UNIVERSITY OF GOTHENBURG

User Authentication: Passive USB tokens, an alternative to Passwords?

Kristoffer Rosvall
Kristofer Niklasson

Bachelor of Applied Information Technology Thesis

Report No. 2009:064
ISSN: 1651-4769

University of Gothenburg
Department of Applied Information Technology
Gothenburg, Sweden, May 2009

Abstract

Most personal computers today employ knowledge based user authentication e.g. password protection. Nevertheless, password protection is regarded as insecure. This paper investigates if a token based authentication solution for Windows XP (using passive USB storage devices), can counter any of the documented issues inherent with password based authentication. An architecture for such a solution aimed at the home/small business market is presented and evaluated.

The research method is based on the principles of Design Research. The achitecture and prototype is developed using prototyping. A litterature study provides a theoretical framework used as a basis for architecture evalutation. This paper's contribution is mainly an architecture of a token based authentication solution (for Windows XP), capable of addressing many of the known problems regarding password use. Conclusions include the observed pros and cons of the suggested solution, as well as recommendations regarding areas of improvement and future research.

Keywords user authentication, token, password alternatives

1 Introduction

This paper presents and evaluates a prototype architecture for a token based authentication solution. The architecture is evaluated against findings from a literature study, and empirical findings from prototype development.

The research question is: *In the context of home or small business environments, can a token based authentication solution, using passive USB tokens, counter existing, or introduce new issues, compared other authentication methods described and categorized in available literature?*

The literature study investigates the user acceptance and efficiency of the three groups of authentication methods. (*When used for user authentication and not user identification.* E.g. to verify an identity claimed by a user, not to identify a user.) The paper will in part discuss applications of these methods in e-banking, large scale corporate environments and alike. However, only when the authors feel such analysis will help them in the drawing conclusions regarding what would be a suitable alternative for the home/small business market. The scope of this paper will not permit a complete review of available published material but is limited to a mere subset.

The prototype serves as a means of investigating the Win XP applicability, and as a facilitator of empirical findings.

1.1 Background

Much of your information stored on information systems, such as private files on your PC or conversations on your e-mail is meant to be for your eyes only. In order to keep them that way, most of these systems employ some form of user authentication. User authentication is of the essence in order to ensure that only allowed users access an information system. Authentication methods are most often evaluated in terms of effectiveness (i.e. ability to reject unauthorized and accept authorized users), cost and user acceptance(i.e. to what degree the users find the method acceptable).

Research indicate that 97% of organisations use passwords as a means of authentication[1]. Furthermore, numerous resarchers have concluded that users do indeed trust these systems. Nevertheless, passwords do have a well documented history of being insecure. Part of the reason being user behavior[2, 3, 4], part due to vulnerability toward social engineering and hacking efforts[5].

Failings in user authentication causes substantial financial loss; in 2007 US$3.2 billion was lost to phishing [6]. A white paper[7], stipulates that corporations can reduce cost, by implementing alternative methods of authentications (SmartCards), due to a decrease in password related support calls.

These reasons are why the field of user authentication is important. The consequences of it failing involve loss of privacy and money. Furthermore, efficient user authentication can reduce costs.

2 Theory

Determining if a user is a valid or invalid user of a system is the art of user authentication. User authentication is a two step process: First the system needs to identify the user, second, that user's identity need to be verified. This process can be conducted using a number of technologies. User identity is verified using either something the user knows (such as a password), something she has (e.g. a key or other token) or something she is (i.e. fingerprint, voice or other biometric trait). Consequently these groups are often used to categorize authentication techniques. We will use the terms knowledge based, possession based and characteristic based authentication to describe the aforementioned groups.

The central issue related to user authentication is balancing security with ease of use. One wishes to maintain high security whilst not making the process of authentication too cumbersome, stress inducing or intrusive[3].

2.1 Knowledge based methods of authentication

Knowledge based methods of authentication are the most common and does indeed enjoy high acceptance among users[1]. Basically a user's identity is successfully verified if she can provide the secret knowledge linked with her user account such as a password or pass phrase [8].

Knowledge based systems are, though popular, only as secure as the secret information they employ. The key concern is that the chosen secret must be hard to figure out (by computers or humans) yet easy enough to remember without resorting to writing the information down or otherwise duplicating it. Key fields relating to passwords being researched are password generation, recall and guessability[2, 9]. Hence extensive research aimed at developing alternatives has been conducted, as the traditional use of passwords require a complexity that strains the bounds of human memory capabilities[10, 11]. Some of these alternatives are built around the idea of association, others around theories relating to human memory.

A pass phrase is simply a phrase used as a password (e.g. "roses are beautiful bananas are slippery"), generally longer then a password. Keith et al.[10] reasons that an increase of password length render a greater number of possible combinations than an increase of the character set used. Their research also indicates that, even though longer, pass phrases are no harder to remember then conventional "strong" passwords. Keith et al. believes that this is due to aspects of the human memory described in Millers Chunking Theory.

