*Proceedings of the OZCHI 2004 Conference* (Wollongong, Australia, November 21-24, 2004).
University of Wollongong, Wollongong, Australia. Preprint version.

# Usable Security and E-Banking: Ease of Use vis-à-vis Security

Morten Hertzum, Niels Christian Juul, Niels Jørgensen, Mie Nørgaard.
Roskilde University, Denmark
Email: {mhz,ncjuul,nielsj,mnl}@ruc.dk

## Abstract

*Electronic banking must be secure and easy to use. An evaluation of six Danish web-based electronic banking systems indicates that the systems have serious weaknesses with respect to ease of use. Analysis of the weaknesses suggests that security requirements are among their causes and that the weaknesses may in turn cause decreased security. Conceptually we view the conflict between ease of use and security in the context of usable security, intended to match security principles and demands against user knowledge and motivation. Automation, instruction, and understanding can be identified as different approaches to usable security. Instruction is the main approach of the systems evaluated; automation relieves the user from involvement in security, as far as possible; and understanding goes beyond step-by-step instructions, to enable users to act competently and safely in situations that transcend preconceived instructions. We discuss the pros and cons of automation and understanding as alternative approaches to the design of web-based e-banking systems.*

**Keywords**

Usable security, ease of use, security, electronic banking, public key infrastructure, strong passwords.

## INTRODUCTION

E-banking, which we define as web-based electronic banking, is a high-risk area with a potential for substantial economic loss. The high risk makes *security* a prime concern. In 2001, an estimated total of $17 billion was spent on information-security products and services in the United States alone (IDC, 2003). The diversity of e-banking users and the absence of any special training prior to becoming one make *ease of use* a prime concern as well.

The current web-based variant of electronic banking is the latest of several generations of systems: *Automated teller machines* (ATMs) were the first well-known machines to provide electronic access to customers of retail banks. Next came *phone banking* where users call their bank's computer system on their ordinary phone and use the phone keypad to perform banking transactions. *PC banking* superseded phone banking and allowed users to interact with their bank by means of a computer with a dial-up modem connection to the phone network. Phone and PC banking entailed maintenance costs associated with keeping up to date with diverse modems and with avoiding prohibitively complex installation procedures. E-banking uses the web browser for the user interface and the Internet for data transfer and download of software, and so has a potential for reducing maintenance costs. For users, e-banking provides current information, 24-hours-a-day access to banking services – in addition to the familiar browser interface. The primary services provided by e-banks are transferring money among one's own accounts, paying bills, and checking account balances. Loans, brokering, share trading, service bundling, and a host of other financial services are being added to these primary services (e.g., Dewan & Seidmann, 2001). Since the late 1990s e-banking has developed from virtual insignificance to tens of millions of users worldwide, and due to the high penetration, e-banking systems must accommodate a wide range of users. E-banking is widely used in, among other places, the Nordic countries. In 2001, e-banking was used by more than 25% of the population in Norway, Sweden, and Finland, and by 15% of the population in Denmark (OECD, 2001).

Usable security of e-banking may be of some general interest, for example in view of the current endeavours towards further digitalization of public administration. E-government services that are currently on the planning table in many countries, such as services that enable citizens to access their medical or tax data via the Internet, entail security and ease of use requirements similar to those for e-banking.

The purpose of this paper is twofold. At the concrete level, we assess the security and ease of use of contemporary Danish e-banking systems. At the conceptual level, the study is about *usable security* – that is, matching security principles and demands against user knowledge and motivation. To address these issues, emphasis is on use-related aspects of security, such as risk awareness, installation procedures, and whether systems are oriented toward instruction or understanding rather than on purely technical issues, such as encryption algorithms.

In the remainder of the paper, we relate usable security to usability in general, then present our evaluation method and evaluation results, and finally we discuss and conclude.

## USABLE SECURITY

Passwords provide an illustrative example of the conflict between security and ease of use and, thereby, of the need for a concept of usable security (e.g., Adams et al., 1997; Klein, 1990; Morris & Thompson, 1979; Pinkas & Sander, 2002; Schultz et al., 2001). While passwords such as person names or other real words are relatively easy for a user to remember they are weak from a security point of view because they are vulnerable to dictionary attacks. Strong passwords (e.g., x7h!t%C9) are less vulnerable to attack but at the same time more difficult to remember.

