

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

INTEGRATING A USABLE SECURITY PROTOCOL FOR USER
AUTHENTICATION INTO THE REQUIREMENTS AND DESIGN PROCESS

DISSERTATION PRESENTED AS PARTIAL FULFILLMENT OF THE
DOCTORATE IN COGNITIVE COMPUTING

BY

CHRISTINA BRAZ

SEPTEMBER 2011

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs (SDU-522 Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, (l'auteur) concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de (son) travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, (l'auteur) autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de (son) travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de (la) part (de l'auteur) à (ses) droits moraux ni à (ses) droits de propriété intellectuelle. Sauf entente contraire, (l'auteur) conserve la liberté de diffuser et de commercialiser ou non ce travail dont (il) possède un exemplaire.

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

L'INTÉGRATION D'UN PROTOCOLE DE SÉCURITÉ ET D'UTILISABILITÉ
POUR L'AUTHENTIFICATION DES UTILISATEURS/TRICES DANS LA
PHASE D'ANALYSE DES BESOINS ET DE CONCEPTION

THÈSE PRÉSENTÉE COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR
CHRISTINA BRAZ

SEPTEMBRE 2011

ACKNOWLEDGMENTS

A doctoral thesis is not a work of a single person; every thesis is inevitably the outcome of many interactions between the author, the author's advisors, peers, colleagues, friends, and family.

My thanks to my thesis advisors Prof. Dr. Pierre Poirier and Prof. Dr. Ahmed Seffah for whom I have a profound admiration. They have worked with me for years to produce this research work. They have shared with me their time, resources, and judgment. They have guided me and helped me improve several sections of my thesis with regards of errors in thinking, purpose and presentation.

Finally, I would like to express my thanks, appreciation and gratitude to my family who have sustained me on this colossal project.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
TABLE OF CONTENTS.....	iii
LIST OF FIGURES	viii
LIST OF TABLES	x
ABSTRACT	xi
RÉSUMÉ	xii
CHAPTER I	
INTRODUCTION	1
1.1 Creating Systems that are Both Secure and Usable	1
1.2 Justification for the Research.....	4
1.2.1 The Challenging Issues - Why Authentication?.....	9
1.2.2 Strong Authentication.....	11
1.2.3 Authentication Methods - Vulnerabilities Still Remain.....	14
1.3 Research Objectives.....	14
1.4 Assumptions and Hypotheses	18
1.5 Human Computer Interaction Security (HCISec).....	21
1.5.1 Conferences in Usable Security	21
1.6 Types of Users.....	22
1.7 Thesis Roadmap	23
CHAPTER II	
REVIEW OF THE STATE OF THE ART.....	25
2.1 Introduction	25
2.2 User Authentication	26
2.2.1 The context of Authentication in Computer Security	28
2.2.2 Elements of User Authentication	32
2.2.3 Architectural Design Patterns in Authentication.....	33
2.2.4 Authentication Factors	35
2.2.5 User Authentication Methods.....	38
2.2.5.1 Passwords and PINs.....	38
2.2.5.2 Authentication Tokens.....	43
2.2.5.2.1 Non-Contact Tokens	44
2.2.5.2.1.1 One-Time Passwords (OTP)	45
2.2.5.2.1.2 C-R Authentication	47
2.2.5.2.2 Contact Tokens	51
2.2.5.3 Digest Access Authentication.....	51
2.2.5.4 Out-Of-Band Authentication (OOBA)	53

2.2.5.5 Risk-Based Authentication (RBA)	54
2.2.5.6 Public Key Authentication.....	56
2.2.5.6.1 Encryption.....	59
2.2.5.6.2 Digital Signatures.....	59
2.2.5.6.3 No Usability Features of PKI.....	59
2.2.5.7 Single Sign-On (SSO)	60
2.2.5.8 Kerberos.....	62
2.2.5.9 Biometrics.....	63
2.2.5.9.1 Fingerprint Recognition	66
2.2.5.9.2 Optical Recognition	69
2.2.5.9.3 Facial Recognition	69
2.2.5.9.4 Voice/Speaker Recognition	70
2.2.5.9.5 Signature Recognition.....	70
2.2.5.9.6 Keystroke Recognition.....	71
2.2.5.9.7 Under-Skin RFID Chip – AuthenLink.....	72
2.2.5.9.8 Biometrics Trade-Offs - Usable Security	72
2.2.6 To Whom Authentication Is Targeted?	73
2.2.7 Comparative Analysis of User Authentication Methods.....	74
2.3 The GOMS Model	81
2.3.1 Engineering Models for Usable Interface Design	82
2.3.2 GOMS: A Method for Cognitive Task Analysis.....	84
2.3.3 How to Develop a GOMS Model?	87
2.3.4 Natural GOMS Language (NGOMSL).....	91
2.3.4.1 Cognitive Complexity Theory	92
2.3.4.2 NGOMSL Steps Development Process.....	95
2.3.5 Learning Time Predictions	95
2.3.6 Execution time predictions.....	97
2.3.7 NGOMSL Methodology.....	98
2.3.8 NGOMSL Limitations.....	99
2.4 Usability and Usable Security	100
2.4.1 Usability Inspection Methods.....	100
2.4.1.1 General usability principles ("heuristics") for User Interface Design....	101
2.4.1.2 Cognitive Walkthrough	107
2.4.1.3 GOMS Model	108
2.4.1.4 Additional Usability Evaluation Methods	110
2.4.2 Usable Security Principles and Guidelines	113
2.4.2.1 Computer Security Design Principles.....	114
2.4.2.2 Design Guidelines For Security Management Systems	117
2.4.2.3 Guidelines and Strategies For Secure Interaction Design	120
2.4.2.4 Design Principles and Patterns for Aligning Security and Usability	121
2.4.2.5 Criteria for Security Software to Be Usable	123
2.4.2.6 Additional Criteria for Security Software to Be Usable.....	124
2.4.2.7 General Security Usability Principles.....	125

CHAPTER III	
THE USABLE SECURITY PROTOCOL.....	128
3.1 Introduction.....	128
3.2 The Usable Security Protocol Methodology	129
3.2.1 Introduction	129
3.2.2 The Usable Security Protocol Architecture and Methodology	132
3.2.2.1 Step 1: Define the mission and conceptual design objective.....	136
3.2.2.2 Step 2: Identify the most representative user authentication methods categories	138
3.2.2.3 Step 3: Develop the NGOMSL Model (Natural Goals, Methods, Selection Language)	139
3.2.2.3.1 Standard Primitive External Operators	140
3.2.2.3.2 Standard Primitive Mental Operators	141
3.2.2.3.3 Analyst-Defined Mental Operators.....	142
3.2.2.3.4 Total Execution Time and Total Learning Time	143
3.2.2.3.5 A Time Level Analysis of the NGOMSL	173
3.2.2.4 Step 4: Develop the Authentication Risk Assessment Matrix.....	176
3.2.2.5 Step 5: Generate the usable security principles	206
3.2.2.6 Step 6: Formulate the Usable Security Symmetry (USS).....	210
3.2.2.7 Step 7: Demonstrate the Usable Security Symmetry (USS).....	210
3.2.3 The Usable Security Protocol Methodology Reuse	211
CHAPTER IV	
THE COGNITIVE SCIENCE AXIS	213
4.1 Introduction.....	213
4.2 Cognitive Ergonomics.....	218
4.2.1 Methods.....	221
4.2.2 The Cognitive Approach	221
4.2.2.1 Gathering information	222
4.2.2.2 Information processing.....	222
4.2.2.3 Process Modeling	222
4.2.2.4 Simulation.....	223
4.3 Main Cognitive Areas of Focus Relating to User Authentication	223
4.3.1 Perception.....	223
4.3.2 Memory	224
4.3.2.1 Encoding.....	225
4.3.2.2 Storage	225
4.3.2.2.1 Sensory Memory	227
4.3.2.2.2 Working Memory (or Short-Term Memory)	227
4.3.2.2.3 Long-Term Memory (LTM)	228
4.3.2.3 Information Retrieval	233
4.3.2.3.1 Recall	233
4.3.2.3.2 Recognition.....	234

4.3.2.4 Password Memorability Issues	235
4.3.2.4.1 Password Policies.....	237
4.3.2.4.2 Varying Systems	237
4.3.3 Mental Models.....	237
4.4 The Cognitive Model of User Authentication (CMUA).....	240
4.4.1 Why to Use a Cognitive Architecture?.....	241
4.4.1.1 GLEAN3 Cognitive Architecture	243
4.4.1.2 SOAR Cognitive Architecture	244
4.4.2 CMUA Cognitive Architecture	246
4.4.2.1 CMUA Components	249
4.4.2.2 CMUA Processing Cycle.....	253
CHAPTER V	
THE COMPUTER SCIENCE AXIS	256
5.1 Introduction	256
5.2 Security as a Usability Characteristic	259
5.3 Usability Factors and Usability Criteria Mapping	261
5.3.1 User Authentication Use Cases	265
5.3.2 Demonstrating the USS using a Multifunction Teller Machine.....	275
5.4 The Usable Security Symmetry (USS) Inspection Method	276
5.4.1 Definition.....	277
5.4.2 Usable Security Protocol (USP) Sub-Methodology	278
5.4.2.1 Usable Security Symmetry (USS) Inspection Method	278
5.4.2.2 Severity Ratings.....	321
5.4.2.2.1 Usability Severity Ratings	322
5.4.2.2.1.1 Usability Severity Ratings and Recommendations for MTM ..	323
5.4.2.2.2 Security Severity Ratings.....	343
5.4.2.2.2.1 Security Severity Ratings and Recommendations for MTM ..	346
5.4.3 Applicability of USS in the AMDLC.....	375
5.5 The Demonstrational Approach.....	376
5.5.1 Demonstration of One-Time-Password (OTP).....	378
5.5.1.1 Wireless Network	378
5.5.1.2 Hardware Token with OTP Functionality	378
5.5.1.3 Personal Identification Number (PIN).....	380
5.5.1.4 Tokencode	381
5.5.1.5 How the OTP Demonstration Works?.....	381
5.5.1.5.1 System Requirements.....	381
5.5.1.5.2 Demonstration Steps	382
5.5.2 One-Time-Password (OTP) Usability Testing.....	387
5.5.2.1 Objectives of the OTP Usability Testing.....	387
5.5.2.2 Testing Tools	387
5.5.2.3 Testing session.....	388
5.5.2.4 Testing Methods – Participant Tasks	388

5.5.2.5 Data Results	389
5.5.2.6 Findings Summary	390
5.5.3 One-Time-Password (OTP) Usability Issues – Discussion.....	390
CHAPTER VI	
CONCLUSIONS AND FUTURE WORK	396
6.1 Summary of the Research Work	396
6.2 Scientific Contributions	397
6.3 Practical Observations on the Impact of USS in Corporate and Academic Environments	400
6.4 Limitations	405
6.5 Future Work and Recommendations.....	405
6.6 The Future of User Authentication	407
REFERENCES.....	409
APPENDIX A	
COMPARATIVE ANALYSIS OF USER AUTHENTICATION METHODS.....	425
A.1 Introduction	425
APPENDIX B	
DATA GATHERING	452
B.1 Introduction	452
APPENDIX C	
CHECKLIST DEVELOPMENT	457
C.1 Introduction	457
APPENDIX D	
USABLE SECURITY PROTOCOL (USP) REUSE METHODOLOGY	460
D.1 Introduction	460
D.2 Design Artifacts for Reuse	460
D.3 Usable Security Symmetry (USS) Inspection Method Quick Setup.....	464
APPENDIX E	
THE GOMS FAMILY: WHICH TECHNIQUE TO USE?	466
E.1 NGOMSL versus CPM-GOMS, KLM, and CMN-GOMS	466
E.2 GOMS Models Comparative Analysis	467
E.3 Why GOMS Model When Compared With Other CTAs?	473
E.3.1 Characterizing the User's Tasks	474
E.3.1.1 Locus of Control	475
E.3.1.2 Goal-Directness.....	475
E.3.1.3 Skill Dimension of Tasks	475
E.3.1.4 Sequential Versus Parallel Activity	476
E.4 Why GOMS?	476
E.4.1 During Design - GOMS Analysis Guiding the Design.....	478

LIST OF FIGURES

1.1	Usability and Security trade-off: A common solution based on a compromise ...	5
1.2	Artifacts with which users might interact with: An authentication token, a wireless device, and a Web interface.....	6
1.3	Percentages of key types of incident	9
1.4	Levels of security.....	13
2.1	AutoFill Code improving usability in user authentication	26
2.2	User authentication in an ideal organization's security framework	28
2.3	The Authentication Marketplace	31
2.4	The elements of the authentication process	33
2.5	Direct Authentication when a client and service share a trust relationship...	34
2.6	Broker Authentication	34
2.7	RSA SecurID® 700 hardware authenticator with One-Time Password (OTP)...	37
2.8	The password authentication process	39
2.9	RSA SecurID® Token for BlackBerry.....	45
2.10	SafeWord RemoteAccess	45
2.11	RSA SecurID® for iPhones.....	45
2.12	DigiPass Go	45
2.13	Authentication via OTP	46
2.14	CAPTCHA: Telling Humans and Computers Apart Automatically	49
2.15	SiteKey helps prevent unauthorized access to users' accounts.....	50
2.16	RSA SecurID® 800 with One-Time Password	51
2.17	Magnetic stripe token	51
2.18	VeriSign Secure StorageToken	51
2.19	Digest Access Authentication's window login in Firefox.....	52
2.20	SafePass Mobile	54
2.21	The RSA® Engine.....	56

2.22	A network authentication method: Kerberos	63
2.23	Biometric authentication process.....	64
2.24	Eikon To Go fingerprint reader	67
2.25	Fingerprint recognition scheme BioEnable	68
2.26	Facelt Argus: facial recognition system from L1Identity	69
2.27	This is a voiceprint for the passphrase: "My voice is my password"	70
2.28	Dynamic signature verification from Cyber SIGN.....	71
2.29	Balancing security and usability	73
2.30	The GOMS Model	81
2.31	Method for selecting a word	88
2.32	The optimal number of usability evaluators: benefits versus costs	105
3.1	The USP architecture and methodology	133
3.2	USP Input, Process, Output.	135
3.3	Hard-coded password Java code snippet	202
3.4	Reusability Pyramid (Gautam & al 2007)	211
4.1	Worksystem and a domain	220
4.2	The three-stage processing model of memory.....	226
4.3	Classification of memory.....	226
4.4	Mental Models.....	238
4.5	A basic cognitive architecture	242
4.6	The GLEAN3 cognitive architecture.....	244
4.7	The SOAR (State Operator And Result) architecture.....	245
4.8	The CMUA cognitive architecture	247
5.1	Authentication Method Development Life Cycle	258
5.2	Security as a usability characteristic.....	260
5.3	Tokencode: a six-digit number	381

LIST OF TABLES

1.1	Percentages of key types of incident	10
2.1	Guide to Understanding Identification and Authentication in Trusted Systems	36
2.2	Do the 8 golden rules of user interface design apply to security systems? ...	41
2.3	The "six password broken rules", and their corresponding usability and security mismatches	41
2.4	Summarized comparative analysis of user authentication methods	79
3.1	Total Execution Time by task scenario	174
3.2	Execution time by task scenario, authentication method type, and method for goal.....	175
3.3	Authentication Risk-Assessment Matrix	180
3.4	Username and Password Guessing	198
5.1	Usability Factors and Usability Criteria Mapping for a MTM.....	264
5.2	User authentication use cases	266
5.3	Usable Security Symmetry inspection method - data visualization	280
5.4	Usable Security Symmetry checklist.....	285
5.5	Example of a usability problem and its severity rate.....	322
5.6	Example of a security problem and its usability criterion and severity rate.....	345

ABSTRACT

Security and usability are both essential in user authentication processes. One of the biggest challenges facing heterogeneous organizations is providing access control systems, to logical as well as physical resources, that are both secure and usable. To achieve this, it is initially necessary to implement three indispensable components such as Identification (*Who does this user claims to be?*), Authentication (*Is this user in fact who s/he claims to be?*), and Authorization (*Is this user authorized to have the resource or service that s/he is requesting?*). Inquiry particularly on user authentication is vital. Without authentication, a computer system often has no foundation for establishing if access should be granted or not.

So far, there has been very little research on usable security of user authentication methods although a considerable body of research work has been made for computer security mechanisms in general other than authentication methods. Therefore a usable security protocol is needed for user authentication.

My thesis is that there is an intrinsic conflict between creating systems that are secure and systems that are usable. But usability and security can be made synergistic by providing requirements and design tools with specific usable security principles earlier in the requirements and design phase. In certain situations it is -possible to concurrently increase usability and security by revisiting design decisions that were made in the past. In other situations it is possible to align security and usability by changing the regulatory environment in which the computers operate. To these ends, this thesis's main goal is not to address usability and security after the product (authentication method) has been manufactured, but to make security a natural outcome of the requirements and design phase of the authentication method development life cycle.

Keywords: user authentication, usability, computer security, access control.

RÉSUMÉ

L'utilisabilité et la sécurité sont des éléments cruciaux dans le processus d'authentification des utilisateurs. L'un des défis majeurs auquel font face les organisations aujourd'hui est d'offrir des systèmes d'accès aux ressources logiques (par exemple, une application informatique) et physiques (par exemple, un bâtiment) qui soient à la fois sécurisées et utilisables. Afin d'atteindre ces objectifs, il faut d'abord mettre en œuvre les trois composantes indispensables que sont l'identification (c.-à-d., définir l'identité d'un utilisateur), l'authentification (c.-à-d., vérifier l'identité d'un utilisateur) et l'autorisation (c.-à-d., accorder des droits d'accès à un utilisateur). Plus particulièrement, la recherche en authentification de l'utilisateur est essentielle. Sans authentification, par exemple, des systèmes informatiques ne sont pas capables de vérifier si un utilisateur demandant l'accès à une ressource possède les droits de le faire. Bien que plusieurs travaux de recherche aient porté sur divers mécanismes de sécurité, très peu de recherches jusqu'à présent ont porté sur l'utilisabilité et la sécurité des *méthodes d'authentification des utilisateurs*. Pour cette raison, il nous paraît nécessaire de développer un protocole d'utilisabilité et de sécurité pour concevoir les méthodes d'authentification des utilisateurs. La thèse centrale de ce travail de recherche soutient qu'il y a un conflit intrinsèque entre la création de systèmes qui soient sécurisés et celle de systèmes qui soient facile d'utilisation. Cependant, l'utilisabilité et la sécurité peuvent être construites de manière synergique en utilisant des outils d'analyse et de conception qui incluent des principes d'utilisabilité et de sécurité dès l'étape d'Analyse et de Conception de la méthode d'authentification. Dans certaines situations il est possible d'améliorer simultanément l'utilisabilité et la sécurité en revisitant les décisions de conception prises *dans le passé*. Dans d'autres cas, il est plus avantageux d'aligner l'utilisabilité et la sécurité en changeant l'environnement régulateur dans lequel les ordinateurs opèrent. Pour cette raison, cette thèse a comme objectif principal non pas d'adresser l'utilisabilité et la sécurité postérieurement à la fabrication du produit final, mais de faire de la sécurité un résultat naturel de l'étape d'Analyse et de Conception du cycle de vie de la méthode d'authentification.

Mots-clé : authentification de l'utilisateur, utilisabilité, sécurité informatique, contrôle d'accès.

CHAPTER I INTRODUCTION

1.1 Creating Systems that are Both Secure and Usable

This chapter describes the research objectives and approach, the justification for the research, the economics of strong user authentication, and the main hypothesis of this research.

All systems demand some form of user account. A user is a single entity whose behavior is solely identified within a computer-based system (i.e. Personal Digital Assistant (PDA), workstation, server login, Web sites, etc.). Individual users classically correspond to individual people, but they might also represent particular system services or resources. Most accounts are protected by an easy keyboard password that even a novice hacker can crack in less than 10 minutes. Once inside, hackers use the attacked account for a diversity of nefarious activities, such as launching distributed denial of service (DOS) attacks, distorting Web sites, stealing billing and credit card information or making counterfeit purchases.

A new report from Penn (2008) Forrester Research shows that security spending is on the rise in some enterprises. The Cambridge, Massachusetts-based research firm interviewed practically 1,000 firms for its State of Enterprise IT Security: 2008-2009 report and found that the security segment of Information Technology (IT) budgets is expected to increase 12.6 percent in 2009, up from 7.2 percent in 2007 and 11.7 percent in 2008. As a matter of fact even during difficult economic conditions, IT security remains an essential portion of business operations as enterprises try to preserve their current environment as well as plans for the implementation of novel initiatives. Security is getting a bigger portion of the IT pie, with the focus less on reactive vulnerability defenses and more on looking at what is required to protect businesses. The focus now is more on protecting the data itself which means information security.

Distribution of budget for new security initiatives, information security, has increased from 17.7 percent in 2008 to 18.5 percent in 2009. There has been a major shift from what was the broadly recognized state of security just a few years ago. Protecting the organization's information assets is the top concern facing security programs: data security (90 percent) is most frequently mentioned as a vital concern for IT security organizations, followed by application security (86 percent), and business continuity/disaster recovery (84 percent). Data security as well tops the list of business objectives for security, with 89 percent mentioning protection of corporate data and 87 percent mentioning protection of personal data as essential business objectives. Most of this 2008 spending on information security countermeasures has purchased confidentiality and integrity solutions: products like firewalls to protect the information perimeter of an enterprise (or encryption), Virtual Private Networks (VPNs)¹, and anti-virus and intrusion detection to safeguard the actual information. Spending on user authentication products to safely identify users has followed in the security market. Inefficient user authentication marginalizes perimeter security and access controls, showing vulnerabilities in the confidentiality and integrity areas. The growing trend toward identity theft, or employing stolen names, birthdays and identification numbers to perpetrate fraud, would meet firm resistance if strong authentication practices were universally employed. Privacy violations take place as well due to compromised user authentication. Authentication is behind confidentiality and integrity because exactly identifying huge numbers of users has proven a costly and overwhelming task.

The central research question of this thesis is the following:

How is it possible to ensure usability of user authentication without compromising security and vice-versa?

Security and usability are both essential in the authentication process. It is broadly held that security and usability are two opposing goals in system design

¹ A VPN is a network that uses a public telecommunication infrastructure, such as the Internet, to provide remote offices or individual users with secure access to their organization's network.

(Cranor and Garfinkel, 2005; Jøsang *et al.*, 2007; Nielsen, 2000) but there are several cases in which security and usability can be synergistically enhanced by reviewing the usable security approach. In addition, the human portion of computer security is effortlessly exploited and continually overlooked. Companies spend millions of dollars on firewalls, encryption and secure access devices, but most of the time they forget to address issues related to the weakest link in the security chain: the human being.

In considering the extent that users are important in the authentication process, a company's goal is to select an Authentication Method (AM) that is suitable to the risk involved and as easy to use as possible. Applying too low a level of security might compromise the integrity of the company's process. But applying too high a level for a low-risk process means the process will be too hard and will confront low adoption rates. As stated by Penn (2008), the key criteria when assessing such solutions are ease of use, portability, cost, security, manageability, and cross-channel utility.

My thesis is that there is an intrinsic conflict between creating systems that are secure and systems that are usable. But usability and security can be made synergistic by providing requirements and design tools with specific usability and security principles earlier in the requirements and design phase. In certain situations it is possible to concurrently increase usability and security by revisiting design decisions that were made in the past. In other situations it is possible to align security and usability by changing the regulatory environment in which the computers operate. To these ends, this thesis's goal is not to address usability and security after the product (authentication method) has been manufactured, but to make security a natural outcome of the requirements and design phase of the authentication method development life cycle.

1.2 Justification for the Research

Security and usability are both essential in user authentication processes. One of the biggest challenges facing heterogeneous organizations is providing access control systems, to logical as well as physical resources, that are both secure and usable. To achieve this, it is initially necessary to implement three indispensable components such as Identification (*Who does this user claims to be?*), Authentication (*Is this user in fact who s/he claims to be?*), and Authorization (*Is this user authorized to have the resource or service that s/he is requesting?*). Inquiry particularly on user authentication is vital. Without authentication, a computer system often has no foundation for establishing if access should be granted or not.

Furthermore, the majority of contemporary computer users for example need to authenticate to a company network several times during their work day. Another particular concern in authentication according to Cranor and Garfinkel (2005) is that authentication systems *do not fail gracefully*. It means that if an average consumer computer user forgets her username but gets right the password the system does not enable her partial access to an online magazine, for instance, or for an average corporate computer user access to the system's less important files, or an emergency or temporary access. However there are a few companies that are in the initial stages of implementing some of these “fail gracefully” functionalities² in the corporate area. There is no established and recognized mechanism to accommodate user error, which means that most likely the productivity will be strongly compromised and the user's dissatisfaction with the system will be high. Figure 1.1 models the relationship between usability and security.

General principles for User Interface Design (UID) have already been well recognized in the Human Computer Interaction (HCI) field. These general principles

² Users who have lost their hardware authenticator, for instance, can still log in to their accounts with an emergency access code through the on-demand authentication method, without having to contact the system administrator. RSA SecurID® On-demand (SMS) Authenticator. RSA - The Security Division of EMC <<http://www.rsa.com/node.aspx?id=3481>>

are called "heuristics" because they are more in the nature of rules of thumb than specific usability guidelines (Molich and Nielsen, 1990) (e.g. "User Control and Freedom" is one of these principles).

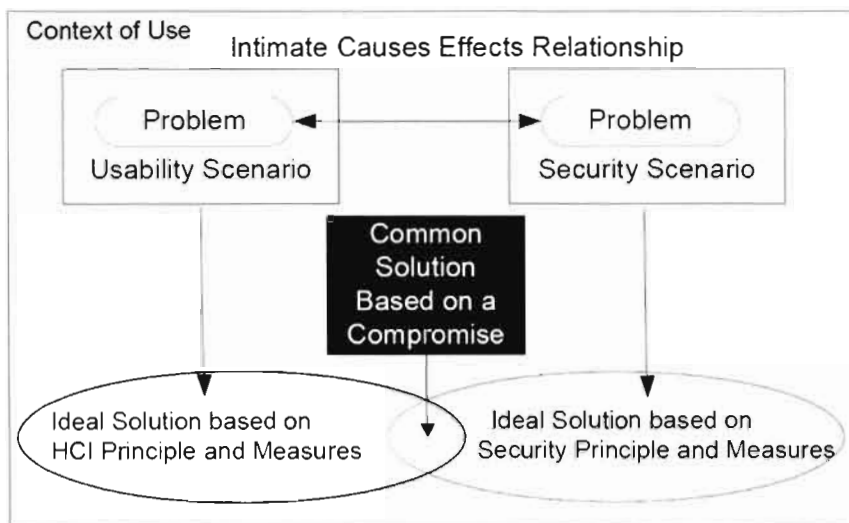


Figure 1.1: Usability and Security Trade-off: A Common Solution Based on a Compromise (Braz and Seffah, 2007).

So far, there has been very little research on the security usability of user authentication methods, although a considerable body of research work in usable *security* (a terminology adopted in this thesis when referring to security and usability) has been made for computer security mechanisms in general other than authentication methods. Therefore a usable security protocol is needed for user authentication.

This thesis defines the concept of Usable Security as the study of how security information and usability factors should be handled in either front-end or back-end user authentication processes, taking into consideration resources and costs. But why take front-end/back-end processes into consideration? Graphical User Interface (GUI) Developers should have knowledge of user interface design, and tools for implementing designs correctly. This knowledge will result in better front-end design, minimize the number of bugs in the software, and result in lower development costs per feature. GUI Developers should be assigned responsibility for accurate

implementation of front-end design, as well as back-end functionality. GUI Developers should understand interface issues sufficiently well to know when to raise design issues during implementation, rather than disregarding them or implementing them inaccurately.

It is crucial to note that the term *-interface-* in this thesis is not only related to GUI, but also to a shared limit through which the information flows (Maffezzini, 2006). It consists of a hardware or software component that makes the junction between the interface and the user with the purpose of transiting information between them (e.g. an OTP token is an interface between the authentication server and the user) (Figure 1.2).

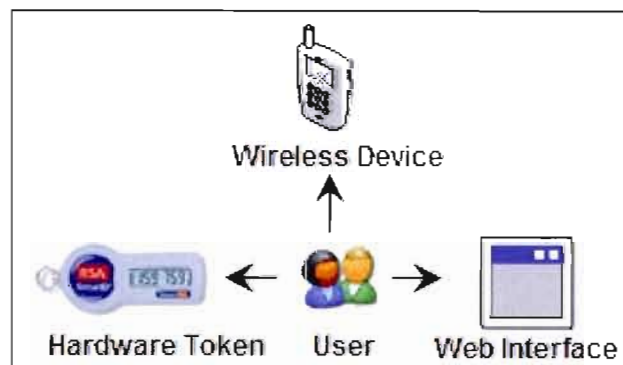


Figure 1.2: Objects with which users might interact: An authentication token³, a wireless device, and a Web interface.

According to Sasse (2004), one of the most recognized researchers in usable security-, “Don’t focus on UIs to security tools - the big problems are in security requirements, job design and user involvement.” That is exactly what this thesis is all about: requirements and design. Additionally, according to Whitten and Tygar (1999), most of the research in HCI Sec focuses on providing better UIs, but it is obvious that usability problems with secure systems are more than only UIs and need application of HCI factors and design methodology. Whitten and Tygar (1999) claim that using conventional methods for usability evaluation that concentrate on the

³ <http://www.rsa.com/node.aspx?id=3049>

impact of usability on security effectiveness will assist developers to discover usability problems that threaten security. Both analytical and empirical evaluations were performed in testing the usability goals of Pretty Good Privacy (PGP) Desktop E-mail software (i.e. public key encryption software for desktops and laptops) (Whitten and Tygar, 1998). A number of usability problems causing security failures were discovered in the study, providing the foundation in the Whitten and Tygar (1999) study that specific usability goals are needed for usability evaluation of security mechanisms. It is important to note that the PGP software has been cited throughout this dissertation as an example of public key authentication. This is due to the fact that PGP is one of the most common solutions for email encryption, supports major email security standards, and interoperates with most accepted email security software solutions.

The value of usable security was pointed out as early as 1883 by the Belgian cryptographer and linguist Auguste Kerckhoffs in two articles on cryptography. Kerckhoffs is most famous for establishing the principle that security should not be based on obscurity. Moreover, four out of six of Kerckhoffs' cipher principles of design (Kerckhoffs, 1883) are related to usable security (3 to 6) in bold) as follows:

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concurrence of several people;
6. Finally, it is necessary, given the circumstances that command its application that the system should be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

The initial data gathered on usable security related to user authentication methods are basically research regarding an evaluation of Pretty Good Privacy (PGP) (Whitten and Tygar, 1998), a public key encryption program primarily intended for authentication and email privacy, anti-phishing authentication mechanisms (Dhamija *et al.*, 2006; Dhamija and Tygar, 2005), security toolbars (Wu *et al.*, 2006), user authentication mechanisms (pictorial passwords) (Angeli *et al.*, 2003), security user studies (Chiasson and Biddle, 2007), secure User Interface (UI) for network applications (i.e. authentication of the communication) (Jøsang and Patton, 2003), design principles and patterns for computer systems that are secure and usable (Cranor and Garfinkel, 2005), and some general white papers about user authentication. Although, Human Computer Interaction-Security (HCI-Sec) researchers have been applying HCI techniques in security software on a very small scale, there are no methods or techniques to effectively design secure and usable user authentication systems yet. This is area which this thesis is going to explore: integrating usable security in the requirements and design phase.

Moreover, authentication services are critical to authorization and auditing services. If users' identities are not appropriately authenticated, an organization has no guarantee that access to resources and services is correctly monitored. Regardless of how well controlled a company's authorization services are, everything stems from the exact identity of the users. Also, correspondingly, without accurately authenticated identities, audit trails, though complete and well monitored, will be untrustworthy and give no accountability (e.g., a forged user ID could be linked to auditing actions).

On October 12, 2005, the Federal Financial Institutions Examination Council (FFIEC) issued the updated guidance, "Authentication in an Internet Banking Environment." FFIEC requires that financial institutions provide consumers of online financial services with the same security protection enjoyed by customers buying groceries or gas with a debit card: strong authentication. The -FFIEC (2005) guidance states the following: "Single-factor authentication, as the only control mechanism, to

be inadequate for high-risk transactions involving access to customer information or the movement of funds to other parties. The authentication techniques employed by the financial institution should be appropriate to the risks associated with those products and services. Account fraud and identity theft are frequently the result of single-factor (e.g., ID/password) authentication exploitation."

1.2.1 The Challenging Issues - Why Authentication?

Without a proper user authentication system (the "door-entry" of any system), organizations are susceptible to potential attackers who can compromise the whole organization's computer and network system, and consequently undermine its infrastructure and assets as well. For example, the -CSI/FBI-- Computer Crime and Security Survey (2008) defined 13 types of attacks or computer mishandling resulting in direct financial loss to the survey's participants (Table 1.1). The survey asks about a number of different sorts of computer attacks and incidents. The areas marked with red squares in Table 1.1 highlight the type of incidents related to authentication. A significant percentage (44%) is responsible for attacks coming from inside an organization. In Figure 1.3, a subset of the mentioned attacks is graphed, with data stretching back to 1999 by percentages of key types of incidents. In Particular, this chart illustrates the four categories of highest incidence: viruses, insider abuse, laptop theft, and unauthorized access to systems.

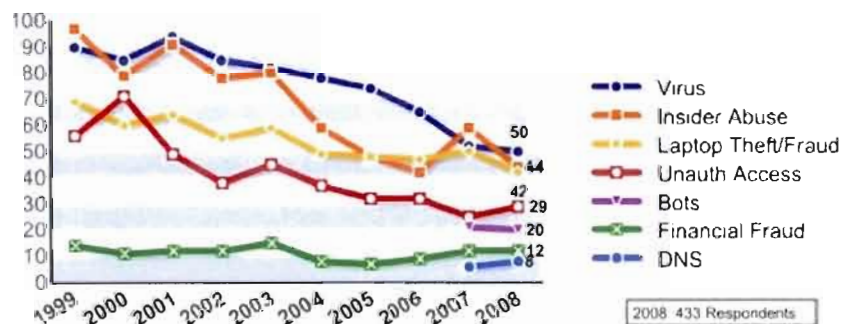


Figure 1.3: Percentages of key types of incident (CSI/FBI, 2008).

	2004	2005	2006	2007	2008
Denial of service	39%	32%	25%	25%	21%
Laptop theft	49%	48%	47%	50%	42%
Telecom fraud	10%	10%	8%	5%	5%
Unauthorized access	37%	32%	32%	25%	29%
Virus	78%	74%	65%	52%	50%
Financial fraud	8%	7%	9%	12%	12%
Insider abuse	59%	48%	42%	59%	44%
System penetration	17%	14%	15%	13%	13%
Sabotage	5%	2%	3%	4%	2%
Theft/loss of proprietary info	10%	9%	9%	8%	9%
from mobile devices					4%
from all other sources					5%
Abuse of wireless network	15%	16%	14%	17%	14%
Web site defacement	7%	5%	6%	10%	6%
Misuse of Web application	10%	5%	6%	9%	11%
Bots				21%	20%
DNS attacks				6%	8%
Instant messaging abuse				25%	21%
Password sniffing				10%	9%
Theft/loss of customer data				17%	17%
from mobile devices					8%
from all other sources					8%

Table 1.1: Percentages of key types of incident (CSI/FBI, 2008).

Virus incidence fell below insider abuse last year, but regained its position of the most common occurrence this year. That said, both categories dropped compared to last year, and actually all four of the most widespread types of incidents fell. There seems to be an obvious trend of lower and lower percentages of incidence being reported in these categories over the past several years. Table 1.1 also shows that only four categories showed to some extent increased percentages.

In the real-world, organizations struggle to enforce security policies—even the most basic ones (e.g. password). When a user has unsupervised physical access to a

mobile device, for example, he can usually do whatever he wants with it, even authenticate himself through the software token installed in the mobile device since he knows his friend's username and password. As a result, most of these policies violate the Big Stick principle: *Whoever has physical access to the device is allowed to take it over* (Stajano, 2003) (as in the previous example). These policies are extremely hard to enforce and thus scarcely of practical usage. The Big Stick Principle is a very high level security policy model which identifies a set of cases in which authentication is superfluous.

In the Web area, it is worth noting that 5 out of the top 10 Web application security vulnerabilities are directly or indirectly related to authentication according to OWASP (2009).

1.2.2 Strong Authentication

Strong authentication relates to systems that entail rigorous user identity verification, which is accomplished through multiple factors for authentication. It allows us to irreversibly determine the user's identity or the integrity of precise data. Strong authentication also presumes that access to a network is extremely hard to break, thus creating a secure network. The goal of strong authentication is to strengthen the security by replacing the classic authentication method of password for a software-only authentication solution with dynamic password generators, or software-hardware authenticators like smart cards, tokens, biometrics, etc. Traditional authentication assumes we know something: the user and the password. In contrast, strong authentication presupposes we know the username and the password, but also employs something that will generate the password, like password generators. A password generator offers the user the choice to allow the system to assign passwords to usernames and logins. Password generators use an amalgamation of case sensitive letters, numbers, and symbols mathematically generated to offer the user with the strongest, hardest to hack passwords.

A single factor authentication is not secure. Actual information security requires an amalgamation of mechanisms (i.e. multi-factor user authentication) to verify who the user is, what the user knows, what the user has, or where the user is. Verifying who the user is typically requires a Personal Identification Number (PIN) to attest what the user knows. The PIN combined with a biometric method, such as a fingerprint or iris scan, attests what the user has., or a smart card or digital certificate also assures what the user has and a Global Positioning Satellite (GPS) receiver (e.g., a Blackberry with a GPS application installed) corroborates- where the user is.

Combining multiple user authentication methods generates almost infallible user authentication on the Web, just as multiple levels of identification provide security for the physical access control. For example, a user who enters the top secret area of a military building might be asked to present two pieces of identification which is information known only to the user, match a fingerprint, and finally type in the combination for an electronic door lock. Once inside, the user still has to log onto the computer. Multi-factor user authentication such as this has been employed for a long time in physical world security systems.

There are currently several authentication technologies to select from, and they each verify the identity of a user and grant access to resources. Nevertheless, they essentially diverge in the level of security they offer as shown in Figure 1.4. While passwords are usually considered weak forms of authentication, token and especially Biometrics have been established as much stronger forms of authentication.

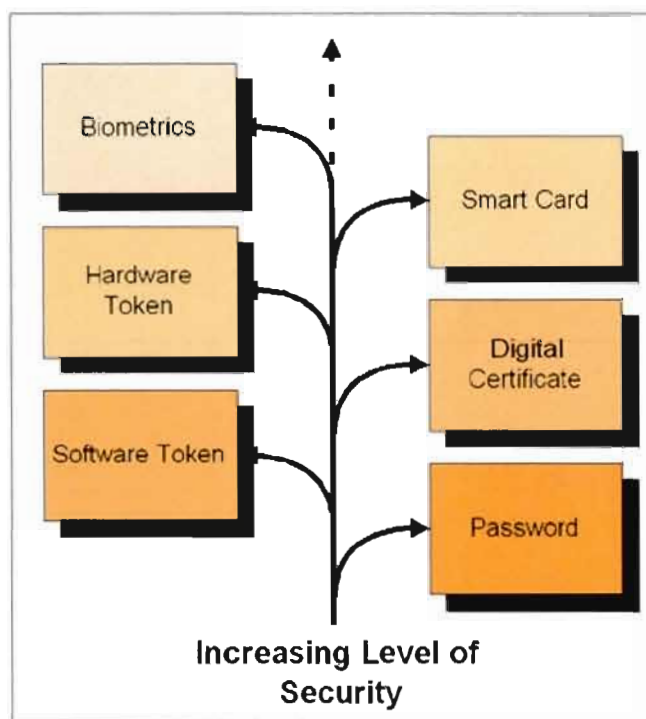


Figure 1.4: Levels of security (I/O Software, 2005).

Users frequently and understandably resist strong authentication because it adds additional steps to their login and Internet sessions. Once they are authenticated, users' identities are securely established. As expected, corporate users are more receptive to strong user authentication, especially since it is intrinsic to their jobs. Generally speaking consumers have shown more resistance to additional or intrusive steps that eliminate anonymity. Many security experts foresee equivalent trends toward stricter user authentication for Internet consumers as e-commerce continues to increase and an increase in novel kinds of services that require strong authentication in the market.

Authentication policies are required to manage how the authentication methods interoperate. These policies orchestrate user authentication methods, such as the methods to employ for specific resources, the order in which to employ them, and the back-up activities to be carried out should the selected methods fail. Developing user

authentication policies typically requires the expertise of highly skilled security system designers to put -the system into operation on- a long-term basis. Automated user authentication management systems are only in their infancy to ease the human-intensive effort usually associated with deploying and operating strong authentication. However, that same automation on one end- pushes the burden to the other end of the chain, which is the end-user.

The greatest challenge of strong authentication is to make fraud more difficult for an attacker while respecting the constraints associated with an application: the technical, economical, and organizational environment (Braz and Aïmeur, 2005).

1.2.3 Authentication Methods - Vulnerabilities Still Remain

Despite the efforts that have been made by organizations to provide suitable authentication methods, vulnerabilities still remain. Mechanisms and models that are complicated to the user will be misused. When an authentication method is too demanding the user might not keep up with the increasing workload (e.g. a user might refuse to sign up to a Web site because she cannot cope with the strong authentication method). Thus, organizations often tend to blame the users for the human failure of not handling complex and demanding technical systems. However, Norman (1988) argues that what we often view as human error is the result of design flaws that may be surmounted. According to the Computing Technology Industry Association (CompTIA, 2002), human errors turn out to be one of the major causes of security breaches in organizations; they account for 84% of security breaches in 900 private and public American organizations.

1.3 Research Objectives

There is a common sense among security professionals that it is crucial to find ways of designing secure systems that individuals can use, including user authentication systems. However, there is still less agreement on how to achieve this

goal (Adams and Sasse, 1999; Whitten and Tygar, 1999; Garfinkel, 2005; Sasse and Flechais, 2005; Yee, 2005).

The main goal of the research presented in this dissertation is to investigate the relationship between usability and security and address the problem of usable security in user authentication methods in the field of computer security. This dissertation has the following objectives:

- Find the right trade-off between Security and Usability: The design of usable yet secure systems raises crucial questions when it comes to properly balancing security and usability. Finding the right tradeoff between these two important attributes is not an easy endeavor. Usability problems in these systems can lead to security vulnerabilities which can consequently impact a company's bottom line. One of the difficulties in developing human interfaces to security systems is anticipating the response of users to the huge space of possible system states and design options (Harris, 2007). It would be useful to have a computational representation of the user that would allow the designer to simulate user responses to a diversity of situations and design options. Although the HCISec field is far from having a complete model of the user, a first step has been taken in this research with the development of a Usable Security Protocol (USP) for user authentication by taking into consideration the cognitive as well as computer science aspects.
- Provide practical and specific guidelines on the design of authentication methods to support usable security: The HCISec research community has gradually been developing a good body of work in usable security, but this consists primarily of general guidelines (See Chapter 2, Review of the State of the Art, for more details.) These do not provide a method for pointing out and solving specific design problems (e.g., for one-factor authentication like a password, have strict password policies been established?). Contrary to general design guidelines, which are mostly descriptive- and simply specify

“nice to have” general design features, this thesis adopts a constructive approach by specifying how a problem can be solved. The bottom line is that companies don’t base their security decisions on “general solutions”.

- Develop a robust design tool for the development of an authentication method: Even if a project is small and the requirements are simple, there is still a design process that occurs between understanding the requirements and starting to construct. Design becomes progressively important as the project becomes larger and more complex, which is usually the case for the development of security products and mechanisms. The Requirements and Design phase is an important prelude to extracting and gathering the requirements and especially because it defines the problem that the stakeholder is trying to solve, no matter what model of software development process is adopted (e.g., waterfall, iterative, etc.). It is broadly held that gathering and agreeing on requirements and design is crucial in the whole *development* process and also important to the overall success of any project (Perks, 2003). Therefore a robust design tool is recommended to influence an authentication method’s reliable design and bring security and usability together earlier in its life cycle.
- Establish a solid *ground basis* for developing a Usable Security Symmetry Web-based application (USSWebApp): The research work in this thesis represents exactly what should be done when following best practices for software development, which is starting with a robust and well defined Requirements and Design phase prior to going through the Implementation, Deployment, Testing, and Evaluation phases.
- Publish research papers in conferences and journals:
 - Published - Braz, C., Poirier, P., & Seffah, A.: 2010. Designing Usable, Yet Secure User Authentication Service: The Cognitive

Dimension. e-Review of Tourism Research (eRTR), Web-based international research network for tourism professionals.

- Published - Braz, C., Seffah, A. (1) & Poirier, P. (2): 2009. User Authentication: Adding Usable Security Symmetry into Design and Requirements. First International Workshop on Software Security Process (SSP09) .in conjunction with the IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT 09), August 29-31, Vancouver (Canada); (1) Department of Computer Science, Concordia University; (2) Department of Philosophy, University of Quebec at Montreal.
- Published - Braz, C., Seffah, A. (1) & M'Raihi, D. (2): 2007. Designing a Trade-Off between Security and Usability: A Metrics - Based Approach. INTERACT 2007 Socially-Responsible Interaction The Eleventh IFIP TC13 International Conference on Human Computer Interaction, 10-14 September 2007, Rio de Janeiro, RJ (Brazil); (1) Department of Computer Science, Concordia University; (2) Principal Scientist, Innovation Group, VeriSign, Inc.
- Published - Braz, C. & Robert, J.M.: 2006. Security Usability: The Case of User Authentication Methods. In 18th French-Speaking Conference on Human Computer Interaction (HCI2006), École Polytechnique de Montréal, 18-21 April, Montreal, Quebec (Canada).
- Published - Braz, C. & Aïmeur, E. (1): 2005. ASEMCM: Authentication for a Secure Mobile Commerce. RFID Journal, RFID White Papers, Security White Papers (June 2005); (1) University of Montreal.
- Published - Braz, C.: 2004. AUTHENLINK: A User-Centered Authentication System for a Secure Mobile Commerce. In ERGO-IA 2004, November 17-29, 2004, Biarritz (France).
- Published - Braz, C. AUTHENLINK: 2004 A User-Centered Authentication System for a Secure Mobile Commerce. In

Proceedings of the 1st French-Speaking Conference on Mobility and Ubiquity Computing UBIMOB 2004, June 1-3, 2004, Nice, Sophia-Antipolis (France).

- Published - Braz, C. & Aimeur, E. AUTHENLINK: 2004. Authentication System for a Secure Mobile. In 3rd International Workshop on Wireless Information Systems (WIS-2004), April 13-14, 2004, Porto (Portugal).

1.4 Assumptions and Hypotheses

This thesis project assumes the following hypotheses:

- Usable security is critical to the effective adoption and deployment of user authentication methods. As a matter of fact there is no set of recognized usable security standards particularly targeted to user authentication methods but rather only to security mechanisms in general. As expected, there are numerous examples that fully characterize this hypothesis such as the so-called password complexity, locking Personal Identification Number (PIN) systems, cumbersome data input of challenge-response calculators, lack of usability in security software, “negative redundancy” (this term has been coined by this research) of biometrics systems when users are authenticating to a system (e.g., combine a username/PIN with fingerprint), and so on. Moreover, to reduce management and support costs, organizations are placing more and more of the burden of authentication on the user (i.e. key stakeholders like employees, partners, end-users, etc.), forcing them to perform - at the enterprise’s discretion - lifecycle-management tasks (i.e. self-service user authentication) such as token activation, password replacement, and certificate renewal.
- The development of a user authentication method, irrespective of being a software or hardware authenticator, should include usable security measures

at an early stage as part of the Requirements and Design phases in a trusted security framework of an organization.

- A user authentication method might be adapted to a computer system's infrastructure already in place in an organization. In some cases the specification of an authentication method may be infrastructure-dependent on a computer system within an organization since authentication should be integrated into an existing security infrastructure (i.e. post-implementation).
- The choice of an authentication method depends also on industry norms and well as on legal and business needs such as the environmental characteristics of the electronic communication (e.g. online shopping). These needs basically prescribe the computer requirements of an organization, putting in place controls on processes and technical infrastructure.
- Security designers address the diverse authentication needs of several different users, including system administrators, employees, business partners, customers, and end-users. For example, System Administrators might require a strong Public Key Infrastructure (PKI) solution. Employees who want to mix physical access security with strong network authentication might require a smart card, aggregating use of a One-Time Password (OTP), PKI, and building-access credentials. External users such as remote Virtual Private Network (VPN) users, business partners, and customers may prefer clientless OTP tokens (VeriSign, 2010). Users who conduct high-value transactions and need non-repudiation capabilities may require a hybrid authentication platform that can combine OTP and PKI VeriSign functionality such as the RSA SecurID® 800 hybrid authenticator⁴ (i.e. all in-one-devices). Lastly,

⁴ RSA hybrid authenticators: The RSA SecurID® 800 authenticator is a hybrid device that secures the end-user environment by combining features of the RSA SecurID hardware. authenticator with a smart chip - all in a single USB form factor. RSA-The Security Division of EMC. May 23, 2010 <<http://www.rsa.com/node.aspx?id=1215>>

organizations setting large-scale consumer applications may prefer consumer scratch cards over more expensive electronic devices for authentication.

- Security tools (including authenticators) have been developed, but their successful use in real applications is fairly limited because of their complexity, “hard-of-use”, and the necessity of previous advanced technical knowledge on the part of end-users. The “hard-of-use” restrains not just novice users but also the average corporate and consumer computer user. This originates from the fact that several steps and parameters have to be set on those security applications so that security services, which range from security policy development (i.e. authentication included) to intrusion detection support, can be properly executed. However such configuration complexity leads to unwelcome circumstances in which some users are ready to give up security to meet their project deadlines or to attain higher system performance. For example “maintaining the secrecy of authentication keys is a particular problem because humans are famously the weak link in information security. People trust each other and will sometimes disclose classified information upon request” (Renaud, 2003). Even when people are security conscious enough not to disclose their authentication key, they will often write down codes they should be memorizing. They will do this in self-defense if the codes change too often or if they have too many. “These weaknesses are caused by the human factor in security, and no authentication mechanism can succeed in meeting its dual roles of permission and prevention until the human factors are taken into account” (Mitnick and Simon, 2002). There is an increasing understanding of the user’s role in the security of any system, as just one of many links of a chain which can be considered to surround and secure the system. One way to make the user link stronger is to consider essential factors such as the user’s needs, abilities, inclinations and skills in formulating security mechanisms and policies (Renaud, 2003).

Therefore security architects and designers who are in charge of selecting, implementing, and managing IT security services for an organization - cautiously assess their options before selecting resources that will be delegated to meet their specific IT security program requirements. Development of methods and techniques to diminish complexity in usage of security services is therefore required.

1.5 Human Computer Interaction Security (HCISec)

Human Computer Interaction Security (HCISec) is the field of research that studies how humans interact with computers, especially information security. It aims, in plain terms, to improve the usability of security features in end user applications. Traditionally, security features demonstrate poor usability for the following reasons:

- Security features are usually added in casual afterthought;
- Security features are quickly patched in to deal with newly revealed security bugs;
- Security features deal with very complex use cases;
- Interface designers usually lack understanding of security concepts; and
- Interface designers are not often usability experts but application developers.

1.5.1 Conferences in Usable Security

Since 2004, conferences in Usable Security have been held and have steadily been gaining attention within the HCISec research community and Computer Security industry as follows:

- Computer Human Interaction (CHI) 2003 - Workshop on HCI & Security Systems: <<http://www.andrewpatrick.ca/CHI2003/HCISEC/>>
- DIMACS Workshop on Usable Privacy and Security Software: <<http://dimacs.rutgers.edu/Workshops/Tools/>>

- SOUPS Symposium On Usable Privacy and Security:
<<http://cups.cs.cmu.edu/soups/2009/>>
- Usable Security (USEC):
<<http://usablesecurity.org/>>

1.6 Types of Users

This research includes five different types of users related to Computer and Web Security areas. Each type of user has different attributes and responsibilities. A user can be defined as described below:

- Super Administrator (Super Admin): An Information Technology (IT) professional who acts as the chief administrator, for instance for the authentication manager software. This person is presumed to be highly skilled, commensurate with the type of IT professional who would be assigned to administer a mission critical application. The Super Admin has all permissions to configure and administrate the system and other administrators, and is typically the person who would be responsible for planning, deploying, and configuring the software. The Super Admin can also act as an Approver or Distributor. The Super Admin may grant administrative permission to approve credential manager requests from and distribute tokens to End-Users.
- Domain Administrator: Generally speaking, each domain has its own Administrator, or a Domain Administrator can look after a number of domains. The role of the Domain Administrator is to configure and maintain the authentication manager software for the portion of the enterprise for which they are responsible. They can manage, for example, objects such as users, users groups, tokens, and password policy. The Domain Administrator can also act as an Approver or Distributor.

- **Help Desk Administrator:** Like the Super Admin, the Help Desk Admin is an IT professional who acts as an administrator for the mentioned software. This person is assumed to be of reasonable skill, commensurate with the type of professional who would be assigned to administer parts of a mission critical application. A Help-Desk Admin is a person who provides first-tier or second-tier help desk technical troubleshooting support for end users.
- **Developer:** The Developer is a person who designs and writes software.
- **Customer:** The customer is the buyer of the authenticator (i.e., authentication method solution). The Customer should not be confused with the End-User.
- **End-User:** A person who will ultimately use a software or hardware authenticator to enable her/him to perform a job function (e.g. an individual employs a hardware token to authenticate to a corporate network). The End-User has knowledge of basic Web browsing with typical technical expertise or some previous training in the use of computer interfaces, and can be considered -an average corporate or/consumer computer user.

1.7 Thesis Roadmap

This thesis contains six (6) Chapters and four (4) Appendices.

Chapter 1: Introduction

Justification for the research, Research objectives, Assumptions and hypotheses, Human Computer Interaction Security (HCISec), Author's Publications in Usable Security, Types of users, and Thesis roadmap.

Chapter 2: Review of the State of the Art

The context of Authentication in Computer Security, Elements of User Authentication, User Authentication Methods, To Whom Authentication Is Targeted? Comparative Analysis of User Authentication Methods, Usability and Usable Security Principles and Guidelines.

Chapter 3: The Usable Security Protocol

The Usable Security Protocol Methodology, Step 1: Define the mission and conceptual design objective, Step 2: Identify the most representative user authentication methods categories, Step 3: Develop the NGOMSL Model (Natural Goals, Methods, Selection Language), Step 4: Develop the Authentication Risk Assessment Matrix, Step 5: Generate the usable security principles, Step 6: Formulate the Usable Security Symmetry (USS) inspection method, and Step 7: Demonstrate the USS, and The Usable Security Protocol Methodology Reuse.

Chapter 4: The Cognitive Science Axis

Cognitive Ergonomics, Main Cognitive Areas of Focus Relating to User Authentication, and the Cognitive Model of User Authentication (CMUA).

Chapter 5: The Computer Science Axis

Security as a Usability Characteristic, User Authentication Use Cases, Usability Factors and Usability Criteria, The USS Inspection Method, Demonstrating USS using A Multifunction Teller Machine (MTM), Usability Severity Ratings, Security Severity Ratings, and One-Time-Password (OTP) demonstration.

Chapter 6: Conclusions and Future Work

Summary of the Research Work, Scientific Contributions, Practical Observations on the Impact of USS in Corporate and Academic Environments, Limitations, and Future Work and Recommendations, and The Future of User Authentication.

Summary of the topics discussed in Chapter 1: Introduction.

In this Chapter, the Justification for the research has been presented as well as the challenging issues in authentication (Strong authentication and vulnerabilities in user authentication methods), Research objectives, Assumptions and hypotheses, the Human Computer Interaction Security (HCISec), and Types of users.

CHAPTER II

REVIEW OF THE STATE OF THE ART

2.1 Introduction

This chapter presents a review of the state of the art in User Authentication, Usability inspection methods, Usable Security principles and guidelines, and the GOMS model.

Security systems are conceived to allow authorized users in, and to keep unauthorized users out of an organization's network resources. In addition, the security system needs to make sure that users only perform actions they are authorized to perform. To this end, user authentication is the entry point to different computing networks or facilities in which a set of services are rendered to users or a set of tasks can be performed. For example, once successfully authenticated, the user can gain access to a company's Intranet, databases, applications, facilities, etc.

Usability of the authentication mechanisms has seldom been investigated, and since security mechanisms are conceived, implemented, put into practice and violated by people, human factors should be taken into account in their design (Adam and Sasse, 1999). For example, Social Engineering attacks precisely target the human link, and represent a very effective attack vector. A reformed and world-famous controversial computer hacker Kevin Mitnick found that he never had to crack passwords by technical means because he could constantly get them from people.

The GOMS model, first proposed by Card *et al* (1983), is the general term for a family of human information processing techniques that attempt to model and predict user behavior.

2.2 User Authentication

Usability becomes a strategic issue in the implementation of user authentication methods in organizations. Usability can be defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (ISO9241-11:98). Usable Security is concerned with the study of how security information and ease-of-use should be handled in the user interface.

Authentication is an enabling task that needs to be completed to get to the resources required to do real work (Sasse *et al*, 2001). The function of user authentication should not be to educate users to better manage their security (authentication) issues, nor should it be so difficult to use that it requires either mental strain or the knowledge of a long series of rules to observe (Kerckhoffs, 1883). Instead, it should follow the least (if not zero) user interaction principle, meaning that authentication procedures are unobtrusive involving almost no user input (or none at all) and are intuitive, helping users to authenticate themselves. Good examples of the latter are the "zero" interaction authentication (Corner and Noble, 2002), and the RSA Security Toolbar Token with an auto fill code feature that drastically reduces user interaction (Figure 2.1).



Figure 2.1: AutoFill Code⁵ improving usability in user authentication.

Identification, Authentication, and Authorization are distinct and necessary components-that allow users to securely access a computer system. Authentication is the process of establishing whether someone is who s/he declares her/himself to be. In private and public computer networks (encompassing the Internet), authentication is popularly done through the use of logon passwords. The logon is the process used

⁵ Auto fill code. RSA, The Security Division of EMC <<http://www.rsa.com/node.aspx?id=3031>>

to gain access to an operating system or application, generally from a remote computer. Usually a logon requires that the user have a user ID (username) and a password. Authentication is one of the critical elements of a set of services that constitute a security sub-system in a communications infrastructure and encompasses the following security services:

- *Authentication*: The verification of a claimed identity.
- *Confidentiality*: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes.
- *Integrity*: The property that data has not been modified or destroyed in an unauthorized manner.
- *Non-repudiation*: The process of ensuring that the author of a document cannot later claim not to be the author.
- *Access Control*: Encompasses any mechanism of granting access to data or performing an action. An authentication method is used to check a user login; then the access control mechanism grants and revokes privileges based on predefined rules.
- *Availability*: Demands that a computer system's assets be available to authorized parties when needed (Braz and Aïmeur, 2005).

The three essential security properties of *confidentiality*, *integrity*, and *availability* rely on differentiation between authorized and unauthorized users. In order to differentiate between them, authentication must be present. The authentication process is based on a risk criterion. High-level risk systems, applications, and information necessitate distinct forms of authentication that more precisely affirm the user's identity than would a low-level risk application, where the confirmation of the identity is not as significant from a risk standpoint (e.g. anonymous authentication in a library). This is typically referred to as "stronger authentication".

2.2.1 The context of Authentication in Computer Security

The authentication services are located beneath the Security Operations ring in an ideal organization's computer security framework (Figure 2.2). The trusted security framework is no longer a limited technological matter: It supports strategic initiatives and provides a platform for taking a business to new levels of competitiveness. The enterprise security framework must provide confidentiality, integrity, and availability throughout the enterprise, bringing it into conformity with corporate objectives.



Figure 2.2: User authentication in an ideal organization's security framework (Accenture, 2004).

The framework is centered on the Business Assets to be protected, which are identified and prioritized through the Security Strategy and Management. The Security Management is in charge of coordinating and supervising the different Security Services (Security Operations, Security Compliance, Security Policy and

Standards, and finally Security Awareness). Security services are the services supplied by a system for implementing the security policy of an organization. A standard set of such services includes identification and authentication, access control and authorization, accountability and auditing, data confidentiality, data integrity and recovery, data exchange, object reuse, non-repudiation, and finally reliability.

And why focus on authentication? Authentication is more important than ever due to the collapse of network security perimeters, the expansion in the number of devices wanting to access company networks, and the rising number of remote users and wireless devices (including laptops) since users want to access increasingly diverse applications. The information users (i.e. average corporate/consumer computer users) need to access has broadened to comprise all aspects of both personal and business purposes, including e-mail, a greater range of applications, and various types of data. In particular, there has been an impressive increase in corporate users' need to access their organizations' network resources, characterized by a growing number and variety of users (e.g., local and mobile users, telecommuters, etc.), applications, access methods, and extensions of enterprise networks to include third parties (i.e., customers, suppliers, partners, employees, consumers, etc.), hence exposing organizations to significant risks unless they take protective measures.

According to JanneyMontgomery (2005), the total authentication market- (comprising tokens, smartcards, and biometrics), will achieve \$6.77 billion in revenue by 2008. The same report estimates the authentication market will grow at a 15% CAGR (Compound Annual Growth Rate) through 2009, biometrics at 40% CAGR through 2008, and the smartcard market at 7.4% CAGR through 2008. Also, more than half of the enterprises surveyed in Forrester's Enterprise and SMB Security Survey, North America and Europe, Q3 2007 (Forrester, 2007) have either implemented strong authentication at desktop logon or are planning to start or finish such a project in 2008. Although strong multifactor authentication surely improves security, there is no guaranty that *end users will accept it as a convenient and usable second factor*. "Reflecting the real-world difficulties security managers have had in

keeping users happy with the choice of strong authentication, the vendor landscape is complex and fragmented. Vendors will continue to expand their product lines until enterprise adoption of identity management, including strong authentication, becomes more widespread in the coming years” (Forrester, 2007).

Finally, an organization’s authentication service should be suitable to the risks, and should consider the impact on users, as well as the cost of integration with its existing technology architecture, and total cost of ownership.

According to Allan (2007) from Gartner (Figure 2.3), RSA, the Security Division of EMC (EMC, 2010) has dominated the remote-access OTP token space for years, and now provides a wide range of authentication methods concentrated on the enterprise and consumer spaces, including an In-The-Cloud Authentication Service (ITCAS) complemented by fraud detection capability. Vasco has had the most success in the consumer authentication space in Europe and Australia, and now provides a generic infrastructure - a versatile authentication server (VAS) - that supports multiple authentication methods. There are progressively more authentication vendors, but the bulk is focused on a narrow range of authentication methods, and frequently only one (Allan, 2007). “Lightweight OTP methods” employ a form factor other than PC software, a purpose-built handheld device or a smart token.

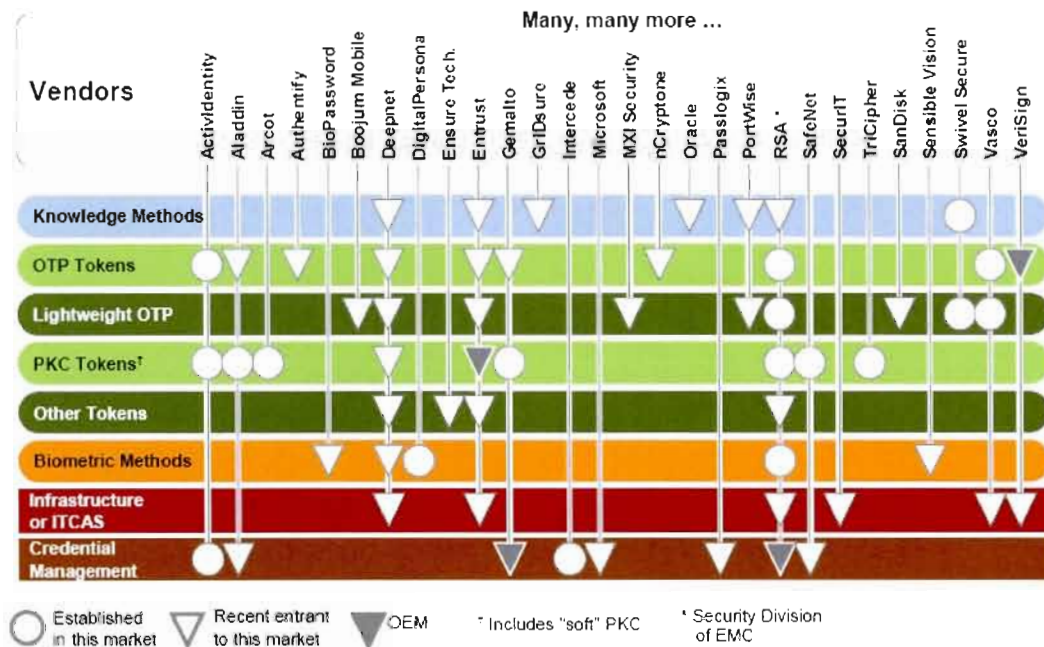


Figure 2.3: The Authentication Marketplace - Gartner Research (Allan, 2007).

Entrust also provides a VAS along with a wide range of authentication methods, including low-cost OTP tokens and fraud detection for online consumer security. VeriSign⁶, a key proponent of the Initiative for Open Authentication (OATH)⁷, entered the authentication market with an ITCAS as an Original Equipment Manufacturer (OEM) (i.e. a computer firm whose products are produced by customizing basic parts supplied by others). OEMs for tokens from other vendors, like Entrust and RSA, complement raw authentication with fraud detection. Smart card vendors that focus on the user authentication market include Gemalto, which offers a variety of authentication tokens (including OTP tokens) that embed smart chips. Most of the newer vendors focus on a single method or set of related methods, but some, such as Deepnet Security, provide open, flexible infrastructures along with a wide range of methods.

⁶ VeriSign, Inc. Mountain View, California (USA).

⁷ Initiative for Open Authentication (OATH). February 12, 2008
<http://www.openauthentication.org/>

2.2.2 Elements of User Authentication

In an authentication process there are some elements that are present as shown in Figure 2.4 as follows:

- A *user (or principal)* to be authenticated. The principal is the entity requesting authorization. It is generally some combination of user, device, and/or service.
- A *credential* which is possessed by the user who submits it as proof of identity. The main types of credentials are shared key (password), One-Time-Password (OTP), digital certificate, and biometric credential.
- A *distinguishing* characteristic that sets apart that particular user.
- A *proprietor* who is responsible for the system in use.
- An *authentication mechanism* to verify the existence of the distinguishing characteristic.
- A *server* which is the authentication key storage.
- A privilege when the authentication is successful by employing an *access control mechanism* which rejects the privilege if authentication is unsuccessful.
- And finally, *contextual information* of the authentication request that encompasses the network and physical location of the request (e.g., Geo location, IP address, workstation), the kind of access provided (e.g., check balance), the time of day, and other elements such as network load, security threat level, and so on.

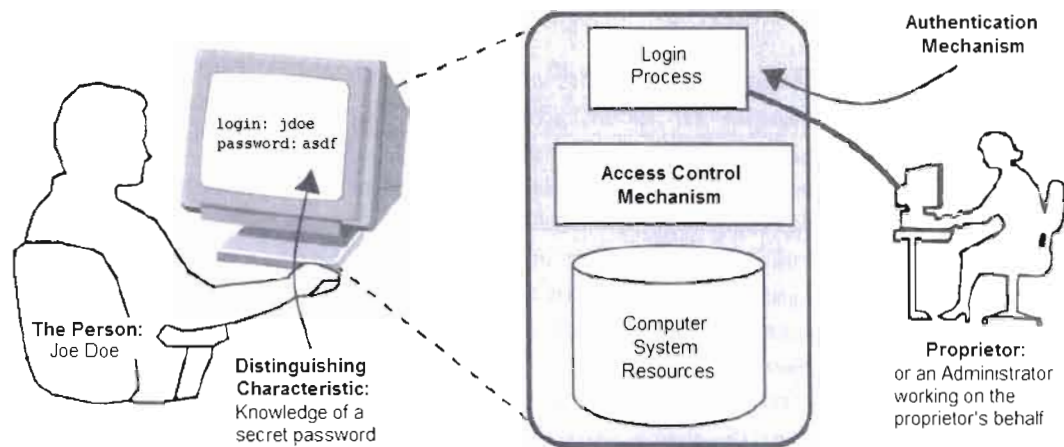


Figure 2.4: The elements of the authentication process (Smith, 2002).

2.2.3 Architectural Design Patterns in Authentication

There are some architectural design patterns that have frequently been encountered in the deployment of authentication systems (Smith, 2002). These patterns analyze problems in terms of space and people, not data and processing. Below is a description of the architectural design patterns employed in authentication:

- **Local Authentication:** This is related to single desktop systems and laptops. The whole system (comprising its authentication and access control mechanism) lives within a single physical security perimeter.
- **Direct Authentication:** Figure 2.5 shows the direct authentication when a client and service share a trust relationship. Direct authentication requires the presentation of credentials, which are usually a user name and password. The service employs these credentials to authenticate the request.

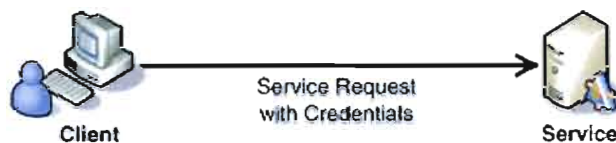


Figure 2.5: Direct Authentication when a client and service share a trust relationship (Microsoft, 2005).

- **Broker Authentication:** Using a broker to carry out authentication when client and service do not share a trust relationship, as shown in Figure 2.6. The broker authenticates the client and then issues a security token that the service can employ to authenticate the client. The security token is constantly verified, but usually the service does not need to interact with the broker to carry out the verification. The reason for this is that the token itself can include proof of a relationship with the broker, which can be employed by the service to verify the token.

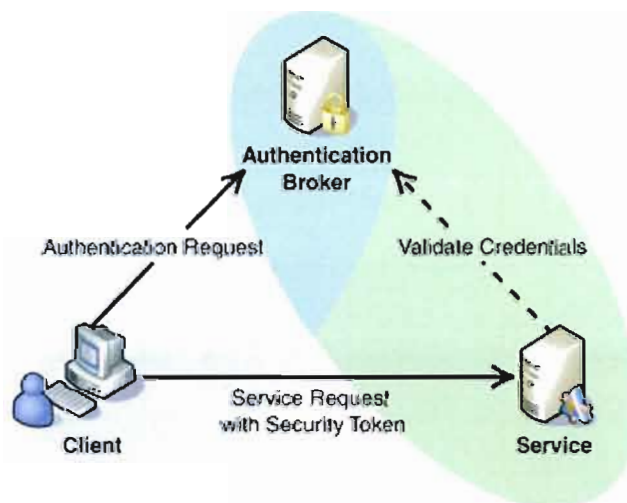


Figure 2.6: Broker Authentication (Microsoft, 2005).

- **Offline Authentication:** Public-key certification software follows an offline authentication pattern, which recognizes authorized users, is stored in multiple locations throughout the system, and is accessible offline. For example, if a legitimate user wants to authenticate to her bank's server, her workstation first acquires the bank's public key certificate and authenticates it with a pre-established public key. Second, it uses the public key within the bank's certificate as part of some other protocol, such as Secure Socket Layer (SSL), to authenticate the bank as the real owner of the private key, which is mathematically related to the certificate's public key (Public key authentication is described later on in this chapter).

2.2.4 Authentication Factors

In some applications, there is no need for users to be authenticated by the system, as in the case of browsing the internet in a public library. So, if the data is public, there is no need to limit access to users. If no authentication is taking place, the user is said to be "anonymous". This is the initial status, after a connection has been opened to the server. However, when the resources are protected, for example in an online banking website, the user must be authenticated in order to have access to the services. The fundamental purpose of security is to control who has access to valuable property, whether physical or logical.

An authentication factor is a piece of information used to authenticate or verify a person's identity. There are four factors of user authentication that might be employed in combination to increase the level of security in the claimed identity of a user according to Table 2.1: Something you HAVE (e.g., a smart card), something you KNOW (e.g., a password or PIN), and something you ARE (e.g., iris recognition). In addition, a fourth authentication factor has been also proposed by Braz and Aïmeur (2002), which is something you CONVEY (e.g., a mobile user who authenticates to a system by conveying a chip user's ID by radio frequency signal from an under-skin chip tag to a wireless device).

Classification	Factor	Examples
Type 1: Authentication by Knowledge	Something you KNOW	<ul style="list-style-type: none"> • A password or passphrase⁸ • A Personal Identification Number (PIN) • Information about the user or family members
Type 2: Authentication by Ownership	Something you HAVE	<ul style="list-style-type: none"> • A physical key • A magnetic-stripe card • A token that generates a One-Time Password (OTP)
Type 3: Authentication by Characteristic	Something you ARE (or a physical attribute)	A Biometric trait: <ul style="list-style-type: none"> • Fingerprint • Iris pattern • Hand geometry • Voice
Type 4 ⁹ : Authentication by Emanation (Braz and Aïmeur, 2002)	Something you CONVEY	<ul style="list-style-type: none"> • A microprocessor-chip computer (ChipTag) implanted under human skin. This ChipTag is able to authenticate users' access to systems and- connect them wirelessly- through -Radio Frequency Identification (RFID) (Braz and Aïmeur, 2002)
Or a combination of the above		

Table 2.1: Guide to Understanding Identification and Authentication in Trusted Systems (NCSC, 1983).

Using any authentication factor alone provides single-factor authentication. Any two authentication factors may be combined to provide two-factor authentication; NCSC-TG-017 calls these combinations Type 12 (“one-two”), Type 13 (“one-three”), and Type 23 (“two-three”) as shown in Table 2.1. Associating all three factors presents three-factor authentication, Type 123 (“one-two-three”). So, associating two or more factors introduces greater security. The most well-known

⁸ Passphrase: is generally longer than a password and includes letters, numbers, words, and random characters. In encrypted communications, one should always use a passphrase rather than a password. For example: *I must g!o down to the sea again, to t7he lonely s8a and the sly.*

example is the magnetic-stripe card (smart card) and PIN (Two-Factor Authentication) used with an Automated Teller Machine (ATM): To access the network, a legitimate user must have both “factors”, just as he must have an ATM card and a PIN to withdraw money from a bank account. Another example is the RSA SecurID® 700¹⁰ hardware authenticator (Figure 2.7), which enforces strong “two-factor” user authentication by requiring the user to present two forms of credentials to prove his identity: something you KNOW (a password or PIN), and something you HAVE (authenticator). An example of a multi-factor authentication method is the combination of password, smart card, and iris recognition, resulting in far greater security (e.g., it can be employed for higher risk transactions).



Figure 2.7: RSA SecurID® 700 hardware authenticator with One-Time Password (OTP).

Consider the following contextual authentication scenario: Before a system grants a legitimate user access to a company’s protected network resources, the system must determine who she is, if she belongs to the system, if she has the right to access the system, and if she is the person she says she is. Actually, the system has required three distinct elements - *identification, authentication, and authorization* - that together comprise the so-called *access control*. However, how does the system confirm that she is who she says she is? For example, entering her password does not prove it is her. Hence, the system needs the identification and authentication to authorize access for her. The AI may be gathered from one of the -authentication factors- shown in Table 2.1.

¹⁰ RSA, The Security Division of EMC. May 25, 2010 <http://www.rsa.com/products/secuid/datasheets/10306_SID700_DS_0709.pdf>.

As already mentioned, authenticated identities are the foundation for several other information security services. Typically an organization needs to do the following:

- Control individual users' access to its information systems (Authentication);
- Control individual users access to the resources and services supplied by those systems (Authorization);
- Generate an audit trail of users' access or attempted access to those systems, resources, and services.

2.2.5 User Authentication Methods

This section describes the most representative user authentication methods currently available for IT systems, such as Passwords and PINs, Authentication Tokens (also known as authenticators), Kerberos, Biometrics, and Single Sign-On (SSO). It also introduces an advanced user authentication method called Under-Skin RFID Chip.

2.2.5.1 Passwords and PINs

To avoid other users from using your account through your username, you are required to have a password. A username is a unique identifier which can be system-defined or user-defined¹¹. Usernames are usually built of text so that people might easily remember and type in their user names during the logon process. A password is a secret word or string of characters that is employed for authentication, to prove identity or gain access to a resource. A password allows you (and only you) to access protected network resources in a computer system. A simple password such as

¹¹ There are several types of passwords such as weak, strong, system-defined, or user-defined passwords, passphrases, and PINs. One of the goals of this thesis was to identify the main user authentication task scenarios. They have been built only taking into consideration its main use cases, which in the case of Password/PIN scenarios, is "a user successfully logs into a system." So this use case did not consider all possible and alternative password scenarios such as changing a password, resetting a password, or providing passphrases hints, and other alternative scenarios due to the fact that they are out of the scope of this thesis.

hello, which is in fact easy to remember, has weak password strength, while a complex password such as hCytW7m9!, which is hard to remember, has strong password strength. *Password strength* is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. The strength of a password is a function of length, complexity, and unpredictability. A password essentially proves to the computer system that you are who you say you are. The use of a password is by far the most common knowledge-based (Type 1) authentication method (Figure 2.8) in user authentication. A computer system may employ other Knowledge-Based Authentication (KBA) methods, such as secret questions and answers to check the identity of a user by asking for a password.

A long password, especially one with inserted spaces, is called a *passphrase*. A passphrase is a special type of token-based password where the tokens are words instead of symbols from a character set (e.g. "Where is my checked shirt?"). In principle, the longer the password the stronger it is. (In fact, it is less guessed and less exposed to certain kinds of attacks).

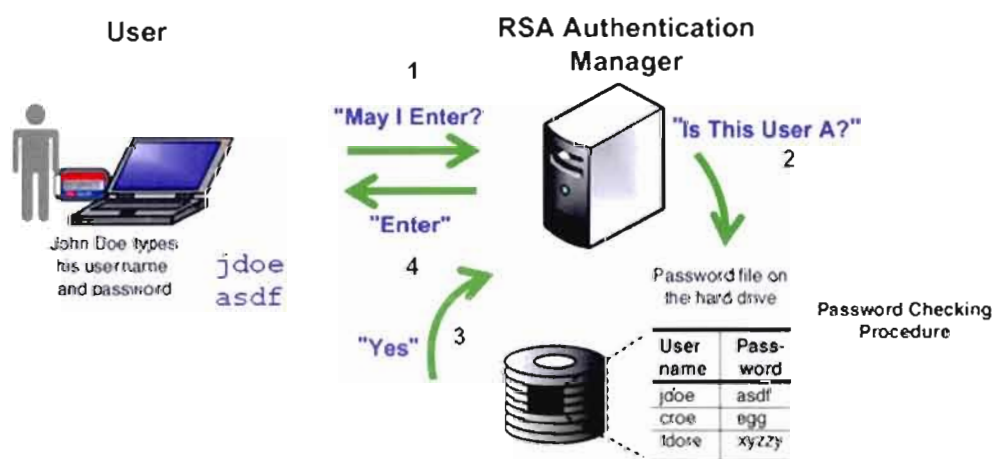


Figure 2.8: The password authentication process (adapted from Smith (2002) and RSA® Authentication Manager 7.1¹²).

¹² Enterprise-class security engine for RSA SecurID® authentication. RSA-The Security Division of EMC May 23, 2010 <http://www.rsa.com/products/securid/datasheets/9239_SIDAM_DS_0208-lowres.pdf>

Passwords are the first line of defense against attacks to a computer system. The rules for password choice can be definitely a burdensome problem for a user and a security problem for a system. For example, trivial choices that are easy to guess are broken within seconds using password cracking techniques - the longer the password the more difficult it is to crack. To prevent hackers from gaining access to a company's computer or files, experts recommend -using a complex password (strong password) that is capable of, in a first instance, increasing the Short-Term Memory (STM) load of users causing frequent errors. As already mentioned, the capacity of STM is usually limited to 7 ± 2 items (e.g. letters, digits, words, etc.) (Miller, 1956). Traditional password systems include many design features for the purpose of making trial-and-error attacks as difficult as possible. Actually, they violate most of the recognized usability standards for computer systems. From the eight "Golden Rules" for interface design recommended by Schneiderman and Plaisant (2005), password interactions break six rules as shown in Table 2.2. Table 2.3 presents the "six password broken rules"- and their corresponding usability issues through a couple of examples and security benefits. Users also should follow a set of rules (i.e. password security policy) especially related to password creation: "Your password must contain: 8 to 32 characters, at least 6 alphabetic characters (a mix of upper and lower cases), at least 1 non-alphanumeric character, not allowed: @<>&%. You may not re-use one of your last 3 passwords. Do not write anything down" (EMC, 2009). If users don't pay proper attention to this password policy, their accounts become vulnerable to intrusion by spoofers.

Golden Rules of User Interface Design	Adequate for Passwords?
1. Strive for consistency	Yes
2. <i>Frequent users can use shortcuts</i>	No
3. <i>Provide informative feedback</i>	No
4. Dialogs should yield closure	Yes
5. <i>Prevent errors and provide simple error handling</i>	No
6. <i>Easy reversal of any action</i>	No
7. <i>Put the user in charge</i>	No
8. <i>Reduce short-term memory load (F)</i>	No

Table 2.2: Do the 8 golden rules of user interface design apply to security systems? (Schneiderman and Plaisant, 2005).

Item	Usability	Security
Frequent users cannot use shortcuts	Users can't take shortcuts: the system won't match the first few letters typed and -fill in the rest.	Prevents dictionary ¹³ and cavedropping ¹⁴ attacks.
Don't provide informative feedback	Users hardly see the password they type: they can't find out repeated letters/accidental misspellings.	Prevents guessing attacks and Social Engineering ¹⁵ .
Don't prevent errors and don't provide simple error handling	Most systems only mention success or failure: they don't show how close the password guess was, or even discern between a mistyped username and password.	Prevents guessing, eavesdropping, and social engineering attacks.
Difficult reversal of any action	Most systems keep track of incorrect guesses and take irreparable action (locking the user's account) if several bad guesses happen.	Prevents guessing, eavesdropping, and social engineering attacks.
Don't put the user in charge	The system makes users be "responders" of actions rather than the initiators.	Prevents guessing, eavesdropping, and social engineering attacks.
Don't reduce short-term memory load	Users must follow a set of security policies related to password creation recommended by EMC, (2009). STM is usually limited to 7 ± 2 items.	Prevents guessing, eavesdropping, and social engineering attacks.

Table 2.3: The "six password broken rules" and their corresponding usability and security mismatches (Schneiderman and Plaisant, 2005).

¹³ A form of attack in which an attacker uses a large set of likely combinations to guess a secret.

¹⁴ Electronic eavesdropping is the intentional surveillance of data: voice, fax, e-mail, mobile telephones, etc. often for nefarious purposes;

¹⁵ To infiltrate a physical building or information systems using non-technical means (e.g. searching user desks for passwords on notes).

Users have too many passwords with different values, expiration periods, and composition rules. Usually these new complexity requirements drive users to forget their passwords and call the help desk, write down their passwords, and finally chose common, insecure passwords. A Web-based pool was developed by Strauss and Corbin (1990) to gather quantitative and qualitative data on user behaviour and perceptions concerning password systems. The key findings are respectively: A large number of passwords diminish their memorability and raises insecure work practices, causing serious usability problems; users' understanding of what it is secure password content is deficient; and at last, the users are not well-informed about security issues.

In an extremely networked world as the Internet for example, wherein users may access numerous applications, password protection is regarded as expensive, awkward, and insecure. The requirement of authentication to access different applications, services, or facilities might generate frustration among users on a daily basis, because users might need to frequently access the same secured applications in a short period of time.

The problem with passwords is that if they are not encrypted, or if the encryption is easy to break, passwords and passcodes (i.e. a PIN plus a tokencode) are vulnerable to eavesdropping and replay. And if it is encrypted, there are other types of attacks that may be used. A brute force or dictionary attack consists of an attack that just tries possibility after possibility until the right one is found. Utilities to help an attacker with this kind of attempt are easily found on the Internet. Short passwords, made of one simple word, are the easiest to figure out with this kind of attack. Therefore, many Super Administrators require pass phrases, complex combinations of words. Controls will also often require the use of a password policy as mentioned above, which makes it more random in nature and harder to guess. In some environments, users must remember many complex passwords and pass phrases

and end up writing them down near the computer. This becomes the vulnerability (EMC, 2009).

