Designing Usable, Yet Secure User Authentication Service: The Cognitive Dimension

Christina Braz[a],
Pierre Porrier[a], and
Ahmed Seffah[b]

[a] University of Quebec a Montreal, Canada
{braz.christina, pierre.poirier}@uqam.ca

[b] EHL Lausanne Switzerland, Switzerland
ahmed.seffah@ehl.ch

Abstract

User authentication is key in many interactive tourism software applications and Websites including online booking and reservation applications, customer relationship management systems, etc. However, the design of a user authentication service raises crucial questions when it comes to properly balancing between security and usability. Furthermore, there is a common false design belief that security is only related to the underlying software functionality and can be designed independently from the usability which is related to the User Interface (UI) component and the user experience, in our context the tourist. Finding the right trade-off between these two quality attributes is not an easy endeavour. In this paper, we introduce a new cognitive model that aims to model the tourist task when using a user authentication system. This can help security designers to specify, design, inspect, and evaluate the security as well as usability aspects of user authentication mechanisms. Our model integrates usable security concerns earlier into the requirements and design phase of the development lifecycle. We also show how the proposed model contributes to usable security in a real-world application based on a Multifunction Teller Machine (MTM).

Keywords: Usability, security, user authentication, ATM, tourism interactive applications.

1 Introduction

Like many other domains, security is a critical concern in many types of applications for tourism and in diverse computing devices that can be used by tourist accessing to different tourism-related online services. Due to the fact that such systems are characterized by their user interface components and the user experience that it should support, usability is also another critical concern. User authentication is one example of the basic security services, and tourism e-commerce is a representative application where a trade-off between usability and security is needed in user authentication. However, there is a common but false design 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 & Metzger, 2004). In fact, the meaning of what is a UI and how usability is defined are perhaps a major underlying obstacle that explains such belief. 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 independently from the other quality factors such as security. We define Usable Security 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 management's perspective (e.g., enforcing a strong password policy can be a major constraint to the usability of a system). The fundamental question is therefore how to ensure usability without compromising security and vice-versa. The aim of this paper is to propose a new cognitive model to design usable security system. It aims to help security designers to design, inspect, and evaluate the usability as well as the security aspects of user authentication mechanisms. From our 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. Our model aims also to integrate usable security earlier into the requirements and design phases of the user authentication development lifecycle.

2 Related Work

The Human Computer Interaction Security (HCI-SEC) research community has been constantly reporting the bad usability of security systems and its consequences, vulnerabilities and threats (Whitten & Tygar, 1999; Sasse, Brostoff & Weirich, 2001; Stiegler et al. 2004). Also a significant number of usability problems causing security failures were found in the Pretty Good Privacy (PGP) study (Whitten & Tygar, 1998). We agree with the idea fundamentally supported by the authors that there is a need for a comprehensive model of usable security more specifically for user authentication methods. This model includes either process-and-product related usability characteristics such as effectiveness, efficiency, satisfaction, security, and learnability.

The three essential security properties of confidentiality, integrity, and availability rely on differentiation between authorized and unauthorized users. In order to differentiate them, authentication must be present to grant the network resource. Therefore, authentication plays a fundamental role in our lives. Authentication is the process of establishing whether someone is who s/he declares her/himself to be. This process is based on a risk criterion. High-level risk systems necessitate distinct forms of authentication that more precisely affirm the user's identity as being who s/he claims to be 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". In private and public computer networks (encompassing the Internet), authentication is popularly done through the use of logon passwords. An authentication factor is a piece of information used to authenticate or verify a person's identity. There are three factors of user authentication that might be employed in combination to increase the level of security in the claimed identity of a user: Something you HAS (a smart card), something you KNOW (a password or PIN), and something you ARE (iris recognition). In addition, a fourth authentication factor has been also proposed by Braz and Aïmeur (2003) which is something you convey. The HCI community has been gradually developing research work in usable security guidelines for software such as Computer Security Design Principles (Saltzer & Schroeder, 2000), Guidelines for designing and evaluating usable secure software (Yee, 2005), "Principle of Least Authority" to individual programs: Polaris provides protection against viruses while simultaneously improving usability and functionality (Stiegler et al., 2004), and others. But to date, there is no theoretical framework to provide an evaluation method that considers security and usability synergistically for user authentication.