The term ease of use is used in this paper in the sense of learnability and understandability of user interfaces, and both ease of use and security are considered to be contained within the concept of usability. Within human-computer interaction, the ISO 9241-11 (1998) definition of usability has gained widespread acceptance. The standard defines usability 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." This broad definition equates usability with the quality of a system in use and, thereby, addresses the entire issue of whether the system meets the needs of actual users. Specifically, effectiveness includes whether a system enables users to achieve their goals at an appropriate level of security, and likewise, ease of use is also contained within the ISO 9241-11 definition of usability. In contrast, several other studies allude to a conflict between usability and security (e.g., Adams et al., 1997; Dourish & Redmiles, 2002; Schultz et al., 2001; Smetters & Grinter, 2002); these studies assume a narrower definition of usability, as conventional in software engineering (see ISO/IEC 9126-1, 2001).

Although security has become an aspect of virtually any system, and users habitually interact with access-control mechanisms for, among other things, personal computers, email accounts, and e-commerce or e-government sites, the usable security aspect of such systems appears to be not fully understood. Contributions include the set of characteristics proposed by Whitten and Tygar (1999) of the usability problem for security, including the *weakest link property*, the *unmotivated user* property, and the *barn door property*. (The latter is the property that once a secret such as a private key is compromised, then closing the barn door, e.g. setting up a firewall, does not restore security.) Several studies including Schultz et al. (2001) suggest that security measures that are inconvenient for users may weaken security, for example because of lack of user acceptance or outright resistance. Dourish and Redmiles (2002) propose a distinction between *theoretical* and *effective* security. Theoretical security concerns the level of security that is technically possible; for example, digital signatures provide strong authentication under the assumption that various difficult computational problems related to prime numbers will not be solved within some time frame. Effective security concerns the level of security achieved in practice, and is typically lower than theoretical security, due to weaknesses with respect to, among other things, algorithm implementations, protocol design – and ease of use.

Whitten and Tygar (1999) suggest the following definition, against which we measure the e-banking systems:

Security-related software is usable if the people who are expected to use it -

1. are reliably made aware of the security tasks they need to perform;
2. are able to figure out how to successfully perform those tasks;
3. don't make dangerous errors; and
4. are sufficiently comfortable with the interface to continue using it.

This definition addresses the effectiveness (issues 1 and 3), efficiency (issues 1 through 3), and satisfaction (issue 4) of security-related software, and so accords with ISO 9241-11 (1998).

There are two ways in which it seems appropriate to re-interpret Whitten and Tygar's notion of a security task for use with our evaluation. First, many security tasks in e-banking are tightly *integrated* into proper e-banking tasks, eg. typing a password when a transaction is confirmed, rather than independent tasks such as many of those considered in Whitten and Tygar's evaluation of PGP, a tool for secure email. Second, we stress that some security tasks in e-banking may be difficult for the system designers to *predict* in advance, which seems to contradict an underlying assumption of issue 1. Thus, security tasks may be part of other tasks, and may not be predicable in advance.

## WHAT WE EVALUATED

A single e-banking service, transfer of money, was selected to represent ordinary retail banking, that is, non-commercial account holders' ordinary interactions with their e-bank. We believe this service is the most important for most ordinary users, and certainly every e-bank should provide it in a secure and easy to use way.

A sample of six Danish e-banks was selected. The sample includes the three e-banks offered by the three largest Danish banks. Each of the three other e-banks is based on software from one of three independent Danish

manufacturers of generic software for e-banking. The three biggest banks develop their e-banking software in-house while all other Danish e-banks are based on software from one of the three manufacturers. Thus, apart from differences due to customization and extension of a generic solution, the sample covers the entire Danish e-banking market.

The task of conducting a money transfer in the six e-banks requires that two or three other tasks are carried out, namely installation (five of the e-banks) and logon and logoff (all six e-banks). We include these prerequisite tasks in the evaluation.

The prerequisite tasks serve to attain various security requirements. All six e-banks have pledged to comply with the security requirements laid down in the industry code defined by The Danish Bankers Association. The code contains the following requirements, rephrased here to comply with standard security terminology (see ITU 1991): *User authentication* to verify the identity of the user prior to completing transactions and prior to getting access to account information. *Nonrepudiation* to enable banks to prove that a given transaction has been conducted by a given user. *Data confidentiality* to protect transaction or account data communicated between the user and the bank from disclosure. There are various more specific requirements, most notably that user authentication is based on two independent secrets:

1. a secret the user *knows* (e.g., a memorised password)
2. a secret the user *possesses* (e.g., a private key stored on the user's computer).