A Personal Identification Number (PIN) is in turn a unique personal character string used as a password, usually with a four-digit number, which must be entered by the user before a remote terminal (e.g. desktop, mobile device, ATM, etc.) or Point-of-Sale terminal (e.g. kiosk) can be used to transfer information or complete a transaction. PINs are often employed with a magnetic-stripe card or smart cards at an ATM to authenticate, for example, a bank customer. PINs are also employed with authentication tokens such as the RSA SecurID® Token for BlackBerry (Figure 2.9) and SafeWord Remote Access (Figure 2.10). A classic strategy to defend against Personal Identification Number (PIN) guessing attacks in authentication tokens is to lock a user's account usually upon three consecutive invalid PIN attempts. However, this "classic" strategy could seriously undermine the usability of the system or what is called a locking PIN system. Since the PIN is locked, it can only be unblocked by the Help Desk Administrator. Until then, the user will be not able to logon to the system. SafeWord tokens¹⁶, for instance, will only generate the correct password after the correct PIN has been entered, and has attack lockouts if the wrong PIN is entered too many times. In the worst case scenario, the token becomes totally blocked, and service is not available.

2.2.5.2 Authentication Tokens

An Authentication Token (AT or sometimes also referred to as an authenticator), broadly defined, is a unique hardware or software object given to specific users to prove their identities. ATs provide a means of authenticating and identifying an end-user. To verify the identity of the token's owner, the host system performs its authentication protocol using data encoded on the token. Some ATs

¹⁶ Aladdin Knowledge Systems Ltd. June 23, 2009
<<http://www.securecomputing.com/index.cfm?skey=643>>

contain advanced components like a microprocessor and semi-conductor memory¹⁷, and they support sophisticated authentication protocols which provide a high level of security. Generally, ATs allow the use of SSO systems that enable users to utilize an AT to sign on only once to a wide range of applications or Websites for which they demand access.

ATs come in a variety of physical forms. The size, shape, and materials from which a token is manufactured are referred to conjointly as the token's *form factor*. A token's form factor involves trade-offs that must be evaluated for deployment in different applications so security professionals can choose the form factor that is best suited to a particular application. An AT may have different form factors depending on the authentication method used and the vendor, i.e., a handheld device with or without a keypad, a key fob (small hardware device with built-in authentication engines), and a smartphone executing a certain type of software from a vendor.

There are two types of token form factors: Non-contact Tokens and Contact-Tokens.

2.2.5.2.1 Non-Contact Tokens

Non-contact Tokens demand no electrical or physical contact with a token reader device such as proximity cards (e.g., an employee brings it close to the card-reader in order to gain physical access into the office). One-Time-Password (OTP) generators and handheld Challenge-Response calculators are microprocessor-based authentication tokens which do not require a physical connection to host systems. These devices communicate directly with human users through an onboard display and some form of keypad. Users relay authentication data, such as passwords or encrypted challenges, between tokens and host systems manually.

¹⁷ RAM and ROM are semi-conductor memories. One of the characteristics of the semi-conductor memory is the ability to write information at an extremely high speed and read it out.

2.2.5.2.1.1 One-Time Passwords (OTP)

A security system that requires a new password every time users authenticate to a system, One-Time-Password (OTP) generators- make it more difficult to gain unauthorized access to restricted resources, such as by a hacker replaying an intercepted password. An OTP system usually employs microprocessor-based ATs, which do not demand a physical connection to host systems. These devices communicate directly with users through an embedded display and some form of keypad. Users transmit authentication data, such as passwords or encrypted challenges, between tokens and host systems manually. Examples of OTP generators are the following: RSA SecurID® Token for BlackBerry (Figure 2.9), SafeWord RemoteAccess (Figure 2.10), RSA SecurID® Software Token 1.1 for iPhone Devices (Figure 2.11), and DigiPass from Vasco (Figure 2.12).



Figure 2.9:
RSA
SecurID®
Token for
BlackBerry.



Figure 2.10: SafeWord
RemoteAccess¹⁸.



Figure 2.11: RSA
SecurID® for
iPhones¹⁹.



Figure 2.12: DigiPass
Go 3²⁰.

¹⁸ Aladdin Knowledge Systems Ltd. June 23, 2009

<<http://www.securecomputing.com/index.cfm?skey=643>>

¹⁹ RSA SecurID® Token for BlackBerry. RSA, The Security Division of EMC. May 23, 2010

<<http://www.rsa.com/node.aspx?id=1165>>

²⁰ Vasco Data Security Inc. October 15, 2007

<http://www.vasco.com/products/digipass/digipass_go_range/digipass_go3.aspx>.

OTP generators produce a sequence of passwords that are synchronized with host systems. Each password is only valid for one authentication, and so cannot be recorded and replayed to obtain access. Synchronization is frequently based on a secret initial source value, which is swapped at specific time intervals, or each time an authentication event takes place. Then, without knowledge of the secret value and the number of times it has been swapped, an attacker may not foresee the next password in the sequence even if one or more previous passwords are known. When an OTP is combined with a *Personal Identification Number* (PIN), two-factor authentication is achieved because the client needs to have something (the token) and know something (the PIN). But how exactly does authentication via OTP work? Figure 2.13 shows the OTP authentication process.

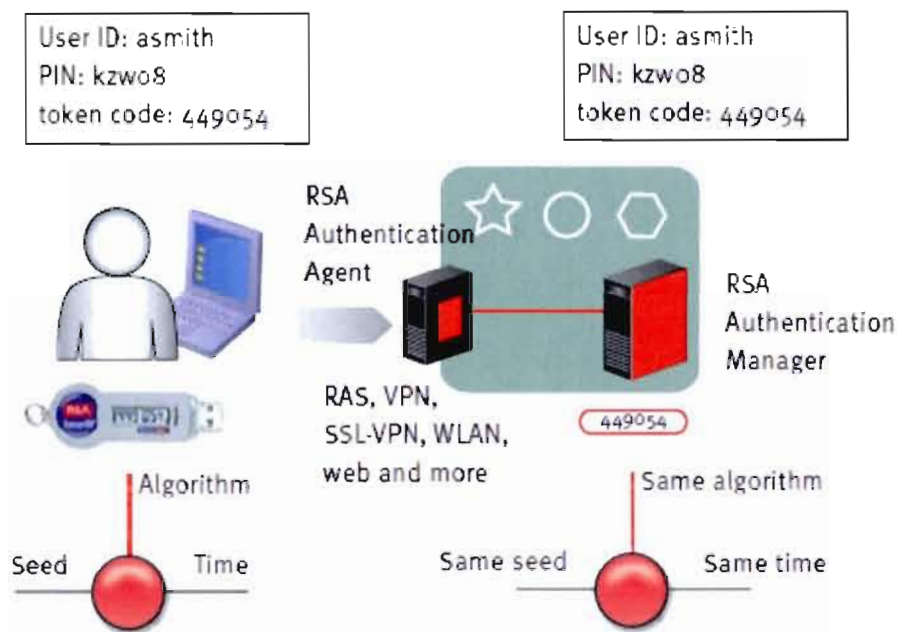


Figure 2.13: Authentication via OTP²¹: **Pre-condition:** Authentication server randomly generates OTPs (six-digit numeric token numbers) on the user's token display (e.g. 435961) every 60 seconds. User has an assigned authentication token.

²¹ Authentication Manager - Enterprise-class security engine for RSA SecurID® authentication. RSA, The security division of EMC <http://www.rsa.com/products/secuid/datasheets/9239_SIDAM_DS_0208-lowres.pdf>

1. User opens and connects to the Virtual Private Network (VPN) application from his computer;
2. User enters his username (jdoe) and a 4-digit numeric PIN (7234) in the username and passcode fields on the login application screen-;
3. User types and appends the six-digit numeric token number (currently displayed on her token) to the entered PIN on the login screen. This number (7234435961) becomes the *passcode*, which is in fact the combination of PIN and token number.
4. User clicks “OK”.
5. If the authentication is successful, the authentication server validates the passcode and grants user access to the network’s protected resources.

2.2.5.2.1.2 Challenge Response (C/R) Authentication

There are currently three main types of Challenge Response (C/R) generators: Handheld C/R, CAPTCHA: Telling Humans and Computers Apart Automatically, and SiteKey. The following describes the operation of these methods in general terms, but many variations are possible.

A Handheld CR (HC/R) is a security mechanism for verifying the identity of a user or system without the need to transmit the actual password across the wire or wireless network. The server sends a *challenge* (string of alpha or numeric characters) to a client; this client then combines the string with its *response* (password), and from this a new password is generated. The new password is sent to the server; if the server can generate the same password from the challenge it sent the client and the client's password, then the client must be authentic. Some vendors provide software tokens (Figure 2.11) as an alternative to their hardware tokens. The main advantage of the software token is ease of use, since the user does not have a separate token; however, this reduces security. To solve this problem, some vendors provide smart cards or

Universal Serial Bus²² (USB) keys (Figures 2-16 and 2-18). Other vendors make simpler the authentication process by providing a connection without interface between the hardware token and the user's computer. Typically, this is a Radio-Frequency²³ (RF) or wireless (Bluetooth or (IEEE802.11, 1999) interface. RF/wireless tokens provide an advantage over traditional tokens because a user can be continuously authenticated; hence going away from the computer automatically locks the computer or logs the user off.

These HC/Rs require even more data input in comparison with other authentication methods. Regardless of whether it is an average consumer or corporate computer user, the interaction with the device is always difficult since the following data have to be entered: a conventional user ID and password, a PIN, and the challenge. The difficulty of input is nowhere more obvious, since, for example, HC/Rs do not "echo" the password back on the screen as it is typed, but instead asterisks. Even if we make use of an OTP, the input error will be not evident until the server rejects the try. For that reason, the usability of the system suffers greatly. For example, in order to authenticate users to the network resources using a C/R Windows LAN Manager²⁴ (LANMAN), a 14-byte password product feature concatenates with 0's if the password is less than 14 bytes, converts to upper case and splits into 7-byte halves for encryption. This security mechanism of encryption indirectly provides a good practice of usability since users can make a shift key error (e.g. typing a lowercase password) and the system will still recognize their password. On the other hand, this feature reduces the password entropy²⁵.

The CAPTCHA: Telling Humans and Computers Apart Automatically (Ahn *et al*, 2003) is a type of authentication employed to tell humans apart from machines by

²² A serial bus with a data transfer rate of 12 Mbps for connecting peripherals to a microcomputer.

²³ Radio Frequency (RF) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags or transponders.

²⁴ Windows NT LAN Manager version 3 client with first logon prevents subsequent logon activity. Retrieved on June 1, 2009. <<http://support.microsoft.com/default.aspx?scid=kb;ENUS;241338>>

²⁵ We can estimate the number of guesses, on average, the attacker must make to disclose a base secret (e.g. username and password); this metric is named *Average Attack Space*.

avoiding automated responses. It falls into the category of C/R authentication. A CAPTCHA is a program that protects websites against Internet bots (i.e. a computer program which performs automated tasks) by generating and grading tests that humans can pass but existing computer programs cannot. For example, humans can read distorted text as shown in Figure 2.14, but existing computer programs can't. It is usually a word or set of numbers and letters -presented to the user that are in fact obscured or modified in some manner to prevent computers from responding to a prompt.

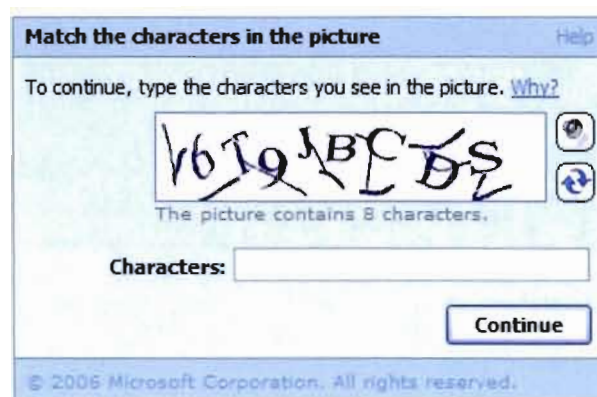


Figure 2.14: CAPTCHA: Telling Humans and Computers Apart Automatically (Ahn *et al.*, 2003).

Another C/R authentication method is SiteKey (BankofAmerica, 2009), a server authentication via images. It is a web-based security system that offers one type of mutual authentication between end-users and websites. Its main purpose is to deter phishing. SiteKey (Figure 2.15) has been employed by Bank of America, a large American financial institution, since 2006. SiteKey is owned by RSA Security (2010).

SiteKey uses the following C/R techniques:

1. User identifies (not authenticates) herself to the site by entering her username (but not her password). If the username is a valid one the site proceeds (screen 1).
2. Site authenticates itself to the user by displaying an image and accompanying phrase that she has earlier configured. If the user does not recognize them as

her own, she is to assume the site is a phishing site and immediately abandon it. If she does recognize them, she may consider the site authentic and proceed (screen 2).

3. User authenticates herself to the site by entering her password. If the password is not valid for that username, the whole process begins again. If it is valid, the user is considered authenticated and logged in (screen 2).

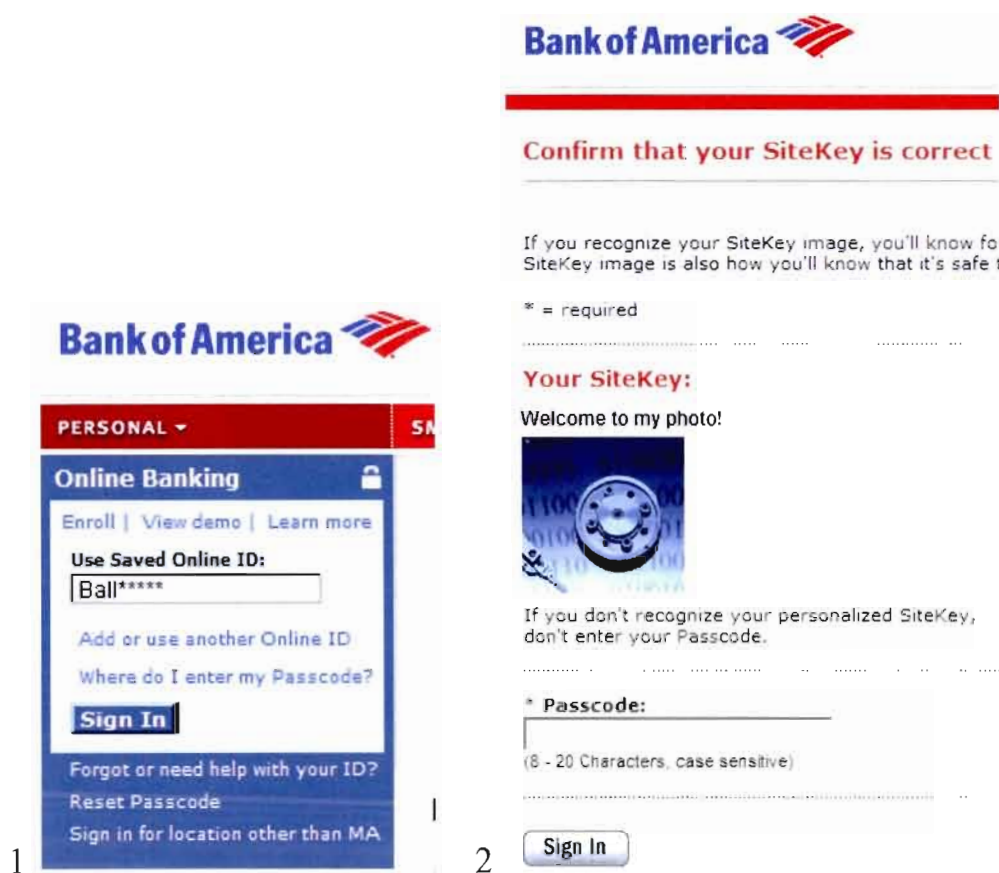


Figure 2.15: SiteKey²⁶ helps prevent unauthorized access to users' accounts. It employs a combination of username (left), then phrase, image and passcode (right).

²⁶ SiteKey. Bank of America. November 19, 2007
http://www.bankofamerica.com/onlinebanking/index.cfm?template=site_key&state=CA

2.2.5.2.2 Contact Tokens

To transfer data, most tokens must make physical contact with a reader device. Examples of such devices are magnetic stripe tokens used in Automated Teller Machines (ATM)²⁷ (Figure 2.17) and hardware authentication USB tokens, which are small hardware devices that can be plugged into a USB port on a computer and can also be used with a password or PIN. Examples of USB tokens are the RSA SecurID® 800 Hardware Authenticator (Figure 2.16) and the VeriSign Secure Storage Token²⁸ (Figure 2.18). But what if you lose your token, or lock them in your car, or what if it is stolen? This is a common problem for tokens.



Figure 2.16: RSA SecurID® 800 with OTP.



Figure 2.17: Magnetic stripe token.



Figure 2.18: VeriSign Secure StorageToken.

2.2.5.3 Digest Access Authentication

Digest Access Authentication²⁹ (DAA) is one of established methods a web page can employ to negotiate credentials with a web user. It employs the Hyper Text Transfer Protocol (HTTP)³⁰. This method builds upon (and obsoletes) the basic authentication scheme, enabling a user's identity to be established without having to send a password in plain text over the network. A DAA scheme provides no encryption of message content. The goal is simply to create an access authentication

²⁷ Retrieved on June 25, 2008

<http://www.1stsource.com/personal_banking/products/resource_plus.htm>

²⁸ <<http://www.verisign.com/static/DEV016111.pdf>>

²⁹ <http://www.ietf.org/rfc/rfc2617.txt>

³⁰ The basic authentication scheme was originally defined by RFC 1945 (Hypertext Transfer Protocol – HTTP/1.0).

method that prevents the most serious flaws of basic authentication. One advantage of the basic authentication scheme is that it is supported by roughly all popular web browsers.

The classical transaction is comprised of the following steps: The user asks for a page that requires authentication but does not provide his credentials (i.e. user name and password). Typically this is because the user simply enters the address or follows a link to the page. The server responds with the “401” response code, providing the authentication realm and a randomly-generated, single-use value called nonce³¹. At this moment, the system will show the authentication realm (normally a description of the computer or system being accessed) to the user and prompt for her credentials through the login window (Figure 2.19). Once the credentials have been provided, the system re-sends the same request but adds an authentication header that contains the response code. In this example, the server accepts the authentication and the page is returned. If the credentials are invalid, the server may return the “401” response code and the system would prompt the user again.

³¹ Nonce is a random or non-repeating parameter value that is included in data exchanged by a protocol, usually for the purpose of guaranteeing liveness and thus detecting and protecting against replay attacks. A nonce can be a time stamp intended to limit or prevent the unauthorized replay or reproduction of a file.

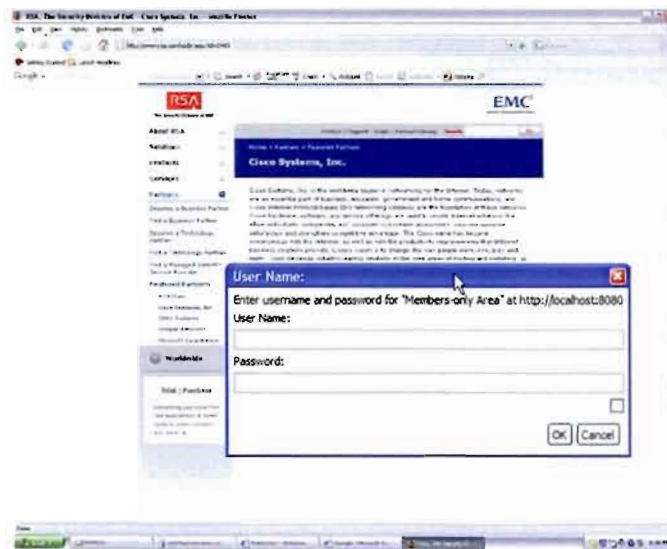


Figure 2.19: Digest Access Authentication's window login in Firefox.

2.2.5.4 Out-Of-Band Authentication (OOBA)

Out-Of-Band Authentication (OOBA) is the utilization of two separate networks working concurrently to authenticate a user. Therefore it prevents an intruder from having a chance to hack the password. It is as a strong defense against man-in-the-middle attacks and sophisticated online hackers. OOBA works well because even if an intruder gains all security credentials to a user's account, a transaction cannot be completed without access to the second authentication network. Consider the mobile phone scenario to verify the identity of the user involved in a Web transaction. Some of the advantages of using a mobile phone-based OOBA are the following:

- No supplementary hardware, software, or training is necessary for the end user. However it is still difficult for the average consumer computer user to cope with.
- Users already carry phones and keep close track of them.
- Mobile phone communication can take place in true real time.

- Mobile phone authentication can require interaction with a human being.
- The Public Switched Telephone Network (PSTN) is a secure network.
- A strong, humanly understandable audit trail of the transaction is captured.

OOBA using a mobile phone enables legitimate account owners to be made aware of attempts to breach their accounts. If an account is protected by mobile phone-based OOBA, the user will receive a call to authenticate a large dollar transaction before it can be completed. If the rightful account owner is not involved in the web transaction, he cannot complete the mobile phone-based authentication, and the counterfeit transaction will be cancelled before losses are incurred.



Figure 2.20: SafePass Mobile.

To improve online security, Bank of America introduced SafePass³² (Figure 2.20). To register for the SafePass OOBA service, customers add their mobile phone

³² SafePass: Link retrieved on May 22, 2009
 <http://www.bankofamerica.com/privacy/index.cfm?template=learn_about_safepass>

numbers to the accounts overview section on Bank of America's Web site. When a customer initiates certain online transactions, the user will be prompted to enter a six-digit code, which is sent via text message to the user's mobile phone. The code is required for transactions such as money transfers for amounts greater than the current limits, adding new bill payees, or adding new accounts for online transfers. The code expires within 10 minutes of being issued or immediately after it is used.

2.2.5.5 Risk-Based Authentication (RBA)

Similar to layered authentication, RBA requires various levels of proofs, depending on the risk level of the transaction. This term is used interchangeably for systems where risk assessment is used in two different ways: 1) In some systems, risk assessment is employed to determine the toughness of the processes and procedures to sign up and use a particular set of resources. The same credentials will be employed in every session, but users who need different kinds of resources possibly will use different credentials. A user name and password will be sufficient for some users, whereas others with more access to sensitive information may need, for example, a two-factor hardware token. 2) RBA is employed when systems require different authentication levels for the same user, based on the specific transaction, not identity. For example, many web services will use a cookie, placed on the browser from an earlier session, as proof of identity for browsing catalogue pages, but will ask for a user name and password to make a purchase.

An example of RBA is the RSA® Adaptive Authentication, which is an RBA and fraud detection platform that measures over one hundred risk indicators to identify high-risk and suspicious activities. Adaptive Authentication is powered by RSA® Risk Engine (Figure 2.21) that conducts a risk assessment of all users behind the scenes. A unique risk score is assigned to each activity, and users are *challenged* when an activity is identified as high-risk and/or an organizational policy is violated. Adaptive Authentication monitors and authenticates activities based on risk, profiles,

and policies by correlating device identification profiles, behavioral patterning profiles, user profiles, RSA® eFraudNetwork™ feeds, and fraud intelligence.



Figure 2.21: The RSA® Engine³³ measures a number of factors in generating a risk score.

The RSA Risk Engine is a proven, self-learning technology that evaluates each online activity in real-time, tracking over one hundred indicators in order to detect fraudulent activity. A unique risk score, between 0 and 1000, is generated for each activity. The higher the risk score, the greater the likelihood -that an activity is fraudulent. Adaptive Authentication protects users while accessing Websites & portals, SSL VPN applications, and WAM applications, for example.

2.2.5.6 Public Key Authentication

In conventional cryptography, the sender and receiver of a message know and use the same secret key; the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. This method is known as secret key or symmetric cryptography. The main challenge is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, a phone system, or some other

³³ Adaptive Authentication for the Enterprise (RSA® Engine) datasheet. May 21, 2010 <http://www.rsa.com/products/consumer/datasheets/10087_AAVPN_DS_0409.pdf>

transmission medium to prevent the disclosure of the secret key. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. The generation, transmission and storage of keys are called *key management*; all cryptosystems must deal with key management issues. Because all keys in a secret-key cryptosystem must remain secret, secret-key cryptography often has difficulty providing secure key management, especially in open systems with a large number of users.

In order to solve the key management problem, Diffie and Hellman (1976) introduced the concept of public-key cryptography. Public-key cryptosystems have two primary uses, encryption and digital signatures. In their system, each person gets a pair of keys, one called the public key and the other called the private key. The public key is published, while the private key is kept secret. The need for the sender and receiver to share secret information is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. In this system, it is no longer necessary to trust the security of some means of communications. The only requirement is that public keys be associated with their users in a trusted (authenticated) manner (for instance, in a trusted directory). Anyone can send a confidential message by just using public information, but the message can only be decrypted with a private key, which is in the sole possession of the intended recipient. Furthermore, public-key cryptography can be used not only for privacy (encryption), but also for authentication (digital signatures) and various other techniques.

In a public-key cryptosystem, the private key is always linked mathematically to the public key. Therefore, it is always possible to attack a public-key system by deriving the private key from the public key. Typically, the defense against this is to make the problem of deriving the private key from the public key as difficult as possible. For instance, some public-key cryptosystems are designed so that deriving the private key from the public key requires the attacker to factor a large number; in

this case it is computationally infeasible to perform the derivation. This is the idea behind the RSA public-key cryptosystem³⁴.

Another important component often used in Public Key Authentication is a smart card. Smart cards are plastic cards that include integrated circuit cards. They are tamperproof and can be employed to store users' certificates and private keys. Smart cards can execute complex public key cryptography operations, for instance digital signing and key exchange.

It is possible to deploy smart cards (and smart card readers) to offer stronger user authentication and non-repudiation for a variety of security solutions, including logging on over a network using fingerprint (i.e. the smart card includes cryptographic keys and biometric fingerprint data), secure email, and other methods. But what are the benefits of making use of smart cards?

The benefits are the following:

- Private keys are stored on the smart card (tamper-resistant) instead of, for instance, on a users' hard disk (not secure). Therefore smart cards provide stronger security for user authentication and non-repudiation.
- As cryptographic operations are disassociated from the operating system (OS), smart cards are not subject to attacks on the OS (e.g., memory dump attacks which may expose private keys or other cryptographic secrets).
- Logon credentials follow users, so the system administrator for example can issue a single smart card to each network user to provide a set of logon credentials for logging on to local and remote networks, which can reduce the cost of managing separate user accounts for logging on to a network and logging on remotely (Microsoft, 2010).

Additionally, because the administrative support that is needed to administer user passwords is an important cost for large organizations, smart cards can be

³⁴ 3.1.1 What is the RSA cryptosystem? RSA Laboratoires. RSA-The Security Division of EMC. May 23, 2010<<http://www.rsa.com/rsalabs/node.asp?id=2214>>

deployed to reduce the cost of forgotten or expired passwords. Smart cards use PINs instead of passwords. The PIN protects the smart card in case of misuse due to the fact that the PIN is known only to the smart card's owner. A smart card scenario is a user who inserts the card in a smart card reader that is attached to a computer and, when prompted, enters the PIN. The smart card can be employed only by a user who possesses the smart card and has knowledge of the PIN.

2.2.5.6.1 Encryption

When Alice wishes to send a secret message to Bob, she looks up Bob's public key in a directory, uses it to encrypt the message and sends it off. Bob then uses his private key to decrypt the message and read it. No one listening in can decrypt the message. Anyone can send an encrypted message to Bob, but only Bob can read it (because only Bob knows his private key).

2.2.5.6.2 Digital Signatures

To sign a message, Alice does a computation involving both her private key and the message itself. The output is called a digital signature and is attached to the message. To verify the signature, Bob does a computation involving the message, the purported signature, and Alice's public key. If the result is correct according to a simple, prescribed mathematical relation, the signature is verified to be genuine; otherwise, the signature is fraudulent, or the message may have been altered.

2.2.5.6.3 No Usability Features of Public Key Authentication

The “Usability of Security: A Case Study” (Whitten and Tygar, 1998) was developed in order to evaluate the usability of Pretty Good Privacy (PGP) 5.0. PGP is standard software, which uses Public Key Infrastructure to encrypt, decrypt, and digitally sign data for the encryption of Electronic Mail developed by Phil

Zimmermann. The study's authors have chosen PGP because it has a very good user interface by established standards, and they wanted to know whether that was sufficient to allow non-programmers who know little about security to in fact use it effectively. The most meaningful results, obtained from a cognitive walkthrough and user test methods, unequivocally show that users had considerable difficulty with the following: avoiding dangerous errors, encrypting -messages, understanding the public key model, figuring out the correct key to encrypt with and how to encrypt with any key, decrypting a message, publishing the public key, and finally verifying a signature on an email message. Well, these are in fact the basic tasks needed to run - the program correctly! Therefore, PGP has not been considered usable as a way to provide effective security for most email users, according to the authors, because of the fact that there is a "mismatch between the design philosophy behind its user interface and the usability needs of a security utility".

2.2.5.7 Single Sign-On (SSO)

“External compliance requirements and internal security initiatives are driving the need for more complex passwords with more frequent expirations. Without a solution that helps end users, corporate password policies only frustrate end users. Users will write down passwords or use the same password for multiple applications - weakening security and hindering regulatory compliance efforts. Additionally, supporting multiple passwords also hits the IT bottom line as employees use costly help desk resources for password resets” (RSA Security, 2010a).

Single Sign-On describes the ability to use one set of credentials, an ID and a password or passcode, for example, to authenticate and access information across a system, an application, or even organizational boundaries. It may be called Web SSO when everything is accessed through a browser. With SSO, users authenticate only one time for a particular working session, in spite of where the information that they

want to access is located. This process provides improved security over a simple synchronization of passwords.

SSO is quite useful these days when users need to access an ever increasing numbers of applications. SSO has major security and user benefits, as well as the capacity to reduce the helpdesk costs of password management significantly.

There is a security risk with static password-based SSO because a breach of password security means all systems accessible by a particular user can be compromised. Typically, SSO uses are in conjunction with some form of two-factor authentication.

Although security vendors like RSA Security openly tout the benefits of RSA Federated Identity Manager (i.e. SSO)³⁵ as convenient and easy to use, users still need to create a secure (strong) password, which is- a cumbersome task, and remember it, increasing memory workload. This is the very same usability problem encountered with traditional password systems.

A variant of the SSO is OpenID (2010) (not owned by any one company such as AOL), a decentralized sub-set of SSO for the Web that is different in that end-users own an identity URL instead of a password. “You may choose to associate information with your OpenID that can be shared with the websites you visit, such as a name or email address. With OpenID, you control how much of that information is shared with the websites you visit. With OpenID, your password is only given to your identity provider, and that provider then confirms your identity to the websites you visit. Other than your provider, no website ever sees your password, so you don’t need to worry about an unscrupulous or insecure website compromising your identity” (OpenID, 2010). Also, you are able to make use of a single, existing account (e.g. AOL) to sign in to numerous websites without needing to create another username and password. OpenID is an easy method of joining new sites. Yet there are also some challenges associated with using OpenID, such as the struggle of creating a

³⁵ RSA Federated Identity Manager. RSA, The Security Division of EMC. May 23, 2010 <<http://www.rsa.com/node.aspx?id=1191>>

username and password; however, this burden is minimized since you create your credentials only once. Still, users might not understand- or may be confused by openID since it is not the standard username/password process well known even to Novices users. You should not use OpenID for high-privacy sites such as online banking, e-commerce sites, or healthcare sites since you trust only one provider for your –credentials (only one), so it is fine to use it for trivial things (e.g., library). Finally, an openID provider may track your habits since they receive all of the authentication requests.

2.2.5.8 Kerberos

Kerberos³⁶ is an authentication method created by MIT (2006) as a solution to network security problems. Kerberos is a network authentication protocol that supplies strong authentication and shares temporary base secrets for client/server applications by using secret key cryptography. It uses strong cryptography so a client may prove its identity to a server (and vice versa) across an unsecure network session. Thus, all authentication processes happen between clients and servers. In Kerberos ontology, a “Kerberos client” is an entity that obtains a service “Ticket” for a Kerberos service. A client is commonly a user. The designation “Kerberos Server” usually appeals to the Key Distribution Center (KDC). The KDC carries out the Authentication Service (AS) and the Ticket Granting Service (TGS). The KDC has a copy of each password related to every client or server. Hence, it is crucial to keep the KDC as secure as possible.

³⁶ The name Kerberos comes from Greek mythology; it is the three-headed dog that guarded the entrance to Hades.

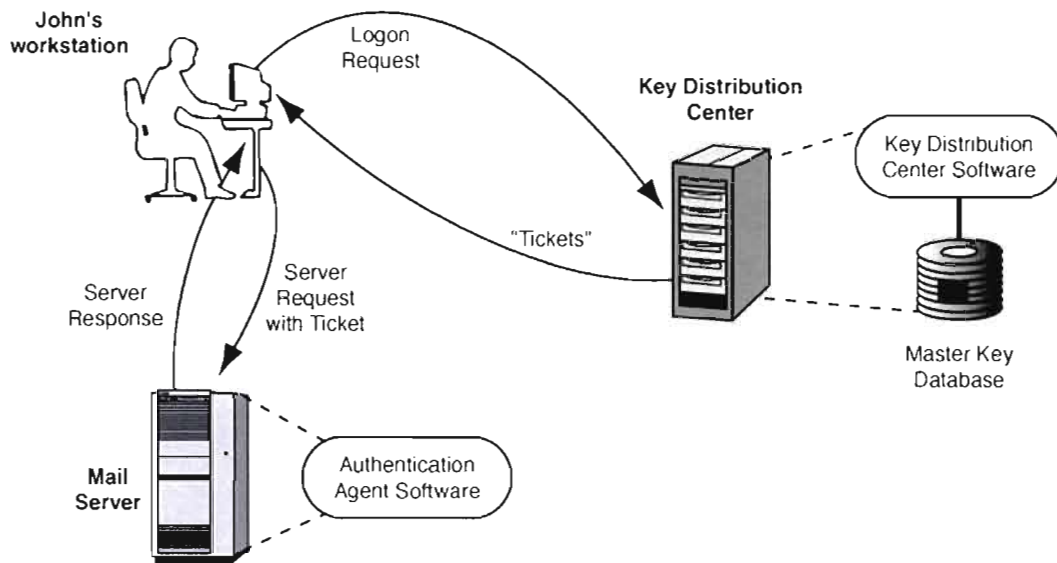


Figure 2.22: A network authentication method: Kerberos (Smith, 2202).

How does it work? An Indirect Authentication design pattern, which appears in the deployment of an authentication system like Kerberos, works according to the following (Figure 2.22): John logs on to the KDC, which supplies him with two encrypted credentials that are designated “Tickets” (one encrypted for his Master Key and the other encrypted with the Mail Server’s Master Key). Then, John decrypts his Ticket to gather the shared secret key (e.g. something that the user and the system hold in common: a password) and forwards the other to the Email Server, which uses that Ticket to authenticate John. In fact, a Ticket is an encrypted copy of a temporary base secret, and it is encrypted with the Master Key known only by the KDC and the Ticket’s supposed receiver.

2.2.5.9 Biometrics

Biometrics is a form of authentication that employs the user’s physical or behavioral characteristics to verify her claimed identity. Physical characteristics like fingerprints, retinas and irises, hand geometry, facial structure, voice/speech, and under skin-based authentication and behavioral characteristics like signature- and keystroke recognition are some of the existing biometric authentication methods.

They may be employed alone or integrated with other technologies such as smart cards, encryption keys, or digital signatures.

Biometric-based authentication applications encompass workstation, network, and domain access, single sign-on, application logon, data protection, remote access to resources, transaction security, and Web security. A Biometric system operates in two modes: *enrolment* and *verification*. In the *enrolment* process, the user's biological characteristics - are acquired and stored (*template*) for later use. This *template* is then placed in a back-end database for later retrieval. In the *verification* process, the user's characteristics are measured and compared against the stored *template*. Biometric authentication is potentially the strongest single authentication method. Nevertheless, it is not infallible, and certain vulnerabilities have far-reaching consequences. Fingerprint, voice, and face recognition are the most appropriate biometrics for restricting access to Web pages, as the sensors for these AMs are small, cheap, and available at large as standard options on PCs from many computer vendors.

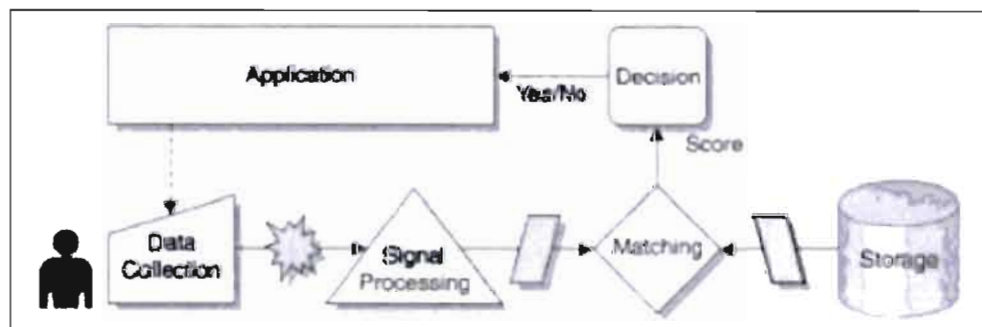


Figure 2.23: Biometric authentication process (Woodward *et al.*, 2003).

How does Biometrics work?

1. The Data Collection process- captures the individual's biometric data using a physical biometric reader (e.g., a fingerprint scanner or an iris camera) and sends the unprocessed biometric data (shown by the splash) to Signal Processing.

2. Signal Processing refines the unprocessed biometric data and sends the biometric sample (shown by the parallelogram) to Matching Process.
3. The Matching Process compares the biometric sample with the individual's template and sends a Score to the Decision process.
4. The Decision process (shown separately from the application in this example) determines whether the Score is above or below some pre-established thresholds, and sends it the final Yes or No to the Application.

The redundancy factor of Biometrics systems is an important aspect of usable security. The best practices in authentication state that multi-factor authentication is generally stronger than any single-factor authentication method. Biometrics is generally recognized as a “good candidate” to be used with another authentication technique: a two-factor authentication. In a two-factor technique (e.g. coupling biometrics with smart card technology) the *redundancy* of the authentication augments the security level, but at the same time diminishes the user experience. In such cases, the authentication process must have built-in *redundancy*, so that a second method must be provided in order to confirm the user's identity. Secondly, physiological traits such as fingerprint or hand recognition generally require a single data scanning of the acquisition device (e.g. fingerprint scanners, iris recognition cameras, etc.) and are stable physical characteristics. On the other hand, behavioural characteristics like voice or signature recognition are more susceptible to alterations (e.g. illness, aging, emotion, etc.) and usually require- multiple- samples from the user in order to generate an accurate template. Therefore, behavioural characteristics require extensive user collaboration and also fast data acquisition devices. The former demands ease of use of the system, while the latter at present doesn't offer data speed or user convenience, since both enrolment and verification generally bother users. In theory, any good security policy means using multiple forms of security to create a *positive redundancy* and make it more difficult for an attacker, but as already stated, this undermines usability.

This thesis argues that it is feasible to balance (*trade-off*) the security usability constraints of the biometric systems by the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) - two measurements often quoted as identifying the capabilities of biometric systems. These rates refer to the number of false negative or false positive matches returned during a biometric evaluation and verification. A FAR of a non-legitimate user may cause damage to the system it is supposed to protect. Frequent FRR of a legitimate user, on the other hand, would have an undesired consequence for the user and drop user acceptance for the system. The risk of having an FRR is clear, for instance, in the case of a user having a terrible cold. The *trade-off* between the FRR and the FAR rate depends on the extent that the system is able to tolerate (i.e., if the decision threshold is increased, FRR increases, and FAR decreases, and vice versa) at the moment of a signature or hand verification, for example.

Biometric authentication is susceptible to capture and replay attacks (i.e. between the scanning device and client software) or between the client and the database server. If an attacker can capture the image or trial template of a user's biometric, then the attacker can replay that data to masquerade as that user. Once an individual's biometric is compromised, that user can no longer employ that characteristic on that system, or on any other analogous system, for life. In contrast to a password or token, a biometric is not able to be reissued, so in order to participate in the biometric system again, the user must re-enroll.

2.2.5.9.1 Fingerprint Recognition

Fingerprint Recognition (FR) is the most common form of biometrics. One of the most widely known and commonly used biometric identification schemes is fingerprint. FR compares users' fingerprints to a previously stored template and determines validity and authenticity based on this comparison. For example, users place their finger on a device that reads the thumbprint (Figure 2.24) on a laptop. To

authenticate them, the system compares the current fingerprint pattern with a previously collected thumbprint, and access is granted only when the patterns are identical.



Figure 2.24: Eikon To Go fingerprint reader³⁷.

Figure 2.24 illustrates a wireless network and electronic access control-based task using a portable Universal Serial Bus (USB) device coupled with software that allows remote employees to simply swipe their finger, for example to log -into the organization's system or access password-protected Web sites.

³⁷ <<http://www.upek.com/solutions/mac>>

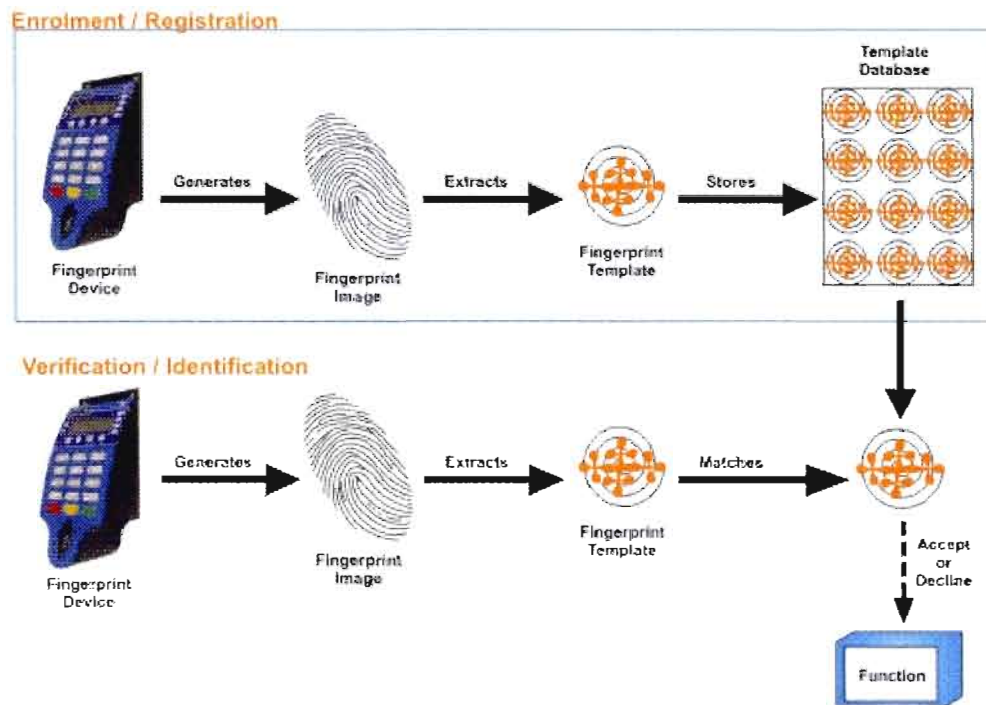


Figure 2.25: Fingerprint recognition scheme BioEnable³⁸.

Fingerprint recognition technology is divided into two distinct processes: verification and identification (Figure 2.25). In the verification process, the user states who she/he is and a fingerprint is taken and compared to the user's previously registered fingerprint. If the fingerprints match, the user is "verified" as who he/she says he/she is. Since the newly acquired fingerprint is compared to only one stored fingerprint, this is called a one-to-one matching process (1:1).

As in the enrollment process, when fingerprint verification is done, only the fingerprint template is used in the comparison, not the actual image of the fingerprint.

In the identification process, the user doesn't need to state who she/he is. A fingerprint is taken and compared to each fingerprint in the database of registered users. When a match occurs, the user is "identified" as the existing user the system found. Since the newly acquired fingerprint is compared to many stored fingerprints, this is called a one-to-many matching process (1:N). As in the verification process,

³⁸ <<http://www.bioenabletech.com/>>

when fingerprint verification is done, only the fingerprint template is used in the comparison, not the actual image of the fingerprint.

2.2.5.9.2 **Optical Recognition**

There are two common types of optical biometrics: retinal and iris. Retinal and iris scanning devices now enable individuals to be scanned even through eyeglasses and contact lenses. Commercial authentication systems are produced by Panasonic³⁹ and LG IrisAccess⁴⁰.

2.2.5.9.3 **Facial Recognition**

An image is examined for overall facial structure. In authentication applications, the system has a camera that searches a user’s face and matches it against the face stored in the user record (Figure 2.26). Commercial face recognition systems are produced by L1Identity.



Figure 2.26: Facelt Argus: facial recognition system from L1Identity⁴¹.

³⁹ <http://www-images.panasonic.com/business/security/products/biometrics.asp>

⁴⁰ <http://www.lgiris.com/>

⁴¹ L-1 Identity Solutions <<http://www.l1id.com/pages/17>>

2.2.5.9.4 Voice/Speaker Recognition

Voice/Speaker recognition is well suited for telecommunications applications, and the latest mobile devices already have the necessary hardware to utilize these applications. Speaker Recognition systems prompt a user for some spoken words, and authenticates the user based on distinguishing speech patterns (voiceprints). Apple Computer (Figure 2.26)- and Sensory Technologies⁴² are vendors which produce speaker recognition systems for user authentication.

One of the drawbacks of Voice/Speaker recognition is the impersonation attack, where an unauthorized individual changes her biometric to appear to be an authorized individual. Also it is more susceptible to alterations (e.g. illness, aging, emotion, etc.).

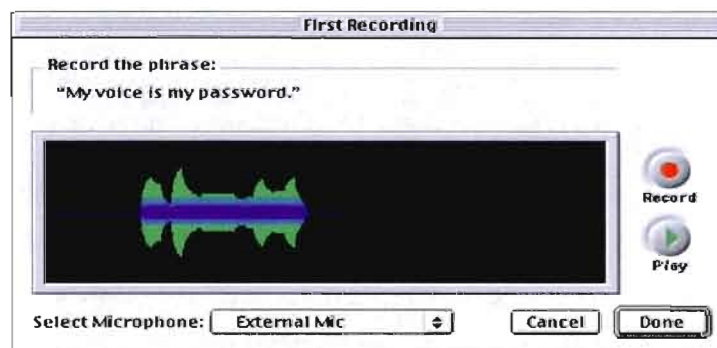


Figure 2.27: This is a voiceprint for the passphrase: *"My voice is my password"*. Spoken passphrase gathered by Apple's Mac OS 9.0⁴³.

2.2.5.9.5 Signature Recognition

Signature recognition operates in a three-dimensional environment. It measures height and width as well as the amount of pressure applied in a pen stroke. Signature authentication software is produced by Cyber-SIGN. Dynamic signature verification (Figure 2.28) takes into account how the signature was realized. It is the alterations in speed, pressure, and timing that take place during the act of signing that are

⁴² Sensory, Inc. <<http://www.sensoryinc.com/>>

⁴³ <http://www.apple.com/macosx/what-is-macosx>

significant, and not the shape or look of the signature. In fact, only the original signer can reconstruct those alterations in timing and X, Y, and Z pressure.

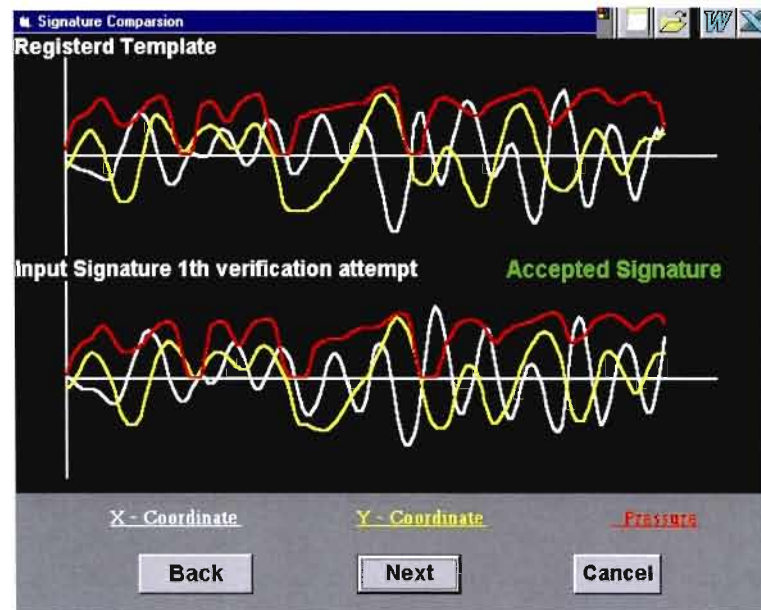


Figure 2.28: Dynamic signature verification from Cyber SIGN⁴⁴.

2.2.5.9.6 Keystroke Recognition

In this type of technology, not only must the attackers know the correct password, but they must be also able to reproduce the user's rate of typing and intervals between letters. AdmitOne⁴⁵ uses two methods to identify individuals: First, the user must know both the correct username and password, and second, the user's typing rhythm must match the biometric template that has been stored and secured by the system. Keystroke Dynamics: Unique Keyboard Signature of an Individual is an example of a keystroke recognition program.

⁴⁴ Witswell Consulting & Services, Inc. <<http://www.cybersign.com/com/CSlacrobat.html>>

⁴⁵ Keystroke Dynamics: Unique Keyboard Signature of an Individual. AdmitOne Security is the identity assurance division of Scout Analytics.
<http://www.admitonesecurity.com/keystroke_dynamics_advantages.asp/>

2.2.5.9.7 Under-Skin RFID Chip - AuthenLink

This thesis's author envisions a Personal Area Network (PAN)⁴⁶ wherein humans communicate (intercommunicate) continually with wireless devices without additional user authentication inputs like passwords, passphrases, PINs, Biometrics, or other existing authentication methods. To this end, an authentication method dealing with human-implanted RFID chips has been developed by Braz and Aïmeur (2003): AuthenLink. It is a wireless and user-centered authentication system to authenticate humans to remote systems. It is specifically designed to protect against fraud, counterfeit, and theft, and particularly well-suited for high-risk security systems. The system achieves its goal through a microprocessor chip (ChipTag) computer implanted under human skin. This ChipTag is able to authenticate a user's access to systems; to connect them wirelessly through the RFID technology; and to enable mobile devices to perform mobile transactions, access files, and so forth. For more information, go to: <http://www.er.uqam.ca/nobel/d362040/masterThesis.htm>.

2.2.5.9.8 Biometrics Trade-Offs Related to Usable Security

From the usability perspective, the ideal systems only require a single biometric signature to enrol a user. For example, fingerprints and hand palm recognition typically require a single reading, whereas behavioural systems like voice recognition or written signatures are more exposed to variation, and frequently require multiple readings to train the system to recognize each user. The secret is to balance the likelihoods of FRR and FAR- so the system barely locks out legitimate users and it doesn't fall for masquerades.

⁴⁶ A Personal Area Network (PAN) is a computer network used for communication among computer devices (including wireless devices) close to one's person. The reach of a PAN is typically a few meters. PANs can be used for communication among the personal devices themselves (intrapersonal communication), or for connecting to a higher level network and the Internet (an uplink).

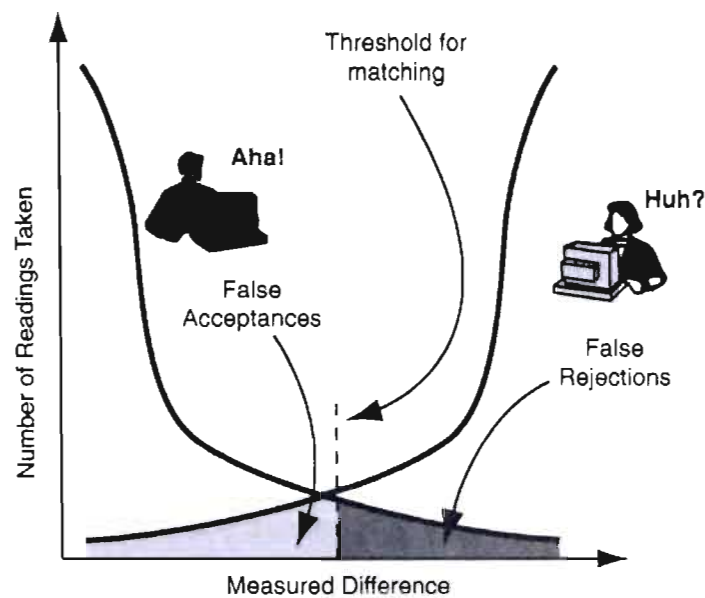


Figure 2.29: Balancing security and usability (Smith, 2002).

According to Smith (2002), as shown in Figure 2.29, Biometrics readings from a particular user should closely match that user's biometric pattern; at worst, the difference should never exceed the matching threshold shown by the dashed line. The risk of masquerades (i.e. FAR) is reduced by moving the curve to the left, which reduces the amount of grey area on the left. But this reduces usability, given that it increases the likelihood of FRRs (the grey area on the right).

For a comparison of the most representative Biometric technologies (Jain, 2004), see Appendix A, Comparative Analysis of User Authentication Methods.

2.2.6 To Whom is Authentication -Targeted?

An important point is which type of users employs authentication systems, or more specifically, to whom is authentication targeted? An end-user is the final or ultimate user of a computer system. The term "end-user" usually implies an individual with a moderately low level of computer expertise. Typically there are four classes of users: *Security User* (a computer security expert user), *Techy User* (a

computer programmer or expert user), *Corporate User* (a user who works for an organization and has - relatively - good knowledge of computer systems), and a *Novice User*.

As already mentioned in Chapter 1, the *end-user* has knowledge of basic Web browsing with typical security and technical expertise or some previous training in the use of computer interfaces, and can be an average corporate computer user or an average consumer computer user. Typically this individual ultimately uses a software or hardware authenticator to enable them to perform their job function (e.g. an individual employs an authentication token to authenticate to a corporate network or access an online banking account).

2.2.7 Comparative Analysis of User Authentication Methods

As part of one of the tasks to understand what authentication methods are, how they work, and what kind of different features are found on them, a comparative analysis of the main user authentication methods has been developed according to Table 2.4. (The full version is available in Appendix A).

The frame of reference for this comparative analysis has been acquired from specific sources indicated in the footnotes and also from observation, experience, and secondary data. It has been established according to the different attributes contained or possessed by the authentication methods such as characteristics/acquisition device, definition, advantages, disadvantages, security, costs, usability and place of use, acquisition time, industrial application, and finally accuracy. The primary grounds for comparison have to do with the most representative and diverse authentication methods currently on the market, and an example that describes a specific advanced method created by this thesis's author which has not been yet commercialized.

Following are explanations of the definitions, abbreviations, and notes/footer sections used in Table 2.4:

- To describe the authentication methods attributes, subjective rating scales have been used, such as:
 - "Security" and "Usability" range from 1=Minimum to 5=Maximum in order to measure the degree of severity issues related to each authentication method.
 - "Automation versus Human" range from 1=Human is better; 5=Machine is better.
- Total Transaction Time (seconds) corresponds to the time it takes for a single user to present the biometric (acquisition time), processing time, and, optionally, might include entry of a PIN or user identifier (Woodward *et al.*, 2003).
- "Accuracy" has two measure rates of authentication by biometrics:
 - False Reject Rate (FRR) where a legitimate user is rejected by the acquisition device.
 - False Acceptance Rate (FAR) where a false user is accepted.
- "Average Attack Space": corresponds to the number of guesses made by an attacker in order to disclose the base secret (e.g. passwords, PINs, etc.).
- Abbreviations used are the following:
 - C/R=Challenge/Response
 - PK=Public Key
 - PRK=Private Key
 - SSO= Single-Sign-On
 - TGS=Ticket Granting Service.

Notes are displayed as numbers between square brackets (e.g. (1), (2), n), and can be found right after the comparative analysis table. They are not displayed in order of entry, given that different items may use the same note. References are also used due to the variety of data provided and the need to minimize the number of notes and/or footnotes on the pages. They can be found in the "References" section.

Characteristics/ Authentication Methods	Pass words (PW)	PIN	Proximi ty card	OTP	C/R	Micro proces sor Chip Card	Public Key	Kerberos	Finger Print, Hand, Face	Voice	Signatu re	Retina/ Iris	Keystro ke	Chip implan ted under skin
Classification	Knowl edge- Based (4 to 8 digits).	Knowle dge- Based (4 to 8 digits).	Authen tication Token. Acquis ition device.	Authen tication Token. Acquis ition device (7)	Authen tication Token. Rando m challen ge (7).	Smart Card. Embed ded integrat ed circuit chip	Public Key Crypto graphy. PK and PRK (8)	Key Distrib ution Center (KDC) (MIT, 2006)	Biomet rics	Biomet rics	Biomet rics	Biomet rics	Biomet rics (user's typing rhythm)	Human chip- based
Advantages	Easily implem ented. If system- generat ed, they are more robust.	Not sent across the networ k.	Last longer, no physic al contact (11).	Very difficul t to guess: passco de (user PIN + code from server)	PW "can be" a PIN (4 digits), easier to remem ber.	More secure than the regular user ID and passwo rd	PK provide s much stronger identity checking	Trusted third- party	Easily sample d & non- intrusive	People use instinct ively	Highly accepte d by users	Unchan geable during lifetime	Enrolm ent/veri fication don't bother the regular work flow	Forger y, stealing , or removi ng the chip is too difficult
Disadvantages	Can be forgott en. Users create easily identifi able base secrets.	Can be forgotten	Forgery (too easy to copy it), loss, theft possible	Brute- force and diction ary attacks	Users share their access permiss ions	Need a smart card reader	Key distribu tion/ex change	Scalabi lity (central ized administ ration)	Crimin al affiliati on	Change s over time	Signatu re can change at any time	Requir es much user cooper ation	Impers onation attack	Impers onation attack

Security (Minimum=1 Maximum=5)	2	2	3	4	3	4	4	4	4	1	3	4	3	4
Characteristics/ Acquisition device (or Data Generator)	Pass words (PW)	PIN	Proxi mity card	OTP	C/R	Multi functio n card	Public Key	Kerbero s	Finger Print, Hand, Face	Voice	Signatu re	Retina/ Iris	Keystro ke Recogni tion	Chip implant ed under skin
Costs (CAD\$)	\$110 Passwo rd Cost Calcula tor (1)	\$110 Passwo rd Cost Calcula tor (1)	\$30 Card	\$80 Token	\$80 Token	\$80 Card	\$220 per seat for around 5000 users	Free software	\$100 to \$500 acquisit ion device	\$5 acquisit ion device	\$300 (Smith, 2002) acquisit ion device	\$300- 700 (Smith, 2002) acquisit ion device	\$45 (keybo ard) Free software is availab le.	\$300 Chip
Usability (Minimum=1 Maximum=5)	1 Break 6 rules of UI design (Schneid erman, 1998).	2 Software general ed and more robust.	4 Practical for users (2)	2	2	3 Used in combin ation with PIN (2)	2 (2) (PGP)	2 SSO	4 (10)	4 Novel Neural Net (3) (10)	4 (10)	4 (10)	4 (10)	4 (10)
Human Versus Automation (15) (Human is better=1; Machine is better=5)	5 Computer generate more secure, automatic Passwords than human.	5 Computer generate more secure & automatic PINs than human.	5	5	5	5	5	5	1	1	1	1	1	1

Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	5-15	5-8	2-6	10-20	10-15	5-15	5-15	5-15	Face= 10-15; Fingerprint=2-9; Hand=4-10	10-12	10-15	4-12	10-15	3-6
	Pass words (PW)	PIN	Proximity card	One Time PW Generator	C/R	Multi function card	Public Key	Kerberos	Finger Print, Hand, Face	Voice	Signature	Retina/Iris	Keystroke Recognition	Chip implanted under human skin
Industrial Application	Unix (Smith, 2002), Windows NT/2000, Mac Keychain	RSA SecurID authenticators (4)	XyLoc (12)	RSA SecurID® 800 Authenticator	Safe Word, Crypto Card, ActivCard	Gemalto	Pretty Good Privacy (PGP) (5)	Kerberos 5 1.3.2 (MIT, 2006)	Verifier 300 LC 2.0 (finger print) L-1 Identity (face)	Apple Mac OS X, Voice Security	Cyber-SIGN	iCamT D 100 (Iritech)	KeyLog PC (6)	Not yet implemented
Accuracy	Average Space Attack: Dictionary attack= 2^{15} to 2^{23} (Smith, 2002)	13-bit (Smith, 2002)	Fair (no available quantitative data)	Average Space Attack = 2^{19} to 2^{63} (Smith, 2002)	Average Space Attack = 54 bits	Distance and cycle delay do have an impact on the accuracy of Read/Write.	Average Space Attack = 1024-bit Public Key= 2^8 (Smith, 2002) (13)	Requires clock synchronization between machines on the network (9)	FR= 3 to 7 in 1,000 (0.3-0.7%); FA= 1 to 10 in 100,000 (0.001-0.01%) (Bollec <i>et al.</i> , 2004)	FR= 10 to 20 in 100 (10-20%); FA= 100 to 1,000 in 100,000 (Bollec <i>et al.</i> , 2004)	FR= 2-try: 2.10% FA= 2-try: 0.58% (Bollec <i>et al.</i> , 2004)	FR= 2 to 10 in 100 (2-10%); FA= 10 to 5 (0.001%) (Bollec <i>et al.</i> , 2004)	No available data	No available data

Table 2.4: Summarized comparative analysis of user authentication methods.

Explanation for Notes (only) for Table 2.4:

- (1) The Password Cost Estimator shows the direct and recurring costs to your organization regarding the use of passwords. Although passwords seem to be free, they actually cost organizations a significant portion of its IT support budget. This silent budget killer is merely the time the technical support staff devote to resetting users passwords. This does not include the abstract costs associated with lost productivity of the user or security breaches, etc., but the labour cost of the help desk personnel physically resetting passwords on the system. Enabling Compliance with Password Policies, Password Cost Calculator. MandyLion Research Labs, LLC. May 20, 2010 <<http://www.mandyionlabs.com/PRCCalc/PRCCalc.htm>>
- (2) Average swiping speed. The ideal swiping speed deals with your self-confidence: shy people swipe slower, anxious people swipe too fast, and confident people swipe at the ideal speed.
- (3) Novel Neural Net Recognizes Spoken Words Better than Human Listeners. May 18, 2010 <<http://www.sciencedaily.com/releases/1999/10/991001064257.htm>>
- (4) RSA SecurID authenticators. RSA-The Security Division of EMC May 18, 2010 <<http://www.rsa.com/node.aspx?id=3049>>
- (5) Pretty Good Privacy (PGP) Email encryption program from PGP Corporation May 18, 2010 <http://www.pgp.com/products/desktop_email/index.html>
- (6) Verification is built up on the concept that the rhythm with which the user types is distinguishing. KeyLogPC KeyStroke Logger May 18, 2010 <<http://www.keylogpc.com/>>
- (7) RSA hardware authenticators. RSA-The Security Division of EMC May 18, 2010 .<<http://www.rsa.com/node.aspx?id=1158>>
- (8) To reduce risk exposure and comply with security regulations, companies rely on public key infrastructure (PKI) and digital certificates. VeriSign Inc. May 18, 2010 <<http://www.verisign.com/authentication/enterprise-authentication/pki-infrastructure-solutions/>>

- (9) Maximum tolerance for computer clock synchronization: this is the maximum time that can be tolerated between a ticket's timestamp and the current time at the Kerberos Distribution Center (KDC). Kerberos configuration by Jan De Clercq, October 8, 2004, Elsevier Digital Press <http://searchwindowserver.techtarget.com/news/article/0,289142,sid68_gci1014049,00.html>
- (10) User data collection can impact - usability as well. User data collection is the time period a person must spend to have her/his biometric reference template successfully created (i.e. enrolment and verification time) but can vary dramatically depending on the biometric device.
- (11) Cards are intended to operate within up to 10cm of the reader antenna at a frequency of 13.56 MHz. ISO/IEC 14443-1:2000. January 23, 2007 <http://www.iso.org/iso/catalogue_detail.htm?csnumber=28728>.
- (12) XyLoc provides full-time access control by determining a user's location and automatically locking the computer when the user is not physically present. Ensure Technologies Inc. May 18, 2010 <<http://www.ensuretech.com/>>
- (13) Cost of cryptographic operations (1,280-bit Rabin-Williams keys on 550 MHz K6). Mazieres, D. & Zhu, Y.: 2002. Machine Learning course - G22.3033-001. Topics in Computer System Security, New York University, NY (USA):

Operation	Time (seconds)
Encrypt	1.11
Decrypt	39.62
Sign	40.56
Verify	0.10
Total	81.39

- (14) Form of identifier presented by the user: PIN, memory card, etc.

2.3 The GOMS Model

The GOMS model, first proposed by Card *et al.* (1983), is the general term for a family of human information processing techniques that attempts to model and predict user behavior.

The acronym GOMS stands for Goals, Operators, Methods, and Selection Rules, as illustrated in Figure 2.30. Briefly, a GOMS model consists of descriptions of the Methods needed to accomplish specified Goals. The Methods are a series of steps consisting of Operators that the user performs. A Method may call for sub-Goals to be accomplished, so the Methods have a hierarchical structure. If there is more than one Method to accomplish a Goal, then Selection Rules choose the appropriate Method, depending on the context. Describing the Goals, Operators, Methods, and Selection Rules for a set of tasks in a formal way constitutes doing a GOMS analysis, or constructing a GOMS model.

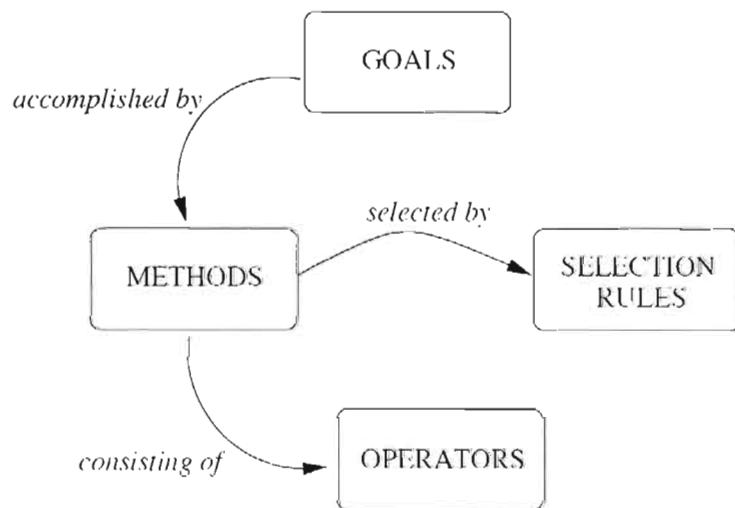


Figure 2.30: The GOMS Model (Grasso, 2008).

2.3.1 Engineering Models for Usable Interface Design

The major extant form of an engineering model for interface design is the GOMS model. The standard accepted technique for developing a usable system, empirical user testing, is based on iterative testing and design revision using actual users to test the system and help identify usability problems. It is widely agreed that this approach, derived from Human Factors, does indeed work when carefully applied (Card, 1983). The GOMS model is appropriate for modern software development practice since it is faster and cheaper than empirical user testing, especially when difficult-to-get domain experts are the target user group. Another important point is the concept of *engineering models* for usability, which has been evolving since the work by Card *et al* (1983). Similarly to the models used in other engineering disciplines, engineering models for usability produce quantitative predictions of how well humans will be able to perform tasks with a proposed design. Such predictions can be used as a substitute for actual empirical user data, making it possible to iterate through design revisions and evaluations much more rapidly. Furthermore, unlike purely empirical assessments, an engineering model for an interface design can capture the essence of the design in an inspectable representation, making it easier to reuse successful design insights in the future.

The overall scheme for using engineering models in the user interface design process is as follows: Following an initial task analysis and proposed first interface design, the interface designer would then use an engineering model as applicable to find the usability problems in the interface. However, because there are other aspects of usability that are poorly understood, some form of user testing is still required to ensure a quality result. Only after dealing with design problems revealed by the engineering model would the designer then go on to user testing. If the user testing reveals a serious problem, the design might have to be fundamentally revised, but again the engineering models will help refine the redesign quickly. Thus the slow and

expensive process of user testing is reserved for those aspects of usability that can only be addressed at this time by empirical trials. If engineering models can be fully developed and put into use, then the designer's creativity and development resources can be more fully devoted to more challenging design problems, such as devising entirely new interface concepts or approaches to the design problem at hand.

The major extant form of an engineering model for interface design is the GOMS model. The standard accepted technique for developing a usable system, empirical user testing, is based on iterative testing and design revision using actual users to test the system and help identify usability problems. It is widely agreed that this approach, inherited from Human Factors, does indeed work when carefully applied (Card, 1983). The GOMS model is appropriate for modern software development practice since is faster and cheaper than empirical user testing, especially when difficult-to-get domain experts are the target user group. Another important point is the concept of *engineering models* for usability, which has been evolving since the work of Card *et al.* (1983). Similarly to the models used in other engineering disciplines, engineering models for usability produce quantitative predictions of how well humans will be able to perform tasks with a proposed design. Such predictions can be used as a substitute for actual empirical user data, making it possible to iterate through design revisions and evaluations much more rapidly. Furthermore, unlike purely empirical assessments, an engineering model for an interface design can capture the essence of the design in an inspectable representation, making it easier to reuse successful design insights in the future.

The overall scheme for using engineering models in the user interface design process is as follows: Following an initial task analysis and proposed first interface design, the interface designer would then use an engineering model as applicable to find the usability problems in the interface. However, because there are other aspects of usability that are poorly understood, some form of user testing is still required to

ensure a quality result. Only after dealing with design problems revealed by the engineering model would the designer then go on to user testing. If the user testing reveals a serious problem, the design might have to be fundamentally revised, but again the engineering models will help refine the redesign quickly. Thus the slow and expensive process of user testing is reserved for those aspects of usability that can only be addressed at this time by empirical trials. If engineering models can be fully developed and put into use, then the designer's creativity and development resources can be more fully devoted to more challenging design problems, such as devising entirely new interface concepts or approaches to fit the design problem at hand.

2.3.2 GOMS: A Method for Cognitive Task Analysis

Advances in technology have augmented demands on the cognitive skills of workers, whereas the human system has remained moderately stable. Workers have been employing and/or operating complex computer technology. User interfaces (UIs) help users interact with programs and in their tasks. Users employ programs for performing their tasks. A UI should not reflect the structure of the underlying program, but the structure of the task domain and/or the task solution process. Users should not interact with the computer, but with their tasks. Cognitive Task Analysis (CTA) can boost human performance by guiding the development of tools and programs that support the cognitive processes required for a task (Chipman *et al.*, 2000). CTA is conducted for a wide variety of purposes such as system development, instruction and training, and human-computer interface design. For our particular study, CTA provides a description of the conceptual and procedural knowledge utilized by users as they perform, for example, authentication tasks such as accessing a protected network resource using a KBA method (e.g. security questions as an emergency access method). Based on extensive research among a variety of current CTA strategies (Kirwan *et al.*, 1992; Hollnagel, 2003; Kieras *et al.*, 1995; Diaper

and Stanton, 2003; Hackos and Redish, 1998), and in the context of the problem under consideration, user authentication, NGOMSL (Natural Goals, Methods, Selection Language) (Kieras, 1996) was selected as the most appropriate CTA method.

According to Proctor and Vu, (2005), GOMS is the most widely used method of cognitive task analysis. GOMS is both a performance model and a cognitive task analysis method. A GOMS model represents users' knowledge in a hierarchical stack structure consisting of *goals*, the state of affairs to be achieved; *operators*, elementary perceptual, motor, cognitive acts whose execution is necessary to change any aspect of a user's mental state or to affect the task environment; *methods*, sequences of operators that describe a procedure for accomplishing a goal; and *selection rules*, rules that determine which method is used when there are several methods for accomplishing the same goal. A GOMS task analysis is a process of creating a GOMS model by decomposing user task knowledge into GOMS components. The general strategy is similar to hierarchical task analysis: Begin by identifying the top-level user goals, emphasizing breadth over depth. Then refine each goal into sub-goals, methods, and operators. It differs from hierarchical task analysis in that a GOMS model has a specific format. GOMS is specifically designed to represent procedural knowledge of well-learned cognitive tasks. Additionally, a GOMS model can be translated into executable programs for evaluating the consistency of the model and obtaining quantitative measures of the interface being designed.

In the analysis and implementation stages, a GOMS model of existing or proposed tasks can be used to determine if the functional requirements are derived from the tasks performed by users: Every task goal should have a specific method for achieving the goal. The GOMS analysis can identify benchmark tasks (i.e. set of benchmark cases that represent important user tasks) and establish performance criteria for usability tests at later stages. A GOMS model also can be used to

determine the consistency of procedures: Similar goals should be accomplished with similar tasks. When more than one method is provided, the GOMS analysis can determine whether there are selection rules that determine the method to be used. When a GOMS model is constructed, quantitative assessments of the design can be made from assumptions about operator execution times. For example, a GOMS model can be used to evaluate alternative design concepts in terms of time to learn to use the system, time to perform some tasks, and possibly time for recovering from errors.

GOMS analyses and models are most appropriate for user tasks having well-defined goals (e.g. logging into the system) and requiring the application of learned cognitive skills. GOMS also does not permit analysis of interface issues related to the layout of components, readability of text, and so on. Instead it is focused on the procedures that the user must learn and execute when performing tasks.

It is worth noting that GOMS represents only the procedural aspects of usability. GOMS models can predict the procedural characteristics of usability; these concern the amount, consistency, and effectiveness of the procedures that users must pursue. Since the usability of numerous systems depends profoundly on the simplicity and effectiveness of the procedures, the GOMS model has significant value in guiding interface design. The reason why GOMS models can predict these characteristics of usability is that the methods for achieving user goals has a tendency to be strongly constrained by the design of the interface, making it possible to build a GOMS model given just the interface design, prior to any prototyping or user testing (Kieras, 1996).

Clearly, there are other important characteristics of usability that are not related to the procedures entailed by the interface design. These concern both lowest-level perceptual issues like the font's readability on Cathode-Ray Tubes (CRTs)⁴⁷, and also

⁴⁷ CRT is the technology used in most televisions and computer display screens.

very high-level issues such as the user's conceptual knowledge of the system (e.g. whether the user has an appropriate mental model) (e.g. Kieras and Bovair, 1984), or the degree to which the system fits properly into an organization (i.e. the social or organizational impact of the system and the resulting influence on productivity) (John and Kieras, 1994). The lowest-level issues handle fine in terms of standard human factors methodology, while understanding the higher-level concerns is at present an issue of good judgment on the part of the practitioner and the higher-level task analysis techniques (Kieras *et al.*, 1995).

For detailed information on why to choose the GOMS model over other CTA techniques, see Appendix E.1, NGOMSL versus CPM-GOMS, KLM, and CMN-GOMS: Which Model to Use?

2.3.3 How to Develop a GOMS Model?

In all GOMS analysis techniques, the analyst must start with a list of high-level user goals. Typically, this list of goals can be obtained from other task analyses, including observations of users of similar or existing systems. The GOMS steps are described as follows:

- Identify user's goals (The analyst can then express in a GOMS model how the user can accomplish these goals with the system being designed). A goal is something that the user tries to accomplish. The analyst attempts to identify and represent the goals that typical users will have. A set of goals usually will have a hierarchical arrangement in which accomplishing a goal may first require -accomplishing one or more sub-goals. A goal description is an action-object pair in the form- <verb noun> (e.g. *Access a file*). The verb can be complicated if it is necessary to distinguish between methods (see below on selection rules). Any parameters or modifiers, such as where a “to-be-deleted” word is located, are represented in the task description.

- Define methods (Write Method for accomplishing Goal – may invoke sub-goals). The example consists of a list of methods for each system. For example

Select a word expressed in the NGOMSL notation is shown in Figure 2.31.

```
Method for goal: select word
Step 1. Locate middle of word.
Step 2. Move cursor to middle of word.
Step 3. Double-click mouse button.
Step 4. Verify that correct text is selected
Step 5. Return with goal accomplished.
```

Figure 2.31: Method for selecting a word (Kieras, 1996).

- Define Operators (Standard Primitive External Operators, Standard Primitive Mental Operators, and Analyst-Defined Mental Operators). They are mostly determined by the hardware and lowest-level software of the system (e.g. `Move finger to the USB fingerprint reader`). External operators are the observable actions through which the user exchanges information with the system or other objects in the environment (e.g. a perceptual operator such as `Read your online ID from a screen`). Mental operators are the internal actions performed by the user. In the notation system presented here, some mental operators are “built in”. These operators correspond to the basic mechanisms of the cognitive processor, the cognitive architecture.

A particular task analysis assumes a particular level of analysis which is reflected in the “grain size” of the operators. If an operator will not be decomposed into a finer level, then it is a primitive operator. But if an operator will be decomposed into a sequence of lower-level, or primitive, operators, then it is a high-level operator. Exactly which operators are

primitives depends on the finest grain level of analysis desired by the analyst. Some typical primitive operators are actions like pressing a button- or moving the hand. All built-in mental operators are primitive. High-level operators would be gross actions, or stand-ins for more detailed analysis, such as LOG-INTO-SYSTEM. The analyst recognizes that these could be decomposed, but may choose not to do so, depending on the purpose of the analysis (Kieras, 1996).

In the case of the Standard Primitive External Operators, the designer defines the primitive motor and perceptual operators based on the basic actions required by the system being analyzed. These correspond straightforwardly to the physical and some of the mental operators used in the Keystroke-Level Model (KLM). KLM uses only keystroke level operators, no goals, methods or selection rules. The analysis simply lists the keystrokes, mouse movements, and mouse-button presses that a user must perform to accomplish a task, and then uses a few simple heuristics to place a single type of trivial “mental operator” which approximates many kinds of internal cognitive actions (e.g. think of).

Standard Primitive Mental Operators is in turn divided into *Flow of Control* and *Memory storage and retrieval*. *Flow of Control* represents a sub-method which is invoked by declaring its goal: Accomplish goal: <goal description>. Control passes to the method for the goal, and returns here when the goal has been accomplished. The operator Return with goal accomplished marks the end of a method. A decision is represented by a Decide operator; a Decide operator contains either one IF-THEN conditional with an optional ELSE, or any number of IF-THEN conditionals. In *Memory storage and retrieval*, the memory operators reflect the distinction between

Long Term Memory (LTM) and Working Memory (WM), They are usually used in computer operation tasks as follows:

Recall that <WM-object-description>

Retain that <WM-object-description>

Forget that <WM-object-description>

Retrieve-from-LTM that <LTM-object-description>

Finally, Analyst-Defined Mental Operators represent- psychological processes that are too complex to be practical to indicate as methods in the GOMS model, so -the designer can circumvent these processes by defining operators that act as place holders for the mental activities. They are mostly high-level operators. For example, Think-of <description> represents a process of thinking of a value for some parameter designated by <description> and putting the information into working memory (Kieras, 1996).

- "The purpose of a Selection rule is to route control to the appropriate method to accomplish a goal (Kieras, 1996). If there is more than one method for a goal, then a selection rule is logically required as shown below:

Selection rule set for goal: <general goal description>

If <condition> Then accomplish goal: <specific goal description>.

If <condition> Then accomplish goal: <specific goal description>.

...

Return with goal accomplished.

An example of the GOMS model is shown below:

```
GOAL: CLOSE-WINDOW
. (select GOAL: USE-MENU-METHOD
. MOVE-MOUSE-TO-FILE-MENU
. PULL-DOWN-FILE-MENU
. CLICK-OVER-CLOSE-OPTION
. GOAL: USE-CTRL-W-METHOD
. PRESS-CONTROL-W-KEYS)
```

For a particular user:

```
Rule 1: Select USE-MENU-METHOD unless another rule applies
Rule 2: If the application is GAME,
        select CTRL-W-METHOD
```

2.3.4 Natural GOMS Language (NGOMSL)

In psychology, researchers fit the parameters of their models to data they've collected on the task they're studying. But in interface design, system developers need quantitative *a priori* predictions for systems that have not yet been built. Thus, HCI researchers have done extensive theoretical and empirical work to estimate parameters that are robust and reliable across tasks and can be used *without further empirical validation* to make predictions (e.g. usability testing). NGOMSL is one of the GOMS models that does quantitative *a priori* predictions for systems that have not yet been built.

NGOMSL is a structured natural language notation for representing GOMS models and a procedure for constructing them (Kieras, 1996). An NGOMSL model is in program form, and provides predictions of operator sequence, execution time, and time to learn the methods. An analyst constructs a- NGOMSL model by performing a top-down, breadth-first expansion of the user's top-level goals into methods, until the methods contain only primitive operators, typically keystroke-level operators (e.g. Click on "Sign In" button with left mouse button). Like CMN-GOMS, NGOMSL models explicitly represent the goal structure, and therefore can represent high-level goals.

The NGOMSL technique refines the basic GOMS concept by representing methods in terms of a cognitive architecture called Cognitive Complexity Theory (CCT) (Kieras and Polson, 1985). This cognitive theory allows NGOMSL to incorporate internal operators such as manipulating working memory information or setting up sub-goals. Because of this, NGOMSL can also be used to estimate the time

required to learn how to achieve tasks. An example of a NGOMSL model taken from John and Kieras (1996) is shown in Figure 2.31.

The GOMS modeling technique has proven extremely successful in developing accurate cognitive task models (Williams and Voigt, 2004). Some of the types of applications in which cognitive task models have been applied in their research include assessing human-computer interaction complexity, determining the productivity of human-computer interfaces, and analyzing an interface design to determine whether methods can be automated.

For detailed information on why to choose NGOMLS over other GOMS models such as CPM-GOMS, KLM, and CMN-GOMS, see Appendix E.1, NGOMSL versus CPM-GOMS, KLM, and CMN-GOMS: Which Model to Use?.

2.3.4.1 Cognitive Complexity Theory

When you have learned to perform a task with a particular interface and have to switch to- doing the same task with a new interface, how much better off will you be than someone just learning to do the task with the new interface? That is, how much is the knowledge gained from using the old interface “transferred” to using the new interface? Cognitive Complexity Theory (CCT) (Bovair *et al.*, 1990; Kieras and Polson, 1985) is a psychological theory of transfer of training applied to HCI. It seeks to decompose user goals for completing computer tasks with a greater degree of granularity than GOMS in order to obtain more accurate predictions of how long it will take users to learn to complete tasks online with fewer errors. In contrast to GOMS models, CCT investigates learners rather than skilled users. CCT has been shown to provide good predictions of execution time, learning time, and transfer of procedure learning.

CCT assumes a simple serial stage architecture in which Working Memory (WM) triggers production rules that apply at a fixed rate. These rules alter the

contents of working memory or execute primitive external operators such as making a keystroke. GOMS methods are represented by sets of production rules in a prescribed format. Learning procedural knowledge consists of learning the individual production rules. Learning transfers from a different task if the rules have already been learned. The complexity of a task will be reflected in the number and content of the production rules. The time it takes to learn a task is a function of the number of new rules that the user must learn; if the user already has a production, and a new task requires a rule that is similar, then the rule for the new task need not be learned. It is important to note that some predictions about errors and speedup with practice can also be collected from the contents of the production rules.

The association between the NGOMSL notation and the CCT architecture is in fact direct: There is basically a one-to-one relationship between statements in the NGOMSL language and the production rules for a GOMS model written in the CCT format. Therefore, the CCT prediction results can be used by NGOMSL models to estimate not only execution time like Keystroke-Level Model (KLM) and CMN-GOMS⁴⁸, but also the time to learn the procedures. CMN-GOMS stands for Card, Moran and Newell GOMS.

CCT and NGOMSL models have been empirically validated at the KLM of analysis (operators like DETERMINE-POSITION and CLICK-MOUSE-BUTTON); therefore, models at that level can generate trustworthy quantitative estimates. Because NGOMSL models specify methods in program form, they can characterize the procedural complexity of tasks- both in terms of how much must be learned- and how much has to be executed.

