Internet Banking Login with Multi-Factor Authentication

Sirapat Boonkrong
Faculty of Information Technology,
King Mongkut's University of Technology North Bangkok
Bangkok, 10800 - Thailand
[e-mail: sirapat.b@it.kmutnb.ac.th]

Received May 12, 2016; revised September 6, 2016; revised September 11, 2016; accepted November 15, 2016; published January 31, 2017

Abstract

Internet banking is one of many services provided by financial institutions that have become very popular with an increasing trend. Due to the increased amount of usage of the service, Internet banking has become a target from adversaries. One of the points that are at risk of an attack is the login process. Therefore, it is necessary to have a security mechanism that can reduce this risk. This research designs and develops a multi-factor authentication protocol, starting from a registration system, which generates authentication factors, to an actual authentication mechanism. These factors can be categorised into two groups: short term and long term. For the authentication protocol, only three messages need to be exchanged between a client and a financial institution's server. Many cryptographic processes are incorporated into the protocol, such as symmetric and asymmetric cryptography, a symmetric key generation process, a method for generating and verifying digital signatures. All of the authentication messages have been proved and analysed by the logic of GNY and the criteria of OWASP-AT-009. Even though there are additional factors of authentication, users do not really feel any extra load on their part, as shown by the satisfactory survey.

Keywords: Multi-Factor Authentication, Internet Banking, Security

A preliminary version of this paper appeared in the 7th Workshop 'Autonomous System', 26th – 30th June 2014, in Mallorca, Spain. This version includes a detailed analysis, implementation results and user satisfactory survey on the protocol. This research was supported by a research grant from the National Research Council of Thailand (NRCT) [Cyber Security Research Grant, 2013].

https://doi.org/10.3837/tiis.2017.01.027 **ISSN : 1976-7277**

1. Introduction

Internet banking has been around as one of many services provided by financial institutions for some time now. The service is deemed as one of the most popular since it provides many conveniences to users. With this availability, users are allowed to perform many financial related tasks such as looking at their personal information, checking their account balance, making payments and transferring money.

Unfortunately, Internet banking as well as any financial transactions have been under many forms of attacks [1]. They include phishing, social engineering, espionage and, more importantly, impersonation. All of these attacks have affected the confidence of users in using these online services. It is, therefore, the job of the financial instutions and academic researchers to find a way to improve the security of transactions on the Internet.

Many researchers have studied various aspects of Internet banking security, including threats and vulnerabilities [1] [2] [3]. However, there have not been many that have focused direcly on the first line of defence mechanism or authentication to such service. Some have only worked on authentication in a broad sense [4] without emphasising on Internet banking, while some have concentrated on the security of transaction on mobile applications [5] [6]. It is, therefore, necessary to put the emphasis of this research on a security mechanism that provides the first protection to Internet banking, namely authentication process.

From our preliminary survey of Internet banking provided by six major banks in Thailand, the followings can be learned. Firstly, five out of six had an authentication mechanism that required users to enter their username and password. This is known as *password-only authentication*. The other bank had an extra process of asking users to enter CAPTCHA. However, in theory and practice, CAPTCHA is not really an authentication mechanism. Instead it is known to be an authorisation mechanism.

Moreover, five out of six banks that were studied let users choose their own user ID while the other one selected one for the users. At the same time, all six banks allowed users to choose their own password albeit with basic instruction on how to choose a strong one. In addition, when loggin into the service, five out of six banks required users to enter their password using their own keyboard. There was one bank, however, that provided a randomly generated keypad on the screen for the users to enter their password.

The results of the preliminary survey of authentication mechanisms on Internet banking (in Web banking format) provided by banks in Thailand are shown in **Table 1**. Note that all of the Web sites were accessed and studied in January 2014.

Table 1. Authentication Mechanisms Provided by Banks in Thailand

Banks	Authentication			Choosing User ID		Other Mechanisms
	User ID	Password	Others	Self	Bank	
BKK	✓	✓			✓	
KK	✓	✓		✓		
KTB	✓	✓	CAPTCHA	✓		
SCB	✓	✓		✓		
TMB	✓	✓		✓		
TNC	✓	✓		✓		Random Keypad

It can be seen from our study that most, if not all, of the banks seemed to adopt *one-factor authentication* in using a password only when logging into the system. This has led to problems

stated in [1] [2] [3]. Even though all of the banks used SSL (Secure Socket Layer) as an extra protection mechanism, this technology is still vulnerable to an SSL stripping attack [7].

Furthermore, it was to our surprise that while other transactions such as making payments and money trasfers required another factor of authentication from a *one-time password* or *OTP* sent from the bank to users' mobile phone in addition to users' personal password, no banks seemed to care to adopt the same strategy to the login process.

From this, it can be said that people have appeared to forget how important the login process actually is although a lot of personal and financial information can be seen as soon as users have been authenticated. This information includes account name, account number, account balance and transaction history. Using this information, it is not difficult for an adversary to carry out an attack known as impersonation.

In fact, 2014 and 2015's security reports [8] [9] indicated that online banking was one of the most vulnerable category to attacks, accounting for almost six million attempts to compromise the system and user's identity. In addition, the Bank of England simulated a cyber attack on British financial systems and found that many of the UK's largest institutions were unprepared for large-scale identity-based attacks [10].

The objective of this research is, therefore, to design and implement an authentication mechanism that can reduce the risk of an attack on the login process on Internet banking, with the focus on the service provided via the Web.

The paper is organised as follows. Section 2 gives an overview of existing authentication mechanisms used for Internet banking. Section 3 provides the design of the proposed authentication protocol as well as a proof of security using the logic of GNY. The results of the implementation will be explained in Section 4. Section 5 discusses various aspects of the proposed protocol, including the performance issues and security issues according to the OWASP-AT-009 criteria as well as security against known attacks. Section 6 then concludes the paper.

2. Related Work

2.1 Internet Banking Threats

Since the beginning of the Internet banking services, there have been attempts to compromise them. This section, therefore, provides the description of the existing threats to Internet banking.

Threats to Internet banking, specifically to one-factor authentication, are described in [11] and [12]. Generally, they can be categorised into three groups: credential stealing, channel breaking and content manipulation.

Credential stealing is when an adversay attempts to steal user's personal information such as username and password via the use of malware or phishing.

Channel breaking is when an adversary attempts to intercept information being transmitted between user and the bank's server. This is done by impersonating in such a way that the user thinks he or she is really communicating with the bank, and vice versa. This is very similar to an attack known as a man-in-the-middle attack, which can be solved by the use of transaction authentication [13].

Content manipulation or, more commonly, man-in-the-browser is when a user's Web browser is infected by malware in such a way that it is possible for an attacker to read, write, edit and delete the information on that Web browser without the knowledge of the user. This

type of attack is very popular since it is able to bypass two-factor authentication methods. The detail of man-in-the-browser attack can be studied in [11] and [12].

2.2 Existing Authentication Mechanisms

Many of today's financial institutions have adopted a mechanism known as password-only authentication. That is, users only have to enter their username and password to gain access to their Internet banking account. As stated earlier, many are using it with the help of SSL as an extra protection, but the authentication process is still not secure [7]. This is because the password-only authentication is still vulnerable to the use of key logger and, more so, phishing attack [14] [11]. Moreover, threats to one-factor authentication have been pointed out and explained in Section 2.1 that they can be divided into three main categories, which are credential stealing, channel breaking and content manipulation. Therefore, it appears that a one-factor authentication is not enough to secure the authentication during the login process.

With the threats to one-factor authentication, many banks have gone for two-factor authentication. A two-factor authentication mechanism [2] [15] requires users to still enter their personal password, but they will be provided with a new temporary password which can either be generated by another device called a password generator or sent to users via SMS. Even though this method of authentication is claimed to be more secure than the one-factor authentication, the disadvantage of it is the need for an extra device during authentication. Moreover, sending a temporary password (known as a one-time password) via SMS [16] is also considered insecure these days due to the widespread of malware installed on mobile devices.