The installation task is a prerequisite in the five banks that implement the industry code's two-secret requirement by means of a *public key infrastructure (PKI) solution*. In contrast, no installation task is required to conduct money transfer or other services in the e-bank that makes use of a *physical code card*. Table 1 summarizes how the two solutions implement the industry code's two-secret requirement.

The logon and logoff tasks establish and terminate a secure connection between the user's computer and the e-bank's server by means of the SSL/TLS (secure socket) protocol, to attain data confidentiality. All six e-banks require the possessed and the known secret at logon.

The full evaluation comprises the following four tasks having various specified security subtasks:

*Installation,* with two security subtasks: password definition, and signature file management.

*Logon,* with one security subtask: user authentication.

*Money transfer*, with one security subtask: user authentication.

*Logoff*, which is in itself a security task.

The goal of the evaluation of these tasks is to answer the following questions, which are based on the previously discussed definition by Whitten and Tygar of usable security:

*Is the user made reliably aware of, and able to successfully perform the security subtasks ?*

*Is the user prevented from making dangerous errors ?*

*What is the cost of the security subtasks in terms of the added complexity of the user interface - is there a risk that the user does not feel sufficiently comfortable to continue using it ?*

| | **Public key infrastructure (PKI)** | **Code card** |
|---|---|---|
| Possessed secret | The private key. The key is stored in a signature file, typically on the user's computer, and is stored in encrypted form, to protect it from disclosure in case of theft of the signature file. | The physical code card. The card is the size of a credit card and contains 80 pairs of keys and codes. It has no electronic or magnetic components; all the information it contains is conveyed in print. When prompted the user must type the code that corresponds to a given key. |
| Known secret | A password used to decrypt the private key. | A password which is unrelated to the code card. |
| Installation required ? | Yes. Generation by software on the user's computer of a pair of private and public keys, and storage of them in the signature file. | No. Each key-code pair is used only once, in a random order chosen by the e-bank system. Eventually a new code card is sent to the user. When the user receives a code card it is ready for use. |

Table 1. User authentication by PKI vs. code card, and how they implement the two-secret requirement of the Danish Bankers Association's industry code.

## HOW WE EVALUATED

The e-banking systems were evaluated by a walkthrough of the four tasks (i.e., by analytic inspection). Accounts were opened in the six banks so that we could conduct the four tasks as ordinary customers. The walkthroughs consisted of constructing sequence models of the actions involved in performing the four tasks. The sequence models, inspired by Beyer and Holtzblatt (1998), document the information provided by the systems and delineate the correct sequence of user actions required to perform the tasks. As part of the evaluation of the installation task, we attempted to define weak passwords, deliberately disregarding any advice given by the system.

To attain an approximate, quantitative measure of the complexity of the user interfaces, we counted the number of steps, codes, and concepts involved in each task, as recorded in the sequence models. A *step* is an action users must perform to provide input to the system, either by filling in a field or clicking a link. A *code* is a character string users must key in to identify themselves, including account numbers and passwords. *Concepts* are security concepts presented to users as part of on-screen explanatory text.

The evaluation was conducted on a PC with Microsoft's Windows 2000 operating system and Internet Explorer 6.0 browser. The security settings of the browser were set to "custom" and the privacy level to "medium".

## RESULTS

In all six e-banks, the user authentication subtasks of logon and money transfer are tightly integrated. The user merely has to type the relevant passwords etc. when prompted. Serious errors are probably prevented during these tasks, because if the user fails to provide the input, the tasks can not be completed at all. This integration facilitates daily use of the e-banks. On the other hand, the e-banks suffer from three main weaknesses:

1. The installation task associated with the five PKI-based e-banks is highly demanding, and more complex than any other task. In contrast, the code-card solution relieves the user of installation, but on the other hand, each logon is more cumbersome in the code card-based e-bank (see Table 2).
2. Security alerts and other messages from the browser are likely to confuse the user during installation (if any) and also during logon and logoff.
3. The user may not be made sufficiently aware of the security subtasks of installation; the user is not likely to be able to figure out how to perform them; and the error of defining a weak password is not consistently prevented.