CCT employs production systems in the form IF (condition) THEN (action) in order to explain the cognitive demands related to task performance. The condition

⁴⁸ CMN-GOMS adds hierarchical structure to the KLM version of GOMS. Tasks are organized as a series of goals and sub-goals and operators are organized into subroutines called methods. CMN-GOMS can provide task execution times and afford- a better view of the task structure than KLM.

component of production rules relates to either the contents of WM or environmental factors (e.g. screen display). The action component relates to manipulations of either the environment (e.g. key presses) or the contents of WM (e.g. deleting current goals). The clauses included within the “condition” component are combined using logical AND. If the pattern of goals, notes, and external information in WM matches the condition clauses, the rule is said to “fire” and the action operators are executed. Current goals and variables have to be stored in WM, and it is -on this basis that task demands are estimated. Once added to WM by a production, GOALS and NOTES must be preserved in WM until deleted by later productions. A number of production systems may be produced with the purpose of describing complete task performance. The sequence in which these productions are performed may depend upon selection rules that specify different methods of achieving the current task goals (Card *et al.*, 1983).

To illustrate a CCT production rule, consider the following example:

- Editing with VI (Visual editor)⁴⁹:
 - Production rules are in LTM.
 - Model working memory as attribute-value mapping:


```
(GOAL perform unit task)
(TEXT task is insert space)
(TEXT task is at 5 23)
(CURS0R 8 7)
```
- Rules are pattern-matched to WM.
- e.g., LOOK-TEXT task is at %LINE %COLUMN is true, with LINE = 5
COLUMN = 23.

⁴⁹ VI (visual) editor is available on all major computer systems. VI is a display oriented, interactive text editor which allows a user to create, modify, and store files on the computer via a terminal.

2.3.4.2 NGOMSL Steps Development Process

This section lists the steps involved in the development of the NGOMSL model. Each step contains information on which processors its operators require. *Interstep mental operators* (e.g. Recall_Passcode) require use of the cognitive processor, but *intrastep mental operators* (e.g. Return_with_goal_accomplished) do not. The steps are the following:

- Generate Task Description;
- Describe a List of High-Level User Goals;
- Define Operators, Write Methods, and Selection Rules for Accomplishing Goals;
- Estimate Pure-Learning Time.

2.3.5 Learning Time Predictions

NGOMSL models have been demonstrated to be superior predictors of the time it takes to learn how to use a system, keeping in mind that what is predicted is the Pure Learning Time for the *procedural knowledge* represented in the methods. As already mentioned, the user is assumed to already know how to execute the operators; the GOMS methods do not represent the knowledge involved in executing the operators themselves, but rather only represent the knowledge of which operators to apply and in what order to accomplish the goal. Innovative interface technology frequently results in new operators; moving the cursor with a mouse was a new operator, and selecting objects with an eye-movement tracker or manipulating 3D objects and flying about in virtual space with data-glove gestures will be new operators as these technologies move into the workplace. Undoubtedly, the time to learn how to execute new operators is a decisive aspect of the value of new interface devices, but a GOMS model that *assumes* such operators cannot predict their learning

time. The time to learn new operators themselves would have to be measured, or simply not included in the analysis.

The total elapsed time to learn to use a system depends not simply on how much procedural knowledge must be learned but on how much time it takes to complete the training curriculum itself. It means that nearly all learning of computer use occurs in the context of the new user performing tasks of some sort, and this performance would take a certain amount of time even if the user were fully trained. As a result, the total learning time includes the time to execute the training tasks in addition to the extra time required to learn how to perform the tasks, the Pure Learning Time (PLT). As Gong (1993) has demonstrated, training-task execution times can be predicted from a GOMS model of the training tasks.

The fundamental empirical result is that the time needed to learn a particular procedure is roughly linear with the number of NGOMSL statements that must be learned. Thus, the PLT for the methods themselves can be estimated just by counting the statements and multiplying by an empirically determined coefficient. The transfer of training effects can be calculated by deducting the number of NGOMSL statements in methods that are identical, or highly similar, to ones already known to the learner (Bovair *et al.*, 1990). This description of interface consistency in terms of the quantitative transferability of procedural knowledge is possibly the most important contribution of the existing CCT research and the NGOMSL technique.

A supplementary element of the PLT is the time required to memorize chunks of declarative information required by the methods, such as the menu names under which commands are found. Such items are assumed to be stored in LTM, and while not rigorously part of the GOMS methods, are required to be in LTM for the methods to execute correctly. Including this component in learning time estimates is an approach to representing the learning load imposed by menu or command terms.

“The validity and utility of the learning time estimates depend on the general requirements of the learning condition. Obviously, if the learner is engaged in problem-solving, or in an unstructured learning situation, the time required for learning is more variable and ill-defined than if the learner is trained in a tightly controlled situation” (John and Kieras, 1996). Also, it seems logical that in spite of the learning situation, systems whose methods are longer and more complex will require more time to learn them, because more procedural knowledge has to be acquired, either by explicit study or inferential problem-solving. Summarizing the above discussion of estimating learning using the values determined, Gong (1993) presents the following

$$\text{Total Learning Time} = \text{Pure Method Learning Time} + \text{Long Term Memory Item Learning Time} + \text{Training Procedure Execution Time}$$

$$\text{Pure Procedure Learning Time} = \text{NGOMSL Method Learning Time} + \text{LTM Item Learning Time}$$

$$\text{NGOMSL Method Learning Time} = 17 \text{ sec}$$

$$\text{No. of NGOMSL Statements to be Learned}$$

$$\text{LTM Item Learning Time} = 6 \text{ sec} * \text{Number of LTM Chunks to be Learned}$$

These formulas give a pure procedure learning time estimate for a whole set of methods in an usual learning situation, assuming no previous knowledge of any methods, and assuming that learning the appropriate command words for the two menu terms will require learning three chunks each (John and Kieras, 1994).

2.3.6 Execution time predictions

As already mentioned, the execution time for a task is predicted by simulating the execution of the methods required to perform the task as shown below:

$$\text{Execution Time} = \text{NGOMSL statement time} + \text{Primitive External Operator Time} + \text{Waiting Time}$$

The execution time predictions are founded on the sequence of operators executed while carrying out the benchmark tasks. Execution time might be approximated by a constant, by a probability distribution, or by a function of some parameter. For example, the time to type a word might be approximated by a constant (e.g., the average time for an average typist to type an average word), or a statistical distribution, or by a function involving the number of letters in the word and the time to type a single character (which could, in turn, be approximated by a constant or a distribution) (John and Kieras, 1994). The precision of estimates obtained from a GOMS model depends on the precision of this assumption and on the precision of the duration estimates.

2.3.7 NGOMSL Methodology

According to Kieras (2006), the NGOMSL methodology starts with the following:

- Top-down breadth-first task decomposition:
 - Start with the user's top-level goals.
 - Write a step-by-step procedure for accomplishing each goal in terms of sub-goals or keystroke-level operators.
 - Use NGOMSL syntax for the procedure.
 - Recursively write a method for each sub-goal until all methods contain only keystroke-level operators.
 - Write a selection rule to specify which method to use if more than one for a goal.
- Count number of statements in methods to predict learning time.
 - Consistency can be directly reflected by the presence of re-used sub-methods, reducing learning time.
 - Similar methods also reduce learning time.
- For a specific task scenario, count number of statements and operators executed to predict execution time.

2.3.8 NGOMSL Limitations

The GOMS model has a number of limitations, the most significant of which is that the predictions are only valid for expert users who do not make any errors. This is in fact a significant deficiency, because even expert users will make mistakes. Given that one of the goals of HCI is to seek - maximum usability for all users, including novices, this is a grave insufficiency in the model. However, at the same time, the only GOMS model that mitigates this deficiency is NGOMSL, which attempts to model the time required to learn a task. As already mentioned, the GOMS model has been used in this research work as a cognitive task analysis tool and not for evaluating a specific user interface. Therefore, this dissertation is not interested in measuring the user performance related to user authentication methods, but rather in identifying and understanding the cognitive processes involved specifically in those types of interaction.

Another limitation is that in NGOMSL models all tasks are goal-directed, neglecting the problem-solving nature of some tasks. They do not take into account individual differences among users. One of the main premises of this thesis is that tasks should be goal-directed (e.g. make an electronic funds transfer) and include the user authentication portions in them. The goal-directness "feature" is in fact very appropriate to the type of tasks described in this dissertation. Computer security applications, especially the user authentication applications, -usually have a clear-cut, narrow user interface and function. This entails limited application features to users, but at the same time, simplicity is gained.

GOMS models cannot provide information on how valuable or pleasant the product under design will be. Finally, as already mentioned, GOMS does not address the social or organizational impact of the product under development.

2.4 Usability and Usable Security

Usability has been defined differently in several standards (ISO/IEC 9126, 2004), (ISO 9241-11, 1998)- and (IEEE 1061, 1998). Each of these standards emphasizes somewhat different sets of usability factors, such as effectiveness, efficiency, learnability, or user satisfaction. Thus, a more comprehensive model of usability should include both process-related and product related usability characteristics such as effectiveness, efficiency, satisfaction, security, and learnability. Moreover, usability is a generally relative measure of whether a software product enables a particular set of users to achieve specified goals in a specified context of use (Abzan *et al.*, 2003). On the other hand, - according to Jøsang and Patton (2003), Usable Security deals with how security information should be handled in the UI. Both usability and security can vary depending on the context of use that includes user profiles (i.e., who are the users), task characteristics, hardware (including network equipment), software, and physical or organizational environments (Seffah *et al.*, 2006). As previously mentioned, Usable Security is imperative from the user's perspective, from the developer's perspective, and from the management's perspective (e.g., software with weak security support can be a major constraint to the usability of the system).

2.4.1 Usability Inspection Methods

Usability evaluation methods can be divided into three categories: Test, Inspection, and Inquiry. Each category can be applied to one or more phases of the design lifecycle as follows:

- *Testing method*: It makes use of representative users to work on typical tasks using the system (or the prototype) (e.g., the conventional usability testing), and their performance is usually measured.

- *Inspection method:* In this approach, *usability* experts – and sometimes software engineers, or domain experts - inspect usability related aspects of a user interface (e.g., Heuristic Evaluation) (Nielsen, 1994), Usable Security Symmetry (Braz *et al.*, 2010). One interesting characteristic of this method when compared to the other two categories is that they can be used at any stage of design, from product definition to final design. Usability inspection methods -are particularly efficient in terms of a high benefit-cost ratio, and are able to find many usability problems that are overlooked by user testing (Karat *et al.*, 1992; Desurvire *et al.*, 1992; Desurvire, 1992).
- *Inquiry method:* This method collects information regarding users' preferences, desires, and behavior (e.g., a focus group), and aims to formulate the requirements of a design.

There is at present a multiplicity of methods used to evaluate usability. To choose a specific method you must consider human and facilities resources, costs, time constraints, and the suitability of the method for the product at hand.

This section presents an overview of the three most representative usability inspection methods relevant to this thesis: Heuristic Evaluation, Cognitive Walkthrough, and the GOMS Model. These methods are recognized as the most actively employed and researched usability inspection methods by the HCI Research and Industry communities according to Hollingsed and Novick (2007) and Kieras (2006).

2.4.1.1 General usability principles ("heuristics") for User Interface Design

One of the most widely recognized usability evaluation methods, "Heuristic Evaluation" is a usability inspection method employed to analyze the usability problems of a user interface or a system against a set of ten established - principles

called "heuristics" (Molich and Nielsen, 1990). These heuristics are more based on empirical data than on specific usability guidelines. After evaluating numerous sets of heuristics, Nielsen (1994) developed a set of heuristics as follows:

- *Visibility of system status:* The system should always keep users informed about what is going on, through appropriate feedback within a reasonable time.
- *Match between system and the real world:* The system should speak the users' language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- *User control and freedom:* Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo;
- *Consistency and standards:* Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- *Error prevention:* Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- *Recognition rather than recall:* Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- *Flexibility and efficiency of use:* Accelerators - unseen by the novice user - may often speed up the interaction for the expert user such that the system can

cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

- *Aesthetic and minimalist design*: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- *Help users recognize, diagnose, and recover from errors*: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- *Help and documentation*: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search for, focused on the user's task, - concrete in regards to the steps to be carried out, and not overly complex.

An example of such a (usability) heuristic is as follows:

Visibility of system status: The system should always keep users informed about what is going on, through appropriate feedback within a reasonable timeframe.

Heuristic evaluation is not limited to the list of heuristics above. In fact, the list of heuristics can be as long as the evaluators consider appropriate for the task at hand. For instance, you may create a specific list of heuristics for specific audiences, like senior citizens, children, or disabled users, based on a review of the literature (UPA, 2010).

Heuristic evaluation falls into the general category of usability inspection methods, and it is considered a "discount usability engineering" method (i.e. smaller and cheaper usability studies for projects with small budgets for usability).

Heuristic Evaluation is performed by one or more evaluators, preferably experts, who often possess the knowledge to design and carry out comprehensive performance tests with human subjects and the ability to analyze the resultant data. Their training is usually an advanced degree in human factors, behavioral science, industrial engineering, human-computer interaction, industrial design, computer science, or a related field. The greatest results are achieved by evaluators who are usability specialists with domain knowledge. Independent research has found heuristic evaluation to be extremely cost-efficient, confirming its value in circumstances where limited time or budgetary resources are available. "Overall, the heuristic evaluation technique as applied here produced the best results. It found the most problems, including more of the most serious ones, than did any other technique, and at the lowest cost" (Jeffries *et al.*, 1991).

Another important factor to be taken into consideration is the ideal number of evaluators to perform the evaluation. According to Nielsen (1994), it is recommended to use of three to five evaluators, since different evaluators tend to find different problems.

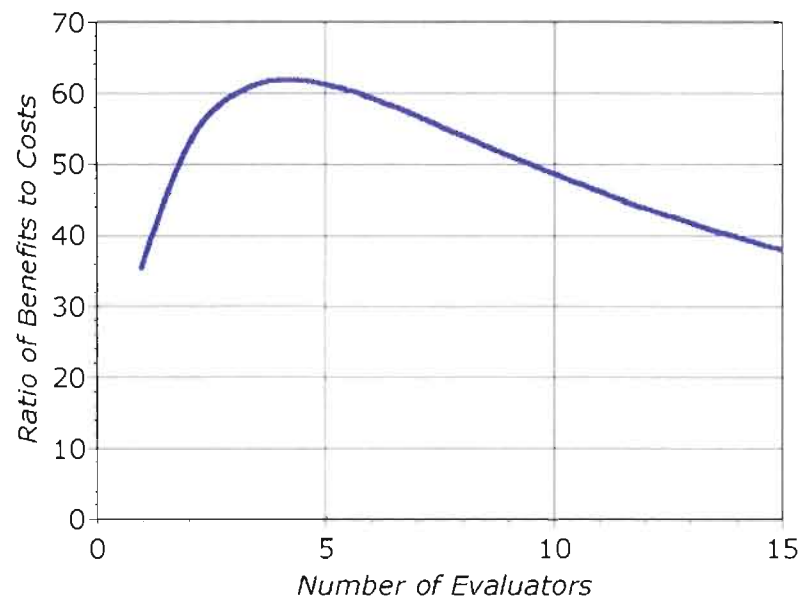


Figure 2.32: The optimal number of usability evaluators: benefits versus costs (Nielsen, 1994a).

The curve showing how many times greater the benefits are in comparison to the costs associated with heuristic evaluation (sample project) is presented in Figure 2-32. The optimal number of evaluators in this example is four, with benefits that are 62 times greater than the costs. It also shows the varying ratio of the paybacks to the costs for different numbers of evaluators in the sample project mentioned by Nielsen (1994a). In fact, the curve illustrates that the optimal number of evaluators is four, validating “the general observation that heuristic evaluation seems to work best with three to five evaluators. In the example, a heuristic evaluation with four evaluators would cost \$6,400 and would find usability problems worth \$395,000.”

With regards to Expert review versus usability testing, Molich and Dumas (2008) argue that Expert reviews with highly experienced practitioners may be more efficient than usability tests, in terms of number of issues found as a function of resources expended.”

The next paragraphs describe the required procedures to perform a Heuristic Evaluation according to UPA (2010):

- Decide which aspects of a product and what tasks you want to review. For most products, you cannot review the entire user interface, so you need to consider what type of coverage will provide the most value.
- Decide which heuristics will be used.
- Select a team of three to five evaluators (you can have more, but the time to aggregate and interpret the results will increase substantially) and give them some basic training on the principles and process.
- Create a list of representative tasks for the application or component you are evaluating. You might also describe the primary and secondary users of your product if the team is not familiar with the users.
- Ask each evaluator to perform the representative tasks individually and list where the product violates one or more heuristics. After the evaluators work through the tasks, they are asked to review any other user interface objects that were not directly involved in the tasks and note violations of heuristics. You may also ask evaluators to rate how serious the violations would be from the users' perspective.
- Compile the individual evaluations and ratings of seriousness.
- Categorize and report the findings so they can be presented effectively to the product team.

The advantages of heuristic evaluation are that it is inexpensive, intuitive (i.e. we apply a set of predefined rules/heuristics), very easy to plan, and -can be used early in the design process. Feedback can also be obtained early in the design process. Assigning the right heuristic can aid to recommend the best corrective measures to the designer. The disadvantage is that there is a focus on problems rather than on

solutions. Also if the wrong heuristics are assigned to potential problems, it will mislead designers into applying the wrong solutions to the problems.

2.4.1.2 Cognitive Walkthrough

Cognitive walkthroughs are performed at any stage of design using a prototype, a conceptual design document, or the final product. This is a more specific version of a design walkthrough, focusing on cognitive principles. Based on a user's goals, a group of evaluators steps through tasks, evaluating at each step how difficult it is for the user to identify and operate the interface element most relevant to their current sub-goal and how clearly the system provides feedback for that action.

Cognitive walkthroughs take into consideration the user's thought processes that contribute to decision making, such as memory load and the ability to reason. For example, finding the Usability First website can be broken down to several levels of tasks. At a general level, it requires opening up a browser, remembering the Uniform Resource Locator (URL), and typing it in the text box at the top of your browser. Or, if you do not remember the URL, you must choose a search engine, think of a search term, view the results, scroll through the results, and then click on the link. Each of these actions can be further decomposed. This approach is intended specifically to help understand the usability of a system for first-time or infrequent users, that is, for users in an exploratory learning mode.

But how does one- perform a Cognitive Walkthrough? The following procedures must be taken into consideration, according to Richardson (2000):

- Define the inputs:
 - Identify the users and tasks, create a description of the interface (screenshot or prototype), and define the action sequences for completing each task.

- Gather the walkthrough team:
 - A facilitator for the discussion and a scribe for recording information.
 - Participants walk through the tasks relating to the user interface.
 - Participants probe the task on hand (e.g. does the label for the correct action match the user's goal? What is the user's goal?, etc.).
- Walk through the action sequences for the task(s).
- Record critical information (i.e. the happy (or failure) path(s), problems, etc.).
- Review the interface to fix the problems (i.e. re-implement rapid prototype or new screenshots).
- Repeat. (i.e. iterative design, which means prototyping, testing, analyzing, and refining the user interface, and repeat if needed).

2.4.1.3 GOMS Model

GOMS is a family of techniques proposed by -Card *et al*- (1983) for modeling and describing human task performance. However, GOMS models have also been - used as a usability inspection method as shown by Card *et al*. (1983), Lecerof and Paternò (1998), Schrepp (2010), John and Kieras (1996), and Kieras (2006).

GOMS is an acronym that stands for Goals, Operators, Methods, and Selection Rules, the components of which are used as the building blocks for a GOMS model. Goals represent the goals that a user is trying to accomplish, usually specified in a hierarchical manner. Operators are the set of atomic-level operations with which a user composes a solution to a goal. Methods represent sequences of operators, grouped together to accomplish a single goal. Selection Rules are used to decide which method to use for -achieving a goal when several are applicable.

According to John and Kieras (1996), the GOMS model has been one of the few extensively recognized theoretical concepts in HCI, and has been used in real-world design and evaluation situations. The paper by John and Kieras (1996) actually summarizes the previous work on GOMS by offering a unified analysis of GOMS models, showing how these models can be used in design, and describing several examples of the application of GOMS to the design and evaluation of the interfaces for a multiplicity of real-world systems such as the Computer-Aided Design (CAD) system for mechanical design, the Space operations database system, and the Mouse-driven text editor, among others.

One recent application of the GOMS model as a usability inspection method is the Guideline compliance, a necessary but insufficient condition to guarantee the usability of web units by disabled users, since efficiency-related issues can be as exclusive for disabled users as violations to basic guidelines. This paper shows that Goals, Operators, Methods and Selection rules (GOMS) analysis, which is an established method in user interface design, can be adapted to evaluate the efficiency of interface designs for disabled users. As examples, several GOMS models for the interaction behavior of disabled users with web units are described, showing how such models can be used to answer concrete accessibility-related questions. Advantages and limitations of GOMS analyses are also discussed.

Another relatively recent application of using GOMS for evaluating user interface is “A GOMS Model for Keyboard Navigation in Web Pages and Web Application” developed by Schrepp and Fischer (2006). Generally speaking, technology should be accessible and usable by users effortlessly, including users with disabilities. According to the study authors, 81% of the websites were incompliant with basic standards for accessibility as recommended by the World Wide Web

(W3C) Consortium⁵⁰. Also the study infers that a task that can be carried out in one minute using a keyboard- might need 5 to 10 minutes using a mouse. Many disabled (and even expert) users favour handling desktop and Web applications by using the keyboard, as it is often faster than using a mouse. Therefore, offering efficient keyboard support is crucial to increasing the usability of those applications. To achieve this, it is necessary to make use of a method to evaluate mouse and keyboard navigation; the GOMS model is a very appropriate method for doing this, as it allows for the comparison of diverse methods to operate a Web application. According to the GOMS model, “the average time taken to carry out a task such as following a link will be 11.83 seconds using a mouse and 28.56 seconds using the keyboard. If the time taken to work on a Web page using a keyboard should not be twice or even more than twice the time taken using a mouse, then additional keyboard support should be implemented for this page. Furthermore, the GOMS models can be used to verify whether the amount of keyboard support for a Web application is adequate to guarantee that there are no inadmissible drawbacks to keyboard users.

2.4.1.4 Additional Usability Evaluation Methods

It is worth noting that there are other recent usability evaluation methods (Law *et al.*, 2008), not inspection methods, that have been used on a limited scale in industry settings, but they are worth mentioning in the context of this thesis. This section gives an overview of these methods as follows:

- *Condensed Contextual Inquiry (CCI)* (Kantner and Keirnan, 2003): Traditional CI requires long hours with each user, which usually can take a full day per visit. Although this long session time allows researchers to gather much important information, organizations refrain from spending the time to

⁵⁰ W3C <<http://www.w3.org/>>

gather and analyze so much data. Additionally, organizations are reluctant to interrupt employees for such an extended period of time. This method is basically a one-on-one observation of work practice in the user's real context. CCI in fact identifies -a more restrained suite of concerns to examine than the traditional version of a Context Inquiry, which often requires -a full day per visit. CCI tries to accommodate the restricted time that product development teams have to learn about users' work processes and motivations. *The primary difference between traditional CI and CCI* is the restricted nature of the work under observation. CCI looks at the bigger picture of the users' motivations and contextual artifacts for accomplishing work. Therefore, this method is not suitable for designing a complex system such as authentication management applications.

- *Remote Usability Testing (RUT)*: This evaluation method remotely gathers - key data from larger populations (e.g., 40, 80, 120 and so on) in a single usability testing. Participants are observed by usability evaluators in real-time sessions to collect instantaneous behaviors and comments. RUT collects performance measures such as the number of errors, types of errors, time on task, and automatic data collection regarding user behavior such as logging (Client and Server sides), browser logs, etc. According to Law *et al.* (2008), these requirements have led evaluators to conduct usability testing with larger sample sizes as mentioned. The RUT method is typically used to compare two or more products, designs, or product features (e.g., from competitors). According to Paternò and Santoro (2008), the key dimensions for analyzing the different methods for assessing remote usability evaluation are:
 - The type of interaction between the user and the evaluator.
 - The platform used for the interaction (desktop, mobile, vocal, etc.)

- The techniques used for collecting information about the users and their behavior (graphical logs, voice/or Webcam recordings, eye-tracking, etc.).
- The type of application considered in terms of the implementation environment (Web, java-based, .NET, etc.).
- The type of the evaluation results (task performance, emotional state) provided.
- *Longitudinal Usability Evaluations:* Longitudinal studies follow a user over an extended period of time (i.e., a month or two), with observations made at periodic intervals. After each experiment, for instance, participants are typically asked to answer two questionnaires: one questionnaire intended to assess the hardware, software, and technological issues encountered- and the other intended to assess the extent and quality of the cooperation between users. According to Sy (2009), the research methods most appropriate for longitudinal studies are:
 - Diary studies.
 - Usage logs and clickstream/instrumented data analysis.
 - Periodic field ethnography.
 - Periodic interviewing (both on site and remote).
 - Periodic usability testing (both on site and remote).
 - Retrospectives.

There is no method that is the most appropriate- for longitudinal studies, but what is critical here is to triangulate data compilation from several methods.

An example is a longitudinal laboratory-based usability evaluation of a health care information system (Kjeldskov *et al.*, 2010). The goal of this study was to inquire into the nature of usability problems experienced by novice and expert users, and to see to what extent usability problems of a health care

information system would or would not disappear over time, as the nurses got more familiar with it. The authors conducted a longitudinal study with two main sub-studies: a usability evaluation was conducted with novice users when an electronic patient record system was being employed in a large hospital. After the nurses had used the system in their daily work for 15 months, the authors repeated the evaluation. The results demonstrate that time does not heal. Even though some problems were not -as severe, they still remained after 1 year of extensive use.

2.4.2 Usable Security Principles and Guidelines

The research work of Whitten and Tygar (1998) and Whitten and Tygar (1999) on the usability of the Pretty Good Privacy (PGP), a public key encryption application, is considered pioneering in the Usable Security field.

To date, there is no theoretical framework to provide an inspection method that considers security and usability synergistically for *user authentication methods*. However, the HCI/Sec community has been steadily developing research work in usable security *guidelines and standards* for computer security software in general, such as Computer Security Design Principles (Saltzer and Schroeder, 2000), Design guidelines for security management systems (Chiasson *et al.*, 2007), Guidelines and Strategies for Secure Interaction Design (Yee, 2005), Design Principles and Patterns for Aligning Security and Usability (Garfinkel, 2005), and finally, Properties of the Usability Problem for Security (Whitten and Tygar, 1998). The next paragraphs explain all of these usable security guidelines and standards.

2.4.2.1 Computer Security Design Principles (Saltzer and Schroeder, 1975)

The work of Saltzer and Schroeder (1975), "The Protection of Information in Computer Systems", presents the basis required for designing and implementing secure software systems. Their principles describe useful practices that are suitable mainly to architecture-level software decisions regardless of the platform or language of the software. Software developers, whether they are developing new software or assessing existing software, should always apply these design principles as a benchmark for making their software more secure. The eight design principles that apply particularly to protection mechanisms are the following:

1. Keep the design as simple and small as possible. The most natural way to do any task should also be the most secure way. This well-known principle applies to any aspect of a system, but it deserves emphasis for protection mechanisms for this reason: Design and implementation errors that result in unwanted access paths will not be noticed during normal use (since normal use usually does not include attempts to exercise improper access paths). As a result, techniques such as line-by-line inspection of software and physical examination of hardware that implements protection mechanisms are necessary. For such techniques to be successful, a small and simple design is essential.
2. Fail-safe defaults: Base access decisions on permission rather than exclusion. It means that the default situation is lack of access, and the protection scheme identifies conditions under which access is permitted. The alternative, in which mechanisms attempt to identify conditions under which access should be refused, presents the wrong psychological base for secure system design. A conservative design must be based on arguments as to why objects should be accessible, rather than why they should not. In a large system, some objects will be inadequately considered, so a default of lack of permission is safer. A

design or implementation mistake in a mechanism that gives explicit permission tends to fail by refusing permission, a safe situation, since it will be quickly detected. On the other hand, a design or implementation mistake in a mechanism that explicitly excludes access tends to fail by allowing access, a failure which may go unnoticed in normal use. This principle applies both to the outward appearance of the protection mechanism and to its underlying implementation. Complete mediation: Every access to every object must be checked for authority. This principle, when systematically applied, is the primary underpinning of the protection system. It forces a system-wide view of access control, which in addition to normal operation includes initialization, recovery, shutdown, and maintenance. It implies that a foolproof method of identifying the source of every request must be devised. It also requires that proposals to improve performance by remembering the result of an authority check be examined skeptically. If a change in authority occurs, such remembered results must be systematically updated.

3. Open design: The design should not be secret. The mechanisms should not depend on the ignorance of potential attackers, but rather on the possession of specific, more easily protected- keys or passwords. This decoupling of protection mechanisms from protection keys permits the mechanisms to be examined by many reviewers without concern that the review may itself compromise the safeguards. In addition, any skeptical user may be allowed to convince himself that the system he is about to use is adequate for his purpose. Finally, it is simply not realistic to attempt to maintain secrecy for any system that receives wide distribution.
4. Separation of privilege: Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key. The relevance of this observation to

computer systems was pointed out by R. Needham in 1973. The reason for this is that once the mechanism is locked, the two keys can be physically separated, and distinct programs, organizations, or individuals can be made responsible for them. From then on, no single accident, deception, or breach of trust is sufficient to compromise the protected information. This principle is often used in bank safe-deposit boxes. It is also at work in the defense system that fires a nuclear weapon only if two different people both give the correct command. In a computer system, separated keys apply to any situation in which two or more conditions must be met before access is permitted. For example, systems providing user-extendible protected data types usually depend on the separation of privilege for their implementation.

5. Least privilege: Every program and every user of the system should operate using the fewest privileges necessary to complete the job. Primarily, this principle limits the damage that can result from an accident or error. It also reduces the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privileges are less likely to occur. Thus, if a question arises related to the misuse of a privilege, the number of programs that must be audited is minimized. Put another way, if a mechanism can provide "firewalls," the principle of least privilege provides a rationale for where to install the firewalls. The military security rule of "need-to-know" is an example of this principle.
6. Least common mechanism: Minimize the number of mechanisms common to more than one user and depended on by all users. Every shared mechanism (especially one involving shared variables) represents a potential information path between users, and must be designed with great care to be sure it does not unintentionally compromise security. Furthermore, any mechanism

serving all users must be certified to the satisfaction of every user, a job presumably harder than satisfying only one or a few users. For example, given the choice of implementing a new function as a supervisory procedure shared by all users or as a library procedure that can be handled as though it were the user's own, choose the latter option. Then, if one or a few users are not satisfied with the level of certification of the function, they can provide a substitute or not use it at all. Either way, they can avoid being harmed by a mistake in it.

7. Psychological acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly. Also, to the extent that the user's mental image of his protection goals matches the mechanisms he must use, mistakes will be minimized. If he must translate his image of his protection needs into a radically different specification language, he will make errors.

2.4.2.2 Design Guidelines For Security Management Systems (Chiasson *et al.*, 2007)

Although end-users are the major concern for the field of usable security, interfaces for security professionals are equally important, because the consequences of usability problems can potentially leave entire networks vulnerable to attack. For example, an Administrator might miss an attack entirely or misdiagnose it. For example, poor upgrades through security patches can lead to unstable systems that need to be rolled back. Also, despite the fact that the knowledge acquired to design for end-users can assist in designing interfaces for security experts, these two user-categories are quite different in terms of the domain knowledge, their level of responsibility, the amount of information these users have to process, and the

consequences of their actions. To this end, Chiasson *et al* (2007) developed a preliminary set of design guidelines for security management interfaces as follows:

1. Administrators should reliably and promptly be made aware of the security tasks they must perform;
2. Administrators should be able to figure out how to successfully perform those tasks;
3. Administrators should be able to tell when their task has been completed;
4. Administrators should have sufficient feedback to accurately determine the current state of the system and the consequences of their actions;
5. Administrators should be able to revert to a previous system state if a security decision has unintended consequences;
6. Administrators should be able to form an accurate and meaningful mental model of the system they are protecting;
7. Administrators should be able to easily examine the system from different levels of encapsulation in order to gain an overall perspective and be able to effectively diagnose specific problems;
8. The interface should facilitate the interpretation and diagnosis of potential security threats;
9. Administrators should be able to easily seek advice and take advantage of community knowledge to make security decisions;
10. The interface should encourage administrators to address critical issues in a timely fashion.

These design principles recognize that users will need to -make key decisions and be supported in this process. The majority of the interactions will take place because of unpredicted events that the system cannot handle on its own, and as such it should strive to give clear, pertinent, and sufficient information so that users are able to precisely identify and address the problem.

Furthermore, it is to be expected that users will occasionally make mistakes when dealing with these novel situations so the system must allow users to easily revert to a previous state. Such mistakes differ from the “dangerous errors” addressed in usable security because these mistakes may not be possible to predict (whereas dangerous errors such as entering a password in a phishing site are always considered bad). For example, poor upgrades through security patches can lead to unstable systems that need to be rolled back. Occasional mistakes are unavoidable, and thus the systems must be flexible enough so that recovery is possible.

When faced with a new security threat, it is likely that others are also being similarly attacked. The interface should support and facilitate interaction within the security community not only to more quickly analyze a new threat and determine appropriate counter-measures, but also to facilitate propagation of such security measures. Social navigation could also be used to provide trusted feedback about what steps others have taken in similar situations, and could be further customized by defining a specific group of trusted sources from which to gather information. Integrating the communication and social navigation into the system could be faster, have less noise, and be harder to spoof than current ad hoc methods. Security systems still generate a sufficiently large number of false alarms to potentially lure administrators into ignoring alarms or deeming them as non-urgent, or otherwise lead to situations where it is impossible to address all alarms. This may result in unnecessarily vulnerable systems. The interface should attempt to recognize such situations and encourage the administrators to take corrective action. The interface should alert administrators if the majority of other security professionals have taken some preventative measure that has yet to be addressed in the current system, especially if related to a severe threat given the specific system configuration.

2.4.2.3 Guidelines and Strategies for Secure Interaction Design (Yee, 2005)

This section presents a preliminary set of guidelines for secure interaction design. The criterion used by Yee (2005) for admitting something as an essential principle is that it should be a *valid* and *non-trivial* concern. Each principle is valid by showing how a violation of the principle would lead to security vulnerability. In the statement of these principles, the term “actor” is used to mean “user or program”. The term “authority” only refers to the capability to take a particular action.

1. Path of Least Resistance: The most natural way to do any task should also be the most secure way.
2. Appropriate Boundaries: The interface should expose, and the system should enforce, distinctions between objects and between actions along boundaries that matter to the user.
3. Explicit Authorization: A user's authorities must only be provided to other actors as a result of an explicit user action that is understood to imply granting.
4. Visibility: The interface should allow the user to easily review any active actors and authority relationships that would affect security-relevant decisions.
5. Revocability: The interface should allow the user to easily revoke authorities that the user has granted, wherever revocation is possible.
6. Expected Ability: The interface must not give the user the impression that it is possible to do something that cannot actually be done.
7. Trusted Path: The interface must provide an unspoofable and faithful communication channel between the user and any entity trusted to manipulate authorities on the user's behalf.
8. Identifiability: The interface should enforce that distinct objects and distinct actions have unspoofably identifiable and distinguishable representations.

9. Expressiveness: The interface should provide enough expressive power (a) to describe a safe security policy without undue difficulty and (b) to allow users to express security policies in terms that fit their goals.
10. Clarity: The effect of any security-relevant action must be clearly apparent to the user before the action is taken.

2.4.2.4 Design Principles and Patterns for Aligning Security and Usability (Garfinkel, 2005)

Garfinkel's (2005) doctoral thesis's philosophy was to identify patterns that can make systems that are in fact secure, rather than the conventional goal of creating systems that are in theory securable. This section introduces the six general design principles and patterns for aligning security and usability:

1. The Principle of Least Surprise. This principle is an interpretation of Saltzer and Schroeder's (1975) principle of "psychological acceptability." This principle holds that the computer should not surprise the user when the user expects the computer to behave in a manner that is secure. The Principle of Least Surprise is violated when there is a mismatch between the user's expectations and the computer's implementation.
2. The Principle of Good Security Now. Computer security is an engineering discipline. Even though it is impossible to have a computer system that is completely secure, there is always a tension between deploying good systems that are available today and waiting for better systems that can be deployed tomorrow. This principle holds that it is a mistake not to deploy good systems that are available now: if good systems are not deployed, end-users who are not trained in security will create their own- poor security solutions.
3. Provide Standardized Security Policies. Today's security subsystems provide too many choices and configuration options that are relevant to security.

These choices are frequently overwhelming to end-users. Worse, relatively minor changes in a security policy or configuration can have a drastic impact on overall security. Most users need security experts to make decisions for them, because - by definition - users are not experts. This is not to say that users need to be locked in tightly to a few inflexible policies from which they can never deviate. What is needed is a range of well-vetted, understandable, and teachable policies, and then the ability to make understood, controlled, contained and auditable deviations from these policies when needed.

4. **Consistent Meaningful Vocabulary.** Usability is promoted when information is presented with a vocabulary that is consistent and meaningful. But there is a natural tendency among computer engineers to be loose with their choice of language. A guiding principle for aligning security and usability is that security information, at least, must be standardized and used consistently.
5. **Consistent Controls and Placement.** In addition to standardizing vocabulary, it is important that security-related controls in graphical user interfaces be likewise standardized, so that similar functionality is presented in a similar manner and in a consistent location in user interfaces.
6. **No External Burden.** Security tools must not pose a burden on non-users who do not otherwise benefit from their use. Otherwise, non-users will push back on users through social channels and encourage the users to discontinue the use of the tools.

According to Grudin (1989), these principles should be adapted rationally to the tasks that are at hand since there are many cases in which a simple application of consistent UI rules does not lead to interfaces that are easy-to-use.

2.4.2.5 Criteria for Security Software to Be Usable (Whitten and Tygar, 1998)

The authors studied the usability of Pretty Good Privacy (a public key encryption application) which was considered to have a good GUI, but the results showed that PGP 5.0 was not suitably usable to provide effective security for most users.

Security has some inherent properties that make it a difficult problem domain for user interface design. Design strategies for creating usable security will need to take these properties explicitly into account, and generalized user interface design does not do so. Five properties are described below; it is possible that there are others that have not yet been identified.

1. The unmotivated user property: Security is usually a secondary goal. People do not generally sit down at their computers wanting to manage their security; rather, they want to send email, browse web pages, or download software, and they want security in place to protect them while they do those things. It is easy for people to put off learning about security, or to optimistically assume that their security is working, while they focus on their primary goals. Designers of user interfaces for security should not assume that users will be motivated to read manuals or to go looking for security controls that are designed to be unobtrusive. Furthermore, if security is too difficult or annoying, users may give up on it altogether.
2. The abstraction property: Computer security management often involves security policies, which are systems of abstract rules for deciding whether to grant accesses to resources. The creation and management of such rules is an activity that programmers take for granted, but which may be alien and unintuitive to many members of the wider user population. User interface design for security will need to take this into account.

3. The lack of feedback property: The need to prevent dangerous errors makes it imperative to provide good feedback to the user, but providing good feedback for security management is a difficult problem. The state of a security configuration is usually complex, and attempts to summarize it are not adequate. Furthermore, the correct security configuration is the one which does what the user “really wants”, and since only the user knows what that is, it is hard for security software to perform much useful error checking.
4. The barn door property: The proverb about the futility of locking the barn door after the horse is gone is descriptive of an important property of computer security: Once a secret has been left accidentally unprotected, even for a short time, there is no way to be sure that it has not already been read by an attacker. Because of this, user interface design for security needs to place a very high priority on making sure users understand their security well enough to keep from making potentially high-cost mistakes.
5. The weakest link property: It is well known that the security of a networked computer is only as strong as its weakest component. If a cracker can exploit a single error, the game is up. This means that users need to be guided to attend to all aspects of their security, not left to proceed through random exploration as they might with a word processor or a spreadsheet.

2.4.2.6 Additional Criteria for Security Software to Be Usable (Chiasson *et al.*, 2006)

In their usability study regarding two password managers, Chiasson *et al.* (2006) proposed two additional criteria which actually support items 2 and 3 from Whitten and Tygar (1998) above:

1. Be able to tell when their task has been completed: It concerns a usability problem seen in both the (Whitten and Tygar, 1998) and (Chiasson *et al.*,

2006) studies: users were unable to tell whether their task had been successfully completed, and sometimes incorrectly assumed success. This can cause security vulnerabilities (e.g., as information believed to be secure can be left unprotected).

2. Have sufficient feedback to accurately determine the current state of the system: This criterion uses the well-known usability guideline of feedback, which is especially important for supporting accurate mental models in security interfaces. Transparency in this case can be dangerous because it leaves users free to make assumptions about the system that could lead to security breaches.

2.4.2.7 General Security Usability Principles (Identity Management) (Jøsang *et al.*, 2007)

Direct user involvement in a security service is often required, and a distinction can be made between two types of involvement. A security action is when users are required to produce information and security tokens, or to trigger some security relevant mechanism. For example, typing and submitting a password is a security action. A security conclusion in turn is when users observe and assess some security relevant evidence in order to derive the security state of systems. For example, observing a closed padlock on a browser- and concluding that the communication is protected by SSL is a security conclusion.

Usability principles related to security actions and security conclusions are described below.

- Security Action Usability Principles:
 - The users must understand which security actions are required of them.

- The users must have sufficient knowledge and the practical ability to take the correct security action.
- The mental and physical load of a security action must be tolerable.
- The mental and physical load of making repeated security actions for any practical number of transactions must be tolerable.
- Security Conclusion Usability Principles
 - The user must understand the security conclusion that is required for making an informed decision. This means that users must understand what is required of them to support a secure transaction.
 - The system must provide the user with sufficient information for deriving the security conclusion. This means that it must be logically possible to derive the security conclusion from the information provided.
 - The mental load of deriving the security conclusion must be tolerable.
 - The mental load of deriving security conclusions for any practical number of service access instances must be tolerable.

This chapter described the most representative usability inspection methods currently used in the HCI landscape. These methods have served as foundation (and data source) for the development of the USS. Also, it is important to note that due to the absolute lack of specific standards and guidelines for user authentication, HCISec research, including both previous and current work related to usable security of computer security mechanisms, has been presented. It was crucial to research the usable security related to a broader spectrum of security mechanisms, first- because of the referred lack of user authentication guidelines, and then secondly to understand what they are, how they work, and verify if any of these guidelines could be used or re-adapted for user authentication.

Summary of the topics discussed in Chapter 2: Review of the State of Art.

Chapter 2 described the state of the art of User Authentication, the GOMS models, and the Usable Security Principles and Guidelines. Regarding user authentication, the following topics have been described: The context of user authentication in Computer Security, Elements of user authentication, Architectural design patterns in Authentication, Authentication factors, User Authentication Methods (Passwords and PINs, Authentication tokens, Digest access authentication, Out-Of-Band Authentication, Risk-based authentication, Public Key Authentication, Single Sign-On, Kerberos, Biometrics), To whom authentication is targeted?, and Comparative Analysis of User Authentication Methods. Next, the GOMS models have dealt with the Engineering models for usable interface design, GOMS as a method for cognitive task analysis, GOMS analysis guiding the design, Natural GOMS Language (NGOMSL), Cognitive complexity theory, Learning and Execution time predictions, NGOMSL methodology, and NGOMSL limitations. Finally this chapter has also described the most representative usability inspection methods in HCI, and discussed HCISec research, both previous and current, in regards to the principles and guidelines related to usable security of computer security mechanisms from (Saltzer and Schroeder, 2000; Chiasson & al, 2007; Yee, 2005; Garfinkel, 2005; Whitten and Tygar, 1998).

CHAPTER III

THE USABLE SECURITY PROTOCOL

3.1 Introduction

Numerous studies have shown that usability inspection methods are capable of finding many usability problems that are disregarded by user testing but also that user testing also involves some problems that are disregarded by inspection, meaning that the best results can often be achieved by combining several methods (Desurvire, 1994; Desurvire *et al.*, 1992; Karat *et al.*, 1992).

Both empirical usability testing and usability inspection methods appear to be in extensive use by designers who choose the most suitable method for their purposes and their context. While inspection methods require expert evaluators to be effective, their strengths are that they can be implemented in the early phases of the development cycle and offer an opportunity in which changes to an interface can be agreed upon.

Empirical methods can also be used early in the development process- through low-fidelity versions of interfaces. Also designers frequently combine multiple inspection methods - heuristic evaluation and the cognitive walkthrough - so that it is feasible to obtain better coverage of usability issues. Finally, adding multiple perspectives such as the range of stakeholders or types of usability problems seems to improve the efficiency of inspection methods.

This thesis's author agrees with Jøsang *et al.* (2007) that poor usable security clearly represents a significant vulnerability. It seems that poor security usability still does not appear on standard vulnerability checklists used by security analysts and experts. This is an urgent issue that has to be addressed, and this thesis addresses this issue by proposing the Usable Security Symmetry inspection method.

3.2 The Usable Security Protocol Methodology

This chapter details the Usable Security Protocol (USP) methodology step by step, including the goal and the logistics behind each step and the utility of using a cognitive and computer science approaches in developing a usability inspection method for user authentication. This chapter starts with an overview of the USP methodology and discussion of how the USP methodology brings together the cognitive and computer science approaches, detailing the theoretical and demonstrational basis for the Validation and Verification (V&V) of the protocol. Each step of USP is then presented in sequence, and finally, at the end it shows how all the steps fit together to provide a design requirements inspection method tool for the design of user authentication methods.

3.2.1 Introduction

Before delving into the specifics of the USP methodology, this thesis's author considers it important to define what the *notion of methodology* is for the purposes of this thesis. The notion of "methodology" encompasses a description of processes as they relate to the particular discipline of software engineering. The reason for adopting such a notion is that based on the professional experience of this thesis's author, the development of a software product is similar to the development of an authentication method product. Therefore this thesis's author borrows this definition, and the term *methodology* is used throughout this thesis.

The USP development methodology refers to the framework that is used to structure, plan, and control the process of developing a usable security user authentication inspection method. The framework of the USP methodology consists of multiple methods- to assist in the USP development process, having as the final output the USS inspection method.

This thesis is based on applied research that encompasses both theoretical and demonstrational approaches developed from the analysis and gathering of primary, secondary, and tertiary data. Both approaches are then validated by the V&V phase within the protocol. The theoretical approach is comprised of usable security principles, which

when relevant, are confronted against their cognitive dimensions as presented by the cognitive model and also by the Usable Security Symmetry (USS) inspection method in Chapter 6.

The demonstrational approach presents the theoretical approach through a demonstration comprised of a representative authentication method named RSA SecurID® SID700 hardware authenticator with One-Time Password (OTP), which is completely described in Chapter 6. The theoretical approach is in turn validated by the Cognitive Model in Step 4, showing the respective cognitive dimensions using GOMS (Natural Goals, Operators, Methods, Selection Language) or more specifically Natural GOMS Language (NGOMSL).

Methodologically speaking, here is a summary of the main activities developed for the accomplishment of this thesis:

- Initially, a comparative analysis of the existing user authentication methods is developed in order to understand what authentication methods are, how they work, and what kind of features are contained in them. Then a classification analysis is undertaken from the literature review (e.g. authentication marketplace) to establish the main user authentication methods to be used for the purposes of this thesis.
- Afterward-, the most representative user authentication methods categories according to Allan (2007) from Gartner are identified as follows: *Password/PINs* (wired network-based task): username and password login operation in a desktop environment; *One-Time-Passwords*(wireless/token network-based task): real-time generated OTPs based on the challenge-response method; *Out-of-Band Authentication* (wired and wireless network-based task): utilization of two separate networks working concurrently to authenticate a user (e.g., computer and mobile device interaction, such as a cell phone, blackberry, etc.); and finally *Biometrics* (wired network and electronic access control-based task): logical and physical access control (e.g. fingerprint). In parallel, types of users (e.g. Super Admin, End-Users, etc.) and working contexts are identified.

- From the user authentication methods categories identified previously, the task scenarios are created to perform the NGOMSL analysis, which includes the following: Check Business E-mail, Update the SecurID token User Interface Specification, Make an electronic funds transfer, and Access a file on a personal laptop.
- Also in parallel, the identification of the main cognitive areas of focus relating to user authentication (i.e. perception, attention and memory, mental models) is established followed by the definition of the appropriate cognitive architecture (i.e. adaptation of EPIC and SOAR), which guides the construction of the final cognitive architecture: the Cognitive Model of User Authentication (CMUA). CMUA helps to determine how and what cognitive processes are involved specifically for user authentication tasks (e.g. attention and memory, etc.). It serves as the basis for the development of the USS inspection method.
- Next, the NGOMSL model is developed by i) specifying Standard Primitive External (i.e. Type <string of characters>, example: Type <username>) and Mental Operators (i.e. Recall <STM-object-description>, example: Recall <passcode>), and Analyst-Defined Mental Operators (i.e. Think-of <description>, example: Think-of <VPN Dialer>) and ii) generating a Task Description, a list of High-Level User Goals, Operators and Write Methods for Accomplishing Goals, and Total Execution and Learning Times estimates for each of the user authentication use cases.
- After that, an Authentication Risk Assessment matrix is undertaken to identify the most critical vulnerabilities and threats related to online user authentication. This assessment determines which Security Review should be considered within each usability criterion in the USS.
- Next, the specification of the usability factors and usability criteria is undertaken by classifying and prioritizing the cognitive processes generated by the NGOMSL model.

- Afterward, the development of the USS inspection method itself is carried out by: i) plotting the usability factors and usability criteria in the USS matrix, ii) defining the rating severity of identified security problems, its rating severity representation, and recommendations, and iii) defining the rating severity of identified usability problems, its rating severity representation, and recommendations.
- The V&V phase of the USS is undertaken by using the Multifunction Teller Machine (MTM) example and the RSA SecurID® 700 hardware token.
- Finally, a usability Testing has been performed to identify high-priority usability issues. The testing assesses the usability of designs for end-user authentication tasks involving remote access; Secure Socket Layer (SSL); and Virtual Private Network (VPN), which is commonly known as SSL-VPN user authentication, a two-factor OTP system that provides strong authentication⁵¹.

3.2.2 The Usable Security Protocol Architecture and Methodology

An orderly and sequential seven-step methodology, which makes clear the process of enquiry through which knowledge materializes, is undertaken to generate the USP architecture as depicted in Figure 3.1. The USP architecture is comprised of Primary, Secondary, and Tertiary Data in the development phase. Then the Cognitive Science Model {Cognitive Ergonomics} is developed followed by the Computer Science Model {Demonstration}, which both form the Theoretical and Demonstrational approaches.

⁵¹ Adaptive Authentication for the Enterprise. RSA-The Security Division of EMC. May 19, 2010 <<http://www.rsa.com/node.aspx?id=3018>>

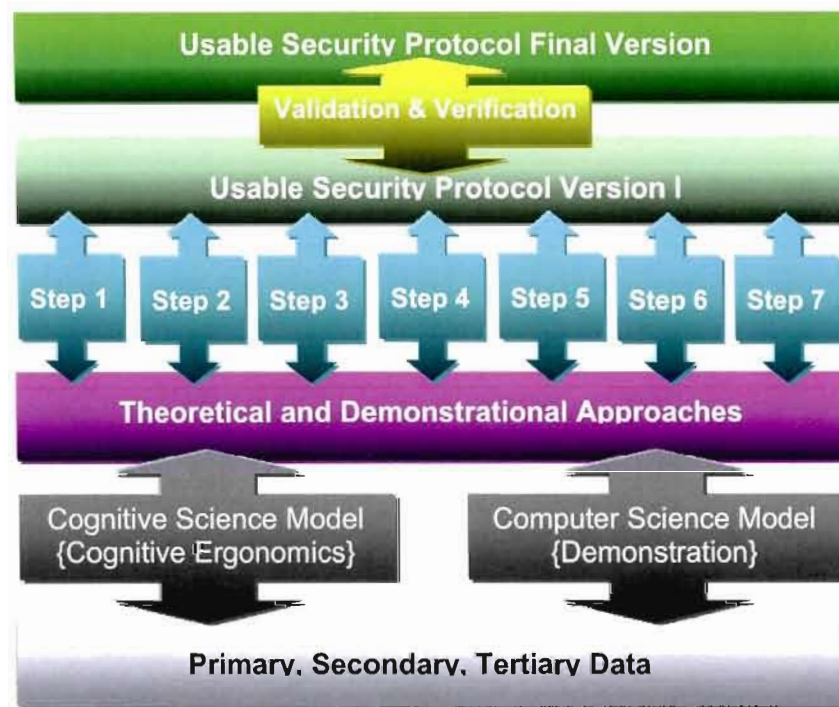


Figure 3.1: The USP architecture and methodology.

- Step 1: Define the mission and conceptual design objective.
- Step 2: Identify the most representative user authentication methods categories.
- Step 3: Develop the Natural GOMS Language (NGOMSL).
- Step 4: Develop the Authentication Risk Assessment Matrix.
- Step 5: Generate the usable security principles from the NGOMSL model.
- Step 6: Formulate the Usable Security Symmetry (USS) inspection method.
- Step 7: Demonstrate the Usable Security Symmetry (USS).

Finally, the V&V phase is undertaken in Step 7, where the USP is validated by the authentication method demo. The V&V will demonstrate the protocol, fulfilling its intended purpose. As already mentioned, the key point of this thesis is the creation, development, and integration of the USS into the Requirements & Design process of user authentication method design.

Using one of the most essential design patterns, the Input-Process-Output (IPO) model, the different functions and interactions between the USP (information system) and the outside world (users in the environment) are represented in Figure 3.2. Data is provided for the USP (Input) (e.g., gathering and capturing data for the comparative analysis of user authentication methods), which is afterward analyzed and reorganized (Process) (e.g., analyzing and classifying the most representative user authentication methods), and finally it is displayed (Output) (e.g. selecting the most representative user authentication methods to be used in the GOMS model). The Process stage in fact entails translating or converting data into useful Outputs. Thus processing can involve formulating comparisons, making calculations (i.e. execution time in NGOMSL), or classifying data for future use.


OUTPUT	Usable security definition and use cases, types of users, comparative analysis of user AMs	User AMs, task scenarios: Password, PINs, OTP, OOBAs, Fingerprint	Total execution and learning times, execution time by task scenario, AM type Method for goal (Log into the system)	Authentication assets / targets and their respective risk reduction strategies	Usable security principles targeted to user authentication	USS inspection method	USS MTM case and OTP usability testing
							
PROCESS (Operations)							
INPUT (Actors, Resources)	Security and usability definitions, Tasks, usability and security scenarios, users, user AMs	Comparative analysis of user AMs	Operators, tasks descriptions, high-level users goals, methods for goal using NGOMSL model	OWASP CVE, SANS, CIO, Top 10 threats and vulnerabilities	Cognitive processes from NGOMSL and usability and security principles	Usability factors/ criteria, user authentication use cases, project lead/ development activities, usability and security severity ratings	Demo of OTP and USS MTM inspection method
ACTORS	<i>R, SA, EU</i>	<i>R</i>	<i>R, EU</i>	<i>R, SA, EU</i>	<i>R, SA, EU</i>	<i>R, EU</i>	<i>R, EU</i>

Figure 3.2: USP inputs, processes, outputs, and actors (R = Researcher, SA = System Admin, EU = End-user).

The Human Actors and their roles in the USP:

- *Researcher*: The individual (graduate student) who undertakes this thesis research and is the Subject Matter Expert (SME).
- *System Administrators*:
 - *Super Administrator*: See Section 1.6 Types of Users.
 - *Domain Administrator*: See Section 1.6 Types of Users.
 - *Help Desk Administrator*: See Section 1.6 Types of Users.
- *End-User*: See Section 1.6 Types of Users.

The System Actors and their roles:

- *Multifunction Teller Machine (MTM)*: A fully integrated cross-bank MTM network providing functionalities which are not straightforwardly associated with the management of one's own bank account, such as loading monetary value into pre-paid cards. Also, the MTMs can provide advanced authentication capabilities such as Biometrics (e.g., palm recognition).
- *EMC VPN Application*: Client Version 4.8.02.0010 2006. Client type: Windows winNT.
- *Authenticator*: RSA SecurID® 700 hardware token.

3.2.2.1 Step 1: Define the mission and conceptual design objective.

Sub-Step 1.1 Formalize a usable security definition:

As stated in the introduction of this thesis, Usable Security is the study of how security information and usability factors should be handled in the system, including both front and back-end processes, and taking into consideration the resources and costs involved. Front-end processes are represented by the interface that has been already highlighted, and are in fact regarded as a shared limit through which the information flows (Maffezzini, 2006).

Sub-Step 1.2 Specify usable security scenarios and use cases. What are usable security scenarios - all about?

Although the HCI and HCISec literature contains plenty of definitions of what task and usability scenarios are, this thesis refines them in the context of usable security, and also introduces a novel definition for a *security scenario* as follows:

- **Task Scenario:** A task scenario refers to a description of the task at hand, including its context of use. According to Figure (1-1), the Context of Use (CoU) analysis refers to a broad technique to determine the characteristics of the User, Tasks, and their Environments. The application of the CoU analysis is primarily -used to support the data gathering requirements to build the basic components at the early developmental stages of the application, and also to establish the end results, which consist of effectiveness, efficiency, and satisfaction.
- **Usability Scenario.** A usability scenario details a user problem when doing a task in a certain context. Therefore, a usability scenario is a problem related to a task scenario, but it should be well known, meaning defined in a usability model, standard, or evaluation method.
- **Security Scenario.** A security scenario refers to a description of a task scenario that includes the use of a particular security mechanism. A security scenario can be *tangible* or *intangible*. A Tangible Security Scenario (TSS) includes physical infrastructure, such as controlling a user's access to buildings and facilities using Biometrics, or sending a silent alarm in response to a threat at MTM. An Intangible Security Scenario (ISS) includes data or other digital information, for example, a user who enters sensitive information at registration in order to purchase a concert ticket at an MTM. A Security Scenario might (or might not) be a combination of TSS and ISS.

Examples of Task, Usability, and Security Scenarios are described in Chapter 5 in Table 5.2, which describes the user authentication use cases as follows: 1. authenticate to an MTM (Multipurpose Contactless Smart Card (MpCC)), 2. transfer funds to an international bank account,

3. buy a ticket concert, 4.access your MTM offline with your cell phone, 5. deposit your check by using checking image, and 6. send a silent alarm.

Sub-Step 1.3 Identify users and working context:

See Chapter 1, Section 1.6: Types of Users.

Sub-Step 1.4 Perform a comparative analysis for user authentication methods:

As part of one of the tasks to understand what authentication methods are, how they work, and what kind of different features are found in them, a comparative analysis of the main user authentication methods has been developed in Chapter 2, Section 2.2.7, Table 2.4.

3.2.2.2 Step 2: Identify the most representative user authentication methods categories

Step 2 will serve as the task scenarios for the NGOMSL model:

Sub-Step 2.1 Password/PINs (wired network-based tasks):

Username and password login operation in a desktop environment.

Sub-Step 2.2 One-Time-Passwords {wireless/token network-based tasks}:

Real-time generated OTPs based on the challenge-response method.

***Sub-Step 2.3 Out-of-Band Authentication (wired and wireless network-based tasks)
(e.g. computer and mobile device such as a cell phone, smartphone etc.)-:***

Utilization of two separate networks working concurrently to authenticate a user (e.g. when a user initiates certain online transactions, the user will be prompted to enter a 4-digit code, which is sent via text message to the user's mobile device).

Sub-Step 2.4 Utilization of Biometrics {wired network and electronic access control-based task}:

Logical and physical access control (e.g. a fingerprint).

3.2.2.3 Step 3: Develop the NGOMSL Model (Natural Goals, Methods, Selection Language)

NGOMSL Model (Natural Goals, Methods, Selection Language) is a method in which learning time and execution time are predicted based on a program-like representation of the procedures that the user must learn and execute to perform tasks with the system. Under NGOMSL, methods are represented in terms of an underlying cognitive theory known as Cognitive Complexity Theory (CCT). This cognitive theory allows NGOMSL to incorporate internal operators (i.e. actions that the user executes) such as manipulating working memory information or setting up sub-goals. Because of this, NGOMSL can also be used to estimate the time required to learn how to achieve tasks. To this end, the NGOMSL task analysis identifies and measures the execution and learning times of key perceptual, cognitive, and motor processes undertaken by users. They are based on expert users and well defined tasks. There is an emphasis on the analysis of those cognitive processes involved in the user authentication processes.

The task scenarios descriptions and a list of goals, methods, and operators for each user authentication method category is described throughout this current step. All data resources for developing the NGOMSL model have been gathered and/or developed from usability tests and user interviews, authentication token demonstrations, published research papers for estimating learning and execution times (Gong and Elkerton, 1990; Gong, 1993; Gong and Kieras, 1994; Card *et al.*, 1983), and observations of real users employing the RSA SecurID® 700 (Figure 2-7) for authentication tasks in their real environments at the RSA Security Corporation in Bedford, MA (US).

A vital decision as to what (and what not to) describe was made when developing the NGOMSL analysis, which is that mental processes should be basically treated as "black boxes" due to their overwhelming complexity. As a matter of fact, this is recommended by Kieras (1996). It means that trying to explain in detail, for instance, what Read mental

process is would be extraordinarily difficult. Hence this thesis treats the user's reading, verification (as looking or seeing), forgetting, reading, and thinking mechanisms as "black boxes" (For further details, see Section 2.3.3, How to Develop a GOMS Model).

This thesis implements a set of basic operators for constructing the NGOMSL model as described in the sub-sections below.

3.2.2.3.1 Standard Primitive External Operators

The analyst⁵² defines the primitive motor and perceptual operators based on the elementary actions required by the system being analyzed. The standard primitive external operators used in this NGOMSL analysis are the following:

- Click mouse button
- Move cursor to <target coordinates>
- Type <string of characters>
- Locate <name> value from screen is equivalent to the process of scanning specific spots on a screen that supply a value for some parameter specified by <name> to determine the location of this value and placing the information into working memory. Basically, there should be a `Locate` operator executed prior to a `Double-click` or `Click`, to reflect that before an object can be clicked, its location must be known (e.g. `Locate the password field`).
- Wait for <description>: The waiting time is the time when the user is waiting idly for the system's response (e.g., the user has entered her username and password, the system processes the user authentication, and logs the user into the system).

⁵² The "Analyst" is the person who performs a GOMS analysis as referred to by (Kieras_96).

3.2.2.3.2 Standard Primitive Mental Operators

As previously mentioned, mental operators are the internal actions (steps within the Methods) performed by the user. These operators include actions like making a basic decision, recalling an item in STM, retrieving information from LTM, etc. The mental operators used in this NGOMSL analysis are described below:

- Read <name> value from screen is equivalent to the process of interpreting characters on a screen that supply a value for some parameter specified by <name> and placing the information into working memory (e.g. Read the password displayed on the digital readout window on the SecurID Token). Verify <name> value from screen is equivalent to the process of representing how the user is expected to notice and make use of that feedback information. A Verify operator should normally be included at the point where the user must commit to the entry of information (e.g. Verify that the password has been correctly typed in).

For memory storage and retrieval, the memory operators reflect the distinction between LTM and STM as they are typically used in computer operation tasks. The standard primitive mental operators used in this analysis are the following:

- Recall <LTM-object-description> Recall means to fetch from LTM. Searches LTM for an item whose specified properties have the specified values, and stores its symbolic name in STM.
- Recall <WM-object-description> Recall means to fetch from STM. Searches STM for an item whose specified properties have the specified values, and stores its symbolic name in STM.
- Retrieve <WM-object-description> Retrieve means to get back an item from memory, which can either be from LTM or STM, during the method execution.
- Retain <WM-object-description> Retain means to store in working memory.
- Forget <WM-object-description> Forget means that the information is no longer needed, and thus can be deliberately dropped from working memory.

- `Listen <Auditory stimulus-object-description> Listen (auditory stimulus)` means that the user listens to either speech or sound inputs. After a standard time delay representing auditory working memory decay time (currently 1000 ms), this object is deleted and the auditory stimulus information is no longer available.
- `Return with goal accomplished` is a basic flow of control. A sub-method is invoked by asserting that its goal should be accomplished, and returns here when the goal has been accomplished. The operator: `Return with goal accomplished` is analogous to an ordinary `RETURN` statement, and marks the end of a method.

3.2.2.3.3 Analyst-Defined Mental Operators

Analyst-Defined Mental Operators represent psychological processes that are too complex to be practical to designate as methods in the GOMS model. The designer can in fact circumvent these processes by defining operators that act as place holders for the mental activities as follows:

- `Think-of <description>` represents a process of thinking of a value for some parameter designated by `<description>` and putting the information into working memory (Kieras, 1996) (e.g. `Think-of "VPN Dialer"` padlock icon which indicates that it is a fully secure connection).
- `Read <name> value from screen` is equivalent to the process of interpreting characters on a screen that supply a value for some parameter specified by `<name>` and placing the information into working memory (e.g. `Read the password` displayed on the digital readout window on the SecurID Token).

As stressed by Kieras (1996), the methods have been represented at the standard primitive operator level, so the calculations for predicting learning and execution times will generate realistic, accurate, and useful results.

3.2.2.3.4 Total Execution Time and Total Learning Time

NGOMSL predicts the Learning Time that users will take to learn the procedures represented in the GOMS model- and the Execution Time (ET) users will take to execute particular task instances by following the procedures. It is important here to understand the difference between these two usability measures.

The **Total Execution Time** comprises the methods, steps, and operators needed to carry out a specific task. The time needed to complete a task instance is determined by the number and content of NGOMSL statements that have to be executed in order to get that particular task done. The time needed by each statement is the sum of a small fixed time for the statement plus the time required by any external or mental operator executed in the statement.

The execution time for a task is predicted by simulating the execution of the methods required to perform the task. Each NGOMSL statement (i.e. each step) is assumed to require a small fixed time to execute, and any operators in the statement, such as a keystroke, will then take additional time depending on the operator.

$$\text{Execution Time} = \text{NGOMSL statement time} + \text{Primitive External Operator Time} + \text{Waiting Time}$$

Execution Time = Time for the execution of the methods required to perform the task itself by the user.

NGOMSL statement time = Number of statements executed x 0.1 sec

Primitive External Operator Time = Total of times for primitive external operators

Waiting Time = Total time when user is inactive while waiting for the system's response. Note: For the purposes of this thesis, the waiting time is irrelevant due to the fact that - the system is considered to be fast enough, and it is not the main focus of our GOMS analysis related to user authentication. Therefore it will be not measured and indicated in the time measurement analysis.

The **Total Learning Time** is the *total number and length of all methods*. The time to learn a set of methods is fundamentally specified by the total length of the methods, which is provided by the number of NGOMSL statements in the whole GOMS model for the interface. This is the quantity of procedural knowledge that the user has to acquire in order to know *how to use the system for all tasks under consideration* (Kieras, 2006). The time required to learn how to perform the methods themselves is defined as the Pure Learning Time.

$$\text{Total Learning Time} = \text{Pure Method Learning Time} + \text{LTM Item Learning Time} + \text{Training Procedure Execution Time}$$

Total Learning Time = the total time needed to complete a training process.

Pure Method Learning Time = Learning Time Parameter x Number of NGOMSL Statements to be learned (the time required to learn how to perform the methods):

Learning Time Parameter = 30 sec for rigorous procedure training or 17 sec for a typical learning situation.

Long Term Memory (LTM) Item Learning Time = the time required to memorize items that will be retrieved from LTM during method execution. Gong (1993) has estimated that in general this takes approximately 6 secs x Number of LTM Chunks* to be Learned.

* *Retrieving a chunk from memory*: A chunk is a common unit such as a file name, command name, or abbreviation. For instance, if the user wants to list the contents of directory `foo`, they need to retrieve two chunks, `dir` and `foo`, each of which takes an **M**.

According to Kieras (1996), there is no recognized and verified method for counting how many chunks are implicated in “to-be-memorized” information, so what is presented next is heuristic-based information. Count the number of chunks as follows:

- one chunk for each common pattern in the retrieval cue.
- one chunk for each common pattern in the retrieved information.

- one chunk for the association between the retrieval cue and the retrieved information.

For instance, presume that the “to-be-stored” association for a command is *move cursor right by a word* is *CTRL-RIGHT ARROW*. Then:

(move cursor right) (by a word) = 2 chunks for retrieval cue

(ctrl) (right-arrow) = 2 chunks for retrieved information

association between the two = 1 chunk

Training Procedure Execution Time = If the procedures to be used in training are known, it may be useful to estimate the total learning time by adding the time required to execute the training procedures.

The following KLM (Keystroke-Level Model) GOMS Operators have been used for some of the operators’ duration times (Kieras, 2001) according to the table below. Although those KLM Operators have been counted and included as part of the time calculations, they are not explicitly shown in *Sub-Steps 1, 2, 3, and 4* in the next sections as specific values within the time measurements so as to simplify data presentation.

When considering a user’s typing skills, it is important to note that the typing time depends on the typing skill of the user, so for the purposes of this thesis it has been specified by the Average Skilled Typist (55 wpm*) = 0.20 secs (Card *et al.*, 1983).

*wpm=words per minute is a measure of input or output speed.

Operator	Abbreviation	Duration (secs)
Mental	M (1)	1.20
Keystroke <key>	K	0.28
MouseDown or Up	B	0.10
Click (mouseDown & Up)	BB	0.20
Homing (2)	H	0.40
DoubleClick	BBBB	0.40
Point w/ mouse	P (3)	1.10
Type characters	T(n) (4)	0.20
System Response	R (5)	t

(1) Thinking time. (2) Homing is the process of determining the location of something, and going to it. (3) Point with mouse to a target on a display. (4) T(n): Type a sequence of N characters on a keyboard.

(5) The system response time during which the user has to wait for the system. The duration (t) can drastically vary depending on the system being analyzed.

An example of TET and TLT time calculations for “Access a file on a personal laptop”, Method for goal: Log into the system, follows:

Total Execution Time (TET):

Method for goal: Log into the system	NGOMSL Statement (secs)	Operator (Type)	Operator Time (secs) (1)	Sub-Total Execution Time (secs)
Step 1. Read fingerprint logon Welcome screen containing finger image and "Password" field on the laptop computer.	0.10	M	2.01	2.11
Step 2. Refer to the Universal Serial Bus (USB)-based biometric fingerprint reader.	0.10	H	2.01	2.11
Step 3. Locate the fingerprint sensor on the USB fingerprint reader.	0.10	M	2.01	2.11
Step 4. Move finger to the USB fingerprint reader.	0.10	P	1.01	1.11
Step 5. Position last knuckle joint over the center of the fingerprint sensor.	0.10	H	1.01	1.11
Step 6. Swipe the finger without lifting it over the fingerprint sensor.	0.10	P	1.21	1.31
Step 7. Verify that you have been granted access to the system.	0.10	M	1.11	1.21
Total Execution Time (secs)				9.16

(1) A 0.01 milliseconds have been added to the Operator Time as a margin of error.

Total Learning Time (TLT):

Total Learning Time = Pure Method Learning Time + Long Term Memory Item Learning Time + Training Procedure Execution Time

$$\text{Total Learning Time} = 119 \text{ secs} + 30 \text{ secs} + 0 \text{ secs} = 149 \text{ secs}$$

(Pure Method Learning Time = Learning Time Parameter x Number of NGOMSL Statements to be learned -> 17 secs x 7 steps = 119 secs)

The following task- scenarios with their corresponding authentication methods have been created to develop the NGOMSL model:

- Check Business E-mail incorporates the Username and Password login authentication method;
- Update the SecurID token User Interface Specification incorporates One-Time-Passwords (OTP);
- Make an electronic funds transfer incorporates Out-Of-Band Authentication (OOBA);
- Access a file on a personal laptop incorporates Biometrics (fingerprint recognition).

These tasks were conceived of only to serve as a basis for demonstrating the authentication portions which are embedded in them. Security is a secondary goal for many users, an indispensable step in the way of achieving their primary goals such as the task-scenarios mentioned previously. Therefore, it would be odd to describe only the authentication activity given that users don't authenticate to a system, and in fact do nothing. There is always a goal involved when authenticating to a system.

As already mentioned, a method is a series of steps that accomplishes a goal. A step in a method typically consists of an external operator, such a pressing a key, or a set of mental operators involved in setting up and accomplishing a sub-goal. Much of the work in analyzing a user interface consists of specifying the actual steps that users carry out in order to accomplish goals, so describing the methods is the focus of the task analysis. According to NGOMSL, the structure for a method is as follows:

```
Method for goal: <goal description>
Step 1. <operator>...
Step 2. <operator>...
Step 3. <operator>...
...
Step n. Return with goal accomplished.
```

The NGOMSL analysis lays down the foundation for the design of the USS inspection method. Finally, this section presents the results of quantitative and qualitative aspects of NGOMSL applied to user authentication.

Sub-Step 3.1 Username and Password Login {wired network-based task}:

This authentication task scenario is specified as Task_scenario: T1.

- *Generate Task Description*

This task is related to the username and password login operation in a wired network-based desktop environment. Password authentication is the most common method of authentication, and also one of the least secure. Basically the computer asks the user to type in a username and a password. The computer searches the system's password file for an entry matching the username in the database. If the password in that entry matches the password just typed, then the login succeeds. (For more details, see section 2.2.5.1 Passwords and PINs). This task scenario makes use of the Microsoft Windows NT⁵³ operating system, which controls - user access to systems within and across domains (i.e. local and remote access). When a user logs on to an NT system, NT validates the user's account and authorizes access to the appropriate system or domain (i.e. Windows NT authentication).

- *Describe a List of High-Level User Goals*

The topmost user's goal is: Check Business E-mail. The set of a user's high level goals - includes the following:

- Log into the system
- Open -Microsoft Office Outlook
- Read E-mail Message
- Return with goal accomplished.

This analysis is based on the premise that this thesis is about user authentication, so a particular level of analysis or granularity is required for the Log into the system high-

⁵³ Microsoft Windows NT <<http://www.microsoft.com/technet/archive/winntas/default.msp?mfr=true>>

level operator. It means that this operator is decomposed into finer levels (i.e. a series of lower-level, or primitive, operators), whereas the `Open -Microsoft Office Outlook` and `Read e-mail message` high-level operators function as supportive methods within the task, and are therefore not decomposed. This is in fact a common decision-making point when undertaking GOMS: specifically, whether the operator should or should not be decomposed into finer levels, depending on the finest grain level of analysis desired by the analyst. It is worth noting that users do not authenticate to a system *per se* since *authentication is not a goal but rather a means* to accomplishing a goal within the context of this task: `Check Business E-mail`.

- *Define Operators and Write Methods for Accomplishing User Goals*

`Method for goal: Log into the system`

- Step 1. Read Windows Logon Welcome screen on the desktop computer.
- Step 2. Locate Ctrl+Alt+Del key combination on the keyboard and simultaneously hold them down.
- Step 3. Verify that the Windows pop-up window is opened.
- Step 4. Locate the username field on the screen.
- Step 5. Move the cursor to the username field.
- Step 6. Recall the username "jdoe", retrieve it from LTM, and retain it.
- Step 7. Type the username in the username field.
- Step 8. Verify that the username has been correctly typed in.
- Step 9. Forget the username.
- Step 10. Locate the password field.
- Step 11. Move the cursor to the password field.
- Step 12. Recall that the password is "Boat6paper!", retrieve it from LTM, and retain it.
- Step 13. Type the password in the password field.
- Step 14. Verify that the password has been correctly typed in.
- Step 15. Forget the password.
- Step 16. Locate the "Submit" button.

Step 17. Double-click the “Submit” button.

Step 18. Verify that you have been granted access to the system.

Step 19. Return with the goal accomplished.

Method for goal: Open -Microsoft Office Outlook

Step 1. Locate the Microsoft Office Outlook icon in the task bar.

Step 2. Double-click on the Microsoft Office Outlook icon.

Step 3. Verify that Microsoft Office Outlook has opened up the inbox.

Step 4. Return with the goal accomplished.

Method for goal: Read E-mail Message.

Step 4. Double-click on any e-mail message row.

Step 5. Verify that the e-mail message has been opened.

Step 6. Return with the goal accomplished.

- *Estimate Total Execution Time:*

The time measurement related to the *execution time* for Task_scenario: T1 Check Business E-mail is listed in seconds below.

Method for goal: Log into the system	Sub-Execution Time(s)
Step 1. Read Windows Logon Welcome screen on the desktop computer.	1.21
Step 2. Locate Ctrl, Alt, and Del keys on the keyboard.	1.21
Step 3. Hold down Ctrl+Alt+Del keys simultaneously.	0.77
Step 4. Verify that the Windows pop-up window is opened.	1.21
Step 5. Locate the username field on the screen.	1.21
Step 6. Move the cursor to the username field.	1.11
Step 7. Recall the username "jdoe", retrieve it from LTM, and retain it.	1.21
Step 8. Type the username in the username field.	2.16
Step 9. Verify that the username has been correctly typed in.	1.21
Step 10. Forget the username.	1.21
Step 11. Locate the password field.	1.21
Step 12. Move the cursor to the password field.	1.11
Step 13. Recall that the password is "Boat6paper!", retrieve it from LTM, & retain it.	1.21
Step 14. Type the password in the password field.	2.16
Step 15. Verify that the password has been correctly typed in.	1.21
Step 16. Forget the password.	1.21
Step 17. Locate the "Submit" button.	1.21
Step 18. Double-click on the "Submit" button.	0.21
Step 19. Verify that you have been granted access to the system.	1.21
Total Execution Time	23.25

Method for goal: Open the Microsoft Office Outlook (MOO)	Sub-Execution Time(s)
Step 1. Locate the MOO icon in the task bar.	1.21
Step 2. Double-click on the MOO icon.	0.41
Step 3. Verify that MOO has opened up the inbox.	1.21
Total Execution Time	2.83

Method for goal: Read E-mail Message	Execution Time(s)
Step 1. Double-click on any e-mail message row.	0.41
Step 2. Verify that the e-mail message has been opened.	1.21
Total Execution Time	1.62

The Total Execution Time for **Task_scenario: T1 - Check Business E-mail** is shown below:

Task_scenario: T1 Check Business E-mail	
Methods	Sub-Execution Time(s)
Method for goal: Log into the system	23.25
Method for goal: Open -Microsoft Office Outlook	2.83
Method for goal: Read E-mail Message	1.62
Total Execution Time	27.70

- *Estimate Total Learning Time:*

The Total Learning Time for **Task_scenario: T1 - Check Business E-mail** is shown below:

Total Learning Time = Pure Method Learning Time + LTM Learning Time + Training Procedure Execution Time

Total Learning Time = 408 secs + 24 secs + 0 secs = 432 secs

Sub-Step 3.2 One-Time-Passwords (OTP) {wired network-based task}:

This authentication task scenario is specified as **Task_scenario: T2**.

- *Generate Task Description*

This is a hardware token and wireless network-based task with real-time generated OTPs based on the challenge-response authentication method. The typical scenario where OTP is used is with Secure Sockets Layer (SSL)⁵⁴/Virtual Private Network (VPN)⁵⁵ servers and web portals.

In a simple scenario, John Doe, a user, connects to the VPN application and opens it in his computer. John enters his username (jdoe) and a 4-digit numeric PIN (7234) in the username and passcode fields in the login application screen. Then John

⁵⁴ Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data – a public key known to everyone and a private or secret key known only to the recipient of the message.

⁵⁵ Virtual Private Network (VPN) is a private network that uses a public network (usually the Internet) to connect remote sites or users together.

types and appends the six digit numeric token number (currently displayed on his hardware token, e.g. 435961) to the entered PIN on the login screen. This whole number (7234435961) becomes the **passcode**, which is in fact the combination of the PIN and the token code. John clicks “OK”. If the authentication is successful, the authentication server validates the passcode and grants user access to the network.

Pre-condition for this task scenario: The authentication server randomly generates - OTPs (six-digit numeric token numbers) on the user token’s display (e.g. 435961). John has possession of a hardware authentication token which has been assigned to him by the company’s IT administrator. For more details, see Section 2.2.5.2.1.1: One-Time-Passwords (OTP).

- *Describe a List of High-Level User Goals*

The topmost – goal for users is- **Update the SecurID token user interface specification** using a client/server configuration management system⁵⁶.

The set of -high level user goals considered are:

- **Open the EMC Virtual Private Network (VPN) application.**
- **Log into the system.**
- **Get authorization from the system to the protected resource.**
- **Enable the configuration management system.**
- **Open the SecurID token user interface specification.**
- **Return with goal accomplished.**

- *Define Operators and Write Methods for Accomplishing User Goals*

Method for goal: Open the EMC VPN application (EMC, 2006)

Step 1. Locate the EMC-VPN icon in the bottom taskbar on the screen in Windows.

Step 2. Move mouse over the EMC-VPN icon and double-click the icon of the VPN.

Step 3. Verify that the EMC-VPN pop-up window is opened.

⁵⁶ <http://www.perforce.com>

Step 4. Return with the goal accomplished.

Method for goal: Log into the system

Step 1. Locate and verify that the "Americas East Coast" connection entry is highlighted in the EMC-VPN pop-up window.

Step 2. Move the mouse over to it.

Step 3. Double-click it with left mouse button.

Step 4. Verify that the status bar on the bottom left corner of the pop-up window is displaying "Authenticating user...".

Step 5. Verify that the "VPN Client | User Authentication for Americas East Coast" secondary login pop-up window is opened.

Step 6. Verify that the username field has been automatically filled in (e.g. joedoe).

Step 7. Verify that the cursor is automatically placed within the "Passcode" field.

Step 8. Recall the 4-digit Personal Identification Number (PIN), retrieve it from LTM, and retain it.

Step 9. Type the 4-digit- PIN in the "Passcode" field.

Step 10. Verify that asterisks are displayed while entering the PIN within the "Passcode" field.

Step 11. Forget PIN.

Step 12. Refer to the SecurID 700 token to get the ever-changing (i.e. each 30 seconds) 6-digit numerical password.

Step 13. Read the 6-digit numerical password displayed on the digital readout window on the SecurID token.

Step 14. Retain, memorize, and store the 6-digit numerical password in the STM.

Step 15. Retrieve the 6-digit numerical password from the STM.

Step 16. Verify that the cursor is in the correct place within the "Passcode" field.

Step 17. Append the 6-digit numerical password to the PIN that has been already entered in the "Passcode" field.

Step 18. Verify that asterisks are displayed while entering the 6-digit numerical password in the "Passcode" field.

Step 19. Forget the 6-digit numerical password.

Step 20. Move the mouse over to the "OK" button.

Step 21. Double-click the "OK" button.

Method for goal: Get authorization from the system to the protected resource.

Step 1. Wait for the authentication server to check user's username and passcode against the database.

Step 2. Verify that the system displays "Contacting the security gateway at 137.69.115.17..." message in the status bar at the bottom left corner, and the progress bar at the bottom right is running.

Step 3. Verify that the system displays the "Negotiating security policies..." message (after successfully contacting the security gateway)- in the status bar bottom in the left corner, and ensure that the progress bar at the bottom right is running.

Step 4. Verify that the system displays the "Connected to Americas East Coast" message (after successfully negotiating the security policies) in the status bar bottom in the left corner, and ensure that the progress bar at the bottom right is running.

** The system displays the Internet Protocol (IP) address of the authentication server.

Method for goal: Enable the configuration management system.

Step 1. Double-click on "Continue" button to enable the protected network resource.

Step 2. Verify that the system displays the "VPN Dialer | Banner" pop-up window.

Step 3. Read the "VPN Dialer | Banner" statement.

Step 4. Locate the "VPN Dialer | Banner" padlock icon, think-of as locked (i.e. user is connected with the software VMS) in the status bar bottom right corner, and put this information into the STM.

Step 5. Think-of also that the "VPN Dialer | Banner" padlock icon -indicates that it is a fully secure connection.

Step 6. Return with goal accomplished.

Method for goal: Open the SecurID token user interface specification.

Step 1. Locate the SecurID token user interface specification in the directory of the VMS.

Step 2. Double-click with left mouse button.

Step 3. Go to the "Request a Token" section in the specification.

Step 4. Add a new software token type.

Step 5. Return with goal accomplished.

- *Estimate Total Execution Time*

The time measurement related to the *execution time* for Task_scenario: T2 **Update the SecurID token user interface specification** is listed in seconds below.