Associative password systems are, as the name implies built around the idea of word association. The user is logged on to a system by supplying a series of response words corresponding to cues (e.g. ocean, blue) selected from a template (one per user)[8]. Zviran and Haga refined this concept when developing cognitive passwords[11]. As with associative passwords each user has a template stored in the system, it contains answers to a number of fact and opinion based questions e.g. *"What is your mother's maiden name?"* or *"What is your favorite kind of flower?"*. If the user can answer a subset of these, she is successfully authenticated.

Passwords are usually stored in a database. Login is achieved by comparing a user supplied password to the stored equivalent using the given user name. A hacking effort is either directed toward a single user or aimed at gaining access to the entire database[5, 12]. Many efforts to prevent or mitigate the effects of a single stolen password will impact the security of the database itself. For instance, one of the core security problems retaining to knowledge based authentication includes password reuse and password aging[1, 5]. Password aging occurs when passwords are not changed frequently, allowing an intruder access to a system for substantial periods of time by way of a recovered password. A preventive course of action taken to combat this is enforcing regular password changes for all accounts. Such an effort would mitigate the effects of a compromised database as well[5]. Studies have shown that this is not common practice[1], nor user friendly as such a requirement leads to password duplication or users resorting to writing down authentication information, thereby compromising system integrity[3].

Another noteworthy issue with most authentication methods is sniffing, i.e when an intruder obtains a users authentication information as it is transmitted between user and system. Knowledge based systems are particularly vulnerable to Van Eck and keystroke sniffing. A password can be captured off the screen as it is being displayed by reading the Van Eck radiation. Keystroke sniffing is the act of secretly reading keystrokes and capturing information as it is being typed. Van Eck attacks can be fended off by assuring that passwords are not displayed on screen. Keystroke sniffing attacks by assuring that the keyboard memory buffer can not be read by any software except the OS. However knowledge based authentication is very vulnerable against sniffing attacks in general. Typically passwords are stored and transmitted as hashes e.g. obscured but such passwords can still be exploited[12].

One example of such an attack took place in one of Kinko's facilities; keystroke-capture software were installed on a computer, the software sent over 450 user names and passwords to an intruder that attacked the users bank accounts[5].

All knowledge based authentication approaches is susceptible to social engineering. Social engineering is the art of manipulating a user to willingly supply authenticating information to a third party. Often by using trickery or simply asking for the information, possibly misrepresenting oneself or ones intent [13].

2.2 Possession based methods of authentication

Possession based authentication methods use something the user own to verify user identity. An every day example of this is the way we gain access to a car. If we have the car key we are believed to be the rightful owner (or driver) of that car. In authentication, the item we possess is called the token. All tokens contain a base secret which is the equivalent of the secret in knowledge based authentication systems.

Primarily one differs between passive and active tokens. Passive tokens does not conceal the base secret but submits it as it is(e.g. ATM cards). In contrast active tokens (e.g. SmartCards) have the capability to process the base secret in some way and display the result of that process (i.e. a One-Time password) rather then the base secret itself [12]. SmartCards are active credit-card sized plastic tokens with a small processor, capable of encryption. There are active tokens with the same capabilities as a SmartCard utilizing USB connectors. The gain of using such tokens is that USB has become somewhat of a standard interface, hence no additional hardware is required. Furthermore, USB is faster in comparison to conventional SmartCard readers[14].

Token systems often make use of so called Two-Factor authentication e.g. one combines several authentication methods such as a Magnetic Card (e.g. ATM card) and PIN code[15]. Other combinations such as characteristic based methods in conjunction with SmartCards have been proposed[16].

The concept of Two-Factor authentication also applies to Challenge-Response authentication schemes. Such schemes enable a user to be authenticated without transmitting the actual base secret. This is achieved by having the server issue a challenge to the user, e.g. enter a number sent by the server in the token. The token generates a One-Time password using the issued numbers and the base secret (the latter encoded in the token). The same computation is done serverside. If the results on both sides match the user is granted access. The goal of using One-Time password is to eliminate re-use of a sniffed password, as such passwords are only valid for a limited time period[17]. They are becoming increasingly common in e-commerce and banking[18]. However, according to Schneier[17] Two-Factor authentication do not protect against Man-In-The-Middle-Attack [12] or Trojan attacks. Other suggestions include using SSL/TLS session-aware user authentication[19] in an effort to prevent the aforementioned Man-In-The-Middle-Attack .

Token Theft can be made more cumbersome by employing Two-Factor authentication e.g. require a PIN code before use. Tokens without key pads can employ this strategy by requiring the PIN code prior to, or in conjunction with the token.

2.3 Characteristic based authentication

All biometric systems are built around the same principle. Users enroll by suppling a user name and creating a biometric signature template (i.e. a template that the system use to authenticate the user against). When authentication commences the system captures the user's signature (e.g. reads her fingerprint) and compares it to the template. If the user signature matches the template to a sufficient degree, the system authenticates the user, otherwise the user is rejected.

Advanced biometric systems combines several readings when creating the template. This is done to capture subtle variations between readings [12]. The reason is that template and user signature are matched through comparison, unlike passwords the match is never exact. A threshold determines the degree to which a sample must match the template. This fact is the basis for false rejection (of valid users) and false acceptance (of unauthorized user) [12]. Different thresholds will result in different false acceptance and rejection rates(FAR and FRR). The optimal threshold would be the value where the two curves coincide (Equal Error Rate). However this is seldom the case as the designers often premiere one over the other[20].