In the remainder of this section we detail observations 1-3. The full evaluation results, including the sequence models, can be seen in Hertzum et al. (2004).

| | **Ebank1 (code card)** | **Ebank2 (PKI)** | **Ebank3 (PKI)** | **Ebank4 (PKI)** | **Ebank5 (PKI)** | **Ebank6 (PKI)** |
|---|---|---|---|---|---|---|
| Installation | No installation | 13 | 14 | 19 | 15 | 23 |
| Logon | 9 | 5 | 5 | 6 | 6 | 7 |
| Transfer | 3 + data | 5 + data | 5 + data | 6 + data | 6 + data | 6 + data |
| Logoff | 1 | 2 | 2 | 1 | 2 | 2 |

Table 2. Approximation of the complexity of the four steps in the evaluation by the number of steps they require. The transfer task additionally requires typing in data to specify the transfer, such as the amount to be transferred.

### The complexity of the installation task in the PKI-based e-banks

The complexity of installation in the six e-banking systems is described in more detail in Table 3 based on quantitative data about the number of steps, codes, and concepts involved in the installation process (see the previous section for a definition of these terms).

A significant part of the installation process is the typing of various passwords and other codes, for identification and authentication. The user must type: temporary identification and password, for initial identification and authentication, and permanent identification and password for subsequent use; all five PKI-based e-banks require the users to define their own permanent passwords. Some of the banks merge the two identification codes, reducing the number of codes from four to three, as can be seen in the table. In addition to the codes users are also asked to provide other security-related input, such as where to store the signature file.

The security concepts presented to the user during installation include concepts such as authenticity, verification, signing, certificate, secure connection, private key and signature file. These are foreign to most users, intricately related, and not easily understood. Users are presented with 8-14 such concepts during installation of their e-banking system (Table 3). This includes concepts appearing in browser messages (see below). Of course, the number of steps increases beyond those in the table if the user chooses to inspect some of the links to explanatory information or has to backtrack to make changes or correct mistakes.

The complexity of the installation tasks is aggravated by scant information about how far the user has progressed toward completion of the installation.

The code card-based e-bank avoids installation altogether. Assessment of the merits of the code card-technology against a public key infrastructure must consider the trade-off between avoiding installation and introducing some inconvenience in daily operation. Installation may become a barrier that precludes further use of the e-banks, but is performed only once. The three other tasks will be executed repeatedly during everyday use of the e-banks. The code card is an external device and everyday use is somewhat less convenient because the card must be available. Also, logon to the code card-based bank involves more steps (nine) than in the other e-banks (see Table 3). The user is presented with 11 security concepts during logon. For logon and money transfer the user must type in four codes, including code card number and a key, both of which must be read from the card.

| | **Ebank1** | **Ebank2** | **Ebank3** | **Ebank4** | **Ebank5** | **Ebank6** |
|---|---|---|---|---|---|---|
| Steps | no installation | 13 | 14 | 19 | 15 | 23 |
| Codes | no installation | 3 | 4 | 4 | 3 | 4 |
| Concepts | no installation | 8 | 8 | 14 | 12 | 8 |

Table 3. The complexity of the installation process. The table gives the number of steps users must go through, the number of different security-related codes users must key in, and the number of security concepts users are presented with during installation (includes concepts that appear in browser messages).

**The browser messages**

In all six e-banking systems, information from the banks is interspersed with security messages from the browser during installation (if any) and during logon and logoff. None of the e-banks mention the browser messages in their introductory material. Users are unprepared for messages such as the security warning of Figure 1, and may experience these messages as an indication that there is something wrong. In fact they are