Two-factor authentication does not have to consist of typing a temporary password received from the bank into a computer. Several researchers [14] [17] were against the idea of typing a password on a computer by saying that it was like giving a password to a computer that might actually be public. They, therefore, had suggested another variation of two-factor authentication by proposing that the user's mobile device or smart phone should be the second factor of authentication. In other words, instead of entering the password on the computer, the password would be entered on the user's own device. The password was to be encrypted on the mobile device before being transmitted either via wi-fi or bluetooth to the computer. Using this method, no plaintext password is to be entered on a computer and only the encrypted version will be seen by the computer. However, there are problems with the method. First of all, we can see the inconvenience of having to have a smart phone and an extra application. Secondly, the communication channel used by the smart phone and the computer must be secure in some way, hence extra work is needed. Thirdly, if the smart phone is not malware-free then an adversary can still have access to the entered plaintext password [13].

Another interesting two-factor authentication scheme was presented by *Wang et al* in [18]. The authors proposed an efficient and secure two-factor authentication scheme with user anonylity. Similar to our proposed work, [18] also consisted of two phases, which were the registration and authentication phases. However, having a smart card as a second factor of authentication does not appear attractive to Internet banking authentication because of the additional piece of hardware required.

Other variations of two-factor authentication are also available. A two-factor authentication using a bookmark on a Web browser as the second factor was proposed in [19]. Some have used a password together with graphics or pictures [20] [21] [22] to

increase security. Another popular second factor of authentication is using the "Something You Are" or biometric as a part of authentication process [23] [24] [25].

Even though two-factor authentication has received a lot of support of interests, we do not think that it provides enough security for authentication purposes, especially during the login. This is supported by [13] who said that two-factor authentication was useful because a password was changed each time we would like to enter the system. However, this mechanism was not the life saver, because attacks in the forms of phishing and identity theft were still possible. [13] also suggested that it would be better to authenticate the transactions rather than the entities.

Two-factor authentication mechanisms have not been the only method studied by resesearchers, some have also worked on multi-factor authentication [4] [26]. However, their work did not directly focus on Internet banking authentication.

Stebilia *et al* [4] proposed that an authentication mechanism should include four factors. They were a long-term password, a one-time password, cryptographic key and biometric. Users would have to illustrate to the server that they knew or possess all the four factors. The author claimed that the scheme was more secure because an adversary would have to compromise all four factors before gaining access to the system. Similarly, Huang *et al* [27] provided a solution for an authentication protocol in a large-scale system and fragile communications. The factors of authentication included a password, biometric as well as a smart card.

Even though their work was not proposed to work with Internet banking, it seems possible that the scheme could be adapted to function in such environment. However, there would still be a problem with the required additional piece of biometric and smart card equipments which would bring burden to users.

It can be seen that many researchers have worked on improving the security of authentication mechanisms. They have applied the idea of having extra factors as parts of the process. However, the authentication process still needs to be designed and developed so that it is more secure and more suitable to the Internet banking login process. This is the objective of this paper.

In addition, although two-factor authentication provides better security than the one-factor authentication methods, it is still not enough to prevent such attacks as cresential stealing, impersonation and man-in-the-browser. Therefore, it is necessary to come up with an authentication mechanism that is more difficult to attack. One of the ways to achieve this is to add extra factors of authentication to obtain what we call multi-factor authentication, which will be discussed in the next section.

3. The Design

This section begins with the two main requirements of our proposed authentication protocol for the Internet banking login process. The first is mutual authentication. The second is transaction authentication.

First of all, mutual authentication or two-way authentication is a must in the authentication process for Internet banking. This is because it is essential for a user to know that the entity he or she is communicating with is really the bank. At the same time, the bank also needs to be sure that the entity at the other end is really a legitimate user.

Secondly, transaction authentication is another aim of our proposed authentication protocol. This is also supported by [13]. Transaction authentication is used for identifying a user at a transaction level, rather than at a session level. That is, every message being transmitted is to be verified by the other party to ensure that it is really sent by the expected party or known entity.

In addition to the two main aims, basic security properties cannot be forgotten. Firstly, a lot of the information being transferred during the login process is usually credential information that needs to be protected, so confidentiality is necessary. Secondly, it is important to make sure that the transmitted messages are not modified in any ways. Therefore, the integrity of the message is another essential property that needs to be considered.

3.1 Registration

It needs to be understood here that an authentication or a login process must begin with registration. That is, an Internet banking user needs to register with a bank before being able to use the service. That means our first task is to design a registration process for Internet banking.

The registration system has been designed and proposed in such a way that the information needed to be entered is not different from what a typical bank requires. What most banks usually do at this stage is to store information and user's credentials, which include username and password. However, we propose that other factors of authentication for each user need to be generated and saved during registration, too.

In order to achieve that, an extra field is to be added to the proposed registration process. The added field is to be called the "Iterations" field. This field requires that a user enters any whole number bigger than one. The value will have to be remembered by the user. This number is a necessary component in a key generation process that will be explained later. It should be acknowledged that this field is not much different from a "safety question" field. However, the reason that the iterations field has been proposed in this research is because it is believed that in all sexes and ages, the memory-span for digits is more superior than the memory-span for letters [28].

Once the information has been entered, it will be stored on the bank's server. In addition, we think that it is necessary to generate some more credentials for each user during the registration. The process of generating these extra credentials or authentication factors will be run in the background of the registration stage, so that it is seamless to the user. Note that each registered user will have a completely different set of authentication factors after registering to use the system.

3.1.1 Authentication factors

The first authentication factor to be generated is, of course, the user's chosen password. It is recommended that users follow Ma *et al*'s criteria [29] for choosing their password. Ma *et al* stated that a password should be at least eight characters long and consist of some special characters and numbers.

For password storage, [30] describes many methods for storing a password. However, it is suggested that a salted hash method should be applied to enhance the security. This implies that a second factor of authentication is to be generated here. This factor is a salt value. A *salt* is a value that is randomly generated by the system and can be used as a part of the password storing method. In other words, instead of storing a password by just hashing it, a

salt value is concatenated to the password before hashing using a cryptographic hash function such as MD5, hence the name salted hash.

Next, the third and fourth factors of authentication are to be generated. During this registration process, a pair of *public key* and *private key* for this particular user will be computed. As the name suggests, the public key can be seen by anyone. However, the private key needs to be kept secret and can only be known by the user. One possible way to achieve this is to encrypt it using symmetric cryptography so that only the user can access it. This implies that a *symmetric key* needs to be produced. This will be the fifth authentication factor.

In order to generate a symmetric key, it is proposed here that a key derivation function (KDF) is to be applied. For this particular purpose, SHA-256 [31] can be used with two inputs, which include the salted password (*Salt*||*Password*) and the number of iterations. Both are also factors of authentication computed and entered earlier in the registration process. In other words, a symmetric key can be derived as *DK = SHA-256(Salt*||*Password; iterations)*, where *DK* is the derived key and iterations denotes the number of rounds the KDF or SHA-256 has to be computed. It should be noted that the iterations value serves three purposes. Firstly, it provides the number of rounds or iterations that has to be computed when attempting to derive a symmetric key. Secondly, by having a different number of computing rounds for each user rather than one round of SHA-256 for all users, it becomes more difficult for an attacker to compromise the symmetric key. Thirdly, the value can help reduce the risk of two or more users producing the same symmetric key.

It needs to be understood here that a different symmetric key will be computed for every user. This is because each user will choose a different password, a different salt value will be generated differently and each user will be likely to choose a different value of iterations, too. Even if two users happen choose the same password and the number of iterations, the values of the salt will still be different as they are randomly chosen. Two different symmetric keys will be obtained as a result.

After the symmetric key has been produced, the private key that has been generated earlier will be encrypted using the AES-256 algorithm. This is the reason that SHA-256 is used as a symmetric key derivation function. Once the private key encryption is finished, we are ready to transfer information or factors of authentication to the bank. This information to be transmitted consists of (1) username, (2) a salt value, (3) the hash value of the salted password or MD5(*Salt*||*Password*), (4) a public key and (5) an encrypted private key. Everything can also be encrypted by using the bank's public key to enhance the security of the message, which is constructed as follows.

$$user \rightarrow bank: \{userID, salt, MD5(Salt||Password), +K_{user}, \{-K_{user}\}_K\}_{+K_{bank}}$$

where $+K_x$ means the public key of *x*,

$-K_x$ means the private key of *x*,

$\{M\}_{+K_x}$ means the encryption of message *M* using the public key of user *x*, and

$\{M\}_K$ means the encryption of message *M* using the symmetric key *K*.