3 The Cognitive Model of User Identification

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 essentially involve the interaction that occurs between the user and authentication method (e.g. log into a system through an OTP token). Most usability inspection techniques do not overtly take into account users' thinking, even though psychology-based inspection techniques provided key insights into how thinking shapes interaction. Another evidence is that the well known KBA does not take into account how people think. Also empirical research (Zurko & Simon, 1996) has shown that cognitive dimensions have definitely influence in the usability of security mechanisms under which user authentication methods are included. Authors argue that security concepts used in security mechanisms are not easily graspable by intuition to many users. Hence security designers should place an additional effort into understanding the cognitive demands placed on users, and employing concepts they can recognize and cope with. 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 motor (e.g. Alice types in a password on a desktop keyboard), and cognitive ones (e.g. Alice tries to remember a strong password such as <Gyz!152#> which results in a huge demand on her memory). According to the company's security policy, a strong password must be enforced given that it makes the attacker job much harder in guessing predictable passwords. Enforcing a strong password is not an easy task given that the cognitive capacity of a user to remember a password is quite limited (Saltzer & Schroeder, 2000). Alice keeps trying to log into the system but after three unsuccessful attempts Alice is locked. At this moment the system has blocked her account. Another example is the PGP which is poorly understood and hard to use by users according to usability evaluations (Whitten & Tygar, 1999). These facets of understanding how users cope (or not) with different types of user authentication methods explain our interest in studying its cognitive dimensions. Our aim is to provide a Cognitive Ergonomics (CE) account of user authentication design using GOMS (Goals, Operators, Methods, and Selection Rules) or more specifically the Natural GOMS Language (NGOMSL).

3.1 Cognitive Ergonomics

HCI involves systems comprising of people, computers and their interactions. CE though is concerned with the analysis of cognitive processes such as perception, memory, reasoning, and motor response required from operators in modern industries. CE also studies the competencies and limitations of workers in their interaction with the work system (e.g. errors, strategies, cognitive workload), in particular with the cognitive artefacts 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), high-tech, or automated systems. Conventionally CE has employed the human information-processing model of cognition (Wickens, 1992), which models human cognition through a computer metaphor. Both HCI and CE aim to support the optimisation of human computer interactions for effectiveness. 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.). Therefore this "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 increasingly placing the burden of authentication on the user forcing them to perform - at the enterprise's deliberation - 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 to generate a response, in this particular study, when authenticating to a system (the assessment item). The cognitive processes are typically comprised of recall and recognition (e.g. facial recognition authentication), and identification and classification (e.g. KBA such as SiteKey1: first you recognize a unique image you chose and image title you created to accompany your image. Then you group image and title carrying out in this way collection and comparison.

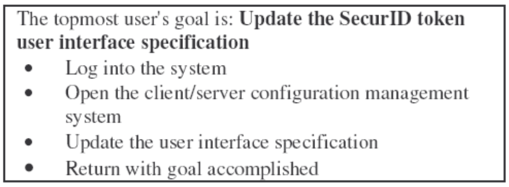
3.2 Cognitive Task Analysis

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. It elicits information from individuals about the cognitive processes they use in the course of completing specific tasks such as authenticating to a computer system using a KBA method (e.g. security questions as an emergency access method). CTA is conducted for a large collection of purposes such as system development, instruction and training, human-computer interface design, and others. The outcome in our research is a description of the conceptual and procedural knowledge used by users as they perform a task involving authentication. After extensive research among a variety of current CTA strategies, we concluded that in context of the problem under consideration, user authentication, NGOMSL (Natural Goals, Methods, Selection Language) (Kieras, 2006) was the most appropriate CTA method.

Natural GOMS Language (NGOMSL). The GOMS model is a type of engineering model for interface design. It is a description of the knowledge that a user must have

[1] http://www.bankofamerica.com/privacy/sitekey

in order to carry out tasks with a system; it is represented in a way that can truly be executed. It means that a user (or a programmed computer) can undergo the GOMS description, running the described actions, and really carry out the task. The acronym GOMS stands for Goals, Operators, Methods, and Selection Rules. Briefly, a GOMS model consists of descriptions of the Methods required to achieve particular Goals (Figure 1). The Methods are a sequence of steps consisting of Operators that the user performs (Table 1). A Method might call for sub-Goals to be accomplished, thus 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. The GOMS modeling techniques has proven extremely successful in developing accurate cognitive task models (Williams & Voigt, 2004).