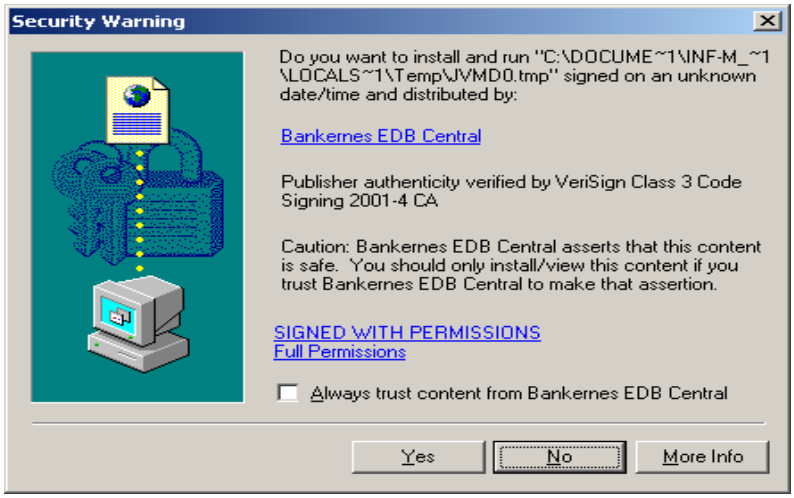


Figure 1. A browser message that pops up during installation of the PKI-based e-banks.

part of normal operation, where sound browser design principles prescribe that the user is informed and asked about download of foreign code. The reason that the e-banks do not attempt to explain the browser messages may be that users have different browsers, versions, and configurations, so that the banks are unable to predict what messages are presented to the user by the browser.

Most users, even if they recognize that the browser messages are normal, will be unable to understand and assess the security implications of the messages. The security warning in Figure 1 asks for the user's permission to install and run a piece of software on the user's computer. There are, however, multiple problems, including:(1) Users cannot be sure that the piece of software is in fact distributed by the entity that claims to be the distributor. (2) Hardly any users will know what an authenticity verification by "VeriSign Class 3 Code Signing 2001-4 CA" is, or whether it is trustworthy. (3) The security warning violates established usability principles such as speak the user's language (Nielsen & Molich, 1990) and support internal locus of control (Shneiderman, 1998).

At the bottom line, there is no alternative to permit the installation and execution of the software, unless the user is prepared to discard the e-banking systems altogether. This may overshadow the user's legitimate security concerns.

**The user's awareness of and ability to successfully perform the security subtasks during installation**

The security subtasks of installation are password definition and signature-file management. The user is prompted to define a password and a path (possibly a suggested default) for the signature file during installation.

When prompted for password definition, the user is about halfway through installation and has been presented with a browser message such as the one in Figure 1. Unfortunately many users at this stage may have already refrained from understanding the relevant security issues, started to disregard his or her security concerns, and may proceed simply by following instructions and accepting defaults. Keeping in mind that the user-defined password serves to protect the private key from disclosure, the dialogues for creating and changing passwords display severe shortcomings:

*Not enforcing strong passwords*. Four of the five PKI-based banks do not enforce creation of strong passwords. In fact, three banks accept passwords that consist of multiple instances of the same character (e.g., 'aaaaaaaa') and one bank accepts passwords consisting of a digit followed by a sequence of instances of the same letter. Though Ebank3 enforces several strong-password rules the only rule consistently enforced by the banks is that passwords must be at least eight characters long.

*Insufficient information about strong passwords*. Three of the banks provide no information about what constitutes a strong password. These banks leave it entirely to users to be aware of possible threats, to realize the consequences of these threats in relation to password creation, and to think of ways of creating strong but memorisable passwords. The two remaining PKI-based banks warn against using real words and personal information (e.g., phone number). One of these banks also suggests how to create strong passwords, including several examples. The examples make the suggestions easier to understand but are at the same time a security weakness because experience shows that a disproportionate number of users will choose the examples, which are visible to attackers, as their passwords.

*Preventing strong passwords (!).* Three of the banks restrict passwords to letters and digits, and Ebank3 uses case-insensitive passwords. This significantly reduces the number of possible passwords, and increase the vulnerability to dictionary attacks. (Of course these simplifications make password typing easier.)

Management of the signature file may involve defining a non-standard file path, or storing the file on an external medium instead of on the computer. Such measures may reduce the risk of theft of the signature file, but are only partly supported. An obvious and easily predictable task is a change of password in the situation where a backup copy has been made. Only two of the five PKI-based e-banks tell the user, during the dialogue for changing passwords, that a new backup should be made (the reason is that the old backup still needs the old password for decryption of the private key). There are many other further issues in signature file management, such as transferring it to a new computer, that are not explained to the user at all.