It needs to be clarified here that even though all five pieces of information are transmitted to and stored at the bank, it is not possible for the bank to compute or find the secret credentials of each user, namely the user's password and private key. This is because the user's salted password has already been hashed by a one-way cryptographic hash function, MD5. In addition, the user's private key has already been encrypted using the AES-256 with a symmetric key that has been produced from the user's password and the number of iterations, both of which are never transmitted anywhere.

After the bank has received the message, all five pieces of information will be stored on its server. This brings us to the end of the registration process. The registration process is depicted in **Fig. 1**.

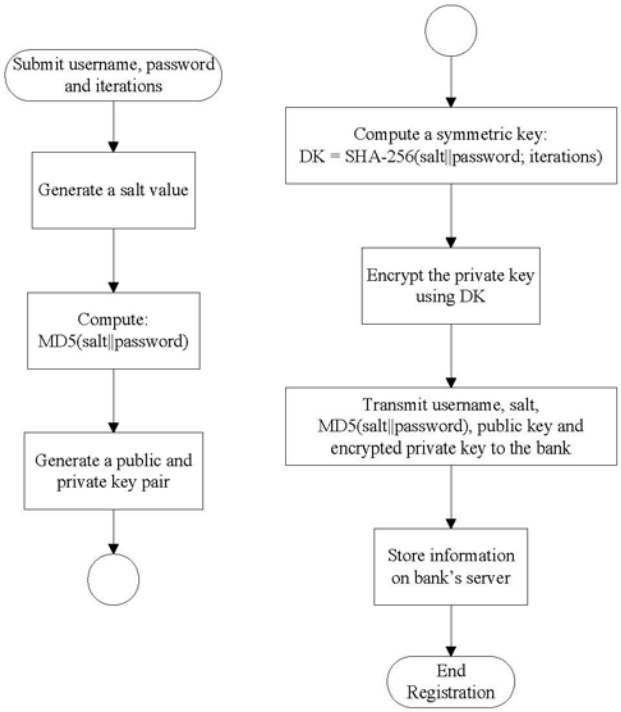

Fig. 1. Proposed Registration Process

3.2 Authentication

When a user would like to use an Internet banking service, an authentication or a login process will have to be carried out. As stated earlier, the main aims of the authentication process include mutual authentication and transaction authentication. In addition, three other security characteristics will also be considered. They are confidentiality, integrity and non-repudiation. It will be shown in this paper that the proposed authentication protocol can achieve these objectives.

Before going into detail of the authentication design, it has to be assumed that a bank holds a valid digital certificate and the client computer has the ability to verify it. This is not overly stated in any way, because banks usually hold a certificate and any today's Web browser on the user side has the ability to carry out the certificate verification anyway.

The proposed authentication protocol consists of two parts. The first is for a user to fetch his or her own encrypted private key from the bank's server. This private key will be a very important component for the second part of the protocol, the mutual authentication. At the end of the protocol, users will be assured that he or she is really communicating with the bank, and vice versa.

In order to have the same understanding in the proposed authentication protocol, the process prior to the actual authentication is as follows. When a user visits a bank's Web site, the bank's server sends its digital certificate to the user. At the same time, the server records the IP address of the client's machine used in this particular session. On the client's side, the

client machine verifies the bank's certificate. Assuming that the bank's certificate is valid and the verification process is a success, the proposed authentication protocol consists of the following steps.

1. On the client's machine, the user fills in the authentication form via the bank's Web site. The form consists of the username, password and number of iterations fields. The client machine will only sends the username to the bank's server at this step.

2. The bank's server, having received the username, searches for two pieces of information which belongs to this particular user. They are the salt value and the encrypted private key. Note that this information has been previously generated and stored during the registration process. The server then sends both pieces of information to the user together with its digital signature.

3. The client's machine carries out the verification process on the bank's signature. If successful, the client holds the following information. First of all, the client's machine holds the user's password and number of iterations, both of which were entered in Step 1. Secondly, the client holds the salt value, which has just been received from the bank. These three pieces information are enough to generate a symmetric key that will used to decrypt the received encrypted private key.

4. The client's machine generates a symmetric key using the method proposed in Section 3.1.1. That is, the client uses *DK = SHA-256(Salt||Password; iterations)* to derive the key. The derived key is then used to decrypt the encrypted private key received from the previous step. At the end of this step, the user holds his or her own private key.

5. The client's machine computes the hash value of the salted password before encrypting it using the bank's public key, which was in the bank's certificate received earlier. The user then generates his or her digital signature using the private key obtained in the previous step. The user appends the signature to the encrypted hash value. The message is sent to the bank's server.

6. The bank's server verifies the signature on the received message using the user's public key that has already been stored during the registration process. If successful, the server decrypts the message and verifies the hash value of the salted password. If the value matches the one stored on the server then the user will be allowed to enter and begin the Internet banking application.

From the proposed design, it can be seen that mutual authentication is achieved. In other words, the client's machine knows that the other entity is the bank's server by the verification of the bank's signature. The bank also knows that the other party is the authorised user by the verification of the user's signature as well as the hash value of the password.

The designed authentication protocol can be written in an idealised form as follows.

Message 1.	$User \rightarrow Bank$:	$Req, UserID, \{UserID, Nonce_{User}\}_{+K_{Bank}}$
Message 2.	$Bank \rightarrow User$:	$\{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt,$ $\{SHA1(\{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}},$ $\{-K_{User}\}_K, Salt, IP_{Bank})\}_{-K_{Bank}}$
Message 3.	$User \rightarrow Bank$:	$\{UserID, Nonce_{Bank}, MD5(Salt\|\|Password)\}_{+K_{Bank}},$ $\{SHA1(\{UserID, Nonce_{Bank}, MD5(Salt\|\|Password)\}_{+K_{Bank}},$ $IP_{User})\}_{-K_{User}}$

where $Nonce_x$ means a set of characters randomly generated by entity x,

$+K_x$ means a public key of entity x,
$-K_x$ means a private key of entity x,
$\{M\}_K$ means encryption of message M using a symmetric key K,
$\{M\}_{+K_x}$ means encryption of message M using a public key of entity x,
$\{M\}_{-K_x}$ means a digital signature on message M by entity x,
$SHA1(\ )$ means SHA-1 cryptographic hash function,
$MD5(\ )$ means MD5 cryptographic hash function, and
IP_x means an IP address of entity x.

4. Protocol Analysis

Having designed the authentication protocol for Internet banking login process in the previous section, it is necessary to provide a formal proof and analysis to make sure that it is secure and correct. In this section, the logic of GNY [32] is used in order to show that the main objectives of mutual authentication and transaction authentication are achieved at the end of the protocol.

It is assumed that the symmetric encryption scheme used here, specifically AES-256, is secure against chosen-plaintext attack. It is also assumed that the RSA encryption and signature schemes are secure against chosen-ciphertext and factoring attacks. Moreover, the security of the cryptographic hash functions in SHA-1, SHA-256 and MD5 is also assumed.

The analysis begins with an additional set of assumptions, some of which are actually obtained during the registration process and the visit to the bank's Web site.

$Bank \ni Cer_{Bank}$	$Bank \ni +K_{Bank}$	$Bank \ni -K_{Bank}$	$Bank \ni BankID$
$Bank \ni +K_{User}$	$Bank \ni Salt$	$Bank \ni \{-K_{User}\}_K$	$Bank \mid \equiv \#Nonce_{Bank}$
$User \ni +K_{Bank}$	$User \ni Cer_{Bank}$	$User \ni Password$	$User \ni UserID$
$User \ni iterations$	$User \mid \equiv \#Nonce_{User}$		

where $\ni$ means to possess,
$\mid \equiv$ means to believe, and
$\#M$ means the component M is fresh.