Method for goal: Open the EMC-VPN application	Sub-Execution Time(s)
Step 1. Locate the EMC-VPN icon in the bottom taskbar on the screen in Windows	1.21
Step 2. Move mouse over the EMC-VPN icon and double-click the icon of the EMC-VPN	1.11
Step 3. Verify that the EMC-VPN pop-up window is opened	1.21
Total Execution Time	3.53

Method for goal: Log into the system	Sub-Execution Time(s)
Step 1. Locate and verify that the "Americas East Coast" connection entry is highlighted in the EMC-VPN pop-up window	1.21
Step 2. Move mouse over to it	1.11
Step 3. Double-click it with left mouse button	0.41
Step 4. Verify that the status bar on the bottom left corner of the pop-up window is displaying "Authenticating user..."	1.21
Step 5. Verify that the "VPN Client User Authentication for Americas East Coast" secondary login pop-up window is opened	1.21
Step 6. Verify that username field has been automatically filled in (e.g. joedoe)	1.21
Step 7. Verify that the cursor is automatically placed within the "Passcode" field	1.21
Step 8. Recall the 4-digits Personal Identification Number (PIN), retrieve it from LTM and retain it	1.51
Step 9. Type the 4-digits PIN within the "Passcode" field	2.16
Step 10. Verify that asterisks are displayed while entering the PIN within the "Passcode" field.	1.21
Step 11. Forget PIN	1.21
Step 12. Refer to the SecurID 700 token to get the ever-changing (i.e. each 30 seconds) 6-digit number password	2.1
Step 13. Read the 6-digit number password displayed on the digital readout window on the SecurID token	3.0
Step 14. Retain, memorize and store the 6-digit number password in the STM	1.51
Step 15. Retrieve the 6-digit number password from STM	1.21

Step 16. Verify that the cursor is at correct place in the "Passcode" field	1.21
Step 17. Append the 6-digit number password to the PIN that has been already entered in the "Passcode" field	1.50
Step 18. Verify that asterisks are displayed while entering the 6-digit number password in the "Passcode" field	1.21
Step 19. Forget the 6-digit number password	1.21
Step 20. Move mouse over to "OK" button	1.11
Step 21. Double-click the "OK" button	0.41
Total Execution Time	28.13

Method for goal: Get authorization from the system to the protected resource.	Sub-Execution Time(s)
Step 1. Wait for the authentication server to check user's username and passcode against the database.	1.21
Step 2. Verify that the system displays "Contacting the security gateway at 137.69.115.17..." message in the status bar bottom left corner, and the progress bar bottom right is running.	1.21
Step 3. Verify that the system displays "Negotiating security policies..." message (after successfully contacted the security gateway), in the status bar bottom left corner, and the progress bar bottom right is running.	1.21
Step 4. Verify that the system displays "Connected to Americas East Coast" message (after successfully negotiated the security policies) in the status bar bottom left corner, and the progress bar bottom right is running.	1.21
Total Execution Time	4.84

Method for goal: Enable the software configuration management system (CMS).	Sub-Execution Time(s)
Step 1. Double-click on "Continue" button to enable the protected network resource.	0.41
Step 2. Verify that the system displays the "VPN Dialer Banner" pop-up window.	1.21
Step 3. Read the "VPN Dialer Banner" statement.	1.11
Step 4. Locate the "VPN Dialer Banner" padlock icon, think-of as locked (i.e. user is connected with the CMS) in the status bar bottom right corner, and put this information into the STM.	1.51
Step 5. Think-of also that "VPN Dialer Banner" padlock icon as indicating that it is a fully secure connection.	0.29
Total Execution Time	4.53

Method for goal: Open the SecurID token user interface specification.	Sub-Execution Time(s)
Step 1. Locate the SecurID token user interface specification in the directory of the VMS.	1.21
Step 2. Double-click with left mouse button.	0.41
Step 3. Go to the "Request a Token" section in the specification.	2.21
Step 4. Replace token type wording from RSA SecurID Toolbar to RSA SecurID Windows Mobile ⁵⁷	0.45
Total Execution Time	4.28

⁵⁷ RSA-The Security Division of EMC. May 23, 2010 <<http://www.rsa.com/node.aspx?id=2571>>

The Total Execution Time for Task_scenario: T2 Update the SecurID token user interface specification is shown below:

Task_scenario: T2 Update the SecurID token user interface specification	
Methods	Sub-Execution Time(s)
Method for goal: Open the EMC-VPN application	3.53
Method for goal: Log into the system	28.13
Method for goal: Get authorization from the system to the protected resource.	4.84
Method for goal: Enable the software configuration management system (CMS).	4.53
Method for goal: Open the SecurID token user interface specification.	4.28
Total Execution Time	45.31

- *Estimate Total Learning Time:*

The Total Learning Time for Task_scenario: T2 Update the SecurID token user interface specification is shown below:

Total Learning Time = Pure Method Learning Time + LTM Learning Time + Training Procedure Execution Time

Total Learning Time = 629 secs + 42 secs + 0 secs = 671 secs

Sub-Step 3.3 Out-of-Band Authentication {wired and wireless network-based task}:

This authentication task scenario is specified as Task_scenario: T3.

- *Generate Task Description*

Out-Of-Band Authentication (OOBA) is essentially the use of two separate networks, for instance wired and wireless networks, working concurrently in order to authenticate a user. Consider this scenario: Alice, who has her Blackberry mobile device number stored on the bank authentication server, wants to transfer an amount of money to a different bank account. First she logs onto the bank's website with her credentials (username and password). Then she goes to the money transfer section and selects from the drop down menu list to transfer more than \$15,000 to another bank account; this selection triggers the server to send a code to Alice. According to the bank's security policy, this transaction requires an additional authentication method. Third, the bank sends a code to Alice's Blackberry via Short Messaging Service (SMS). Finally, Alice types this code in the TextField on the bank's Website screen and clicks Submit; if this code matches the one the bank has just sent then the transaction is successful. For more detail, see Section 2.2.5.4 Out-Of-Band Authentication (OOBA).

- *Describe a List of High-Level User Goals*

The topmost goal of the use is: **Transfer \$15,000 to the Bank of America**. The set of the user's high level goals considered are:

- Go to the bank website.
- Log on to the system.
- Make an electronic funds transfer.
- Return with goal accomplished.

- *Define Operators and Write Methods for Accomplishing User Goals*

Method for goal: Go to the bank website.

Step 1. Open the Web browser on a desktop computer.

Step 2. Type the bank's website address.

Step 3. Return with the goal accomplished.

Method for goal: Log on to the system.

Step 1. Locate the "Online Banking" log-on section on the bank's home page.

Step 2. Read your online ID that has been already filled in by the system (e.g. a saved online ID such as "go4t****").

Step 3. Locate the "Sign In" button.

Step 4. Move the cursor to the "Sign In" button.

Step 5. Click on the "Sign In" button with the left mouse button.

Step 6. Verify that it is a secure connection by checking if the bank's address bar contains the prefix "https" when the "Confirm that your SiteKey is correct" page is displayed.

Step 7. Verify that it is a secure connection by checking if the Uniform Resource Locator (URL) is correct and no errors were encountered on the same page.

Step 8. Think- of the closed yellow padlock icon as indicating that it is a secure connection on the bottom right of the Windows task bar.

Step 7. Locate the SiteKey phrase field.

Step 8. Read the SiteKey phrase.

Step 9. Recognize the SiteKey phrase (e.g. "Whales are fascinating creatures"), retrieve it from LTM, and retain it.

Step 10. Verify that the SiteKey phrase is correct according to the user's online account setup.

Step 11. Read the SiteKey image.

Step 12. Recognize your SiteKey image (e.g. "a whale image"), retrieve it from LTM and retain it.

Step 13. Verify that the SiteKey image (e.g. "a whale image") is correct according to the user's online account setup.

Step 14. Locate the "Passcode" field.

Step 15. Verify that the cursor has been already placed in the "Passcode" field on the page load.

Step 16. Recall the passcode (e.g. Light7ocean), retrieve it from LTM and retain it.

Step 17. Type the passcode within the "Passcode" field.

Step 18. Verify that asterisks are displayed while entering the passcode within the "Passcode" field.

Step 19. Verify that the passcode has been correctly typed in.

Step 20. Forget passcode

Step 21. Locate the "Sign In" button.

Step 22. Click on "Sign In" button with left mouse button

Step 23. Verify that the system has been granted access and the "Accounts" page is displayed.

Method for goal: Make an electronic funds transfer.

Step 1. Verify that you are in the "Accounts Overview" tab.

Step 2. Locate the "Transfers" tab and click on it.

Step 3. Verify that you are in the "Transfers" page.

Step 4. Verify that you are in the "Make Transfer" sub-tab.

Step 5. Locate the "From" field and select the account.

Step 6. Locate the "To" field and select the account.

Step 7. Locate the "Amount" field and type the amount of \$15,000.

Step 8. Locate the "Continue" button and click on it.

Step 9. Read a warning message on the top of the next page which says that "You will have to provide an additional authentication credential in order to proceed with transfers over \$10,000. Do you want to continue?"

Step 10. Verify the "OK" and "Cancel" buttons.

Step 11. Locate the "OK" button and click on it.

Step 12. Verify that you are in the "Additional Authentication Credentials" page.

Step 13. Locate the instructions section on this page.

Step 14. Read the instructions about the 4-digit numerical code that has to be sent to the Blackberry mobile device using Short Messaging Service (SMS). After 15 seconds, verify that an SMS text message has been sent by the bank to the Blackberry mobile device. Then enter this same code in the "Code" field on this page.

Step 15. Wait for a sound alert for SMS notification to the Blackberry after 15 seconds, indicating that the 4-digit numerical code has been sent by the bank. Then open your "Messages" application on your Blackberry.

Step 16. Listen to the auditory stimulus in the form of a sound alert for SMS notification on the Blackberry with the 4-digit numerical code that has been sent by the bank to the Blackberry-, retrieve it from the auditory LTM, and retain it.

Step 17. The auditory stimulus information (sound alert) disappears from LTM and is no longer available.

Step 18. Open your "Messages" application.

Step 19. Verify that you have received an SMS text message with the 4-digit numerical code sent by the bank.

Step 20. Read the 4-digit numerical code directly from the subject field of the message row.

Step 21. Memorize the 4-digit numerical code- and retain it in the STM.

Step 22. Locate the "Code" field on the "Additional Authentication Credentials" Web page.

Step 23. Locate the "Code" field on the page.

Step 24. Type the 4-digit numerical code in the "Code" field.

Step 25. Verify that asterisks are displayed while entering the code within the "Code" field.

Step 26. Forget code.

Step 27. Verify the "Send" and "Cancel" buttons are displayed below the "Code" field on the page.

Step 28. Click on the "Send" button with the left mouse button.

Step 29. Verify that you have been directed to the confirmation page which states that you have successfully transferred the amount of \$15,000 to the desired destination account.

Step 30. Return with the goal accomplished.

- *Estimate Total Execution Time*

The time measurement related to the *execution time* for **Task_scenario: T3** **Transfer: \$15,000 to the National Bank of Canada** is listed in seconds below.

Method for goal: Go to the bank website.	Sub-Execution Time(s)
Step 1. Open the Web browser in a desktop computer.	0.91
Step 2. Type the bank's website address.	3.01
Step 3. Verify that the bank's Home page is displayed.	1.21
Total Execution Time	5.13

Method for goal: Log into the system	Sub-Execution Time(s)
Step 1. Locate the "Online Banking" log-on section on the bank's home page.	1.21
Step 2. Read your online ID that has been already filled in by the system (e.g. go4t****).	1.21
Step 3. Locate the "Sign In" button.	1.21
Step 4. Move cursor to "Sign In" button.	1.11
Step 5. Click on "Sign In" button with left mouse button	0.21
Step 6. Verify that it is a secure connection by checking if the bank's address bar contains the prefix "https" when the "Confirm that your SiteKey is correct" page is displayed.	1.21
Step 7. Verify that it is a secure connection by checking if the Uniform Resource Locator (URL) is correct and no errors were encountered on the same page.	1.21
Step 8. Think-of the closed yellow padlock icon as indicating that it is a secure connection on the bottom right of the Windows task bar.	0.29
Step 7. Locate the SiteKey phrase field.	1.21
Step 8. Read the SiteKey phrase.	1.51
Step 9. Recognize the SiteKey phrase (e.g. "Whales are fascinating creatures"), retrieve it from LTM and retain it.	1.21
Step 10. Verify that the SiteKey phrase is correct according to the user's online account setup.	1.21
Step 11. Read the SiteKey image.	1.21
Step 12. Recognize your SiteKey image (e.g. "a whale image"), retrieve it from LTM and retain it.	1.21
Step 13. Verify that the SiteKey image (e.g. "a whale image") is correct according to the user's online account setup.	1.21

Step 14. Locate the "Passcode" field.	1.21
Step 15. Verify that the cursor has been already placed in the "Passcode" field on the page load.	1.21
Step 16. Recall the passcode (e.g. Light7ocean), retrieve it from LTM and retain it.	1.21
Step 17. Type the passcode within the "Passcode" field.	0.51
Step 18. Verify that asterisks are displayed while entering the passcode within the "Passcode" field.	1.21
Step 19. Verify that the passcode has been correctly typed in.	1.21
Step 20. Forget passcode	1.21
Step 21. Locate the "Sign In" button.	1.21
Step 22. Click on "Sign In" button with left mouse button	0.21
Step 23. Verify that the system has been granted access and the "Accounts" page is displayed.	1.21
Total Execution Time	26.83

Method for goal: Make an electronic funds transfer.	Sub-Execution Time(s)
Step 1. Verify that you are in the "Accounts Overview" tab within "Accounts" page.	1.21
Step 2. Locate the "Transfers" tab and click on it.	1.21
Step 3. Verify that you are in the "Transfers" page.	1.21
Step 4. Verify that you are in the "Make Transfer" sub-tab.	1.21
Step 5. Locate the "From" field and select the account.	1.21
Step 6. Locate the "To" field and select the account.	1.21
Step 7. Locate the "Amount" field and type the amount of \$15,000.	1.21
Step 8. Locate the "Continue" button and click on it.	1.21
Step 9. Read a warning message on the top of the next page which says that "You will have to provide an additional authentication credential in order to proceed with transfers over \$10,000. Do you want to continue?"	1.21

Step 10. Verify the "OK" and "Cancel" buttons.	1.21
Step 11. Locate the "OK" button and click on it.	1.21
Step 12. Verify that you are in the "Additional Authentication Credentials" page.	1.21
Step 13. Locate the instructions section on this page.	1.21
Step 14. Read the instructions about the 4-digit number code that has to be sent to the Blackberry mobile device using Short Messaging Service (SMS). After 15 seconds, verify that an SMS text message has been sent by the bank to the Blackberry mobile device. Then enter this same code in the "Code" field on this Web page.	10.01
Step 15. Wait for a sound alert for SMS notification to the Blackberry after 15 seconds indicating that the 4-digit number code has been sent by the bank.	15.00
Step 16. Listen the auditory stimulus in the form of a sound alert for SMS notification on the Blackberry with the 4-digit number code has been sent by the bank to the Blackberry.), retrieve it from the auditory LTM, and retain it.	1.21
Step 17. Auditory stimulus information (sound alert) dissipates from LTM and is no longer available.	1.01
Step 18. Open the "Messages" application on the Blackberry.	2.01
Step 19. Verify that you have received a SMS text message with the 4-digit number code sent by the bank.	1.21
Step 20. Read the 4-digit number code directly from the subject field of the message row.	1.21
Step 21. Memorize the 4-digit number code, and retain it in the STM.	6.00
Step 22. Locate the "Code" field on the "Additional Authentication Credentials" web page.	1.21
Step 23. Type the 4-digit number code in the "Code" field.	2.16
Step 24. Verify that asterisks are displayed while entering the code within the "Code" field.	1.21

Step 25. Forget code.	1.21
Step 26. Verify the "Send" and "Cancel" buttons are displayed below the "Code" field on the page.	1.21
Step 27. Click on "Send" button with left mouse button	0.21
Step 28. Verify that you have been directed to the confirmation page which states that you have successfully transferred the amount of \$15,000 to the desired destination account.	1.21
Total Execution Time	61.81

The Total Execution Time for **Task_scenario: T3 Transfer \$15,000 to the Bank of America** is shown below:

Task_scenario: T3 Transfer \$15,000 to the Bank of America	
Methods	Sub-Execution Time(s)
Method for goal: Go to the bank website.	5.13
Method for goal: Log into the system	26.83
Method for goal: Make an electronic funds transfer	61.81
Total Execution Time	93.77

- *Estimate Total Learning Time:*

The Total Learning Time for **Task_scenario: T3 Transfer \$15,000 to the Bank of America** is shown below:

Total Learning Time = Pure Method Learning Time + LTM Learning Time + Training Procedure Execution Time

Total Learning Time = 833 secs + 102 secs + 0 secs = 935 secs

Sub-Step 3.4 Fingerprint Recognition {wireless network task}:

- *Generate Task Description*

This task is basically a wireless network and electronic access control-based task using a portable Universal Serial Bus (USB) device that allows remote employees to swipe their finger to access corporate network resources (Figure 2-24). It recognizes a fingerprint, providing a secure way of accessing a protected resource. Users have to install the USB fingerprint suite software on the desktop or laptop computer; then plug the USB fingerprint into the USB port; and finally - swipe their finger on the fingerprint reader to log - into Windows, for example, or to access - password-protected Web sites. A pre-condition is that users have to set up a one-time registration for all their accounts to authenticate with the USB, but once that is set up, users can make use of the USB fingerprint. For more details on fingerprint recognition, see Section 2.2.5.9.1: Fingerprint Recognition.

- *Describe a List of High-Level User Goals*

The topmost user's goal is: Access a file on a personal laptop. The set of the user's high level goals considered are:

- Log into the system.
- Go to the file directory.
- Open the file.
- Return with goal accomplished.

- *Define Operators and Write Methods for Accomplishing User Goals*

Method for goal: Log into the system

Step 1. Read fingerprint logon Welcome screen containing finger image and "Password" field on the laptop computer.

Step 2. Refer to the Universal Serial Bus (USB)-based biometric fingerprint reader.

Step 3. Locate the fingerprint sensor on the USB fingerprint reader.

Step 4. Move finger to the USB fingerprint reader.

Step 5. Position last knuckle joint over the center of the fingerprint sensor.

Step 6. Swipe the finger without lifting it over the fingerprint sensor, and read computer screen.

Step 7. Verify that you have been granted access to the system.

Step 8. Return with goal accomplished.

Method for goal: Go to the file directory.

Step 1. Open the file manager application (e.g. Windows or MAC Explorer) on the laptop computer.

Step 2. Verify that the Explorer application is opened.

Step 3. Locate the C: directory on the Explorer.

Step 4. Move cursor to the C: directory and click on it.

Step 5. Verify that the Explorer application is on the C: directory.

Step 6. Return with goal accomplished.

Method for goal: Open the file.

Step 1. Locate the "read.txt" file on C: directory.

Step 2. Move cursor to the "read.txt" file icon.

Step 3. Double-click on "read.txt" file icon.

Step 4. Verify that the "read.txt" file is opened.

Step 5. Access the "read.txt" file.

Step 6. Return with goal accomplished.

- *Estimate Total Execution Time*

The time measurement related to the *execution time* for **Task_scenario: T4**

Access a file on a personal laptop is listed in seconds below.

Method for goal: Log into the system	Sub-Execution Time(s)
Step 1. Read fingerprint logon Welcome screen containing finger image and "Password" field on the laptop computer.	2.11
Step 2. Refer to the Universal Serial Bus (USB)-based biometric fingerprint reader.	1.10
Step 3. Locate the fingerprint sensor on the USB fingerprint reader.	1.21
Step 4. Move finger to the USB fingerprint reader.	1.11
Step 5. Position last knuckle joint over the center of the fingerprint sensor.	1.11
Step 6. Swipe the finger without lifting it over the fingerprint sensor.	1.31
Step 7. Verify that you have been granted access to the system.	1.21
Total Execution Time	9.16

Method for goal: Go to the file directory	Sub-Execution Time(s)
Step 1. Open the file manager application (e.g. Windows or MAC Explorer) on the laptop computer	0.41
Step 2. Verify that the Explorer application is opened	1.21
Step 3. Locate the C: directory on the Explorer	1.21
Step 4. Move cursor to the C: directory and click on it	1.11
Step 5. Verify that the Explorer application is on the C: directory	1.21
Total Execution Time	5.15

Method for goal: Open the file	Sub-Execution Time(s)
Step 1. Locate the "read.txt" file on C: directory.	2.21
Step 2. Move cursor to the "read.txt" file icon.	1.11
Step 3. Double-click on "read.txt" file icon.	0.41
Step 4. Verify that the "read.txt" file is opened.	1.21
Step 5. Access the "read.txt" file.	1.21
Total Execution Time	6.15

The Total Execution Time for Task_scenario: T4 Access a file on a personal laptop is shown below:

Task_scenario: T4 Access a file on a personal laptop	
Methods	Sub-Execution Time(s)
Method for goal: Log into the system	9.16
Method for goal: Go to the file directory	5.15
Method for goal: Open the file	6.15
Total Execution Time	20.46

- *Estimate Total Learning Time:*

The Total Learning Time for Task_scenario: T4 Access a file on a personal laptop is shown below:

Total Learning Time = Pure Method Learning Time + LTM Learning Time +
Training Procedure Execution Time

Total Learning Time = 289 secs + 48 secs + 0 secs = 337 secs

3.2.2.3.5 A Time Level Analysis of the NGOMSL

This section presents a time analysis of the data gathered for each set of four task scenarios. As described above, the design information obtained from NGOMSL has been the operator sequences, execution times, and procedure learning times.

As shown in Table 3.1, the user took 28.85 seconds which, is the total execution time (TET) to check business email (T1) using the Password/PIN authentication method and so forth for the tasks T2, T3, and T4. It is worth mentioning that what is important is not to measure the TET the user has spent in each task as a whole, but rather just the TET to “Log into the system”. It is irrelevance to only measure how long it would take, for instance, to check business e-mail (i.e. 28.85 seconds), update a specification (i.e. 45.31 seconds), make an electronic funds transfer (i.e. 93.77 seconds), or access a file (i.e. 20.46 seconds). These tasks can significantly vary in

terms of time, application type, and the context in which they have been performed. However, the TET does vary depending on the type of the authentication method used. In fact it takes more time if the user employs an Ooba method rather than another method- because the amount of user interaction when authenticating to a system with Ooba is more demanding than the other authentication methods as shown in Table 2.

Task_Scenario	Description	Authentication Method	Total Execution Time(s)
T1	Check Business E-mail	Password/PIN	28.85
T2	Update the SecurID token UI spec	OTP	45.31
T3	Transfer 15,000 to the Bank of America	Ooba	93.77
T4	Access a file on a personal laptop	Fingerprint	20.46

Table 3.1: Total Execution Time by task scenario.

The results of this research show the total execution time for the set of four authentication benchmark methods, which is the profile for the Method for goal: Log into the system in Table 3.2. The profile includes the total time in seconds spent using this method and the percentage of the time spent on it. What this thesis is more concerned with is -an investigation of the authentication portions of the tasks scenarios. These portions are in fact the time related to the Method for goal: Log into the system.

Task_Scenario	Method for goal	Authentication Method	% of Total	Total Execution Time(s)
T1	Log into the system	Password/PIN	83.93	23.25
T2		OTP	12.75	28.13
T3		OOBA	25.16	26.83
T4		Fingerprint	1.88	9.16

Table 3.2: Execution time by task scenario, authentication method type, and Method for goal: Log into the system.

As mentioned previously, the main factors influencing the amount of time a user spends authenticating to a system are the number of different artifacts to interact with and the authentication method type. For instance, in OOBA method Alice needs to interact with the bank's Website and then a mobile device in order to accomplish the Method for goal: Log into the system. Also, the authentication method type such as Password/PIN takes more time to be performed, 23.25 seconds, when compared to Fingerprint recognition, 9.16 seconds. The former is a Knowledge-Based Authentication (KBA), which requires users to prove the knowledge of a single secret, memorize items, and recall them when accessing a specific system. On the other hand, the latter is Biometrics, which recognizes users physically through their fingers; no cognitive process is directly involved.

Using OTP takes a little more time than OOBA, given that either users need to interact with different artifacts and make use of KBA which directly involves cognitive processes. With OTP users are required to refer to a hardware authentication token, then type the code displayed there on their application (e.g. VPN application). In addition, users need to remember the PIN (i.e. 4-digit) but not the password (i.e. strong password like Rtyr78nM!), which facilitates memory retrieval, although this authentication method is the one that takes more time.

As expected, the fingerprint authentication method (Biometrics)- takes the least amount of time out of all the methods. No cognitive process is directly involved (e.g. not KBA), and there is minimal interaction with artifacts when using a USB drive. It

is important to note that it is the acquisition of the user credentials (e.g. the biometric reader captures fingerprint samples from users) that takes less time when compared with the other authentication methods presented in Table 3.2. The exact authentication processing time can vary considerably depending on the infrastructure, the equipment, and also on different versions of (the same) authentication methods.

3.2.2.4 Step 4: *Develop the Authentication Risk Assessment Matrix*

If we don't identify where the most critical vulnerabilities are – when related either to security and usability – then how can we secure our system's information infrastructure? The Authentication Risk Assessment Matrix must be developed prior to the development of the usable security inspection method itself. It is in fact a crucial step to acknowledge and understand the main threats and security vulnerabilities related especially to online user authentication (user-to-machine). It determines which “Security Review” should be considered within each “Usability Criterion” in the USS. To this end, Computer Security research has been conducted in order to identify and classify those threats and vulnerabilities. Among industry recognized security sources such as (OWASP, 2009; CVE, 2009; SANS, 2009; CIO, 2009), the Open Web Application Security Project (OWASP) was finally selected as the primary data source for identifying the top ten security threats and vulnerabilities. The primary attacks considered in this dissertation are the following: eavesdropper, Man-In-The-Middle (MITM), replay, session hijacking, and verifier impersonation attacks.

The Open Web Application Security Project (OWASP) is an open-source application security project. The OWASP community consists of corporations, educational organizations, and a variety of security experts worldwide who share their knowledge of vulnerabilities, threats, attacks and countermeasures.

This community works to generate freely-available articles, methodologies, documentation, tools, and technologies. OWASP is not affiliated with any technology organization or company, although it supports the knowledgeable use of security technology. OWASP has avoided affiliation as it believes freedom from organizational pressures may make it easier for it to offer impartial, practical, cost-effective information about application security. As a matter of fact, the U.S. Federal Trade Commission strongly recommends that organizations use the OWASP Top Ten and ensure that their partners do the same. In addition, the U.S. Defense Information Systems Agency has listed the OWASP Top Ten as key best practices that should be used as part of the DOD Information Technology Security Certification and Accreditation (C&A) Process (DITSCAP). For these reasons, this thesis has adopted OWASP as the primary source for identifying the most critical security vulnerabilities.

The OWASP Top Ten provides a powerful awareness document and minimum standard for web application security (OWASP, 2009). It represents a broad consensus on the most critical web application security flaws, including authentication. For more information regarding the methodology used by OWASP to select the security vulnerabilities, go to: http://www.owasp.org/index.php/Top_10_2007-Methodology.

It is worth noting that 5 out of the top 10 security vulnerabilities are directly or indirectly related to authentication. The top ten most critical Web application security vulnerabilities are described as follows (the symbol “⚙” located next to the security vulnerability means that it is specifically related to authentication):

1. *Unvalidated input*: Information from web requests is not validated before being used by a web application. Attackers can use these flaws to attack back-end components through a web application.
2. *Broken access control* ⚙: Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorized functions.

3. *Broken authentication and session management* 🚩: Account credentials and session tokens are not properly protected. Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.
4. *Cross Site Scripting (XSS) flaws* 🚩: The web application can be used as a mechanism to transport an attack to an end user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user.
5. *Buffer overflows*: Web application components in some languages that do not properly validate input can be crashed and, in some cases, used to take control of a process. These components can include CGI, libraries, drivers, and web application server components.
6. *Injection flaws* 🚩: Web applications pass parameters when they access external systems or the local operating system. If an attacker can embed malicious commands in these parameters, the external system may execute those commands on behalf of the web application.
7. *Improper error handling*: Error conditions that occur during normal operations are not handled properly. If an attacker can cause errors to occur that the web application does not handle, they can gain detailed system information, deny service, cause security mechanisms to fail, or crash the server.
8. *Insecure storage*: Web applications frequently use cryptographic functions to protect information and credentials. These functions and the code to integrate them have proven difficult to code properly, frequently resulting in weak protection.
9. *Denial of service (DOS)*: Attackers can consume Web application resources to the point where other legitimate users can no longer access or use the

application. Attackers can also lock users out of their accounts or even cause the entire application to fail.

10. *Insecure configuration management* ☛: Having a strong server configuration standard is critical to a secure web application. These servers have many configuration options that affect security and are not secure out of the box.

The following paragraphs present the Authentication Risk-Assessment Matrix (Table 3.3), which describes the user authentication assets, threats, and vulnerabilities along with their corresponding descriptions and mitigation strategies. It also shows the types of rating scales for Threat, Vulnerability, CIA (Confidentiality, Integrity, and Authorization model), Probability, Asset Value and Asset Exposure Classifications, Total Impact and Total Risk Ratings, and finally Risk Reduction Strategy.

Matrix Legend:

☛=A specific authentication/OWASP top ten security vulnerability.

T=Threat.

V=Vulnerability.

CP/E=Compromise/Exploit.

RA=Risk Assessment

OP=Overall Probability Rating is the sum of the Threat and the Vulnerability Rating ($OP=T+V$).

TI=Total Impact Rating is the sum of the Asset Value Classification and the Asset Value Exposure ($TI=AVC+AVE$).

TR=Total Risk Rating is the product of the Overall Probability and the Total Impact ($TR=OP \times TI$).

Table 3.3: Authentication Risk-Assessment Matrix.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating	OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Value Exposure	TI= AVC + AEC	RA TR= OP x TI	Risk Reduction Strategy
1. Password Personally identifiable medical Information stored on Structured Query Language (SQL) Server	Account credentials of data entry clerk stolen	Overly complex password requirements cause users to write down passwords and leave them in obvious places.	CIA	3 Medium	3 Medium	6	4 Substantial	4 Serious	8	48	Mitigate by reducing password complexity requirements, - enforcing policy to not leave passwords in obvious places, and providing user training in password use. Compromise of SQL data could result in large fines as a result of HIPAA ⁵⁸ violations. Also, loss of public confidence could result in long-term loss of business.
2. Password Personally identifiable medical Information stored on SQL Server	Help desk resets password for an account used by data entry clerk based on request from unauthorized individual (Social Engineering attack ⁵⁹).	Lack of policies and procedures in place to verify identity of individual requesting password reset.	CIA	2 Low	2 Medium	4	4 Substantial	4 Serious	8	24	Mitigate by implementing tighter procedures to verify identity- and by delegating password change permissions to small business units where requests for password changes come from individuals known to administrator. Compromise of SQL data could result in large fines as a result of HIPAA violations. Also, loss of public confidence could result in long-term loss of business.

⁵⁸ The Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy Rule <<http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html>>

⁵⁹ A social engineering attack is one in which the intended victim is somehow tricked into doing what the attacker requests.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating	OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
3. Password Non-proprietary and non-confidential company data.	Summer intern's password guessed by attacker.	Use of familiar names and words in passwords.	CIA	1 Very Low	3 Medium	4	1 Negligible	1 Negligible	2	8	Mitigate by providing user education to all staff, including temporary staff. Compromised data results in minimal loss- of productivity, etc. Costs of loss easily absorbed.
4. Password Microsoft Exchange Hosted Services (EHS)	Attacker runs online brute force attack to determine password of account used for EHS and causes account lockout on service account.	Account lockout thresholds set on domain. All versions of UNIX/Linux/Mac OS Server may be affected by accounts having weak or dictionary-based passwords for authentication. Most Unix/Linux systems include multiple standard services in their default installation. All versions of Unix/Linux/Mac OS Server are potentially at risk from improper and default configurations.	A	2 Low	3 Medium	5	3 Medium	3 Moderate	6	30	<ol style="list-style-type: none"> 1. Disable account lockout thresholds on domain accounts, set- them to a higher value, or - set- a low value for the lockout duration. Loss of EHS could result in loss of productivity and affect many users. 2. Don't use default passwords on any account. 3. Enforce a strong password policy. Don't permit weak passwords or passwords based on dictionary words to resist brute force attacks. Use public key authentication mechanism for SSH⁶⁰ to thwart such attacks. 4. Limit the number of failed login attempts to exposed services.

⁶⁰ Secure Shell or SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices. Used primarily on Linux and UNIX based systems to access shell accounts.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating	OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
											<p>5. Limit the accounts that can log in over the network; root should not be one of them.</p> <p>6. Prohibit shared accounts and don't use generic account names like tester, guest, sysadmin, admin, etc..</p> <p>7. Log failed login attempts. A large number of failed logins to a system may require a further check on the system to see if it has been compromised.</p> <p>8. Consider using certificate based authentication.</p> <p>9. If your Unix system allows the use of Pluggable Authentication Modules (PAM), implement those that check the password's strength.</p> <p>10. Avoid service interactions and misconfigurations. Where possible, limit the functions of the host⁶¹. Misconfigurations in multiple services may often increase the risk to a service.</p>

⁶¹ A host, network host, or Internet host is a computer connected to the Internet.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall V Rating	Overall PC/E OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
5. Password Company data residing on individual workstations of employees in Human Resources department.	Attacker runs brute force attack against local Security Accounts Manager ⁶² (SAM) of workstation in attempt to acquire account credentials of local administrator account.	Account lockout thresholds not set on member workstations.	CIA	2 Low	3 Medium	5	3 Moderate	4 Serious	7	35	<ol style="list-style-type: none"> 1. Administrator User Account: The password should be unique and complex. For example, mitigate by requiring account lockout thresholds on workstations used by human resources employees. Loss of confidentiality of HR data could subject the company to severe legal sanctions and other consequences; 2. Guest User Account: This account is disabled by default and should remain configured that way. 3. New User Accounts: There should be few, if any, additional user accounts in the local SAM; 4. Administrators Group: The membership should be limited.

⁶² The Security Account Manager (SAM) is a database present on servers running Windows Server 2003 that stores user accounts and security descriptors for users on the local computer.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall V Rating	Overall PC/E OP= T+V	Overall Asset Value Classification	Overall Exposure/Impact Asset Exposure Classification	TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
											<p>5. Account Policies: It is best to have the local SAM account policy meet or exceed the company security policy setting for the domain account policy.</p> <p>6. User Rights: The user rights on every server should be audited to ensure that they are configured properly and no additional users have privileged access to the server.</p> <p>7. Audit Policy: An important setting for both servers and clients to keep track of when users logon and what resources are accessed and when.</p>

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E		Overall Exposure/Impact		RA	Risk Reduction Strategy
					V Rating	OP= T+V	Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	TR= OP x TI
6. Password Network Infrastructure	Account credentials of administrator stolen by individual “shoulder surfing” administrator performing daily tasks.	Lack of multifactor authentication. Lack of secure work area for administrators.	CIA	2 Low	5	7	5 High	5 Severe	10	70
										Mitigate by requiring administrators to use multi-factor authentication (e.g. smart card plus Personal Identification Number [PIN]), by providing secure work area for administrators, and by providing training to administrators regarding the use of credentials when others are able to view them. Theft of administrative credentials could have a severe impact throughout the company.
7. Password Random company data throughout organization	Attacker exploits older authentication mechanisms.	Windows 95 and 98 clients using LAN Manager ⁶³ (LM). It uses a particularly weak method of hashing a user's password known as the LM hash algorithm.	CIA	2 Low	3 Medium	5	4 Substantial	4 Serious	8	40
										Avoid by installing the Active Directory Services client on Windows 9x workstations and disabling LM authentication throughout domain. Possibility for extensive losses of confidentiality; however, actual total impact rating depends on the actual account and data compromised.

⁶³ A network operating system from Microsoft that runs as a server application under OS/2. It supports DOS, Windows and OS/2 clients. LAN Manager was superseded by Windows NT Server, and many parts of LAN Manager are used in Windows NT and 2000.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating	OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
8. Password Horizontal privilege escalation	Potential web application vulnerabilities that may lead to this condition include- easily guessable passwords.	Consider this scenario: Alice has access to her bank account in an Internet Banking application. <ul style="list-style-type: none"> Bob has access to his bank account in the same Internet Banking application. The vulnerability occurs when Alice is able to access Bob's bank account by performing some sort of malicious activity. 	CIA	2 Low	3 Medium	5	4 Substantial	4 Serious	8	40	<ol style="list-style-type: none"> 1. Remove unused services/apps from network devices. 2. Assign and enforce a strict password policy (e.g. strong password, changed regularly) and ensure all device access is password protected. 3. Disable unused system accounts and remove unused user accounts. 4. Tighten privilege levels on apps and storage locations; do not accept defaults such as windows which, allows "everyone" access by default. 5. Regularly study log files and baseline system files for later comparison for tampering.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E		Overall Exposure/Impact		RA	Risk Reduction Strategy
					V Rating	OP= T+V	Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	TR= OP x TI
9. Password	Allowing password aging ⚠	Allowing password aging to occur unchecked can result in the possibility of diminished password integrity.	CIA	1 Very Low	3 Medium	4	1 Negligible	1 Negligible	2	8
10. Password	Buffer overflows in the username or password of a login feature. ⚠	Buffer overflow using long strings of "A" characters in username/password during authentication. Web application components (e.g. CGI, libraries, drivers, web application server) in some languages that do not properly validate input can be crashed and used to take control of a process.	CIA	3 Medium	3 Medium	4	5 High	5 Severe	9	36

The recommendation that users change their passwords regularly and do not reuse passwords is universal among security experts. In order to enforce this, it is useful to have a mechanism that notifies users when passwords are considered old and that requests that they replace them with new, strong passwords. In order for this functionality to be useful, however, it must be accompanied with documentation which stresses how important this practice is and which makes the entire process as simple as possible for the user.

Keeping systems up-to-date with the most current security patches and using for example "McAfee Enterscept"⁶⁴ will protect servers against these powerful threats.

⁶⁴ The Enterscept Management System delivers comprehensive, enterprise-class management for Enterscept's intrusion prevention agents. McAfee Secure Computing, 2009. http://www.securesynergy.com/services/predictabilitymanagement/ips/enterscept/pdf/ds_enterscept_mgt_system.pdf

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E		Overall Exposure/Impact		RA	Risk Reduction Strategy
					V Rating	OP= T+V	Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	TR= OP x TI
10. Password	Cross Site Scripting (XSS) flaws: The web application can be used as a mechanism to transport an attack to an end user's browser. A successful attack can disclose the end user's session token, attack the local machine, or spoof content to fool the user. ☛	XSS using well-formed SCRIPT tags, especially in the Username / Password of an authentication routine.	CIA	2 Low	2 Low	4	3 Medium	3 Moderate	6 Medium	24
11. Password	SQL injection using " ' " in the-Username/ password of an authentication routine; AND "ID" or other identifier field ☛	Attackers that can compromise passwords, usernames, or database fields can defeat authentication and/or have access to sensitive data.	CIA	4 High	4 High	8	4 Substantial	4	8	64
										Developers need to either stop writing dynamic queries and/or prevent user supplied input which contains malicious SQL from affecting the logic of the executed query. There are several techniques for preventing SQL Injection vulnerabilities (OWASP_09).

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating OP= T+V	Overall Exposure/Impact Asset Value Classification Asset Exposure Classification TI= AVC + AVE	RA TR= OP x TI	Risk Reduction Strategy
12. Password Hard-coded or undocumented account/ password	Insiders with privileged password access	Client-side systems with hard-coded passwords	CIA	2 Low	4	3 Medium 3 Medium 6	24	Users should not circumvent password entry with auto logon, application remembering, embedded scripts, or hard-coded passwords in client software for systems that process/store mission critical and/or confidential data. Users should always deny having a password "remembered".
13. Password Authentication Tokens	Broken authentication and session management	Account credentials and session tokens are not properly protected. Attackers that can compromise passwords, keys, session cookies, or other tokens can defeat authentication restrictions and assume other users' identities.	CIA	3 Medium	6	3 Medium 3 Medium 6	36	Unless all authentication credentials and session identifiers are protected with Secure Sockets Layer (SSL) ⁶⁵ at all times and protected against disclosure from other flaws, such as cross site scripting, an attacker can hijack a user's session and assume their identity.

⁶⁵ Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data – a public key known to everyone and a private or secret key known only to the recipient of the message.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E		Overall Exposure/Impact		RA	Risk Reduction Strategy
					V Rating	OP= T+V	Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	TR= OP x TI
14. Passwords Random company data throughout the network.	Attacker intercepts network authentication packets that contain password hashes and attempts to break them offline.	NTLM ⁶⁶ and LM ⁶⁷ authentication used on the network.	CIA	2 Low	3 Medium	5	4 Substantial	4	8	40
										Mitigate by upgrading clients to Windows 2000 or better. Possibility for extensive losses of confidentiality; however, actual total impact rating depends on the actual account and data compromised (It can also be mitigated by securing physical infrastructure of network cabling). One of the reasons for the relatively high value for the overall risk is that it is relatively easy to perform this exploit using downloadable tools.

⁶⁶ NTLM is a suite of authentication and session security protocols used in various Microsoft network protocol implementations. NTLM authentication is a challenge-response scheme, consisting of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge), and Type 3 (authentication).

⁶⁷ The LM authentication protocol, also known as LAN Manager and LANMAN, was invented by IBM and used extensively by Microsoft operating systems prior to NT 4.0. It uses a password encrypting technology that is now considered insecure.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E			Overall Exposure/Impact			RA	Risk Reduction Strategy
					V Rating	OP= T+V	Asset Value Classification	Asset Exposure Classification	TI= AVC + AVE	TR= OP x TI		
15 .OOBA Trojan horse (TH)	Sending out spam email inviting users to access a website that will install a smart Trojan on user's client computer.	The TH will observe activities on the client computer and get into action when, for example, the user starts an online banking session. When the user specifies a funds transfer transaction, the TH will alter the amount and destination account without displaying the alteration on the screen. The online bank will thus receive a transaction request with the	CIA	2 Low	3 Medium	5	3 Substantial	3 Moderate	6	30		<p>This type of attack is difficult to detect. A strong network security policy with no unauthorized downloads is usually the best way to defend against Trojan horses.</p> <p>Also perform random file comparisons of key binaries on hosts to known, good binaries, confirming that key binaries haven't been compromised.</p>

Definitions:

A threat is a circumstance, event, or person with the potential to cause harm to a system in the form of destruction, disclosure, data modification, and/or Denial of Service (DoS).

Vulnerability is a hole or a weakness in the application, which can be a design flaw or an implementation bug that allows an attacker to cause harm to the stakeholders of an application. Stakeholders include the application owner, application users, and other entities that rely on the application.

Risk Assessment is a computation of risk. Risk is a threat that exploits some vulnerability that could cause harm to an asset. The risk algorithm computes the risk as a function of the assets, threats, and vulnerabilities. One instance of a risk within a system is represented by the formula $\text{Risk} = \text{Asset} \times \text{Threat} \times \text{Vulnerability}$. Total risk for a network equates to the sum of all the risk instances.

Ratings:

The ratings described by Shadow (2008) in the next paragraphs have been applied in the Authentication Risk Assessment Matrix.

Threat, Vulnerability and Overall Probability Scales:

Threat Rating	
1.	Very low or negligible probability of threat. Little or no motivation to launch attack. Almost no probability of threat for non-human threat agent.
2	Low probability of threat.
3.	Medium probability of threat.
4.	High probability of threat.
5.	Extremely high, almost certain probability of threat.

Vulnerability Rating	
1.	Very low or negligible. Vulnerability requires extensive effort/knowledge/resources to exploit; exploit of vulnerability does not lead to exposure of additional vulnerabilities in other services/systems/processes.
2	Low. Vulnerability requires significant effort/knowledge/resources to exploit. Low probability exploit will create exposure to additional vulnerabilities in and threats to other services/systems/processes.

3.	Medium. Vulnerability requires moderate amount of effort/knowledge/resources to exploit. Moderate probability exploit will create exposure to additional vulnerabilities in and threats to other services/systems/processes.
4.	High. Vulnerability requires some resources to exploit. High probability exploit will create exposure of additional vulnerabilities in and threats to other systems/services/processes.
s5.	Very High. Vulnerability requires little knowledge, effort, or skills to exploit. Very high probability that exploit will create exposure of additional vulnerabilities in and threats to other systems/services/processes.

Overall Probability Matrix (Threat / Vulnerability)

	Vulnerability:					
Threat Rating	0	1	2	3	4	5
Very Low	1	2	3	4	5	6
Low	2	3	4	5	6	7
Medium	3	4	5	6	7	8
High	4	5	6	7	8	9
Very High	5	6	7	8	9	10

Exposure / Impact Rating Scales

Asset Value Classification	
1.	Negligible asset value. Negligible or no impact on business if confidentiality or integrity of asset is compromised. Compromise of availability results in negligible or no increase of support costs or loss of productivity.
2	Low asset value. Low impact on business that cannot be measured if confidentiality or integrity of asset is compromised. Compromise of availability results in distractions that are easily absorbed by internal business process – possible slight increase in support costs.
3.	Medium asset value. Medium impact on business (internal processes, etc.) if confidentiality or integrity is compromised, resulting in revenue loss and increase in support costs. Compromise of availability results in work delays with noticeable increase in support costs and loss of productivity.
4	Substantial asset value. Serious impact on business if confidentiality or integrity of asset is compromised, resulting in loss of profitability or success. Compromise of availability results in work interruptions, causing a quantifiable increase in support costs or delay in business commitments (e.g., clients and customers are unable to connect to Web sites, unable to make commitments for contract deliverables on time, etc).

5.	High asset value. Severe or catastrophic impact on business if confidentiality of assets is compromised, resulting in high losses to business profitability or success. Compromise of availability results in significant work stoppages, causing substantial increase in support costs or cancellation of business commitments.
----	--

Exposure Classification	
1.	Negligible or no loss or asset confidentiality, integrity, or availability. Effects of compromise to asset severely contained with no subsequent threat of compromise to other assets.
2.	Low loss of asset confidentiality, integrity, or availability. Effects of compromise to assets tightly contained with negligible or low subsequent threat to other assets.
3.	Moderate or limited loss of asset confidentiality, integrity, or availability. Effects of compromise to assets can involve more than one system or service and cause an increased threat to other assets. Compromise or exploit may be externally visible.
4.	Serious loss of asset confidentiality, integrity, or availability. Effects of compromise are likely to have negative effects on other assets and cause a noticeable increase in threats to other assets. Compromise or exploit may be externally visible.
5	Severe or complete loss of asset confidentiality, integrity, or availability. Results in significant increase in threats to other assets. High probability compromise or exploit may be externally visible.

Total Impact Matrix

	Exposure Factor: 1 = Negligible, 2 = Low, 3 = Medium, 4 = Serious, 5 = Severe					
Asset Value Classification	0	1	2	3	4	5
1 = Very low or negligible	1	2	3	4	5	6
2 = Low	2	3	4	5	6	7
3 = Medium	3	4	5	6	7	8
4 = Substantial	4	5	6	7	8	9
5 = High	5	6	7	8	9	10

Overall Risk Matrix (Overall Probability x Total Impact)

	Total Impact										
Overall Probability	0	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10	
2	2	4	6	8	10	12	14	16	18	20	
3	3	6	9	12	15	18	21	24	27	30	
4	4	8	12	16	20	24	28	32	36	40	
5	5	10	15	20	25	30	35	40	45	50	
6	6	12	18	24	30	36	42	48	54	60	
7	7	14	21	28	35	42	49	56	63	70	
8	8	16	24	32	40	48	56	64	72	80	
9	9	18	27	36	45	54	63	72	81	90	
10	10	20	30	40	50	60	70	80	90	100	

Risk Summary Ranges	
Low	1 - 19
Medium	20 - 40
High	41 - 100

Additional explanation of some of the security vulnerabilities mentioned in the Authentication Risk Assessment Matrix is provided in the remainder of this section.

Cross Site Scripting (XSS)⁶⁸

XSS is generally believed to be one of the most common application layer hacking techniques. Generally speaking, XSS refers to that hacking technique that leverages vulnerabilities in the code of a web application to allow an attacker to send malicious content from an end-user and collect some type of data from the victim. Today, websites rely heavily on complex web applications to deliver different output or content to a wide variety of users according to set preferences and specific needs. This arms organizations with the ability to provide better value to their customers and prospects. However, dynamic websites suffer from serious vulnerabilities, rendering

⁶⁸ Cross Site Scripting Attack. Web Application Security. Acunetix, Inc. May 26, 2010
<<http://www.acunetix.com/websecurity/cross-site-scripting.htm>>

organizations helpless and prone to cross site scripting attacks on their data. A web page contains both text and HTML markups that are generated by the server and interpreted by the client browser. Web sites that generate only static pages are able to have full control over how the browser interprets these pages. Web sites that generate dynamic pages do not have complete control over how their outputs are interpreted by the client. The heart of the issue is that if mistrusted content can be introduced into a dynamic page, neither the web site nor the client has enough information to recognize that this has happened and take protective actions. XSS allows an attacker to embed malicious JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable dynamic page to fool the user, executing the script on his machine in order to gather data. The use of XSS might compromise private information, manipulate or steal cookies, create requests that can be mistaken for those of a valid user, or execute malicious codes on the end-user systems. The data is usually formatted as a hyperlink containing malicious content and which is distributed over any possible means on the internet.

Buffer Overflows

Buffer overflow exploits constitute the largest single threat to enterprises today. These exploits have the most power, are the easiest to use, and are all too common. No advanced technical knowledge is necessary to run pre-written buffer overflow exploit codes. Buffer overflow exploits are very powerful, and in many cases, the malicious code that executes as a consequence of a buffer overflow will run with administrator-level privileges, and thus can do anything it wants to the server.

Broken Authentication

For example, when a user provides his login name and password to authenticate and prove his identity, the application assigns the user specific privileges to the system, based on the identity established by the supplied credentials. Hijacking- Control of a connection is taken by the attacker after the user authentication has been

established. This kind of attack is not a technological security hole in the Operating System or server software. Rather it depends -on how securely stored and complex the passwords are and on how easy it is for the attacker to reach the server (network security).

Password Guessing

Password guessing can be one of the most efficient techniques to defeat web authentication. This technique can be carried out either manually or via automated procedures. Table 3.4 shows some common usernames and passwords used by attackers in authentication guessing attacks:

Username Guessing	Password Guessing
(NULL)	(NULL)
root, administrator, admin	(NULL), root, administrator, admin, password (company_name)
operator, webmaster, backup	(NULL), operator, webmaster, backup
guest, demo, test, trial	(NULL), guest, demo, test, trial
member, private	(NULL), member, private
(company_name)	(NULL), (company_name), password
(company_name)	(NULL), (known_name)

Table 3.4: Username and Password Guessing (Scambray *et al.*, 2006).

Brute-Force Attack

A Brute Force Attack is the most widely known password cracking method. If password guessing renders no results, the next step for an attacker is to try other password combinations using special custom tools, such as WebCracker which is freely available on the internet. This custom tool attempts to authenticate into the system, making use of predefined lists of usernames and passwords, dictionary attacks, and brute-force attacks. A dictionary attack uses pre-computed wordlists like dictionaries to try to authenticate on the Web applications by trying thousands of combinations of these dictionary words as usernames and passwords.

LM Hash Algorithm

One setback with this encryption scheme is that all characters are converted to uppercase prior to encryption. This in fact removes 26 characters from the set of choices from which a user may possibly select a password, making a dictionary attack, or even a brute-force attack, considerably less work for a cracker. Another weakness of the LM Hash scheme is an even greater one, however, because of the method used to prepare the password for encryption. The number of characters in an LM password is exactly 14, no matter how many characters a user chooses. Perhaps a 14-character password seems like a good one, but this is not the case. Each user password of less than 14 characters is padded with null characters (ASCII zero) to extend its length. The result is then split into two 7 character parts, each of which is encrypted separately. Along with a predictable parity value, the results are hashed, concatenated, and stored.

Password History and Password Aging

Password expiration is not efficient unless users choose different passwords from those previously used. *Password history* is the retention of one or more prior passwords or password hashes for comparison against new passwords or password hashes. A new password is checked to make sure that it has not been used during the specified history. The period is typically defined as either a certain number of prior passwords or a period of time. *Password age* in turn is an attribute directly related to password history. The *minimum password age* is the amount of time that must pass between password changes. As Scarfone and Souppaya (2009) point out, to diminish the effort necessary in remembering passwords, a significant number of users will cycle through passwords after expiration until they have exceeded the password history retention buffer and then change their password back to the original one. Although enforcing a minimum password age does not prevent this, at least it is a restriction.

There are some password history mechanisms that are also capable of identifying passwords that are not satisfactorily different from previous passwords. When forced to choose a new password, the majority of users has a tendency to employ variations of old passwords (e.g., changing “secret05” to “secret06”). This makes it easy for an attacker who knows the old password to guess or crack the new one rapidly. Some existing password history mechanisms can be configured to refuse new passwords that have a certain number of characters in common with previous passwords. Without such a mechanism, it is usually trouble-free for users to append counters to their passwords (e.g. “secret05”). This makes password expiration mostly unproductive, and may in fact cause users to select weaker passwords than they would have without password expiration.

Password history usually only works on a single authentication mechanism and cannot check the history from multiple mechanisms. This enables users to employ the identical password (and prior passwords) on several systems at once. Users frequently do this because it decreases the number of passwords that they have to remember, but this increases the risk to the enterprise by entitling an attacker who compromises one password to reuse it to gain access to additional resources. Additionally, administrators will sometimes reuse passwords between a local user account on a personal workstation and an account that has domain or centralized administrative privileges. This can pose a major risk to the enterprise because the security of centralized password management is generally higher than on individual workstations. An attacker who compromises the workstation and is able to crack the domain administrator password will have significant access to enterprise resources.

There is generally no easy way to detect password reuse across systems, particularly when both internal and external systems are involved. To attempt to reduce the likelihood of password reuse, organizations can have their password management policies prohibit use of the same or closely-related passwords on the organizational IT system and external systems. The password management policy can also explicitly forbid the reuse of centralized (e.g., domain) administrative level

credentials with user or local (e.g., local administrator or root) accounts. Proper user training that stresses the importance of proper password management and protection and explains the risks of password reuse should also be implemented. However, without an enforcement mechanism, it is unlikely that policies against reuse will be significantly effective in reducing reuse, given the number of passwords that users typically need to remember.

Shoulder Surfing

Shoulder surfing is using direct observation techniques, such as looking over someone's shoulder, to get information. Shoulder surfing is an efficient way to get information in packed places because it's quite simple to stand next to someone and watch as they fill out a form, enter a PIN number at an ATM machine, or use a calling card at a public pay phone. Shoulder surfing can also be done long distance with the assistance of binoculars or other vision-enhancing devices. To prevent shoulder surfing, experts advise that you protect paperwork or your keypad from view by using your body or hand.

Phishing

Phishing is the criminally fraudulent process of attempting to obtain sensitive information such as usernames, passwords and credit card details by masquerading as a trustworthy entity in an electronic communication. Phishing is typically carried out by e-mail or instant messaging. A typical example is an e-mail that directs users to visit a website where they are asked to update personal information, such as passwords and credit card, social security, and bank account numbers that the legitimate organization already has. The Web site, however, is forged and set up only to steal the user's information.

Phishing is one of the social engineering techniques employed to trick users, and exploits the poor usability of existing web security technologies.

Hard-Coded Password

Using a hard-coded password greatly increases the possibility of password guessing. The consequences in Authentication are the following: If hard-coded passwords are used, it is almost certain that attackers will gain access through the account in question. They continue to be used to this day, sometimes in high-profile software, despite the significant risk they pose. An example of a hard-coded password in the Java programming language is shown in Figure 3.3. The 8-digit characters Boat6sea! is the hard-coded password.

```
int VerifyAdmin(String password) {  
    if (passwd.Equals("Boat6sea!"))  
    {  
        return(0)  
    }  
    return(1);  
}
```

Figure 3.3: Hard-coded password (Boat6sea!): JAVA code snippet.

Trojan Horse

A Trojan horse is a rogue program that takes the identity of a trusted application to collect information or avoid detection. In a typical Trojan horse attack, the user is presented with a logon screen that appears to be genuine. The user enters their user name and password, and are either logged on, or presented with an error message that she has to type their logon credentials again. Often, the rogue logon application exits after the first request passed the user on to the real logon. Users are easily fooled into thinking that they probably typed the wrong password and must re-enter the information again, never suspecting that their logon credentials are compromised.

Man-In-The-Middle Attack

In a Man-In-The-Middle (MITM) attack, a malicious party intercepts a legitimate communication between two friendly parties. The malicious host then

controls the flow of communication and can eliminate or alter the information sent by one of the original participants without the knowledge of either the original sender or the recipient. In this way, an attacker can fool a victim into disclosing confidential information by “spoofing” the identity of the original sender, who is presumably trusted by the recipient. Here is an example of a MITM attack against a Web-based financial system: A bank demands authentication from the user (i.e., a password, a one-time code from a token, etc.). The attacker sitting in the middle receives the request from the bank and passes it on to the user. The user responds to the attacker, who passes that response to the bank. Now the bank assumes it is talking to the legitimate user, and the attacker is free to send transactions directly to the bank. This kind of attack completely bypasses any two-factor authentication mechanisms, and is becoming a more popular identity-theft tactic.

Broken Authentication

User authentication on the web usually involves the use of a user ID and password. Stronger methods of authentication are commercially available, such as software and hardware based cryptographic tokens or biometrics, but such mechanisms are cost prohibitive for most web applications. A wide array of account and session management flaws can result in the compromise of user or system administration accounts. Development teams frequently underestimate the complexity of designing an authentication and session management scheme that adequately protects credentials in all aspects of the site. Web applications must establish sessions to keep track of the stream of requests from each user. HTTP⁶⁹ does not provide this capability, so web applications must create it themselves. Frequently, the web application environment provides a session capability, but many developers prefer to create their own session tokens. In either case, if the session tokens are not properly

⁶⁹ Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. Its use for retrieving inter-linked resources led to the establishment of the World Wide Web.

protected, an attacker can hijack an active session and assume the identity of a user. Creating a scheme to create strong session tokens and protect them throughout their lifecycle has proven obscure for many developers. Unless all authentication credentials and session identifiers are protected with SSL⁷⁰ at all times and protected against disclosure from other flaws, such as XSS, an attacker can hijack a user's session and assume their identity.

SQL Injection

SQL Injection attacks are very common, and this is due to two factors: the significant prevalence of SQL Injection vulnerabilities and the attractiveness of the target (e.g., the database usually contains all the appealing and critical data for your application). There are many successful SQL Injection attacks that occur, because it is tremendously simple to introduce SQL Injection vulnerabilities in the code. Basically, SQL Injection flaws are introduced when software developers create dynamic database queries that include user supplied input. To avoid SQL injection flaws is simple. Developers need to either - stop writing dynamic queries- and/or prevent user supplied input which contains malicious SQL from affecting the logic of the executed query.

Privilege Escalation

Privilege escalation occurs when a user gains access to more resources or functionality than they are normally allowed, and such elevation/changes should have been prevented by the application. This is typically caused by a flaw in the application. The result is that the application performs actions with more privileges

⁷⁰ Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data – a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with <https:> instead of <http>.

than those intended by the developer or system administrator. The degree of escalation depends on which privileges the attacker is authorized to possess and which privileges can be obtained in a successful exploit. For example, a programming error that allows a user to gain extra privilege after successful authentication limits the degree of escalation, because the user is already authorized to hold some privilege. Usually, we refer to *vertical escalation* when it is possible to access resources granted to more privileged accounts (e.g., an Internet banking account that acquires administrative privileges as an Admin), and to *horizontal escalation* when it is possible to access resources granted to a similarly configured account (e.g., a user accesses information related to a different user on an online banking website).

Real-Time Man-in-the-middle Attacks: Session-Hijacking Trojans

As mentioned in 15.00BA Trojan horse Authentication Asset/Target item in the matrix, this attack installs a type of proxy on the user's computer that interacts with the financial institution's genuine site on the user's behalf. As the Trojan interacts with the financial institution's site through the user's computer, it allows the fraudster to imitate the user's profile. Some Trojans of this type wait until the user logs onto the genuine site and performs a concurrent web session automatically. Thus the Trojan will appear to be transacting from the same IP and device as the user. This type of Trojan circumvents many existing security methods that rely on the compromised computer to communicate. In other words, it uses compromised devices to prompt users to supply challenge questions, one-time passwords, and other types of information that can be used to perpetrate fraud. Session-hijacking Trojans are one of the most difficult forms of attacks to combat.

3.2.2.5 Step 5: Generate the usable security principles

Sub-Step 5.1 Classify and prioritize the cognitive processes generated by the NGOMSL model:

See description in Section 3.2.2.3 Step 3: Develop the NGOMSL Model (Natural Goals, Methods, Selection Language).

Sub-Step 5.2 Classify and develop a cross-cognitive and usable security principles analysis in order to specify the final usable security principles targeted to user authentication.

Our interest in this section is to describe the most relevant correlations between some of the most recognized principles of HCI (Nielsen, 1994) and each of the main categories of authentication methods. This will serve as part of the basis for the construction of the Usable Security Symmetry inspection method.

Password/Passphrases/PINs

Principle	Correlation
Visibility	Users hardly see the password they type, or if they do, the password is hidden under asterisks. The interface cannot provide visual cues, reminders, lists of choices, or other aids; the system cannot display the typed password as it is since it would be an open door for eavesdropping and Social Engineering attacks.
Feedback (Error Handling)	Most systems only mention success or failure. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error. However, if it gives us clues like "the password has to contain letters" we will be exposed to dictionary, eavesdropping, and Social Engineering attacks.
Consistency	This authentication method in general consistently presents - the same layout and terminology (e.g. prompts for passwords, username/password windows). The prompts are well recognized by users (e.g. corporate and techy users) as a security mechanism for accessing network systems. Besides, the association of a typed password with asterisks is very common among users.
Compatibility	This authentication method demands a lot of a user's STM load. The human mind can recognize better than recall, and this authentication method fails to provide enough information to the user so that the user can take action without recalling a lot.
Simplicity	It is simple and straightforward, but some functionalities like "Help" and "Forget Your Password?" are hidden (but can be activated on the links) from the users.

Challenge-Response Calculators (CRC)

Principle	Correlation
Visibility	Users hardly see the password and PIN they type, or if they do-, the password and PIN are hidden under asterisks. The interface cannot provide visual cues, reminders, lists of choices, or other aids; the system cannot display the typed password or PIN as it is since it would be an open door for eavesdropping and Social Engineering attacks. CRC must provide for the ease of use by simple commands, or better, the user's task to authenticate her/himself should become as automatic as possible, increasing its speed, efficiency, and usability.
Feedback (Error Handling)	The system only mentions success or failure related to passwords, PINs, and "challenges". If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.
Consistency	The user has to navigate to multiple screens in order to complete a task. The authentication process is as follows: The user enters a PIN -> and the server send a "challenge" to the user -> The user reads the "challenge" from the token display and enters it on the token keypad -> A <i>response</i> is calculated by the server -> The user reads the response on the display and enters it on the computer terminal -> The responses calculated by the token and by the server are compared, and if they are equal, the user is successfully authenticated. The challenge response security dialog can be very annoying, which can negatively affect the user's experience.
Compatibility	The CRC requires several extra steps (e.g. PIN + challenge + password) and additional memory load for the user, which can be very time consuming. A stronger password including other characters other than digits will demand an even larger CRC keypad, which is not desirable.
Simplicity	A CRC that the user has logged into should be shut off within a given timeframe if not used to prevent illicit users from gaining access while the token is left unattended. However, this can be very annoying for the user, who might have to login with the token several times. Besides, handling an electronic device in a proper manner can be quite a challenge to many users.

Public Key (PK) Authentication

Principle	Correlation
Visibility	The controls of the PK method are not obviously visible and intuitive, and their functions are not recognizable; it is not clear when something has been encrypted and when it has not.
Feedback (Error Handling)	Very few errors are revertible, even if they demand some time and effort to rebuild the intended state. There are numerous irreversible actions in the PK system, such as accidentally deleting the private key, publicizing a key or revoking a key, forgetting the passphrase, and finally, failing to back up the key rings.
Consistency	To encrypt or sign a file, PK presents the user with a status message that indicates it is now "encoding"; it would better to say "encrypting" or "signing", since being able to see terms that unequivocally match the operation being executed helps to create a clear mental model for the user.
Compatibility	The user needs to have a sensorial perception of the PK method in order to allow him/her- to interact with the system in working toward his/her goals, which doesn't occur in this type of authentication method since the PK model is not clear to the user. With this method, users are unlikely to put too much effort into tasks for which they don't understand the need.
Simplicity	The PK system is difficult to learn and use, and displays too much information. A separate certificate is needed for each Certificate Authority for different services and certificate expiration. Using a computer, installing and importing a certificate to other applications might be difficult, as it can only be used at the location where it is installed. To use a smart card, a reader is necessary, and the user must carry the token with her/him. A crucial task such as making a backup revocation certificate is not easy to perform. All these factors are not so convenient from the point of view of the user. Furthermore, the PK model is too demanding for the user as we can see in the authentication process as follows: 1. the user supplies her/his certificate with the Certificate Authority, a signature, and the PK to the authentication system; 2. the user proves possession of the secret private key (PRK)- by presenting the file or inserting the smartcard into a reader; 3. the authentication system checks if the PK in the certificate corresponds to the private secret key. If they correspond, the user has proven possession of the secret key and is successfully authenticated.

Biometrics

Principle	Correlation
Visibility	Generally speaking, registration, identification, and authentication processes are very time consuming ⁷¹ .
Feedback (Error Handling)	Most systems only mention success or failure, and the user might be erroneously rejected.
Consistency	In fingerprint recognition, exact placement of the fingertip on the scanning surface is very significant for reliable performance (there is elastic distortion from one sample to the next) and can be difficult for the user. With face recognition, the concept of recognition itself is well known by the users, generally resulting in high user acceptance.
Compatibility	Much training and education is necessary to prevent false rejects. Many biometrics cannot handle variations to the users' characteristics (e.g. aging, illness, or injury).
Simplicity	With the voice recognition method, for example, it might have to give numerous live samples.

3.2.2.6 Step 6: Formulate the Usable Security Symmetry (USS)

Identify the usability factors and usability criteria (Section 5.2 Usability Factors and Usability Criteria);

Define user authentication use cases (5.3.1 User Authentication Use Cases);

Specify project lead and development activities, (5.4.2 Usable Security Protocol (USP) Sub-Methodology);

Specify usability severity ratings (5.4.2.2.1 Usability Severity Ratings);

Specify security severity ratings (5.4.2.2.2 Security Severity Ratings).

3.2.2.7 Step 7: Demonstrate the Usable Security Symmetry (USS)

See Section 5.5: The Demonstrational Approach.

⁷¹ Alan E. Zuckerman, M.D., Kenneth A. Moon, M.D., and Kenneth Eaddy: 2007. Comparison of Fingerprint and Iris Biometric Authentication for Control of Digital Signatures. May 27, 2010 <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2244356/pdf/procamiasymp00001-1249.pdf>>

3.2.3 The Usable Security Protocol Methodology Reuse

The current push toward shortening product (and design) development times and the urge to *go-to-market* have led customers to adopt new strategies such as reusing to reduce waste and streamlining the product development process.

Reusability of design has been practiced by the vast majority of companies, but mostly at the high granularity level (Figure 3.4). This thesis's author agree with Gautam *et al.* (2007) that "Reuse is not only the process of using physical components or associated designs but also the process of implementing or updating product information using existing 'assets'. Assets can be specifications, designs, user documentation, test plans, test results, etc." The scope for reuse is in fact frequently limited to the physical components (i.e., reuse of hardware components).

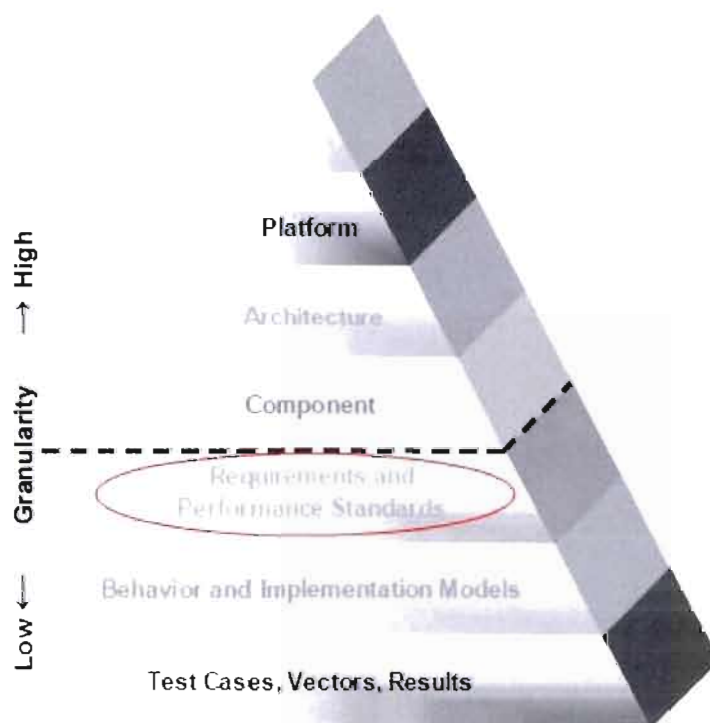


Figure 3.4: Reusability Pyramid (Gautam *et al.* 2007).

Reuse can and should also be implemented at a low granularity level to speed up the product development process. This is where USP methodology reuse can be

applied in the Reusability Pyramid, in the *Requirements and Performance Standards* layer as shown in Figure 3.4 (red oval).

The USP methodology allows users to reuse its framework by essentially specifying their own parameters and adapting them to their own products. It uses a summarized USS Quick Setup process and a set of Design Artifacts which can be all seen in Appendix D: Usable Security Protocol (USP) Reuse Methodology. Additionally, the Usability Factors and Usability Criteria Mapping mentioned in Chapter 6 can be re-used independently of the USS framework to study the relationship between other usability factors, for instance *Usability* and *Portability* (i.e. wireless devices): *Usability* is the extent to which the software is easy to learn and operate; *Portability* is the ease with which the software can be transferred to new operating environments, hardware platforms, and operating systems. For more information, see Appendix D, section D.2 Design Artifacts for Reuse, Design Artifact 1: Specifying the usability factors and usability criteria.

Summary of the topics discussed in Chapter 3: The Usable Security Protocol.

In concluding Chapter 3, the USP methodology has been laid out by presenting the following steps: Step 1: Define the mission and conceptual design objective, Step 2: Identify the most representative user authentication methods categories, Step 3: Develop the NGOMSL Model (Natural Goals, Methods, Selection Language), Step 4: Develop the Authentication Risk Assessment Matrix, Step 5: Generate the usable security principles, Step 6: Formulate the Usable Security Symmetry (USS) inspection method, and Step 7: Demonstrate the USS. Also, this chapter discussed - how to reuse the USP methodology design artifacts by essentially specifying your own parameters and adapting them to your own product.

CHAPTER IV

THE COGNITIVE SCIENCE AXIS

4.1 Introduction

This chapter describes the Cognitive Axis of this thesis by identifying and explaining the main cognitive areas of focus relating to user authentication (e.g. perception, attention and memory, etc.). Also it specifies a Cognitive Model of User Authentication (CMUA) in order to understand how and what cognitive processes (and flow) are involved for each of the main categories of user authentication methods. It serves as the basis for the development of the USS inspection method.

The Cognitive Axis is defined as one part of the two-part vital holistic approach in conjunction with the Computer Science Axis to undertake the usable security of user authentication. It is also the intersection point of perceptual and cognitive processes.

Cognitive analysis for HCI lends itself to two related interpretations: 1) the analysis of cognition-intensive interactions with computers, such as learning, problem-solving, or reading and 2) the analysis of cognitive content, structures, and processes involved in any interaction with a computer. This dissertation addresses both interpretations by providing methods for analyzing cognition with a focus on interactions specifically related to user authentication and that involve cognitive processes such as Perception, Memory (LTM, STM, Visual Recognition Memory), Information Retrieval (Recall and Recognition), and Mental Models. In addition, the analysis of users' cognition should not be restricted to an early design phase (as task analysis typically is), but should be an important activity throughout the entire design process. However, many designers have little training in the methods used to measure cognition.

Security systems must be viewed as socio-technical systems that depend on the social context in which they are embedded to function correctly. Security systems will only be able to provide the intended protection when people actually understand and are able to use them correctly. There are very real differences between the degree to which systems can be considered theoretically secure (assuming they are correctly operated) and actually secure (acknowledging that often they will be operated incorrectly) (Jøsang *et al.*, 2007). In many cases, there is a trade-off between usability and theoretical security. It can be meaningful to reduce the level of theoretical security to improve the overall level of actual security. For example, the strongest passwords, from a theoretical perspective, are randomly generated. However, since it is very difficult to remember such passwords, people will write them down, and thereby undermine the system's security. Thus, it may be meaningful to allow people to choose passwords that are easier to remember. Although this reduces the theoretical strength of the passwords, it increases the security of the system as a whole.

It is important to step back and remind ourselves about the concept of cognition. The term cognition (Latin *cognoscere*, "to know" or "to recognize") is a concept used in different ways by different disciplines, but is generally accepted to mean the process of thought. It refers to a faculty for the processing of information, applying knowledge, and changing preferences. Within psychology or philosophy, the concept of cognition is closely related to abstract concepts such as the mind, reasoning, perception, intelligence, learning, and many others that describe capabilities of the mind and expected properties of an artificial "mind". Cognition is considered an abstract property of advanced living organisms and is studied as a direct property of a brain (or of an abstract mind) on - the factual and symbolic levels. For example, in psychology and cognitive science it refers to an information processing view of an individual's psychological functions.

Cognitive Informatics (CI) is an emerging discipline that studies the natural intelligence and internal information processing mechanisms of the brain, as well as

the processes involved in perception and cognition. CI provides a coherent set of fundamental theories, in conjunction with contemporary mathematics, which form the foundation for most information and for knowledge-based science and engineering disciplines such as computer science, cognitive science, neuropsychology, systems science, cybernetics, software engineering, and knowledge engineering.

In psychology and in artificial intelligence, cognition is used to refer to the mental functions, mental processes (thoughts), and states of intelligent entities (humans, human organizations, highly autonomous machines). In particular, the field focuses on the study of specific mental processes such as comprehension, inferencing, decision-making, planning, and learning.

The advent of (and constantly evolving) Information Technology (IT) has placed heavy cognitive demands on workers under normal conditions. These demands are amplified significantly when workarounds are needed, when problems occur, and when time is short. Howell and Cooke (1989) found that advances in technology and machine intelligence had in fact augmented, not lowered, the cognitive demands on humans. Basically what has remained for humans are the complex aspects of works such as tasks demanding judgment, assessment, diagnostic power, decision making, and the ability to plan and anticipate. Also the complexity of the work can boost cognitive demands such as the number of different factors to track, their diversity, and their level of interaction. Workers struggle to figure out how things interact and how outputs are produced from inputs. Braz *et al.* (2007) have demonstrated that users have to manage complexity when authenticating to an MTM with a cell phone. They are equipped with a special chip that is able to communicate with the MTM. In addition, users will still be required to authenticate to the system by entering a PIN. Basically, the scenario here involves a user making her monthly mortgage payment. In this case, the user has to deal with different services offered through different types of communication channels such as MTM, the Web, and Wireless Networks. Although it might be considered a convenient service when one does not have physical access -to

an MTM, it does place the burden on the users with regards to the coordination of the MTM with the cell phone.

User authentication systems are ultimately used by people, so their ease of use, understandability, satisfaction, and their implicit cognitive dimensions must be addressed as well. The cognitive dimensions can be essentially considered as the interaction occurring between users and security mechanisms (e.g. logging into a system, interacting with an authentication token or smart card, etc.). When a user performs a task, that is, activities that are undertaken to achieve a goal (e.g. a user logs on to *ieee.com* using her/his BlackBerry to access her/his myIEEE account), some of these activities can be considered *physical* (e.g. a user enters a password on the BlackBerry's keyboard), while others can be considered *cognitive* ones (e.g. a user retrieves a password stored in her/his memory). In particular, *interface* is viewed by this thesis's author either as a software component (e.g. login Web page) or as a hardware component (e.g. an authentication token, a smartphone, etc.) through which the interaction/information travels between the interface and the user (Maffezzini, 2006).

Most usability inspection techniques do not overtly take into account users' thinking, "even though psychology-based inspection techniques supplied key insights into how thinking shapes interaction" (Hornbæk and Frøkjær, 2004). Evidence shows that the well known Knowledge-Based Authentication does not take into account how people think (Adam and Sasse, 1999). Also, empirical research (Zurko and Simon, 1996; Whitten and Tygar, 1998; Chiasson and Biddle, 2007) has shown that cognitive dimensions definitely influence the usability of security mechanisms under which user authentication methods are included. Researchers argue that security concepts used in security mechanisms are not easily understood by many users. Hence security designers should place additional effort into understanding the users' mental model and be certain to employ concepts the users can recognize. For example, in a typical authentication task, Alice tries to log into a corporate computer system with a

user ID and password. The activities that are undertaken to achieve this goal can be considered physical (e.g. Alice types in a password on a desktop keyboard)- or cognitive - (e.g. Alice retrieves a strong password such as Gyz!l52# stored in her memory which often results in a huge demand on her memory). A strong password must be enforced, given that it makes the attacker's job much harder in guessing predictable passwords. This is not an easy task given that the cognitive capacity of a user to remember a password is quite limited (Sasse *et al.*, 2001). But what is a strong password policy? It must contain at least eight characters, one uppercase alphabet (A-Z), one lowercase alphabet (a-z), one Arabic numeral (0-9), one non-alphanumeric character excluding “ @ ~”. At this moment the system has blocked her account due to three unsuccessful attempts to log into the system. An authentication system should *a priori* promote strong passwords which account for security while still preserving memorizability, which in turn accounts for usability. Another example is the poorly understood, overly complex, and hard to use (for end users) Public Key authentication method according to usability evaluations (Whitten and Tygar, 1999; Williams and Voigt, 2004). For example, when Pretty Good Privacy (PGP) is in the process of encrypting or signing a file, it gives the user a status message indicating that it is presently “encoding.” However, a better term would be “encrypting” or “signing”, given that employing terms that overtly match the operations being executed helps to create an understandable mental model for the user.

These facets of understanding how users cope (or not) with different types of user authentication methods explain our interest in studying its cognitive dimensions in order to give a cognitive ergonomics account of user authentication design using Cognitive Task Analysis (CTA) (Hollnagel, 2003). CTA describes the physical tasks and cognitive plans required of a user to accomplish a particular work goal. The GOMS model was the method chosen to perform the CTA for this dissertation. The CTA and GOMS model are explained in detail- in Section 2.3.2 GOMS: A Method for Cognitive Task Analysis. The GOMS CTA method provides the capability of

identifying, describing, and detailing the cognitive thought processes involved in the preparation and successful execution of user authentication procedures.

As stated previously, our objective is to develop a usable security protocol for user authentication, also taking into consideration their respective cognitive dimensions. Therefore, a cognitive model for user authentication has been developed as described in Section 4.4: The Cognitive Model of User Authentication (CMUA).

There are three reasons why two basic sections of this thesis, the state-of-the-art of cognitive processes and the Cognitive Model of User Authentication (CMUA), have been brought together here in Chapter 5: i) background knowledge is provided first, followed by its related implementation. For instance, the background of a cognitive process like memory (e.g. definition, types of memory, etc.) is located right before its implementation (function) within the CMUA. ii) the Cognitive Computing doctoral program explicitly requires to separate the Cognitive axis from the Computer Science axis. iii) to facilitate the flow of thinking, reading, and –accessing related information (see example in i) above).

4.2 Cognitive Ergonomics

HCI involves systems comprised of people, computers, and their interactions. Cognitive Ergonomics (CE), however, is concerned with the mental aspects of the interaction, that is, the analysis of cognitive processes such as perception, memory, reasoning, and motor response required of operators in modern industries. CE is also concerned with developing specifications of the knowledge required by the human to interact with the computer to perform work effectively. These specifications are implementable as an interaction. It places particular emphasis on the analysis of cognitive processes (e.g., diagnosis, decision making and planning) required of operators in modern industries. CE aims to enhance the performance of cognitive tasks through - several interventions, including:

- user-centered design of human-machine interaction and human-computer interaction (HCI),
- design of information technology systems that support cognitive tasks (e.g., cognitive artifacts),
- development of training programs, and
- work redesign to manage cognitive workload and increase human reliability.

CE also studies the competencies and limitations of workers in their interactions with the work system (e.g. errors, strategies, cognitive workload), in particular with the cognitive artifacts they use to achieve their goals as well as with the co-operation with other actors. CE is mostly important in the design of complex (e.g. computer security applications), high-tech, or automated systems.

A major factor when designing security applications that must be taken into consideration by designers is that for the vast majority of users, security is an “enabling task” to one or more “production tasks” (e.g. access a database, shop online, etc.). Such an “enabling task” is perceived as an obstacle. In addition to that, cognitive demands required by authentication tasks are becoming increasingly complex. To reduce management and support costs, organizations are placing more and more of the burden of authentication on the user, forcing them to perform - at the enterprise’s discretion - lifecycle-management tasks such as token requests and activation, password replacement, certificate renewal, etc.

The cognitive demands required by an assessment item are related to the number and strength of connections of concepts and procedures that a user needs to make in order to generate a response, in this particular thesis, when authenticating to a system (the assessment item). The cognitive processes are typically comprised of recall and recognition (e.g. face recognition authentication) and identification and classification (e.g. KBA such as SiteKey (BankofAmerica, 2009): first you recognize a unique image you chose and image title you created to accompany your image, then

you mentally group image and title carrying out in this way collection and comparison).

Traditionally CE has used the “human information-processing” model of cognition (Wickens, 1992), which models human cognition through a computer metaphor.

Central in CE is the notion of *domain*: Domain is the larger environment in which the worksystem must operate, and presents both constraints and opportunities for the worksystem. The domain influences the approach followed, as the degree of coupling among its constituents, the level of top-down causality, and the degree of human intentionality in decision making shapes the validity of the models used (Dowell and Long, 1998) (Figure 4.1).

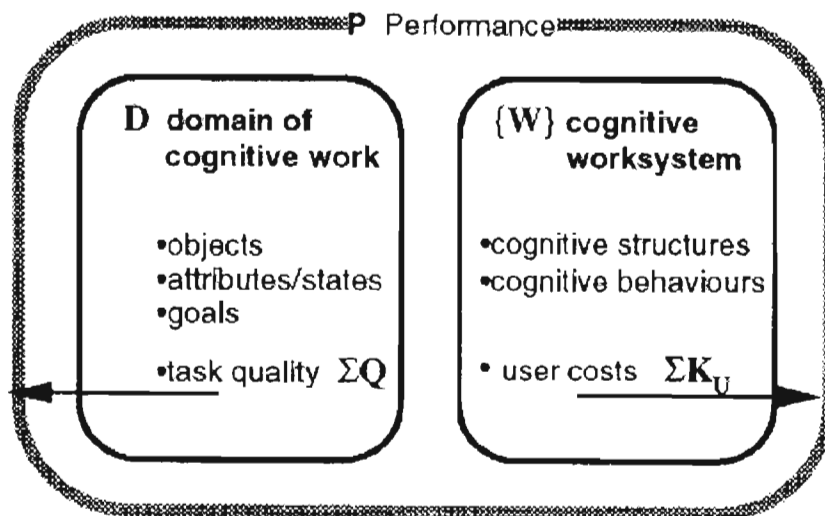


Figure 4.1: Worksystem and a domain (Dowell and Long, 1998).

CE also studies the competencies and limitations of the workers in their interactions with the worksystem in general (e.g. attention, perception errors, strategies, cognitive workload), and in particular the cognitive artifacts they use to achieve their goals as well as their co-operation with other actors. The worksystem is

the system of agents interacting with each other to perform work by intentionally changing the states of domain objects.

CE is especially important in the design of complex, high-tech, or automated systems. A poorly designed cellular phone user-interface may not cause an accident, but it may well cause great frustration on the part of the consumer and result in a marketplace-driven business failure. A poor interface design on industrial automated equipment, however, may result in decreased production and quality, or even a life-threatening accident.

4.2.1 Methods

The methods used by CE are those whose product yields specifications of knowledge. That is, the representations and processes required to support the user's behavior, such that, in interacting, the user and the computer achieve the desired performance. A wide range of methods are used in CE, but the most common methods are the Hierarchical Task Analysis (Kirwan *et al.*, 1992) and Cognitive Task Analysis (Hollnagel, 2003; Kieras, 1996). Section 2.3.4 Natural GOMS Language (NGOMSL) describes the adopted GOMS model (Kieras, 1996) as the basis for the development of the CTA.