## THREE APPROACHES TO USABLE SECURITY

Several of the weaknesses of the e-banking systems discussed above could, in our judgement, be overcome without fundamental changes in the design of the systems. Suggested changes include explaining and enforcing strong passwords and using the same code for temporary and permanent identification, and using a consistent terminology throughout in the same user interface. However the confusing browser messages (found in all six e-banks) and the complexity of the installation task (the five PKI-based e-banks) would remain, suggesting that one at least considers more fundamental changes.

Three approaches to the design of usable secure systems are illustrated in Table 4. Automation merely requires that users know of security in the sense that they are aware of the existence and importance of security. However, automation alone is not enough because security is critically dependent on steps and precautions taken by users. Instruction focuses on providing users with the how-to knowledge they need to complete security tasks in a step-by-step manner, without necessarily knowing why steps taken or how individual steps are related to each other. Understanding attempts to provide users with knowledge of security principles and relies on users to understand these principles sufficiently well to be able to take competent action in an unspecified range of situation. It may be useful to discuss usable security from the point of view of how various designs combine these three approaches. The six e-banking systems are biased toward instruction, and one may consider if their usable security could be improved via greater emphasis on automation and/or understanding.

### Instruction

The advantage of instruction, the main approach of all six e-banks, is that the complexity inflicted upon users is reduced by instructing them what to do at each step of the process. This can be viewed as a standard approach, and as already noted, Whitten and Tygar's definition of usable security appears to assume that there is a given set of security tasks which the user can be made aware of and instructed about how to perform.

However, instructions can only cover tasks predicted by system designers; they are useless when users are faced with novel tasks not covered by the instructions. This is a critical limitation because unpredicted tasks tend to be quite frequent in complex domains (Rasmussen et al., 1994). The most glaring example of the limitations of instructions in relation to the e-banking systems is that part of the functionality and user interaction is outside the control of the e-banking systems because it is performed by the browser. Users may have different browsers, different versions of the same browser, or their browsers may be set up differently. This defeats thorough instruction because neither the sequence of steps a user will go through nor their exact contents is known to the e-banking system. Relying on instruction in such a situation involves a risk of confusing and frustrating users due to mismatches between the instructions and the actual system behaviour. Since instruction lends itself to a dialogue where the system is in command and the user is assigned a residual role of filling in missing pieces of information, instruction also entails a risk of compromising security by confronting users with issues not covered by the instructions and which they need to make informed decisions about.

### Automation

Automation may be considered as a road to simplify the user interfaces of the e-banking systems in this study, for example to minimize the number of steps users must go through (cf. Tables 2 and 3). One should note that the processes already automated by the e-banks are complex. For example, the PKI-based e-banks partially automate key generation and digital signing of a payment transaction. Of the seven steps listed by Schultz et al. (2001) as typical for digital signing, four are automated. This is due to the integration of the digital signature into the e-banking system.

A candidate for further automation would be a 'Remember password' function, for the password that protects the digital signature. Schultz' et al. we find that this would introduce a security task that is likely to be unacceptable (when the legitimate user may leave the computer unattended). A better way to reduce the number of times the password must be typed is to allow for packaging of transactions, similarly to collecting multiple items in a shopping basket prior to purchase on an e-commerce site.

Another candidate for automation is the browser-based dialogs, because the browser messages are confusing and the user always must type the same answer, eg., "Yes" in the dialog shown in Figure 1. It might be possible to automate this step if the banks provide the user with a specialized user interface, for example a preconfigured browser dedicated to e-banking. However, this approach appears as a step backwards to PC-banking, the previous generation of electronic banking, that eliminates the advantage for users (in terms of convenience) and banks (in terms of maintenance costs) of reusing existing and well-known web-components.

A third candidate for automation is the selection of a pathname for the signature file. However, removing the option of selecting a non-default path weakens security, because it makes it easier for hostile programs,

transferred by a remote attacker to the user's computer, to obtain a copy of the file. All in all, further automation does not seem to be a viable road given the web-based architecture and the requirement that authentication is based, in part, on a secret the user knows, as required – rightly, we think - in the Danish industry code for e-banking.