Biometric technologies are often divided into two sub categories, one being those that are physiological, the other those being behavioral. Much of the research on the subject indicate that behavioral biometrics have higher acceptance among users then physiological biometrics , but it is under debate[21].

Security threats toward biometrics include Replay, By-Passing and Overloading Attacks. All biometric techniques are potentially vulnerable against replay attacks, i.e. an intruder replays a valid biometric signature for the system's sensor and is accepted as a genuine signature. A variation of this type of attack could be enacted utilizing a forged trait (such as a false thumb)[22]. Forged traits can be detected by way of a liveness check in conjunction with reading the sample (e.g. ensuring the presence of a pulse) [8]. Sniffing is an issue in biometrics as with most methods. Such an attacks can be made more costly for an intruder. For instance, the system could alternate the trait used for authentication, hence forcing an intruder to intercept a multitude of biometric readings. Another approach is encrypting the biometric signature as it is transmitted. This would require the use of encryption keys, which might diminish the benefits of a biometric system. Furthermore one can employ authentication techniques in order assure that the reader itself is trusted. This is useful in distributed environments[12].

A By-pass attack aims at circumventing the entire biometric authentication mechanism (i.e. bypassing it). This is achieved by compromising the trait, making it hard for the system to obtain a good enough template. The desired effect is to be switched to an alternative, less secure authentication system. This attack relates to an (as of yet) inherent problem relating to biometrics: The persistent nature of biometric traits. I.e. a biometric trait cannot be changed nor replaced if compromised or otherwise unsuitable(e.g. sniffed, copied or damaged by injury)[23], however, IBM is working on developing Cancellable Biometrics [22]. Furthermore Lee[22] mentions overloading attacks; i.e. to weaken or circumvent the authentication method by overloading the biometric reader itself resulting in diminished security.

2.4 Use and awareness of Two-Factor authentication

Two-Factor authentication methods such as SmartCards are being used more and more. Researchers have seen an increase in use, for instance Forrester Research forecasted that in Europe alone, over 130 million people will be using remote e-banking services by 2007, up 75 million compared to 2005[15]. According to RSA CSO perspective survey , 25% of respondants use Two-Factor authentication (with SmartCards) moderatly in their cooperation, 8% use it universally. Approximatly 11% employ SmartCards alone for authentication purposes moderatly in their organization while 4% use the solution universally within their organisation[24]. A pilot study in England investigating the use of Two-Factor authentication at the point of sale (as a replacement of signatures) showed that 83% of the customers was positive about the launch of such a system[25].

Granted the these studies differ in contexts, some not directly relevant to this paper. However, the results above could indicate the status quo regarding possession/ Two-Factor authentication methods.

A survey testing the security and usability of three Two-Factor authentication solutions utilizing active tokens in e-banking, showed that users preferred a simple token generating a One-Time password, over an alternative using a Challenge-Response scheme in conjunction with a PIN Code. The reason being that the latter was more time consuming to use and harder to learn [18].

2.5 Attitudes toward biometrics

Deane et al.[21] found behavioral biometrics technologies had lower acceptability then physiological alternatives. Out of the physiological biometrics technologies included in their survey fingerprint and hand geometry were the only technologies found acceptable(based on a mean acceptability rating among respondents). Among the generally lesser accepted behavioral biometrics methods only voice recognition was considered acceptable. The high acceptance of fingerprint use is strengthened by other research as well: In a study by Magnusson and Giarimi[26] 95% of those positive towards use of biometrics preferred fingerprints, 44% preferred voice biometrics. Further indications of fingerprint being the trait of choice is found in Furnell et al's. survey from 2005 where fingerprint had the highest preference next to passwords[1].

Another interesting observation made by Dean et al. is that as the perceived sensitivity of information increases so does the perceived acceptability of some biometric authentication methods (specifically keystroke verification and fingerprint recognition). This increase might help justify the fact that both keystroke analysis and mouse dynamics was ascribed high acceptance when used for continuous authentication by other researchers[1]. (Note however that Dean et al. do not differ between continuous and "at log in" authentication in their survey.)

Retina scanning was classified as unacceptable in Dean et al.'s survey based on mean values, however, in percentages 49 % considered the method unacceptable[21]. Assuming that the remainder concidered the technology acceptable to some degree, it is in the same range of acceptance as iris scanning had in Furnell et al.'s study[1] (47% when used for initial login), and 44% in Magnusson et al's study (when applied on mobile phones)[26]. Indeed, these results are not comparable, some pertain to the context of mobile phones. Furthermore Dean et al. measured acceptance of retina scanning as opposed to iris scans. However, the fact that the results are in the same range could indicate the general acceptance level of both these methods. Such reasoning is strengthened by the fact that a later survey by Furnell et al.[4] show similar results regarding acceptance of iris scanning as a authentication method (for mobile phones).

Both Dean et al. and Furnell et al. acknowledge trust, and specifically Work Performance Monitoring, as important factors in achieving acceptance of biometric authentication methods. User awareness of the policies surrounding monitoring is crucial in order to successfully launch such systems[21, 23], a fact that resonates in Furnell et al. survey result where in excess of 80% states the importance of user awareness. A survey showed that 40% of participants would consider monitoring an invasion of privacy, 45% felt that they would not trust their organization to use collected information soley for security purpouses [1]. Dean et al. speculate that the fear of WPM might partially explain behavioral biometrics methods lesser acceptance in their study[21].

