A Feasible and Cost Effective Two-Factor Authentication for Online Transactions

Jing-Chiou Liou  
Department of Computer Science  
Kean University  
1000 Morris Ave.  
Union, NJ 07083, USA  
jlou@kean.edu

Sujith Bhashyam  
Department of Computer Science  
Kean University  
1000 Morris Ave.  
Union, NJ 07083, USA  
bhashyas@kean.edu

ABSTRACT
Authentication is the process of verifying a user’s identity when the user is requesting services from any secure IT system. By far, the most popular authentication is a basic username-password based method that is commonly considered to be a weak technique of authentication. A more secure method is the multi-factor authentication that verifies not only the username/password pair, but also requires a second or third unique physical or biological factor. However, the feasibility of multi-factor authentication is largely restricted by the deployment complexity and cost. In this paper, we propose a technique of two-factor authentication, called SofToken, that eases the deployment process and greatly reduces the cost, while maintaining the same level of security as achieved by current available techniques. A RFID application is also proposed to improve the portability of SofToken.

Keywords
Authentication, security, OTP, online transaction.

1. INTRODUCTION

For the last twenty years, most parts of the world have evolved into an electronic era. In modern society, the Information Technology (IT) and the Internet have played a significant role among people’s daily life. E-business, E-government, and even E-learning are reaching people everywhere wherever computers and the Internet connections are available.

In the E-era, all kinds of online applications not only provide convenience for people to access information and services anytime and anywhere, but also dramatically change the way how people live.

All the conveniences brought by the advanced IT come with a cost: IT security. To make sure the information and services do not fall into wrong hands, a user identity must be confirmed before any online transaction. To achieve the goal, two processes are used in any online transaction: Authentication and access control. The authentication process verifies the user identity as if he/she is a valid user to the IT system. Access control is applied after the authentication that gives the user some specific privileges in accessing certain components of the IT system.

We will discuss current available techniques of authentication in next section. In Section 3, sofToken for two-factor authentication will be presented. The new technique will be followed in section 4 by another technique utilizing RFID devices for different application in either two-factor or combining with the sofToken for multi-factor authentication. We then compare the sofToken technique with current available authentication techniques in Section 5. Finally, in Section 6, we conclude our discussion and project possible future study.

2. BACKGROUND

During the authentication process, some authentication factors are required for verification of user identity. An authentication factor is a piece of information that is supplied by the user and used to authenticate or verify the user identity that is requesting access under security constraints. The authentication factor is typically one of the three methods: “something you know” (e.g., username/password or PIN), “something you have” (e.g., smartcard or token), or “something you are” (e.g., fingerprint or iris scan) [1, 2].

2.1 Single-Factor Authentication

Single factor authentication (S-FA) relies on only one factor. The most common method in S-FA is a set of username/password or PIN. In security lexicon this is referred to as the what-you-know factor. Although still widely used by most merchants and financial institutions due to lack of a low cost alternative, authentications based solely on passwords have been considered to be a weak technique especially when used for valuable online transactions.

Security concerns for S-FA are obvious. Secure passwords are difficult to remember, not to mention that people have quite a few passwords to remember. For passwords that are easy to remember, they are conveniently suffered from various forms of software attacks. In a study by a data security firm [3] that analyzes 32 million passwords exposed in the Rockyou.com breach in December, 2009, it finds some of the most commonly used passwords. The top five most common passwords among those 32 million users are: 123456, 12345, 123456789, Password, and iloveyou.
Even using secure passwords, phishing and spoofing attacks may use a site that looks like a legitimate one to tricks the user into supplying the password. As a matter of fact, a news on October 8, 2009 reported that phishing scheme almost catches FBI Chief [4].