4.2.2 The Cognitive Approach

The Cognitive approach permits the analyst to gather information and understand operation up to the thought process level. It allows a deeper understanding of the business problems or needs. This thorough understating can then be translated into better decision making. Overall, the Cognitive approach is comprised of the rigorous practice of gathering information, human information processing, analysis, business modeling, and simulation.

4.2.2.1 Gathering information

Instead of relying only on meetings, surveys, or internal documentation, information is gathered in the field with “thinking out loud” techniques while people are performing their tasks. This ensures a deeper understanding of the current situation. Even if a process is totally changed, gathering information with the cognitive approach exceeds the risks of not doing so.

4.2.2.2 Information processing

To understand the thinking process, goals, and knowledge, a CTA is performed which is in this thesis undertaken by the GOMS model. Cognitive goals, sub-goals, and methods are then described hierarchically. Methods are extracted with “how” questions, and goals are extracted with “why” questions. At the end of the process, management will have a deep understanding of the operations, problems, and strategies. This ensures an effective way to optimize any process. CTA serves also as input for defining the requirements of an information system.

4.2.2.3 Process Modeling

The business is modeled as a hierarchy of systems and processes. The highest level is the mission, followed by generic functions, specific functions, and ultimately, at the most detailed level, the structural elements. The gathering techniques along with the CTA ensure that the business model will be grounded in reality. This provides a complete and exact picture of the operation to management.

4.2.2.4 Simulation

Before executing a plan, each risk is analyzed and addressed by simulation and calculation. For example, in Information Technology (IT), user acceptance is often the prime risk. Simulating the user interface prior to writing any line of code ensures that the both the user's and the business's needs are met first.

The application of the Cognitive Approach helps an organization learn and translate that learning into rapid action, which is the vital competitive advantage.

4.3 Main Cognitive Areas of Focus Relating to User Authentication

This section addresses the main cognitive processes specifically related to Human Computer Interaction (HCI) and user authentication. As already stressed in Section 5.1 Introduction, the cognitive dimensions are considered as the interaction occurring between users, security (authentication) mechanisms, and Web/Mobile/Wireless components.

4.3.1 Perception

Perception is our awareness and understanding of the elements and objects of our environment through the physical sensation of our various senses, including sight, sound, smell, and so forth. Each sense organ is part of a sensory system which receives sensory inputs and transmits sensory information to the brain. Perception is influenced in part by experience. We classify stimuli based on models stored in our memories, and in this way achieve understanding. Basically, we tend to match objects or sensations perceived to things we already know. Other perceptual characteristics that are relevant to the user authentication subject matter include:

- Proximity: Our eyes and mind see objects as belonging together if they are near each other in space.

- Similarity: Our eyes and mind see objects as belonging together if they share a common visual property, such as color, size, shape, brightness, or orientation.
- Matching patterns: We respond in the same way to the same shape in different sizes. For example, the letters of the alphabet have the same meaning, regardless of physical size.

According to Dowell and Long (1998), the user's cognitive behaviors are the processing of representations. So, perception is a process whereby a representation of the domain, often mediated by tools, is created. Neisser (1964) emphasizes that human experience depends on the stored mental schema, which guide exploring behavior and the perception of external contexts.

There are numerous theoretical accounts of perception, which can, in general, be divided into two groups: Bottom-up processing and Top-down processing. Bottom-up processing is also known as data-driven processing, because perception begins with the stimulus itself. Processing is carried out in one direction from the retina to the visual cortex, with each successive stage in the visual pathway carrying out ever more complex analyses of the input. Top-down processing refers to the use of contextual information in pattern recognition. For example, understanding difficult handwriting is easier when reading complete sentences than when reading single, isolated words. This is because the meaning of the surrounding words provides a context to aid understanding.

4.3.2 Memory

Memory is just one of many phenomena that show the brain's complexity⁷². On a basic level, memory is the capacity for storing and retrieving information, but memories are not simply recorded and neatly stored. Our memories are selected,

⁷²As already pointed out, one of the main goals of this thesis was to identify the main user authentication task scenarios. Thus the employed use cases for all tasks scenarios did not consider all possible and alternative memory related scenarios such as changing a password, resetting a password, losing a password, etc. due to the fact that they are outside the scope of this thesis.

constructed, and edited not just by us but by the world around us. We have an amazing, unlimited capacity for memory, but our memories are also faulty, full of holes and distortions, and vulnerable due to unreliable data retrieval systems.

Memory can store, identify, and classify- detailed sensory images, facts about the world, tasks mechanics, and experiences. Three processes are involved in memory: *encoding*, *storage*, and *retrieval*. All three of these processes determine whether something is remembered or forgotten.

4.3.2.1 Encoding

Processing information into memory is called *encoding*. People automatically encode some types of information without being aware of it. For example, most people probably can recall where they ate lunch yesterday, even though they didn't try to remember this information. However, other types of information become encoded only if people pay attention to it. College students will probably not remember all the material in their textbooks unless they pay close attention while they're reading. There are several different ways of encoding verbal information:

- Structural encoding focuses on what words look like. For instance, one might note whether words are long or short, in uppercase or lowercase, or handwritten or typed (e.g., a strong password `Blitz4three$`).
- Phonemic encoding focuses on how words sound.
- Semantic encoding focuses on the meaning of words. This requires a deeper level of processing than structural or phonemic encoding and usually results in better memory.

4.3.2.2 Storage

After information enters the brain, it has to be stored or maintained. To describe the process of storage, many psychologists use the three-stage model of memory

proposed by Atkinson and Shiffrin (1968) as shown in Figure 4.2. According to this model, information is stored sequentially in three memory systems: Sensory Memory, STM, and LTM.

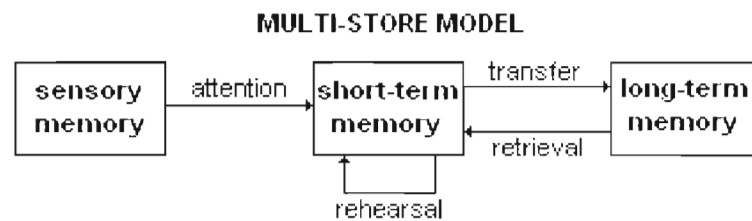


Figure 4.2: The three-stage processing model of memory
(Atkinson and Shiffrin, 1968).

The memory system can be classified into two main types of memory: - LTM and STM, as shown in Figure 4.3. The LTM is divided into Declarative and Procedural memories. The Declarative memory is in turn sub-divided into Semantic and Episodic memories, followed by the Procedural Memory, which is sub-divided into Priming and Procedural Memory.

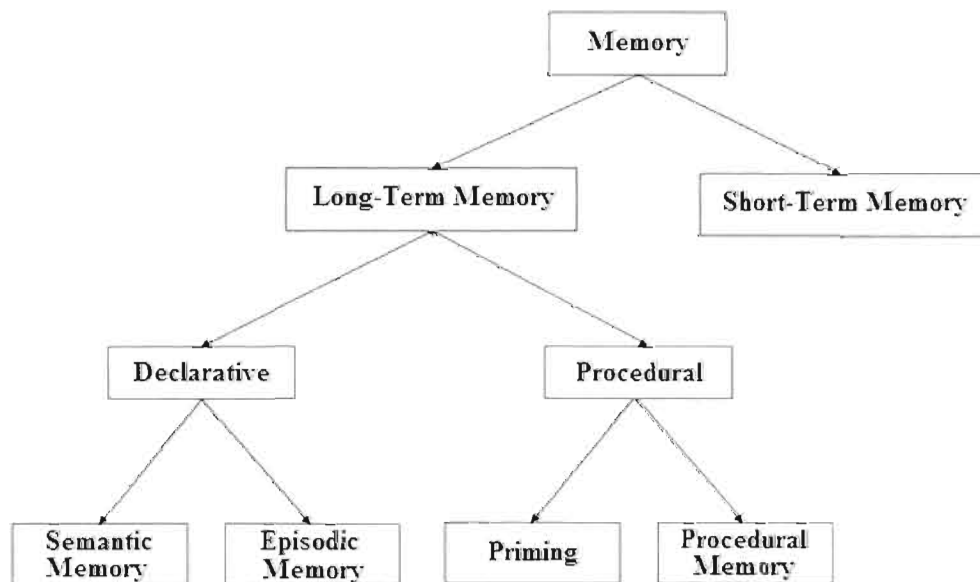


Figure 4.3: Classification of memory.

A detailed explanation of each type of memory is given in the next paragraphs.

4.3.2.2.1 Sensory Memory

Sensory memory, or sensory register, notes or registers sensory stimuli as they are experienced. It consists of representations of the outside world as experienced through the senses such as touch, sight, or smell. It holds information for approximately one to two seconds. If, for instance, you glance at the ocean and turn away, the image of the ocean will be lost in one to two seconds unless the image is quickly transferred into the STM system. The contents of sensory memory are constantly changing as new stimuli are perceived. Information that does not fade from sensory memory enters STM.

4.3.2.2.2 Working Memory (or Short-Term Memory)

Working Memory (WM) is thought to process information by actively repeating, grouping, and summarizing it to aid its storage in LTM. Information is thought to last within STM for only a short period of time before it is either passed into LTM or discarded. For information to be transferred into LTM, it must be rehearsed or repeated. WM is generally considered to have limited capacity. The earliest quantification of the capacity limit related with WM was the 7 ± 2 (i.e. the magical Number Seven, Plus or Minus Two) “chunks” of information rule introduced by Miller (1956). He noticed that the memory span of young adults was around seven elements, called chunks, regardless of whether the elements were digits, letters, words, or other units. The 7 ± 2 is that the capacity of STM is seven plus or minus two pieces of information (some people can hold five or six items, while others can hold eight or nine).

However, in a more recent study about the mental storage capacity of the STM, Cowan (2001) noted a number of other limits of cognition that relates to a "magical

number four". He argues that Miller's number was in fact meant more "as a rough estimate and a rhetorical device than as a real capacity limit. Others have since suggested that there is a more precise capacity limit, but that it is only three to five chunks." In his study Cowan put together a broad range of data on capacity limits, asserting that the "smaller capacity limit is real".

In general, WM can hold five to nine units of information for between twenty seconds to one minute in length. It holds information for as long as it is actively thought about, or until new information basically forces it out. Unless we repeat the information and purposely try to retain it, most (or all) of it will be lost. A good example of this process can be seen when you look up a new phone number and repeat it to yourself as you dial it. After dialing it, within a few seconds you will usually forget it. However, if you do this repeatedly (repetition or rehearsal), as in the case of a friend with a new phone number, it will in the end up entering LTM. These "units" of information can represent single pieces of information, such as an individual's name, or the units can be single pieces of information that represent a number of different pieces of information, as in the last name of a family representing all of the family's members. The process of using a single item to represent a number of items is called chunking, and researchers have found that WM's information holding capacity can be significantly improved with this process.

It seems there are many factors that determine what information enters LTM, two of the strongest being repetition and intense emotion. If something is repeated often enough, such as multiplication tables, it will enter LTM. And it is hard to forget intensely emotional experiences, such as being involved in a serious car accident or falling in love.

4.3.2.2.3 Long-Term Memory (LTM)

LTM has been the focus of most research and theory on the memory system. It holds all the information that has managed to pass through the sensory and STM

systems. In contrast to both of those systems, LTM is thought to be able to hold potentially unlimited amounts of information for an indefinite period of time, possibly for a lifetime. There is a structure for storing a representation of the knowledge we accumulate over time in the LTM. It is thought to hold all of the memories of our life, as well as our knowledge of the world in general. Information entering the LTM is assumed to be permanent. In LTM, one might find memories as diverse as the first person you ever had a crush on, knowledge of how to ride a bike or cook scrambled eggs, or a second language. It is also where mental models are stored. Whereas STM generally holds between five and nine items, scientists say there are no limits on the capacity of LTM given that people have associations for those memories. That is why, for example, people have a natural inclination to choose passwords based on familiar things such as children's birthdays or favorite sports team rather than incomprehensible strings like 3B#\$1r or 7*\$3fg.

LTM stores and operates quite differently depending on the type of information involved. - One of the most influential theoretical divisions of LTM is the division between Declarative Memory (episodic memory and semantic memory) and Procedural memory as described below:

- Declarative memory (DM) is recall of factual information such as dates, words, faces, events, and concepts. It is so called because it refers to memories that can be consciously discussed, or declared. It applies to standard textbook learning and knowledge, as well as remembering the capital of Germany, the rules for playing ice hockey, and what happened in the last game of the Montreal Canadiens, for instance, as each of these involves declarative memory. DM is often considered to be explicit because it involves conscious, intentional remembering. It is subject to forgetting, but regularly accessed memories can last indefinitely. DMs are best established by using active recall combined with mnemonic techniques and spaced repetition. A mnemonic device is a memory and/or learning aid. Mnemonics are frequently verbal,

something such as a very short poem or a special word used to help a person remember something, particularly a list, but they may also be visual, kinesthetic, or auditory (Tulving and Schacter, 1990). Declarative memory is divided in two types:

- Semantic and Episodic. *Semantic Memory* is the recall of general facts, while *Episodic Memory* is the recall of personal facts. Remembering the capital of Germany and the rules for playing football uses semantic memory; remembering what happened in the last game of the Alouettes uses episodic memory.
- Episodic memory is the conscious recollection or recall of specific experiences from a person's life. These memories often include the time and place of the experience, as well as a representation of the role the individual who is remembering played in it. *Episodic Memories* seem to be more affected by the passage of time than are procedural or semantic memories, such that if the event is not recalled and thought of relatively often, details of the event, if not the event itself, seem to fade or be forgotten over time. Two specific types of mental representations hypothesized to be used by the semantic memory system to organize information are schemas (Anderson, 1977) and categories. Schemas are ordered frameworks or outlines of world knowledge that help us organize and interpret new information. They are like maps or blueprints into which new related information will be fitted. Knowledge of your home town or city, with its streets, various buildings, and neighborhoods is an example of a schema. Schemas also help people to reconstruct, or try to remember, information that may have been forgotten. For example, if a friend brings up a time when you both went out to eat dinner a few months ago and you don't remember it clearly, you might ask for more information, and then use

your schema for the usual sequence of events in eating out to try to remember or reconstruct what happened. Categories are another representational form of thought used by semantic memory to organize information. Categories are sets of objects, experiences, or ideas that are grouped together because they are similar to one another in some respect. For example, apartments, houses, huts, and igloos, might be grouped under the category of dwellings. Like schemas, categories help us make sense of, and organize, the countless aspects of the world.

- Typically, recognition memory has been defined as the ability to assess accurately that a stimulus has been encountered before. There can also be discrimination components in which the learner may be able to distinguish between a stimulus that had been previously presented and a new stimulus, without any further knowledge of either one.

Visual Recognition Memory is the ability to recognize elements in the surrounding environment, such as faces or places, as well as the ability to learn about and orient ourselves within that environment, both of which are crucial to our functioning in the world. We need to recognize individuals, such as family members, and to be able to navigate from one place to another. Thus neural systems have evolved to interpret incoming sensory information, with neurons that are capable of distinguishing novel and familiar visual elements. Systems based on recognition of visual items for authentication have been receiving much attention lately. For example, both *Déjà Vu* (Dhamija and Perrig, 2000) and *Passfaces™* (Passfaces, 2009) present users with panels of images, from which they have to recognize and select their pass images. The most significant difference between the two systems involves the content of the images: *Déjà Vu* employs randomly generated art, while *Passfaces* uses photographs of strangers' faces in

an attempt to exploit people's ability to process and remember faces. These systems have performed well in laboratory-style tests, producing recall rates of up to 80% even after up to 3 months of non-use (Valentine, 1998).

- Procedural Memory (PM) pertains to the storage of skills and procedures. This type of memory has also been referred to as "tacit knowledge" or "implicit knowledge". PM is involved in tasks such as remembering how to play squash or how to ride a bike. This is "know how" memory; it often can only be expressed by performing the specific skill, and people have problems verbalizing what they are doing and why. Procedural memory is therefore very important in human motor performance. An important feature of procedural memory is that it tends to persist; it's resistant to change -can be useful since you don't want to have to keep re-learning behaviors. But this also means that you can't change a procedure, unless and until you pay attention to how and when it operates. An interesting characteristic of PM is that procedural patterns take a while to unlearn. Consider this scenario: You like to play tennis and have played for years. You decide to take some lessons. The instructor shows you how to swing the racquet more effectively. But you soon discover that you just can't - tell yourself to swing it differently. The old pathways interfere with the new ones. It's hard to interrupt a well-established procedure. In fact, those original neural pathways, though weakened, will always be there, for we currently have no reason to think that they will deteriorate. Under conditions resembling the initial circumstances in which they were laid down, they may even be reactivated. However, the new regulated pathways will eventually override the old ones.
- Priming refers to an increased sensitivity to certain stimuli due to prior experience. Because priming is believed to occur outside of conscious awareness, it is different from memory that relies on the direct retrieval of

information. Direct retrieval utilizes explicit memory, while priming relies on implicit memory. Priming can be conceptual or perceptual. *Conceptual priming* occurs where related ideas are used to prime the response, and is enhanced by semantic tasks. For example, *table* will show priming effects on *chair*, because *table* and *chair* belong to the same category. *Perceptual priming* is based on the form of the stimulus and is enhanced by the match between the early and later stimuli, for example, where a partial picture is completed based on a picture seen earlier.

4.3.2.3 Information Retrieval

There are two types of information retrieval: recall and recognition⁷³. In recognition, the presentation of the information provides the knowledge that the information has been seen before. Recognition is of lesser complexity, as the information is provided as a cue. However, the recall can be assisted by the provision of retrieval cues, which enable the subject to quickly access the information in memory.

4.3.2.3.1 Recall

One of the most critical HCI principles is to avoid unaided recall wherever possible, since it is known to place a considerable burden on users' cognitive load and overall ability to perform. There are authentication mechanisms that use cued recall and recognition, for example:

⁷³ As already mentioned, one of the goals of this thesis was to identify the main user authentication task scenarios. They have been -built only taking into consideration its main use cases, which- in the case of Passwords/PINs scenarios, is "a user successfully logs into a system." So this use case considered a strong password as the standard example of passwords (e.g., "Boat6paper!" as shown in Sub-Step 3.1 Username and Password Login {wired network-based task}). Also, this use case did not consider all possible and alternative recall and/or recognition scenarios such as passwords that have been changed, passphrases hints, and others due to the fact that they are out of the scope of this thesis.

- SiteKey (BankofAmerica, 2009), as already described in Chapter 2, is a web-based security system that provides one type of mutual authentication between end users and websites. Its primary purpose is to deter phishing.
- *Cognitive passwords* involve a series of questions (e.g. security questions) about the user's personal preferences and history: After a certain number of correct answers, the user is considered to have passed authentication.
- *Associative passwords* employ word pair or phrase associations in a similar manner (e.g. Dear-God, Spring-Step), while avoiding word association stereotypes.
- The *pass sentence* mechanism is an unaided recall mechanism in the first place. However, if the user does not get the secret completely right, the user is prompted with questions about the *pass sentence*, and when the user answers enough questions correctly, login is allowed.

4.3.2.3.2 Recognition

Recognition is one of the three basic memory tasks. It involves identifying objects or events that have been encountered before. Recognition (re+cognition) is a process that occurs in thinking when some event, process, pattern, or object recurs; it involves knowing or feeling that someone or something present has been encountered before. Coming from the base cognition, recognition has various uses in different fields of study, and has generally been accepted as referring to- the process of awareness or thought.

In psychology, cognition is used for information processing view of a person's psychological functions. This takes place as we process stimuli in relation to previous memories and experiences; also we make connections between the current stimuli and our memories. Thus, in order for something to be recognized, it must be familiar. This recurrence allows the recognizer to more properly react. Hence recognition is a survival mechanism. Humans and animals will recognize certain foods, which are

poisonous through taste, as they have tasted them before. This works also for sounds and alarms, which we are trained to react to, such as fire alarms.

Without Recognition, we would go through life reliving everything without learning from the past. Experiences would be pointless, as they would not be remembered. Recognition is the easiest of the memory tasks. That is why multiple-choice tests are often considered easier than other tests. In multiple-choice tests, you only need to recognize the right answer. You do not have to come up with the answer on your own. Recognition uses the memories we have in place to help with the current situation. When the recognizer has correctly responded, this is a measure of understanding.

4.3.2.4 Password Memorability Issues

The user characteristic that has the major impact on password design is memorability. The Password authentication mechanism is in fact a huge component of the study of usable security of user authentication. As a Knowledge-Based Authentication (KBA) mechanism, it requires users to memorize items and recall them when accessing a specific system. Asking users to recall a single password and user ID for one system may seem reasonable, but with the proliferation of passwords, users are increasingly unable to cope. Research on human memory is extensive, but according to Sasse *et al.* (2001), the most important issues related to passwords can be summarized as follows:

- the capacity of working memory is limited;
- memory decays over time, meaning that people may not recall an item, or may not recall it 100% correctly;
- recognition of a familiar item is easier than unaided recall;
- frequently recalled items are easier to remember than infrequently used ones, and retrieval of very frequently recalled items becomes ‘automatic’;

- people cannot “forget on demand” items will linger in memory even when they are no longer needed;
- items that are meaningful (such as words) are easier to recall than non-meaningful ones (sequences of letters and numbers that have no particular meaning);
- distinct items can be associated with each other to facilitate recall, yet similar items compete against each other on recall.

Research conducted by Sasse *et al.* (2001) regarding login failure (i.e. users forgetting passwords) found that the login usually failed because:

- they recalled the password partly, but not 100% correctly;
- they recalled a different password from the one required (i.e. a previously used password for the same machine, or a password for a different machine).

As pointed out by Sasse *et al.* (2001), this demonstrates the basic memory mechanisms (described above) in action: Items decay in memory unless they are frequently recalled, and recall of similar items causes interference. The likelihood of 100% correct recall of infrequently used items is extremely low. This means that a password mechanism that demands 100% accurate recall every time is an extremely bad match for infrequently used systems. That the results for 6-digit PINs are even worse confirms the importance of password content. It also indicates that a token-PIN combination, frequently proclaimed as a more usable substitute for passwords, is likely to cause more problems with infrequently used systems than a standard password. In Study 2 (the analysis of the password resets), the author found that 91.7% of resets were caused by “normal users” (i.e., more than 90% of users cannot cope with the password mechanism in the way they were expected to, which is a negative result in terms of the usability of password mechanisms).

4.3.2.4.1 Password Policies

The growing number of systems with which users have to interact creates memory problems (see section above). The problem is often exacerbated by password policies, which usually state rules, for example, that:

- passwords must be strong, such as a pseudo-random mixture of letters (upper/lowercase), numbers, and characters;
- users should have a different password for each system;
- passwords should be changed at regular intervals, and accounts of users who do not comply are deleted or suspended.

4.3.2.4.2 Varying Systems

In most password systems, there is a great variability of user IDs and passwords across different systems (e.g., UNIX takes up to 8 characters, Windows 95/98 up to 14, and Windows 2000 up to 127). Some systems have highly elaborate content restrictions, or more specifically, password policies (e.g. a password must include upper/lowercase, at least six characters, alpha-numeric, and special characters), but these vary from system to system. The result is a huge demand on users' memories:

- users not only have to remember passwords, but also the system and user ID with which it is associated;
- users have to remember which password restrictions apply to which system;
- users have to remember whether they have changed a password on a particular system, and what they have changed it to.

4.3.3 Mental Models

Mental models are representations of the function and/or structure of objects in peoples' minds. Designing something requires that you understand what the person

wants to get done. Empathy with a person is distinct from studying how a person uses something. Empathy extends to knowing what the person wants to accomplish regardless of whether she has or is aware of the thing you are designing. You need to know the person's goals and what procedure and philosophy she follows to accomplish them. So mental models give you a deep understanding of people's motivations and thought-processes, along with the emotional and philosophical landscape in which they are operating.

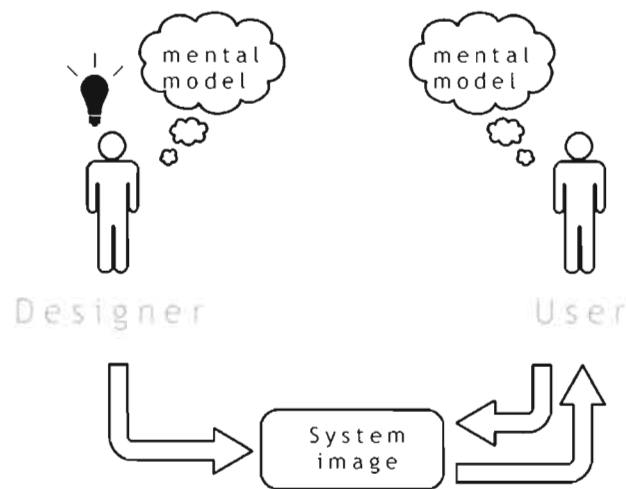


Figure 4.4: Mental Models (Norman, 1988).

Another important point is the "system causality conveyance" raised by Norman (1988), who made this a popular HCI debate within the research community. He used mental models to describe how a system is designed and implemented on the basis of the designer's mental model. Similar to the reader of a passage of text, the user develops a mental model of how s/he thinks the system works through interaction with the system as shown in Figure 4.4. This model is used to come to an understanding of the system, to anticipate system behavior, and to explain why the system reacts as it does. In other words, the designer materializes her mental model of a given design (e.g. a computer system), which becomes the only means of conveying her mental model to the user. Some characteristics of mental models are:

- may be incorrect or incomplete;

- can be "executed";
- are analogical representations, or a combination of analogical and propositional representations;
- are dynamically constructed when required.

There are two main types of mental models:

- functional models (good for everyday use);
- structural models (good for breakdown situations; difficult to acquire from usage experience only).

In short, computer systems should be designed in such a way that users can quickly acquire a good functional model of the system which is in accordance with their task model.

Chiasson and Biddle (2007) have found that in the usable security literature, and within their own studies, discussions invariably turn to the problem of mental models. User interfaces for security fall short of fostering useful mental models for users. One frequently cited explanation is that security is a complex issue and that users need more education in the area. This thesis' author disagrees with this argument. Not only is it shortsighted to assume that users will be adequately trained, but it is unrealistic to place such a burden on users.

The user interface should convey the information necessary for users to be able to easily predict and understand the consequences of their actions. This does not mean that users need to know the intricate details of how the system operates, but that they can form a reliable explanation in their minds that lets them interact successfully. The file managing metaphor is a good example: Users understand that files can be placed in folders, opened, closed, thrown into the recycle bin, and so on. But at no point do users need to know the underlying details of file storage and manipulation, such as disk blocks, index tables, and disk head scheduling.

Security interfaces do not yet help users form such mental models, and in fact still assume that users will have an understanding of underlying security concepts.

This places users in a vulnerable position. They lack the necessary knowledge, they must rely on inadequate interfaces to deduce what is happening, and they must make decisions that could potentially place them at risk. A wrong decision can give attackers valuable information or leave a user's system vulnerable. Alternatively, a wrong decision can also hinder a user's productivity because the security mechanisms now prohibit desired activities. It is not surprising that users prefer not to deal with security issues if they can avoid them.

Security interfaces must foster useful mental models. Therefore, researchers and designers must also be careful to accurately identify users' mental models when running usability studies so that an accurate and unbiased understanding of the usability of systems can be obtained. These are not easy tasks, but ones that must nevertheless be accomplished to achieve usable security.

4.4 The Cognitive Model of User Authentication (CMUA)

Based on the careful consideration of the information presented in the previous chapter, this thesis presents the Cognitive Model of User Authentication (CMUA) to explain how and what cognitive processes such as attention and memory, are involved specifically in user authentication tasks. The goal of developing this model is to have an understanding of human cognition related to user authentication and the interactions involved⁷⁴. CMUA also serves as the basis for the development of the Usable Security Symmetry inspection method. In addition, the GLEAN3 (GOMSL) and SOAR cognitive architectures which have been adapted for the development of the CMUA are briefly described in this section.

⁷⁴ An alternative approach to demonstrate the CMUA would be by showing how (Miller, 1956)'s works would appear in SOAR (Laird, 2008), then in CMUA, and afterward- how these works would influence the creation of the USS review questions and operators-or even further, what - impact they would have at the end on the execution time of the operators or other factors. Despite the fact that this is an interesting approach, it is out of the scope of this thesis.

4.4.1 Why Use a Cognitive Architecture?

Before proceeding with CMUA, it is worth noting what a cognitive architecture is and what the approach to modeling is. Cognitive architectures are an approach to modeling behavior that presupposes that there are two components to behavior, the architecture and knowledge. The architecture is comprised of cognitive mechanisms that are fixed across tasks and essentially fixed across individuals. These mechanisms usually comprise some type of perception and motor output, some sort of central processor, some working memory or activation of declarative memory, and some way to store and apply procedures. These mechanisms are used to apply task knowledge to generate behavior. The aim is to outline a cognitive architecture that captures a selection of cognitive processes in an integrated manner, and therefore to provide integrated explanations of a broad array of data (Laird and Congdon, 2009; Ritter, 2004).

Cognitive architectures must embody strong hypotheses about the building blocks of cognition that are shared by all tasks, and how different types of knowledge are learned, encoded, and used, making a cognitive architecture a software implementation of a general theory of intelligence.

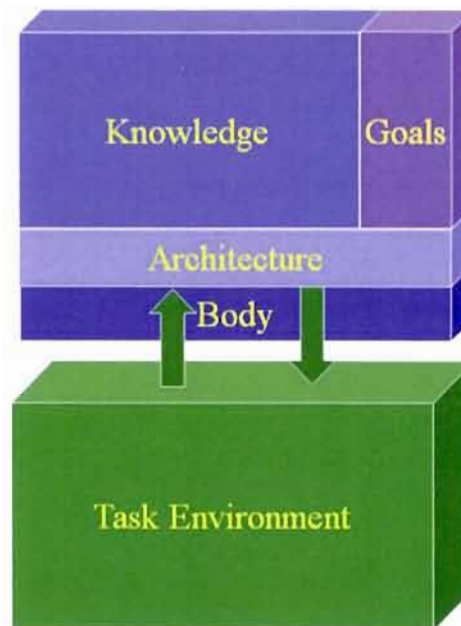


Figure 4.5: A basic cognitive architecture (Laird, 2009).

A basic cognitive architecture for which the Goal is to bring all existing Knowledge to allow select- actions (Task Environment, Body) to achieve goals is shown in Figure 4.5. Learning is implied in Knowledge.

There are several types of architectures that are or that could be used for evaluating interfaces and predicting task time and errors. These include descriptive architectures (Kieras, 1999), symbolic (Laird *et al.*, 1987) and hybrid architectures, intelligent agent architectures, and connectionist architectures. CMUA looks at how cognitive processes take place in a particular cognitive architecture. To this end, this thesis adapts and refines two recognized architectures for cognitive modeling such as GLEAN3 (GOMSL) (Kieras, 1999) and SOAR (Laird, 2008) in order to build the CMUA. The result is a descriptive architecture which helps in system design, and offers a first view of how the human interaction process occurs in user authentication.

4.4.1.1 GLEAN3 Cognitive Architecture

GLEAN3 (GOMS Language Evaluation and Analysis) is a computationally realized version of the Model Human Processor (MHP), being based more on the EPIC (Executive-Process/Interactive Control) architecture (Kieras and Meyer, 1997) for human information processing that precisely accounts for the thorough timing of human perceptual, cognitive, and motor activity. EPIC provides a framework for constructing models of human-system interaction that are precise and comprehensive as much as necessary to be useful for practical design purposes. EPIC depicts a state-of-the-art synthesis of results on human perceptual/motor performance, cognitive modeling techniques, and task analysis methodology, implemented in the form of computer simulation software. GLEAN has been used in numerous domains including military command and control, aircraft maintenance, and web-applications. Current development is extending GLEAN to better support error analysis and error-tolerant design (Kieras, 1996).

The GLEAN3 basic architectural structure is shown in Figure 4.6.

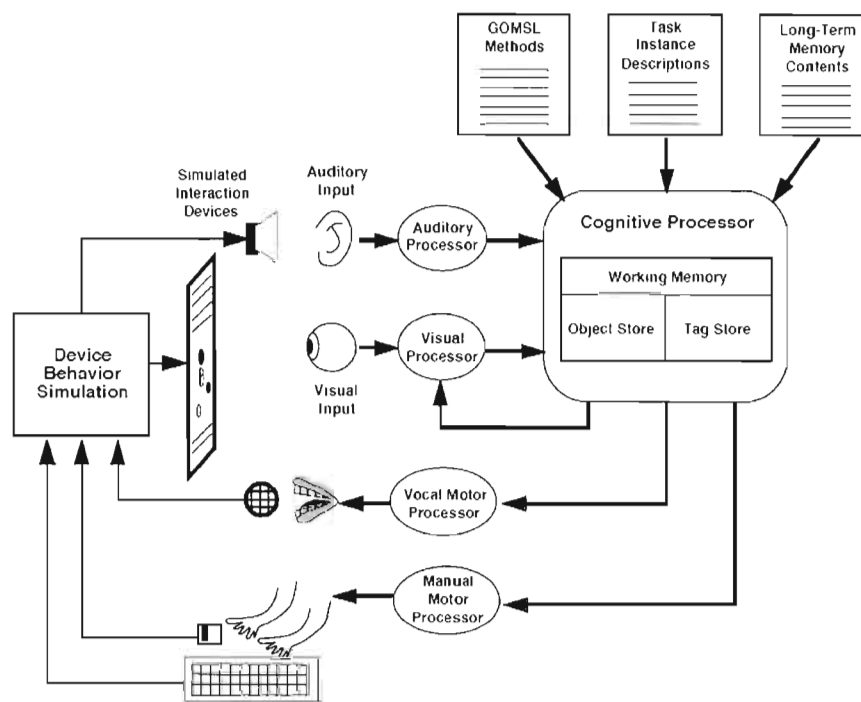


Figure 4.6: The GLEAN3 cognitive architecture (Kieras, 1999).

Human performance in a task is simulated by programming the cognitive processor with production rules organized as methods for accomplishing task goals. The EPIC model then is run in interaction with a simulation of the external system and performs the same task as the human operator would. The model generates events (e.g. eye movements, key strokes, vocal utterances) whose timing is accurately predictive of human performance.

4.4.1.2 SOAR Cognitive Architecture

The SOAR (State Operator and Result) architecture is a symbolic cognitive architecture, created by Laird *et al.* (1987) (Figure 4.7). It is both a vision of what cognition is and an implementation of that vision through a computer programming architecture for Artificial Intelligence (AI). SOAR has been broadly used by AI researchers to model diverse aspects of human behavior.

The main goal of the SOAR project is to be capable to handle the full range of capabilities of an intelligent agent, from highly routine to exceedingly complex open-ended problems. To this end, it needs to be capable to generate representations and use appropriate forms of knowledge (i.e. procedural, declarative, episodic, and possibly iconic). SOAR should then deal with a collection of mechanisms of the mind. Also underlying the SOAR is the vision that a symbolic system is required and enough for general intelligence. This is known as the physical symbol system hypothesis by Newell and Simon (1997) which states that “A physical symbol system has the necessary and sufficient means for general intelligent action.”

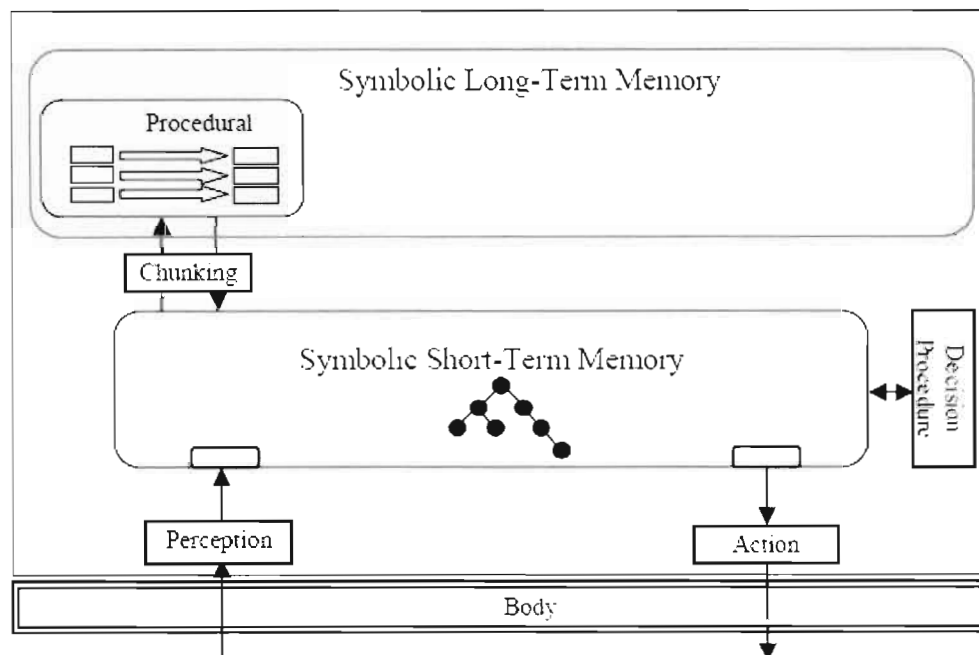


Figure 4.7: The SOAR (State Operator and Result) architecture (Laird *et al.*, 1987).

SOAR consistently illustrates Short-Term Knowledge (STK) as a network of active symbols. Long-term knowledge (LTK) is a collection of condition-action rules. The conditions of each rule build up a pattern to match against the active symbol network. When a rule's condition matches, the rule runs by carrying out its actions. These actions might entail -adding (or deleting) symbols in the STK structure.

To deal with complexity, SOAR includes a goal hierarchy, allowing successive decomposition of problems into component sub-problems. SOAR includes mechanisms to generate new goals automatically in response to a system's LTK and existing situation. SOAR depicts perceptual and conceptual knowledge consistently in STM; therefore, new actions flow from preceding actions and from changes in the external environment. The rule system also incorporates pattern-matching technology, allowing quick processing. Hence, SOAR is well suited for the development of intelligent systems that must produce actions in time analogous to human decision time.

SOAR includes an automatic learning mechanism based on the psychological concept of chunking. SOAR learns new chunks by compiling sequences of actions that change STM in particular ways. New chunks fit consistently into a system's existing long-term rule set. Therefore, a SOAR system can incrementally learn new facts about the world, as well as more proficient representations of its original LTK.

4.4.2 CMUA Cognitive Architecture

A cognitive model is a representation of some aspects of the user's understanding, knowledge, intentions or processing. Building a model of how a user works allows us to foresee how s/he will interact with the interface. One of the available modeling techniques, as already explained in previous section, is Model Human Processor (MHP). A basic model of human performance, it is intended to offer gross predictions of system behavior. MHP also considers humans as information processing systems such as a collection of memories and processors, as well as a set of principles ("principles of operation").

The CMUA is a cognitive model which has been built for the purposes of comprehension of the user authentication methods related to cognitive processes. It is focused on how and what cognitive processes interact (e.g., perception, memory, etc.). Figure 4.8 shows the basic architectural structure of CMUA, version 0.1.

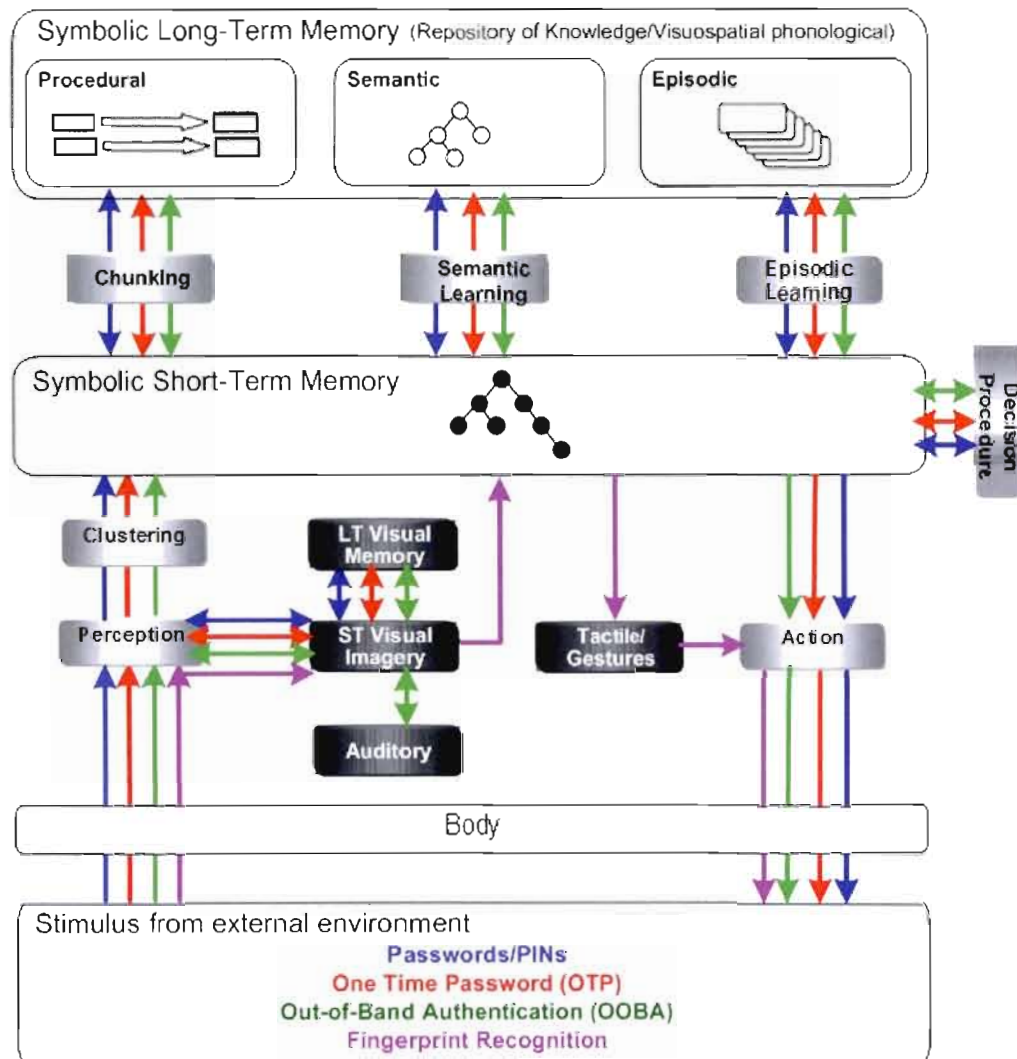


Figure 4.8: The CMUA cognitive architecture.
Adapted from (Kieras, 1999) and SOAR (Laird, 2008).

The CMUA consists of a LTM, which is encoded as production rules, and a STM, which is encoded as a symbolic graph structure so that objects are able to be represented with properties and relations. Symbolic STM holds the agent's evaluation of the current situation derived from perception and via retrieval of knowledge from its LTM. Action in an environment takes place through creation of motor commands in a buffer in STM. The decision procedure selects *operators* and detects *impasses*. At

the lowest level, CMUA's processing consists of matching and firing rules. Rules provide a flexible, context-dependent representation of knowledge, with their conditions matching the current situation and their actions retrieving information relevant to the current situation. In fact, many rule-based systems select a single rule to fire at a given time, and this serves as the venue of choice in the system – where one action is chosen instead of another.

CMUA lets additional knowledge to exert influence on a decision by introducing *operators* as the venue for choice and employing rules to propose, assess, and apply operators. Rules perform as an associative-memory that retrieves information pertinent to the present situation, so because of this, rules fire in parallel. In CMUA, there are rules that *propose* operators that generate a data structure in working memory representing the operator and an *acceptable preference* so that the operator is able to be taken into consideration for selection. There are also rules that *evaluate* operators and generate other categories of preferences that prefer one operator to another or provide some sign of the usefulness of the operator for the present situation. Lastly, there are rules that *apply to* the operator by making changes to working memory that reflect the actions of the operator. These changes might be solely internal or start external actions in the environment. This approach supports a flexible representation of knowledge about operators. There can be many reasons for proposing, selecting, and/or applying an operator, some that are very specific and others that are quite general.

CMUA possesses a shared STM where knowledge from Perception and LTM are combined to offer an integrated representation of the current situation. It has an “incline” decision procedure that supports context-dependent reactive behavior, but in addition supports automatic impasse-driven sub-goals and meta-reasoning. Chunking is CMUA's learning mechanism that converts the outcomes of problem solving in sub-goals into rules compiling knowledge and behavior from deliberate to reactive (Laird, 2008).

The four most representative categories of user authentication methods are demonstrated within the CMUA as follows: *Password/PINs (PPs)*, *One-Time Password (OTP)*, *Out-Of-Band-Authentication (OOBA)*, and *Fingerprint Recognition (FR)*. Figure 4.8 depicts the user authentication tasks with their corresponding cognitive and motor process- flows shown in colored arrows as follows: → PPs → OTP → OOBA → FR.

A description of the CMUA components is given in the next paragraphs.

4.4.2.1 CMUA Components

- **Stimulus from External Environment:** The external environment consists of PPs, OTP, OOBA, and FR user authentication tasks.
- **Body:** Users receive stimuli through their sense organs, the five senses (sensory information). Sensory memory corresponds approximately to the initial 200-500 milliseconds after an item is perceived.
- **Perception:** The Perception component is represented by the brain, which selects, organizes, and interprets sensory information.
- **Auditory:** The Auditory memory component accepts either speech or sound inputs and makes them available to the WM. It is the capability to remember what an individual has heard. It involves being able to take in information that is introduced to you, process that information, store it in STM, and then recall what you have heard. Mostly, it involves the task of attending to, listening, processing, storing, and recalling.
- **Visual Imagery:** The CMUA provides a set of processes and memories to support visual imagery, which includes depicted representations. The *ST Visual Imagery* module includes a STM where images are built and manipulated; a LTM that includes images that can be retrieved into the STM; processes that manipulate images in STM, and processes that generate symbolic structures from the visual images. Visual imagery is controlled by

the symbolic system, which emits commands to build, manipulate, and inspect visual images. Additionally, visual imagery allows processing that is not possible with only symbolic reasoning, such as determining which letters in the alphabet are symmetric along the vertical axis (A, H, I, M, O, T, U, V, W, X, Y) (Laird, 2008).

- **Visual Memory:** Visual memory is an individual's capability to remember what she has seen. Typically, three types of visual memory are found: Photographic memory, Iconic memory, and Spatial memory. *Photographic memory* is the capability to recall images and/or objects in memory with great precision and in plentiful volume). *Iconic memory*, is the sensory store for vision, a type of ST visual memory. Experiments performed by Sperling (1960) offers indication for a speedily decaying sensory trace, lasting only roughly 250 milliseconds after the offset of a display. Finally, *Spatial memory* can be considered a sub-category of visual memory because it relies on a cognitive map. Cognitive mapping is a sort of mental processing by which a person is able to acquire, code, store, recall, and decode information about the relative locations and attributes of events in their daily or symbolic spatial environment.
- **Tactile/Gestures:** A gesture is "an imprecise, context-dependent event that conveys the user's intentions" (Voyles *et al.*, 1995). In this case, the gestures are force impulses - nudges - on the end-effector. Because gestures are context dependent, state information must be associated with each gesture. This state information is generally application specific. Using a linguistic analogy, the raw gestures form a gestural alphabet along with the state information. Gestural words are assembled from the raw gesture and its associated context by the gesture recognizers (preprocessors in Figure 5).
- **Clustering:** Classification is a basic human conceptual activity. For example, children gain very early knowledge of classifying objects in their environment

and correlating the resulting classes with nouns in their language. “Cluster analysis” is a broad term used for a multiplicity of procedures that can be employed to create subtypes (i.e. classification). These procedures form “clusters” or groups of highly similar entities. More particularly, a clustering method is a multivariate statistical procedure that starts with a data set including information about a sample of entities and attempts to rearrange these entities into somewhat homogeneous groups. The Clustering component detects statistical regularities in the flow of experiences and automatically and dynamically creates new symbolic structures that represent those regularities, providing a mechanism for automatically generating new symbols and thus concepts that can be used to classify perception (Laird, 2008). Those new symbolic structures enrich that state representation. Clustering is in fact sub-symbolic, where non-symbolic perceptual structures are amalgamated collectively to create symbols.

- **Symbolic Short-Term Memory:** Symbolic STM holds the agent’s assessment of the current situation derived from perception and via retrieval of knowledge from its LTM. STM allows recall for a period of several seconds to a minute without rehearsal, and its capacity is also very restricted. According to Miller’s (1956) experiments, the store of STM was 7 ± 2 items, the Magical Number Seven, Plus or Minus Two.
- **Decision Procedure:** The decision procedure selects *operators* and detects *impasses*. It helps context-dependent reactive behavior, but also helps automatic impasse-driven sub-goals and meta-reasoning.
- **Chunking:** Chunking signifies arranging items into familiar, manageable units. Each chunk collects a number of parts of information from the environment into a particular unit. Chunking refers to an approach for making more

proficient use of STM by recoding information. In general, Herbert A. Simon⁷⁵ has used the term *chunk* to indicate LTM structures that can be employed as units of perception and meaning, and *chunking* as the learning mechanisms guiding -the acquirement of these chunks.

- **Semantic Learning:** Declarative knowledge can be separated into elements that are known (i.e. facts) and elements that are remembered (i.e. episodic experiences). Semantic Learning and memory provide the capability to stock up and retrieve declarative facts about the world, such as cars have wheels, eggplant is a vegetable, and pyramids are in Egypt. This capability increases the ability to create agents that reason and employ general knowledge about the world. Semantic Learning is built up from structures that take place in STM. A structure from Semantic Learning is retrieved by generating a cue in a particular buffer in STM. The cue is then employed to seek for the best partial match in semantic memory, which is then retrieved into STM.
- **Episodic Learning:** Episodic memory is the type of memory that remembers events that are observed through experience (Nuxoll and Laird, 2004) (e.g. a snapshot from one's past experience). It includes specific instances of the structures that occur in STM at the same time, providing the capability to remember the context of past experiences as well as the temporal relationships between experiences (Nuxoll and Laird, 2004). An episode is retrieved by generating a cue in a particular buffer in STM. The cue is then employed to seek -the best partial match in semantic memory, which is then retrieved into STM. The next episode can also be retrieved, providing the capability to replay an experience as a succession of retrieved episodes. Episodic memory is task-

⁷⁵ Herbert. A. Simon was an American psychologist whose research ranged across the fields of cognitive psychology, computer science, public administration, economics, management, philosophy of science and sociology. He was a professor at Carnegie Mellon University, Pittsburgh, PA (US). With nearly a thousand frequently -quoted publications, he is one of the most authoritative social scientists of the 20th century.

independent and therefore available for every problem, providing a memory of experience not available from other mechanisms.

- **Symbolic Long-Term Memory:** Symbolic LTM contains images that can be retrieved into the STM, and which are encoded as production rules. See also 4.3.2.2.3 Long-Term Memory (LTM).
- **Action:** Action in an environment takes place through generation of motor commands in a buffer in STM.

4.4.2.2 CMUA Processing Cycle

- *Input.* Users receive stimuli through their sense organs, the five senses. “Stimulus from the external environment” is represented by the user authentication tasks (sensory information).
- *Perception* is carried out through different perceptual processors such as visual memory, visual imagery, tactile gestures, and auditory. In visual memory SiteKey is an example where first you recognize a unique image you chose and an image title such as "Whales are fascinating creatures" that you created to accompany your image). In visual imagery, an example is when a user visualizes a 4-digit- PIN like “2222” in his mind and associates it visually with swans). In tactile gestures, a user places his finger on a device that reads the thumbprint), and finally in auditory, a user gets a sound alert such as a beep or a tone- on her wireless device warning her that she has just received a text message, which includes the token code that she must authenticate with when using OOBA) depending on the current type of user authentication task being performed.
- *Changes to perception* are processed, and clustering is undertaken if needed. Then changes are sent to the Symbolic STM.

- *Chunking, Semantic Learning, and/or Episodic Learning* are undertaken, if needed. *Chunking*, in a random 10-character password, a chunk is a symbol, and as Miller (1956) probably would claim, -the majority of people cannot remember 10 random symbols. Users are probably more certain to remember a 10-character pass phrase comprised of 2 or 3 words or chunks, let's say memorizing a *passphrase* (i.e. a long password with inserted spaces) such as "My voice is my password"). In *Semantic Learning*, an emphasis is given on the importance of the interactive and psychological situations in which learning occurs. Learning is identified with acquisition of knowledge (e.g. user is required to memorize a sequence of 4 images of the *same category* (enrollment): let's say that the category is `racquet sports` so this would represent by the following sports: badminton, racquetball, squash and tennis). Then when later she authenticates to the system, she is presented with a series of images from categories that she pre-selected mixed with images from random categories. After that she retrieves it from LTM by entering that sequence in a fixed order). Finally in *Episodic Learning*, a change in behavior takes place as a result of an event (e.g. to change a password, users enter and confirm a new password that is at least eight characters long and which includes at least one number. Users cannot re-use any of their previous passwords).
- *Elaboration*. Rules compute entailments of STM. For example, a rule might test if the goal is to grasp a hardware token, the hardware token's distance, and the user's reach, and then create a structure signifying whether the hardware token is within reach.
- *Operator Application*. The actions of an operator are carried out by rules that match the present situation and the present operator structure. Several rules can shoot in parallel and in sequence offering a means for encoding operator actions.

- *Output (Action)*. Any output commands are passed on to the motor system.

Summary of the topics discussed in Chapter 4: The Cognitive Science Axis.

This chapter introduced the cognitive axis approach as follows: Cognitive Ergonomics, the Main Cognitive Areas of Focus Relating to User Authentication, and the Cognitive Model of User Authentication (CMUA). The CMUA provides a relevant contribution to the understanding of what and how cognitive processes are involved in user authentication. On the basis of this formalization, CMUA is the first attempt to build a cognitive model (architecture) for user authentication.

CHAPTER V

THE COMPUTER SCIENCE AXIS

5.1 Introduction

The previous chapter has dealt with the cognitive processes involved in user authentication and their application in a cognitive architecture - the Cognitive Model of User Authentication (CMUA). Now that we understand what the cognitive processes are and how they should produce responses when users authenticate to a system, it is time to put those cognitive processes into operation by making use of the Computer Science Axis.

The Computer Science Axis is defined as one of the two-part vital holistic approach in conjunction with the Cognitive Science Axis to demonstrate the Usable Security Protocol (USP) which encompasses the USS inspection method and the demonstrational approach.

Human-computer interaction (HCI) is an area of research and practice that emerged in the early 1980s, initially as a specialty area in Computer Science. HCI has expanded rapidly and steadily for three decades, attracting professionals from many other disciplines and incorporating diverse concepts and approaches. To a considerable extent, HCI now aggregates a collection of semi-distinct fields of research and practice in human-centered informatics.

Security has been an important quality factor in many types of computer-based information systems, including, for instance, authentication mechanisms, -banking software such the ones used in MTMs, and many others. Due to the fact that such systems are characterized by their user interface components, usability is also required. Furthermore, there is a common but false belief that security is only related to the software functionality and can be designed independently from the software usability which is related to the User Interface (UI) component (Seffah and Metzker, 2004). In fact, the meaning of what -a UI is and how usability is defined are perhaps

major underlying obstacles that explain such erroneous conceptions. Indeed, it gives the impression that the UI is a thin layer sitting on top of the “real” system and that usability can be conceived of independently from the other quality factors such as security.

The Human Computer Interaction Security (HCI-SEC) research community has constantly been reporting the bad usability of security systems and its consequences, vulnerabilities, and threats (Whitten and Tygar, 1999; Stiegler *et al.*, 2004; Saltzer and Schroeder, 2000). Also a significant number of usability problems causing security failures were found in the Pretty Good Privacy (PGP) study (Whitten and Tygar, 1999), a public key encryption program mainly intended for email privacy and authentication. This thesis’s author agrees with the idea fundamentally supported by Whitten and Tygar (1999) that there is a need for a comprehensive model of *usable security* more specifically for user authentication methods. This model should include either process-and-product related usability characteristics such as effectiveness, efficiency, satisfaction, security, and learnability.

Usable Security is defined in this thesis as the study of how security information and usability factors should be handled either in the front-end and back-end processes, taking into consideration resources and costs. Usable Security is imperative from the *user's perspective* (e.g., authenticating appropriately in a computer system without circumventing the security policy), from the *developer's perspective* (e.g., success or breakdown of a token provisioning application), and from the *management's perspective* (e.g., enforcing a strong password policy can be a major constraint to the usability of a system).

Modeling is always a goal-driven activity since every model has a purpose. Testing and validation (or demonstration) has to be done with the purpose of the model in mind. To this end, the validation phase is based on a *design-driven demonstration* of the Cognitive Model of User Authentication (CMUA) by using one

of the most representative authentication methods used for user authentication: One-Time-Password (OTP).

The Development of an Authentication Method (AM) is subject to the following phases: Evaluation, Definition, Development, and finally Readiness, as shown in Figure 5.1.

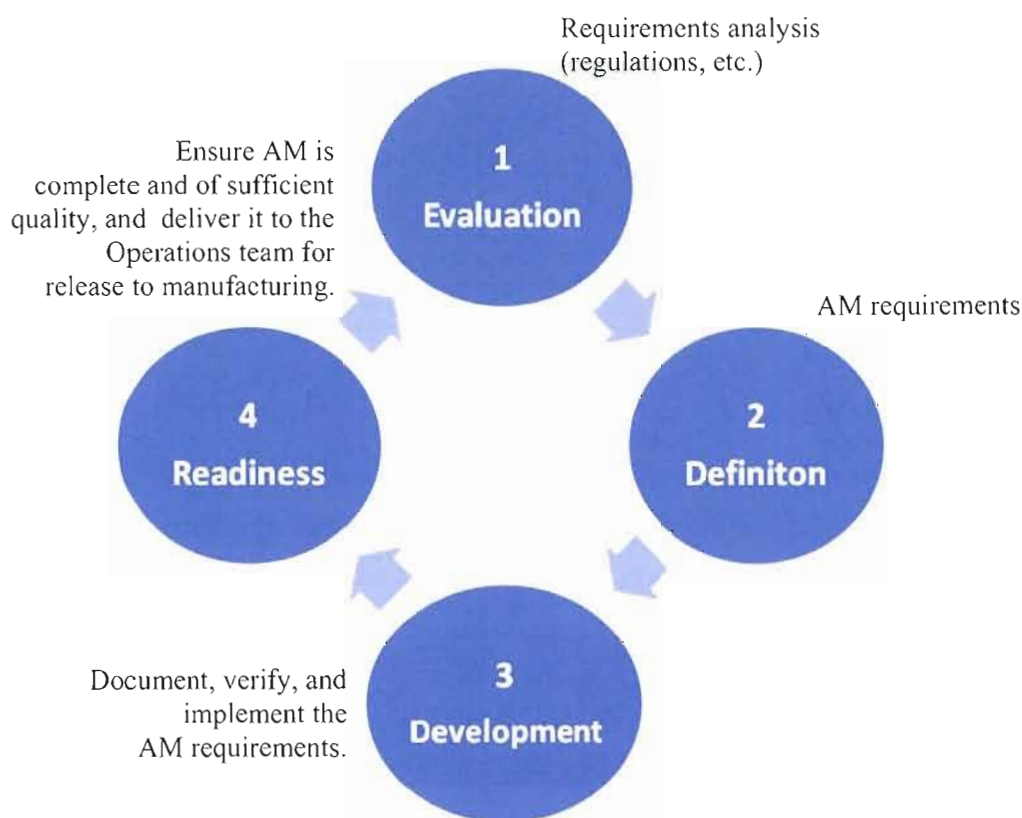


Figure 5.1: Authentication Method Development Life Cycle.

In Step 1, the “Evaluation” is made by establishing the AM Requirements Document (AMRD) version 0.x, which is the input, and the AMRD version 1.0 which is the output comprised of a list of AM requirements. Next, in Step 2, the “Definition” takes over by defining the AM System Requirements (AMSR), a- UXD Spec 0.x document, and the AM schedule. Then, in Step 3, the development phase proceeds by executing the UXD Spec version 1.0, prototyping, usability testing, and testing (i.e. Quality

Engineering); finally in Step 4, the Readiness phase releases the AM to manufacturing and market.

5.2 Security as a Usability Characteristic

Researchers, as well as standard organizations, have provided an additional perspective on usability that refers to a specific usability characteristic, which is *Security*. Figure 5.2 lists some of the standards where security is included within their usability model as follows:

- ITSEC: Information Technology Security Evaluation Criteria IEC 300: It presents software as security-critical.
- International Standards Organization (ISO) ISO/IEC 13407: It describes human-centered design as a multidisciplinary activity incorporating human factors and ergonomic and technical knowledge with the objective of raising efficiency and effectiveness, improving human working conditions, and opposing possible unfavourable effects of use on human health, security, and performance.
- ISO/IEC 9126: It defines security, which is a sub-characteristic, as a set of software attributes that relate to its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.
- Federal Aviation Administration (FAA): Security is a characteristic of the CHI, which is particularly important in an industrial context.

These standards consider that good usability is a significant condition for human security in critical systems, such as medical apparatus or nuclear power stations. Within our model, this thesis adopts this perspective of security.

The usable security community acknowledges that for a system to be secure, it has to be usable. This means that even the most secure system can fail if it is not used appropriately. In Whitten and Tygar's (1998) study, the authors demonstrated that

usability of security has different requirements than usability of IT in general. As already mentioned in the introduction of this thesis, it is broadly held that security and usability are two opposed goals in system design (Cranor and Garfinkel, 2005; Jøsang *et al.*, 2007; Nielsen, 2000). However, there are several cases in which security and usability can be synergistically enhanced by reviewing the usable security approach. For example, improving the interface and changing the way users interact with the system (Yee, 2004 Nielsen, 2000; Sasse *et al.*, 2001). Additionally, Polaris (Stiegler *et al.*, 2004) allows users to configure most applications so that they launch with only the rights they need in order to get the job done (i.e. Principle of Least Authority), thereby demonstrating that it is feasible to build systems that are more secure, more functional, and easier to use.

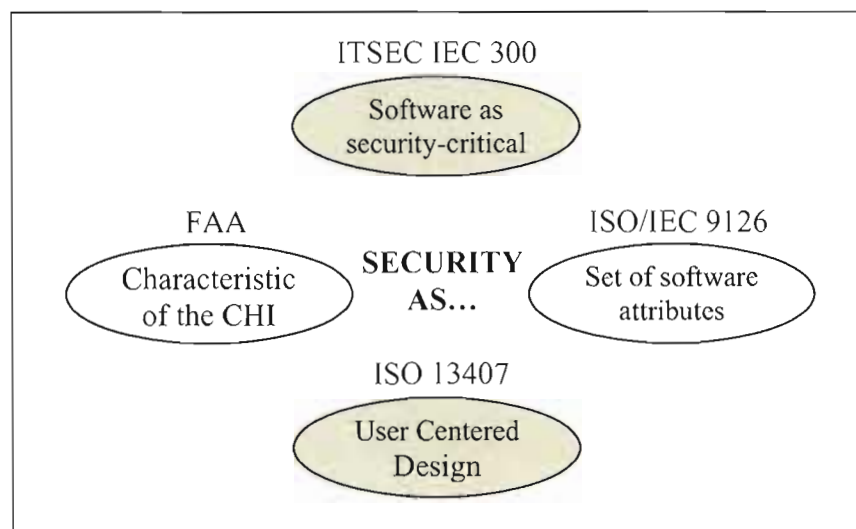


Figure 5.2: Security as a usability characteristic.

End-users, IT administrators, developers, and security designers are the main concern for the field of usable security. According to Zurko and Simon (1996), software developers working on UI design and evaluation lack UCD tools. Tools are needed to support security designers in acquiring and sharing UCD and software engineering best practices. This is especially the case with user authentication. USS aims to help security designers to design, inspect, and evaluate the usability as well as the security

aspects of user authentication mechanisms. From this thesis research perspective, a security designer is an expert in computer security and possesses a reasonable understanding of the skills, mindset, and background of the users who are expected to perform an authentication task. USS integrates usable security earlier into the requirements and design phases of the user authentication methods development lifecycle.

5.3 Usability Factors and Usability Criteria Mapping

Generally speaking, the quality of a software product is specified by its internal and external capability to assist users to achieve their goals, and the organization's goals, therefore improving productivity and human health. The (ISO/IEC 9126-1:2001) standard is founded on a quality model for software products that consists of two parts: (1) external and internal quality and (2) quality in use. In brief, internal quality refers to properties of the non-executable portion of a software product during its development, and metrics for internal quality in general refers to the quality of intermediary deliverables, for instance the source code for a prototype version. External quality in turn refers to the behavior of the computer system of which the software product is a part.

To build the usability factors and usability criteria mapping, an adaptation of the Quality in Use Integrated Measurement (QUIM) (Seffah *et al*, 2005) hierarchical model has been used. "Quality in use is a kind of higher-order software quality construct, and it concerns whether a software product enables particular users to achieve specified goals with effectiveness, productivity, safety, and satisfaction in a specific context of use."

QUIM adopts the viewpoint of most HCI standards that serve to foster quality in use in different factors: Usability is broken down into factors, then into criteria. For the purposes of this thesis, nine (9) usability factors and eight (8) usability criteria have been developed as follows:

1. *Efficiency*: the capability of the software product to provide appropriate performance, relative to the amount of resources used under stated conditions.
2. *Effectiveness*: the capability of the software product to enable users to achieve specified goals with accuracy and completeness in a specified context of use
3. *Productivity*: the capability of the software product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.
4. *Satisfaction*: the capability of the software product to satisfy users in a specified context of use.
5. *Learnability*: the ease with which a user can master the required features for achieving his or her goals.
6. *Safety*: whether a software product limits the risk of harm to people or other resources.
7. *Trustfulness*: the degree of faithfulness a software product offers to its users.
8. *Accessibility*: the capability of a software product to be used by permanently or temporarily disabled persons (i.e. vision, hearing, motor, cognitive and language impairment).
9. *Universality*: whether a software product accommodates a diversity of users (e.g., takes cultural considerations into account).
10. *Usefulness*: the degree to which a software product actually helps to solve users' practical problems.

Each factor is broken down into criteria as follows:

1. *Minimal Action*: capability of the application to help users achieve their tasks in a minimum number of steps.
2. *Minimal Memory Load*: whether a user is required to keep minimal amount of information in mind in order to achieve a specified task.
3. *Operability*: amount of effort necessary to operate and control an application.
4. *Privacy*: whether users' personal information is appropriately protected.
5. *Security*: capability of the application to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access.
6. *Load Time*: time required for the application to load (i.e., how fast it responds to the user).
7. *Resource Safety*: whether resources including people are handled properly without any hazard.

Consider this example to illustrate the applicability of the usability factors and their corresponding criteria using a MTM as shown in Table 5.1. It demonstrates 9 usability factors and 7 usability criteria. The relation between, for instance, the usability factor *Efficiency* is assumed to correspond to the criteria *Minimal Action*, *Operability*, *Privacy*, *Resource Safety*, and *Minimal Memory Load*.

#	Task Scenario	Security Problem/Threat	Usability Criteria	USABILITY FACTORS								
				Efficiency	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality	Usefulness
1	Authenticate yourself	Storage of Information: replay attacks, eavesdropper and session hijacking, and Man-in-the-middle and verifier impersonation.	Minimal Action	●	●		●			●		
2	Transfer funds to an international bank account	Access Control	Operability	●	●				●	●		●
3	Buy a ticket concert	Sensitive Information	Privacy	●	●				●	●		●
			Minimal Action	●	●		●			●		
4	Access your MTM with your cell phone	Credentials across several channels	Security					●	●			●
5	Deposit your check using checking image	Encryption	Loading Time	●		●					●	●
6	Send a silent alarm	User physical safety	Minimal Memory Load	●	●		●			●	●	●
			Resource Safety					●				

Table 5.1: Usability factors and Usability criteria mapping for a MTM.

5.3.1 User Authentication Use Cases

The following user authentication use cases have been developed to serve as the task scenarios (Table 5.2): Authenticate to the MTM and sub-systems, Transfer funds to an international bank account, Buy a concert ticket, Access your MTM with your cell phone, Deposit your check using checking image, and finally Send a silent alarm.



Table 5.2: User authentication use cases.

USE CASE			USABILITY		SECURITY	
#	Name	Scenario	Required Features	Problem	Scenario	Problem
1	Authenticate to a MTM and other systems (Multipurpose <u>C</u> ontactless <u>S</u> mart <u>C</u> ard [MpCC])	Customer must log in using a multipurpose contactless (*) token-based authentication in combination with KBA method (PIN) in order to have access to the MTM's system. This MpCC can provide access to other services than those supplied by the MTM such as keeping medical records, physical access to buildings (when used with a reader), and make small purchases with an electronic purse. (*) A contactless smart card includes an	To log into the MTM: <ul style="list-style-type: none">• Insert your MpCC in the MTM's card slot;• Enter your PIN. To authenticate to a medical institution logging on to a computer with the MpCC: <ul style="list-style-type: none">• Insert your card in the smart card reader;• Enter your PIN. To authenticate to a facility (c.g., building) that is physical access control which authenticates individuals and permits access to physically secure areas:	User Convenience (dealing with multipurpose vs. one purpose smart cards).	An MpCC improves in fact user convenience, not having to carry several cards and usually memorizing different PIN codes. However, it raises the risk if the card is lost or gets stolen, and also if the card is forgotten by the customer in the reader of the MTM. Using a one purpose card is more secure, but means the user will need to carry one card for each application which is not so convenient.	Storage of Information

						<p>without being aware of it and the system and the application need to ensure that transactions come to a regular end. For example, in ticketing applications, the system needs to ensure the user is not charged twice, but actually just once); <i>covert transactions</i> (fraudulent merchants communicate with the user's card, triggering forged transactions using forged readers. For example, such merchants could process a number of transactions instead of only one); and finally <i>denial of service</i> (the user and the issuer should be aware of the denial of service attack, in which, for example all monetary units could be debited from the card at a distance, thus denying the user access to the service s/he has paid for) (Handschuh2004).</p>
<p>embedded smart card secure microcontroller, internal memory and a small antenna and communicates with a reader through a contactless RF interface.</p>	<ul style="list-style-type: none"> Bring your MpCC close to a card reader. To debit an amount from a electronic purse(i.e., the card can be loaded with "electronic" value that can be decremented as purchases are made): Pay your lunch with your multipurpose contactless smart card in the cafeteria. 					

USE CASE				USABILITY		SECURITY	
#	Name	Scenario	Required Features	Problem	Scenario	Problem	Scenario
2	Transfer funds to an international bank account	Customer must prove his identity again to protect access to particularly sensitive applications (i.e., authentication for high-value financial transactions) through a biometric-based authentication method (See figures in the next page).	When prompted, customer places his hand on the palm scanner.	Weighty Workload	Customer has to deal with both access control and strong authentication, that is, in this case three-factor authentication as described below: Biometric authentication (palm scanning): customer is not habituated with biometric identification. In fact, this customer has already been identified by a two-factor authentication method according to the task 1. However, he wants to perform a higher risk transaction so customer has to prove again his identity to the system.	Access Control	In high-value financial transactions environment, procedures to control access to several areas of the card become predominantly important. The degree of security changes with the degree of sensitivity of the data related to the application. The issue of data security becomes more complex for example in a high-value financial transaction because the application requires a different level of security. Some applications may require no security; others may be sufficiently protected by a PIN; others may require the use of biometrics to protect access to particularly sensitive applications which is the case here.

Continued from # 2.

#	USE CASE		USABILITY		SECURITY	
	Name	Scenario	Required Features	Problem	Scenario	Problem
2		 <p>MTM with a palm scanner embedded in the software (to the right on the screen).</p>  <p>MTM with a palm scanner placed outside the MTM (on the right).</p>				<p>In fact, we add the biometric feature as another security layer to the current system. The Palm Recognition authentication system is build up around taking a three dimensional view of the hand in order to determine geometry and metrics of such variables as length and thickness of fingers. How it works? First, an infrared scan is taken of the veins on the customer's palm, and the data from the scan is stored in an Integrated Circuit (IC) chip embedded in the cash card, or directly at the MTM biometric mechanism. When using the ATM, the customer places his hand on the palm scanner, which verifies whether the pattern of veins matches that stored on the card or in the MTM.</p> <p>Hand geometry can only be used for verification.</p>

USE CASE			USABILITY		SECURITY	
#	Name	Scenario	Required Features	Problem	Scenario	Problem
3	Buy a ticket concert	After the customer has made a bank transaction, she decides to buy a concert ticket that she has just seen in the “Buy a Concert Ticket to “The Cure!”” session at the MTM. This customer has already bought a concert ticket from an MTM once. At that time, she entered sensitive information such as her credit card number before to buy the ticket. However, now the MTM requests again the same customer to enter her sensitive information.	<ul style="list-style-type: none"> Click on “Buy a Concert Ticket to “The Cure” Today!” session; Select the number of seats; Select your credit card brand; Tap your credit card number; Tap credit card’s month and year of expiration. 	Cumbersome user input requirements	Customer has to enter sensitive information, here the credit card number, each time she purchases a concert ticket. It does not provide convenience for the customer. The customer should be able to enter sensitive information only at registration.	Sensitive Information
					Should the customer enter sensitive information only at registration or each time a concert ticket is purchased? The first method makes a “one-click feature” possible, meaning the customer only has to select an option to order the ticket, providing convenience for her. Nevertheless the second method is much better from a security perspective, but requesting more activity on the customer part.	

USE CASE				USABILITY		SECURITY	
#	Name	Scenario	Required Features	Problem	Scenario	Problem	Scenario
4	Access your MTM off-line with your cell phone	Customer accesses a MTM via cell phone in order to make her mortgage monthly payment. The cell phone is equipped with a special chip that enables to communicate with the MTM.	<ul style="list-style-type: none"> • Select "Access my MTM" from the cell phone main menu; • Enter your 4-digit PIN (the PIN is entered on the customer's cell phone keypad then transmitted to a central server and checked against file saved there); • Select "Make a Payment" from the MTM's menu; • Select the type of payment which is "Mortgage"; • Tap the exact amount; • Select "Submit". 	Overwhelm customers with complexity when dealing with different communication channels.	Customers have to manage complexity when dealing with different services offered through different types of communication channels such as MTM, Web, and WAP. Although it might be considered a convenient service when one does not have access physically to a MTM, it does place the burden on the customer with regards to the co-ordination of the MTM with the cell phone. In addition, customers will still be required to authenticate to the system by entering a PIN. Unlike passwords, PINs have no meaning to the customer, and then it might be even harder to remember than a password (i.e., passwords can be created to be pronounceable). PINs become harder to remember for customers who have many different ones to keep track of.	Credentials across several channels	Using the same authentication credentials for both WAP and MTM channel, can provide convenience for the customers. However, PIN code is the only acceptable alternative for the WAP channel, and is not considered to provide good enough security (i.e., longer PINs (6 or 8-digit PINs) would be more secure than 4-digit PINs). Additionally, when PIN is used for authentication over the phone, the risk of eavesdropping the telephone line is a supplementary threat, especially since it cannot be encrypted.

USE CASE				USABILITY		SECURITY	
#	Name	Scenario	Required Features	Problem	Scenario	Problem	Scenario
5	Deposit your check using checking image	Customer has to deposit a cheque where the MTM accepts envelope-free check deposit.	<ul style="list-style-type: none"> • Select “Deposit a Check”; • Select “Scan my Check”; • Select if you want a deposit’ paper or electronic receipt (i.e. it will be saved in your checking account). 	Lower response time to the customer.	Although it might be considered to reduce “perceivably” transaction speed (i.e., deposit a check without envelope), it might take more time up front to process the data (i.e., capture the check’s image) and the speed with which the MTM responds to the customer. If the checking image “seems slow”, customers might be less inclined to not use it.	Strive for security using encryption.	The speed with which the MTM captures the check’s image and the speed with which the MTM responds to the customer are very important. However, in this particular case the check’s image must ideally be encrypted in order to provide secure communication and storage of information. While improving security, encryption takes time and makes the system slower which may be very inconvenient for the customer. In short, there is a trade-off between having a secure communication and having a short response time.

#	USE CASE		USABILITY		SECURITY	
	Name	Scenario	Required Features	Problem	Scenario	Problem
6	Send a silent alarm	Customer has in his hands a certain amount of money which is going to be deposited in his account. He starts the process of authentication (but not finalize it) when he notices a suspicious person approaching to him. The intention of this person is obvious which is to steal the customer's money.	Customer enters an emergency PIN at the MTM alerting the bank's security system.	Difficulty remembering password or PIN especially under pressure.	People have even more difficulty remembering password or PIN specially under pressure. In case of stealing money of a customer the threat is clear.	Customer security while using MTMs
						Adoption of an emergency PIN system for MTM. Customer is able to send a silent alarm in response to a threat and get help from the bank. This is can be the case for example of a customer who has subscribed for this type of service with his bank.

5.3.2 Demonstrating the USS using a Multifunction Teller Machine

The good old ATM cannot defeat what this thesis designates as the new Multifunction Teller machine (MTM) in several aspects especially in the “transaction” factor. Being capable to perform up to 150 kinds of transactions ranging from straightforward cash withdrawals and deposits to fund transfer, to trading in stocks, to purchasing mutual funds or to cash a check using check imaging, to something as ordinary as processing the payment of electricity bills, booking air-tickets, purchasing concert tickets and making hotel reservations. A MTM is in effect the next generation of an ATM, fully integrated cross-bank MTM network providing functionalities which are not straightforwardly associated to the management of one's own bank account, such as loading monetary value into pre-paid cards (e.g., cell phones, tolls, service and shopping payments, etc.). Also the MTMs can provide advanced authentication capabilities such as palm recognition.

To illustrate how USS can be applied in a real world application, and how the USS elements such as usability and security factors were selected and determined for the MTM example, this thesis depicts one of the use cases described in the previous section: a three-factor authentication which is “Transfer funds to an international bank account”. The USS has been applied to determine which type of authentication method should be used for this particular use case.

A user, Bob, needs to transfer US\$5,000 to an international account by dealing either with access control and strong authentication. He first authenticates himself to the MTM using a smart card and a PIN (the bank PIN policy states that a PIN must have 4 digits and 1 letter). In high-value financial transactions environment, procedures to control access to several areas of the card become predominantly important. The degree of security changes with the degree of sensitivity of the data related to the application. The issue of data security becomes more complex in a high-value financial transaction because the application requires another layer of

security to the current system: Biometrics. As this represents a high-value transaction, the MTM asks for Bob to prove again his identity. So in addition to the bank card and PIN, Bob must employ a biometric authentication such as palm recognition - a multiple factor authentication.

5.4 The Usable Security Symmetry (USS) Inspection Method

The goal of making a system secure and usable will be successful only if it is a *pre-hoc* consideration. This strengthens the argument made by other HCI-SEC researchers (Balfanz *et al.*, 2004; Flechais *et al.*, 2003; Yee, 2004) that security and usability must be developed in unison from conception right through to development as an integral part of the system if they are ever to align perfectly. This thesis's author agrees with Yee (2006) that security and usability not only must be taken into consideration early and iteratively, but also *together*. According to Yee (2006), integrated iterative design means iterative development processes based on repeated analysis, design, and evaluation cycles, rather than linear processes in which security or usability testing occurs at the end. Although many teams have adopted iterative processes, few seem to incorporate security and usability throughout. Not only is it important to examine these issues early and often, it is vital to design the UI and security measures together. Iterating offers the opportunity to see how security and usability decisions affect each other. Moreover, since usable security requires UI design priorities that are not similar as those of universal consumer software, it should also require usability evaluation methods that are appropriate to security. Standard usability evaluation methods possibly will treat security functions as if they were primary rather than secondary goals for the user, leading to flawed conclusions.

Symmetry also plays an important role in understanding the framework of the USS and how it has been built. Among a variety of definitions of symmetry, a generalized concept of the term has been adopted which is defined as follows: "Symmetry is a relationship of characteristic correspondence, equivalence, or

identity among constituents of an entity or between different entities”⁷⁶. The entities are defined as security and usability, and the mentioned relationship is the final outcome which is the *usable security*. Another (notable) definition which relates to the definition of Symmetry for the purposes of this thesis is from Weyl (1952) who states that: “Symmetry, as wide or as narrow as you may define its meaning, is one idea by which man through the ages has tried to comprehend and create order, beauty, and perfection.” The word, *order*, is in fact a synonym of harmony. The utmost goal is that security and usability will no more be two separate entities, but will work in harmony to produce secure and easy to use authentication methods.

5.4.1 Definition

USS is a usable security inspection method which involves having a group of evaluators -systematically examine a user interface and judge its compliance with security and usability principles. *Interface* is regarded, in this thesis, as both software (e.g. user login Web page) and hardware components (e.g. authentication token as shown in Figure 2-13) toward- which the interaction/information transits between software and/or hardware components, networks, and users.

USS can be used to *guide a design decision* or to *assess a design* that has already been created. It integrates usable security earlier into the requirements and design phase which helps security designers make more informed and therefore better decisions, and influence the design in its early stage when traditionally the bulk of the feature design is done.

As USS provides very specific and practical review questions (not general ones), it is common to unfold issues and as well opportunities for feature improvement other than only those related to security and usability. This can be

⁷⁶ The Free Dictionary by Farlex. 2010. <<http://www.thefreedictionary.com/symmetr>>. Retrieved on July 29, 2010.

helpful when designing an authentication method and it is in fact much appreciated given that we are talking about a computer security application.

According to Nielsen (1992), usability specialists were much better than those without usability expertise at finding usability problems by heuristic evaluation. Moreover, usability specialists with specific expertise (e.g., security) did much better than regular usability specialists without such expertise, especially with regard to certain usability problems that were unique to that kind of interface. Thus, USS is developed as a usable security inspection method for system designers (acting also as evaluators) who have knowledge in computer security (especially user authentication), and a general knowledge of usability techniques and requirements.

5.4.2 Usable Security Protocol (USP) Sub-Methodology

The USS inspection method is the *sub-methodology* within the Usable Security Protocol Methodology (USP), therefore continuing on from the work done in Section 4.2 The USP Methodology, which details step by step how the USS is created and generated, including goals, logistics, and the content behind each step. Also it shows how the USP methodology brings together the cognitive and computer science approaches to finally generate, as the outcome, the design requirements inspection method tool for the design of user authentication methods: the Usable Security Symmetry.

5.4.2.1 Usable Security Symmetry (USS) Inspection Method

USS is a checklist-based inspection method. A checklist is a valuable evaluation method when carefully developed and applied. A robust evaluation checklist clarifies the criteria that as a minimum should be considered when evaluating something in a particular area; aids the evaluator not to forget key criteria; and finally enhances the assessment's impartiality, reliability, and reproducibility.

Another relevant benefit of employing checklists lies in the fact that they offer an organizational framework for quick recall of critical information and current best practices.

Such a checklist is useful in the authentication method life cycle process such as planning an authentication method, monitoring and guiding its operation, and assessing its outcomes.

Moreover, checklists are useful for both formative and summative evaluations (Stufflebeam, 2000). USS makes use of the *Formative evaluation*, which is a process of ongoing feedback on performance. The purposes are to specify aspects of performance that need improvement and to provide corrective suggestions. *Summative evaluation* in turn is a process of specifying larger patterns and trends in performance, and judging these review statements against criteria to get performance ratings.

A snapshot of the USS checklist short form is shown below. Depending on the evaluation been carried out, the checklist can be quite long, so that users should be able to collapse or expand each checklist item (e.g., # 1.3), thereby facilitating results data visualization (Table 5.3).

The long form checklist will show all rows expanded as can be seen in the next paragraphs.

Usable Security Symmetry Inspection Method

1. Usability Criterion: MINIMAL ACTION									
Capability of the application to help users achieve their tasks in a minimum number of steps (i.e. the length of transactions and procedures). It concerns perceptual and cognitive workload for individual inputs or outputs.									
#	Usability Review	Occurrence			Comments	Security Review			Comments
		Y	N	NA		Y	N	NA	
1.1	Can the user select the				Users could do	Is a single ID credential using a			-
1.2	Is the workload low and simple				-	Have strict password policies			Security policy
1.3	Is the authentication carried out				It seems pretty	Is authentication of principals to			-
1.4	Has the user to reauthenticate				This can take	Does the permissions strategy be			-

Table 5.3: Usable Security Symmetry inspection method - data visualization.