Another facet of trust central to biometric authentication is privacy: Research indicate that there is a a remarkable difference in acceptance depending on where the biometric template is stored. In cases where the user is the only one having access to the biometric template the positive responses ranged from 72% to 83%[26] while lesser when others had access. This preference is evident in another suvey where 50% prefered local storage of biometric templates(34% prefered network storage)[4]. Bernecker[23], along with others[16], promote storing of the template (i.e. the secret) using a SmartCard carried by the user as the permanent housing of the template. Such a solution would also conform with the principle of keeping information stored on the authentication site to a minimum[27].

2.6 Dual relationship between memory and password security

As stated in 2.1, passwords need to be complex enough to withstand cracking attempts, and changed regularly in order to prevent password aging. Guidelines and recommendations aimed at achieving this, is however largely ignored due to the resulting effects on memorability. Results from one study indicated that 70% of participents could not remember a password they had created a week earlier[28]. Bersch found that only 35% could recall a non-assigned password after three months (23% when the password was assigned)[2]. The tendency to forget passwords is indicated further by fact that results from a 2004 survey showed participants forgot their passwords between one and two times every month depending on the number of passwords used by the respondent[9].

The poor recollection of passwords is understandable when considering that most people use multiple password protected systems concurrently[2, 1, 4]. Researchers have found that the use of multiple password protected systems lead to password duplication. A survey showed that only 7,1% of the participants used a unique password on each system. The survey also showed that 65,1% of all services shared a password[2].

Apart from duplication, users tend to forego regular password changes: Furnell et al. found that a mere 28% changed passwords monthly or more often (the remainder doing so twice a year 18%, more seldom 20% or never 34%)[1]; A survey on mobile phone practices reveal even more compelling results as a total of 13% reported changing their PIN code monthly or yearly, 45% never did it (remainder changed pin at purchase)[4]; Brown et al's survey showed that 11,9% of respondents where forced to change their passwords[2].

Ignoring password aging and duplicating passwords to this extent exposes the users to the so called domino effect, where an intruder by way of one recovered password could access several other systems, e.g. compromising several systems[5]. If an entire database of authentication were to be compromised, the effects could be devastating. However, solutions such as the one proposed by Szydlo et al.[29] aim to make such an attempt more cumbersome by incorporating two or more servers in the authentication process. As aforementioned (in 2.1) tools, such as keyloggers can be used to ultimately exploit the domino effect in a password system. Online fraud do indeed result in substantial loss: In March 2007 APACS released statistics revealing that on-line banking frauds was increasing in the UK, in 2005 it was a £23.2 million issue; the following year the amount was £33.5 million[15]. Bruce Schneider[17] reasons the use of Challenge-Response and One-Time passwords do mitigate the effects of sniffing attacks and compromised databases. However Schneier also observes that Man-In-The-Middle-Attack nor Trojan attacks would be impeded.

There is a dual relationship between ease of use and security of a system recognised by several authors[3, 1]. As they imply that the use of stronger passwords and polices (regarding password change etc.) may compromise simplicity of use. When passwords become complex, along with the use password changing policies, users resort to other harmful behavior such as writing passwords down, along with further password duplication. One user was cited saying *"...because I was forced into changing it every month I had to write it down."*, a practice adopted by 50% of asked users[3]. Other researchers have found evidence supporting this prevalence: A 2004 survey showed that 27% of participants admitted to writing down their passwords[9]; Dhamija et al. found that a vast majority of participants wrote down all or a subset of their passwords[28]; Bunnel et al. that in excess of 40% admitted to writing down thier password in order to remember it (an assigned password), one fifth when memorising a self-generated password as well[30].

Knowledge based authentication methods developed in an effort to lessen the memory strain inherent with strong passwords include the Pass phrase, associative and cognitive passwords. Key when evaluating these methods is recall and guessability rates.

Zviran and Haga are two prominent researchers advocating cognitive passwords. Their studies of recall and guessability showed that fact based questions was easier to recall then opinion based[11], average recall rates after 3 month among respondents was about 78% (fact based 88,3%, opinion based 74%). Later the same year the authors published another study with similar results [31]: average recall rates after 3 month among respondents was about 83% (fact based 94%, opinion based 88%). In both studies Recall rates where significantly better compared to convensional passwords. Their 1993 study[32] revealed somewhat lower recall rates (an all over average of 68%, 83,7% fact based, 70% opinion based).

Bunnell et al.[30] raises concerns about Zviran and Hagas 1993 study[32], criticizing the extensive timespan and the testing of multiple password knowledge based alternatives. Furthermore they note that Zviran and Haga did not investigate guessability. Bunnell et al. study mimics that of Zviran and Haga, but use a shorter timespan, a larger set of cognitive questions, and they generate cue words (asking participants to generate only response words). Their results are very similar with 88% recollection of the fact based items and 72% of opinion based. Their investigation of guessability reveals that 56% of fact based and 23% of opinion based items where guessed by significant others.