In addition, people usually don’t change their passwords frequently. It was reported, in some cases, that less than 25% of people change their password monthly and some 34% in a survey said they never change their passwords [5]. Hence, a keystroke logger can be installed physically [6] or in the form of a software on a computer to catch password entered manually on a login screen. As there are many passwords to remember, many people keep a file to book-keeping their passwords on their computer. Hackers who are able to reach that file can obtain the person's all username/password information.

One improvement in S-FA is to utilized password management utility. Password management is achieved by using various password valet applications, such as RoboForm [7] and KeePass [8], that store user passwords and can automatically enter the required fields in a web form. The software typically has a local database or files that hold the encrypted password data. Many password managers also work as a form filler, thus they fill the user and password data automatically into forms. Password managers have three types:

- Desktop - desktop software storing passwords on a computer hard drive.
- Portable - portable software storing passwords and program on a mobile device, such as a USB device or smart phone.
- Web based - Online password manager where passwords are stored on a provider's website.

Password managers typically use a user-selected master password or passphrase to form the key used to encrypt the protected passwords. However, this master password can also be attacked and discovered using key logging or acoustic cryptanalysis.

### 2.2 Two-Factor Authentication

Two-factor authentication (T-FA) is a process wherein two different factors are used in conjunction to authenticate. Using two factors as opposed to one factor generally achieves a higher level of authentication assurance. The FFIEC issued supplemental guidance on this subject in August 2006 [9], in which they clarified, "By definition true multifactor authentication requires the use of solutions from two or more of the three categories of factors. Using multiple solutions from the same category ... would not constitute multifactor authentication."

Generally, this second factor takes the form of a physical security token or smart card that the user has in his/her possession. This is referred to as the what-you-have factor. In this case, some application may also use mobile phone and other personal devices. One of the examples is the use of ATM card issued by any bank. One authentication factor is the physical ATM card that the customer slides into the machine. The second factor is the PIN the customer enters. Without both, authentication cannot take place.

Another application of the second factor may be a biological factor, such as a fingerprint scan. This is referred to as the what-you-are factor. Use of the what-you-are factor requires special hardware to scan the input data, thus has a higher complexity and cost in deployment.

To improve on security, the information in the what-you-have factor should be changed along the time. So the information is no longer valid when it is stolen and re-used. This is called One Time Password (OTP).

#### 2.2.1 Smart Card

Smart card [10] is a successor of magnetic card that is widely used in credit cards, debit cards, ATM cards and ID badges. In a magnetic card, unique characteristics called magnetic fingerprint is stored. Each swipe of the card provides a correlation number called a dynamic digital identifier that can be scored and “matched” to the originating value to determine the cards authenticity. Since the number changes each time (an OTP), it cannot be re-used as long as all processing is authenticated.

Smart cards are about the same size as a credit card and require special reader. The downside is that the smart card is not a small device and the card reader is an extra expense. It also needs special middleware application due to the mismatch between smart card communication standards [11] and the communication protocols [12] used by mainstream PC applications.

The deployment complexity and cost have limited its application within the government or enterprise environments that sometimes perform both the function of a proximity card and network authentication. Users can authenticate into the building via proximity detection and then insert the card into their PC to produce network logon credentials. They can also serve as ID badges.

#### 2.2.2 Biometrics

Users may biometrically authenticate via their fingerprint, voiceprint, or iris scan using provided hardware and then enter a PIN or password. For many biometric identifiers, the actual biometric information is rendered into string or mathematic information.

The device scans the physical characteristic, extracts critical information, and then stores the result as a string of data. Comparison is therefore made between two data strings, and if there is sufficient commonality a pass is achieved. It may be appreciated that choice of how much data to match, and to what degree of accuracy, governs the accuracy/speed ratio of the biometric device.

One problem is, when a large number of users are being authenticated at the same time, the technique may become unacceptably slow and comparatively expensive. It is also an easy target for reply attack. Once the biometric information is compromised (for example, fingerprint is being copied from something the user had held), it may easily be replayed unless
the reader is completely secure and guarded. Also spoofing is another way of security breach with this technique.