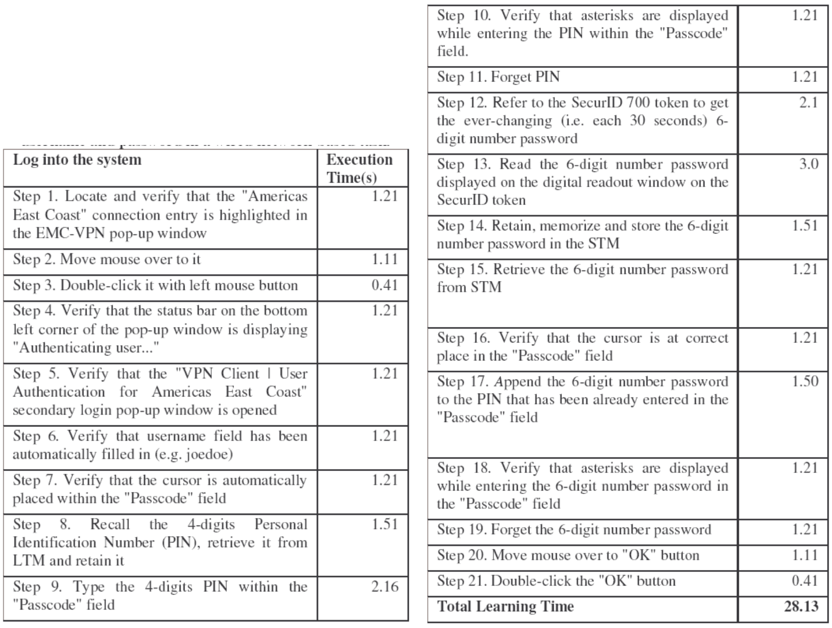


The topmost user's goal is: **Update the SecurID token user interface specification**

- Log into the system
- Open the client/server configuration management system
- Update the user interface specification
- Return with goal accomplished

Fig. 1. High-level user goals.

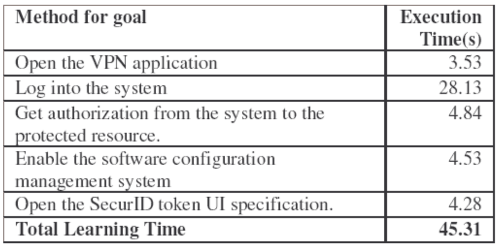
Table 1. Method for goal: "Log into the system" using username and password in a wired network-based task.



Log into the system	**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 Learning Time	**28.13**

A more rigorously defined GOMS version called NGOMSL presents a process for identifying all the GOMS components, expressed in a form analogous to an ordinary computer programming language. NGOMSL comprises rules-of-thumb about how many steps can be in a method, how goals are set and terminated, and what information needs to be remembered by the user while doing the task. NGOMSL 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 (Kieras & Polson, 1985). It allows NGOMSL to incorporate internal operators (i.e. actions that the user executes) such as manipulating working memory information or setting up sub-goals. 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 (Table 2). They are based on expert users and well defined tasks. An emphasis is given to the analysis of the cognitive processes involved in user authentication.

Table 2. Total Learning Time for Task_scenario: Update the SecurID token user interface specification.



Method for goal	Execution Time(s)
Open the VPN application	3.53
Log into the system	28.13
Get authorization from the system to the protected resource.	4.84
Enable the software configuration management system	4.53
Open the SecurID token UI specification.	4.28
Total Learning Time	**45.31**

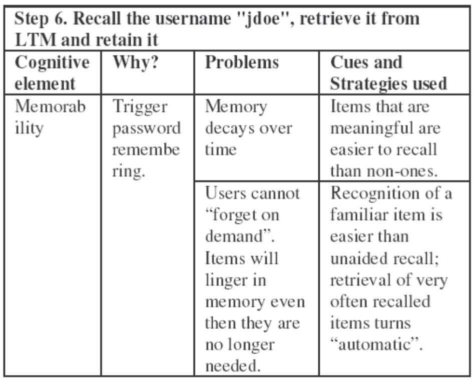
As mentioned, carrying out a GOMS analysis involves defining and then describing in a formal notation the user's Goals, Operators, Methods, and Selection Rules. Consider this NGOMSL excerpt from our complete NGOMSL model2. The topmost user's goal is: Update the SecurID token user interface specification. The first required step to develop the NGOMSL model is to describe the set of user's high level goals (Figure 1). The Total Learning Time is the total time needed to complete a training process while the execution time is the time for the execution of the methods required to perform the task itself by the user. The Total Learning Time for Task_scenario: T1 - Check Business E-mail is shown in Table 2.