Associative password recollection results differ significantly between the researchers studies 69%[32] versus 39%[30]). Bunnell et al. reason that their poor results might be attributed to the fact that they chose to use cue words that where known to produce heterogeneous responses in an effort to reduce guessability (which was indeed low). This fact might have impacted memorability in a negative way[30].

Research indicates that even though longer, pass phrases are no harder to remember then conventional strong passwords. Nevertheless, pass phrases are more prone to typing errors then (both strong and simple) passwords[10, 11].

All varieties of knowledge based authentication are highly susceptible to social engineering. A survey was conducted on the streets of London, people passing by was asked for the password to their computer. In the end 75% of them revealed their password to the researchers[33]. Furnell et al. found that 29% admitted that they have shared their password with coworkers, 21% admitted that they have used someone elses password without their consent[1]. Another study concluded that in working environments, people are liable to share their account information in order to better facilitate working together on group tasks[3]. Granted this kind of behavior does not directly relate to Social Engineering , nevertheless it does indicate that users handle their account information in a manner considered insecure.

3 Research Method

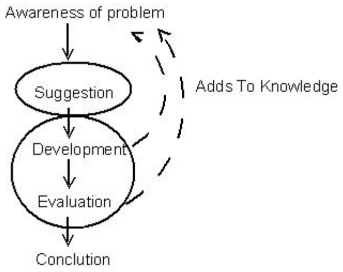

Figure 1: Overview of knowledge flow in the research process

The research method (see Figure 1) is based on the principles of Design Research as described by Takeda et al.[34].

The problem observed by the customer is validated by a short literature study, aimed at finding documented support for the perceived flaws in knowledge based authentication. This activity result in an Awareness of the problem and a research proposal, including a Suggestion on improving the problem. The research proposal also specifies restrictions and scope of the suggested solution (see 3.1.1)

Architecture development is done in an exploratory manner by way of prototyping (described in 3.1). A process that generates a deeper Awareness of the problem, while allowing refinement of the Suggestion itself. The final outcome of prototyping process is an architecture (4) and a partial implementation of the same.

After development, a more comprehensive literature study (3.2), extending the initial background research is conducted (section 2). Findings from this effort is used as a framework in Evaluation of the Suggestion.

Evaluation is done by observing what deficiencies, in knowledge based authentication the architecture can mitigate or counter. Evaluation also includes discussing risks introduced by and possible improvements of, the architecture (i.e. the Suggestion).

Finally, the Conclusions from the Evaluation is accounted for, summing up pros and cons of the architecture, in conjunction with suggesting suitable areas of further research.

3.1 Prototyping

Evolutionary prototyping is well suited for developing systems where the requirements are vague and dynamic[35]. Prototype development is conducted in short, iterative, development cycles in close contact with the customer followed by customer meetings regarding updates and/ or requirements issues. Requirements are derived and refined through semi-structured interviews with the customer. In addition informal use cases and sequence diagrams are tools used, both during the requirements engineering process (with customer) and for documentation.

3.1.1 Scope of prototype

A company working with secure email applications wanted a prototype for a possession based authentication system, replacing the traditional password system default in Windows XP. The purpose was primarily to investigate feasibility and potential security issues countered or introduced by such a solution.

The prototype was to address local login i.e. not addressing remote login scenarios. Architecture is limited to using standard USB storage devices as tokens (i.e. not active tokens).

3.2 Literature study

The primary literature sources are various journal articles found through digital libraries. In addition, select books found through Chalmers University Library Databases or electronic libraries such as Books24x7 are included. Binary search strings such as "User authentication" or "Biometrics" are used when searching for material.

A collection of articles were selected based on title and abstract relevance. A subset of these were selected for a full text review. References deemed relevant were then examined in the same manner.

The literature used is published in 1989 or later. The reason for this is twofold: First literature pre 1989 on relevant technologies tend to be somewhat out of date (e.g. the technological environment changes). Second, focusing on newer sources (dated 1990 or later) results in a more current view of the acceptance concerning the different methods of authentication.

4 Result

4.1 Purpose of Prototype

A company working with secure email applications wanted a prototype for a possession based authentication system, replacing the traditional password system default in Windows XP. The purpose was primarily to investigate feasibility and explore potential fall pits involved in developing such a system.

Scope of prototype The prototype was to be limited to local login i.e. not addressing remote login of any kind. The prototype was to use standard USB storage devices, not active token technologies.

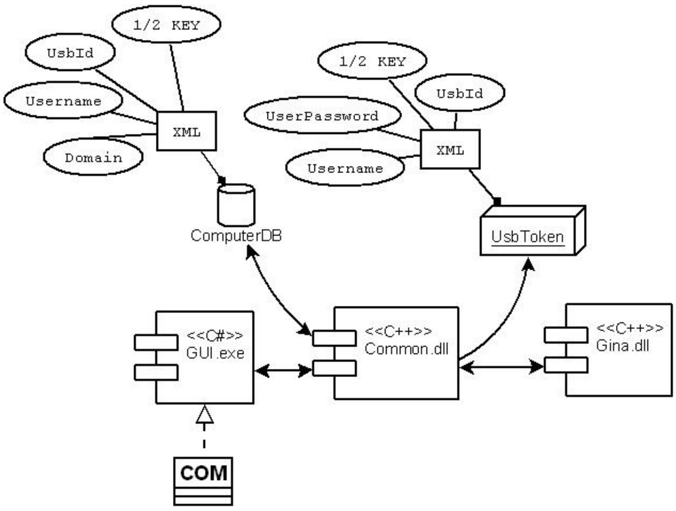

Figure 2: High level illustration of the prototype architecture.