The *sub-methodology* is described as follows:

- *Project Lead activities:*
 - Identify and define the usability criteria that will be used to evaluate the authentication method.
 - Identify one (or up to 5) security designers and/or usability professionals to examine the system on an individual basis.
 - Gather materials that facilitate the evaluators to become familiar with the purpose of the system and of its users (e.g., system specification, user tasks, use case scenarios, etc).
 - Gather and analyze primary, secondary, and tertiary data available for building the inspection method.
- *Development activities:*
 - Develop the usability Review Questions in conjunction with the Occurrences.
 - Asking questions is a crucial component of finding information. The questioning method adopted for USS is a combination of Review and Survey Questions which has been named for the purposes of this thesis, a Review Question. A Review Question presents users with a question or statement to which they answer using a predefined set of scales (see *Occurrences* section below). The goal of the Review Question is to sum up and ask for agreement or otherwise.
 - The Review Questions should be developed by taking into account the following quality guidelines:
 - i) Ensure each Review Question is completed properly with the ultimate goal of providing usable security for user authentication.

- ii) Make certain that the Review Question is pertinent to each specific - usability criterion (e.g. Review Question: *Minimal action - Has the requirement of data entry by the user been limited when the data can be derived by the application or browser?*);
 - iii) Include the cognitive process involved, implicitly or explicitly;
 - iv) Identify particular error-prone areas (e.g. Review Question: *Minimal Action – Should PINs longer than 6 digits be avoided?*);
 - v) Utilize nomenclature well-known in the domain (Computer Security and Usability) for efficient communication;
 - vi) Consider the expertise of the Computer Security and/or Usability experts (i.e. target user).
- Occurrences are represented by the following letters: **Y** (Yes), **N** (No), and **NA**. **Y** (Yes) represents that the authentication method being reviewed for example complies with the Usability Review question; **N** (No) represents that the authentication method does not comply with the Usability Review question; and finally, **NA** (Not Applicable) represents that the Usability Review question doesn't apply for that particular authentication method. The default value for the Occurrences fields is empty (none).
 - Each Occurrence has a specific value field that has to be filled out with a symbol when applicable. For example if the authentication method does comply with the Usability Review question then a symbol has to be entered in the **Y** field, and- the

N and **NA** fields are left blank. It is important to note that the importance of visual effectiveness on the inspection method: The appearance of information and the use of visual characteristics can influence largely the effectiveness of the inspection method. Therefore, it is recommended to use a colored square in the Occurrences value fields for easier data visualization: green for **Y**, red for **N**, and gray for **NA**. For visually impaired evaluators some alternatives are the use of gray scales, check marks, or even numbers.

- Create a Comments column. Include in this column any comments needed. If there are no comments, include a dash “-” (or any other sign meaning “no data”) within the respective field since leaving it blank may mislead evaluators into thinking that data are missing.
- *Evaluating the system:*
 - Review the materials provided to familiarize with the system design. If possible carry out the user actions that will be taken to perform the user tasks.
 - Identify and list any areas of the system that might be opposed to the usability principles. List all of the concerns that you note in the Comments fields.
- *Showing the outcome:*
 - The main outcome of the USS is a list of usability and security problems in the interface with references to those usability criteria and security aspects (See Sections 5.4.2.2.1.1 Usability Severity Ratings and Recommendations for MTM Study Case and 5.4.2.2.2.1 Security Severity Ratings and Recommendations for MTM Study Case).

- *Analyzing the results:*
 - Review each of the concerns that have been written down in the checklist evaluated.
 - Assess and judge each concern for its compliance with your defined criterion.
 - Allocate a severity level for each grouped concern based on the impact to the end-user (See section 5.4.2.2 Severity Ratings).
 - Establish recommendations to fix the problem. Ensure that each recommendation relates to the criterion.
 - If needed, finally, use a holistic approach to review all recommendations and verify that there are no major (or none at all) conflicts among them. If any, make an assessment and implement only those recommendations that address high-priority functional system requirements as defined by your stakeholders.

The output of the USS inspection method is a list of usability and security problems and their severity ratings (and their recommendations), which will be described in the next paragraphs. But before addressing the severity ratings, the USS inspection method checklist is shown in Table 5.4. Usable Security Symmetry checklist.

Abbreviations used in the inspection method:


AM = Authentication Method.

Table 5.4: Usable Security Symmetry checklist.

Usable Security Symmetry Inspection Method

1. Usability Criterion: MINIMAL ACTION

Capability of the application to help users achieve their tasks in a minimum number of steps (i.e. the length of transactions and procedures). It concerns perceptual and cognitive workload for individual inputs or outputs.

#	Usability Review	Occurrence			Comments	Security Review	Occurrence			Comments
		Y	N	NA			Y	N	NA	
1.1	Does the system employ multiple-technology ⁷⁷ contactless technology ⁷⁸ for physical and logical access ⁷⁹ applications?				Currently the system adopts a contact card. All-in-one contactless technology card would be nice to have. Very convenient to users struggling with several debit, credit, and other plastic cards in their wallets.	For stronger security, has a 3 rd -factor authentication introduced (e.g., biometrics)?				Hand geometry is used for authentication in conjunction with the smart card plus PIN.

⁷⁷ Multiple-technology enables a card to be produced with contact or contactless smart chip technology, magnetic stripe, bar codes, optical stripe and/or 125 kHz proximity antenna. A card containing several types of read/write media is generally called a multiple technology card.

⁷⁸ Contactless technology offers reliable and fast throughput. If another authentication factor is introduced the throughput advantages offered by contactless technology are decreased, but the strength of security and authentication is increased.

⁷⁹ Logical access refers to the connection of one device or system to another through the use of software. The software may run, say as the result of a user powering a PC, which then executes the login sequence, or it may be the result of internal processing between systems.. Whereas physical access is being able to physically touch and interact with computers and network devices.

Physical access:

However plastic cards will be soon obsolete since contactless technology is growing with a variety of form factors (e.g. a mobile phone). Some technology analysts such as Penn (2008) from Forrester Research say that within five years, mobile phones in US will be able to make electronic payments, open doors, access subways, clip coupons and act as another form of identification.



Logical access:
Currently, contact technology provides a convenient and cost-effective way to transfer significant amounts of data between a card and a reader/host system quickly and perform complex cryptographic operations for authentication applications. For these reasons, contact smart cards are a prominent solution for network security implementations.

1.2	Is the workload low and simple (e.g., input workload kept to a minimum)?		There is a lot of user interaction when dealing with the different MTM functionalities. Perhaps defining a simpler feature set can mitigate this issue.	Have strict password policies been established (i.e., for 1-factor authentication like password, a strong password policy must be set up)?		It also depends on the use of MTM other features which will require 1 st level or 2 nd level authentication.
1.3	Is the authentication (verify/authenticate the identity of the user) process simple to users?		If there's only 1st level authentication (smart card and password); if users need more privilege or access critical services it may need a 2 nd level authentication (e.g., hand recognition)	Is authentication of principals to resources required?		---

1.4	Has the user to reauthenticate when effectuating other transactions than financial (e.g. buying a concert ticket with a credit card)		Reauthentication is cumbersome to users.	In a reauthentication scenario, are the different AMs integrated?		It needs to be implemented.
1.5	Has the user to reauthenticate when accessing information from other channels?		This is really bothersome and bad user experience to users.	Is security being varied with the task itself ⁸⁰ ?		This also depends on integrating a RBAC as one of the software functionalities. Software includes access control mechanism which grants and revokes privileges based on predefined rules.






⁸⁰ For example, different kinds of security measures are called for protecting information that is in transit (128-bit AES encryption using an ephemeral encryption key that is destroyed when it is no longer used) VS. information that is to be stored permanently on a hard drive (128-bit AES encryption for storing documents using key escrow, secret splitting, or even a secondary encryption key so that the document's content can be recovered in the event of a problem).

1.5.1	From Web-based application?		Users have already logged into the MTM with the conventional smart card + PIN. Using another AM (e.g., OTP) really becomes a heavy workload to users.	Is strong authentication used with at least two-factor authentication?		Smart card plus PIN: smart cards require a PIN so they add a 2 nd layer (smart card/PIN in place of a password) that an impersonator would have to obtain to log onto a system.
1.5.2	From Electronic Purse (EP)?		It seems a little wearying to users carry out another plastic card but EP is less bulky than cash. However EP is very convenient to users.	Is strong authentication used with at least two-factor authentication?		EP is more secure than cash since they can be "closed" by a single key stroke and reopened by entering a four-digit PIN. However the advantage of the EP is really in its comparison with cash: once users have obtained their cash from a MTM, anyone can spend it.

1.9	Are the items displayed succinctly (i.e., shorter reading times, smaller errors)?		---	Are required items associated with sensitive information kept to a minimum?		Basically username and email address are required.
1.10	Is the password usable (i.e., meaningful and concatenate /interspersed words w/ characters)?		This depends on a predefined password policy. Users can choose a meaningful and concatenate password as long as it respects the standard password policy.	Do passwords follow strict security policies?		Password policy requires a strong password which must contain at least eight characters, one uppercase alphabet (A-Z), one lowercase alphabet (a-z), one Arabic numeral (0-9), one non-alphanumeric character excluding @ ~.
1.11	Is a particular cue given to users for remembering passwords?		---	Are textfield' postlabels that provide tips for remembering passwords and PINs avoided?		When creating a password, a link is displayed such as <u>What's a valid password? next</u> to the password field. When users mouseovers it a pop window



1.12	Is Single Sign-On login system used?		As mentioned, memorability is further improved by the use of mnemonics to aid their recall. Interoperability is very difficult for now.	Are strong passwords used?			shows up and display the password policy. Sometimes a reminder of what the policy is displayed in single one sentence next to the textfield.
							When users log into the application to buy a concert ticket which requires 2nd-level authentication, they will first need to complete 1st-level authentication as they normally do by using their smart card and PIN. After successful first-level authentication,

1.14	Does the system speed data entry by setting default values?		Although default values are different than using a cookie for username and password it is somewhat related to 1.14	Are default values avoided especially for sensitive information (e.g. username and password)?		Although default values are different than using a cookie for username and password it is somewhat related to 1.14
1.15	Does the system accommodate both experienced and novice users (e.g., short cuts are available to experienced users)?		After entering PIN at the MTM expert users generally click directly on the "Fast Cash" (short cut) button instead of "Enter" to complete the login process. Novice users usually click on "Enter" first.	Are succinct error messages displayed for novice users avoiding serious errors?		Succinct error messages are displayed for both novice and expert users.
1.16	Are PINs longer than 6 digits avoided?		4-digit PIN.	Does the system use weak PIN change protocols?		Users circumvent PIN forgetfulness through insecure behaviors (e.g., writing down their PINs, make them all the same, or disclose them to friends and family).

1.17	Are PINs employed mostly for frequently used systems (infrequently used PINs are the ones that are most often forgotten)?		4-digit PIN is used.	Are smart cards used in conjunction with PIN to provide strong authentication?	
1.18	Does the smart card provide interoperability across services?		Users sometimes get lost with so many options and also different levels and types of authentication.	Has the user the option to choose to use just one PIN or separate PINs for each application?	 PIN is required only once when users first login at the MTM.
1.19	If users must switch between different systems are dual-interface ⁸² chip smart cards ⁸³ used?		Users switch to Web applications not from the card.	Can card issuers record and update appropriate privileges from a single central location?	 ---

⁸² Single card solution for contact and contactless applications.

⁸³ A microcontroller chip card (e.g., a card that opens a door by just bringing the card close to the card reader and that same card also provides logical access to computers and networks).

1.20	For physical and logical access ⁸⁴ applications, does the system use multiple-technology ⁸⁵ contactless technology ⁸⁶ ?		Currently the system adopts a contact card. All-in-one contactless technology card would be nice to have. Very convenient to users struggling with several debit, credit, and other plastic cards in their wallets.	For stronger security, has a 3 rd - factor authentication introduced (e.g., biometrics)?		Hand geometry is used for authentication in conjunction with the smart card plus PIN. <i>Physical access:</i> However plastic cards will be soon obsolete since contactless technology is growing with a variety of form factors (e.g. a mobile phone). Some technology analysts such as Penn (2008) from Forrester Research say that within five years, mobile phones in US will be able to make electronic
------	--	---	---	---	---	---



⁸⁴ Logical access refers to the connection of one device or system to another through the use of software. The software may run, say as the result of a user powering a PC, which then executes the login sequence, or it may be the result of internal processing between systems.. Whereas physical access is being able to physically touch and interact with computers and network devices.

⁸⁵ Multiple-technology enables a card can be produced with contact or contactless smart chip technology, magnetic stripe, bar codes, optical stripe and/or 125 kHz proximity antenna. A card containing several types of read/write media is generally called a multiple technology card.

⁸⁶ Contactless technology offers reliable and fast throughput. If another authentication factor is introduced the throughput advantages offered by contactless technology are decreased, but the strength of security and authentication is increased.

payments, open doors, access subways, clip coupons and act as another form of identification.

Logical access: Currently, contact technology provides a convenient and cost-effective way to transfer significant amounts of data between a card and a reader/host system quickly and perform complex cryptographic operations for authentication applications. For these reasons, contact smart cards are a prominent solution for network security implementations.


1.21	Does the system provide the capability to users to be authenticated by group/role (e.g. a group of users (two or more) can authenticate with the same credentials)?		It needs system review to evaluate the feasibility of this functionality.	Is the Role-Based Access Control (RBAC) ⁸⁷ used to manage users who has signed up for this service?		RBAC diverges from Access Control Lists (ACLs) used in conventional discretionary access control systems in that it assigns permissions to certain groups to specific operations.
------	---	---	---	--	---	---

⁸⁷ Role-based access control (RBAC) is a general security model that simplifies administration by assigning roles to users and then assigning permissions to those roles.







Usable Security Symmetry Check-List

2. Usability Criterion: OPERABILITY

Amount of effort necessary to operate and control an application

#	Usability Review	Occurrence		Comments	Security Review		Occurrence		Comments
		N	NA				Y	N NA	
2.1	Can (some) system's mechanisms ⁸⁸ be configured by users to operate in certain way (e.g., select a preferred authentication method)?			Having the possibility of select what authentication method users would like to make use of is still in its infancy. Organizations are just starting to offer different ways of authenticating to a system but this is taking place in corporate environments (e.g. usually authentication options are given to system administrators). Consumers cannot select their preferred authentication mechanisms so far.	Does the system offer different combinations of authentication methods to users that includes strong authentication (e.g. OTP plus biometrics, or contactless smart card plus password)?				---

⁸⁸ For customized applications (personalization applications) in order to enhance customer experience reducing transaction speed while at the same time adding value.

2.2	Can users customize the user interface to their specific needs (e.g., personalized look and Feel) at the MTM?		It needs to be implemented.	Can users customize only NO sensitive information and security-related content items?		---
2.3	If a variety of authentication methods is required, is that the simplest ones?		---	Is 3-rd authentication factor required to protect access to sensitive applications (e.g., biometrics)?		When making an international funds transfer above \$5,000 at the MTM.
2.4	Is the biometric method usable and easy to use?		Palm recognition.	Has the chosen biometric method the lowest False Reject Rate (FRR) (i.e., a legitimate user is rejected by the acquisition device?)		Users like palm and fingerprint recognition because they are not intrusive, reliable, low cost, and only have a 0.1% FRR which is one of the lowest in the biometrics industry.

2.5 Is the user task' workload light?

Relatively simple for the corporate/consumer computer average user. As mentioned the system uses smart card plus PIN (1st level authentication) then 2nd level username/password , hand recognition, or OTP.

Is the user authentication flow simple to avoid dangerous behaviors (e.g., sensitive data leakage)?

Generally speaking smart card plus PIN takes 2 steps. For Username/ Password, according to the GOMS analysis developed in this thesis, there are precisely 19 steps (24.40 seconds comprising cognitive and motor processes). However the core process takes 3 steps that is user inputs username, then password and finally clicks on "Submit" button. For OTP, according to the GOMS analysis, precisely 21 steps (28.13 seconds comprising cognitive and motor processes). However the core process takes 4

steps that is user inputs PIN, then gets tokencode, then appends tokencode to the PIN, and finally clicks on "Submit" button. For Hand Palm Recognition (pretty similar to Fingerprint), according to the GOMS analysis, there are precisely 7 steps (9.16 seconds comprising cognitive and motor processes). However the core process takes 2 steps that is user places her hand palm over the palm sensor, then gets authenticated by the system.

2.6	Is the transactions' sequence smooth when varying levels of access?		---	When varying levels of access, is authentication or user validation required?		---
2.7	For logical access (i.e. access to MTM software and Web applications), are there very few varying levels of access?		But it depends on which services users are accessing. For example buying a concert ticket requires going online whereas using a check image feature is a local service and does not need another level of authentication.	Have the system considered a high security-computing environment (i.e., this is required for Public Key Infrastructure (PKI) and/or biometrics services)?		<p>Users access services from the MTM and from external applications, mostly e-commerce ones. In order to run Web applications based on MTM a real-time Web service structure.</p> <p>Also as people are more concerned with privacy risks in Biometrics systems and MTM does make use of network and distributed systems, so that the use of encryption is critical. The security of the biometric system relies on the integrity and authenticity of the biometric information, which can be accomplished using PKI once the individual has been enrolled.</p>

2.8	For biometrics, are local and remote accesses available to user authentication (i.e., if on-line systems are down during an outage or other emergency)?		---	Is the local access set up to default (i.e., no need to be transmitted over the network)?		It needs to be implemented.
2.9	Is enrollment data maintained centrally or locally (i.e. user convenience - a key issue in selecting card issuance procedures)?		Users should enroll once and this be served as the basis of future authentication across a range of systems.	Is a smart card used to store user's biometric data including user profile and enrolment?		It needs to be implemented. Although users may see the benefit of leveraging an existing enrolment as mentioned in the 2.9 Usability review in comments (on the left), they may be suspicious of potential data misuse.
2.10	Is biometric authentication used (i.e., more user convenience)?		---	For a transaction with high risk of hacker attack on the system, is biometrics used?		Man-in-the-Middle (MITM) attacks using biometrics are more difficult to go through than the other AMs.





2.11	If biometrics adopted, is hand geometry authentication used (i.e., ease-of-use, non-intrusive)?		Hand palm recognition.	For stronger security, is hand geometry authentication method used (i.e., more accurate and secure than fingerprint biometric method ⁸⁹)?		More secure if used in conjunction with other technologies like KBA (e.g. password and PIN) on the Internet.
2.12	If hand geometry is used, are the False Reject Rate (FRR) and False Acceptance Rate (FAR) balanced to provide more convenience to users?		It needs to be implemented.	Has the hand geometry authentication <u>FRR about 1%</u> and the <u>FAR about 0.1%⁹⁰</u> ?		Currently the system presents a FRR of 5% and a FAR of 3%. It needs to be implemented.
2.13	Is there an alternative authentication method, when biometrics is not available in order to provide ⁹¹ availability to the users?		OTP AM.	Is there an alternative authentication method like token, when biometrics is not available?		OTP AM.

⁸⁹ Biometrics-Based Web Access study. Retrieved on January 3, 2009







<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.27.7828&rep=rep1&type=pdf>>

⁹⁰ FRR= False Rejection Rates: the likelihood that a legitimate person will be denied access; FAR= the likelihood that the wrong person will be able to access the system. The FRR ranges from 1 to 20% whereas the FAR ranges from 0,001 to 5%.

⁹¹ There could be a number of potential privacy, personal, religious, cultural, and legal issues associated with the use of a biometric, and also persons with disabilities.

2.14	If users forget their PINs, can users reset them via a Web interface (i.e., ensure customer convenience and satisfaction) rather than in an issuance station?		Users need to visit a bank branch in person to reset their PINs.	If users forget their PINs, can they reset them via a Web-based portal ⁹² ?		Not possible according to bank security policy. But this policy will need to change in the near future in order to provide convenience and stronger security to users. To this end, a contactless technology is needed.
2.15	If biometrics adopted, are mechanisms and software used the most recognized in the industry to facilitate the biometrics process close to users?		---	Does the biometric solution follow applicable standards (i.e., many biometric solutions use their own proprietary algorithms & processes)?		---

⁹² Using this approach, the user can authenticate to the web site and then navigate to a PIN reset screen where the old PIN is required and validated using the rules set on the smart card during the chip personalization process.

2.16	Has the Mean Time Between Failures (MTBF) the lowest rate possible improving user experience ⁹³ ?		It needs to be implemented.	Is the MTM system (components and coupling) designed to be as simple as possible ⁹⁴ ?		---
2.17	Has the Mean Time To Repair (MTTR) ⁹⁵ smaller than an hour improving user experience and availability of services?		It needs system review to be able to implement it.	Is the MTM error rate of <i>simple processing errors</i> ⁹⁶ of one error per 100,000 transactions?		It needs system review to be able to implement it.
2.18	Is the Total Transaction Time (3Ts) to handle a biometric AM equals to minimum 4 and maximum 10 seconds for a single user?		Currently hand geometry takes 15-20 seconds.	Is the biometric template maintained in a smart card ⁹⁷ providing highly secure and portable authentication of the cardholder's identity?		Biometric template is kept in a central repository, mirrored in distributed data bases.

⁹³ The estimated length of time that a system is available and operational between failures. It is a measure for hardware reliability.

⁹⁴ Generally speaking, if the probability of errors augments with the level of complexity, so then the probability of errors that can cause security breaches. As a result, not only is malfunction unavoidable, but the probability that at least one exploitable security hole exists is unavoidable. In a nutshell, multiple components can intermingle in unexpected ways to bring down a relatively fault-tolerant system.








⁹⁵ The estimated length of time needed to bring a system back up and make it fully operational following a system failure.

⁹⁶ A bank's MTM sends a transaction again if the network went down before a confirmation message was received from the mainframe.

⁹⁷ It can be used to authenticate the identity of the cardholder by matching a live scan of a biometric feature (e.g. fingerprint or iris scan) to the template on the card.

Usable Security Symmetry Check-List




3. Usability Criterion: <i>PRIVACY</i>									
Whether users' personal information is appropriately protected									
#	Usability Review	Occurrence			Comments	Security Review			Comments
		Y	N	NA			Y	N	NA
3.1	If smart card used, who owns the personal data stored on it?				The MTM's owner which in this case is a bank. It might be a loyalty program provider including healthcare, entertainment and transportation.	Is the user the "owner" of personal information (user is responsible for keeping personal information on the card up-to-date)?			It needs system review to be able to implement it.
3.2	Does the system owner provide easy access to the privacy policy?				It needs to be implemented.	Has information about privacy protection been included in the privacy policy (card acceptance agreement)?			---
3.3	Does the system provide the user with the right of access to the information and a process for correcting errors?				---	Is the owner's system responsible for the security and accuracy of information?			---
3.4	Are users in control of their private information?				It needs to be implemented.	Is the system application complies with State or Federal laws (e.g., regulation E, State Privacy Acts, etc. in U.S.)?			But owner needs to do an update regarding current or potential laws.

3.5	If the card is lost, can the user go to a single location to repopulate the card?		---	Is there a client registry that provides pointers to all application owner databases for all applications active on the card ⁹⁸ ?		It needs system review to be able to implement it.
3.6	Does the system owner provide integration of multiple databases ⁹⁹ ?		It needs system review to be able to implement it.	Does the system owner create “shadow” files of all transactions and route these at least daily to the application owner’s remote database?		It needs system review to be able to implement it.
3.7	Is the cardholder privacy maintained (i.e. security of the information of the ID credential)?		PKI is used (encryption)	Is “Match on-card” technique used ¹⁰⁰ ?		It needs system review to be able to implement it. It is more portable.
3.8	Is the user assured about transactions made related to loss of privacy/security for payments (transactions are made on the Internet using the MTM)?		But the system needs some improvement in this area (review and updates).	Does the system provide private and secure payments for Internet purchases?		---

⁹⁸ The card issuer uses the client registry to determine which applications are active and queries the application owner for the client backup database in the case of card replacement.

⁹⁹ For example, the contents of the physical access control privilege database and logical access control privilege database can be combined into a single integrated DB maintained as part of the card management system.

¹⁰⁰ This technique stores the enrolment template into the smart card’s secure memory (i.e. the information is secured both during collection and during use of the credential in the ID system).

3.9	Do security protocols adapt their encryption policies based on the content of the data being encrypted (e.g., video encryption algorithms ¹⁰¹)?		It needs system review to be able to implement it.	Are encrypted tunnels ¹⁰² or virtual private networks supported to provide confidential access over insecure wireless links?		---
3.10	Are users insecure and reluctant about their sensitive information (i.e., system have to be aware about the impact of application/network on remote end-users' experiences, and uncover and resolve problems before rollout)?		Especially because is a MTM no matter how much secure it is. But this is changing since MTMs provide almost the same user experience/basic functionalities as a computer.	Does the system provide data corruption ¹⁰³ , detection and prevention, and backups for databases)? For example, tamper-evidence for digital signature generation ¹⁰⁴ or transaction logging, auditing remotely for MTM journal of transactions).		But needs system review.

¹⁰¹ Video encryption algorithms focus on protecting the more important parts of a video stream, thereby reducing the total amount of data encrypted and providing a faster response time which enhances the end-user experience.

¹⁰² Not susceptible to "sniffing" (i.e., a sniffing attack occurs when an attacker gains access to the network TCP/IP traffic path, captures data packets that make up the conversation, and assembles the packets into a format readable to the attacker). For example, the privacy of each user's credit card numbers is crucial. Although the Internet is by no means armored, the most likely location for the loss of privacy to occur is at the last points of the transmission.

¹⁰³ Data corruption is the deterioration or damage of computer data caused by human, hardware and software error.

¹⁰⁴ "Tamper-evidence for digital signature generation" can be used to defend against attacks aimed at covertly leaking secret information held by corrupted signing nodes (<http://www.informatics.indiana.edu/markus/papers/TESig.pdf>).

Usable Security Symmetry Check-List

4. Usability Criterion: *SECURITY*

Capability of the application to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access

#	Usability Review	Occurrence		Comments	Security Review		Occurrence		Comments
		Y	N	NA			Y	N	NA
4.1	If logical access is provided, and users e.g., forget their password, is there any mechanism to avoid the system's blockage after 3 failed attempts to log on ¹⁰⁵ ?				It needs system review to be able to implement it.	Is logical access provided for local & remote networks prevent unauthorized programs/users from accessing confidential data ¹⁰⁶ ?			---
4.2	When using different communication channels, is PIN authentication used (i.e., when accessing MTM, Web, and WAP: PINs are easier to remember)?				---	Is 6-digits PIN used (i.e. PINs have lower level of security since the number of possible combinations is lower ¹⁰⁷)?			It needs system review to be able to implement it.
4.3	When using KBA over the telephone, is PIN used instead of a strong password (i.e. PIN is easier to remember than password)?				It needs system review to be able to implement it.	Are PINs avoided to be used as the only authentication mechanism?			---

¹⁰⁵ If a cardholder blocks their card by entering an invalid PIN, the cardholder should have the capability to unblock the card. When the cardholder's card is initially setup, a special unblock code should be generated, encrypted and then stored in the card management system.

¹⁰⁶ For example, private keys or confidential information in databases.







¹⁰⁷ Long PIN's give stronger security, but bad usability because the PIN is harder to remember and takes longer to type.

4.4	Can the customer make use of multifunction smart cards which provide substantial convenience for them?			---	Are financial and security applications placed in separate card platforms ^{108,?}		It needs to be implemented.
4.5	For logical access application, is biometrics provided enhancing usability (i.e., on-card biometric match and on-card key generation)?			It needs system review to be able to implement it.	For logical access application, is contact technology (e.g. contact cards) provided for network security implementations ^{109,?}		---
4.6	Does the user trust the MTM's system owner?			See also # 3.10	Does the cryptography employ algorithms approved under Federal Information Processing Standard 140-2?		It needs system review to be sure.
4.7	Have users more than a few alternatives to authenticate to the system improving availability and convenience of the system?			---	Does the system use a higher end card to support digital signatures and/or biometric capabilities ^{110,?}		---

¹⁰⁸ The combination of financial and security applications raises potential security risks and interoperability issues that must be addressed in a multi application environment.

¹⁰⁹ It is a convenient and cost-effective way to transfer significant amounts of data between a card and a reader and host system, and to perform complex cryptographic operations for authentication applications. Furthermore, contact chips have microcontrollers while contactless chips may or may not. For these reasons, contact smart cards have been an outstanding solution for network security implementations.

¹¹⁰ With enough memory and/or a cryptoprocessor

4.8	Do users have access to multiple authentication technologies such as PKI and biometrics?		---	Is PKI used (i.e., secure digital certificates for logical access control, remote access and encryption)?		---
4.9	For the transmission of credentials over the phone, is the traffic passing over a telephone network encrypted ¹¹¹ increasing user satisfaction and trust in the system?		---	Is an additional authentication method used in conjunction with the PIN over the telephone providing thus a strong authentication?		It needs system review to be able to implement it.
4.10	Are users able to perform wireless transactions obtaining in this way mobility, connectivity and ubiquity capabilities?		Not completely. It needs system review to be able to implement it.	Is JavaScript used for wireless applications (e.g., malicious Wireless Markup Language Script (WML) scripts have the ability to off-load money from smartcards) ¹¹² ?		It needs system review to be able to implement it.

¹¹¹ This would provide a higher level of security but lower system response time and thereby usability. The risk of eavesdropping the telephone network is a real threat, especially since it cannot be encrypted.

¹¹² Here the attacker has the ability to charge purchases in real time to e-cash stored on the phone's smart card (e.g., purchasing concert tickets).

Usable Security Symmetry Check-List





5. Usability Criterion: <i>LOAD TIME</i>									
Time required for the application to load (i.e., how fast it responds to the user).									
#	Usability Review	Occurrence		Comments	Security Review	Occurrence		Comments	
		Y	N	NA		Y	N	NA	
5.1	Is the computation imperceptible to the user ¹¹³ ?			---	Is PKI used when a high percentage of users perform monetary transactions?				---
5.2	When the computation takes longer, is a progress indicator ¹¹⁴ used if it is not possible to mask it?			It needs system review.	To reduce the workload of a security protocol, are lightweight cryptographic algorithms for various security functions used (e.g., SSL)?				---
5.3	Is the speed of transaction time-consuming?			But there's some room for improvement. A system review should be done.	Is PKI used when a high percentage of users transmit and/or receive data across open networks?				---
5.4	Does the system use Java Card technology ¹¹⁵ in order to have the response time ¹¹⁶ reduced?			---	Is the adopted smart card a high performance one to reduce latency when executing the encryption algorithm?				It needs to be implemented.

¹¹³ For example, the Secure Socket Layer (SSL) protocol requires processing time for encryption and authentication. This cost is easily masked by the latency of loading web pages.

¹¹⁴ The full length of an operation can be determined and you can tell the user how much of the process has been completed.



¹¹⁵ Java Card Technology, Sun Developer Network. Retrieved on August 2nd, 2009 <<http://java.sun.com/javacard/>>

¹¹⁶ The time to read data from and write data to the chip has been reduced substantially.

5.5	If imaging technology is used, is the processing time required to scan a live image (e.g., a check) less than 1 second?		---	Has an adequate period of time been allotted for providing encryption capabilities?		---
5.6	Does the live image require a small memory allocation which will affect time for retrieval of the image by the user?		---	Does the system provide dynamic memory allocation ¹¹⁷ ?		---
5.7	Are the UI screens attractive and easy on the eye?		There are some areas for improvement.	Does the system make use of a Windows-based platform ¹¹⁸ to get the best character recognition software?		System has Windows-based software.
5.8	When using check imaging technology (i.e. digitally imaged checks), if the scanner rejects the inserted check into the MTM's check slot because handwriting is illegible can the user do by the regular way at the MTM without fees?		It sends check images or information electronically to the systems owner. The conventional deposit is available at the MTM as well.	Has the check imaging process been integrated to the system's owner IT infrastructure?		It needs to be implemented.

¹¹⁷ The dynamic memory allocation feature can maximize the concurrent execution of memory allocation routines and also optimize the allocation of both large and small blocks of memory.

¹¹⁸ Software package optimized for text acquisition with hand-held and flatbed scanners.

5.9	Has the customer at least three opportunities to enter the PIN ¹¹⁹ (i.e., limited number of login attempts)?		---	Does the system is blocked after three PIN failed attempts to access the system?		---
5.10	If login fails (e.g., users forget their PIN) is there another authentication option to login to the MTM?		For security reasons, there's no other option for now. But another option should be provided.	If login fails, does the system avoid indicating what part of the authentication information that was incorrect ¹²⁰ (i.e., no error feedback)?		But system needs update with current best practices.
5.11	Does the system's owner make use of the image quality and usability image components in check imaging?		It needs to be implemented.	Does the system provide transactional integrity?		It requires system review to be able to implement it.
5.12	Is Data Encryption Standard (DES) ¹²¹ encryption avoided although provides virtually no latency delays and minimal response time impact compared to Triple-DES (3DES)?		DES encryption is used though less secure. Usability advantages of using 3DES correspond to the Administrator, not End-User level (See 5.4.2.2.1 Usability Severity Ratings, Problem Description 29. Privacy and Integration.).	Is Triple-DES (3DES) ¹²² encryption employed providing thus greater security?		It needs to be implemented.
5.13	Is extensive automatic logs ¹²³ avoided?		---	Does the system provide auditing capabilities?		---

¹¹⁹ If the PIN is entered incorrectly several times in a row (usually three attempts per card insertion), the MTM will attempt to retain the card as a security precaution to prevent an unauthorised user from discovering the PIN by guesswork. However, it reduces usability since users who have difficulties remembering their password, might be locked out of the system.

¹²⁰ To reduce risk of successful intrusion attempts.











¹²¹ Data Encryption Standard (DES) algorithm is based on work by IBM and was published as a federal standard in 1977. For over twenty years, DES has been subject to heavy examination, and there are no known algorithmic flaws.

¹²² Triple DES (or TDEA Triple Data Encryption Algorithm) utilizes a 168 bit key. FIPS 46-2 standard: Triple DES will be the FIPS. Advanced Encryption Standard (AES) is the replacement for DES (NIST_01).

Usable Security Symmetry Check-List

6. Usability Criterion: MINIMAL MEMORY LOAD

Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task.

#	Usability Review	Occurrence			Comments	Security Review			Occurrence			Comments
		Y	N	NA		Y	N	NA	Y	N	NA	
6.1	Is the memory load on the user minimized (i.e., no memorization of long data lists, complicated procedures, or undertake complex cognitive activities)?				---							---
6.2	Are the entries short (i.e., STM capacity is limited ¹²⁴ , so the shorter the entries, the smaller errors and reading times)?				---							---
6.3	Are short PINs used such as four digits or less (i.e., they are easier to memorize and fast to type)?				---							---
6.4	Is a <i>system-defined</i> PIN avoided (i.e. more difficult to memorize since it has no meaning and cannot be pronounced)?				---							---
6.5	Is the MTM's application based on recognition of visual items for authentication (i.e., to avoid unaided recall)?				It requires system review if it is feasible.							---

¹²³ Automatic logs can be used with software-based products to keep a record of all user interactions and transactions but they can have the disadvantage that they will also pick up accidental actions.







¹²⁴ The capacity of STM is usually limited to 7 ± 2 items (e.g. letters, digits, words, etc.).

¹²⁵ For example, for reasons of data protection, it may not be desirable to display passwords and other secure entries.

6.6	Is “time-to-learn” the silent alarm feature (almost) equals to zero so then users can intuitively send a silent alarm?		---	Is the alarm system reliable (i.e., nuisance alarms ¹²⁶ , false alarms, or fails to alarm when called for)?		---
6.7	Are the sequences and interdependencies of the MTM’s artifacts and their corresponding UIs (i.e., streamlined business workflow) “harsh-less” in the user interaction standpoint enhancing thus customer intimacy (providing appropriate choices, information and advice)?		But there are some areas for improvement. For example, the number of AMs should be reduced. This probably depends also on the interoperability of the system and applications.	Does the system’s owner provide enhanced process control and operational risk mitigation regarding the MTM’s business process?		---
6.8	Are <i>system-defined</i> PINs avoided (i.e., no meaning to the user and harder to remember than a password)?		---	Does the system require an additional and faster authentication method to be used in conjunction with user selected PIN?		It needs to be implemented.
6.9	For codes longer than 4 or 5 characters, are mnemonics or abbreviations being used?		Memorability is further improved by the use of mnemonics to aid their recall. However this is not the case.	For PINs and passwords are mnemonics or abbreviations being avoided?		4-digit PIN.

¹²⁶ Nuisance alarms occur when an unintended event evokes an alarm status by an otherwise properly working alarm system.


Usable Security Symmetry Check-List

7. Usability Criterion: RESOURCE SAFETY									
Whether resources (including people) are handled properly without any hazard.									
#	Usability Review	Occurrence		Comments	Security Review	Occurrence		Comments	
		Y	N	NA		Y	N	NA	
7.1	Does the system's owner provide MTMs inside supermarkets (i.e. MTMs inside busy supermarkets are considered safer)?				---				---
7.2	Are users updated concerning of the owner's system security services to better meet customer needs?				It needs to be implemented.				---
7.3	If the physical security of the customer is threatened (e.g., steal a customer's card/money), can the user send a silent alarm ¹²⁸ to the owner's system?				---				With the new MTM card reader slot users only need to insert partially the card in the slot and remove it rapidly. The MTM does not keep the card inside the machine anymore during transaction processing.

¹²⁷ Applications are loaded on an as-needed basis. The system needs to be able to add additional applications to the card platform.

¹²⁸ A silent alarm that makes no noise that is audible to the trespasser and uses an emergency PIN system. The alarm makes an audible noise elsewhere and notifies the owner's system (e.g., bank) and/or the police. For example, a situation might be a customer who is forced by a robber to withdraw money from the MTM. So, s/he can tap (or speak) an emergency PIN alerting the bank's security system.

¹²⁹ When the customer gets frustrated by not getting the card back and walks away from the machine, the criminal is able to remove the card and withdraw cash from the customer's account.

7.4	If the MTM offers a silent alarm feature, is Reverse PIN used providing faster recall?			It requires system review if it is feasible.	Has the system's owner security policy a clause that states that reverse PIN must only be used for emergency?		It requires system review if it is feasible.
7.5	Are audible instructions (or voice PIN) available so that people who cannot read an MTM screen can independently use the machine ¹³⁰ ?			---	Is PIN entered on public keypad then transmitted to central server and checked against saved file avoided ¹³¹ ?		PIN is stored and checked against the smart card and not transmitted over the network where it would be susceptible to "sniffing" (shoulder surfing or direct observation by cameras).
7.6	Does the MTM hardware and aesthetics convey a secure and trusted operating machine to users?			---	Is the outside of the MTM tamper evident (i.e., it makes unauthorized access to the protected object in the MTM easily detected)?		It requires system review for keeping up with new types of attacks.
7.7	Have users the ability to send a silent alarm through their wireless devices due to MTM's system failure?			It needs to be implemented.	Are attacks prevented on the wireless device that hosts the MTM's emergency application?		It requires system security review to be able to implement it.

¹³⁰ All audible information is delivered privately through a standard headphone jack in the front of the machine. Information is delivered to Postal kiosk.

¹³¹ The risk of eavesdropping or replaying a PIN depends on how and where it is entered to the system. This option is not good from the security perspective, since the PIN has to travel over the network anyway.

5.4.2.2 Severity Ratings

The usability problems can greatly be eliminated or reduced through the severity rates where evaluators are able to identify the security and usability problems that should be tackled and fixed. The ratings also aid in the allowance of resources for treating the user interface problems. If the severity ratings indicate that several disastrous usability problems remain in an interface, it will most likely be unadvisable to release it. However one might choose to go forward with the release of a system with several usability problems if they are all judged as being cosmetic in nature.

According to Nielsen (1994), the severity of a usability problem consists a combination of three elements: frequency ranges (i.e. from ordinary problems to atypical ones), impact (i.e., establishes the ease or difficulty with which users recover from a problem), and persistence (i.e., ranges from just one problem that might be surmounted to the problem that continually replicate itself becoming bothersome to users).

Last but not least, it is recommended to assess the *market impact* of the usability problem since certain usability problems can have a destructive effect on the popularity of a product, even if they are fairly easy to overcome.

Severity ratings are gathered in a questionnaire after the actual evaluation session, listing the entire set of usability and security problems that have been discovered, and the evaluator is asked to rate the severity of each problem. The descriptions are synthesized by the evaluator from the comments made for each problem. Typically, the evaluator needs only spend about 30 minutes to give their severity ratings. The experience shows that severity ratings from a single evaluator are unreliable to be trusted (Nielsen, 1994). As more evaluators are asked to judge the severity of usability problems, the quality of the mean severity rating increases

rapidly, and using the *mean of a set of ratings from three evaluators* is satisfactory for many practical purposes.

5.4.2.2.1 Usability Severity Ratings

The *0 to 4 rating scale* can be employed to rate the severity of usability problems (Nielsen, 1994). Table 5.5 shows an example of a usability problem and its severity rate:

- 0 = I don't agree that this is a usability problem at all.
- 1 = Cosmetic problem only: need not be fixed unless extra time is available on project
- 2 = Minor usability problem: fixing this should be given low priority
- 3 = Major usability problem: important to fix, so should be given high priority
- 4 = Usability catastrophe: imperative to fix this before product can be released

The Recommendations column suggests a solution for the problem.

Problem Description	Name of the criterion	Severity Rate	Explanation of the Usability Issue	Relationships with other problems	Recommendations
Vertical and horizontal scrolling is not possible in each window	Consistency and standards	1	Only two screens have scroll Bars.	Speediness of the task is compromised.	Give user more control

Table 5.5: Example of a usability problem and its severity rate.

5.4.2.2.1.1 Usability Severity Ratings and Recommendations for MTM

Problem Description	Name of the criterion	Severity Rate	Explanation of the Usability Issue	Relationships with other problems	Recommendations
1. No user customization (Usability review 1.1 and 2.1)	Minimal Action	1	Customization is everywhere these days, from faceplates ¹³² on cell phones to background color on computer desktops to the arrangement of widgets on one's internet start page.	System-wide color scheme. Users are doing computers and mobile devices for work and leisure 12 or more hours per day. Usually corporate/consumer computer average users set as many Windows elements to black color as possible because for example too much white backgrounds cause eyes problems.	Provide a more balanced color scheme until the system can provide more customization. Long-term solution should be user customization as much as possible.

¹³² Any type of metal or plastic plate designed to fit over a device or computer component to enhance the device's functionality or its looks. Cellphones, iPods and even handheld game devices are types of consumer electronics that consumers can purchase faceplates for.

<p>2. Heavy authentication workload (Usability review 1.2)</p>	<p>Minimal Action</p>	<p>3</p>	<p>Users are required to log in to the system through different types of authentication (levels of authentication) which depends on the type of service users are accessing.</p>	<p>System performance and heavy flow of user interaction. Perception of performance is informed by two things: The speed with which an application processes data and performs operations and the speed with which the application responds to the user (Apple, 2008).</p>	<p>Realize SSO authentication across multiple service providers. This will remove the need for authenticating users at every instance of remote service provision. However this involves more than system's owner decisions to implement it. It also involves third-party players so that an agreement should be settled in order to provide this feature to users.</p>
<p>3. Heavy re-authentication (Usability review 1.4)</p>	<p>Minimal Action</p>	<p>2</p>	<p>If users have to access Web applications or Electronic Purse they have to re-authenticate and this is cumbersome to users.</p>	<p>Fluidness of the interaction is compromised.</p>	<p>Probably the same solution as above but it needs system reevaluation.</p>

4. Increased number of user logins (Usability review 1.13)	Minimal Action	2	Using different credentials (e.g. password) on different systems decrease frequency of use and memorability.	SSO for individual systems.	Reduce the number of logins that a user has to remember and thereby create a more efficient and streamlined work environment for the user. SSO can be a solution for individual systems which tend to store user credentials locally on the workstation. Care must be taken with this kind of approach since administration as well as usability issues will arise if this method is used in a network environment.
5. Software inconvenience (Usability review 1.21)	Minimal Action	2	Users must switch between different systems (e.g., local MTM transactions and a concert ticket portal).	Lack of motivation to explore other functions within the application might compromise task completion (e.g. buy a concert ticket on the Web).	Same as above.

6. Physical and Logical Access Inconvenience (Usability review 1.22)	Minimal Action	3	Users need to carry different cards for different accesses.	Productivity for corporate users, partial fulfilling or task abandonment (e.g. buy a concert ticket) for consumers.	For physical and logical access applications, contactless technology provides quick user throughput (i.e., the number of users that can be processed (or authenticated) per unit time for a given system). Also to accommodate the user's need for a single ID credential, using a contactless card for both physical and logical access could be attractive. Depending on system requirements, a contactless smart card can now be used to provide the required level of security for logical access, while providing a reliable and <i>easy to use</i> solution. Contactless technology has the advantage of not suffering from physical contact contamination or requiring precise insertion and release (GSA, 2004).
7. Lack of customization (authentication) (Usability review 2.1)	Operability	2	Users cannot choose their preferred authentication method (system-defined). Too many steps to login with certain authentication methods (e.g., OTP), and certain Web applications that require users to create an account (pre-registration) before they can access a Web site.	Task abandonment.	Streamline the authentication process and provide AM(s) options to users in order to remove any confusion or frustration which could lead to the user abandoning the process. Therefore users would have a smooth and better user experience while using the system securely.

8. Lack of user interface customization (Usability review 2.2)	Operability	2	<p>New software is continually released with larger numbers of features, and more complex UIs and applications interoperability. As the interfaces of software grow, the user's efficacy when working with the MTM's given program diminishes.</p>	<p>Typically users only use a portion of the tools and menus available at the MTM, and all the extra information in the software that the user does not need becomes a distraction that potentially slows down their work process.</p>	<p>Provide adaptable interfaces where the user herself is in control of customizing. That customization should be direct and simple, that changes would be local as well as global, and that the system would be able to support deep customization (Stuerzlinger <i>et al.</i>, 2006) (i.e. users are able to create new sets of options and change the function of tools).</p>
9. Excessive and varying levels of access and network variety (Usability review 2.7)	Operability	3	<p>Users get irritated when they attempt to access services and get interrupted to log in again. Security is a secondary task for users.</p> <p>It's very confusing the variety of AMs when accessing different services which are not simple. Several levels of access such as physical and logical access for different services which can be located in the MTM Local Area Network (LAN), Wide Area Network (WAN), or Internet (Web apps).</p>	<p>Technology and network variety in supporting those AMs.</p>	<p>MTMs are able to use operating systems such as Microsoft Windows and Linux so that is why users can access different services in other networks not only locally at the machine.</p> <p>Single Sign-On can be the solution.</p>

10. Unbalanced FRR and FAR of hand geometry recognition (Usability review 2.12)	Operability	2	Legitimate users may get irritated if the system doesn't accept them.	Throughput performance	<p>The accuracy of each system can be adjusted. However, there are some drawbacks to doing so (e.g., changing the system sensitivity to reduce false rejections may increase false acceptances).</p> <p>Zhang (2004) has documented recognition system' hand geometry scanners as being able to operate with a low FAR of 0.096% and reasonable FRR of 1.05%.</p>
11. Locked out user – Forgotten PIN (Usability review 2.14)	Operability	4	Users can only reset their PINs by going into the bank and doing it in person. Also users are not allowed to do it over the phone. This security policy represents a disaster translated to inconvenience and dissatisfaction for users who need to access their account at that very moment at the MTM.	Unavailability of services.	<p>For the time being, a short-term solution is the bank to provide a system-generated PIN upon users' request and mail it to users. This typically takes 5 business days but if user is out of the country it will take 2 weeks or more.</p> <p>A long-term solution is to provide a PIN reset via a secure Web interface. Another option is an on-demand authenticator which grants "emergency" access to a traditional token user who may have temporarily misplaced a token (e.g. left a token at home), irretrievably lost a token or forgotten a PIN. By successfully completing life question challenges from the database, users can keep making transactions by requesting on-demand authentication even in off-hour scenarios.</p>

12. Inefficient Mean Time Between Failures (MTBF) – hardware reliability (Usability review 2.16)	Operability	3	<p>The MTM system is unavailable at certain very short periods of time. Users might try to log in to the system at that exact moment. There is no way of estimating on how long the failure could last and what exactly can cause a failure. Failures might occur from the network for a number of reasons.</p> <p>System reliability which affects authentication tasks and availability of services to users.</p>	Unavailability of services.	<p>System needs to be updated regarding technology infrastructure. High availability – often implemented with failover clustering, load balanced clustering, warm standby servers, and log shipping.</p>
13. High rate of Mean Time To Repair (MTTR) (Usability review 2.17)	Operability	3	<p>System reliability which affects authentication tasks and availability of services to users.</p>	Unavailability of services.	<p>System must have an MTTR of zero, which means that it has redundant components which can take over the instant the primary one fails.</p>
14. High Total Transaction Time (3Ts) for biometrics (Usability review 2.18)	Operability	2	<p>User throughput is too high for a biometric AM compared to non-biometric ones. Though isolated transactions that take 15 seconds might be acceptable in some environments, this could be less than ideal for processing a large number of users in the MTM case. Users may feel uncomfortable using the system because it takes too</p>	<p>Task incompleteness, unavailability of services, and productivity for corporate users.</p>	<p>3Ts should be adjusted to a minimum of 4 and maximum of 10 seconds for a single user. Hand geometry uses low computational cost algorithms, which leads to fast results.</p>

				<p>much time to capture their samples (satisfaction) and to learn the interface (learnability). In addition, users cannot use the product quickly (efficiency).</p> <p>As already mentioned, the 3Ts corresponds to the time for a single user to present the biometric (acquisition time), and processing time.</p>			
15. Hidden privacy policy (Usability review 2.18)	Privacy	1		<p>The privacy policy is located under the “Security” link in the MTM’s Welcome screen.</p>	Liability	Provide a “Privacy Policy” link which directs users to the privacy policy in the Welcome screen.	
16. No users control over their private information (Usability review 3.4)	Privacy	3		<p>Private information is precious to many Internet businesses. Users are unaware that they are not in control of their private information. Increasing practice of commercializing privacy by publicly businesses progressively creates new risks for users in return for little to no protection or reward.</p>	Liability and best business practices	<p>User privacy is a serious topic and should be taken very seriously if an organization wants to be credible to its customers as it is the case of this system’s owner.</p> <p>The following policy should be set up:</p> <p>Indicate why system’s owner is collecting data and whether it intends to share it with other organizations;</p> <ul style="list-style-type: none"> • Ask for user’s consent to use their private information when collecting it from online activities. 	

17. Inconvenience to users (Usability review 3.6)	Privacy	2	<p>The system's owner does not maintain backup data in a central location, so if the card is lost, the cardholder must go to several locations to repopulate the card.</p>	<p>Availability of services, system efficiency.</p>	<p>Adopt a centralized database so that it will be easier for users to repopulate the replacement card when the original card is lost.</p>
18. Dangerous information disclosure (Usability review 3.6)	Privacy	3	<p>Private information is precious to many Internet businesses. Users are unaware that they are not in control of their private information. Increasing practice of commercializing privacy by publicly businesses progressively creates new risks for users in return for little to no protection or reward.</p>	<p>Liability and best business practices.</p>	<p>Distributed privacy preserving data mining tools are vital for mining multiple databases with a minimum information disclosure.</p>
19. Inappropriate encryption policy (Usability review 3.9)	Privacy	2	<p>System response time is slow when not using appropriate encryption policies, and by consequence the user experience suffers.</p>	<p>Productivity for corporate users, partial fulfilling or abandonment of a task.</p>	<p>One solution for example is to use video encryption algorithms which focus on protecting the more important parts of a video stream, thereby reducing the total amount of data encrypted and providing a faster response time which enhances the end-user experience.</p>

20. User insecurity regarding sensitive information (Usability review 3.10)	Privacy	2	Loss, misuse, modification or unauthorized accesses to sensitive information (e.g., social security numbers, credit card numbers, and driver license numbers) negatively affect the privacy of users depending on the level of sensitivity and nature of the information.	Higher authentication "abandonment" rates, partial fulfilling or task abandonment.	<ol style="list-style-type: none"> 1. Using strong authentication procedures and other access controls to make information usable by authorized individuals. 2. Reducing the volume of collected and retained information to the minimum necessary; 3. Limiting access to only those individuals who must have such access; 4. Encrypting data.
21. Failed and annoying login (Usability review 4.1)	Security	3	If users forget their PIN (after inserting their cards) the system blocks users after 3 failed attempts to log on. This is extremely annoying for users and more aggravating especially if they are in a hurry or emergency.	Availability of services.	If a cardholder blocks their card by entering an invalid PIN, the cardholder should have the capability to unblock the card. When the cardholder's card is initially setup, a special unblock code should be generated, encrypted and then stored in the card management system.

<p>22. Failed and annoying login when doing KBA over the telephone (Usability review 4.3)</p>	<p>Security</p>	<p>3</p>	<p>When trying to login using KBA over the phone, users need to provide a PIN as a 2nd authentication factor but if they forget their PIN, they will need to go through other AMs in order to provide that 2nd authentication factor (strong authentication). This is annoying to users especially if they forget everything so then they will need to go to the branch to have their credentials reset.</p>	<p>Availability of services.</p>	<p>Most users are much better remembering 4-digit random PINs than any other type of random password. If users forget their PIN, provide them with the security questions option to replace the PIN. For security reasons, if they don't remember them as well so users will need to set up their PIN and/or security questions at the branch or on the phone. Users will receive their resets by mail. Additionally, users must ensure their Telephone Banking PIN is not similar to their 4-digit PIN used for MTM transactions.</p>
<p>23. Slow and unprotected transactions (Usability review 4.5)</p>	<p>Security</p>	<p>2</p>	<p>Inconvenience to users when on-card biometric match is not provided. There is a separate physical access control system database, managed by the facilities organization, which maintains an employee's physical access control privileges and issues the proximity card.</p>	<p>System performance, interoperability, scalability.</p>	<p>Biometrics can be used with the card technologies discussed above (e.g., smart cards), where biometric information is stored on the card and then verified with the received biometric at the point of interaction. On-card biometric match is the concept of either matching and storing hand geometry or fingerprint on a smart card. There's no need of a database. Matching hand geometry (or fingerprint) information in the card</p>

					<p>removes the uncertainty of matching on a network-connected device, an external server, or a database. This provides faster transactions (performance), user acceptance and more security.</p> <p>On-biometric card match technology is a good candidate to replace PINs and passwords irrespective of the card technology, or the application.</p>
--	--	--	--	--	---

<p>24. Difficulty to cope with different communications channels? (Usability review 4.10)</p>	<p>Security</p>	<p>2</p>	<p>Overwhelm customers with complexity, heavy flow of interaction, and memorability issues when dealing with different communication channels. Customers have to manage complexity when dealing with different services offered through different types of communication channels such as MTM, Web, and WAP. Although it might be considered a convenient service when one does not have access physically to a MTM, it does place the burden on the customer with regards to the co-ordination of the MTM with the cell phone. In addition, customers will still be required to authenticate to the system by entering a PIN. Unlike passwords, PINs have no meaning to the customer, and then it might be even harder to remember than a password (i.e., passwords can be created to be pronounceable). PINs become harder to remember for customers who have different ones to keep track of.</p>	<p>Performance, availability of services.</p>	<p>Minimize interaction while keeping the most important services always available. Reduce the number of actions required to perform the required tasks. Provide other services by flagging as optional or customizable services but maintain a collection of “core services”.</p>
--	------------------------	-----------------	--	---	--

<p>25. Lengthy transaction computational load (Usability review 5.2)</p>	<p>Load Time</p>	<p>2</p>	<p>Computational load is the length of time needed to carry out a computational process. Users get irritated when a transaction or page takes too much time to load on the screen. This decreases the user experience significantly. According to Nielsen (1993), there's a limit of people's ability to keep their attention focused while waiting which is no more than ten seconds for Web pages.</p>	<p>System performance, partial fulfilling or task abandonment.</p>	<p>MTM should consume minimum computational load. But how can the system's owner provide feedback while waiting for server responses when required? The full length of an operation can be determined and you can tell the user how much of the process has been completed. <i>A progress indicator</i> helps maintain the user's attention, improves the user's understanding of how the system works, and also communicates that the system is still alive even if a response hasn't yet occurred.</p>
---	-------------------------	-----------------	--	--	--

26. Slow Response time (Usability review 5.4)	Load Time	2	<p>People are impatient at the MTM. They are in a hurry. They want to get things done. And, they do not want you to waste their time. Anything that slows them down will frustrate them. Slow response time can distract users in their interaction with the MTM. This makes it more annoying and difficult for users to complete a transaction (i.e. funds transfer) or pursue other goals (e.g. buy a concert ticket).</p>	<p>Performance, availability of services, partial fulfilling or task abandonment.</p>	<p>Java card is a robust multifunction card containing a cryptographic processor and secure storage token offering a solid platform on which to securely store a digital hand geometry (or fingerprint) template and to execute an on-card biometric-match function. See also <i>Problem Description 23. Slow and unprotected transactions</i> above.</p> <p>Additional measures are: To help the speed of retrieval it is important that only essential information be displayed. Graphic images usually cause the MTM system to download slower. In addition, pages built with too much code (long style sheets, many scripts, etc.). An appropriate balance between speed and design for usability should be established.</p> <p>The ideal standard guidelines for response times according to Nielsen (1993) are as follows:</p> <ul style="list-style-type: none"> • 0.1 second (one tenth of a second): ideal response time. The user doesn't sense any interruption. • 1 second: highest acceptable response time. Download times above 1 second interrupt the user experience. • 10 seconds: unacceptable response time. The user experience is interrupted at an alarming high rate and the user is likely to leave the site or system.
---	------------------	----------	--	---	---

27. Inexistent authentication method option (Usability review 5.10)	Load Time	2	If PIN is forgotten, there's no other authentication method option to login to the system. This is annoying and totally blocks users on their goals.	Availability of services, task abandonment.	The system should provide an authentication screen that prompts them to enter their PIN, and a link that will allow them to create their authentication profile again and, thus, choose a new PIN creating their customer authentication profile again.
28. Inexistent image quality and image usability components in check imaging (Usability review 5.11)	Load Time	2	Check image quality, usability and integrity are undoubtedly hot topics. Poor quality images bring exposure to fraud and liability issues. Also customer is likely to believe the system's owner is responsible for any quality issues. Clearly, this will negatively impact customer confidence.	Availability of services, task abandonment.	Make use of a software that performs a broad range of image quality and usability tests on each image and "flags" those items which represent a quality, usability or negotiability risk for a financial institution. The software should also address image quality concerns on both a per-image and scanner level. Through this approach, system's owner has the ability to "catch-and-correct" image quality and usability issues before they become concerns. <i>Image quality review</i> functionality is also essential, allowing for protection by reviewing images received and rejecting those that are poor quality and unusable. Assurance ensures the images your institution uses for forward presentment meet the image quality and usability definitions.

29. Privacy and Integration (Usability review 5.12)	Load Time	2	<p>DES is not suitable for confidential data. As already mentioned, usability aspects in this case correspond to the Administrator (not End-User) level. Here are the usability aspects:</p> <ul style="list-style-type: none"> • 3DES was the answer to many of the limitations of DES. Since it is based on the DES, it is very easy to modify existing software to use 3DES. • Advanced Encryption Standard (AES) is the substitute for DES (NIST, 2009) but 3DES will be maintained for <i>compatibility reasons</i> for several years after that. • AES will be at least as strong as 3DES and probably <i>much faster</i>. 	<p>Availability of services, interoperability.</p>	<p>Despite the strength of the DES algorithm largely used worldwide, advances in computer speed and processing power are approaching the point where brute-force searches of its 56-bit key space can be accomplished within 7 days (NIST, 2001).</p> <p>The recommendation is to use the 3DES algorithm which answers this problem by increasing the key length to 168 bits. This 3DES implementation was released and described in ANSI (1998).</p> <p>3DES is an outstanding and reliable choice for the security needs of highly sensitive information including PIN (if DES is used to encrypt PIN so PIN is vulnerable to attack).</p> <p>Finally, to improve transmission speed of 3DES there is technology that enables it to reduce backup sizes and speed the transmission of the customer's backup via the Internet (Idera, 2010).</p>
---	------------------	----------	---	--	---

30. Extensive automatic audit logs ¹³³ (Usability review 5.13)	Load Time	2	Extensive automatic logs take a lot of processing time, lowering system performance, which can be frustrating for users. Automatic logs can be used with software-based products to keep a record of all user interactions and transactions but they can have the disadvantage that they will also pick up accidental actions.	Availability of services, task abandonment.	Set up the system to typically <i>no automatic logs</i> and configure the specific types of audit logs the MTM should record. Make a balance between no logs and recording specific types of logs (i.e. transaction logs).
31. Memorability issues with alphanumerical usernames and passwords authentication method (Usability review 6.5)	Minimum Memory Load	2	The most common computer authentication method is to use alphanumerical usernames and passwords. It has been shown to have major drawbacks. For example, users are inclined to choose passwords that can be effortlessly guessed. On the other hand, if a password is tough to guess, then it is frequently tough to remember.	Availability of services, task abandonment.	The main argument for using recognition of visual items or graphical passwords ¹³⁴ is that people are better at memorizing graphical passwords than text-based passwords. Another interesting fact is that it is more difficult to break graphical passwords using the traditional attack methods such as brute force search, dictionary attack, or spyware. A graphical authentication method might be a solution. Examples are SiteKey from Bank of America, Passface, etc.

¹³³ Audit log is a chronological sequence of audit records, each of which contains evidence directly pertaining to and resulting from the execution of a business process or system function. Audit records typically result from activities such as transactions or communications by individual people, systems, accounts or other entities.

¹³⁴ A graphical password is an authentication system that works by having the user select from images, in a specific order, presented in a graphical user interface (GUI). For this reason, the graphical-password approach is sometimes called graphical user authentication (GUA).

32. Failed to communicate updates made regarding latest security services (Usability review 7.2)	Resource Safety	2	<p>Users can get surprised if for example a new emergency access authentication method has been introduced in the system and there was no prior communication about it. Users can be insecure or unsure if they are performing the right authentication method. Good communication especially related to security field is crucial towards customers.</p>	<p>Availability of services, users' insecurity, discomfort.</p>	<p>Automatic updates pushed to the system. Send out notifications/updates concerning security services to customers by mail, email, or provide a link in the MTM's Welcome page to the Security Update Service section in the MTM.</p>
33. No Reverse PIN available (Usability review 7.4)	Resource Safety	4	<p>Customer is able to send a silent alarm using a Reverse PIN in response to a threat at the MTM and get help from the bank. Users have difficulty remembering PINs especially under pressure.</p>	<p>User physical security, system trust.</p>	<p>The adoption of a Reverse PIN for emergency aids users in remembering it. So if the PIN is 2637 users enters 7362. This would avoid having a new emergency PIN issued and the bothersome task to users to remember another PIN. <i>MTM Reverse PIN:</i> After user enters the correct PIN in reverse order, the account-holder's is automatically locked and the system triggers an error message on the screen saying that the user's account has been locked. At the same time the system also dials the preconfigured National Security Force number to have the robber trapped. This is clearly a resource safety feature that MTM offers to its customers. To unlock user's account, he will be required to make a phone call to customer service providing for example security questions as an additional authentication mechanism and/or confirming his private information stored in the system owner's database.</p>

34. Lack of integration with wireless network (Usability review 7.7)	Resource Safety	2	<p>As wireless devices become progressively more pervasive and essential in the users' lives so is their need for constant connection to different networks such as Internet, corporate network, etc. When sending a silent alarm from an MTM emergency application installed in the wireless device, users' devices can be compromised as can MTM processes. Users are able to send a silent alarm through the MTM's emergency application from their mobile devices.</p>	<p>User satisfaction, performance, user physical security.</p>	<p>System's owner should provide as much access as needed to their customers providing connectivity anywhere and anytime especially in the case of user emergencies. Users can send a Reverse PIN from their wireless devices through Short Message Service (SMS) in the case of MTM system failure (i.e. user convenience and physical security). This service is only provided for customers who have subscribed to it through the bank. See also Problem Description 33. No Reverse PIN available, Recommendations.</p>
--	------------------------	----------	--	--	--

5.4.2.2.2 Security Severity Ratings

The security severity ratings are based on six aspects in the USS:

- authentication (i.e. user identity proofing and verification).
- confidentiality (i.e. information is not made available or disclosed to unauthorized individuals, entities or processes).
- integrity (i.e., data has not been modified or destroyed in an unauthorised manner).
- non-repudiation (i.e., the author of a document cannot later claim not to be the author; the “document” may be an e-mail message, or a credit-card order, or anything that might be sent over a network).
- access Control (i.e., granting access to data or performing an action; an authentication method is used to check a user login, then the access control mechanism grants and revokes privileges based on predefined rules).
- availability (i.e., a computer system asset must be available to authorized parties when needed).

The *4 to 1 (Critical to Low) rating scale* can be employed to rate the severity of security problems. It has been developed by taking into account the work done in Section 3.2.2.4 Step 4: Develop the Authentication Risk Assessment Matrix. The rating scale is described as follows:

- *4=Critical Impact:*
 - This rating is set to flaws that could be effortlessly exploited by a remote unauthenticated attacker and lead to system compromise (e.g., arbitrary code execution¹³⁵) without involving user interaction. These categories of vulnerabilities can be exploited by worms. However

¹³⁵ Arbitrary code execution is employed to describe an attacker's ability to execute any commands of the attacker's choice on a target machine or in a target process.

flaws that involve an authenticated remote user, a local user, or an improbable configuration would not be categorized as Critical Impact.

- *3=Important Impact:*
 - This rating is set to flaws that can effortlessly compromise the confidentiality, integrity, or availability of resources. These categories of vulnerabilities allow the following: local users to gain privileges, unauthenticated remote users to view resources that should be secured by authentication, authenticated remote users to perform arbitrary code, or local or remote users to effortlessly originate a denial of service.
- *2=Moderate Impact:*
 - This rating is set to flaws that might be harder or more improbable to be exploitable but given the right conditions could still lead to some compromise of the confidentiality, integrity, or availability of resources. These categories of vulnerabilities are the ones that may well have had a *Critical Impact* or *Important Impact* but are less effortlessly exploited based on a technical evaluation of the flaw, or have an effect on improbable configurations.
- *1=Low Impact:*
 - This rating is set to all other issues that include a security impact. These categories of vulnerabilities require improbable circumstances to be capable of being exploited, or where a successful exploit would entail negligible consequences.

Table 5.6 presents an example of security problems related to usability criteria and their severity rates for a MTM.

Problem Description	Name of the usability criterion	Severity Rate	Explanation of the Security Issue	Relationships with other problems	Recommendations
MTM requires only a PIN (and the smart card) when using common functionalities.	Minimal Action	2	Systems requiring only a PIN and smart card (2-factor authentication) have become a pretty common authentication method and more exploitable by hackers. As technology advances, modern hackers are increasingly more malicious and savvy.	A 2-factor authentication is not as strong for authentication as it was 5 years ago (or less).	Remove PIN as a requirement and implement a biometric authentication method (e.g., fingerprint) used in conjunction with the smart card. This will ease user interaction and improve security.

Table 5.6: Example of a security problem and its usability criterion and severity rate.

5.4.2.2.2.1 Security Severity Ratings and Recommendations for MTM

Problem Description	Name of the criterion	Severity Rate	Explanation of the Security Issue	Relationships with other problems	Recommendations
1. Exposing security holes (Security review 1.1)	Minimal Action	3	Customization is also taking place in the MTM environments which might provide individual customer preferences (e.g. service selection options, language, customized dashboard, etc.). But to achieve the delivery of next generation services via the MTM, customization means not only from end-user but also third party integration. For example, instead of an MTM presenting a generic interface, the machine would be dynamically reconfigured so that when a customer inserts his card his particular bank's screen is displayed (customized to the user's requirements), and different applications are presented to the user.	Third party integration, user preferences.	<ol style="list-style-type: none"> 1. Make a risk assessment to what degree of customization is realistic at the MTM without compromising security. 2. Use a hybrid contactless smart card can be used to provide the required level of security for logical access, while providing a reliable and easy to use solution for physical access. Or use multiple technology cards that can combine either of the ISO/IEC standard contactless smart card technologies with 125 kHz proximity technology. This enables the card to operate with legacy physical access control systems, as well as new ISO/IEC-compliant systems. Providing multiple read/write capabilities on a card can often assist in providing the tools needed to enable a transition from legacy to new technology applications over

			<p>Moreover, a quick and easy customization of service offering is completely feasible (e.g. simple, user-friendly interface would allow nontechnical staff to easily change and update the presentation layer). Although a distributed hardware and software architecture, delivering a fully customizable service this open the door for plentiful security holes if the system is not secure properly. For example, JavaScript that runs in the MTM's browser can be exploited to hack accounts of users who visit a particular user profile page.</p>	<p>time.</p> <p>3. A final point is that organizations should take advantage of the card architecture by linking physical and logical access privileges to increase security within the card.</p>	
<p>2. Authentication task overload (Security review 1.4)</p>	<p>Minimal Action</p>	<p>1</p>	<p>Users are required to switch to another AM varying complexity and familiarity with the authentication methods artifacts.</p>	<p>Availability of services, efficiency, performance.</p>	<p>Implement SSO authentication across multiple service providers. This will remove the need for authenticating users at every instance of remote service provision. However this involves more than the system owner's decisions to implement it. It also involves third-party players so that an agreement should be settled in order to provide this feature to</p>

				<p>users. Any good SSO implementation should have the flexibility to offer the integration of advanced authentication systems in order to boost the security of the user's login.</p> <p>SSO for individual systems tend to store user credentials locally on the workstation. Care must be taken with this kind of approach since administration as well as usability issues will arise if this method is used in a network environment.</p> <p>Single sign-on within network environments tend to store credentials on a central server or within a directory in order to provide access to all workstations. Administration and accessibility is thereby maximized. For enhanced flexibility, a SSO application should be able to provide both central server and local methods of credential storage.</p> <p>Additionally, consideration should be given to an implementation that provides integration with biometric devices and other advanced authentication devices to enhance security.</p>
--	--	--	--	---

<p>3. Security risk by revealing visual cues for remembering passwords/PINs to users (Security review 1.12)</p>	<p>Minimal Action</p>	<p>3</p>	<p>Attackers will have a better probability of guessing users passwords which is usually called <i>password-guessing</i> attacks (e.g. exhaustive search or dictionary attacks) if cues are given to users. These attacks are difficult to control and hence pose a major problem in the functioning of password based systems.</p> <p>The likelihood of remembering several passwords is not at all that great. Half the people who say they never write down their passwords usually need to have their passwords reset because of forgetting.</p> <p>Users get frustrated if they cannot login to the system given that they have a specific goal to achieve (e.g. withdraw money).</p>	<p>Availability of services, user frustration.</p>	<p>Some countermeasures to password-guessing attacks would be the following:</p> <p><i>Account locking</i>: after a few fixed number of unsuccessful login attempts, the account of the particular user is locked for some time.</p> <p><i>Delayed Response</i>: the server provides a delayed response to the user request (e.g., not faster than one answer per second).</p> <p><i>Adopt CAPTCHA authentication</i>: one of its versions allows users to authenticate as a human by recognizing what object is common in a set of images – image-based recognition. This is easy for humans to respond to but rather difficult for computers to answer. It is useful noting here that an online attacker is fundamentally a programmed computer.</p>
--	------------------------------	-----------------	--	--	--

4. Unsecure default cookies (Security review 1.14/1.15)	Minimal Action	3	Whenever the system provides fields that already have values, the system should reduce the time users spend typing and improve their accuracy. Users can still override the default values, if required.	Performance, availability of services, user satisfaction.	A simple and secure solution by still using cookies is the one which indicates that the cookie should only be accessible via SSL on a page using the HTTPS protocol. All other default aspects of the cookie remain the same. To set a secure cookie code, use the "secure" option which creates a secure cookie by setting the "secure" option to true. As mentioned, this solution will only work if the page calling this code uses the HTTPS protocol, otherwise the cookie will be generated with default options.
5. Unsafe PIN credential (Security review 1.17)	Minimal Action	1	PINs take less time to enter than passwords. They are considered low-security compared to passwords. PINs are seldom employed as the only form of authentication for a system access (i.e. one-factor authentication). Besides weak <i>PIN Change protocols</i> , a common threat is <i>PIN Selection</i> : All it takes is a stolen Social Security number and some	Availability of services, back-end effectiveness, productivity.	Some recommendations for fraud prevention are the following: <i>PIN Change Protocol</i> : Anytime a customer calls the Customer Service Center to access accounts, information should be requested such as the most recent statement balance or the answer to customer-selected security questions to verify the identity of the customer. Customer representatives must not provide any personal information or make changes to a customer's account without verifying their

			<p>other private information (e.g., mother's maiden name) and an attacker can call an institution pretending to be a legitimate user requesting a PIN change. Once the attacker has the PIN changed, she can access all the user's accounts the system has tied to that PIN number.</p> <p>Another threat is <i>Automated PIN Checking</i>. Some hackers employ more high-tech methods to discover your PIN number. Computers can be employed to run hundreds of combinations of account numbers and PINs at high speed until the exact PIN is entered.</p>		<p>identity. Finally, it must be forbidden to change customer-selected PINs at any time by telephone.</p> <p><i>PIN Selection:</i></p> <p>These days, the average user must remember a plethora of PINs and passwords. Nevertheless, the best defense against becoming the victim of fraud can be selecting a PIN or password that is not easily guessed.</p> <p><i>Automated PIN Checking:</i></p> <p>Lock access to an account after three incorrect PIN attempts. To unlock, a customer must call and verify their identity.</p>
--	--	--	---	--	---

<p>6. Physical and Logical Access Diminished Security (Security review 1.21)</p>	<p>Minimal Action</p>	<p>2</p>	<p>Although a 3-factor authentication is used (hand geometry), it is not implemented using a contactless technology which affects performance and security.</p>	<p>Ubiquitous computing (i.e., technology retreats into the background of our lives), performance, and reliability.</p>	<p><i>Physical Access Control</i> authenticates individuals and permits access to physically secure areas. <i>Logical Access Control</i> in turn authenticates individuals and permits access to accounts and networks. For physical access applications, contactless technology provides reliable throughput (i.e. interchange of data). If biometrics is used, the throughput advantages offered by contactless technology are decreased, but the strength of security and authentication is increased. As already stressed, usable security involves tradeoffs between security and usability. As the authentication method used is biometric which is considered quite usable there's no significant impact in the usability standpoint. Applications using contactless smart cards support many security features that ensure the integrity, confidentiality and privacy of information stored or transmitted, including the following:</p>
---	------------------------------	-----------------	---	---	--

				<p>1. <i>Enhances Security:</i> A key smart card advantage is its capability to carry either a digital certificate or a biometric template to enhance authentication of the cardholder's identity. Smart cards provide the tools to enable more secure access to buildings, secure areas, and electronic systems. The smart card provides a secure token to hold the key pairs that enable the authentication of the recipient and originator of transactions across public networks, and if desired, that can be used to encrypt transactions. In conjunction with smart cards, biometrics can provide strong security for PKI credentials held on the cards, thus providing greater trust in PKI services, especially digital signatures for non-repudiation.</p> <p>2. <i>Performance:</i> With the advent of improved operating systems such as Java Card and faster processors, the time to read data from and write data to the chip has been reduced substantially.</p>
--	--	--	--	--

					<p>3. <i>Mutual authentication:</i> For applications requiring secure card access, the contactless smart card-based device can verify that the reader is authentic and can prove its own authenticity to the reader before starting a secure transaction.</p> <p>4. <i>Strong information security:</i> Information stored on cards can be encrypted and communication between the contactless smart card-based device and the reader can be encrypted to prevent eavesdropping.</p> <p>5. <i>Strong contactless device security:</i> It's extremely difficult to duplicate or forge and has built-in tamper-resistance.</p> <p>6. <i>Authenticated and authorized information access:</i> The card has the ability to process information and react to its environment allows it to provide authenticated information access and protect the privacy of personal information. The card can verify the authority of the information requestor and then allow access only to the</p>
--	--	--	--	--	---

					<p>information required. Access to stored information can also be further protected by a personal identification number (PIN) or biometric to protect privacy and counter unauthorized access.</p> <p>7. <i>Strong support for information privacy:</i> The card ensures the ability of a system to protect individual privacy. Unlike other technologies, smart card-based devices can implement a personal firewall for an individual, releasing only the information required and only when it is required. The ability to support authenticated and authorized information access and the strong contactless device and data security make contactless smart cards excellent guardians of personal information and individual privacy (GSA, 2004).</p>
--	--	--	--	--	--

<p>7. <i>No standard/best practices for Role-Based Access Control (RBAC)</i> (Security review 1.22)</p>	<p>Minimal Action</p>	<p>1</p>	<p>Users are overwhelmed with so many things to do that it is difficult to cope with everything especially remembering their usernames, passwords, and PINs! As already stressed in this thesis, security is a secondary task for users.</p>	<p>Availability of services, productivity.</p>	<p><i>Best Practices:</i> The use of RBAC to manage user privileges within a single system or application is extensively accepted as a best practice. Systems including Microsoft Active Directory, Microsoft SQL Server, SELinux, grsecurity, FreeBSD, Solaris, Oracle DBMS, PostgreSQL 8.1, SAP R/3 and many others successfully implement some form of RBAC.</p> <p>A key feature of this model is that all access is through roles. A role is essentially a collection of permissions, and all users receive permissions only through the roles to which they are assigned, or through roles they inherit through the role hierarchy.</p> <p>Within a system (organization), roles are created for various (job) functions. The permissions to perform certain operations are assigned to specific roles. Members of system users are assigned particular roles, and through those role assignments acquire the permissions to perform particular system functions.</p>
---	------------------------------	-----------------	--	--	---

					<p><i>“Trusted-authentication” mechanism (or by roles):</i> this mechanism provides a very convenient and secure way to authenticate to login for end-users.</p> <p><i>Increased security:</i> Users' profiles and privileges can be modified rapidly if Administrators manage them. Changing policies and updating user profiles in a timely manner can help maintain high levels of security.</p>
<p>8. Lack of flexibility (Security review 2.8)</p>	<p>Operability</p>	<p>2</p>	<p>As already highlighted, information transiting over the Internet (open network) is subjected to attacks. Although the system uses encryption there's always a risk.</p> <p>Users can get frustrated if remote or local authentication is not functional. Remember, they have a goal to achieve and they want to do it quickly.</p>	<p>Availability of services, performance.</p>	<p>System should be set up as local access by default which represents stronger security. It means that information does not need to be transmitted over the network so avoiding the growing threat of online attacks on consumers.</p>

<p>9. Lack of stronger security (Security review 2.9)</p>	<p>Operability</p>	<p>3</p>	<p>Although a 3-factor authentication is used (hand geometry), it is not implemented using a contactless technology which affects performance and security. The smart card is not used to store user's biometric data including user profile and enrolment.</p>	<p>Portability, performance, Interoperability, flexibility.</p>	<p>Same recommendations as in item 6. The biometric method of Match-on-card protects the initial enrollment template since it is maintained within the smart card and never transmitted off-card (GSA, 2004). A contactless smart card can support biometric authentication. For human identification systems that necessitate the highest degree of security and privacy, a contactless card can be implemented in combination with biometric technology. Smart cards and biometrics are a natural fit to supply two or multi-factor authentication. A smart card is the logical secure storage medium for biometric information. During the enrollment process, the biometric template can be stored on the smart card chip for later verification. Only the authorized user with a biometric matching the stored enrollment template receives access and privileges.</p>
--	---------------------------	-----------------	---	---	---

10. High FAR and FRR for hand geometry (Security review 2.12)	Operability	2	<p>Hand geometry authentication at the MTM currently presents high FRR of 5% and FAR of 3% which are far from the average rates.</p> <p>Users like hand geometry and fingerprint recognition because they are not intrusive, reliable, low cost, and in general only have a 0.1% FRR which is one of the lowest in the biometrics industry.</p>	<p>Availability of services, performance.</p>	<p>System should be adjusted to FRR about 1.05% and the FAR about 0.1%.</p> <p>Zhang (2004) has documented recognition system' hand geometry as being able to operate with a low FAR of 0.096% and reasonable FRR of 1.05%.</p> <p>The accuracy of each system can be adjusted which means that the secret is to balance the likelihoods of FRR and FAR, so the system barely locks out legitimate users and it doesn't fail for masquerades (See also Figure 2.28: Dynamic signature verification from Cyber SIGN.). However, there are some drawbacks to doing so such as changing the system sensitivity to reduce FRRs may increase FARs.</p>
11. PIN reset issues (Security review 2.14)	Operability	1	<p>If users forget their PINs, they cannot reset them via Internet (i.e., Web-based portal) given that the security policy related to PINs does not allow resetting PINs through the Web. According to the system security policy this is currently not possible for security reasons.</p>	<p>Performance, availability of services, effectiveness.</p>	<p>Although keeping the system current security policy is a good security decision, this affects the usability of the system. This policy will be required to change in the near future in order to provide more convenience to users.</p> <p>One solution might be a GUI to the system that allows a user to modify the PIN by providing the old PIN for authentication and then the system allows a new PIN to be set up. Another</p>

	Operability	1	In practice, the error rate seems to lie somewhere between 1 in 10,000 and 1 in 100,000.	Performance, availability of services.	approach is Web-based portal in the card management system where the user can authenticate to the portal and then navigate to a PIN reset screen where the old PIN is required and validated employing the rules set on the smart card during the chip personalization process. It's frequently hard to tell at first sight whether an exception is due to fraud or to error, and what the error rate range should be. So the lower the transaction error rate, the better. No security policy will ever be completely rigid; there will always have to be workarounds for people to cope with real life, and some of these workarounds will create vulnerabilities. In banking, generally the error rate seems to be somewhere between 1 in 10,000 and 1 in 100,000 transactions (Anderson, 2008), but this depends on the application domain.
12. High error rate of processing errors (Security review 2.17)	Operability	1	In practice, the error rate seems to lie somewhere between 1 in 10,000 and 1 in 100,000.	Performance, availability of services.	approach is Web-based portal in the card management system where the user can authenticate to the portal and then navigate to a PIN reset screen where the old PIN is required and validated employing the rules set on the smart card during the chip personalization process. It's frequently hard to tell at first sight whether an exception is due to fraud or to error, and what the error rate range should be. So the lower the transaction error rate, the better. No security policy will ever be completely rigid; there will always have to be workarounds for people to cope with real life, and some of these workarounds will create vulnerabilities. In banking, generally the error rate seems to be somewhere between 1 in 10,000 and 1 in 100,000 transactions (Anderson, 2008), but this depends on the application domain.
13. Compromised security with "off card" authentication process (Security review 2.18)	Operability	2	There is a separate physical access control system database, managed by the facilities organization, which maintains an employee's physical access control privileges and issues the proximity card.	Performance, availability of services, efficiency, portability.	Same recommendations as in item 6. Through the use of locking mechanisms and encryption, data stored on smart card chips can be made very secure (GSA, 2004)

14. Disclosure of personnel and business sensitive information (Security review 3.1)	Privacy	2	<p>Sensitive personnel information consists of personnel and medical files whose disclosure would constitute a clear unjustifiable invasion of privacy (e.g., User IDs, passwords, PINs, bank account information, and credit card numbers, SSNs). The disclosure of personnel sensitive information can result in identity theft. It refers to fraud that entails somebody pretending to be somebody else with the intention of stealing money or getting other benefits.</p>	<p>Disruption of business (i.e. disruption of the MTM owner's ability to generate services and consequently revenue), system distrust by users.</p>	<p>Smart cards help to protect privacy with secure data storage. They provide a means of securely storing data (e.g., PIN, passwords, and biometric template) on the card. This data can just be accessed through the smart card OS by those with appropriate access rights. This characteristic can be used by a system to improve privacy (e.g., storing personal user data on the card rather than in a central database). In this example, the user has better knowledge and control of when and by whom their personal data is being granted access.</p> <p>The encryption algorithm 3DES offers a significantly higher security than DES for the security needs of highly sensitive information including PIN mechanism. Since 3DES is based on the same algorithm as single DES, it can be implemented into the existing Electronic Funds Transfer (EFT) network with a minimum of disruption. The implementation of 3DES is required with the aim of maintaining users trust in payment systems and to guarantee the integrity of confidential cardholder information. For more details, see also 5.4.2.2.1.1 Usability Severity Ratings and Recommendations for MTM Study Case, Problem Description 29. Privacy and Integration.</p>
--	----------------	----------	--	---	---

<p>15. Private information compromised when using different databases from different applications that reside on a multi-application contactless card (Security review 3.5)</p>	<p>Privacy</p>	<p>2</p>	<p>The system (MTM)'s owner is the financial institution (bank) who would "own" the card and "lease" space to the third parties for its users' applications. The financial institution would have control over the card specification and operating environment.</p> <p>There is a client registry but the current system does not provide pointers to all application owner databases active on the card. The card contains different applications but different databases coexist in the same environment without any integration so that if the card is lost, the cardholder must go to several locations to repopulate the card.</p>	<p>Enables significant productivity gains, information system liability.</p>	<p>The cardholder ensures the accuracy of personal data; application owners are responsible for protecting personal data provided by the cardholder and maintaining the accuracy of that data. The recommendation here is to integrate the multiple databases on the card so that the contents of for example badging system, physical access control privilege database, and logical access control privilege database can be amalgamated into a single integrated database maintained as part of the card management system. This approach reduces the need to maintain multiple separate systems and consequently the leakage of sensitive information. In addition, the multiapplication card can securely hold multiple application usernames and passwords, offering the user convenient access through a single PIN (or biometric) and reducing or eliminating the cost of help desk calls.</p> <p>Also decentralized applications would perform all transactions, but have shadow files maintained in the</p>
--	-----------------------	-----------------	--	--	---

<p>16. Low security of match off-card technique (Security review 3.7)</p>	<p>Privacy</p>	<p>2</p>	<p>In a match off-card technique the enrolled template is originally loaded onto the smart card and then dispensed from the smart card via the contactless interface when requested by the external biometric system. The external equipment then compares a new live scan template of the biometric with the one being presented from the smart card. This implementation obviously has some security risks related to transmitting the enrolled template off the smart card for every biometric challenge. System's owner</p>	<p>Interoperability, performance.</p>	<p>centralized database of the application owners. Finally, the system's owner should provide procedures to safeguard the privacy of "shadow" databases, and document these procedures in the card issuer/cardholder agreement (GSA, 2004). The recommendation in this case is to make use of the match on-card technique. Match on-card technique originally stores the enrollment template into the smart card's secure memory. When a biometric match is requested, the external equipment submits a new live scan template to the smart card which then carries out the matching operation within its secure processor and securely communicates the outcome to the external equipment (i.e. when the smart card itself is employed to carry out the one-to-one identity verification rather than external equipment, a high degree of confidence and security of the credential's verification is achieved). This method protects the original enrollment template since it is maintained within the smart card and</p>
--	-----------------------	-----------------	---	---------------------------------------	--

			<p>should ensure the confidentiality and integrity of the released template. With match off-card technique, the smart card is storing a template (or multiple templates), but has no major knowledge of the kind of biometric information, nor the capability to process it in any manner (GSA, 2004).</p>	<p>never transmitted off-card. Matching hand geometry information in the card removes the uncertainty of matching on a network-connected device, an external server, or a database. This could be regarded as weak links in a security chain. User privacy is also maintained with this technique since the user's biometric template information is not readable from the card. With this technique, the card must be a microcontroller-based device and be able of computing the one-to-one match (GSA, 2004).</p>	<p>never transmitted off-card. Matching hand geometry information in the card removes the uncertainty of matching on a network-connected device, an external server, or a database. This could be regarded as weak links in a security chain. User privacy is also maintained with this technique since the user's biometric template information is not readable from the card. With this technique, the card must be a microcontroller-based device and be able of computing the one-to-one match (GSA, 2004).</p>
<p>17. Unsafe PIN length (Security review 4.2)</p>	<p>Security</p>	<p>3</p>	<p>A MTM machine relies on short, low-entropy PINs for authentication. A four-digit PIN can be broken in less than a second, and a 6-digit PIN in about 10 seconds, while a 10-digit PIN would likely take weeks to crack.</p>	<p>Performance, efficiency.</p>	<p>(ISO 9564-1:2002) allows for PINs from 4 up to 12 digits, but also notes that for usability reasons, an assigned numeric PIN should not exceed six digits in length. So ideally, use PINs with a large number of digits for instance a 6-digit PIN. A longer PIN obviously provides greater security against an attacker who tries to guess a user's PIN or who tries to read a PIN over the shoulder of a user. Hence, a bit longer code (more than 4 digits) is not a hardship given the security benefits.</p>

<p>18. Compromised security when placing security and financial applications in single functions application card (Security review 4.2)</p>	<p>Security</p>	<p>2</p>	<p>A single card employed for different purposes runs the risk of creating a centralized storehouse of data about an individual's activities (e.g., banking, medical, and credit cards transactions records).</p> <p>Software liability, scalability, and system trust.</p> <p>Contactless multi-application smart card technology is used in applications that require to protect personal information and/or deliver fast, secure transactions. The majority of manufacturers write proprietary OSs for each category of chip card they produce. However, lately multi-application operating systems have been developed. These OSs are placed on top of the card's proprietary system and can host parallel applications separated by secure firewalls.</p> <p>The Java Card platform provides a secure execution environment with a firewall between different applications in the same card (e.g., IT security: logon to networks, digital signature, biometrics, and encryption, banking and finance, transportation, telecommunications, government, corporation applications, etc.). This allows diverse applications on the same card to function independently from each other as if they were on separate cards.</p> <p>The "walls" between the individual's activities records keep individual</p>
--	------------------------	-----------------	--

	Security	4			<p>privacy in two ways:</p> <ol style="list-style-type: none"> 1. They restrict the damage to individual privacy that takes place through either misuse by an authorized user or unauthorized access by an attacker. 2. They place auditing on the monitoring capacity of each system. <p>Security features that ensure the integrity, confidentiality and privacy of information stored or transmitted, include the following:</p> <p>Smart cards provide a means of secure communications between the card and card readers. Similar in concept to security protocols used in many networks, this feature allows smart cards to send and receive data in a secure and private manner. This capability can be used by a system to enhance privacy by ensuring that data sent to and from the card is not intercepted or tapped into.</p>
<p>19. Lower security when employing inappropriate security algorithms (Security review 4.6)</p>			<p>Currently the system does not employ the algorithms specified within the (FIPS 140-2_02) FIPS 140-2: Security Requirements for Cryptographic Modules.</p>	<p>System trust, scalability, performance.</p>	<p>Meet the security requirements indicated in (FIPS 140-2_02) FIPS 140-2: Security Requirements for Cryptographic Modules.</p>

<p>20. Missing additional authentication method when using KBA over the telephone (Security review 4.9)</p>	<p>Security</p>	<p>3</p>	<p>The risk of eavesdropping through telephone network is a genuine threat, particularly since it cannot be encrypted.</p>	<p>Flexibility, availability of services, user satisfaction.</p>	<p>Introduce an additional authentication method to be used in conjunction with PIN like voice recognition. The ideal recommendation here would be to switch to Voice-over-Internet-protocol (VoIP) phone calls. It means that the system would convert a voice signal into data packets and sends them over the Internet. Voice over IP converts conversations to packets of bits that can be effortlessly encrypted with secret keys.</p>
<p>21. Response Time compromised (Security review 5.4)</p>	<p>Security</p>	<p>2</p>	<p>Processor currently used is not adequate to the high demands of smart card technology.</p>	<p>Availability of services, task abandonment.</p>	<p>An example of a faster processor would be the ARM(R) SecurCore(R) SC300(TM) processor¹³⁶. It provides fast real-time handling of multiple interfaces for new high-speed and contactless applications. Another solution is to employ the Java Card Technology.</p>

¹³⁶ ARM Inc. Retrieved on February 12, 2009 <<http://www.arm.com/products/CPU/families/SecurCoreFamily.html>>

22. Security risks using WMLScript language (Security review 4.10)	Security	2	<p>Malicious scripts will have the ability to falsely ring up charges or potentially off-load money from smartcards or bank accounts.</p> <p>The security risks of WMLScript language are the following:</p> <ol style="list-style-type: none"> 1. WMLScripts is not a type-safe language. Type safety is a property of some programming languages that involve the use of a type system to prevent certain erroneous or undesirable program behavior called type errors. 2. WMLScripts can be scheduled to be pushed to the client device without the user's knowledge. 3. WMLScripts language does not prevent access to persistent storage. <p>Possible attacks:</p> <ol style="list-style-type: none"> 1. Theft or damage of personal information; 2. Abusing user's authentication information; 3. Maliciously offloading money from smart cards. 	<p>Availability of services, applications liability, and system trust.</p>	<p>The MTM hosts different applications in order to provide services to its customers. However the security of those applications cannot be controlled by the MTM's owner when users are accessing these applications over the Internet through the MTM.</p> <p>The recommendation here is that the system's owner should evaluate third-parties applications and its security risks before implementing them on the MTM. Some of the best practices are:</p> <ul style="list-style-type: none"> • MTMs should not allow unauthorized applications to communicate. • MTMs should verify application integrity. • MTM should be invisible to viruses, worms, and hackers. • Lock-down the MTM allowing only authenticated applications and data transfer to execute between trusted endpoints.
--	-----------------	----------	--	--	---

23. Lack of response time (Security review 5.4)	Load Time	2	<p>A smart card with a slow response time can lead to a substantial performance decrease. For example, consider this scenario: the threat from an untrusted card reader that the user sticks her card into. The cardholder does not have infinite patience. If she puts her smart card into a reader and nothing happens for more than a few seconds, she will likely pull the card out and try again. If nothing happens yet again, she will find another reader. The slow response of the card merely allows a fraudulent reader does so much damage before the cardholder removes her card.</p>	Availability of services, task abandonment.	¹³⁷ A high performance FIPS certified smart card with a separate processor and cryptographic chip and memory for encryption, and with a 32-bit CPU is the recommended solution for this issue.
---	------------------	----------	--	---	--

¹³⁷ Federal Information Processing Standards (FIPS). Retrieved on June 2, 2009 < <http://www.itl.nist.gov/fipspubs/> >

24. Lack of system integration (Security review 5.8)	Load Time	2	Without ensuring auditing and data integrity capabilities to the system, users and system's owner will suffer from potential attacks to the check imaging process.	Availability of services, task abandonment.	System should integrate the check imaging workflow process to the IT infrastructure to easily detect fraud and facilitate auditing and reporting.
25. Lack of security when using check imaging (Security review 5.11)	Load Time	3	With various forms of check fraud already on the rise, the trend toward check imaging stands to have an even greater impact on the security of the system's owner. Though check imaging will certainly contribute to cost savings and increased convenience for customers and the system's owner alike, the transformation introduces major risks to everyone involved since these electronic "substitute checks" include all of the printed version's sensitive information, with little of its security features.	Availability of services, system trust, performance, flexibility.	<p><i>Make Alterations Detectable:</i> System's owner should offer an image security feature which makes image alterations detectable by creating an exclusive mathematical digital signature on the original image. As a result, system's owner in the image exchange process can recognize even the slightest alteration to an image.</p> <p><i>Digital signature:</i> System's owner should check processing hardware and image security software to generate an exclusive digital identity for each check image created. The mathematical digital signature is attached to the image file at the point of capture and stays with the image as it moves through the payment system.</p> <p>The benefits of image security for the system's owner and users are: Detects fraud quickly, reduces the risks associated by image exchange, generates a secure identity for each check image, and generates an unquestionable link between the image and the imaging system's owner.</p>

26. Lower security using DES encryption algorithm (Security review 5.12)	Load Time	3	See - 5.4.2.2.1.1 Usability Severity Ratings and Recommendations for MTM Study Case, Problem Description 29. Privacy and Integration.	Availability of services, system performance.	Same recommendations as in Usability Severity Rates item 29.
27. Weak auditing capabilities (Security review 5.13)	Load Time	2	Auditing user activity provides the auditor with assurance that the policies, procedures, and safeguards that management has established are working as intended. However the system does not provide customizable auditing and reporting capabilities only automatic logs which also includes accidental actions.	Availability of services, system trust, performance.	As already mentioned, system should integrate the check imaging workflow process to the IT infrastructure to easily detect fraud and facilitate auditing and reporting. The most important audit files that should be analyzed online is file server's security audit log files The main problem is to sort important audit logs from not so important ones. Very skilful experts are required to prioritize logs.