Cognitive Demands. Cognitive elements have been more and more incorporated into systems design due to the changing environment of the workplace and the effect of technology on countless tasks and functions. Tasks nowadays put to a great extent increased demands on the cognitive skills of workers. Howell and Cooke (1989) have claimed that with progresses in technology, we have increased, rather than lowered, cognitive demands on humans. More procedural tasks are conducted by intelligent

[2] http://www.labunix.uqam.ca/~ha991381/NGOMSL_ model.pdf

machines, while humans have become in charge of tasks that involve inference, diagnoses, judgment, and decision making. For example, in manufacturing technologies, critical skills at present comprise perceptual skills for monitoring equipment, diagnostic skills for interpreting computerized information, and communication skills required for problem solving and co-ordination in distributed decision environments. Therefore, once carried out the NGOMSL analysis we have also developed a cognitive demands mapping to security (an excerpt is shown in Table 3) to sort through and analyze the data. The cognitive demands matrix is intended to provide a format for the security designer working in conjunction with a user experience designer when developing an authentication mechanism. The focus at this point is to analyze the different cognitive processes involved and their respective issues and strategies used.

Table 3. Cognitive demands excerpt: memorability problems and strategies from Sasse et al. (2001).



Step 6. Recall the username "jdoe", retrieve it from LTM and retain it			
Cognitive element	**Why?**	**Problems**	**Cues and Strategies used**
Memorability	Trigger password remembering.	Memory decays over time	Items that are meaningful are easier to recall than non-ones.
		Users cannot "forget on demand". Items will linger in memory even then they are no longer needed.	Recognition of a familiar item is easier than unaided recall; retrieval of very often recalled items turns "automatic".

4 Conclusion

As discussed in this paper, several critical risks are originated by weak usable security. It is therefore required to consider security usability as part of vulnerability analysis and risk assessment in order to properly manage current and emerging risks. We proposed a new taxonomy of security-sebsitive task for dealing with this issue. We have also demonstrated how this task model can be used as a design model in order to understand how cognitive processes influence the user authentication service. We also demonstrated via a Multifunction Teller Machine (MTM) a real world application how to apply this model during requirements and design phase in the development life cycle. Based on cognitive model, we develop specific design guidelines and design pattern that take into account the specific constraints of usability mechanisms and their potential consequences on security.

References

Braz, C. & Aïmeur, E. (2003). AuthenLink: A User-Centred Authentication System for a Secure Mobile Commerce. Dept. of Computer Science, University of Montreal, Canada.

Howell, W. C. and Cooke, N. J. (1989). Training the human information processor: A look at cognitive models, in I.L. Goldstein (ed.), Training and Development in Work Organizations: Frontiers of Industrial and Organizational Psychology, San Francisco: Jossey-Bass, 121 -182.

Kieras, D. E. (2006). A Guide to GOMS Model Usability Evaluation using NGOMSL and GLEAN4. Artificial Intelligence Laboratory Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI (USA).

Kieras D.E. & Polson P.G. (1985). An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies, 22:365-394.

Saltzer, J. & Schroeder, M.D. (2000). The Protection of Information in Computer Systems. University of Virginia, Department of Computer Science CS551: Security and Privacy on the Internet.

Sasse, M.A., Brostoff, S. & Weirich, D. (2001). Transforming the "weakest link" — a human/computer interaction approach to usable and effective security. BT Technol Journal, 19 (3).

Seffah A. & 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): 71-76.

Stiegler, M., Karp, A.H., Yee, K.P. & M. Miller. (1992). Polaris: Virus Safe Computing for Windows XP. Mobile and Media Systems Laboratory, HP Laboratories Palo Alto HPL-221.

Wickens, C.D. (1992). Engineering Psychology and Human Performance, 167-210. Harper Collins, 2nd edition.

Williams, K.E. & Voigt, J. R. (2004). Evaluation of a computerized aid for creating human behavioral representations of human-computer Interaction. (Cognitive Processes), Human Factors.

Whitten, A. & Tygar, J.D. (1998). Usability of Security: A Case Study. Carnegie Mellon University School of Computer Science Technical Report CMU-CS-98-155.

Whitten, A. & Tygar, J.D. (1999). Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In the Proceedings of *8th USENIX Security Symposium.,* Usenix Assoc., 169–184.

Yee, K.P. (2005). Guidelines and Strategies for Secure Interaction Design. In *Security and Usability: Designing Secure Systems that People Can Use*, edited by Lorrie Faith Cranor, Simson Garfinkel. O'Reilly, CA (USA).