The proposed authentication protocol consists of three messages, which can be written in the GNY format as follows.

$$Message\ 1.\quad Bank \lhd *Req, *UserID, *\{*UserID, Nonce_{User}\}_{+K_{Bank}}$$

$$Message\ 2.\quad User \lhd *\{*BankID, Nonce_{User}, *Nonce_{Bank}\}_{+K_{User}}, *\{*-K_{User}\}_K, *Salt, *\{*SHA1(*\{*BankID, Nonce_{User}, *Nonce_{Bank}\}_{+K_{User}}, *\{*-K_{User}\}_K, *Salt, IP_{Bank})\}_{-K_{Bank}}$$

$$Message\ 3.\quad Bank \lhd *\{UserID, Nonce_{Bank}, *MD5(Salt||Password)\}_{+K_{Bank}}, *\{*SHA1(*\{UserID, Nonce_{Bank}, *MD5(Salt||Password)\}_{+K_{Bank}}, IP_{User})\}_{-K_{User}}$$

where $\lhd$ means to hear or to receive, and
$*M$ means the component M is not originated at the receiving entity.

The analysis and proof of security and correctness using the logic of GNY can now begin. The GNY postulates consist of six categories. The first category of GNY postulates is called the *Being-Told Rules*, denoted as *Ti* where *i* is the rule number. The second category is known as the *Possession Rules*, denoted as *Pi* where *i* is the rule number. The third set of postulates is the *Freshness Rules*, denoted as *Fi* where *i* is the rule number. The fourth contains the set of *Recognisability Rules*, denoted as *Ri* where *i* is the rule number. The fifth category is called the *Message Interpretation Rules*, denoted as *Ii* where *i* is the rule number. The sixth group of postulates is the *Jurisdiction Rules*, denoted as *Ji* where *i* is the rule number. We provide a summary of the definition of each of the GNY postulates in **Table 2**, although the detail of the all the GNY postulates can be seen in [32].

Table 2. Definitions of the GNY Postulates

GNY Postulate Notation	**GNY Postulate's Name**	**Definition**
Ti	Being-Told	This set of rules deal with components as well as manipulations of those components a protocol end entity receives. They are regarded as being-told to the entity.
Pi	Possession	This set of rules deal with the fact that the protocol end entity is capable of possessing any components or manipulations of those components he or she has already possessed as well as those that he or she has been told.
Fi	Freshness	The freshness rules ensures that the protocol end entity has at least one component that he or she believes to be fresh, i.e., the same value has never been used before. This set of rules can ensure that replay attacks do not occur.
Ri	Recognisability	The recognisability rules specify that the protocol end entity receives at least one component that he or she believes to be recognisable. Challenge-and-response mechanism falls into this set of rules.
Ii	Message Interpretation	This set of rules enable the protocol end entity to enhance their beliefs on the component he or she receives by examining the components or messages he or she has received previously.
Ji	Jurisdiction	The jurisdiction rules enable the protocol end entity to specify who has conveyed the received components and messages. The decision made is based on the previously received components and messages.

The analysis of the proposed multi-factor authentication protocol is provided below.

Message 1: Applying the GNY postulates T1 and T4, we obtain $Bank \lhd Req, UserID, Nonce_{User}$. That is, *Bank* has received or has been told the components $Req, UserID$ and $Nonce_{User}$.

Applying the postulate P1, we obtain $Bank \ni Req, UserID, Nonce_{User}$. That is, *Bank* now possesses the components $Req, UserID$ and $Nonce_{User}$.

Applying the postulate F1, because the $Nonce_{User}$ is fresh, we obtain $Bank| \equiv \#(Req, UserID, Nonce_{User})$ which means that *Bank* believes that the received message is fresh, i.e., not a replay.

At this stage, the bank looks up the $UserID$ in the database to find the components associated to this particular user. Therefore, the postulate R1 can be applied.

Applying the postulate R1, because the $UserID$ is recognised by the bank, we obtain $Bank| \equiv \phi(Req, UserID, Nonce_{User})$.

The second message is then constructed and transferred to the user.

Message 2: Applying the postulates T1 and T6, we obtain $User \lhd \{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt, SHA1(\{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt, IP_{Bank})$. The postulate T6 is actually the signature verification and message integrity checking stage.

The user can verify the bank's digital signature because he or she possesses the bank's certificate and public key, which has been received when contacting the Web site before logging in. That means a normal signature verification process can proceed.

Applying the postulate P1, we obtain $User \ni \{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt$. That is, the user now possesses the components $\{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}$ and $Salt$.

Furthermore, at this stage the user has already entered his or her password and number of iterations at the login screen. That is, $User \ni Password$ and $User \ni Iterations$. The user has also received and possessed the salt value from the bank. This means that a symmetric key *K* can now be computed, using *SHA-256(Salt||Password; Iterations)*. Hence, $User \ni K$ and the component $\{-K_{User}\}_K$ can now be decrypted.

Applying the postulate P6, we obtain $User \ni \{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, -K_{User}, Salt$. That is, the user has decrypted the component $\{-K_{User}\}_K$ using the symmetric K and now possesses his or her own private key $-K_{User}$, or $User \ni -K_{User}$.

Applying the postulates T4 and P1, meaning that the components $\{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}$ have been decrypted using $-K_{User}$, and we obtain $User \ni BankID, Nonce_{User}, Nonce_{Bank}$. This means that the components $BankID, Nonce_{User}$ and $Nonce_{Bank}$ are now possessed by the user.

Applying the postulate F1 or the freshness rule, we obtain that the user believes that the received message has been freshly constructed. This is true because the component $Nonce_{User}$ had just been generated by the user in the first message and is now sent back to him or her in this second message. Thus, this message must be fresh.

Applying the postulate R1 or the recognisability rule, we obtain that the user believes that the received message is recognisable. This is true because the component $Nonce_{user}$ can be recognised by the user. In other words, the component $Nonce_{User}$ was generated by the user and sent to the bank in Message 1. The same component has now been returned back to the user.

Applying the postulates I4 and I6 known as the message interpretation rules, we obtain $User| \equiv Bank \ni \{BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt, \{SHA1(BankID, Nonce_{User}, Nonce_{Bank}\}_{+K_{User}}, \{-K_{User}\}_K, Salt, IP_{Bank})\}_{-K_{Bank}}$. This means that the user believes that the

bank also possesses all the components that were received by the user. That is, the user believes that the bank is the one who has really sent the message. The analysis of Message 2 ends here. This is because the analysis shows that several objectives have been achieved. First of all, it can be seen that, in Message 2, the user is able to compute a secret symmetric key K. Secondly, the user can use the obtained secret key to decrypt $\{-K_{User}\}_K$ which means that the user now possesses his or her own private key, $-K_{User}$. Thirdly, the analysis shows that the user believes that the message he or she received is really from the bank, and not someone else.

Message 3: Applying the GNY postulates T1 and T6, we obtain $Bank \lhd \{UserID, Nonce_{Bank}, MD5(Salt||Password))_{+K_{Bank}}, SHA1(\{UserID, Nonce_{Bank}, MD5(Salt||Password)\}_{+K_{Bank}}, IP_{User}$. Again, as done in Message 2, the postulate T6 is the signature verification and message integrity check process. If the signature verification and message integrity check are a success, the bank will decrypt the ciphertext part $\{UserID, Nonce_{Bank,} MD5(Salt||Password)\}_{+K_{Bank}}$ using its private key.

Applying the postulates T4 and P1, the decryption process on the ciphertext takes place here. The statement $Bank \ni UserID, Nonce_{Bank}, MD5(Salt||Password)$ is obtained. This means that the bank now possesses the components $UserID, Nonce_{Bank}$ and $MD5(Salt||Password)$. This is where the hash value of the salted password is then compared with the value stored in the bank's database in order to verify the entered password.

Applying the postulates F1 and R1, we learn that the bank believes that the received message is fresh. This is true because the component NonceBank had just been generated by the bank in the previous message and has now been sent back to it in this message. Moreover, the bank believes that the received message is recognisable, which is true because the bank can recognise its own $Nonce_{Bank}$.