28. Lower PIN security (Security review 6.8)	Minimal Memory Load	2	<p>A PIN should be a number that users will easily remember. Users will need it every time they get information on their claim. However it also should not be a number that others can easily guess. Users should avoid their birth date, social security number, telephone number, street address, etc. Because PINs are used often, users may not use consecutive numbers such as 1234 or recurring numbers such as 1111.</p>	<p>Availability of services, user authentication performance.</p>	<p>A fast additional authenticator is the use of “security questions”.</p> <p>A security question is used as an authenticator by banks, cable companies and wireless providers as an additional security layer. They are a type of shared secret. Users must remember the precise spelling (and sometimes even case) of the answers they provide to the security questions. The best answers are straightforward, memorable, can't be guessed easily, and don't modify over time.</p> <p>For example, a financial institution could ask for a customer: “Where does your nearest sibling live?” before issuing a replacement for a lost debit card.</p> <p>Adopting security questions in conjunction with PIN represents a strong authentication (two-factor authentication).</p>
--	----------------------------	----------	---	---	--

29. Lack of enforcement of security policy (Security review 7.4)	Resource Safety	4	<p>MTMs offer a real convenience but at the same time offer an element of risk. Many of the MTM robberies occur after the cash withdrawal. So system's owner should provide a resource safety countermeasure to its customers and enforce its security policy. Without security policies, no enforcement of security standards or configurations is able to be made. By establishing a policy, it is supposed that enforcement can or will pursue. Without security policies, enforcement of them is not achievable.</p>	<p>Availability of services, system trust, and user confidence and satisfaction.</p>	<p><i>MTM Reverse PIN:</i> After user enters the correct PIN in reverse order, the account-holder's is automatically locked and the system triggers an error message on the screen saying that the user's account has been locked. At the same time the system also dials the preconfigured National Security Force number to have the robber trapped. This is clearly a resource safety feature that an MTM offers to its customers. To unlock user's account, he will be required to make a phone call to customer service providing for example security questions as an additional authentication mechanism and/or confirming his private information stored in the system owner's database.</p>
30. Lack of wireless attacks prevention (Security review 7.4)	Resource Safety	2	<p>As wireless devices become progressively more pervasive and essential, they are becoming both a target for attack and also a weapon with which such an attack can be performed. So it is responsibility of the</p>	<p>Availability of services, system trust, user physical security.</p>	<p>Super Administrator is able to wipe information within the wireless device or disable a lost or stolen one with Over-The-Air (OTA) commands. Remote device wipe initiates and tracks a remote wipe command for lost or stolen mobile devices. Remote</p>

			<p>system's owner to offer security in all points of interaction between the MTM itself and components such as the silent alarm application installed in the user's wireless device.</p> <p>When sending a silent alarm from an MTM emergency application installed in the wireless device, users' devices can be compromised as well as MTM processes.</p>		<p>device wipe is a feature that enables a server to set a mobile device to delete all data the next time that device connects to the server.</p> <p>OTA is a standard for the transmission and reception of application-related information in a wireless communications system.</p> <p>OTA is commonly used in conjunction with the Short Messaging Service (SMS), which allows the transfer of small text files even while using a mobile phone for more conventional purposes. In addition to short messages and small graphics, such files can contain instructions for subscription activation, banking transactions, ringtones, and Wireless Access Protocol (WAP) settings.</p> <p>OTA messages can be encrypted to ensure user privacy and data security.</p>
--	--	--	---	--	--

5.4.3 Applicability of USS in the AMDLC:

Although the USS inspection method has been primarily developed to be applied in the Requirements and Design phase of the Authentication Method Development Life Cycle (AMDLC), it can be also used as an evaluation tool after the product, authentication method, has been released to manufacturing or market. The USS can be employed in two ways such as *guiding the design decision* or *assessing the design of a final product* as follows:

- *Guiding the design decision* (requirements and design phase):

In the case of *guiding the design decision* of the authentication method, USS is used in the Requirements and Design phase of the AMDLC. It influences the design in its early stage when traditionally the bulk of the feature design is done. Therefore, it makes the security and usability a natural outcome of the AMDLC (see Section 5.1 Introduction, Figure 5). As mentioned in Section 6.3 Practical Observations on the Impact of USS in Corporate and Academic Environments, by analyzing and answering concurrently the usability and security review questions in the inspection method, USS forces User Experience and/or Security Designers to think in the process as a whole (usability and security) not a part of the whole (usability or security). USS also forces them to initiate - or trigger - potential solutions in their minds for the questions/issues. In addition, while evaluating the review questions User Experience and/or Security Designers will be able to anticipate the identification of potential bugs earlier in the Requirements and Design phase that would otherwise occur when the product is handed-off to the manufacturing.

- *Assessing the design of a final product* (readiness phase):

Conversely, USS can also be applied to *assess the design of a final product* (i.e. authentication method) which has been released to manufacturing or to the market. USS is used in this case in the Readiness phase of the AMDLC.

It can be used to point out what security and usability fixes and/or improvements should be implemented in the next release of the authentication method. This is a very important aspect of the authentication method development life cycle which can represent benefits in terms of improved product reliability, greater business, reduced customer support calls, smaller releases product life cycle, reduced number of bugs to be opened against the product, and finally enhanced user interaction (i.e. finding bugs earlier can force designers to review and improve authentication method functionalities with developers as well as project and product managers).

5.5 The Demonstrational Approach

The Verification and Validation (V&V) phase is undertaken within this thesis with the demonstrational approach. V&V are supplementary techniques aimed at checking the quality of the system generated. Verification is a Quality control process that is employed to assess whether or not a product, service, or system conforms to regulations, specifications, or conditions included at the start of a development phase. Verification is frequently an internal process.

Validation is Quality assurance process of laying down facts that provides a high degree of assurance that a product, service, or system accomplishes its planned requirements. The crucial goal of validation is to make the model useful; it addresses the right problem, and provides correct information about the system being modeled.

Typically validation can be expressed by the question "Are we building the right thing?" and verification by "Are we building the thing right?". "Building the right thing" refers back to the user's needs, while "building it right" checks that the specifications be properly implemented by the system.

In scientific investigation, an experiment is a method of investigating causal relationships among variables. An experiment is a foundation of the empirical

approach to acquiring data about the world and is employed in both natural sciences and social sciences. An experiment can be employed to aid in resolving a practical problem which is the focus of this thesis: the usable security of user authentication methods.

Security systems should be viewed as socio-technical systems that depend on the social context in which they are embedded to function correctly. Security systems will only be able to offer the intended protection when people truly understand and are able to use them correctly. As Jøsang *et al.* (2007) stresses, there are genuine differences between the degree by which systems can be considered theoretically secure (supposing those systems are properly operated) and in reality secure (acknowledging that frequently those systems will be operated erroneously). Often, as already stated throughout this thesis, there is a trade-off between usability and theoretical security. It can be useful to reduce the level of theoretical security to improve the whole level of actual security. For example, the strongest passwords are the ones, from a theoretical perspective, randomly generated. However, given that it is difficult to remember such passwords, people will write them down, and in so doing weaken the system's security. Thus, it may be important to let people to choose passwords that are easier to remember. Even though this reduces the theoretical strength of the passwords, it intensifies the security of the system as a whole.

To this end, the demonstrational approach is translated in this thesis in two forms:

- Through the applicability of the USS: this thesis applies the USS inspection method to evaluate the OTP authentication method.
- Through the demonstration of the One-Time-Password (OTP) authentication method which is described in the next sections.

5.5.1 Demonstration of One-Time-Password (OTP)

This section describes the demonstration of the OTP authentication method which has been subjected to the GOMS analysis in Sub-Step 3.2. The OTP demo has as a goal to demonstrate the difficulties that users are subject to when using this particular authentication method. It focuses on the usability aspects of the user interaction with the system. The OTP demo is a wireless-and-token based authentication task which is comprised of the following elements:

- a Wireless Local Area Network (WLAN);
- a hardware token with OTP functionality;
- a Personal Identification Number (PIN) ;
- a tokencode.

5.5.1.1 Wireless Network

A wireless network refers to any type of computer network that is associated with a telecommunications network whose interconnections between nodes are implemented without the use of wires. Wireless networks are generally implemented with some type of remote information transmission system that uses electromagnetic waves, such as radio waves, for the carrier and this implementation usually takes place at the physical level or "layer" of the network (OSI Model¹³⁸).

5.5.1.2 Hardware Token with OTP Functionality

As already mentioned, RSA SecurID® 700 hardware token is an OTP scheme (Figure 2.7) and an associated authentication mechanism to protect an organization's most critical information assets. It is generally used to secure either local or remote

¹³⁸ OSI Model and Communication Between Systems (Cisco, 2010):
<http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Intro-to-Internet.html#wp1020627>

access to computer networks. Each end-user is assigned an authenticator that generates a one-time-use code which is the *tokencode*. When logging on, the user simply enters this number plus a PIN which is the *passcode* to be successfully authenticated.

RSA SecurID® 700 is a very widely accepted authentication scheme used by many infrastructure solutions. It is a battery powered, hand-held device containing a dedicated microcontroller. The microcontroller stores, in Random Access Memory (RAM), the current time, and a 64-bit seed value that is unique to a particular token. At the specified interval, every 60 seconds, the seed value and the time are combined through a proprietary algorithm stored in the microcontroller's Read Only Memory (ROM) to create the tokencode value.

An authentication server verifies the passcodes. The server maintains a database which contains the seed value for each token and the PIN or password for each user. From this information, and the current time, the server generates a set of valid passcodes for the user and checks each one against the entered value. The PIN can be changed if forgotten. The OTP is the concatenation of the four-digit user PIN and the six-digit tokencode called the *passcode*.

When the token is manufactured, a seed is encoded into the specific token. The RSA SecurID® 700 comes pre-seeded and is ready-to-use out-of-the-box. It has a unique symmetric key that is combined with a proven algorithm to generate a new OTP every 60 seconds. Patented RSA technology synchronizes each authenticator with the security server, ensuring a high level of security. The OTP, something you HAVE, is coupled with a secret PIN, something you KNOW, to generate a combination that is almost impossible for a hacker to guess. Due to the algorithm being secret and only known to a limited number of individuals, attacks against the process of generating new tokencodes are less likely to succeed, unless information about it leaks to the public, or a hacker deliberately reverse-engineers the hardware token and the authentication server. Other attacks, such as network traffic sniffing,

shoulder surfing, keyboard logging, and social engineering, might have limited success. If by one of these methods a hacker manages to obtain a tokencode and knows the user PIN number, the hacker can only use the passcode to authenticate within a limited period of time – up to 60 seconds after the tokencode was initially generated on the RSA SecurID® 700 token. Moreover, the user must not have used this tokencode to authenticate, or the tokencode will be rejected as already used. If the hacker does not try authentication within the 60-second period after having obtained actual credentials, the chance is lost and the hacker needs to find another tokencode. However, if a hacker manages to capture a legitimate tokencode from a user, the hacker already has the PIN as the first four digits of the passcode. The PIN can be used for brute force attacks, decreasing the unknown keyspace from 10^{10} to 10^6 but this is still not likely to lead to success due to the limited time that the hacker has. Beyond the 60-second validity period of the tokencode, other attacks such as password guessing, sniffing, man-in-the-middle, etc. is not possible because the tokencode has already changed.

The RSA SecurID® 700 is also tamper evident, meaning that if someone opened the token for criminal purposes, it would be apparent to the user of the device.

5.5.1.3 Personal Identification Number (PIN)

A Personal Identification Number (PIN) is a secret numeric password shared between a user and a system that can be used to authenticate the user to the system. A PIN is usually a four-digit number (sometimes six-digit number), and the user is required to provide a non-confidential user identifier or token and a confidential PIN to gain access to the system. Upon receiving the User ID and PIN, the system looks up the PIN based upon the User ID and compares the looked-up PIN with the received PIN. The user is granted access only when the number entered matches with the number stored in the system.

5.5.1.4 Tokencode

The hardware token displays a new pseudo-random value, usually a six-digit number, called the tokencode, at a fixed time interval, usually 60 seconds as shown in Figure 5.3.



Figure 5.3: Tokencode: a six-digit number.

5.5.1.5 How the OTP Demonstration Works?

To access resources protected by the RSA SecurID® 700, users combine their secret PIN with the tokencode generated by their authenticators. The outcome is a unique, one-time-use passcode that is used to positively identify, or authenticate, the user. If the code is validated by the RSA SecurID® 700 system, the user is granted access to the protected resource. If it is not recognized, the user is denied access.

5.5.1.5.1 System Requirements

A Virtual Private Network (VPN) client application should be installed in the user's laptop which in this thesis is an EMC VPN Client Version 4.8.02.0010 2006. Client type: Windows winNT.

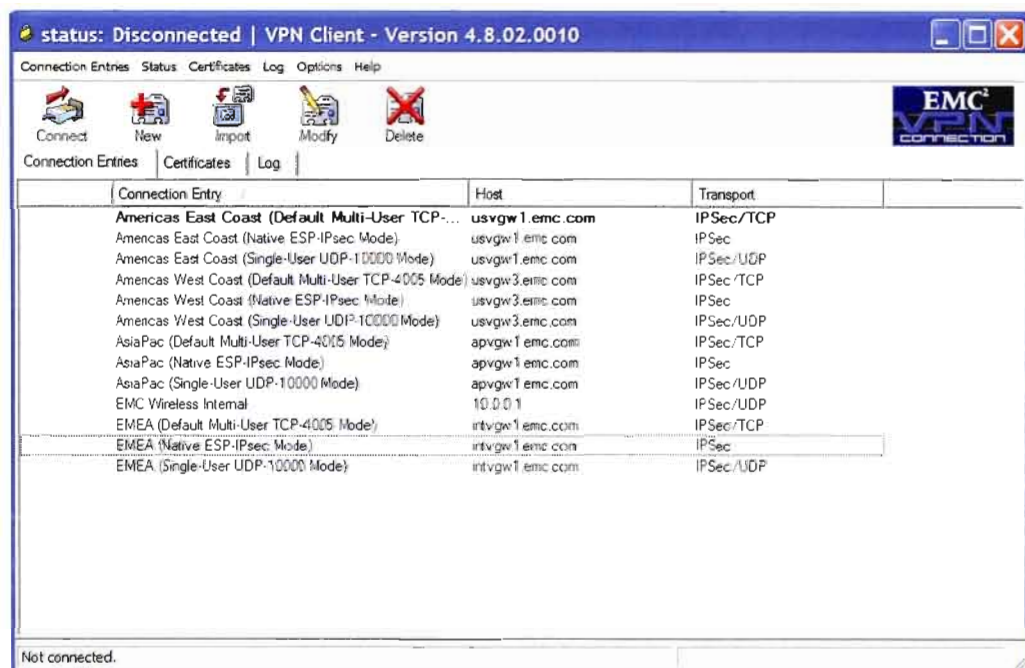
5.5.1.5.2 Demonstration Steps

The demonstration steps of the OTP authentication method with screenshots are described as follows:

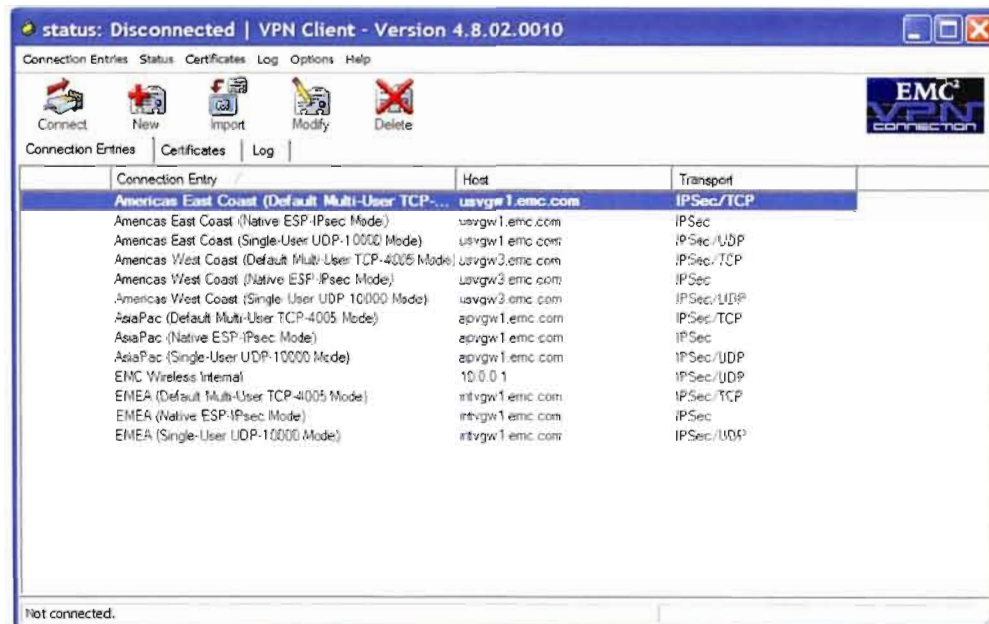
- Step 1: User access the EMC VPN application in the laptop taskbar clicking on the corresponding icon;



- Step 2: System opens up the EMC VPN application on the screen displaying a Connections Entry list for authentication;



- Step 3: User selects and double-click the proper connection entry for her/his area;



- Step 4: System initialize the authentication process by contacting the authentication server which displays the Login screen window;



- Step 5: User enters username in the username field;



VPN Client | User Authentication for "Ame..."

Enter Username and Password.

EMC² VPN CONNECTION

Username: jcarlson

Passcode:

OK Cancel

- Step 6: User enters PIN number in the passcode field. The PIN number is provided when user enrolls in the RSA® Authentication Manager 7.1 application;



VPN Client | User Authentication for "Ame..."

Enter Username and Password.

EMC² VPN CONNECTION

Username: jcarlson

Passcode: ****

OK Cancel

- Step 7: User refers to the RSA SecurID® 700 hardware token by reading and memorizing the tokencode displayed in the token's LCD (Liquid Crystal Display). The system generates a different tokencode every 60 seconds;
- Step 8: User concatenates the tokencode (six-digit number) to the PIN already entered in the passcode field and click on OK button;



VPN Client | User Authentication for "Ame..."

Enter Username and Password.

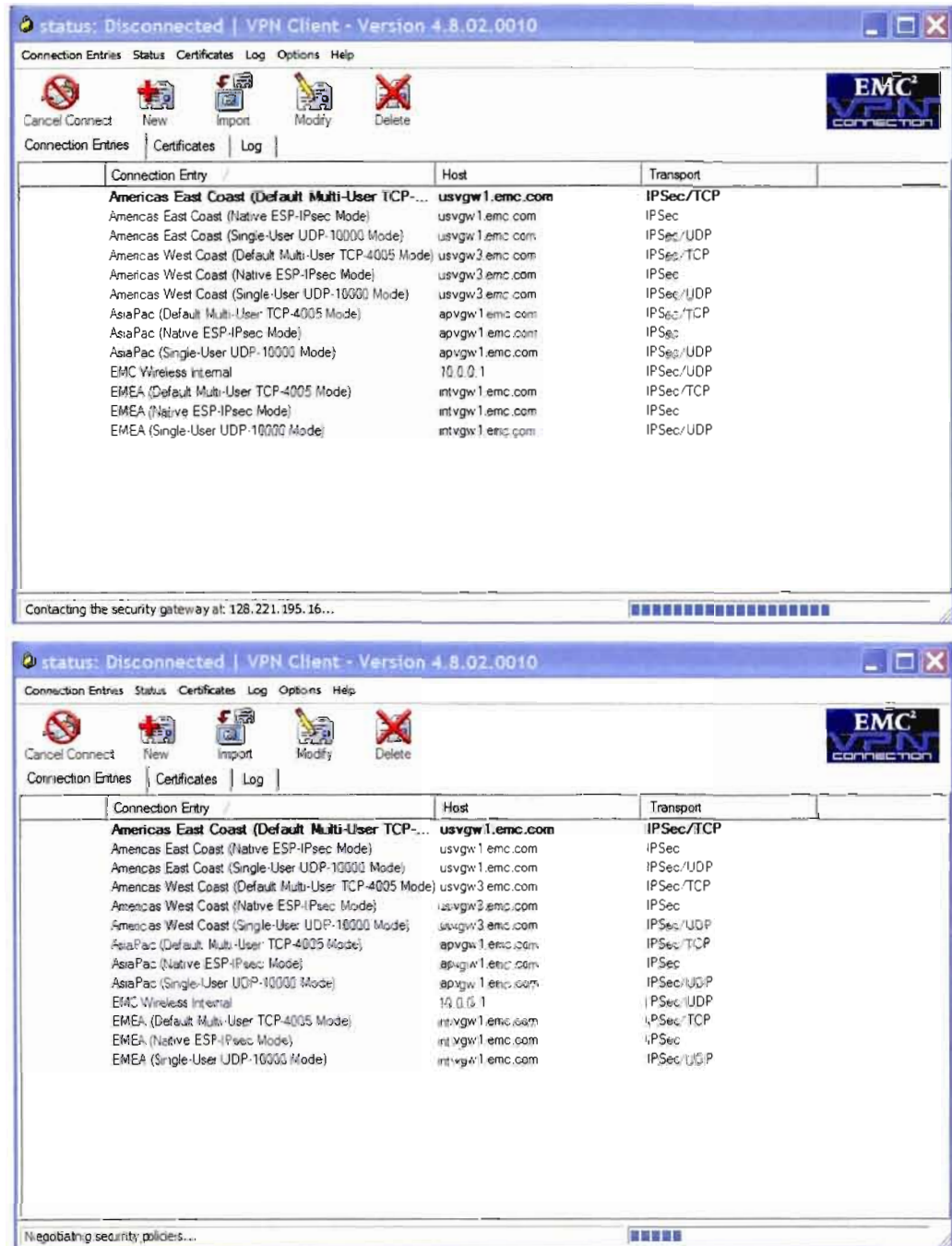
EMC² VPN CONNECTION

Username: jcarlson

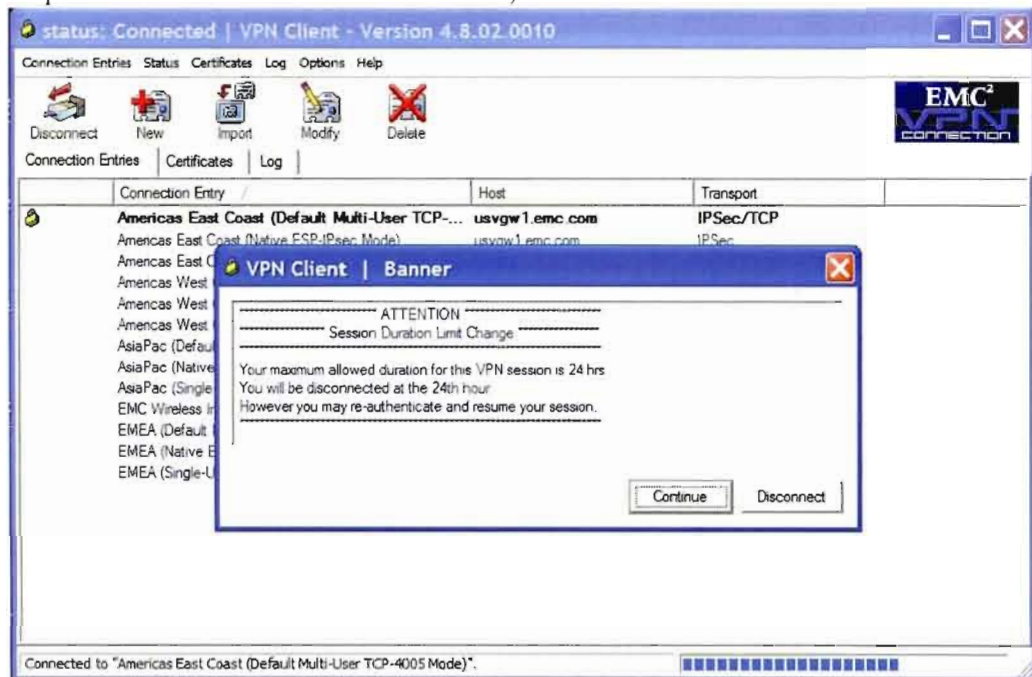
Passcode: ****

OK Cancel

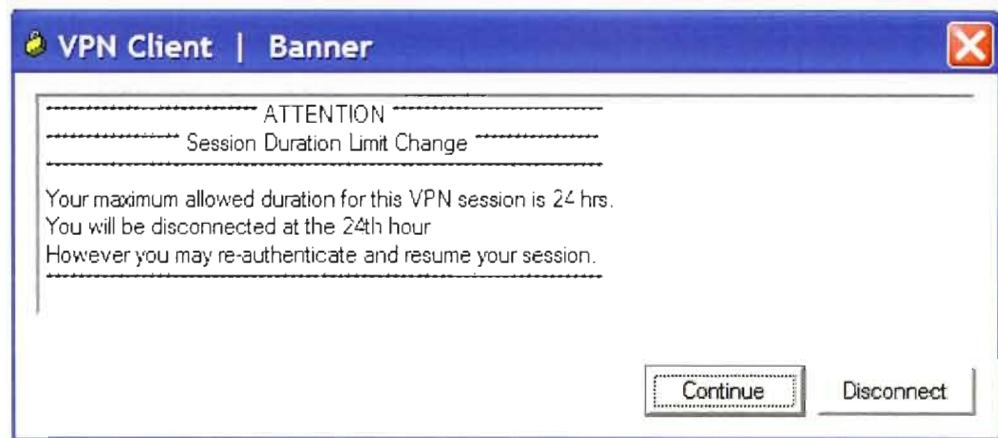
- Step 9: System contacts the security gateway;



- Step 10: User clicks on Continue button;



Detail of the popup screen:



- Step 11: If authentication is successful, system grants access to the user. The yellow padlock icon shown as locked in the taskbar means that the authentication was successful. If not, system prompts the user to enter login details again.



5.5.2 One-Time-Password (OTP) Usability Testing

OTP Usability testing has been performed at RSA - The Security Division of EMC in Bedford, Massachusetts (USA) on June, 2009.

Confidentiality Disclaimer: Certain usability testing questionnaires and tests results regarding the OTP Usability Testing cannot be disclosed to protect RSA Security confidential information.

5.5.2.1 Objectives of the OTP Usability Testing

Identify high-priority usability issues:

- Assess the usability of designs for end-user tasks involving an Invisible User Authentication enabled SSL-VPN.
- Assess the usability of designs for end-user tasks involving account creation and management using the UCM Self-Service Console.

5.5.2.2 Testing Tools

The following testing tools have been used: Medium Fidelity Clickable Prototype, Morae Recording Software, and System Usability Survey (SUS).

- Additional surveys to gauge ease of use.

5.5.2.3 Testing session

Session length = approximately 1 hour, 12 external Participants:

- Recruitment centered on non-admin users with average computer skills and familiarity with remote log in procedures.
- Users were categorized into 3 levels:
 - Low – Consistent computer user, not familiar with the concept of VPNs.
 - Medium – Moderately skilled computer user, accesses remote resources but does not have in-depth knowledge or understanding about the agent or VPN.
 - Advanced – Skilled computer user, frequently accesses remote resources, knowledgeable about VPNs.

5.5.2.4 Testing Methods – Participant Tasks

Participants were divided into 2 groups:

Group 1:

- Authenticate via an IUA-enabled SSL-VPN.
- Configure Challenge Method (Security Questions).
- Log into Self-Service Console.
- Request RSA SecurID® 800 hybrid authenticator.
- Enable Token.

Group 2:

- Create a Self-Service Account.
- Request IUA Enablement.
- Configure Challenge Method (On-Demand Tokencode).
- Authenticate via IUA-enabled SSL-VPN

5.5.2.5 Data Results

System Usability Scale (SUS):

- What does it measure?
 - Effectiveness (can users successfully achieve their objectives).
 - Efficiency (how much effort and resource is expended in achieving those objectives).
 - Satisfaction (was the experience satisfactory).
 - Average Score = 72.5 (scale of 1-100).
 - Historical Testing Results:
 1. FIM (Everest) SUS = 75.00
 2. SecurID Appliance (Clydesdale) Round 1 SUS = 67.92
 3. SecurID Appliance (Clydesdale) Round 2 SUS = 81.92
- Areas that Tested Well:
 - Consistency.
 - Perception of the ability to learn the system quickly.
- Areas of Improvement:
 - Complexity of the system.
 - Need for prior knowledge.
- Interesting data point – When rating the system on “cumbersomeness”, Group 1 gave it the overall worst rating, and Group 2 gave it the overall best of all the survey questions.
- Most tasks were completed with high success rates, but some did need additional help from the facilitator.
- There wasn't a strong correlation between user experience level and ability to complete most tasks easily.
 - The one exception was configuring On-Demand as the challenge method.
 - All other tasks were well tolerated among all user levels.

5.5.2.6 Findings Summary

Strengths:

- 2 page logon process in the SSL-VPN.
- Use of contextual help (i.e., What is this? Links).
- Save this Device.
- Security Questions:
 - Concept.
 - Specific questions available.
- SSL Page Designs.
- Request a Token self-service console pages.

Areas of Improvement:

- Concept of On-Demand service and tokencodes.

5.5.3 One-Time-Password (OTP) Usability Issues - Discussion

This section brings up and discusses usability issues regarding the RSA SecurID® 700 hardware token which is in this thesis subjected to a demonstration test. The discussion is shown right after the **(Christina Braz-n)** parenthesis.

Usability has been always a concern when designing an authentication method according to the main authenticator manufacturers RSA, The Security Division of EMC¹³⁹, VeriSign¹⁴⁰, and Vasco¹⁴¹. However achieving actual usability is in fact a most difficult task when taking into consideration user interaction.

¹³⁹ In "Consumers want security over convenience" section. White Paper: RSA 2010 Global Online Consumer Security Survey. RSA-The Security Division of EMC. May 23, 2010 <http://www.rsa.com/products/consumer/whitepapers/10665_CSV_WP_1209_Global.pdf>

¹⁴⁰ In "Information Security Risk Management, Solution: A Balanced Approach to Better Security" section. VeriSign, Inc. May 23, 2010 <<http://www.verisign.com/authentication/enterprise-authentication/information-security-risk-management/>>

RSASecurity (2010) manufactures the RSA SecurID® 700 and claims the following usability characteristics of its hardware token:

- *Convenient Form Factor*

“With its robust key ring, small size and easy-to-read Liquid Crystal Display (LCD), the SecurID 700 is a convenient form factor for employees, partners and customers.”

(Christina Braz-1) The LCD is not easy to read since due to its design and form, shadow and light reflect off the display, making it very difficult to read the passcode from the LCD. Actually the user has to turn the token in several different positions until the passcode is readable. The token is in fact a bit big not small. What was small 5 years ago (or less) it is in fact nowadays big. For example, the very first portable phones were really big and clunky; then they were small enough to fit in your briefcase, then your pocket, and then the palm of your hand. Due to the growing trends toward miniaturization (e.g., nanotechnology) and portability users will therefore have more “fitable” artifacts and easier to carry artifacts that reflect on the usability aspect of the product itself.

(Christina Braz-2) The token fits to some extent on a ring of keys but because it is a little big it makes the fob too heavy. Also to put the token on the fob ring is difficult; the ring is too thick so you have to use a knife or scissors in order to slide the token ring through the fob ring. Finally, it doesn't fit well in a pocket, because as already mentioned the token is a little big.

Users can easily read the OTP displayed on the authenticator¹⁴² and know when the number is going to change by watching the countdown indicator.”

¹⁴¹ In "Cascades secures its remote network with VASCO's two-factor authentication". Vasco May 23, 2010 <http://www.vasco.com/company/press_room/news_archive/2010/cascades_secures_its_remote_network_with_vascos_two-factor_authentication.aspx>

¹⁴² In Convenient Form Factor section. RSA-The Security Division of EMC. May 23, 2010 <http://www.rsa.com/products/secuid/datasheets/10306_SID700_DS_0709.pdf>

(Christina Braz-3) The LCD is not easy-to-read as per **(Christina Braz-1)** comment.

- *Reliable Authentication Solution*

“The SecurID 700 authenticator is designed to withstand the worst imaginable conditions, offering industry-leading reliability. From temperature cycling to mechanical shocks to being immersed in water, the SecurID 700 is subjected to rigorous tests to ensure that customers do not face hidden costs due to token failures. The combination of this high level of quality with a lifetime warranty allows organizations to reduce the overhead costs of distributing replacement tokens and drive down the overall cost of security while providing a consistent and easy-to-use authentication experience for end-users¹⁴³.”

(Christina Braz-4) An OTP is still a difficult method to handle in user authentication for the vast majority of average computer users. Users have difficulty understanding the concept of concatenation¹⁴⁴; it means users have to first input their PIN (four-digit number) then concatenate the passcode (six-digit number generated by the server) displayed on the hardware token LCD.

An OTP authentication method requires the utilization of different artifacts and different communication channels by the user, such as a hardware token, a Web interface, a laptop, and a wireless network as shown in Figure 1.2.

¹⁴³ In Reliable Authentication Solution section. RSA-The Security Division of EMC. May 23, 2010
<http://www.rsa.com/products/securid/datasheets/10306_SID700_DS_0709.pdf>

¹⁴⁴ The USS methodology may guide experts towards innovate solutions by making use of the “Usability Severity Ratings and Recommendations” and “Security Severity Ratings and Recommendations” artifacts where evaluators can propose new solutions to ease the user authentication task. Also, a comparative analysis could be undertaken by identifying on one hand the potential cognitive problems (e.g., complex concatenation of PIN and tokencode) and on the other hand potential supplementary actions (e.g., filling out an additional field beyond username and passcode) but this is outside scope of this thesis.

The OTP authentication method violates usability criteria within the USS inspection method¹⁴⁵. It concerns perceptual and cognitive workload for individual inputs or outputs which represent the following usability review questions:

- 8-out-of-22 Usability Review questions specified in the *Minimal Action* criterion;
- 3-out-of-7 Usability Review questions specified in the *Minimal Memory Load* criterion;
- 2-out-of-7 Usability Review questions specified in the *Resources Safety* criterion;
- 2-out-of-12 Usability Review questions specified in the *Load Time* criterion;
- 5-out-of-19 Usability Review questions specified in the *Operability* criterion;
- 2-out-of-10 Usability Review questions specified in the *Security* criterion.

A description of each usability criterion and its corresponding usability review questions is given below:

Usability Criterion: MINIMAL ACTION

Capability of the application to help users achieve their tasks in a minimum number of steps (i.e. the length of transactions and procedures).

- Is the workload low and simple (e.g., input workload kept to a minimum)?
- Is the authentication (verify/authenticate the identity of the user) process simple to users?
- Does the user have to authenticate using different communications channels?
- For codes longer than 4 or 5 characters, are mnemonics or abbreviations being used?

¹⁴⁵ The concept of concatenation PIN + passcode is difficult to be understood by users, and it can be tested not only employing the USS inspection method but also usability test (i.e. observing users performing the OTP task) as per section 5.5.2 One-Time-Password (OTP) Usability Testing. Also, other usability inspection techniques can be used such as Pluralistic Walkthrough, Task Analysis, etc.

- Has the requirement of data entry by the user being limited when the data can be derived by the application?
- For data entry, has the system displayed currently defined default values in their appropriate data fields?
- Does the system accommodate both experienced and average computer users (e.g., length of transactions: short cuts are available to experienced users)?
- If users must switch between different systems are dual-interface⁶ chip smart cards used?

Usability Criterion: MINIMAL MEMORY LOAD

Whether a user is required to keep minimal amount of information in mind in order to achieve a specified task.

- Are the sequences and interdependencies of the MTM's artifacts and their corresponding UIs (i.e., streamlined business workflow) "harsh-less" in the user interaction viewpoint enhancing thus customer intimacy (providing appropriate choices, information and advice)?
- Is the memory load on the user minimized (i.e., no memorization of long data lists, complicated procedures, or undertake complex cognitive activities)?
- Are the entries short (i.e., STM capacity is limited⁴³, so the shorter the entries, the smaller errors and reading times)?

Usability Criterion: RESOURCE SAFETY

Whether resources (including people) are handled properly without any hazard.

- Are audible instructions (or voice PIN) available so that people who cannot read an MTM screen can independently use the machine⁴⁹?
- If the MTM offers a silent alarm feature, is Reverse PIN used providing faster recall?

Usability Criterion: LOAD TIME

Time required for the application to load (i.e., how fast it responds to the user).

- If login fails, is there available another authentication option to the user in order to achieve her/his task?
- Does the live image require a small memory allocation which will affect time for retrieval of the image if solicited by the user?

Usability Criterion: OPERABILITY

- Amount of effort necessary to operate and control an application.
- Is the user task' workload light?
- Can users customize the user interface to their specific needs (e.g., personalized look and Feel)?
- Can (some) system's mechanisms be configured by users to operate in certain way¹⁰ (e.g., customer's language and "favorite transaction")?
- Is there an alternative authentication method, when biometrics is not available in order to provide availability to the users?
- If users forget their PINs, can users reset them via a web interface rather than in an issuance station (i.e., ensure customer convenience & satisfaction)?

Usability Criterion: SECURITY

Capability of the application to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access.

- Do users have more than a few alternatives to authenticate to the system to improve the availability and convenience of the system?
- For logical access application, is biometrics provided to enhance usability (i.e., on-card biometric match and on-card key generation)?

Summary of the topics discussed in Chapter 5: The Computer Science Axis.

This chapter presented the computer science approach as follows: Security as a Usability Characteristic, User Authentication Use Cases, Usability Factors and Usability Criteria, The USS Inspection Method, Demonstrating USS using A Multifunction Teller Machine (MTM), Usability Severity Ratings, Security Severity Ratings, and One-Time-Password (OTP) demonstration.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

6.1 Summary of the Research Work

There has been very little research on usable security especially related to user authentication methods, although a considerable body of research work has been made for computer security mechanisms in general other than authentication methods. To be able to build reliable, effective, and usable security systems, we need inspection methods that take into account the specific constraints of security mechanisms and their potential security threats. Systems should be built so as to be easy to learn and use by the average corporate or consumer computer user. Human factors should be incorporated into the development of security solutions where usability is central during the development process. Another particular concern in authentication according to Cranor and Garfinkel (2005) is that authentication systems do not fail gracefully. This means that if an average consumer computer user forgets her username but gets right the password the system does not enable her for instance, to gain partial access to an online magazine, or for an average corporate computer user access to the system's less critical files, or an emergency or temporary access. There is no established and recognized mechanism to accommodate user error which means that most likely productivity will be strongly compromised and users will be largely dissatisfied with the system. According to Sasse (2004), "Don't focus ONLY on UIs to security tools - the big problems are in security requirements, job design and user involvement." That is exactly what this thesis is all about: Requirements and Design. Additionally, according to Whitten and Tygar (1999), using conventional methods for usability evaluation that concentrate on the impact of usability on security effectiveness will assist developers to discover usability problems threatening the security of a system.

As strong user authentication becomes more imperative, technologies to accomplish it will become more convenient through the use of requirements and design tools. Since communication of intent is vital to security, the UI is also a key component for achieving computer security, not the only aspect when designing user authentication methods. To this end, this thesis has investigated the security consequences of usability issues and presented a novel usable security protocol through an inspection method named Usable Security Symmetry (USS) for dealing with usable security of user authentication methods in the HCI-Sec field that this thesis's author hopes will guide the development of truly secure and usable user authentication systems.

6.2 Scientific Contributions

This section summarizes the body of research performed at University of Quebec at Montreal over the past four years and a half. Scientific contributions contained herein include:

1. A methodology for a Usable Security Protocol which is translated into a usable security inspection method for user authentication methods named Usable Security Symmetry (USS) towards the goal of aligning usability and security. It is employed to influence the design of the authentication methods earlier in the Requirements and Design phases. USS can also be used in later stages to assess existing user authentication methods. To date, USS is the *only existing* design (and evaluation) inspection method to improve the process of user-centered design in user authentication methods. An important factor is that USS does not present only a general principle but also specify exactly what the issues are and recommendations for them. Other related work in usable security do not address the usable security issues as a holistic approach as USS according to the sub-products of this research work mentioned below

(1 to 11 items). No related work presents any formal method for evaluating usable security for user authentication or even for computer security in general. Some significant differentiators of USS compared to the general usable security principles presented in Chapter 3 are the following:

- USS points out specific usability and security issues and recommends specific solutions for each of them, avoiding therefore the use of general guidelines which are very broad in scope and at the end do not apply to anything - this is an important aspect of the scientific contribution of USS.
- Heuristic evaluation from Nielsen (1994) is only focused on usability not considering the security aspects of a system. Also heuristic evaluation does not recommend any solutions to the problems only point them out.
- Cognitive walkthrough is not also designed to entail the security aspects of a system, only usability which USS does.
- Computer Security Design Principles (Saltzer and Schroeder, 1975) does consider only security aspects.
- Design guidelines for security management systems (Chiasson *et al.*, 2007) does consider only security aspects.
- Guidelines and Strategies for Secure Interaction Design (Yee, 2005) consider usability and security aspects but it is focused on general principles.
- Design Principles and Patterns for Aligning Security and Usability (Garfinkel, 2005) consider usability and security aspects but it is focused on general principles.
- Criteria for Security Software to be Usable (Whitten and Tygar, 1998) consider usability and security aspects but it is focused on general principles.

2. A GOMS model developed as a cognitive task analysis tool to understand user interaction related to the computer sciences and cognitive processes involved in the most representative user authentication methods such as Password/PINs (wired network-based task), One-Time-Passwords (wireless/token network-based task), Out-of-Band Authentication (wired and wireless network-based task), and Biometrics (wired network and electronic access control-based task). The general guidelines mentioned above do not address formally cognitive aspects of user authentication.
3. GOMS (design) artifacts which supported the cognitive task analysis work such as total learning and execution time measures for each of the user authentication methods as referred in item 2 above.
4. A refined and practical definition of usable security considering resources and costs.
5. A new definition of *Security Scenario* encompassing two new types such as Tangible Security Scenario (TSS) and Intangible Security Scenario (ISS).
6. A set of methods expressed in a programming-like language GLEAN (GOMS Language Evaluation and Analysis) for accomplishing goals for the following tasks scenarios: *Check Business E-mail* (Username and Password), *Update the SecurID authenticator UI specification* (One-Time-Passwords (OTP)), *Make an electronic funds transfer* (Out-Of-Band Authentication (OOBA)), and finally *Access a file on a personal laptop* (fingerprint recognition).
7. A Cognitive Model of User Authentication (CMUA) for understanding how and what cognitive processes are involved in each of the user authentication methods mentioned in item 2 above. The general guidelines mentioned in 1 do not again address formally cognitive aspects of user authentication.
8. A set of user authentication use cases used as basic use cases for a MTM.

9. A comparative analysis of user authentication methods taking into consideration a diverse spectrum of attributes such as definition, advantages, disadvantages, security, usability, human versus automatism, input process time, industrial application, and privacy issues. The general guidelines mentioned in 1 do not present any formal comparative analysis related to user authentication methods.
10. An Authentication Risk Assessment matrix to acknowledge and understand the main threats and security vulnerabilities related especially to online user authentication.
11. A comprehensive literature review of usable security related to security guidelines and user authentication methods.

6.3 Practical Observations on the Impact of USS in Corporate and Academic Environments

This thesis's author has been working on Computer Security especially in user authentication for the past 8 years within the following companies and institutions:

- RSA Security, the Security division of EMC, Bedford, MA (USA).
- VeriSign Inc., Mountain View, CA (USA).
- Center for Research and Analysis of Organizations (CIRANO), Montreal, Quebec (Canada).
- Rogers Communications, Montreal, Quebec (Canada).
- University of Quebec at Montreal and Concordia University within the doctoral program framework – Cognitive Computing, Quebec (Canada).
- University of Montreal within the master program framework – Master of Science in Electronic Commerce (specialization in computer science), Quebec (Canada).

In my point of view, working in the marketplace specifically in very centered authentication company and at the same time following a doctoral program which is

in certain way related to the thesis's subject is a crucial aspect on understanding the subject matter, user authentication, and having access to different resources and information that would be impracticable if I was only restricted to the academic program. My professional experience is translated implicitly and explicitly in this thesis through the i) demonstration of the OTP authentication method that will be performed during my oral defense; ii) the incorporation of questions into the questionnaire used for a formal usability testing of the OTP authentication method at RSA as mentioned in Chapter 6; iii) the participation in a demo regarding the OTP USB fingerprint authentication method at RSA; iv) the access to numerous technical documentation, development data, and usability tests sessions for authentication methods, and authentication manager software targeted to administrators (RSA Security Console) and corporate end-users (RSA Self-Service Console); and finally v) the different types of users involved in user authentication systems (e.g. Super Admin, corporate users, etc.) which is vital for the comprehension of the cognitive and corporate processes aspects.

But how this thesis' author envisions USS in the practical real world? Performing the USS inspection method User Experience Designers while evaluating the usability review questions may come across issues that would not be apparent if they were not also evaluating the security review questions at the same time. An example is the following:

- Usability review question: *7.4 If the MTM offers a silent alarm feature, is PIN number reversal used providing faster recall?* Security review question: *Has the system's owner security policy a clause that states that PIN number reversal must only be used for emergency?* Here it is clear that a PIN security policy should be enforced while providing ease of use to users. Also the reverse PIN functionality being enforced avoid the opening of bugs they will be able to reduce the number of bugs within a CD software release since issues can be brought early in the design phase.

By analyzing and answering concurrently the usability and security review questions force User Experience or Security designers to think in the process as a whole (usability and security) not a part of the whole (usability or security). This also forces them to initiate – or trigger – a solution in their minds for the questions/issues. In addition, while evaluating the review questions User Experience Designers will be able to anticipate the identification of potential bugs earlier in the authentication method design phase that would occur when the product is handed-off and released to the market. This is a very important aspect of the authentication method development life cycle which can represent benefits in terms of improved product reliability, greater business, reduced customer support calls, smaller releases product life cycle, decrease of Quality Engineering (QE) work (i.e., reduction of the number of bugs to be opened against the product), an finally enhanced user interaction (i.e. finding bugs earlier can force designers to review and improve authentication method functionalities with developers, project and product managers). Another important aspect is that companies, in general, are having difficulty fitting the traditional usability testing into their project plans and budgets when working on an Agile software development process (SCRUM framework). Agile software development refers to a set of software development methodologies derived from iterative development, where requirements and solutions develop through cooperation between self-organizing cross-functional teams. Agile is considered a “lightweight method” (especially when compared to the Waterfall method) which is very appropriate in today's dynamic business environment. SCRUM is an iterative, incremental framework for Agile. Using USS can be very appropriate in Agile since one (or more) evaluator can perform the inspection method to find out particularly major interface issues, speed development, and save user testing logistics time, planning, and money. The USS can be applied through a Spike (i.e. a timeboxed investigation, not the original user story), or Non-functional task (e.g., a performance-related issue, a reliability issue, etc.) within a two-week Sprint.

In the academic environment, the USP methodology design artifacts, described in Appendix D, can contribute to the research cycle by providing tools for the requirements and design phase of any software application master or doctoral project. For instance, within the Cognitive Computing program, the USS inspection method could be applied to the study and/or design of a Learning Management Systems (LMS) software application for the administration, documentation, tracking, and reporting of training programs, classroom and online events, e-learning programs, and training content. Another application of the design artifacts would be to be used as learning tools for teaching HCI.

The USP design artifacts can also help in avoiding hurried statements that cannot be sustained by objective data. For example, in GOMS Operators like DETERMINE-POSITION and CLICK-MOUSE BUTTON have been empirically validated at the keystroke-level, thus models that use those operators can generate reliable quantitative estimates. Another example where the design artifacts can be useful in the academic environment is in the evaluation of accessibility in an application: a task that can be carried out in one minute using a keyboard, might need 5 to 10 minutes using a mouse. Many disabled users favor to handle web applications, using the keyboard, given that it is often faster than using a mouse. As researchers are more focused on the conceptualization and modeling activities in a research project, USP design artifacts (especially 4, 5, and 6) can help in the understanding of concepts in a greater extent. Also, the CMUA can be adapted to the study and/or simulate other security mechanisms such as the True 128-bit Extended Validation SSL¹⁴⁶ from VeriSign which provides a simple and trustworthy way to establish trust online by displaying a green address bar with the name of the organization that owns the SSL certificate and the name of the Certificate Authority that issued it. The green bar shows site users that the transaction is encrypted and the

¹⁴⁶ VeriSign, Inc. June 1, 2010 <<http://www.verisign.com/ssl/buy-ssl-certificates/extended-validation-pro-ssl-certificates/index.html>>

organization has been authenticated in accordance with rigorous industry standard. The CMUA can then demonstrate how the process of recognizing and reasoning about the green bar can happen in the user's mind. To wrap up, these examples are only a few contributions that USP methodology can provide to the academic environment, but it is not limited to, and it is not an exhaustive list.

The lessons learned when developing this research work were the following:

- Usable security is critical to the effective adoption and deployment of user authentication methods.
- The development of a user authentication method, irrespective of being a software or hardware authenticator or biometrics needs to include usable security at early stage as part of Requirements and Design phases.
- A user authentication method can be adapted to a computer system's infrastructure already in place in an organization. They depend on environment and implementation factors, and also how well that product fits into the existing IT infrastructure in today's modern network environments which tend to be diverse in their requirements.
- The choice of an authentication method depends also on industrial norms, legal and business needs such as the environmental characteristics of the electronic communication (e.g. online shopping, authentication manager application, etc.).
- Security designers are required to address the diverse authentication needs of several and different users, including system administrators, employees, business partners, customers, and end-users.
- Security tools (including authenticators) have been developed, but their successful use in real applications is fairly limited because of their complexity, "hard-of-use", and the necessity of previous advanced technical knowledge on the part of end-users.

6.4 Limitations

Evaluation inspection methods in general do not offer potential solutions to the usable security problem. However this thesis provides the usability and security severity ratings in order to recommend practical solutions to usable security problems.

Also it is complex to summarize the findings in the inspection methods from multiple evaluators as they report problems differently and at different levels. There is also the issue of severity. Not all usable security problems are equal. Development teams need to be able to prioritize what problems get fixed according to the seriousness of the problem. There is currently no agreement on how to judge the severity of usability problems (e.g. bugs triage).

One question is how accurately these inspection methods predict problems that real user encounter? An early study found that heuristic reviews were better predictors than cognitive walkthroughs and guideline-based evaluations. This was compared to results from laboratory usability tests. However, none of these methods found more than 50% of the problems discovered in laboratory testing (Scholtz, 2003).

6.5 Future Work and Recommendations

First of all, a major goal of a future work is the development of a USS Web-based application which will be named USSWebApp for performing usable security assessment of user authentication methods. It will automate much of the time-consuming psychological and statistical work that is required for a deep interface analysis. An important and solid requirements and design work has been already completed with this doctoral thesis which will pave the way for the development and implementation phases of the USSWebApp.

Secondly, another goal is the development of a cognitive tool that would simulate the Cognitive Model of User Authentication (CMUA) presented in this dissertation. This cognitive tool could be used to simulate the user interaction with different authentication methods and demonstrate the feasibility of their implementation.

Finally, another important research work would specify the best combination among different and existing user authentication methods used in conjunction with different wired and/or wireless computer devices with regards to usable security and performance. This would provide organizations with robust data in order to implement or evaluate user authentication method(s).

User authentication is a foundation stone of IT security, one that is changing quickly. Several areas of technology development will have a major impact on user authentication over the next decade or even in a few years. According to Sasse (2004) and Manning (2009) from RSA Security, there are key challenges with which HCISec researchers will be confronted when making research on user authentication and should be addressed as follows:

- Thinking security by proposing intuitive or seamless solutions that respond to user behavior or what is known about the user;
- Different mechanisms for frequent and infrequently used passwords;
- Consideration of implications regarding physical and mental workload;
- Perceptions of and attitudes to security: ways of persuading and motivating users to be secure and change the image of security;
- Design of specific (goal-and risk-based) security policies that are enforced, and are seen to be enforced;
- Strong user authentication for end users in the cloud where identity can be carried along cloud services (i.e. Cloud Computing is a means to enhance capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. It consists of any

subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities).

6.6 The Future of User Authentication

Let us be practical: new authentication management solutions are needed to exploit authentication methods and bring stronger security at a reduced cost. A feasible novel approach would be artificial intelligence tools embedded within an authentication policy engine. This engine will assess for patterns of fraud during each user authentication request. Authentication attempts with a high likelihood of fraud might activate an alert to an intrusion detection system. Then the user authentication manager would dynamically establish an access control threshold, based on the kind of threats to a protected resource, and automatically pick the best available authentication methods for accessing the protected resource.

An important obstacle to strong user authentication in e-commerce is the false reject (rejecting the authentication attempt of a legitimate user). This will be probably overcome through use of intelligent applications that sense when a legitimate user is having difficulties authenticating. This *usable security* approach will help the legitimate user repeat a failed method or will present an alternative.

These new automated user authentication systems will be possibly implemented in the so called authentication portals which will provide extremely granular user authentication for accessing significant protected resources with different access privileges for different types of users.

- Future users will authenticate:
 - through trusted computing platforms, which will in turn represent the user to the network;
 - via RFID and other wireless devices, as logical and physical authentication technologies converge;

- based on what they know — and what they're able to do — in new and sophisticated ways;
- anonymously in many cases: as to their privileges, not necessarily their identities;
- with passwords sometimes, but the passwords will be better protected, and the authentication will be mutual

This thesis' author claims that user authentication will assume a chief new network control function. In an integrated and advanced network of multiple user devices (e.g. desktops, laptops, PDAs, cell phones, and so on) each carrying out numerous concurrent sessions with a dynamically allocated temporary client address, there will be an infinite mix of user authentication requirements to accurately and securely provide and bill services.

REFERENCES

- Abran, A., Khelifi, A., Suryin, W. and Seffah, A. 2003. "Usability Meanings and Interpretations in ISO Standards". *Software Quality Journal*, 11(4), p. 325-338.
- Accenture Consulting. 2004. "Guiding Principles - Security Framework". <http://www.biztech.pl/wbi/Anders_Carlstedt.pdf>. Retrieved on March 23, 2006.
- Adams, A. and Sasse, M. 1999. "Users Are Not the Enemy". *Communications of the ACM*, 42(12), p. 40-46.
- Ahn, L. V., Blum, M., Hopper, N.J. and Langford, J. 2003. "CAPTCHA: Telling humans and computers apart". *Advances in Cryptology, Eurocrypt '03, Lecture Notes in Computer Science*, 2656, p. 294-311.
- Allan, A. 2007. "WWW.Authentication: Why? When? What? Who?" *Gartner Identity Access Management Summit*. Gartner, Track Session. <<http://agendabuilder.gartner.com/iam2/webpages/SessionDetail.aspx?EventSessionId=811>>. Retrieved on February 2, 2007.
- ANSI. 1998. "Triple Data Encryption Algorithm Modes of Operation". ANSI X9.52-1998. Committee X9 (Financial Services). Online. <<http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+X9.52%3A1998>>. Retrieved on January 5, 2007.
- Anderson, R. C. 1977. "The Notion of Schemata and the Educational Enterprise: Discussion of the Conference. In R. C. Anderson, R. J. Spiro, and W. E. Montague (Eds.) *Schooling and the acquisition of knowledge*. p. 415-431. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Anderson, R. J. 2008. *Security Engineering: A Guide to Building Dependable Distributed Systems, Second Edition*. Indianapolis, IN: John Wiley & Sons, 1040 p.
- Angeli, A.D., Coventry, L., Johnson, G. and Coutts, M. 2003. "Usability and User Authentication: Pictorial Passwords vs. PIN". *Contemporary Ergonomics*. p. 253-258. London, England: Taylor and Francis.
- Apple Inc. 2008. "Apple Human Interface Guidelines". <<http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/OSXHIGuidelines.pdf>>. Retrieved on October 3, 2009.
- Atkinson, R. C. and Shiffrin, R. M. 1968. "Human memory: A proposed system and its control processes". In K. W. Spence and J. T. Spence (Eds.), *The psychology of learning and motivation: Advances in Research and Theory*, 2, p. 89-195.

- Baddeley, A. D. 1998. *Human Memory: Theory and Practice*. Boston, MA: Allyn and Bacon.
- Balfanz, D., Smetters, D.K. and Grinter, R.E. 2004. "In search of usable security: Five Lessons from the Field". *IEEE security and privacy*. 2(5), p. 19-24.
- Bank of America Corporation. 2009. "SiteKey® at Bank of America". <<http://www.bankofamerica.com/privacy/sitekey>>. Retrieved on April 13, 2009.
- Bastien, J.M.C and Scapin, D.L. 1993. "Ergonomic Criteria for the Evaluation of Human-Computer Interfaces". *Behaviour & Information Technology*, 16(4 and 5), p. 220-231.
- Bolle, R., Connell, J., Pankanti, S., Ratha N. and Senior A. 2004. *Guide to Biometrics*. New York, NY: Springer-Verlag.
- Bovair, S., Kieras, D. E., and Polson, P. G. 1990. "The acquisition and Performance of Text Editing Skill: A Cognitive Complexity Analysis". *Human-Computer Interaction*, 5(1), p. 48.
- Braz, C. and Aïmeur, E. 2003. "AuthenLink: A User-Centred Authentication System for a Secure Mobile Commerce". Master Thesis, University of Montreal, Montreal, QC. 101p.
- Braz, C. and Aïmeur, E. 2004. "AuthenLink: Authentication System for a Secure Mobile". *3rd International Workshop on Wireless Information Systems (WIS-2004)*. p. 114-126.
- Braz, C. and Aïmeur, E. 2005. "ASEMC: Authentication for a Secure Mobile Commerce". *RFID Journal*, White Papers, Security.
- Braz, C. and Robert, J.M. 2006. "Security Usability: The Case of User Authentication Methods". *18th French-Speaking Conference on Human Computer Interaction (HCI2006)*, p. 199-203.
- Braz, C., Seffah, A. and M'Raihi, D. 2007. "Designing a Trade-off between Usability and Security: A Metrics Based-Model". *Interact 2007: Socially Responsible Interaction, IFIP TC.13 IFIP Technical Committee on Human Computer Interaction*. p. 114-126.
- Braz, C., Poirier, P. and Seffah, A. 2010. "Integrating a Usable Security Protocol for User Authentication into the Requirements and Design Process". Doctoral Thesis, University of Quebec at Montreal, QC, Canada. 498 p.
- Biederman, I. 1987. "Recognition-by-Components: A Theory of Human Image Understanding". *Psychological Review Journal*, 94(2), p. 115-47.

- Carda, S., Moran, T.P. and Newell, A. 1980a. "The keystroke-level model for user performance time with interactive systems". *Communications of the ACM*, 23(7), p. 396-410.
- Card, S., Moran, T., and Newell, A. 1983. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Chiasson, S., van Oorschot, P.C. and Biddle, R. 2006. A Usability Study and Critique of Two Password Managers. *15th USENIX Security Symposium*. p. 1-16.
- Chiasson, S. and Biddle, R. 2007. Issues in User Authentication. *Workshop: security user studies: methodologies and best practices, ACM CHI*.
- Chiasson, S. , Biddle, R. and Somayaji, A. 2007. "Even Experts Deserve Usable Security: Design guidelines for security management systems". *Workshop on Usable IT Security Management (USM'07)*.
- Christey, S. 2007. "Unforgivable Vulnerabilities. Common Vulnerabilities and Exposures, Documents". The MITRE Corporation. <<http://cve.mitre.org/docs/docs-2007/unforgivable.pdf>>. Retrieved on May 25, 2008.
- CIO.com. 2009. "12 Top Popular Applications with Critical Security Vulnerabilities". <http://www.cio.com/article/477470/_Top_Popular_Applications_with_Critical_Security_Vulnerabilities>. Retrieved on June 9, 2009.
- Computing Technology Industry Association (CompTIA). 2002. "Committing to Security: A CompTIA Analysis of IT Security and the Workforce". Security survey.
- Corner, M.D. and Noble, B.D. 2002. "Zero-Interaction Authentication". *MOBICOM'02*. p.1-11.
- Cranor, L.F. and Garfinkel, S.L. 2005. *Security and Usability: Designing Secure Systems that People Can Use*. Sebastopol, CA: O'Reilly Media Inc.
- Computer Security Institute (CSI). 2008. "CSI - Computer Crime and Security Survey". <<http://i.cmpnet.com/v2.gocsi.com/pdf/CSIsurvey2008.pdf>>. Retrieved on July 5, 2009.
- Common Vulnerabilities and Exposures (CVE®). 2009. "Common Vulnerabilities and Exposures". <<http://cve.mitre.org/index.html>>. Retrieved on March 6, 2009.
- Dallas Semiconductor. 2006. "iButton: Touch the Future". <<http://www.maxim-ic.com/products/ibutton/>>. Retrieved on August 12, 2008.

- Desurvire, H. W. 1994. "Are usability inspection methods as effective as empirical testing?" In Nielsen, J., and Mack, R. L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons.
- Desurvire, H. W., Kondziela, J. M. and Atwood, M. E. 1992. "What is Gained and Lost When Using Evaluation Methods Other than Empirical Testing". In Monk, A., Diaper, D., and Harrison, M.D. (Eds.), *People and Computers*, 7, p. 89-102.
- Dhamija, R., Tygar, J.D. and Hearst, M. 2006. "Why Phishing Works". *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM Special Interest Group on Computer-Human Interaction. p. 581-590.
- Dhamija, R. and Perrig, A. 2000. "Déjà Vu: A User Study - Using Images for Authentication". *Proceedings of the 9th USENIX Security Symposium*, 4 (4).
- Dhamija, R. and Tygar, J.D. 2005. "The Battle Against Phishing: Dynamic Security Skins". *SOUPS'05: Proceedings of the 2005 symposium on Usable privacy and security*, p. 77-78.
- Diaper, D. and Stanton, N. 2003. *The handbook of task analysis for human-computer interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Diffie, W. and Hellman, M.E. 1976. "New Directions in Cryptography". *IEEE Transactions on Information Theory*. 22, p. 644-654.
- Dowell, John and Long, John. 1998. "Conception of the Cognitive Engineering Design Problem". *Ergonomics*, 41(2), p. 126-139
- EMC Corporation. 2006. EMC VPN Client (Version 4.8.02.0010) Software. Hopkinton, MA: EMC Corp.
- EMC Corporation. 2009. "Password Guidelines, Information Security Awareness, Supporting Materials".
- Federal Aviation Administration (FAA). 1998. "Report of the Computer-Human Interface Re-Evaluation of the Standard Terminal Automation Replacement System Monitor and Control Workstation". Human Factors Team. <<http://www.hf.faa.gov/docs/508/docs/STARS-mc.pdf>>. Retrieved on March 29, 2008.
- Federal Financial Institutions Examination Council (FFIEC). 2005. "Interagency Guidance on Authentication in an Internet Banking Environment". <http://www.ffiec.gov/ffiecinfobase/resources/info_sec/2006/frb-sr-05-19.pdf>. Retrieved on March 4, 2006.
- FIPS. 2002. "Security Requirements for Cryptographic Modules" FIPS PUB 140-2. Online Standard. <<http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>>. Retrieved on December 8, 2006.

- Flechais, I., Sasse, A.M. and Hailes, S.M.V. 2003. "Bringing security home: A process for developing secure and usable systems". *Workshop on New Security Paradigms*. p. 49-57. Ascona, Switzerland: ACM Press.
- Forrester Research, Inc. 2007. "Enterprise and SMB Security Survey, North America and Europe Q3 2007". <<http://www.forrester.com/ER/Research/Survey/Excerpt/0,,641,00.html>> Retrieved on June 2, 2007.
- Forsberg, M. 2003. "Why is Speech Recognition Difficult?". Chalmers University of Technology.
- Furnell, S., Papadopoulos, I. and Dowland, P. 2003. "A Long-Term Trial of Alternative User Authentication Technologies". *Information Management & Computer Security*, Emerald Group Publishing Limited, 12(2), p. 178-190.
- Garfinkel, S.L. 2005. "Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable". Doctoral Thesis, Massachusetts Institute of Technology, Cambridge, MA, 472 p.
- Gautam, N., Chinnam, R.B. and Singh, N. 2007. "Design reuse framework: a perspective for lean development". *Int. J. Product Development*, 4(5), p. 485-507.
- Gong, R. J. 1993. "Validating and refining the GOMS model methodology for software user interface design and evaluation." Doctoral Thesis, University of Michigan.
- Gong, R. and Elkerton, J. 1990. "Designing minimal documentation using a GOMS model: A usability evaluation of an engineering approach". *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. p. 99-10, New York, NY: ACM.
- Gong, R. and Kieras, D. 1994. "A Validation of the GOMS Model Methodology in the Development of a Specialized, Commercial Software Application". In B. Adelson, S. Dumais & J. Olson (Eds.), *ACM CHI'94 Conference on Human Factors in Computing Systems*, 1, p. 351-357. New York, NY: ACM Press.
- Gray, W. D., John, B. E. and Atwood, M. E. 1993. "Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance". *Human-Computer Interaction*, 8 (3), p. 237-209.
- Grudin, J. 1989. "The case against user interface consistency". *Communications of the ACM*, 32 (10), p. 1164-1173.
- Harris, D. 2007. "Engineering Psychology and Cognitive Ergonomics". *7th International Conference on Engineering Psychology and Cognitive Ergonomics (EPCE 2007)* in the framework of the *12th International Conference on Human-Computer Interaction (HCI 2007)*. 4562, Springer.

- Holcombe, B. Government. 2004. "Smart Card Handbook. Smart Card Standards and Interoperability". <<http://www.idmanagement.gov/smart/information/smartcardhandbook.doc>>. Retrieved on July 31, 2005.
- Hom, J. 1998. "The Usability Methods Toolbox". <<http://usability.jameshom.com>>. Retrieved on March 14, 2005.
- Hackos, J.T. and Redish, J.C. 1998. *User and Task Analysis for Interface Design*. New York, NY: John Wiley & Sons.
- Hollingsed, T. and Novick, D.G. 2007. "Usability Inspection Methods after 15 Years of Research and Practice". *ACM 25th International Conference on Design of Communication, SICDOC'07*. p. 249-255.
- Hornbæk, K. and Frøkjær, E. 2004. "Two psychology-based usability inspection techniques studied in a diary experiment". In *Proceedings of the Third Nordic Conference in Human Computer Interaction - NordiCHI 2004*, p. 3-12.
- Howell, W. C. and Cooke, N. J. 1989. "Training the Human Information Processor: A look at Cognitive Models". In *J. Goldstein (Ed.), Training and Development in Work Organizations: Frontier Series of Industrial and Organizational Psychology*, 3, p. 121-182, New York: Jossey Bass.
- Hutchins, E. 1995. *Cognition in the Wild*. Cambridge, Mass: MIT Press.
- I/O Software Inc. 2005. "Levels of Security". <<http://www.iosoftware.com/pages/Support/Authentication%20Basics/Selection%20Process/index.asp>>. Retrieved on June 15, 2006.
- IBM, Inc. 2006. "A usability teaching tool to demonstrate poor interface design: EasyChart". <http://www-306.ibm.com/ibm/easy/eou_ext.nsf/publish/3072>. Retrieved on July 8, 2006.
- Idera. 2010. *Idera SharePoint. Backup and Recover SharePoint* (Version 2.7). Software. Houston: Idera.
- IEEE. 1998. "IEEE Standard for Software Quality Metrics Methodology". IEEE Std 1061-1998. Online Standard. <http://standards.ieee.org/reading/ieee/std_public/description/se/1061-1992_desc.html>. Retrieved on March 13, 2006.
- IEEE. 1999. "IEEE Standard for Information technology - Telecommunications and information exchange between systems". Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 1: High-speed Physical Layer in the 5 GHz band". IEEE Std 802.11-1999. Online Standard. <http://ieeexplore.ieee.org/xpl/freeabs_all.js?arnumber=1389197>. Retrieved on July 5, 2005.

- ISO. 1998. "Ergonomic requirements for office work with visual display terminals (VDTs -Part 11: Guidance on Usability". ISO 9241-11. 1998. Online Standard. <http://www.iso.org/iso/catalogue_detail.htm?csnumber=16883>. Retrieved on November 30, 2006.
- ISO. 1999. "Human-Centred Design Processes for Interactive Systems". ISO 13407: 1999. Online Standard. <http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197>. Retrieved on May 24, 2007.
- ISO. 2002. "Banking - Personal Identification Number (PIN) management and security – Part 1: Basic principles and requirements for online PIN handling in ATM and POS systems". ISO 9564-1:2002. Online Standard. <http://www.iso.org/iso/catalogue_detail.htm?csnumber=29374>. Retrieved on June 4, 2007.
- ISO/IEC. 2001. Software engineering - Product quality - Part 1: Quality model, ISO/IEC 9126-1:2001 Edition 1; 2003 Software engineering - Product quality - Part 2: External metrics, ISO/IEC TR 9126-2:2003 Edition 1; 2004 Software engineering - \ Product quality–Part 4: Quality in use metrics, ISO/IEC TR 9126-4:2004 Edition 1. ISO 9564-1:2002. Online Standard. <http://www.iso.org/iso/catalogue_detail.htm?csnumber=22749>. Retrieved on November 11, 2006.
- James, W. 1890. *Principles of Psychology*. New York, NY: Henry Holt and Company.
- Janney Montgomery Scott LLC. 2005. "Janney Montgomery Scott Report". <<http://www.jmsonline.com/jms/>> Retrieved on July 1, 2005.
- Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. 1991. "User interface evaluation in the real world: A comparison of four techniques". *Proceedings ACM CHI'91 Conference*, p. 119-124.
- Jøsang, A. Zomai, M. and Suriadi, S. 2007. "Usability and Privacy in Identity Management Architectures". *Proceedings of the Fifth Australasian Symposium on ACSW Frontiers*. Australasian Information Security Workshop: Privacy Enhancing Technologies (AISW) session, 68, p. 143-152.
- Jøsang, A. and Patton, M. A. 2003. "User Interface Requirements for Authentication of Communication". *Proceedings of the Fourth Australasian user interface conference on User interfaces 2003*. 18, p.75-80.
- Jøsang, A., Zomai, M., McNamara, J. and Grandison, T. 2007. "Security Usability Principles for Vulnerability Analysis and Risk Assessment". *Proceedings of the Annual Computer Security Applications Conference (ASAC 2007)*. p. 269-278.

- John, B.E. and Kieras, D.E. 1994. "The GOMS Family of Analysis Techniques: Tools for Design and Evaluation". Human-Computer Interaction Institute Technical Report CMU. <<ftp://www.eecs.umich.edu/people/kieras/GOMS/John-Kieras-TR94.pdf>> Retrieved on January 6, 2004.
- John, B. E. and Kieras, D. E. 1996. "Using GOMS for user interface design and evaluation: Which technique?". *ACM Transactions on Computer-Human Interaction*, 3 (287), p.319.
- John, Bonnie and Kieras, David E. 1996a. "The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast". *ACM Transactions on Computer-Human Interaction*, 3 (4), p. 320-351.
- Jain, A. K. 2004. "Biometric Recognition: How do I Know Who You Are?". *Proceedings of IEEE 12th Signal Processing and Communications Applications Conference*. 3540, p. 3-5.
- Kantner, L. and Keirnan, T. 2003. "Field Research in Commercial Product Development". In *Proceedings of Usability Professionals Association (UPA) 2003: Ubiquitous Usability - Advanced Topic Seminars*. Scottsdale, AZ.
- Karat, C., Campbell, R. and Fiegel, T. 1992. "Comparison of empirical testing and walkthrough methods in user interface evaluation". *Proceedings of CHI'92*, p. 397-404.
- Karat, J. and Bennett, J. L. 1991. "Working within the design process: Supporting effective and efficient design". In Carroll, J. M. (Ed.), *Designing interaction: Psychology at the human-computer interface*, p. 269-285. Cambridge, England: Cambridge University Press.
- Kjeldskov, J., Skov, M.B. and Stage, J. 2010. "A longitudinal study of usability in health care: Does time heal?". *International Journal of Medical Informatics*. 79(6), p. 135-143.
- Kerckhoffs, A. 1883. "La cryptographie militaire/Military Cryptography". *Journal des sciences militaires*, 9, p.161-191.
- Kieras, D. E. 1996. "A Guide to GOMS Model Usability, Evaluation using NGOMSL". <ftp://ftp.eecs.umich.edu/people/kieras/GOMS/NGOMSL_Guide.pdf> Retrieved on February 3, 2009.
- Kieras, D. E. 2006. "A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN4". University of Michigan, Ann Harbor, MI.
- Kieras, D. 2001. "Using the Keystroke-Level Model to Estimate Execution Times". <<ftp://ftp.eecs.umich.edu/people/kieras/GOMS/KLM.pdf>> Retrieved on February 3, 2009.