### Understanding

A major obstacle to providing the user with an understanding of security aspects of e-banking is the unforgiving nature of security breaches, cf. the barn door property mentioned in the Usable Security section. Widely used usability heuristics such as 'permit easy reversal of actions' (Shneiderman, 1998) carry with them the underlying assumption that designs should encourage exploration or, at least, allow for a trial-and-error approach to learning how to use systems. However, for e-banking and other security-sensitive systems, a trial-and-error approach is generally not acceptable because a security breach caused by an error may be exploited by an attacker before the error is revoked by the user. 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. This adds to the demands involved in learning to use these systems and sets this learning process apart from how most users learn to use text processors, spreadsheets, and similar systems.

Several questionnaire studies have attempted to identify the factors that influence the adoption of e-banking systems (e.g., Liao & Cheung, 2002; Sohail & Shanmugham, 2003; Tan & Teo, 2000). These studies unanimously find that security is important to users. A significant portion of the security understanding users must have to perform their e-banking transactions in a safe manner is not specific to e-banking - in fact, rather it is a general requirement for behaving safely and competently in a digitally networked world (e.g., Claessens et al., 2002; Schneier, 2000; Whitman, 2003).

However, most users may not be well motivated to learn about security but rather tend to experience it as an obstacle that complicates and slows down their business with their e-bank. Users may be tempted to put off learning about security, since they generally do not sit down at their computers wanting to scrutinize the consequences of suggested security defaults. The required security skills and active involvement may be more than most users are prepared to deliver.

| **Approach to security** | **Automation** | **Instruction** | **Understanding** |
|---|---|---|---|
| User's frame of mind | Trust | Procedural compliance | Active involvement |
| Task coverage | Formalized tasks only | Predicted tasks only | Predicted and unpredicted tasks |
| Type of user knowledge | Awareness knowledge | How-to-knowledge | Principles knowledge |

Table 4. Three approaches to usable security. From a pragmatic point of view, instruction appears the main approach, but to handle complex security issues aspects of automation and understanding must be incorporated.

## CONCLUSION

Attaining security and ease of use in e-banking at the same time is difficult because security is not a system feature that can be provided automatically while users focus on their primary goal of accomplishing their business with their bank. In the six surveyed e-banking systems, security and ease of use are traded against each other at three levels:

The *individual-features* level. Passwords provide a prominent example of a feature that involves a trade-off between security and ease of use. Four of the e-banking systems do not enforce strong passwords. Further, three of the systems have drastically restricted the set of characters that can appear in passwords and thereby prevented strong passwords. Some of these restrictions – for example case insensitivity – make passwords easier to use by avoiding frequent usability problems.

The *system-architecture* level. Five of the e-banking systems are Public Key Infrastructure solutions and the last e-banking system makes use of one-time codes drawn from a physical card. Both types of security solutions entail considerable complexity within three areas: the number of steps users must go through to complete transactions, the number of security concepts with which users are presented, and the number of security-related codes users must key in during system installation. By relying on the user's browser as their basic platform, the e-banking systems get access to a standard set of well-tested communication and security facilities. However, in making use of these facilities the e-banking systems give up some control

over how security issues are negotiated with users, and unless the user is prepared to discard the e-banking system there is no alternative to granting the browser the privileges it requests and, hence, no reason to scrutinize even legitimate security concerns.

The *basic-assumptions* level. The six e-banking systems share the assumption that information provided to users should be step-by-step instructions stipulating how to perform specified tasks. The instructions enable users to perform the specified tasks without concerning themselves with the underlying security issues but at the same time leave users without the understanding required to act competently in situations not covered by the instructions. This is a critical trade-off because, on the one hand, e-banking users cannot generally be expected to be motivated to learn about security but, on the other hand, they need to understand it well enough to avoid unsafe actions. The bias toward instruction gives priority to ease of use in predicted situations at the expense of making the systems more difficult to use safely in an unknown range of unpredicted situations.

We conclude that most users are only able to complete installation of the e-banking systems by painstakingly following instructions, accepting defaults, and refraining from any real understanding of the involved security issues. However, the high penetration of e-banking systems in Denmark suggests that many people are prepared to trust e-banking systems in the absence of compelling evidence in the form of publicly known security breaches leading to customer's financial losses.

## REFERENCES


Adams, A., Sasse, M.A., and Lunt, P. (1997) "Making passwords secure and usable" in *People and Computers XII: Proceedings of HCI'97*, Springer, Berlin, pp. 1-19.

Beyer, H., and Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann, San Francisco, CA.

Claessens, J., Dem, V., Cock, D.D., Preneel, B., and Vandewalle, J. (2002) On the security of today's electronic banking systems, *Computers & Security*, 21(3), 257-269.