4.2 Architecture description

The architecture (Fig.2) consists of five major components (each of which is described below). The data holding components are the UsbToken and the ComputerDB, the other are three software components; the GUI.exe, Common.dll and Gina.dll. These three will be described only in terms of required functionality, specific function names or equivalent are not specified. The illustration includes suggested programming languages for each component. These are best viewed suggestions and nothing else.

4.2.1 ComputerDB

ComputerDB is the database containing information about all users registered in the system (i.e. stored on the computer). In our implementation we used an XML file for storage. The attributes in the database are:

UsbId is a unique value identifying a USB device (used as token). The UsbId attribute is used to facilitate the destruction of lost tokens. This value is needed in order to determined what user account (i.e. user registered in the ComputerDB) a UsbToken is assigned to. When the system is supplied a UsbToken the UsbId from that token is read. The value is used to search the ComputerDB for the matching UsbId value.

$\frac{1}{2}KEY$ is one half of an encryption key, the other half resides on the UsbToken. When assembled the complete key is used to decrypt the Username and UserPassword attributes on the UsbToken. Provided the system achieved a positive match between UsbId attributes (i.e. in ComputerDB and UsbToken), the two $\frac{1}{2}$KEY attributes are fetched and assembled.

Username is needed when deleting keys from the database. It is stored unencrypted. Not including the (unencrypted) Username in ComputerDB would mean not being able to know what accounts have been assigned a UsbToken without being in possession of the token.

Domain identifies the domain name of the computer protected by the system. This information is needed at login in Windows XP if the computer belongs to a domain.

4.2.2 UsbToken

The UsbToken is a USB storage device used as a user's token. The information about a user's account (detailed below) is stored in a file (An XML file in the prototype implementation). That file is called the key file. Observe that each key file contains information about exactly one user account. The attributes stored in a key file are:

UsbId is matched to the matching attribute in the ComputerDB in order to assert what user account the UsbToken is registered to. In other words, the UsbId is a unique value used to identify a specific UsbToken. The attribute's use is two fold: One, it facilitates the encryption of Username and UserPassword attributes stored on the UsbToken. Two, it is a preventive measure meant to make the task of copying a UsbToken more difficult. Ideally this value is not be stored in the key file, but rather calculated using information about the UsbToken and the key file, the result stored in the ComputerDB at the time of key creation. Then, the UsbId could be calculated and matched with the UsbId attribute stored in ComputerDB at each login. Regardless, Fig.2 depicts UsbId as an attribute of UsbToken because it is considered to be an attribute of a UsbToken.

$\frac{1}{2}KEY$ is one half of the encryption key used to decrypt the Username and UserPassword attributes stored in the key file. The other half resides in the ComputerDB. When a token key has been inserted and a positive match of UsbId attributes have been achieved (between ComputerDB and the UsbToken), the $\frac{1}{2}$KEY attributes are assembled forming a complete key used to decrypt the Username and UserPassword

Username is the encrypted username of the user issued the UsbToken.

UserPassword is the encrypted user password.

4.2.3 Gina.dll

Gina.dll is a part of the Windows XP OS that handles the login procedure in Windows XP. Basically it is the program that shows up when the user is asked to authenticate herself to the Operating System (OS). The program dictates the authentication procedure as it is perceived by the users. Since we are re writing this procedure, this file is overwritten.

The Gina.dll in the architecture determines how, and from where, the data used when authenticating a user is collected, and how it is manipulated. E.g. Gina.dll dictates that the username and password for an account is to be fetched from a UsbToken, and decrypted using the $\frac{1}{2}$KEY attributes from the UsbToken and the entry in the ComputerDB matching the UsbId found on the UsbToken.

Gina.dll use functions found in the library Common.dll for reading data from ComputerDB and UsbToken, as well as decrypting information on the UsbToken.

A basic high level use case scenario of a successful login by the prototype is described below, actors involved are the System (meaning our prototype) and the User (meaning the user wanting access):

1. *The System* asks for a UsbToken
2. *The User* inserts a UsbToken
3. *The System* gets the UsbId from UsbToken
4. *The System* finds the corresponding UsbId in ComputerDB
5. *The System* assembles the encryption key using $\frac{1}{2}$KEY attributes from both ComputerDB and UsbToken
6. *The System* uses the assembled encryption key to decrypt Username and UserPassword on UsbToken
7. *The System* attempts to login using the decrypted Username and UserPassword
8. *The System* evaluates result of attempt
9. *The User* is successfully authenticated and granted access to *The System*

4.2.4 Common.dll

The library Common.dll provides functionality for reading and writing information to and from both UsbToken and ComputerDB. Furthermore it provides functions for encrypting and decrypting Username and UserPassword attributes. Common.dll is used by both GUI.exe and Gina.dll, hence the name Common.dll.