Applying the postulates I4 and I6, we learn the fact that $Bank| \equiv User \ni \{UserID, Nonce_{Bank}, MD5(Salt||Password)\}_{+K_{Bank}}, \{SHA1(\{UserID, Nonce_{Bank}, MD5(Salt||Password)\}_{+K_{Bank}}, IP_{User})\}_{-K_{User}}$. This means that the bank truly believes that the whole message was really owned, constructed and sent by the user.

The analysis of Message 3 ends here. This is because the analysis shows that the bank believes that the message really comes from the user, and the digital signature was really generated by the user. Hence, the other entity is the legitimate user.

On the whole, the analysis and proof of correctness and security of the proposed protocol shows that the mutual authentication objective has been achieved. In other words, the user believes that he or she is really communicating with the bank. The bank also believes that it is communicating with the legitimate user.

Moreover, the second and third messages contain the bank's and user's digital signature respectively. The analysis on both messages show that both digital signature verification and message integrity check processes are carried out by both the bank and user. Hence, the proposed protocol achieves the second objective in transaction authentication as well.

The resultant protocol can be seen in **Fig. 2** as follows.

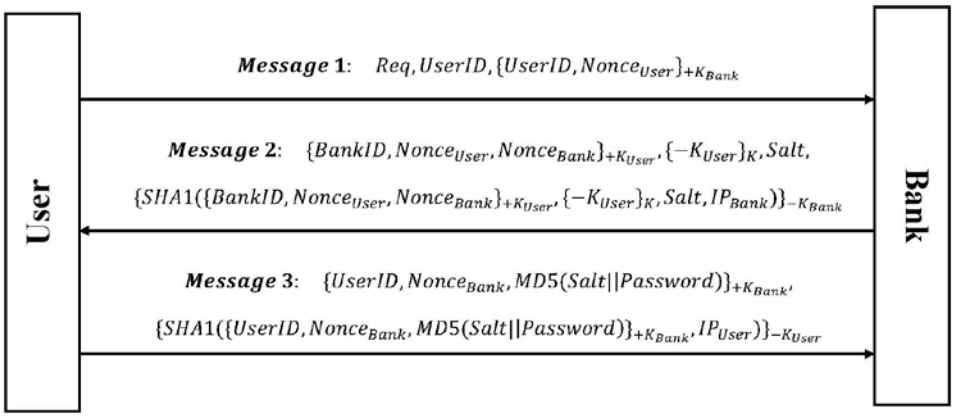

Fig. 2. Proposed Multi-Factor Authentication Protocol

5. Implementation, Testing and Results

The implementation and testing were divided into two main parts. The first was the registration system, and the second was the authentication or the login system. For each system, both the ordinary or existing system as well as the proposed system were implemented so that comparisons could be made.

This section is divided into the following subsections. The first contains the proposed interface of both the registration and the login system. The second describes the test environment as well as the specification of the system used in testing. The third is the performance evaluation of the proposed registration and authentication protocols compared with their respective existing systems. User satisfactory was also surveyed so that the satisfaction level of the proposed systems could be compared with that of the existing systems.

5.1 User interface

There were two main screens to be designed and developed here in this research. They included the user interfaces for the proposed registration process and the multi-factor authentication protocol. The designs for both processes were done in accordance with the protocols proposed in the previous section while making sure that the familiarity for the users on the interface was still there. In other words, an attempt was made to keep the designed interface similar to a typical Internet banking registration and login screens as much as possible.

Fig. 3 depicts the design and implementation of the user interface of our proposed registration process. It can be seen that the screen still resembles the familiar interface of any Internet banking registration screen. That is, it consists of textboxes for personal detail, account detail as well as chosen username and password. There is only one extra field here called *iterations* for users to enter their desired number during this registration process.

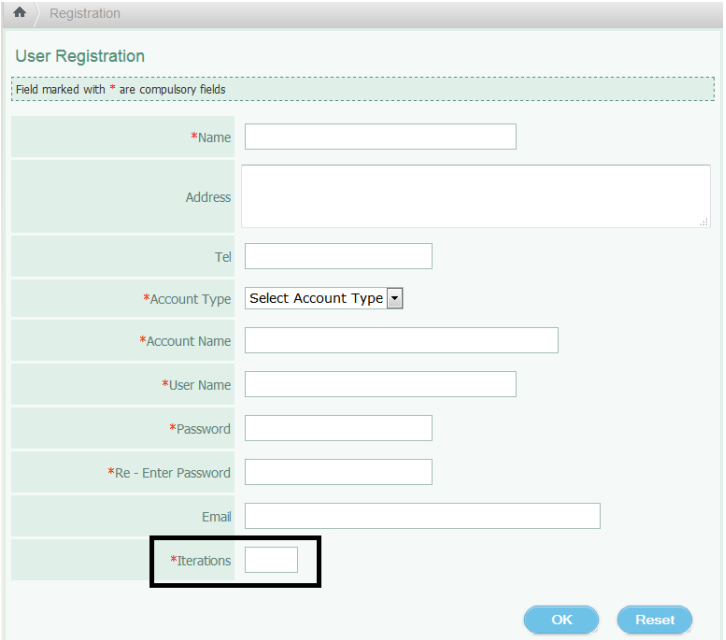

Fig. 3. Proposed Registration Screen

Fig. 4 shows the design and implementation of the user interface of our proposed multi-factor authentication protocol. Usually, a typical login screen for Internet banking consists of two fields, username and password. In our proposed designed, an extra field called *iterations* was added to the interface for users to enter as an additional authentication factor as already explained in the previous section. Even so, the proposed interface still resembles the interface of an ordinary login process.

Fig. 4. Proposed Login Screen

On the whole, it can be said that the design and implementation of the user interface for the proposed registration and authentication schemes have very few differences from those of any typical Internet banking user interface. The difference lays in the extra iterations field on both screen. Apart from that everything else can stay unchanged. Furthermore, although the registration and login screens both contain an extra field as well as extra processes to generate authentication factors, users are not required to install any additional application or software on their device at all.

5.2 Test environment

The next step was to evaluate the performance of the proposed processes so that the differences in terms of time between the ordinary systems and the proposed ones could be seen. We divided the evaluation into two main processes, which included the registration process and the Internet banking authentication process.

However, before showing the evaluation results, it is important to provide the detail of how the proposed systems were developed and what test environment was adopted during the system evaluation.

Four systems were developed for the evaluation purposes. The first was the proposed registration process. The second was the ordinary Internet banking registration. The third was the proposed multi-factor authentication protocol, while the fourth was the ordinary Internet banking login system.

All the four systems were developed using PHP version 5.2 together with Codelobster PHP Edition as the platform for development. MySQL version 5.1 was used as the database while Toad for MySQL was used as the database management program. The interfaces were designed and implemented using Adobe Photoshop CS6 and Dreamweaver CS6.

After the implementation, all the systems were put at an Internet data centre called CAT-IDC in Nonthaburi province in Thailand. They were installed on the Dell PowerEdge R210 II Web server that had the specification shown in **Table 3**.

For the performance evaluation purposes, it was decided that the experiments were to be done using a normal ADSL Internet connection, whose maximum speed was 10 Mbps, from a typical laptop to the Web server. All the tests were carried out this way to mimic the environment of the real Internet banking processes where transactions were done via a normal Internet connection anyway.

Table 3. Web Server Specification

Processor:	Intel Xeon Quad Core E31230 3.20 GHz, 8M Cache
Memory:	8GB (2 x 4GB) 1600Mhz, Dual Ranked UDIMM
Hard Disk:	2 x 500GB 3.5-inch 7.2K RPM SATA II - Non Hotplug
Network Interface:	One Dual Port Broadcom BCM 5716

5.3 Performance evaluation

The performance evaluation was divided into two parts, registration and authentication. For the registration system, the evaluation was done by comparing the time taken to complete the ordinary registration process with the proposed registration process. Similarly, the performance evaluation of the authentication process was the comparison of the time taken to complete the process between the ordinary one and the proposed multi-factor authentication protocol.