- Kieras, D.E., Wood, S.D., Abotel, K. and Hornof, A. 1995. "GLEAN: A Computer-Based Tool for Rapid GOMS Model Usability Evaluation of User Interface Designs". *UIST'95 Proceedings of the ACM Symposium on User Interface Software and Technology*. p. 91-100.
- Kieras, D., Wood, S. and Meyer, D. 1997. "Predictive Engineering Models Based on the EPIC Architecture for a Multimodal High-Performance Human-Computer Interaction Task". *ACM Transactions on Computer Human Interaction*. 4 (3), p. 230-275. ACM: New York.
- Kieras, D. E. and Bovair, S. 1984. "The acquisition of procedures from text: a production-system analysis of transfer of training". *Journal of Memory and Language*. 25, p. 507-524.
- Kieras, D. E., and Polson, P. G. 1985. "An approach to the formal analysis of user complexity". *International Journal of Man-Machine Studies*, 22, p. 365-394.
- Kirakowski, J. 2001. "SUMI Questionnaire". Human Factors Research Group, University College Cork, North Mall, Cork, Ireland.
- Kosslyn, S. M. 1980. *Image and Mind*. Cambridge, MA: Harvard University Press
- Kosslyn, S.M. 1983. *Ghosts in the Mind's Machine: Creating and Using Images in the Brain*. New York, NY: Norton.
- Laird, J. E. 2008. "Frontiers in Artificial Intelligence and Applications". *Proceedings of the 2008 conference on Artificial General Intelligence*, 171, p. 224-235, Amsterdam: IOS Press.
- Laird, J. E. and Congdon, C.B. 2009. "SOAR Users Manual". University of Michigan, MI <<http://soar.googlecode.com/svn/tags/Soar-Suite-9.3.0/Core/Documentation/SoarManual.pdf>> Retrieved on July 22, 2009.
- Laird, J. E., Newell, A. and Rosenbloom, P.S. 1987. "SOAR: An Architecture for General Intelligence". *Artificial Intelligence*, 33 (1), p. 64.
- Law, E.L.C., Hvannberg, E., Cockton, G.(Eds.). 2008. *Maturing Usability. Quality in Software, Interaction and Value*. London, UK: Springer-Verlag, 469 p.
- Lecerof, A. and Paternò, F. 1998. "Automatic Support for Usability Evaluation". *IEEE Transactions on Software Engineering*. 24(10), p. 863-888.
- Maffezzini, I.P. 2006. "Interfaces et Frontières ou Comment Passer de L'autre Côté sans se Faire Mal". University of Quebec at Montreal, Canada.
- Maltoni, D., Maio, D., Jain, A. K. and Prabhakar, S. 2003. *Handbook of Fingerprint Recognition*. New York, NY: Springer-Verlag.
- Manning, B. 2009. "Highlights and Takeaways". *RSA Conference 2009 Internal Presentation*, Slide 5.

- Marr, D. and Nishihara, H. K. 1978. "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes". *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 200 (1140), p. 269-294.
- Microsoft Corporation. 2005. "Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0.". <<http://msdn.microsoft.com/en-us/library/aa480547.aspx>> Retrieved on September 29, 2006.
- Microsoft Corporation. 2009. "Windows 2000 Server. Security with Smart Cards". <<http://technet.microsoft.com/en-us/library/cc962052.aspx>> Retrieved on July 22, 2009.
- Miller, G. A. 1956. "The magical number seven, plus or minus two: Some limits on our capacity for processing information". *Psychological Review*, 63, p. 81-97
- The MIT Kerberos Team. 2006. "Kerberos: The Network, Authentication Protocol". <<http://Web.Mit.Edu/Kerberos>> Retrieved on March 4, 2007.
- Mitnick, K. and Simon, W. 2002. *The Art of Deception: Controlling the Human Element of Security*. Chichester, England: John Wiley & Sons.
- Molich, R. and Nielsen, J. 1990. "Ten Usability Heuristics". <http://www.useit.com/papers/heuristic/heuristic_list.html> Retrieved on April 27, 2006.
- Naur, P. 1995. *Knowing and the Mystique of Logic and Rules*. Dordrecht, Netherlands: Kluwer Academic Publishers.
- Naur, P. 1998. "Human knowing, language, and discrete structures". In Naur, P. 1992, *Computing: A Human Activity*, p. 518-535, ACM Press/Addison Wesley.
- Naur, P. 2000. "CHI and Human Thinking". *Proceedings of the First Nordic Conference on Computer-Human Interaction - NordiCHI*. p. 23-25. Also in [50-2005] Appendix 1, p. 199-207.
- National Computer Security Center (NCSC). 1983. "Guide to Understanding Identification & Authorization in Trusted Systems". 1(5), p. 235-479.
- Neisser, U. 1964. "Visual search". *Scientific American*, 210(6), p. 94-102.
- Newell, A., Rosenbloom, P.S. and Laird, J. E. 1987. "SOAR: An Architecture for General Intelligence". *Artificial Intelligence*. 33(1) p. 1-64.
- Newell, A. and Simon, H. 1997. "Computer Science and Empirical Inquiry: Symbols and Search". In *Mind Design II*, John Haugeland (Ed.), 81-110. Cambridge, MA: MIT Press.
- Nielsen, J. 1992. "Finding Usability Problems through Heuristic Evaluation". *Proceedings of ACM Computer Human Interaction (CHI'92)*. p. 373-380.

- Nielsen, J. 1993. *Usability Engineering*. New Jersey, NY: AP Professional.
- Nielsen, J. 1994. "Heuristic Evaluation". In Nielsen, J. and Mack, R.L. (Eds.), *Usability Inspection Methods*, New York, NY: John Wiley & Sons.
- Nielsen, J. 1994a. "Heuristic Evaluation, How to Conduct a Heuristic Evaluation". http://www.useit.com/papers/heuristic/heuristic_evaluation.html Retrieved on October 9, 2003.
- Nielsen, J. 2000. "Security & Human Factors". Jakob Nielsen's Alertbox. <http://www.useit.com/alertbox/20001126.html> Retrieved on April 10, 2004.
- Nielsen, J. 2006. "Severity Ratings for Usability Problems". <http://www.useit.com/papers/heuristic/severityrating.html> Retrieved on January 13, 2007.
- National Institute of Standards and Technology (NIST). 2001. "Announcing the Advance Encryption Standard (AES)". Online Standard. <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. Retrieved on June 11, 2007.
- Norman, D. A. 1983. "Some Observations on Mental Models". In Gentner, D. and Stevens, A.L., Eds. *Mental Models*. Hillsdale, N.J.: L. Erlbaum Associates, p. 7-14.
- Norman, Donald A. 1988. *The Design of Everyday Things*. New York, NY: Doubleday.
- Nuxoll, A. and Laird, J. 2004. "A Cognitive Model of Episodic Memory Integrated With a General Cognitive Architecture". *International Conference on Cognitive Modeling*. Pittsburgh, PA: p. 22-225.
- O'Regan, J.K. and Noë, A. 2001. "A Sensorimotor Account of Vision and Visual Consciousness". *Behavioural and Brain Sciences*, 24, p. 939-1031, Cambridge, MA, Cambridge University Press.
- Olson, J. R. and Olson, G. M. 1990. "The growth of cognitive modeling in human-computer interaction since GOMS". *Human-Computer Interaction*, 5, p. 221-265.
- Open Web Application Security Project (OWASP). 2009. "SQL Injection Vulnerabilities – SQL Injection Prevention Cheat Sheet". http://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet Retrieved on July 28, 2009.
- Passfaces Corporation. 2009. "Passface® Authentication Method". <http://www.passfaces.com> Retrieved on January 9, 2009.

- Penn, J. 2008. "The State of Enterprise IT Security 2008 to 2009". Business Data Services North America and Europe. Forrester Research survey.
<http://www.forrester.com/rb/Research/state_of_enterprise_it_security_2008_t_o/q/id/47857/t/2> Retrieved on February 2, 2009.
- Perks, M. 2003. "Best practices for software development projects". developerWorks, Technical Library. <http://www.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html>
- Perlman, G. 2006. "Web-Based User Interface Evaluation with Questionnaires". <<http://www.acm.org/~Eperlman/question.html>> Retrieved on June 12, 2006.
- Pierotti, D. 1996. "Usability Techniques - Heuristic Evaluation: A System Checklist", Xerox Corporation, Version 1.0. <<http://www.stcsig.org/usability/topics/articles/he-checklist.html>> Retrieved on March 12, 2004.
- Poirier, P., Hardy-Vallée, B. and DePasquale, J. F. 2005. "Embodied Categorization". In Henri, C. and Claire, L. (Eds.), *Handbook of Categorization in Cognitive Science*. p. 739-765. Oxford: Elsevier Science.
- Proctor, Robert W. and Vu, Kim-Phuong L. 2005. *Handbook of Human Factors in Web Design*. p. 396. Mahwah, NJ: L. Erlbaum Associates.
- Renaud, K.: 2003. "Quantifying the Quality of Web Authentication Mechanisms - A Usability Perspective". *Journal of Web Engineering*. 3(2), p.95-123.
- Richardson, D.J. 2000. "Cognitive Walkthroughs, Human-Computer Interaction". ICS 121 Topic 6: Cognitive Walkthroughs. University of California, CA, United States.
- Ritter, F. E.: 2004. "Choosing and getting started with a cognitive architecture to test and use human-machine interfaces". *MMI-Interaktiv Journal*. 7(17), p. 37.
- RSA, The Security Division of EMC. 2010. Bedford, Massachusetts (USA).
- RSA, The Security Division of EMC. 2010a. "Enterprise Single Sign-On. RSA Laboratories". <<http://www.rsa.com/rsalabs/node.asp?id=2541>> Retrieved on May, 2010.
- Paternò, F. and Santoro, C. 2008. "Remote Usability Evaluation: Discussion of a General Framework and Experiences". In Law, E.L.C., Hvannberg, E., Cockton, G.(Eds.). *Maturing Usability. Quality in Software, Interaction and Value*. London, UK: Springer-Verlag, 469 p.
- Saltzer, J. H. and Schroeder, M. D. 1975. "The Protection of Information in Computer Systems". *Proceedings of the IEEE*, 63(9), p. 1278-1308.
- Sasse, M.A. 2004. "Usability and Security - Why we need to look at the big picture". ISS 2004, University College London, UK.

- Sasse, M.A., Brostoff, S. and Weirich, D. 2001. "Transforming the "weakest link" - a human/computer interaction approach to usable and effective security". *BT Technology Journal*, 19 (3), p. 122-131.
- Sasse, M.A. and Flechais, I. 2005. "Usable Security – Why Do We Need It? How Do We Get It?". In (Cranor and Garfinkel, 2005), Chapter Three, p. 13.
- Scambray, J., Shema, M. and Sima, C. 2006. *Hacking exposed: Web Applications*. San Francisco, CA: McGraw-Hill.
- Scholtz, J. 2003. "Usability Evaluation". National Institute of Standards and Technology. 8 p.
- Sauro, J. and Kindlund, E. 2005 "A Method to Standardize Usability Metrics into a Single Score". <<http://www.measuringusability.com/SUM/p482-sauro.pdf>> Retrieved on May 8, 2006.
- Sauro, J. 2006. "Sample Size for Discovering Problems in a UI, Measuring Usability Website". <http://www.measuringusability.com/samplesize/problem_discovery.php> Retrieved on April 5, 2006.
- Schneiderman, B. 1998. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Third Edition. Reading, MA: Addison-Wesley.
- Shneiderman, B. and Plaisant, B. 2005. *Designing the User Interface: Strategies for Effective Human Computer Interaction*. Fourth Edition. Boston, MA: Pearson Addison-Wesley.
- Schrepp, M. and Fischer, P. 2006. "A GOMS Model for Keyboard Navigation in Web Pages and Web Applications". *ICCHP Conference 2006*. <<http://www.axistive.com/a-goms-model-for-keyboard-navigation-in-web-pages-and-web-applications.html>> Retrieved on November 10, 2007.
- Secure Computing: 2004. "SafeWord® PremierAccess, Strong Authentication." <<http://www.securecomputing.com/index.cfm?skey=643>> Retrieved on October 15, 2006.
- Seffah, A., Abran, A., Suryn, W., Khelifi, A., Rilling, J., and Robert, F. 2003. "Consolidating the ISO Usability Models". Concordia University, Montreal, Canada.
- Seffah A., Donyaee M., Kline R., and Padda H.K. 2006. "Usability Metrics: A Roadmap for a Consolidated Model". *Journal of Software Quality*, 14 (2).
- Seffah, A. and Donyaee, M. 1998. "Metrics and Measurement of Usability." *International Encyclopedia of Ergonomics and Human Computer Interaction*. Second Edition, vol. 1. CRC Press.

- Seffah A. and Metzker, E. 2004. "The Obstacles and Myths of Usability and Software engineering: Avoiding the Usability Pitfalls Involved in Managing the Software Development Life Cycle". *Communications of the ACM*, 47(12), p. 71-76.
- Shadow. 2008. "Authentication Risk Analysis Worksheet Answers".
<<http://www.scribd.com/riskAnalysis.html>>. Retrieved on January 2005.
- Smith, R.E. 2002. *Authentication: From Passwords to Public Keys*. Boston, MA: Addison-Wesley.
- Smith, S. L. and Mosier, J. N. 1986. "Guidelines for Designing User Interface Software". ESD-TR, 86(27), p. 34.
- Sperling, G. 1960. "The Information Available in Brief Visual Presentations". *Psychological Monographs*, 74(498), p. 1-29.
- Stajano, F. 2003. "Security For Whom? The Shifting Security Assumptions of Pervasive Computing". In M. Okada et al. (Eds.): *Software Security—Theories and Systems*, 2609, Berlin, Germany: Springer-Verlag.
- Stiegler, M., Karp, A.H., Yee, K.P. and Mark Miller. 2004. "Polaris: Virus Safe Computing for Windows XP". <<http://www.hpl.hp.com/techreports/2004/HPL-2004-221.pdf>> Retrieved on May 24, 2006.
- Stuerzlinger, W., Chapuis, O., Phillips, D. and Roussel, N. 2006. "User Interface Façades: Towards Fully Adaptable User Interfaces". *Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology-UIST '06*. p. 309-318. New York, NY: ACM Press.
- Stufflebeam, D.L. 2000. "Guidelines for Developing Evaluation Checklists: The Checklists Development Checklist (CDC)." <http://www.wmich.edu/evalctr/checklists/guidelines_cdc.pdf> Retrieved on November 10, 2006.
- Sy, Desirée. 2009. "Usability Over Time: Longitudinal Research Studies". In *Designing the User Experience at Autodesk - Insights on innovation, inspiration, and the practice of design*. <<http://dux.typepad.com/dux/2009/05/usability-over-time-longitudinal-research-studies.html>>. Retrieved on July 22, 2010.
- Tulving, E. and Schacter, D.L. 1990. "Priming and Human Memory Systems." *Science*, 247(4940), p. 301-306.
- United Airlines. 2007. "Delayed and damaged baggage". Baggage Services/Lost and Found. <<http://www.united.com/page/article/0,6722,1037,00.html>> Retrieved on March 4, 2008.

- Usability Professionals Association (UPA). 2010. "Usability - Body of Knowledge, Methods, Heuristic Evaluation (How To), The Usability Body of Knowledge". <<http://www.usabilitybok.org/methods/p275?section=how-to>> Retrieved on May 5, 2010.
- Valentine T: 1998. "An Evaluation of the Passfaces™ Personal Authentication System". Technical Report, Goldsmiths College, University of London, UK.
- Van Someren, M. W., Barnard, Y. F., and Sandberg, J. A. C. 1994. *The Think Aloud Method: A practical guide to modeling cognitive processes*. London, UK: Academic Press.
- Vasco Data Security International, Inc. 2010. <<http://www.vasco.com/>> Retrieved on May 12, 2009.
- VeriSign, Inc. 2010. "VeriSign® Unified Authentication". <http://www.verisign.com/static/DEV016111.pdf>> Retrieved on May 11, 2010.
- Voyles, R.M., Jr., and Khosla, P.K. 1995. "Tactile Gestures for Human/Robot Interaction". *Intelligent Robots and Systems 95 - Human Robot Interaction and Cooperative Robots*. *Proceedings. 1995 IEEE/RSJ International Conference*. 3(7), p. 13.
- Wickens, C.D. 1992. *Engineering Psychology and Human Performance*. Second Edition Chapter 5, p. 167-210. Harper Collins.
- Williams, K.E. and Voigt, J. R. 2004. "Evaluation of a Computerized Aid for Creating Human Behavioral Representations of Human-Computer Interaction". *Human Factors*, 46(2), p. 288-303.
- Whiteside, J. Bennett, J. and Holtzblatt, K. 1988. *Usability Engineering: Our Experience and Evolution. Handbook of Human Computer Interaction*. New York: North Holland, p. 791-817.
- Whitten, A. and J. Tygar, J. D. 1998. "Usability of Security: A Case Study". CMU-CS-98-155, Carnegie Mellon University, Pittsburgh, PA.
- Whitten, A. and J. Tygar. 1999. "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0." *Proceedings of the 8th USENIX Security Symposium*. p. 169-184.
- Woodward, J.D, Orlans, N.M. and Higgins, P.T. 2003. *Biometrics - Identity Assurance in the Information Age*. Berkeley, CA: McGraw-Hill/Osborne Media. 432 p.
- Wu, M., Miller, R.C. and Garfinkel, S. 2006. "Do Security Toolbars Actually Prevent Phishing Attacks?" Massachusetts Institute of Technology, Cambridge, MA.
- Yee, K.P. 2004. "Aligning Security and Usability." *IEEE Security and Privacy*, 2(5), p. 48-55.

- Yee, K.P. 2005. "Guidelines and Strategies for Secure Interaction Design." In (Cranor and Garfinkel, 2005), Chapter Thirteen, p. 253.
- Zhang, D.D. 2004. *Palmprint Authentication*. p. 204. Norwell, MA: Kluwer Academic Publishers.
- Zurko, M. and Simon, R. 1996. "User-centered security." *Proceedings of the UCLA Conference on New Security Paradigms Workshops*. p. 27-33.

APPENDIX A

COMPARATIVE ANALYSIS OF USER AUTHENTICATION METHODS

A.1 Introduction

This appendix presents details of the comparative analysis of user authentication methods presented in Chapter 2. It specifies some definitions and abbreviations used in the following tables. To describe the authentication methods attributes, subjective rating scales have been used for Security and Usability as follows:

- Total Transaction Time (seconds) corresponds to the time for a single user to present the biometric (acquisition time), processing time, and, optionally, might include entry of a PIN or user identifier (Woodward *et al.*, 2003).
- "Security" and "Usability" range from 1=Minimum to 5=Maximum in order to measure the degree of severity issues related to each authentication method.
- "Automation versus Human" range from 1=Human is better; 5=Machine is better.
- "Accuracy" has two measure rates of authentication by biometrics:
 - False Reject Rate (FRR) where a legitimate user is rejected by the acquisition device.
 - False Acceptance Rate (FAR) where a false user is accepted.
- "Average Attack Space": corresponds to the number of guesses made by an attacker in order to disclose the base secret (e.g. passwords, PINs, etc.).
- Abbreviations used are the following: C/R=Challenge/Response; PK=Public Key; PRK=Private Key; SSO= Single-Sign-On; TGS=Ticket Granting Service.

Data Generator: 1. PASSWORDS

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Knowledge-based (4 to 8 digits)	A unique personal character such as alphabet, alphanumeric or passphrase (long password) generally containing 4 to 8 digits. It is generally used with a username.	Easily implemented. If system-generated, passwords are more robust (passwords are frequently random and containing special characters).	Passwords can be forgotten. Users create easily identifiable base secrets (passwords, passphrases, PINs), which leaves a gap for guessing attacks and Social Engineering (e.g. searching user desks for passwords on notes, and lying to staff to gain entry into a system).	2 Prone to brute-force dictionary attacks. Plaintext passwords are generally sent "in clear" on the network. (Eavesdropping risk).	\$110 Password Cost Calculator (1)
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, etc.) Computers can generate more secure and automatic passwords than humans (e.g. a registered user in a database can have her password emailed automatically).	1 Passwords violate most of the accepted usability standards for computer systems. Of the eight rules suggested by Ben Shneiderman (1998) for user interface design, password interactions break at least six. SSO ¹⁴⁷ for instance alleviates a bit problem of frequent password reset requests from users who have difficulty remembering several passwords.	5-15 It depends on the user tapping speed.	General acceptance: the most common KBA method. There is a huge variety of products: Password checking on Unix systems (Smith, 2002), Microsoft Windows NT and Windows 2000 (proactive password checking), Macintosh "keychain" Mac OS9, etc.	3	Average Space Attack Rate (i.e. number of guesses the attacker must take to find the base secret) : Dictionary attack= 2^{15} to 2^{23} Reusable Passwords= 2^1 to 2^{23} Random 8-character Unix password= 2^{45} Mouse Pad Search= 2^1 to 2^4 (Smith, 2002)

¹⁴⁷ Single Sign On (SSO) is an access management system which allows users to sign on once and gain entry to multiple applications and network sessions.

Data Generator: 2. PERSONAL IDENTIFICATION NUMBER (PIN)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Knowledge-based (4 to 8 digits)	A unique personal character string used as a password, which must be entered by a user before a remote terminal (mobile device, ATM, etc.) or Point-of-Sale terminal (kiosk) can be used to transfer information or complete a transaction.	More secure than passwords. PIN itself is not sent across the network and so cannot be intercepted. The system has only a manual interface, not a computer interface: nobody can enter the PIN except by using the device keypad.	It can be forgotten. It is susceptible to shoulder surfing or systematic trial-and-error attack.	2 Prone to brute-force dictionary attacks. A PIN is almost always used as part of a two-factor authentication mechanism.	\$110 Password Cost Calculator (1)
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
4 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc. Computers can generate automatic PINs. A registered user with an email address in a database can have her or his PIN emailed automatically)	2	5-8	Very common KBA. General acceptance: Banking (ATMs), Airlines, Mobile Transactions, etc. Banks distribute randomly generated PINs for ATM (Automatic Telling Machines) cards. RSA SecurID authenticators (4).	3	Average Space Attack Rate: A four-digit PIN presents the attacker with only a 13-bit average attack space (Smith, 2002).

Data Generator: 3. PROXIMITY CARDS (CONTACTLESS CARDS)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Authentication Token (AT)	ATs are physical devices that users carry with them. Users use them to authenticate to a network service: they type in something displayed by the token, as well as some secret information they only know.	Last longer because their reading can take place without removing them from their carrying place and does not require any contact (1).	Forgery (too easy to copy), loss, theft possible. Shoulder surfing (stand next to a victim using the card and note the PIN as it is).	3 The combination of a proximity card and a PIN produces a two-factor authentication hence the risk of masquerade, since the attacker must copy the card and also uncover the PIN.	\$30 Card As the communication between the reader and the chips uses electromagnetic waves and there is not physical contact, the contactless readers cost maintenance is much lower.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 Contactless has been proven very convenient for the users.	2-6 To open the door's office, for example, it is enough to flash the card in front of the card reader at the distance up to 5 cm	They appear everywhere, including ATMs, credit cards, drivers' licenses, and on employee badges to operate electronic doors. XyLoc (12)	3	Fair (no available quantitative data) Contactless cards last longer, do not demand contact with card readers so then their reliability is improved.

Data Generator: 4. ONE-TIME-PASSWORDS (OTP)

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Authentication Token (AT) (7)	A hand-held device the size of a credit card is synchronized with the target system's authentication scheme and displays a one-time password that periodically changes (e.g., every minute). To access the target system, the user enters an assigned UserID and password followed by the one-time password displayed on the hand-held.	Very difficult to guess: tokencode (user PIN + code from server). Force users to choose hard-to-guess passwords. An attacker cannot log on by trying to intercept and reuse a password, since passwords only work once. Usually, these tokens are utilized in association with a password or PIN providing stronger security.	If the network is vulnerable to sniffing, this technique is left exposed to dictionary and brute force attacks.	4 Systems which utilize passwords and/or encryption keys to authenticate an individual's identity or protect against interception of communications reach the highest degree of security when each password or key is used only once.	\$80 Token
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1 = min; 5 = max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1 = min; 5 = max)	Accuracy
5 (Acquisition Device or Data generator" presented by the user: PIN, memory card, fingerprint, etc.).	2 An OTP hardware token is an authentication token that users carry with them.	10-20	RSA SecurID® 800 Authenticator Already common but essentially restricted to enterprise environment. There is a huge variety of products.	4	Average Space Attack= 2^{19} to 2^{63} (Smith, 2002)

Data Generator: 5. HANDHELD CHALLENGE RESPONSE CALCULATORS (HCRC)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Authentication Token. Random challenge (7)	When prompted by the system, the user enters a username and Password, then a PIN into the hand-held. Then the system presents the user with a challenge (e.g., a number) which is entered into the hand-held by the user, and a response is calculated. This response is then entered into the target system by the user, and if the response is that expected by the target system, then the user is authenticated and granted access.	Password "can be" a PIN (4 digits) which is often easier to remember. C/R calculators have a keypad, and thus the password is typically considered to be the PIN required to access the calculator. Users enter their username and response into the system, which is verified by the authentication server.	A considerable disadvantage of software tokens is that they make it possible for people to share their access permissions with other people by sharing the software token's base secret.	3	\$80 Token
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	3	Average speed tapping 15 seconds	Already common but essentially restricted to enterprise environment. Secure Computing-SafeWord ¹⁴⁸ , Vasco DigiPass, etc.	4	Average Space Attack Rate: An (ANSI, 1998) token using 56-bit DES presents the attacker with a 54 bits average attack space (Smith, 2002).

¹⁴⁸ Aladdin Knowledge Systems Ltd. <<http://www.aladdin.com/safeword/authenticators.aspx>>

Data Generator: 6. MEMORY CARDS (MAGNETIC-STRIPES)

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Smart Card	A smart card is a credit card-sized plastic card, a sort of intelligent token, which contains an embedded integrated circuit chip. The process is quite simple: take the card, and swipe it through the card reader, then the user is authenticated and granted access.	Attacks in smart cards are very difficult. The security of authentication systems can be greatly enhanced by requiring more than one factor to grant authentication	They cannot administer files dynamically	3 They provide another authentication factor (what you have).	\$30 ³ Card
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1 = min; 5 = max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1 = min; 5 = max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	3	Average swiping speed: 2 seconds The ideal swiping speed deals with your self-confidence. Shy people swipe slower. Anxious people swipe too fast. Confident people swipe at the ideal speed.	General acceptance: There is a huge variety of products. SchlumbergerSema, Datacard Group, Gemalto, etc.	3	Fair

Data Generator: 7. MICROPROCESSOR CHIP CARD AND MULTIFUNCTION CARDS (MCCs)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Smart Card.	They have built-in dynamic data processing capabilities. MCCs organize card memory into separate sections attributed to specific functions or applications. A microprocessor is embedded in the card or microcontroller chip that administers this memory allocation and file access.	Attacks in MCCs are very difficult since they use public key or secret key authentication. MCCs store the key material (secret key) internally on the card itself when using Kerberos authentication. The security of authentication systems can be greatly enhanced by requiring more than one factor to grant authentication. Smart cards provide another factor (what you have).	Smart cards are currently only deployable in an organization's network due to the requirements for specialized hardware connected to the host machine. Since a smart card is a physical device, it needs an interface to the host computer – the smart card reader.	4 It requires a determined and well-funded adversary to carry out an attack on a smart card.	\$80 Card
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	3 Used often in combination with PIN (2). Smart cards work the same as a credit card except in e.g. Web purchasing is easier. If we have a computer coupled with a smart card reader we can slip in our smart card and it will complete the order form thus saving us valuable time.	5-15 A busy person can speed through the drive-through by simply passing a contactless smart card at a reader placed at the order machine.	General acceptance. Smart cards are typically deployed as part of a Public Key Infrastructure. Gemalto, etc.	4	Distance and cycle delay do have an impact on the accuracy of Read/Write. A fingerprint template stored on a smartcard: enhanced algorithm reduced equal error rate to 1.6% in time of approximately 5 seconds.

Data Generator: 8. PUBLIC KEY AUTHENTICATION

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Public Key Cryptography PK and PRK (8)	Public key encryption involves a pair of keys - a public key (PK) and a private key (PRK) - associated with an entity that needs to authenticate its identity electronically or to sign or encrypt data. Each PK is published, and the corresponding PRK is kept secret. Data encrypted with your PK can be decrypted only with your PRK.	<p>PK provides much stronger identity checking.</p> <p>Unnecessary the "key management" as in secret cryptography. Each user simply publishes their PK, and the sender and the recipient no longer have to worry about key distribution.</p>	<p>Key distribution/exchange. Also if asymmetric encryption is used the PRK is a single point of attack, extremely valuable to the enemy.</p> <p>The process to verify a user's identity and obtain a digital certificate can be tedious, error prone, and time consuming, involving multiple steps for administrators and end-users.</p>	<p>4</p> <p>Safer to distribute across multiple enterprises.</p> <p>Higher resistance to trial-and-error attacks.</p>	<p>Deploying readers can cost \$220 per seat around 5000 users.</p> <p>PK technology often assumes smart card readers to be standard on all workstations and PCs. Current PK algorithms require mathematical operations on numbers hundreds of digits long, which, even with today's processing power, can present a large performance penalty when encrypting large amounts of data.</p>

Human vs. Automatism (1=human better; 5=machine= better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data Generator presented by the user: PIN, memory card, fingerprint, etc.)	2 (2) (PGP)	5-15	Widely used to authentication E- commerce hosts on the World Wide Web. Pretty Good Privacy (PGP) (5), Secure Sockets Layer (SSL), and others.	4	Average Space Attack Rate: 512-Bit Public Key= 2^{63} 1024-Bit Public Key= 2^{86} 2048-Bit Public Key= 2^{116} (Smith, 2002). Cost of cryptographic operations: 81.39 milliseconds (13).

Data Generator: 9. KERBEROS

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Key Distribution Center (KDC) (MIT, 2006)	<p>It makes use of strong cryptography so that a client might prove its identity to a server (and vice versa) across an insecure network session. Thus, all authentication processes happen between clients and servers. The KDC carries out the Authentication Service (AS) and the Ticket Granting Service (TGS). The KDC has a copy of each password related to every client or server. Hence, it is crucial to maintain the KDC as secure as possible.</p>	<p>Trusted-third-party: Kerberos works through a centralized authentication server that all systems in the network inherently trust.</p> <p>Mutual authentication: between a user and a server.</p>	<p>Scalability problem: KDC requires substantial centralized administration.</p>	4	Free software

Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
5 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	2 Single Sign-On	5-15	Widely used to authentication e-commerce hosts on the World Wide Web and also in an enterprise environment for access control to network resources (e.g., Kerberos 5.1.3.2 version (MIT, 2006)).	4	Requires clock synchronization between machines on the network (9)

Data Generator: 10. FINGERPRINT RECOGNITION (FR)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Biometrics (the science of identifying, or verifying the identity of, a person based on physiological or behavioral characteristics)	By having an individual scan a fingerprint electronically to decipher information, the issuer of the data can be sure that the intended recipient is the receiver of the data. To authenticate the user, the system compares user's reading with a previously collected thumbprint. Physiological characteristic	A fingerprint is easily sampled using low-tech means, and non-intrusive. Unique to the individual.	Very strong relationship between fingerprint and criminal history. There is elastic distortion from one sample to the next. There is a large variation of the quality of the fingerprint over the population. Hygiene.	4 Forgery almost impossible. Impersonation attack, where an unauthorized individual change her/his biometric to appear like an authorized individual. Replay attack, where a recording of true data is presented to the sensor.	\$100-500 (sensor). The size and price of fingerprint readers are continually declining. The cost of a biometric sensor is quite different from the total cost of ownership. It depends on the magnitude of each application.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (10)	2-9 Enrolment Time < 3 secs Verification Time on Time < 3 secs (i.e., the time period users must spend to have their biometric reference template successfully created).	Non-criminal, civilian applications such as access control, time and attendance tracking, vault, military related and computer user login. Verifier 300 LC 2.0 ¹⁴⁹ .	3 It might intrude on one's privacy (data confidentiality).	Evaluation Method: Technology (Bolle <i>et al.</i> , 2004). False Reject (A legitimate subject is denied service): 3 to 7 in 1,000 (0.3-0.7%) False Accept (A subject is falsely accepted, causing intruders to enter the system): 1 to 10 in 100,000 (0.001-0.01%).

¹⁴⁹ Cross Match Technologies <<http://www.crossmatch.com/verifier-300-lc-2.php>>

Data Generator: 11. HAND GEOMETRY (HG)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
<p>Biometrics (It refers to the geometric structure of the human hand.)</p>	<p>Typical features include length and width of the fingers, aspect ratio of the palm or fingers, width of the palm, etc. HG is measured when a subject presses the biometric against a platen.</p> <p>Physiological characteristic.</p>	<p>Easily collectible and non-intrusive.</p> <p>Computations are also fairly simple; a standalone system is easy to build.</p>	<p>Somewhat relationship between hand geometry and criminal history.</p> <p>Individual hand features themselves are not very distinctive from one person to another.</p> <p>Such contact may be cause for some public hygiene concerns.</p>	<p>4</p> <p>Impersonation attack, where a unauthorized individual change his biometric to appear like an authorized individual.</p> <p>Replay attack, where a recording of true data is presented to the sensor.</p>	<p>\$100 to 500 (sensor)</p> <p>The cost of a biometric sensor is quite different from the total cost of ownership (cost of maintenance of the sensor, costs of running the facility, training, and other related costs). It depends on the magnitude of each application.</p>

Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (HG recognition systems are surprisingly widespread due to their user- friendliness)	4-10	Much of the available information is in the form of patents or application-oriented description. Examples: 3D hand profile identification apparatus used by Sidlauskas ¹⁵⁰ (patent- based), and a prototype system described by Jain et al ¹⁵¹ (application- oriented).	3 Non-intrusive compared to iris and retina.	Evaluation Method: Scenario and Technology (Bolle <i>et al.</i> , 2004). False Reject (A legitimate subject is denied service): 1 in 33(3%) 1 in 20(5%) 1 in 10(10%) 1 in 3(30%) False Accept (A subject is falsely accepted, causing intruders to enter the system): 1 in 7 (15%), 1 in 10(10%), 1 in 20 (5%) - (0%)

¹⁵⁰ D.P. Sidlauskas, 3D Hand Profile Identification Apparatus, US Patent No. 4,736,203 (1998).

¹⁵¹ A. K. Jain, A. Ross, and S. Pankanti, A Prototype Hand Geometry-based Verification System. In 2nd IEEE International Conference on Audio- and Video-based Biometric Person Authentication, pages 166-171, Washington, DC (1999).

Data Generator: 12. FACE RECOGNITION (FR)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Biometrics	<p>An image is examined for overall facial structure. In authentication applications, the system has a camera that searches for a user's face and matches it against the face stored in the user record.</p> <p>Physiological characteristic.</p>	<p>Easily sampled and non-intrusive.</p> <p>Personal knowledge is unnecessary.</p> <p>The least intrusive from a biometric sampling point of view, requiring no contact, nor even the awareness of the subject.</p>	<p>There is some criminal association with face identifiers since it has long been used by law enforcement agencies ("mug-shots").</p> <p>Similarity of appearance, e.g. of identical twins.</p> <p>Influence by glasses, beard, moustache, hair style.</p> <p>Restrictions by illumination, background, and angle of photographing.</p>	<p>4 (10)</p> <p>The fact that the decisions made by biometrics systems can be used as positive proofs/denials of an individual's authorization and /or presence at a sensor raises serious questions about integrity of the overall biometrics systems.</p>	<p>\$100 to 500 (sensor)</p> <p>The cost of a biometric sensor is quite different from the total cost of ownership (cost of maintenance of the sensor, costs of running the facility, training, and other related costs). It depends on the magnitude of each application.</p>

Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (10)	10-15	More acceptable than most biometrics. It is a fairly good biometric identifier for small-scale authentication application. L-1 Identity Solutions ¹⁵²	1 It might intrude on one's privacy (data confidentiality). Users generally find less intrusive. However, some users may be reluctant to have their faces recorded in a database.	Evaluation Method: Technology and Scenario (Bolle <i>et al.</i> , 2004). False Reject (A legitimate subject is denied service): 10 to 20 in 100 (10-20%) False Accept (A subject is falsely accepted, causing intruders to enter the system): 100 to 1000 in 100,000 (0.1-1%).

¹⁵² Face recognition solutions. L-1 Identity Solutions <<http://www.l1id.com/pages/575-face>>

Data Generator: 13. VOICE RECOGNITION (VR)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Biometrics	VR attempts to identify individuals by how they sound when speaking. Voice is a behavioral characteristic but is dependent on underlying physical traits, which govern frequency, nasal tone, cadence, etc.	Voice is a natural biometric, one that people use instinctively. Users do not have to remember passwords. Users do not have to go to a separated process for verification, since anything they say as part of the transaction dialog can be used to verify their identities, resulting in an integrated and non-intrusive verification process. Voice requires no contact.	Changes considerably over time (illness, aging, emotion, etc.) Background noise.	1 VR is particularly vulnerable to replay attacks because of the ubiquity of sound recording and playback devices. Impersonation attack, where an unauthorized individual change her/ his biometric to appear like an authorized individual.	\$100 to 500 (sensor) The cost of a biometric sensor is quite different from the total cost of ownership (maintenance of the sensor, running the facility, training, and other related costs). It depends on the magnitude of each application.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 Novel Neural Net (3) (10)	10-12	More acceptable than most biometrics. It is a fairly good biometric identifier for small-scale authentication application. Apple Mac OSX, Voice Security	1 It might intrude on one's privacy (data confidentiality). Users generally find less intrusive. However, some users may be reluctant to have their faces recorded in a database.	Evaluation Method: Technology and Scenario (Bolle <i>et al.</i> , 2004). False Reject (A legitimate subject is denied service): 10 to 20 in 100 (10-20%) . False Accept (A subject is falsely accepted, causing intruders to enter the system): 100 to 1000 in 100,000 (0.1-1%).

Data Generator: 14. SIGNATURE RECOGNITION (SR)

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Biometrics (SR operates in a 3D environment measuring height and width as well as the amount of pressure applied in a pen stroke).	Off-line or "static" signatures are scanned from paper documents, where they were written in the conventional way. Online or "dynamic" signatures are written with an electronically instrumented device and the dynamic information (i.e., pen tip location through time) is usually available at high resolution. Behavioral characteristic.	Highly accepted by users.	The characteristic of permanence of signature is questionable since individuals can change their signatures at any time. SR is affected by illness, emotion or aging.	3 Impersonation attack, where an unauthorized individual change her/his biometric to appear like an authorized individual. Replay attack, where a recording of true data is presented to the sensor.	\$300 (sensor) (Smith, 2002) The cost of a biometric sensor is quite different from the total cost of ownership (sensor maintenance cost, costs of running the facility, training, and other related costs). It depends on the magnitude of each application.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1 = min; 5 = max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1 = min; 5 = max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (10)	10-15	Communication Intelligence ¹⁵³ and Corporation ¹⁵⁴ . Cyber SIGN ¹⁵⁴ .	2 It might intrude on one's privacy (data confidentiality).	Evaluation Method: Technology and Scenario (Bolle <i>et al.</i> , 2004). False Reject: 3-try: 2.06%; 2-try: 2.10%; 1-try: 9.10% False Accept: 3-try: 0.70%; 2-try: 0.58%; 1-try: 0.43%.

¹⁵³ CIC eSignature Solutions for Enterprise-Wide Paperless Business Processes. Communication Intelligence Corp. <<http://www.cic.com/solutions/anymethod/>>

¹⁵⁴ Witswell Consulting & Services, Inc. <<http://www.cybersign.com/CSlacrobat.html>>

Data Generator: 15. KEYSTROKE RECOGNITION (KR)

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Biometrics	User's typing rhythm. When a user attempts to log onto the system (domain server), the user name and password are compared to the stored keystroke biometric template. Behavioral characteristic.	Neither enrolment nor verification bothers the regular work flow because users would be tapping the keys anyway.	If someone is with you, for example, s/he could possibly observe your key clicks (impersonation attack).	3 KR is used as a means of certain attacks (timing attacks) in order to infer taped text's content and nature in a way to put up for example a password. Not only must the attackers know the correct password, but they must also be able to reproduce the user's rate of typing and intervals between letters.	\$45 (keyboard) Free software is available.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1 = min; 5 = max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1 = min; 5 = max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 Everyone knows how to tap in a keyboard	Between 7 to 30 seconds, depending on the user's tapping speed.	KeyLogPC (6) Limited acceptance.	2 It might intrude on one's privacy (data confidentiality).	Limited use for professional applications due to an insufficient accuracy.

Generator: 16. IRIS RECOGNITION (IR)

Classification	Definition	Advantages	Disadvantages	Security (1 = min; 5 = max)	Cost (CAD\$)
Biometrics	The colored part of the eye bounded by the pupil is the iris, which is extremely rich in texture. Current commercial systems require users to position their eyes within the field of view of a single narrow-angle camera. Physiological characteristic	Unchangeable during lifetime. Unique to individual. There is no elastic distortion from one sample to the next. No contact. It claimed and may widely believed to be the most accurate biometric, especially when it comes to FA rates. Iris has very few FA, an important security aspect.	IR requires much user cooperation and complex and expensive input devices. The performance of iris authentication may be impaired by glasses, sunglasses, and contact lenses.	4 Forgery almost impossible. The fact that the decisions made by biometrics systems can be used as positive proofs/denials of an individual's authorization and/or presence at a sensor raises serious questions about integrity of the overall biometrics systems.	\$300-700 acquisition device (Smith, 2002) The cost of a biometric sensor is quite different from the total cost of ownership (sensor maintenance cost, costs of running the facility, training, and other related costs). It depends on the magnitude of each application.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1 = min; 5 = max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1 = min; 5 = max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (10) Personal knowledge is unnecessary. Collating may take time.	4-12 Positive Identification= 5 seconds Enrollment < 10secs	Very limited acceptance. iCamTD 100 (Iritech) ¹⁵⁵ . IR might be more readily accepted due to the fact that there is no criminal association.	2 It might intrude on one's privacy (data confidentiality).	Evaluation Method: Scenario (Bolle <i>et al.</i> , 2004). False Reject 2 to 10 in 100 (2-10%) False Accept: $\geq 10^{-5}$ (0.001%).

¹⁵⁵ Iris ID <http://www.irisid.com/ps/products/icam_td100.htm>

Generator: 17. RETINA RECOGNITION (RR)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Biometrics	RR seeks to identify a person by comparing images of the blood vessels in the back of the eye, the choroidal vasculature. Physiological characteristic	Unchangeable during lifetime. Unique to individual. Personal knowledge is unnecessary. No contact. Hygienic.	Requires much user cooperation or complex input devices. It is not possible to form images from people whose eyes suffer from strong astigmatism or very poor eyesight. Resistance to irradiation of infrared rays. Environmental lighting conditions. To obtain a retina image, the user is inconvenienced to a greater extent because an image of inside the eye is needed.	4 Forgery almost impossible. The fact that the decisions made by biometrics systems can be used as positive proofs/denials of an individual's authorization and/or presence at a sensor raises serious questions about integrity of the overall biometrics systems.	\$300-700 (Smith, 2002) The cost of a biometric sensor is quite different from the total cost of ownership (sensor maintenance cost, costs of running the facility, training, and other related costs). It depends on the magnitude of each application.
Human vs. Automatism (1=human is better; 5=machine is better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
1 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.)	4 (10) Personal knowledge is unnecessary. Collating time is fast.	4-12 Positive Identification= 45 seconds	Very limited acceptance. iCamTD 100 (Iritech)	1 It might intrude on one's privacy. The fact that a retina image could reveal certain medical conditions makes the RR more controversial (e.g., data confidentiality).	Evaluation Method: Scenario (Bolle <i>et al.</i> , 2004). Retina Recognition 1% FR (False Reject Rate)= 2^6 Retina Recognition n with 0.01% FAR (False Accept Rate)= 2^{12} Retina Recognition with "One in a million"= 2^{19}

Generator: 18. HUMAN UNDER SKIN MICROPROCESSOR CHIP (AUTHENLINK)

Classification	Definition	Advantages	Disadvantages	Security (1= min; 5= max)	Cost (CAD\$)
Human Chip-based (Emanation characteristic)	AuthenLink (Braz and Aïmeur, 2003) is a RFID-based wireless user authentication system which integrates a microprocessor chip implanted under human skin, and a mobile device antenna-embedded. It provides a user automatic access to network resources without the need of any KBA method.	Forgery, stealing, and removing of the chip is extremely difficult. AuthenLink involves a new authentication factor: something you CONVEY. Processing speed. No contact. Don't need password synchronization.	Impersonation attack, where an unauthorized individual gets an implant under her/ his skin (ChipTag) with the legitimate user's data (Chip_User_ID, Chip_Username, Chip_Tag_Number, etc.)	4	\$300 (Chip) The cost of the AuthenLink's ChipTag is quite different from the total cost of ownership (costs of equipments such as authentication server, enrolment, cost of maintenance, database, and microchip implant for humans, and other related costs).
Human vs. Automatism (1=human better; 5=machine better)	Usability (1= min; 5= max)	Total Transaction Time (in seconds) (Woodward <i>et al.</i> , 2003)	Industrial Application	Privacy Issues (1= min; 5= max)	Accuracy
3 (Acquisition device or data generator presented by the user: PIN, memory card, fingerprint, etc.). RF energy passes through the skin energizing the inactive ChipTag, which then emits a RF signal conveying the ChipUser's unique ID to the mobile reader.	3 Ease of operation and no need to remember passwords, PINs or even carry a token.	3-6	No industrial application (human authentication)	4 Probably poor acceptance from users due to privacy concerns, especially due to a chip implanted under skin and ID stored in a database.	No data available.

Notes:

- (1) The Password Cost Estimator shows the direct and recurring costs to your organization regarding the use of passwords. Although passwords seem to be free, they actually cost organizations a significant portion of its IT support budget. This silent budget killer is merely the time the technical support staff devote to resetting users passwords. This does not include the abstract costs associated with lost productivity of the user or security breaches, etc., but the labour cost of the help desk personnel physically resetting passwords on the system. Enabling Compliance with Password Policies, Password Cost Calculator. Mandyllion Research Labs, LLC. May 20, 2010 <<http://www.mandyllionlabs.com/PRCCalc/PRCCalc.htm>>
- (2) Average swiping speed. The ideal swiping speed deals with your self-confidence: shy people swipe slower, anxious people swipe too fast and confident people swipe at the ideal speed.
- (3) Novel Neural Net Recognizes Spoken Words Better than Human Listeners. May 18, 2010 <<http://www.sciencedaily.com/releases/1999/10/991001064257.htm>>
- (4) RSA SecurID authenticators. RSA-The Security Division of EMC May 18, 2010 <<http://www.rsa.com/node.aspx?id=3049>>
- (5) Pretty Good Privacy (PGP) Email encryption program from PGP Corporation May 18, 2010 <http://www.pgp.com/products/desktop_email/index.html>
- (6) Verification is built up on the concept that the rhythm with which the user types is distinguishing. KeyLogPC KeyStroke Logger May 18, 2010 <<http://www.keylogpc.com/>>
- (7) RSA hardware authenticators. RSA-The Security Division of EMC May 18, 2010 <<http://www.rsa.com/node.aspx?id=1158>>
- (8) To reduce risk exposure and comply with security regulations, companies rely on public key infrastructure (PKI) and digital certificates. VeriSign Inc. May 18, 2010 <<http://www.verisign.com/authentication/pki-infrastructure-solutions/index.html>>

- (9) Maximum tolerance for computer clock synchronization is the maximum time that can be tolerated between a ticket's timestamp and the current time at the Kerberos Distribution Center (KDC). Kerberos configuration by Jan De Clercq, October 8, 2004, Elsevier Digital Press <http://searchwindowsserver.techtarget.com/news/article/0,289142,sid68_gci1014049,00.html>
- (10) User data collection can impact on usability as well. User data collection is the time period a person must spend to have her/his biometric reference template successfully created (i.e. enrolment and verification time) but can vary dramatically depending on the biometric device.
- (11) Cards are intended to operate within up to 10cm of the reader antenna at a frequency of 13.56 MHz. ISO/IEC 14443-1:2000. January 23, 2007 <http://www.iso.org/iso/catalogue_detail.htm?csnumber=28728>.
- (12) XyLoc provides full-time access control by determining a user's location and automatically locking the computer when the user is not physically present. Ensure Technologies Inc. May 18, 2010 <<http://www.ensuretech.com/>>
- (13) Cost of cryptographic operations (1,280-bit Rabin-Williams keys on 550 MHz K6). Mazieres, D. & Zhu, Y.: 2002. Machine Learning course - G22.3033-001. Topics in Computer System Security, New York University, NY (USA).

Operation	Time (seconds)
Encrypt	1.11
Decrypt	39.62
Sign	40.56
Verify	0.10
Total	81.39

(14) Form of identifier presented by the user: PIN, memory card, etc.

A.2 Comparison of Biometric Technologies

This section presents a comparison of various biometric technologies from Jain (2004) study. The author argues that it is feasible to understand if a human characteristic can be employed for Biometrics by taking into consideration the following parameters:

- Universality: each person should have the characteristic.
- Uniqueness: is how well the biometric separates individually from another.
- Permanence measures: how well a biometric resists aging?
- Collectability: ease of acquisition for measurement.
- Performance accuracy, speed, and robustness of technology used.
- Acceptability degree of approval of a technology.
- Circumvention: ease of use of a substitute.

The following Table A.2-1 shows a comparison of existing biometric systems in terms of those parameters. Jain (2004) ranks each biometric based on the groups as being low, medium, or high. A low ranking designates weak performance in the evaluation criterion while a high ranking designates a strong performance.

H=High, M=Medium, L=Low.

Biometrics:	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention*
Face	H	L	M	H	L	H	L
Fingerprint	M	H	H	M	H	M	H
Hand Geometry	M	M	M	H	M	M	M
Keystrokes	L	L	L	M	L	M	M
Hand veins	M	M	M	M	M	M	H
Iris	H	H	H	M	H	L	H
Retinal Scan	H	H	M	L	H	L	H
Signature	L	L	L	H	L	H	L
Voice	M	L	L	M	L	H	L
Facial Thermograph	H	H	L	H	M	H	H
Odor	H	H	H	L	L	M	L
DNA	H	H	H	L	H	L	L
Gait	M	L	L	H	L	H	M
Ear canal	M	M	H	M	M	H	M

Table A.2-1: A Comparative Analysis of Existing Biometric Systems (Jain, 2004). *Circumvention is listed with inverted colors because low is desirable here rather than high.

APPENDIX B DATA GATHERING

B.1 Introduction

Data gathering has been developed through a combination of primary, secondary, and tertiary data as follows:

- The primary data have been gathered from observations, and feedback derived mostly from personal communications with experts within the field of HCI and HCISec as below. This thesis had as one of its goals to understand the user authentication process, and the attitudes and behavior of users.
 - Professor Dr. Bonnie E. John
Pittsburgh, PA (USA)
<http://www.cs.cmu.edu/~bej/>
CogTool
<http://cogtool.hcii.cs.cmu.edu/>
 - Professor Dr. Audun Josang
Kjeller, Norway
<http://persons.unik.no/josang/>
 - Professor Dr. Rick Smith
Hastings, MN (USA)
<http://www.cryptosmith.com/about>
Authentication: From Passwords to Public Keys (Smith, 2002)
- The secondary data have been gathered from journal articles, scientific papers, review articles, literature reviews, dissertations, and theses in HCI and HCISec as indicated in the References section.

- The tertiary data have been in turn gathered from standards developed by research institutions and private organizations such as International Organization for Standardization (ISO)/International Electro-technical Commission (IEC) norms and standards: (IEEE 802.11, 1999), (IEEE 1061, 1998), (ISO 13407, 1999), (ISO/IEC 7816, 1998), (ISO/IEC 9126, 2004), (ISO 9241-11, 1998), scientific journal articles, proceedings of meetings, conferences and symposia, technical reports, and graduate dissertations as indicated in the References section. The following international norms and standards have been researched as below:

ISO3407:99: Human Centred Design Processes for Interactive Systems:

This standard explains human-centred design processes for interactive systems, and proposes four categories of human-centred design activities:

- Understand and specify the context of use
- Specify the user and organizational requirements
- Produce design solutions
- Evaluate designs against requirements.

ISO9126: Parts 01, 02, 04: Software Engineering - Product Quality - Part 1: Quality Model ISO/IEC 9126-1:2001 Edition 1; (2001):

Parts 1, 2, and 4 illustrates a two-component model for software product quality: i) Internal quality and external quality, and ii) Quality in use.

“Briefly, internal quality concerns properties of the non-executable portion of a software product during its development, and metrics for internal quality generally concern the quality of intermediate deliverables, such as the source code for a prototype version. In contrast, external quality concerns the behavior of the computer system of which the software product is a part.”

(Seffah and Donyaee, 1998). In i), the model determines six characteristics for internal and external quality, which are further subdivided into sub

characteristics; whereas in ii) the model determines four quality in use characteristics. Quality in use is the combined result for the user of the six software product quality characteristics. The following have been investigated and employed on this dissertation in terms of user's perspective, that is, usability:

- In “Quality Model for external and internal quality”, item 6.3 Usability: Understandability, Learnability, Operability, and Usability Compliance.

Some important questions to be asked are:

- Is the UI intuitive?
- Is it easy to perform easy operations?
- Is it feasible to perform difficult operations?
- Does the authentication method give sensible error messages?
- Is the UI self-explanatory/ self-documenting?
- Is the UI responsive or too slow?

Software engineering - Product Quality – Part 3: Internal metrics ISO/IEC TR 9126-3:2003:

The internal metrics might be applied to a non-executable software product during its development stages (e.g. requirements definition, design specification, etc.). Internal metrics provide us the ability to measure the quality of the intermediate deliverables and thus predict the quality of the final product. This enables us to identify quality issues and make corrective action as early as possible in the development life cycle. This thesis makes use of the *functionality* metrics (security metrics), *usability* metrics (understandability metrics, learnability metrics, operability metrics, operability metrics, attractiveness metrics, and usability compliance metrics), *portability* metrics (adaptability metrics, instability metrics, and portability compliance metrics).

Product quality - Part 4: Quality in use metrics ISO/IEC TR 9126-4:2004:

Part 4 explains the measures that may be employed to specify or evaluate the impact of the use of the software when operated by the user. The ISO/IEC TR 9126-4 can be broken up into three factors such as effectiveness (i.e., usefulness), productivity, and safety. Only the effectiveness (e.g. task completion, error frequency) and productivity (e.g., task time, task efficiency, relative user efficiency) factors have been considered in this thesis work. *“Effectiveness metrics assess the tasks performed by users achieve specified goals with accuracy and completeness in a specified context of use. They do not take account of how the goals were achieved, only the extent to which they were achieve”. Productivity metrics assess the resources that users consume in relation to the effectiveness achieved in a specified context of use. The most common resource is time to complete the task, although other relevant resources could include the user’s effort, materials or the financial cost of usage”.*

ISO9241-11:98: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTS - PART 11: GUIDANCE ON USABILITY)

ISO9241-11:98, PART 11 defines usability in terms of efficiency, effectiveness, user satisfaction, and whether specific goals can be achieved in a specified context of use. *“Usability is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.* Effectiveness is the accuracy and completeness with which users achieve particular goals. Efficiency in turn is related to the resources spent in relation to the accuracy and completeness with which users achieve goals. Finally, satisfaction is the freedom from discomfort, and positive attitudes towards the use of the product.

It is important to highlight that the framework of (ISO9241-11:98) may be used to identify the aspects of usability and the components of the context of use to be taken into account when designing, specifying or evaluating the usability of the authentication method. The performance (*effectiveness and efficiency*) and satisfaction of the users may be used to measure the extent to which a product is usable in a particular context. *“Measures of the performance and satisfaction of the users can provide a basis for the comparison of the relative usability of products with different technical characteristics which are used in the same context”*. As a matter of fact, effectiveness and satisfaction factors are more appropriate to be employed to authentication methods.

To specify and measure the usability of the authentication methods through the task scenario identified beneath the prototype phase, the following sections have been employed within the norm (ISO9241-11:98) as basis for this research work: “5 Specifying and Measuring the Usability of Products”, “Specification and Evaluation of Usability during Design”, and “Specifying and measuring a Work System in Use”.

APPENDIX C

CHECKLIST DEVELOPMENT

C.1 Introduction

Checklists are valuable evaluation instrument when carefully developed, validated, and applied. A sound evaluation checklist clarifies the criteria that at least should be considered when evaluating something in a particular area; aids the evaluator not to forget important criteria; and enhances the assessment's objectivity, credibility, and reproducibility. Moreover, such a checklist is useful in planning an enterprise, monitoring and guiding its operation, and assessing its outcomes. In the evaluation dialect, checklists are useful for both formative and summative evaluations.

Checklists development Checklist (CDC) (Stufflebeam, 2000):

1. Focus the checklist task:

Define the content area of interest.

Define the checklist's intended uses.

Reflect on and draw upon pertinent training and experience.

Study the relevant literature.

Engage and have conversations with experts in the content area.

Clarify and justify the criteria to be met by the checklist (e.g., pertinence, comprehensiveness, clarity, concreteness, ease of use, parsimony, applicability to the full range of intended uses, and fairness).

2. Make a candidate list of checkpoints:

List descriptors for well-established criteria of merit.

Briefly define each of the initial checkpoints.

Add descriptors for checkpoints needed to round out a definition of merit for the content area.

Provide definitions for each of the added descriptors.

3. Classify and sort the checkpoints:

Write each descriptor and definition on a separate 4" x 6" card

Sort the cards in search of categories

Identify the main candidate categories and label each category

4. Define and flesh out the categories

Define each category and its key concepts and terms

Write a rationale for each category

Present relevant warnings about being overzealous in applying the checkpoint

Review the checkpoints in each category for inclusiveness, clarity, and parsimony

Add, subtract, and rewrite checkpoints as appropriate

5. Determine the order of categories

Decide if order is an important consideration regarding the intended uses of the checklist

If so, write a rationale for the preferred order

Provide an ordering of the categories

6. Obtain initial reviews of the checklist

Prepare a review version of the checklist

Engage potential users to review and critique the checklist

Interview the critics to gain an in-depth understanding of their concerns and suggestions

List the issues in need of attention

7. Revise the checklist content

Examine and decide how to address the identified issues

Rewrite the checklist content

8. Delineate and format the checklist to serve the intended uses

Determine with potential users whether category and/or total scores are needed or desired

Determine with users what needs exist regarding differential weighting of categories and/or individual checkpoints

Determine with users any checkpoints or categories of checkpoints that must be passed for a satisfactory score on the overall checklist

Determine with users what needs exist regarding profiling of checklist results

Format the checklist based on the above determinations

9. Evaluate the checklist

Obtain reviews of the checklist from intended users and relevant experts

Engage intended users to field-test the checklist

Generally, assess whether the checklist meets the requirements of pertinence, comprehensiveness, clarity, applicability to the full range of intended uses, concreteness, parsimony, ease of use, and fairness

10. Finalize the checklist

Systematically consider and address the review and field-test findings

Print the finalized checklist

11. Apply and disseminate the checklist

Apply the checklist to its intended use

Make the checklist available via such means as journals, professional papers, web pages, etc.

Invite users to provide feedback to the developer

12. Periodically review and revise the checklist

Use all available feedback to review and improve the checklist at appropriate intervals.

APPENDIX D

USABLE SECURITY PROTOCOL (USP) REUSE METHODOLOGY

D.1 Introduction

The USP reuse methodology provides you the capability to reuse its design artifacts so then you can customize the design of your own products and/or services. As already mentioned, when reuse is implemented at low granularity level, it can assist in shrinking the product development process wastes (Gautam et al., 2007) and fasten the *go-to-market* strategy.

D.2 Design Artifacts for Reuse

Reusing USS is straightforward, you basically replace the USS keys (e.g. (usabilityCriterion_n)) with you own parameters. The USS keys are pretty much descriptive as shown in the system's design artifacts below.

#	Task Scenario	Security Problem/Threat	Usability Criteria	Usability Factors							
(n)	(taskScenario)	(securityProblem)	(usabilityCriteria)	Efficiency	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality
				Usefulness							

Design Artifact 1: Specifying usability factors and usability criteria.

Design Artifact 2: Defining the tasks scenarios.

USE CASE				USABILITY		SECURITY	
#	Title	Scenario	Required Features	Problem	Scenario	Problem	Scenario
(n)	(title)	(uc scenario)	(r feature)	(u problem)	(u scenario)	(s problem)	(s scenario)

Design Artifact 2: Defining the tasks scenarios.

Usable Security Symmetry Inspection Method

1. Usability Criterion: (usabilityCriterion_n) (usabilityCriterionDescription n)											
#	Usability Review	Occurrence			Comments	Security Review			Occurrence		
		Y	N	NA		Y	N	NA	Comments		
1.1	(usabilityReview n)				(usabilityComments n)	(securityReview n)					
1.2	(usabilityReview n)				(usabilityComments n)	(securityReview n)					
...	(usabilityReview n)				(usabilityComments n)	(securityReview n)					

Design Artifact 3: Building USS inspection method matrix.

Design Artifact 4: Developing Method for goal.

Method for goal: Log into the system

Step 1. (stepDescription)

Step 2. (stepDescription)

Step n. (n)

Method for goal: Log into the system	NGOMSL Statement (secs)	Operator (Type)	Operator Time (secs)	Sub-Total Execution Time(secs)
Step 1...	(0.00)	(operatorType)	(0.00)	(0.00)
Step 2...	(0.00)	(operatorType)	(0.00)	(0.00)
Step n...	(0.00)	(operatorType)	(0.00)	(0.00)
Total Execution Time (secs)				(0.00)

Design Artifact 5: Calculating sub and total execution times.

Task_scenario: (Tn) (taskScenario)	
Methods	Sub-Execution Time(s)
Method for goal: (goal_1)	(timeSecs 1)
Method for goal: (goal_2)	(timeSecs 2)
Method for goal: (goal_1)	(timeSecs n)
Total Execution Time	(timeSecs n)

Design Artifact 6: Calculating the sub and total execution times for the task matrix.

Authentication Asset/Target	Threat (T) Description	Vulnerability (V) Description	CIA	Threat Rating	Overall PC/E V Rating	OP= T+V	Overall Exposure/Impact Asset Value Classification	Asset Value Exposure	TI= AVC+ AEC	RA TR= OP x TI	Risk Reduction Strategy
(asset)	(T)	(V)	(CIA)	(TR)	(V)	(OP)	(AVC)	(AVE)	(TI)	(TR)	

Design Artifact 7: Performing the Authentication Risk-Assessment Matrix

(For more information, see Chapter 4: The Cognitive Science Axis).

Matrix Legend:

T=Threat.

V=Vulnerability.

CP/E=Compromise/Exploit.

RA=Risk Assessment

OP=Overall Probability Rating is the sum of the Threat and the Vulnerability Rating (OP=T+V).

TI=Total Impact Rating is the sum of the Asset Value Classification and the Asset Value Exposure (TI=AVC+AVE).

TR=Total Risk Rating is the product of the Overall Probability and the Total Impact (TR=OP x TI).

D.3 Usable Security Symmetry (USS) Inspection Method Quick Setup

A summarized quick setup process to build the USS inspection method is described below. The product key (i.e. `(product_n)`) used within each step represents your own product.

1. Develop a comparative analysis of the `(product_n)` in order to understand what the `(product_n)` is, how it works, and what types of features is contained on it.
2. Undertake a classification analysis of the `(product_n)` from the literature review (e.g. `(product_n)` marketplace) to establish the main `(product_n)` to be used.
3. Identify the most representative `(product_n)` categories, and create the tasks scenarios by using the main `(product_n)` categories identified in item 2.
4. Specify Standard Primitive External operators, Standard Primitive Mental Operators, and Analyst-Defined Mental Operators.
5. Generate a Task Description, a list of High-Level User Goals, Operators and Write Methods for Accomplishing Goals, and Total Execution and Learning Times estimates for each of the `(product_n)` use cases.
6. Identify, in parallel, the main cognitive areas of focus relating to `(product_n)` (e.g. perception, attention and memory, etc.).
7. Define an appropriate cognitive architecture to be used for the defined `(product_n)` in order to understand how and what cognitive processes and their respective flows are involved for each of the `(product_n)`. It serves as the basis for the development of the USS inspection method.

8. Develop an Authentication Risk Assessment matrix to identify the most critical vulnerabilities and threats related to (product_n). This assessment determines which Security Review should be considered within each usability criterion in the USS.
9. Specify the usability factors and usability criteria by classifying and prioritizing the cognitive processes generated by the NGOMSL model. Also gather and analyze primary, secondary, and tertiary data available for building the inspection method.
10. Plot the usability factors and usability criteria in the USS matrix.
11. Assign one (or up to 5) security designers and/or usability professionals to examine the system on an individual basis.
12. Perform the USS inspection method.
13. Gather materials that facilitate the evaluators to become familiar with the purpose of the system and of its users (e.g., system specification, user tasks, use case scenarios, etc). If possible carry out the user actions that will be taken to perform the user tasks. Also identify and list any areas of the system that might be opposed to the usability principles. List all of the concerns that you note in the Comments fields.
14. Define the rating severity of identified *security* problems, its rating severity representation, and recommendations, and also define the rating severity of identified *usability* problems, its rating severity representation, and recommendations.
15. Verify and validate your USS by using a (product_n) business use case. For example you can make use of a *device* (e.g. MTM, BlackBerry, etc.) and an *authentication method* of your choice (e.g. fingerprint, hardware token, etc.).

APPENDIX E

THE GOMS FAMILY: WHICH TECHNIQUE TO USE?

E. 1 NGOMSL versus CPM-GOMS, KLM, and CMN-GOMS

The basic difference between execution time predictions for NGOMSL, KLM and CMN-GOMS is how time is assigned to cognitive and perceptual operators. Actually the differences relate to how large mental operators are supposed. For example, the NGOMSL has more **M** (Mental)-like operators than do the CMN-GOMS and KLM models. A more important difference is in the character of the unobservable operators. There are essentially two types of operators: the ones directly observable when looking at human performance (i.e., motor operators) and those that are not or usually not observable (i.e., perceptual and cognitive operators such as eye-movements). The KLM has a particular basic **M** operator that comes first each cognitive unit of action. NGOMSL, because it is founded on Cognitive Complexity Theory (CCT), constantly demands some cognitive execution time for every step, manipulating goals and working memory, and for entering and leaving methods. On the other side, CMN-GOMS allocates no time to such cognitive overhead. But all three models include **M**-like operators for substantial time-consuming mental actions such as finding information on the screen and checking entries. As shown in Table E-1, each version of GOMS is applicable for specific types of tasks and for getting particular types of information. The KLM and CMN-GOMS are the original models developed by Card *et al.* (1983). KLM characterizes task performance as a sequence of low-level operators and offers quantitative assessments of task performance times. CMN-GOMS requires a strict goal-method-operation-selection rules structure. Natural GOMS Language as already stressed in this section is an extension of CMN-GOMS that illustrates tasks in English-like statements. Finally, the Critical Path Method (or Cognitive, Perceptual,

Motor) GOMS (CPM-GOMS) (Gray *et al.*, 1993) extends GOMS analysis to tasks performed concurrently. The critical path in a schedule chart provides the prediction of total task time (John and Kieras, 1996; John and Kieras, 1996a).

GOMS Model	Description	Design Information Obtained
Card Moran and Newell (CMN) GOMS	Original formulation of GOMS.	Operator sequences, execution times and error recovery for sequential tasks.
Keystroke-Level Model (KLM)	List of keystrokes and mouse movements to perform a task.	Execution times and error recovery for sequential tasks.
Natural GOMS Language	Procedure for identifying all GOMS components expressed in a programming language.	Functionality consistency, operator sequences, execution times, procedure learning times, and error recovery for sequential tasks.
Critical Path Method (or Cognitive, Perceptual, Motor) GOMS	GOMS applied to tasks performed in parallel, uses cognitive, perceptual, and motor operators (Card <i>et al.</i> , 1983), in which the sequential dependencies between the user's perceptual, cognitive, and motor processes are mapped out in a schedule chart, whose critical path predicts the execution time.	Operator sequences, execution times, and error recovery for sequential and parallel tasks.

Table E-1: Summary of different versions of GOMS models. Adapted from John and Kieras (1996).

E.2 GOMS Models Comparative Analysis

The search and selection process to determine the most adequate GOMS model with regards to the cognitive task analysis is the following:

- Perform extensive research on all GOMS models.
- Narrow down the two most relevant GOMS models: CPM-GOMS and NGOMSL.
- Develop a comparative analysis of the CPM-GOMS and NGOMSL models.

- Demonstrate the advantages of using NGOMSL model over CPM-GOMS as shown in Table E-2.
- Adopt the most relevant model for this thesis research.

NGOMSL Model	CPM-GOMS Model
<ul style="list-style-type: none"> • Information about learning time is provided only by NGOMSL model, and predictions cover the time to learn the methods in the GOMS model and any LTM information they require. • NGOMSL is the only method of the four GOMS techniques (KLM, CMN-GOMS, NGOMSL, and CPM-GOMS) that makes learning time predictions. These predictions are limited to the effects of the amount of procedural knowledge and related LTM information to be learned, and to learning situations for which the coefficients have been empirically determined. • The relationship between the NGOMSL notation and the CCT architecture is direct: there is a one-to-one relationship between statements in the NGOMSL language and the production rules for a GOMS model written in the CCT format. Therefore, the CCT prediction results can be used by the NGOMSL model to estimate not only execution time like KLM and CMN-GOMS, but also the time to learn the procedures. 	<p>No learning time predictions. Also relatively unspecified multiple parallel processor architecture in CPM-GOMS.</p>
<ul style="list-style-type: none"> • Information can be gathered about functional consistency by comparing methods and the knowledge necessary to carry out diverse commands. NGOMSL is mainly appropriate to an analysis of consistency, because the structure and content of NGOMSL methods can be inspected, and the learning time predictions of NGOMSL take this form of consistency into account. That is, a consistent interface is one in which the same methods are used throughout for 	<p>No such consistency functionality is present in CPM-GOMS.</p>

<p>the same or similar goals, resulting in fewer methods to be learned (Kieras, 1996). In fact, transfer of training effects can be calculated by deducting the number of NGOMSL statements in methods that are similar, to ones already known to the learner (Bovair <i>et al.</i>, 1990). This categorization of interface consistency in terms of the quantitative transferability of procedural knowledge is possibly the most significant contribution of the CCT research and the NGOMSL technique (John and Kieras, 1996).</p>	
<ul style="list-style-type: none"> • NGOMSL includes a more rigorous set of rules for identifying the GOMS components and information such as the number of steps in a method, how goals are set and terminated, what information needs to be remembered while performing the task. 	<p>No rigorous set of rules</p>
<ul style="list-style-type: none"> • Because NGOMSL models specify methods in program form, they can characterize the procedural complexity of tasks, both in terms of how much must be learned, and how much has to be executed. NGOMSL provides predictions of operator sequence, execution time, and time to learn the methods. • An additional component of the <i>Pure Learning Time</i> is the time required to memorize chunks of declarative information required by the methods, such as the menu names under which commands are found. The total learning time consists of the time to execute the training tasks plus the extra time required to learn how to perform the tasks (i.e. the <i>pure</i> learning time). 	<ul style="list-style-type: none"> • CPM-GOMS predicts only execution time based on an analysis of component activities. It doesn't provide Pure Learning Time. • In addition, it's difficult to use CPM-GOMS models due to the intrinsic difficulty of identifying and describing in detail how perceptual, cognitive, and motor processing activities are coordinated in time. - CPM-GOMS model significantly underpredicts the execution time related to the other models. According to Kieras

	(1994) this is to the conjecture of extreme expertise in the current CPM-GOMS technique: using maximum operator overlapping, finer-grain time estimates for the individual operators, and assuming the minimum of cognitive activity allowed by the MHP.
Elaborated sequential architecture with a working memory and specified procedure knowledge representation.	Powerful cognitive architecture but relatively unspecified multiple parallel processor architecture.
NGOMSL technique currently involves more M (mental)-like operators than CPM-GOMS, as well as some cognitive overhead due to method step execution.	The main distinction with NGOMSL (and other GOMS models as well) is that CPM-GOMS does not include M-like operators; it does not include any substantial cognitive activity associated with selection of methods or complex decisions in the case of the extreme expertise. Such cognitive activity is represented in the other GOMS variants with M-like operators of about a second in duration.
NGOMSL takes time for the unobservable activity related to the production-rule cycling assumed in the basic architecture and represented with the 0.1 sec/statement cognitive overhead.	CPM-GOMS does not take into account unobservable activity.

<p>Since NGOMSL is founded on CCT, it has particular properties that make it unique. CCT not only gives estimations for execution times, but also predicts learning time. NGOMSL represents methods in terms of a cognitive architecture that is CCT. CCT has been shown to provide good predictions of execution time, learning time, and transfer of procedure learning (Bovair <i>et al.</i>, 1990).</p>	<p>It is based directly on MHP (Card <i>et al.</i>, 1983), which is a cognitive modeling method used to calculate how long it takes to perform a certain task. As mentioned CPM-GOMS doesn't provide learning time, and learning transfer.</p>
<p>CCT and NGOMSL model have been empirically validated at the keystroke-level of analysis (operators like DETERMINE-POSITION and CLICK-MOUSE-BUTTON), therefore, models at that level can generate reliable quantitative estimates. In theory, other levels could be researched and empirically validated, but this has not yet been done (Kieras, 1996).</p>	<p>Not empirically validated.</p>
<p>There are also differences in the distribution of mental time. The KLM has a tendency to place mental time in the preparation for action, while CMN-GOMS mental time tends to come at the end of actions in VERIFY operators, and NGOMSL has mental time in both places.</p>	<p>Duration and dependencies of unobservable operators are specified in the templates used to construct the CPM-GOMS model. However, the other operators needed to accomplish a task and their dependencies make every critical path different, and no one estimate of "mental time" is meaningful in CPM-GOMS (John and Kieras, 1996).</p>
<p>The GOMS family consists of task analysis techniques that are related to models of human information processing. Current research involves additional computational cognitive architectures, but only CCT is shown as a "ready-to-use" technique. Only CCT, a production-rule architecture based on the serial stage model, has been used as the basis for a specific GOMS technique, NGOMSL, which incorporates CCT's assumptions about working</p>	<p>CPM-GOMS does not include assumptions about working memory management, flow of control, and other architectural mechanisms. CPM-GOMS is too academic and complex.</p>

memory management, flow of control, and other architectural mechanisms.	
Information can be obtained about functional consistency by comparing methods and the knowledge necessary to perform different commands. NGOMSL is particularly suited to an analysis of consistency, because the structure and content of NGOMSL methods can be inspected, and the learning time predictions of NGOMSL take this form of consistency into account. That is, a consistent interface is one in which the same methods are used throughout for the same or similar goals, resulting in fewer methods to be learned.	No functional consistency.
User authentication tasks are essentially serial tasks so NGOMSL is the most relevant and simpler GOMS model for that particular type of tasks.	CPM-GOMS models are too detailed for tasks that can be usefully approximated by serial operators (Kieras, 1994).
Non-expert behavior for learning time.	CPM-GOMS model makes an assumption of extreme expertise in the user. That is, they usually model performance that has been optimized to proceed as fast as the MHP and information-flow dependencies will allow. The user is extremely experienced and executes the task as rapidly as the MHP architecture permits.

Table E-2: The advantages of using NGOMSL over CPM-GOMS.

E.3. Why GOMS Model When Compared With Other CTAs¹⁵⁶?

The GOMS task analysis method has been offered as a valuable method for comparing different design solutions to the same user interface problem (John and Kieras, 1994; John and Kieras, 1996). The approach is restricted to a comparison of performance times for error-free expert execution of everyday (i.e. skill-level) tasks. On the other hand such analysis of learning and execution times involved in NGOMSL model which is the one used in this thesis, can be of value, particularly when analyzing user interaction within demanding and complex computer security applications including authentication tasks. Despite of the usefulness of the GOMS model for comparison purposes, this thesis explores the applicability of GOMS analysis, more specifically NGOMSL, to understand and identify the cognitive processes involved in user authentication and also investigate its programming-like capabilities for future work. This thesis is focused on NGOMSL as it assures to yield a thorough analysis of interaction which has a psychological basis, which in turn, relates strictly to the Cognitive Axis of this research work.

But why exactly the GOMS task analysis was chosen in relation to other existent cognitive task analysis? As a matter of fact carrying out a GOMS analysis involves defining and then describing in a formal notation the user's Goals, Operators, Methods, and Selection Rules. Most of the work seems to be in defining the Goals and Methods. That is, the Operators are mostly determined by the hardware and lowest-level software of the system, such as whether it has a mouse, for example. Thus the Operators are fairly easy to define. The Selection Rules can be subtle, but usually they are involved only when there are clear multiple methods for the same

¹⁵⁶ Cognitive Task Analysis (CTA) identifies aspects of system design that place heavy demands on the user's cognitive resources including memory, attention, and decision-making. It is used to determine the thought processes that users follow to perform tasks at various levels, from novice to expert.

goal. In a good design, it is clear when each method should be used, so defining the Selection Rules is relatively easy as well.

Identifying and defining the user's goals is often difficult, because you must examine the task that the user is trying to accomplish in some detail, often going beyond just the specific system to the context in which the system is being used. This is especially important in designing a new system, because a good design is one that fits not just the task considered in isolation, but also how the system will be used in the user's job context.

Once a Goal is defined, the corresponding method can be simple to define because it is simply the answer to the question “how do you do it on this system?” The system design itself determines what the methods are (John and Kieras, 1994).

A *task description* describes a generic task in terms of the goal to be accomplished, the situation information required to specify the goal, and the auxiliary information required to accomplish the goal that might be involved in bypassing descriptions of complex processes. Thus, the task description is essentially the “parameter list” for the methods that perform the task.

Example: A sample task description for deleting text with a certain word processor contains the following items:

- the goal is to delete a piece of arbitrary text;
- the starting location of the text;
- the ending location of the text;
- a find string for locating the beginning of the text.

E.3.1 Characterizing the User's Tasks

Although user tasks can be characterized in many different ways, three dimensions are important for deciding whether a GOMS analysis technique is applicable and most suitable to the user's task: the degree of “routinized” skill involved in the user's task, the “sequentiality” of the user's task, and the degree to

which the interaction is under the control of the user versus the computer system or other agents involved in the task (John and Kieras, 1996).

E.3.1.1 Locus of Control

Computer system tasks can be roughly categorized into *passive-system* tasks and *active-system* tasks. In passive-system tasks, the user has control over the pace and timing of task events; the computer simply sits and waits for inputs from the user (e.g. text editing in username and password fields). In active-system tasks, the system can produce spontaneous, asynchronous events outside of the user's control. Thus the user must be prepared to react to the system, which can also include other people who are providing information or making requests.

E.3.1.2 Goal-Directness

Many computer applications today are to support work-related goals: find information, do analyses, produce reports, and so on. In these examples, the user has a task goal and the application should support that goal as efficiently as possible, both in terms of learning how to accomplish the goal and in essentially accomplishing the goal. Nevertheless, some applications are less goal-directed, like an electronic magazine through which a user would browse primarily for entertainment as opposed to trying to find an article for example about a particular security breach.

E.3.1.3 Skill Dimension of Tasks

The skill dimension of tasks goes from one extreme of problem-solving, where the user does not know how to perform a task and must search for a solution, to routine cognitive skill, where the user knows precisely what to do in the task situation and merely has to identify that situation and perform the appropriate actions. The existing GOMS techniques apply only to the routine end of this dimension.

E.3.1.4 Sequential Versus Parallel Activity

Several HCI tasks can be handily approximated as sequential application of operators, such as text-editing. However other tasks entail so much overlapping and parallel activities that this simplification does not usefully approximate the task, as in the telephone operator tasks analyzed by Gray *et al.* (1993). Although this is currently only applicable to CPM-GOMS, in reality a parallel case, it is important to consider when a task involving some parallel operations can be helpfully approximated by a sequential model. Sometimes parallel operations can be represented as a simple modification to the sequential model. For example, it is rationally necessary that users must visually locate an object before they point at it with a mouse. In a sequential analysis, there would be an operator such as VISUALLY-LOCATE-OBJECT followed by a POINT-TO-OBJECT operator. But practiced users can locate a fixed object on the screen (e.g., the password field) while pointing at it with a mouse, meaning that these two operators can execute in parallel.

E.4 Why GOMS?

Since the introduction of *The Psychology of Human-Computer Interaction* by Card *et al.* (1983), the GOMS model has been a leading theory in cognitive modeling and has been called the most mature engineering model of human performance. GOMS is one of the most validated methods in Human Computer Interaction (HCI). GOMS models continue to be applied to the evaluation of software systems, and it remains an active area of scientific research. Kieras *et al.* (1995) lists many successful applications of GOMS to practical design problems.

The GOMS task analysis method has been provided as a valuable method for comparing different design solutions to the same user interface problem (John and Kieras, 1994; John and Kieras, 1996). The approach is restricted to a comparison of performance times for error-free expert execution of everyday (i.e. skill-level) tasks.

Conversely such analysis of learning and execution times involved in NGOMSL model can be of value, particularly when analyzing user interaction within demanding and complex computer security applications including authentication tasks. Despite of the usefulness of the GOMS model for comparison purposes, this thesis explores the applicability of GOMS analysis, more specifically NGOMSL, to understand and identify the cognitive processes involved in user authentication and also investigate and partially implement its programming-like capabilities for future work. In a nutshell, this thesis is interested in NGOMSL as it assures to yield a thorough analysis of the user interaction which has a psychological basis, which in turn, relates strictly to the cognitive axis of this research work.

Some important benefits of using GOMS model are the following:

- Provide a step-by-step description of how users interact with your system, software, etc.
- Predict time needed to complete tasks/goals.
- Do systematic task analysis - use GOMS tasks in studies with real people.
- Use GOMS to speed up iterations of new UI designs.
- Evaluate an existing system through a GOMS analysis.
- Write something for expert users who can't be bothered to help test the software.

GOMS is based on the Model of Human Processor (MHP) (Card *et al.*, 1983) which makes GOMS truly appropriate and relevant for the cognitive computing subject matter. As mentioned, the MHP model is a cognitive modeling method used to calculate how long it takes to perform a certain task. This engineering approach produces a simple model that could generate quantitative predictions for human performance and help designers make low-level design decisions. This model integrates psychological knowledge of human perception and performance with the design process and translated those findings to a form suitable for HCI analysis.

GOMS models can foresee the procedural aspects of usability. These concern the amount, consistency, and effectiveness of the procedures that users must follow. Since the usability of many systems depends closely on the straightforwardness and effectiveness of the procedures, GOMS model has significant value in guiding interface design. The reason why GOMS models can foresee these aspects of usability is that the methods for accomplishing user goals tend to be tightly constrained by the design of the interface, making it doable to build a GOMS model given just the interface design, previous to any prototyping or user testing.

Finally, data on real tasks at this level of complexity which encompasses computer security tasks is expensive to obtain and likely to be problematic for model validation.

E.4.1 During Design - GOMS Analysis Guiding the Design

As stated by Diaper and Stanton (2003), rather than analyze an existing or specified design, the interface could be designed concurrently with describing the GOMS model. That is, by starting with listing the user's top-level goals, then defining the top-level methods for these goals, and then going on to the sub-goals and sub-methods, one is in a position to make decisions about the design of the user interface directly in the context of what the impact is on the user. For example, bad design choices may be immediately revealed as generating inconsistent, complex methods, leading the designer quickly into considering better alternatives. Clearly, this approach is possible only if the designer and analyst are closely cooperating, or is the same person.

Additionally, there is little difference in the approach to GOMS analysis between doing it *during* the design process and doing it *after*. Doing the analysis during the design means that the analyst and designer are making design decisions about what the goals and methods *should be*, and then immediately describing them in the GOMS model. Doing the analysis *after* the system is designed means that the

analyst is trying to determine the design decisions that were made, *sometime in the past*, and then describing them in a GOMS model. For example, instead of determining and describing how the user does a `recall_password` within an existing RBA authentication method, the designer-analyst *decides* and describes how the user *will* do it. It seems clear that the reliability of the analysis would be better if it is done during the design process, but the overall logic is the same in both cases.