Dewan, R., and Seidmann, A. (eds.) (2001) Current issues in e-banking (Special section), *Communications of the ACM*, 44(6), 31-57.

Dourish, P., and Redmiles, D. (2002) "An approach to usable security based on event monitoring and visualization" in *Proceedings of the 2002 Workshop on New Security Paradigms*, ACM Press, New York, pp. 75-81.

Hertzum, M., Juul, N.C., Jørgensen, N., and Nørgaard, M. (2004) *Usable Security and E-Banking: Ease of Use vis-à-vis Security.* Technical Report. URL: http://www.ruc.dk/~nielsj/research/papers/ebanking-tr.pdf.

IDC (2003) *Total IT security market – including software, hardware, and services – to reach $45 billion by 2006, according to IDC* (press release), IDC, Framingham, MA. Available at: www.idc.com/getdoc.jhtml?containerId=pr2003_01_28_085549. Consulted: September 19, 2003.

ISO 9241-11 (1998) *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability*, International Organization for Standardization, Geneva.

ISO/IEC 9126-1 (2001) *Software Engineering – Product Quality – Part 1: Quality Model*, International Organization for Standardization, Geneva.

ITU (1991) *Security Architecture for Open Systems Interconnection for CCITT Applications. Recommendation X.800*, International Telecommunication Union, Geneva.

Klein, D.V. (1990) "'Foiling the cracker': a survey of, and improvements to, password security" in *Proceedings of the Second USENIX Security Workshop*, USENIX, Berkeley, CA, pp. 5-14.

Liao, Z., and Cheung, T. (2002) Internet-based e-banking and consumer attitudes: an empirical study, *Information & Management*, 39(4), 283-295.

Morris, R., and Thompson, K. (1979) Password security: a case history, *Communications of the ACM*, 22(11), 594-597.

Nielsen, J., and Molich; R. (1990) "Heuristic evaluation of user interfaces" in *Proceedings of the ACM CHI '90 Conference on Human Factors in Computing systems*, ACM Press, New York, pp. 249-256.

OECD (2001) *Electronic Finance: Economics and Institutional Factors*, Occasional Papers No. 2, Financial Affairs Division, Organisation for Economic Co-operation and Development, Paris, France.

Pinkas, B., and Sander, T. (2002) "Securing passwords against dictionary attacks" in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ACM Press, New York, pp. 161-170.

Rasmussen, J., Pejtersen, A.M., and Goodstein, L.P. (1994) *Cognitive Systems Engineering*, Wiley, New York.

Schneier, B. (2000) *Secrets and Lies: Digital Security in a Networked World*, Wiley, New York.

Schultz, E.E., Proctor, R.W., Lien, M.-C., and Salvendy, G. (2001) Usability and security: an appraisal of usability issues in information security methods, *Computers & Security*, 20(7), 620-634.

Shneiderman, B. (1998) *Designing the User Interface: Strategies for Effective Human-Computer Interaction, Third Edition*, Addison Wesley, Reading, MA.

Smetters, D.K., and Grinter, R.E. (2002) "Moving from the design of usable security technologies to the design of useful secure applications" in *Proceedings of the 2002 Workshop on New Security Paradigms*, ACM Press, New York, pp. 82-89.

Sohail, M.S., and Shanmugham, B. (2003) E-banking and customer preferences in Malaysia: an empirical investigation, *Information Sciences*, 150(3&4), 207-217.

Tan, M., and Teo, T.S.H. (2000) Factors influencing the adoption of internet banking, *Journal of the Association for Information Systems*, 1, Article 5. Available at: http://jais.isworld.org/articles/default.asp?vol=1&art=5. Consulted: December 5, 2003.

Whitman, M.E. (2003) Enemy at the gate: threats to information security, *Communications of the ACM*, 46(8), 91-95.

Whitten, A., and Tygar, J.D. (1999) "Why Johnny can't encrypt: a usability evaluation of PGP 5.0" in *Proceedings of the 8th USENIX Security Symposium*, USENIX, Berkeley, CA.

## ACKNOWLEDGEMENTS

This study was supported, in part, by the Development Centre for Electronic Business, Copenhagen Business School, and the IT University of Copenhagen.

## COPYRIGHT