5.3.1 Registration

Seventy users were created for the performance evaluation of the proposed registration system. Ten users were used for each number of iterations, which included 1, 5, 10, 20, 100 and 500 iterations. The other ten users were used for the evaluation of the ordinary registration system. The time taken to complete the registration process was measured from the moment the OK button was pressed until the moment when all the information was transferred to and stored on the server. **Table 4** shows the average time taken for each registration scenario.

Table 4. Average Time Taken to Complete Registration

Type	Iterations	Time (s)
Ordinary	-	0.0000077
Proposed	1	0.02506
	5	0.03316
	10	0.03345
	20	0.03362
	100	0.03399
	500	0.03551

In order to make the comparison between the ordinary and proposed registration systems fair, only the proposed system with one iteration will be discussed. The average time taken to complete the registration process in the ordinary system is approximately 0.0000077 seconds, while the average time taken for the proposed system is approximately 0.02506 seconds. This shows that the ordinary systems works around 3200 times faster.

The main reason that the proposed registration system works more slowly is because there was a need to carry out many more processes. They were public key generation, private key generation, symmetric key generation and the encryption of the private key. None of these were done in the ordinary system. All of these processes were carried out in an exchange for more authentication factors than the existing method. Furthermore, even though 0.02506 seconds were higher than the time of the ordinary method, it was deemed fast enough for users not to feel any delay. This claim was made by [33] which stated that 0.1 seconds was the time that humans could feel that there was a delay.

5.3.2 Authentication

Similar to the above, the evaluation of the authentication protocols was done by comparing the proposed method with the ordinary one. The time taken to complete the authentication or login process was measured from the moment the login button was pressed until the moment the users were allowed to be in the Internet banking system.

For the ordinary login system, usernames and passwords of ten users were entered and the times it took to complete the authentication process were measured.

For the proposed login method, sixty sets of usernames and passwords were used, ten for each of the 1, 5, 10, 20, 100 and 500 iterations. Once the username, password and number of iterations were entered, the proposed authentication protocol, explained in Section 3.2, would be carried out.

Table 5 shows the average time taken to complete the authentication protocols for both the ordinary login method and our proposed one.

Table 5. Average Time Taken to Complete Authentication

Type	Iterations	Time (s)
Ordinary	-	0.00396
Proposed	1	0.10750
	5	0.10780
	10	0.10790
	20	0.10810
	100	0.10845
	500	0.10883

Again, in order to make the performance evaluation of the authentication protocols fair, only the ordinary login process and the proposed authentication protocol with one iteration would be compared. It can be seen from **Table 4** that the average time taken to complete the login process was approximately 0.00396 seconds for the ordinary system and approximately 0.10750 seconds for the proposed authentication protocol. The results imply that the login process of the ordinary system was around twenty-seven times faster than the proposed method. This is because in the proposed authentication protocol, three messages had to be exchanged between the user and the server. Moreover, during the protocol run, the proposed protocol had to carry out many processes. They included digital signature verification, public key encryption and decryption, symmetric key generation and the decryption of private key. However, with the higher amount of time, the proposed protocol came with higher security as already proved earlier.

Having examined the time taken to complete the proposed multi-factor authentication protocol, it is time to turn the attention to the communication cost, in terms of the number of messages and the number of bits. The proposed protocol was designed and implemented with the following components: *UserID* and *BankID* were 8 bits long, the $Nonce_{User}$ and $Nonce_{Bank}$ were 32 bits long, the *salt* value was 64 bits long, the *password* was at least 64 bits long, the *IP* addresses were 32 bits long, the *symmetric key* was 256 bits long and the *RSA key* was 2048 bits long.

The proposed multi-factor authentication protocol consists of three messages. The communication cost, in terms of the number of bits, is summarised in **Table 6**.

Table 6. Total Number of Bits of the Authentication Protocol Messages

Message Number	Message Size (Bits)
1	2,120
2	4,416
3	4,096
Total	19,632

It can be seen that the total number of bits of three messages of the protocol is 19,632 bits, with the first having 2,120 bits, the second having 4,416 bits and the third having 4,096 bits. The reason that the sizes of the messages are over two-thousand bits long is because the RSA encryption and digital signature schemes were an integral part of the protocol. The size of the key used here contributed to the size of the messages.

5.4 User satisfaction evaluation

Apart from performance evaluation, it was necessary to carry out user satisfaction evaluation, too. For this evaluation, we divided users into two groups. The first group of one hundred users had to evaluate both the ordinary registration and login systems. The other group of one hundred users had to evaluate the proposed registration system and the proposed multi-factor authentication process. It should be noted here that no users were informed of the type of registration and authentication system. This was done to ensure that there was no bias towards any particular system.

During the evaluation, users were asked several questions regarding the satisfaction towards the system being evaluated. The users were also asked to rate on the scale of 1 - 5, with 1 being the least satisfied and 5 being the most satisfied. **Table 7** presents the results of the satisfaction evaluation.

From **Table 7**, the satisfactory of users towards the registration and login systems can be explained as follows. In both the registration and login systems, the average user satisfaction scores were not significantly different. In other words, on average, our proposed registration method and the existing method received the satisfactory score of 4.15 and 4.27, respectively. For the user satisfaction of the login system, the satisfactory score of our proposed protocol achieved a slightly lower score than the existing method in 4.29 and 4.46, respectively. One reason that caused the proposed system to receive the lower score is the fact that users had to enter an extra piece of information, the iterations field, when registering and logging in.

Table 7. User Satisfaction Evaluation

User satisfaction on registration system		
Factors of Evaluation	**Satisfaction Level on Ordinary System**	**Satisfaction Level on Proposed System**
1. Easy and convenient to use	4.36	4.02
2. System response time	4.23	4.23
3. Overall satisfaction	4.23	4.19
Average	4.27	4.15
User satisfaction on login system		
Factors of Evaluation	**Satisfaction Level on Ordinary System**	**Satisfaction Level on Proposed System**
1. Easy and convenient to use	4.62	4.26
2. System response time	4.34	4.29
3. Overall satisfaction	4.41	4.33
Average	4.46	4.29

There was also a comment from one user that the iteration number might be difficult to remember, and whether or not there could be an alternative. Although [28] claimed that the memory span for digits was better than letters, it is believed that asking users a safe question instead could reduce the pressure of memorising the iteration number. However, there are two shortfalls. Firstly, there is an overhead of transforming user's answer from text to number, which will be used in the generation of a symmetric key. Secondly, there must be more than one safe question. Otherwise, there could be a risk of having users with the exact same answer, which could mean the same number would be used in the symmetric key generation process.

From the evaluation of user satisfaction, since the levels of satisfaction of the proposed systems were not much different from the existing methods, it can be claimed that the proposed registration and authentication protocols can be used in place of the existing systems without any difficulties from the user's perspective.

5.5 Security evaluation against OWASP criteria

OWASP or OpenWeb Application Security Project [34] was founded in 2004 with an aim of defining guidelines for application development as well as specifying security evaluation criteria for Web applications. One of the practices specified by OWAS is a set of evaluation criteria for multiple factors authentication system (MFAS), which can be found in the OWASP-AT-900 document [34]. OWASP have identified five threats or risks that can occur in a multi-factor authentication system on the Web environment. This is known as the 5T

model, which include (1) credential theft, (2) weak credentials, (3) session based attack, (4) malware and (5) password reuse.

The analysis and evaluation of a multi-factor authentication system using the OWASP specification is usually done to find the strength and weakness of such system. OWASP state that there are three levels of threat countermeasures. They are address, mitigate and not remediate. **Table 8** shows the results of the evaluation of our proposed protocol against the 5T model and three countermeasure levels.