2.2.3 Security Token

Security tokens, also called OTP tokens, have an LCD screen that displays fixed number of alphanumeric characters. The OTP tokens are mainly based on two types of algorithms: time synchronized and event-based. Time synchronized algorithm produces pseudo-random number with a built in pseudo-random number generator. Pseudo-random number changes at pre-determined intervals, usually every 60 seconds. But they can also change at other time intervals. Event-based algorithm such as that proposed by the Open Authentication (OATH) consortium [13] uses a user event, such as the user pushing a button on the token.

On these time-synchronized OTP systems, time is an important part of the password algorithm since the generation of new passwords is based on the current time rather than, or in addition to, the previous password or a secret key. Some devices, such as RSA SecurityID [14] and VeriSign [15], display 6 digits pseudo-random number and require periodically resynchronize the server with the token.

To taking into account the portability, these security token must use materials that are small and consume less power. Still, these tokens need to be replaced every few years when the battery is dead. Cost effectiveness is not the only challenge for security token. Security token management is another key issue that prevents it from popular market acceptance. How to manage tokens issued by different service providers is a big concern for user and provider as well. In addition, once token is lost, the time and cost to replace is frustrating. Finally, the security tokens do not prevent Man-in-the-Middle based attacks against online transaction. Also a malicious user who has a complete control over the user's client computer (due to some kind of malware installed on the victim's client) could use the legitimate user's credentials for authorizing an illegitimate operation as explained in [16].

2.2.4 Virtual Token

Virtual tokens are a comparably new concept in multi-factor authentication first introduced in 2005 by a security company, Sestus [17]. Other virtual token techniques [18, 19] were gradually implemented. Virtual token enables any portable storage devices to work as an authenticate token, that a protected drive, Registry, and even in cell phones [20]. Howevr, distribution of virtual tokens cannot be controlled. This leads to a significant problem for using it to implement enterprise grade security [19].

2.2.5 Software Token

Some primitive types of software tokens are available on the market based on the same concept of hardware tokens. There are two primary architectures for software tokens: Shared secret and public-key cryptography. Shared secret architecture is considered more vulnerable than the hardware token. The configuration file can be compromised if it is stolen and the token is copied.

As an example, RSA SecurID software tokens [21] basically support the same algorithms as their RSA SecurID hardware authenticators. Therefore, like its hardware token, its software token produces either 6 or 8 digits number, called tokencode, and display next tokencode, every 30 or 60 seconds. For online transaction service, it requires, in addition to a web server, RSA Authentication Manager for token provisioning.

![Figure 1 RSA SecurID Software Token](image)

The generation of tokencode is not triggered by the server, but is on clien’t device(s). User enters locally the PIN to the installed application, and the client software generates the tokencode. The major concern with such time-based software tokens is that it is possible to borrow an individual's cell phone or laptop, to set the clock forward, and to generate tokencodes that will be valid in the future. In addition, anyone who provides the PIN correctly can retrieve the tokencode and use it for two-factor authentication on a web server from any cloned devices, such as an SIM card in a cell phone, or a USB installed with such application.

3. SOFTOKEN TECHNIQUE

Most two-factor authentication products require users to deploy client software to make T-FA systems work. Some vendors even have created separate installation packages for network login, Web access credentials and VPN connection credentials. For such products, there may be four or five different software packages to be installed on to the client PC in order to utilize such security token or smart card.

This means to four or five packages on which version control has to be performed and four or five packages to check for conflicts with business applications. If access can be operated using web pages, it is possible to limit the overheads outlined above to a single application. With other T-FA solutions, such as virtual tokens and some hardware token products, no software must be installed onto end user’s computer, but a security file must be created on a portable devices.

Nonetheless, something, either software or a file, needs to be installed for such product to work. This initiates a concept of SofToken. A SofToken is a client software installed on a user’s
computer that generates a time synchronized OTP. SofToken does not require any hardware to store software applications or files.