4.2.5 GUI.exe

GUI.exe is the desktop application used for creating, changing, deleting keys for existing accounts in Windows. It has functionality for creating, replacing and deleting UsbToken for Windows XP user accounts, and can only be used by Administrators. It uses Common.dll when reading, writing, encrypting and decrypting data from ComputerDB and UsbToken.

5 Discussion

The literature study revealed strong indications of human memory limitations being a contributing factor to insecure password systems. The multitude of systems utilizing knowledge based authentication used by users lead to memory strain and in turn the adoption of bad password practices. The vulnerability toward social engineering inherent in all knowledge based systems are another contributing factor to these systems insecurity (as described in 2.6). Ergo, the potential gain of using a non knowledge based alternative is apparent.

In an architecture such as the one proposed (see 4), the secret (i.e. password) is stored on the token. Therefore, users need not remember it. Thereby permitting an increased password length and complexity without adding strain to a users memory. Password complexity could be guaranteed if the an implementation of such an architecture could generate passwords for users at enrollment (i.e. assign passwords to existing user accounts). Prevention of password aging can be facilitated by having the system re-generate passwords for user accounts periodically. However, such features are beyond the scope of the implemented prototype.

Of course, memory strain would become a non issue if using a characteristic based authentication solution. Several biometric technologies do indeed have acceptable user acceptance rates according to surveys. However one might argue that biometrics are not suitable because:

All secrets can be compromised or stolen, this includes all three groups of authentication methods mentioned here. One common denominator of knowledge and possession based authentication is the non-persistent nature of the secret(s): If lost or stolen the secret is easily changed, whereas a biometric template or sample is not. Furthermore, a compromised password or biometric sample might go undetected until an intruder causes damage. A token often used on the other hand, is more likely to be missed. In addition, a biometric templates persistent nature makes it highly desirable to intruders, hence secure storage is of the essence. During development we found that secure storage most often raises the question of how to encrypt sensitive information and subsequent questions regarding where to hide the encryption key (the so called hide the key problem). In order to circumvent this problem, the proposed architecture splits the encryption key and stores it in to separate locations (see section 4). This effectively lessens the amount of sensitive information stored in one location.

To mitigate the effects of a lost token, the architecture of our solution is designed in such a way that no information identifying the computer hosting the account is stored on the token. However, one issue introduced by our solution (due to the use of a storage device as token) is the risk of users storing content revealing the location, name or similar of the computer on the token. (Or using a USB device with a company logo, address or similar as token). Another issue linked to token loss/theft is the risk of the token being copied. If obtained, the key file(4.2.2) can be copied and used to make a working duplicate. Our prototype use a generated value as UsbId. Never the less, as is stated in 4.2.2, calculating a value using the data from the USB device itself and the key file stored on it would be preferred. It would provide a way of assuring that the provided USB device is the one used when creating the UsbToken without storing the value on the device. Hence, making key duplication more cumbersome.

One flaw we have observed in our architecture is that we store the user names of accounts issued a key. This is a duplication of somewhat sensitive information. What motivates the duplication is that without access to this information, it would be impossible to know what users have been issued a key without being in possession of it. Such a limitation would make replacing lost keys very difficult.

The very purpose of our solution was to not increase the number of passwords or equal imposed on a user, hence we opted not to employ Two-Factor authentication (i.e. PIN and token). Of course the degree of confidence in the identity of the token holder lessens because of this choice. However an area of future research would be to investigate the use of ones fingerprint for Two-Factor authentication. Besides effectively tying the user to a token, a fingerprint could be used to encrypt account information on the token. If the biometric template is stored on the token, the user would be the only one holding the token. According to our findings in literature, this kind of storage is preferred by users.

One of the major insecurities of knowledge based authentication methods is Social Engineering. Several scholars have investigated guessability rates of such methods. We perceive another risk, not relating to guessability, that has not been mentioned in any of the literature we have studied: In todays society where most information about a person's past is publicly available (e.g. on the Internet), acquiring the information most likely used as answers in for instance a cognitive password system, is probably not that difficult a task. Using an alternative method of authentication such as biometrics or tokens would, if our assumption above has any merit, increase security.

6 Conclusion

Our findings show that knowledge based authentication is despite its documented flaws, the most used and popular method of authentication.

The solution we have proposed eliminates the need to remember passwords for user accounts. (Not including the administrator account, which is needed when creating and replacing tokens, i.e. only one password per machine is required.)

Not having to remember passwords leads to fewer restrictions when generating passwords. Hence permitting an increased complexity and length, thus increasing security. Aging of these passwords can be prevented by having the system regenerate account passwords for users during active sessions, therefore evading what is known as the domino effect.

Token loss is mitigated by the fact the token does not contain any information identifying the computer it is used with. However we would like to stress that policies instructing users not to store such information on the token is needed.

Our assessment is that the only sensitive information possible to obtain from a stolen token is the UsbId and the $\frac{1}{2}$KEY attributes. It is recognised that the $\frac{1}{2}$ KEY could be used in an attempt to brute force a complete encryption key used to decrypt account information resident on the token. Any calculations on the magnitude of such an effort has not been attempted by us as it is beyond the scope of this paper.