Table 8. OWASP Analysis

Threat	Detail	Level
Credential Theft	The prevention of credential theft is the strength of the proposed protocol. This is because the protocol makes use of both symmetric and asymmetric encryption, which makes eavesdropping difficult. Moreover, IP addresses, digital signature and hash function are also used to prevent man-in-the-middle attack.	Address
Weak Credentials	It has to be accepted that this threat can occur within the proposed system, because the system does not ask users to pick a strong password. However, the proposed method offers a way to securely store passwords by integrating a salt value. If we look at the symmetric key as a personal credential, it can be seen that the way they key is derived must contain the components specified by Ma *et al.*	Mitigate
Session Based Attack	The session based attack is an attack that reuses an old session, which can be prevented by the use of randomly generated tokens. This is in accordance with our proposed protocol, which uses a fresh nonce in every message. Hence, a replay attack is not possible.	Address
Malware	Banking malware has the ability to modify information that is being exchanged between client and server. It has to be accepted that our proposed protocol cannot completely prevent this threat. However, the protocol can use the digital signature and one-way hash function mechanisms to detect when unauthorised modifications on messages occur.	Mitigate
Password Reuse	It is accepted that the proposed protocol has a weakness in this area, because the password is not changed every time the user logs in. Having said that, changing the password every time either by using a password generator or by using SMS decreases the convenience for users as already mentioned previously.	Not Remediate

From the analysis and evaluation of the proposed multi-factor authentication protocol using the OWAST-AT-900 criteria or the 5T model, it can be seen that the proposed protocol has mitigated the weak credential and malware threats, while addressing the credential theft and session based attack threats altogether. Unfortunately, the threat of password reuse has not been addressed nor mitigated due to the reasons stated earlier. However, in order to reduce the risk of this particular threat it is recommended that the "something you process" authentication method provided by [35] can be applied.

In addition to the above evaluation, the protocol admittedly has one limitation to it. That is, if the user's password is guessed or known by an adversary, the system will become more vulnerable. This is because the password is one component that is used to generate the user's symmetric key, which in turn can unlock his or her private key. Fortunately, in order to reduce the risk of passwords being compromised, such methods as [36] and [37] can be applied.

5.6 Security evaluation against known attacks

This section defines an adversary model which specifies the capabilities of the adversary. The adversary model here is inspired partly by the work of [38]. The evaluation of the security of the proposed multi-factor authentication protocol against the adversary model or the known attacks is also presented in this section. Since the formal proof of correctness and security of the protocol as well as the OWASP analysis have been provided in Section 4 and Section 5.5 respectively, this is not intended to be another proof. It is, however, to provide the discussion and evaluation of the protocol security against known attacks.

In the typical Internet environment, it is possible that an adversary has a complete control of the network and its communication channel and at times has control of the end entities participating in the protocol [38] [39]. That means an adversary has the capabilities to carry out both passive and active attacks [39].

In passive attacks, the attacker is modeled to have the ability to violate confidentiality by eavesdropping the traffic between two protocol participants, i.e., user and bank. In active attacks, the attack can violate the integrity of protocol messages by insertion, deletion and modification. Moreover, the attacker can carry out replay and man-in-the-middle attacks.

First of all, all three messages of the proposed authentication protocol are encrypted – the first by asymmetric cryptography, the second by both symmetric and asymmetric cryptography and the third by asymmetric cryptography. As a result, passive eavesdropping will not reveal any useful security information. That is, the nonce values of the user and the bank, the private key of the user as well as the user's password will not be seen or known by any unauthorised entities. Hence, it is difficult for an adversary to violate confidentiality here. This will always remain true, provided that the decryption keys are unknown to the attacker.

Secondly, if there are any message modification and message forgery by an unauthorised manner during the protocol run, they can be and will be detected because a cryptographic hash function together with digital signature are applied in the protocol messages. That means if the content in a message is modified by an attacker, it will be detected. The message will, therefore, be discarded by the recipient.

Thirdly, replay attacks are not possible due to the use of random nonce values in all the protocol messages. The nonce values are freshly generated by either the user or the bank in every message and are also encrypted. It is unlikely that the attacker will be able to provide the correct nonce. Moreover, using the nonce as an anti-replay mechanism has been proven by the GNY analysis to be true.

Fourthly, man-in-the-middle attack differs from the above attacks in that it aims to attack the identities of the protocol participants, rather than the protocol messages. The inclusion of the user's IP address and the bank's IP address in the second and third messages respectively can reduce the risk of man-in-the-middle attack. That is, the IP addresses of the end entities included in the signature generation process should assist in detecting a man-in-the-middle since the attacker's IP address will be different from those of the end entities. If the different IP address is used by the attacker, this abnormality will be detected. Furthermore, digital signature of each of the protocol participants is there to help the process of peer entity authentication, which is also a mechanism to prevent man-in-the-middle attack [39].

Fifthly, user anonymity can be achieved in two ways. The first is the fact that password exposure is dealt with by submitting *MD5(Salt||Password)*, rather than just *Password* or *MD5(Password)*. Secondly, similar to the argument in [38], the *UserID* and *BankID* are all concealed and encrypted in the Messages 2 and 3 of the protocol.

On the whole, it should be noted that successful attacks in the scenarios described above are not likely. This is because they involve the breaking of all cryptographic algorithms used in the protocol, including RSA, SHA-1, SHA-256, MD5 and AES-256. Moreover, for an adversary to successfully carry out one of the mentioned attacks, secret keys will have to be compromised, which again is not a simple task to carry out.

6. Conclusion

Financial institutions have provided Internet banking service to their users for some time. This service allows users to carry out financial transactions such as checking their account balance and making payment. With this, financial institutions must store personal information of users, which also include transaction history and account information. It is, therefore, necessary to have a mechanism to reduce the risk of being attacked. That mechanism is authentication.

Authentication has already been used by Internet banking provided by all financial institutions, with one-factor authentication being the most popular. This paper has shown that one-factor authentication comes with many vulnerabilities. Even two-factor authentication does not guarantee security these days.

This paper, therefore, has provided an attempt to design and develop a multi-factor authentication protocol that would be used during the login process. The protocol was designed with the following objectives. Firstly, it should not affect how users used the Internet banking services. Secondly, mutual authentication between user and the bank's server must be achieved. Thirdly, transaction authentication during the protocol run must be accomplish. Fourthly, there must be more than two factors used for authentication. Finally, the protocol must at least provide better security.

The proposed multi-factor authentication protocol began with the generation of authentication factors during the registration process. These factors would then be used in the actual authentication protocol during the login process. The factors of authentication to be generated and used include a username, a password, number of iterations, a public key, a private key, a symmetric key, a digital signature and an IP address. All of these are unique to each user. They are also kept confidential, except for the salt value and the user's public key which, in theory, can be exposed anyway.

Table 9 shows the differences in factors of authentication between the existing authentication protocols and our proposed multi-factor authentication protocol.

Table 9. Authentication Protocol Comparison

Authentication Factors / Method of Authentication	Username	Password	One-Time Password	Biometric	Number of Iterations	Public Key	Private Key	Symmetric Key	Digital Signature	IP Address
One-Factor Authentication	✓	✓								
Two-Factor Authentication	✓	✓	✓							
Two-Factor with Biometric Authentication	✓	✓		✓						
Proposed Multi-Factor Authentication	✓	✓			✓	✓	✓	✓	✓	✓

The proposed method only requires three messages between the bank's server and the client to complete the authentication process. In each of the three messages various authentication factors are applied so that the objectives of the protocol are met.

Even though the analysis showed that there is a limitation to the proposed protocol, especially the risk of password reuse, it is still believed that the proposed multi-factor authentication protocol is correct and secure. This is in accordance with the proof and analysis by the logic of GNY as well as the OWASP-AT-900 criteria.

References

[1] G. D. Williamson, "Enhanced Authentication in Online Banking," *Journal of Economic Crime Management*, vol. 4, no. 2, pp. 1-42, 2006.

[2] M. Johnson, "A New Approach to Internet Banking," *University of Cambridge Computer Laboratory*, Cambridge, UK, 2008.

[3] Y. Espelid, L.-H. Netland, A. N. Klingsheim and K. J. Hole, "A Proof of Concept Attack against Norwegian Internet Banking Systems," *Financial Cryptography and Data Security*, pp. 197-201, 2008. Article (CrossRef Link)

[4] D. Stebila, P. Udupi and S. Chang, "Multi-Factor Password Authenticated Key Exchange," in *Proc. of the 8th Australasian Conference on Information Security*, pp. 55-66, 2010.