3.1 SofToken Process

A SofToken process commences when a user requests an account for online transaction from a service provider, such as an online banking, an e-business site, or an enterprise intranet, etc. When the user successfully establish the user account through online access by providing sufficient information to the service provider, the server delivers a client software to the user’s computer. This client software installs two components onto user’s computer with user’s consensus: A logon application and a pseudo-random number generator.

During the initialization process, an encrypted public key will be created and issued to the user’s computer as the seed of pseudo-random number generation. The key can be produced based on either a user’s favored challenge-response or by the server. This encrypted key will be stored at the user’s computer as part of the pseudo-random number generator.

The logon application is responsible to set up a direct communication between the server and user’s computer without going through a web-based logon screen. The communication can be encrypted so sufficient degree of security can be maintained. The user supplies the username/password as the first factor to the server.

Once the server successfully verify the first factor, the server sends a request to the pseudo-random number generator installed on the user’s computer to trigger the generation of a pseudo-random number, called codeword. Along with the request, a new encrypted public key will be sent to the user’s computer as the next seed of pseudo-random number generation. With OPT binding process (challenge/response), the request will not trigger any client software that does not has the correct seed. So obtaining the username/password will not generate the OTP codeword from any other devices.

The user is now able to enter the codeword as the second factor for authentication. The codeword will be verified again by the server. If it is entered incorrectly, the user has to start over from the first factor in order to generate a new codeword.

3.2 OTP Generation and Binding

Since the codeword is an OPT, it changes every time it is produced. In addition, the codeword is triggered by the server’s request, it does not hold for 60 seconds as some security tokens do. If the user does not enter the codeword immediately, say within 10 seconds, the codeword expires. And the user has to use the first factor to re-initiate the codeword. So any keystroke loggers will not be able to log the codeword for re-use.

Moreover, since the codeword is triggered by the request of the server, supplying the first factor to the server will only generate the codeword from the computer that has the pseudo-random number generator installed. That also means the computer is registered to the server. After every successful authentication, a new encrypted public key will be sent to the user’s computer as the next seed of pseudo-random number generation Therefore,
another user of the service provider will not be able to hijack other user’s username and password to gain access to the server because the codeword generated would be different from different computers.

In some cases, a computer may host two or more such user accounts and be installed with multiple instances of the authentication software. For example, multiple family members use a home computer for the same online transaction service provider with different user accounts. In this particular case, each family member has its own user authentication software installed and uses his/her own logon application.

Another case is that a user may request to have the authentication software installed onto multiple computers, for example, on a desktop and a laptop. This is feasible because the user can, while installing the software onto a computer, register the computer to the server. In this case, two instances of authentication software are installed. Each authentication software will produce its codeword (pseudo-random number) independently. Therefore, there is no need to synchronize the authentication software on both computers.

4. COMPARISON OF CURRENT AUTHENTICATION TECHNIQUES

Each technique used for two-factor authentication today addresses certain security issues while bringing in some feasibility issues and other security concerns. We performed comparison of most current available two-factor authentication techniques to show their strengths and weaknesses. In such comparison, we compare the single-factor and 4 popular two-factor techniques, along with SofToken, using 9 measures for feasibility and online transaction security. There are 6 feasibility measures and 3 security measures. Those best in the class is highlighted in bold and italic font. Detailed discussion of these feasibility and security measures can be found in [22].

5. CONCLUSION

In this paper, we proposed the SofToken, a two-factor authentication that is feasible and cost effective for any online transaction services. The SofToken is strong in online security, while maintaining a simple deployment process. We compared the SofToekn with the single-factor authentication and other available two-factor authentication techniques. The comparison proves that the SofToken technique outperforms than all others in almost every measure. Only in the portability measure it yields to the non-OTP fashioned authentication techniques. But it still performs as well as other OTP type two-factor techniques.

6. REFERENCES