Furthermore, as described in both 4.2.2 and 5, storing the UsbId attribute on the token makes it possible to produce a duplicate. Calculating such a value at each login would make duplication more cumbersome. This has not been implemented in our prototype.

Lastly we feel that user acceptance of possession based authentication technologies when used in personal computers is a research area in need of renewal. A great deal of the published material on possession based authentication that we found was either related to corporate/e-commerce applications or out of date.

References

[1] Furnell MS, Dowland PS, Illingworth HM, Reynolds PL. Authentication and Supervision: A Survey of User Attitudes. Computers and Security. 2000;19(6):529–539.

[2] Brown AS, Bracken E, Zoccoli S, Douglas K. Generating and Remembering Passwords. APPLIED COGNITIVE PSYCHOLOGY. 2004;(18).

[3] Adams A, Sasse MA. Users Are Not The Enemy. Communications of the AMC. 1999;42(12).

[4] Furnell S, Clarke NL. Authentication of users on mobile telephones - A survey of attitudes and practicies. Computers & Security. 2005;.

[5] Ives B, Walsh KR, Schneider H. The Domino Effect of Password Reuse. Communications of the AMC. 2004;47(4).

[6] Skinner E. Risk and Reward. Biometric Technology Today. 2008;.

[7] Datamonitor. The ROI case for smart cards in the enterprise; 2004.

[8] Magno MB. Survey of User Authentication Mechanisms. NAVAL POSTGRADUATE SCHOOL MONTEREY CA; 1996.

[9] Sater Carstens D, MacCauley-Bell PR, Malone LC, DeMara RF. Evaluation of the Human Impact of Password Authentication Practices on Information Security. Informing Sience Journal. 2004;7.

[10] Keith M, Shao B, Steinbart PJ. The usability of passphrases for authentication: An empirical field study. Int J Human Computer Studies. 2007;(65):17–28.

[11] Zviran M, Haga WJ. User Authentication By Cognitive Passwords: An Empirical Assessment. In: Information Technology, 1990. 'Next Decade in Information Technology', Proceedings of the 5th Jerusalem Conference on (Cat. No.90TH0326-9); 1990. p. 137–144.

[12] Smith RE. Authentication: From Passwords to Public Keys. Addison-Wesley; 2001.

[13] Karakasiliotis A, Furnell MS, Papadaki M. Assessing end-user awareness of social engineering and phishing. In: Proceedings of 7th Australian Information Warfare and Security Conference; 2006. p. 60–72.

[14] Rao HR, Gupta M, Upadhyaya S. In: Managing Information Assurance in Financial Services. IGI Publishing; 2007. p. 157.

[15] Reavley N. Flexibility,Fraud and Two-Factor Authentication. Computer & Security. 2007;.

[16] Vanderhoof R. Smart cards, biometrics and privacy. Card Technology Today. 2003;.

[17] Schneier B. Two-Factor Authentication: Too Little, Too Late. Communications of the ACM. 2005;.

[18] Weir CS, Douglas G, Carruthers M, Jack M. User Perceptions of security, convenience and usability for ebanking authentication tokens. Computers & Security. 2008;.

[19] Oppliger R, Haus R, Basin D. SSL/TLS session-aware user authentication revisited. Computer & Security. 2008;.

[20] Furnell S, Clarke N. Biometrics: no silver bullets. Computer Fraud and Security. 2005;.

[21] Deane F, Barrelle K, Henderson R, Mahar D. Perceived acceptability of biometric security systems. Computers and Security. 1995;p. 225–231.

[22] Lee V. Biometrics and Identity Fraud. Biometric Technology Today. 2008;.

[23] Bernecker O. Biometric Security: An end user perspective. Information Security Technical Report. 2006;.

[24] Security R. The CSO Perspective on security threats, Data Protection and Identity and Access Management Solutions; 2003.

[25] Chip and PIN gets thumbs up. Card Technology Today. 2003;15.

[26] Giarimi S, Magnusson H. Investigation of User Acceptance for Biometric Verification/Identification Methods in Mobile Units. Stockholm University and Royal Institue of Technology; 2002.

[27] Price G. The benefits and drawbacks of using electronic identities. Information Security Technical Report. 2008;p. 95–103.

[28] Dhamija R, Perring A. Deja Vu: A User Study Using Images for Authentication. In: Proceedings of the 9th USENIX Security Symposium; 2000. .

[29] Szydlo M, Kaliski B. In: Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2005. p. 227–244.

[30] Bunnell J, Podd J, Henderson R, Napier R, Kennedy-Moffat J. Cognitive, associative and convensional passwords: Recall and guessing rates. Computers & Security. 1997;.

[31] Zviran M, Haga WJ. Cognitive Passwords: The Key to Easy Access Control. Computers & Security. 1990;.

[32] Zviran M, Haga WJ. A comparison of password techniques for multilevel authentication mechanisms. The Computer Journal. 1993;.

[33] Stinchcombe N, editor. Infosecurity Europe 2003 Information Security Survey; 2003.

[34] Takeda H, Veerkamp P, Tomiyama T, Yoshikawa H. Modelling Design Processes. AI Magazine. 1990;.

[35] Sommerville I. Software Engineering, 5th edition. Pearson Education; 2004.