[5] F. Aloul, S. Zahidi and W. El-Hajj, "Two Factor Authentication using Mobile Phones," in *Proc. of the IEEE International Conference on Computer Systems and Applications*, pp. 641-644, 2009. Article (CrossRef Link)

[6] D. v. Thanh, I. Jrstad, T. Jonvik and D. v. Thaun, "Strong Authentication with Mobile Phone as Security Token," in *Proc. of the 6th IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 777-782, 2009.

[7] M. Marlinspkie, *New Tricks for Defeating SSL in Practice*, BlackHat Conference, 2009.

[8] D. Emm, M. Garnaeva, R. Unuchek, D. Makrushin and A. Ivanov, "IT Threat Evolution in Q3 2015," *Kaspersky Lab*, Moscow, Russia Federation, 2015.

[9] K. C. Park, J. W. Shin and B. G. Lee, "Analysis of Authentication Methods for Smartphone Banking Service using ANP," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 6, pp. 2087-2103, 2014. Article (CrossRef Link)

[10] The Telegraph, "BoE Cyber Attack Exercise Shows Banks Unprepared," 2014. [Online]. Available: http://www.telegraph.co.uk/finance/bank-of-england/10620937/BoE-cyber-attack-exercise-shows-banks-unprepared.html. [Accessed April 2016].

[11] A. Hiltgen, T. Kramp and T. Weigold, "Secure Internet Banking Authentication," *IEEE Security and Privacy*, pp. 21-29, March - April 2006. Article (CrossRef Link)

[12] Y. Desmedt, I. Karaolis, M. Adham and A. Sadr-Azodi, "How to Attack Two-Factor Authentication Internet Banking," in *Proc. of the 17th International Conference on Financial Cryptography and Data Security*, pp. 322-328, 2013. Article (CrossRef Link)

[13] B. Schneier, "Two-Factor Authentication: Too Little, Too Late," *Communications of the ACM*, vol. 48, no. 4, p. 136, April 2005. Article (CrossRef Link)

[14] M. Mannan and P. C. V. Oorschot, "Using a Personal Device to Strengthen Password Authentication from an Untrusted Computer," in *Proc. of the 11th International Conference on Financial Cryptography and 1st International Conference on Usable Security*, Scarborough, Trinidad and Tobago, pp. 88-103, 2007. Article (CrossRef Link)

[15] R. Rittenhouse and J. A. Chaudhry, "A Survey of Alternative Authentication Methods," in *Proc. of the 2015 International Conference on Recent Advances in Computer Systems*, Saudi Arabia, pp. 179-182, 2015. Article (CrossRef Link)

[16] A. M. Hagalisletto and A. Riiber, "Using the mobile phone in two-factor authentication," in *Proc. of the 1st International Workshop on Security for Spontaneous Interaction*, Innsbruck, Austria, 2007.

[17] R. D. Pietro, G. Me and M. A. Stangio, "A Two-Factor Mobile Authentication Scheme for Secure Financial Transactions," in *Proc. of the International Conference on Mobile Business*, Sydney, Australia, 2005. Article (CrossRef Link)

[18] D. Wang, N. Wang, P. Wang and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, pp. 162-178, 2015. Article (CrossRef Link)

[19] B. Adida, "Beamauth: Two-Factor Web Authentication with a Bookmark," in *Proc. of the ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, pp. 48-57, 2007. Article (CrossRef Link)

[20] A. P. Sabzevar and A. Stavrou, "Universal Multi-Factor Authentication Using Graphical Passwords," in *Proc. of the IEEE International Conference on Signal Image Technology and Internet Based Systems*, Bali, Indonesia, pp. 625-632, 2008. Article (CrossRef Link)

[21] K. Najan, P. Ragava, A. Sawant and S. Madchane, "Image Steganography, Compression and Image Morphing for Banking Website," *International Journal for Innovative Research in Science and Technology*, vol. 2, no. 10, pp. 56-58, 2016.

[22] S. Mahitthiburin and S. Boonkrong, "Improving Security with Two-Factor Authentication using Image," *KMUTNB: International Journal of Applied Science and Technology*, vol. 8, no. 1, pp. 33-43, January-March 2015. Article (CrossRef Link)

[23] K. M. Apampa, T. Zhang, G. B. Wills and D. Argles, "Ensuring Privacy of Biometric Factors in Multi-Factor Authentication Systems," in *Proc. of the International Conference on Security and Cryptography*, Porto, Portugal, 2008.

[24] H. Al-Assam, H. Sallahewa and S. Jassim, "On Security of Multi-Factor Biometric Authentication," in *Proc. of the International Conference for Internet Technology and Secured Transactions*, London, UK, 2010.

[25] L. T. Premakumari and A. S. Jothi, "Multimodal Biometric Endorsement for Secure Internet Banking using Skin Spectroscopy, Knuckles Texture and Finger Nail Recognition," *International Research Journal of Engineering and Technology*, vol. 3, no. 2, pp. 1086-1090, 2016.

[26] M. Al-Fairuz and K. Renaud, "Multi-channel, Multi-level Authentication for More Secure eBanking," in *Proc. of the International Conference on Information Security for South Africa*, 2010.

[27] X. Huang, Y. Xiang, E. Bertino, J. Zhou and L. Xu, "Robust Multi-Factor Authentication for Fragile Communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568-581, November-December 2014. Article (CrossRef Link)

[28] C. W. Crannel and J. M. Parrish, "A Comparison of Immediate Memory Span for Digits, Letters and Words," *The Journal of Psychology*, vol. 44, pp. 319-327, 1957. Article (CrossRef Link)

[29] W. Ma, J. Campbell, D. Tran and D. Kleeman, "Password Entropy and Password Quality," in *Proc. of the 4th International Conference on Network and System Security (NSS)*, pp. 583-587, 2010. Article (CrossRef Link)

[30] S. Boonkrong, "Security of Passwords," *Journal of Information Technology*, vol. 8, no. 2, pp. 112-117, July - December 2012.

[31] Information Technology Laboratory, "Secure Hash Standard (SHS)," 2012.

[32] L. Gong, R. Noodham and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," in *Proc. of the 1990 IEEE Symposium on Research in Security and Privacy*, Oakland, California, USA, pp. 234-248, 1990. Article (CrossRef Link)

[33] R. B. Miller, "Response Time in Man-computer Conversational Transactions," in *Proc. of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, San Francisco, California, pp. 267-277, 1968. Article (CrossRef Link)

[34] OWASP, "Testing Multiple Factors Authentication (OWASP-AT-009)," [Online]. Available: https://www.owasp.org/index.php/Testing_Multiple_Factors_Authentication_(OWASP-AT-009). [Accessed July 2014].

[35] N. Usavapipatkul, K. Yochanang and S. Boonkrong, "Authentication by One-Time Password using the Solution of Random Numeric and Simple Calculation," in *Proc. of the 8th National Conference on Computing and Information Technology*, Chonburi, Thailand, pp. 303-310, 2012.

[36] K.-P. Yee and K. Sitaker, "Passpet: Convenient Password Management and Phishing Protection," in *Proc. of the Second Symposium on Usable Privacy and Security*, Pittsburgh, Pennsylvania, USA, pp. 32-43, 2006. Article (CrossRef Link)

[37] S. Gaw and E. W. Felton, "Password Management Strategies for Online Accounts," in *Proc. of the Second Symposium on Usable Privacy and Security*, Pittsburgh, Pennsylvania, USA, pp. 44-55, 2006. Article (CrossRef Link)

[38] D. Wang and P. Wang, "Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, 2016. Article (CrossRef Link)

[39] E. Rescorla and B. Korver, "RFC 2552: Guidelines for Writing RFC Text on Security Considerations," IETF, 2003.

Sirapat Boonkrong is an associate professor at the Faculty of Information Technology, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He received his B.Sc. and Ph.D. in Computer Science from the Department of Computer Science at the University of Bath, UK. His main area of research is information and network security. Previously, Sirapat worked as a researcher at National Electronics and Computer Technology Center (NECTEC) in Thailand. He also has experience in industry as a project manager at an IBM-partnered company. He is currently a full-time lecturer at the Faculty of Information Technology, KMUTNB and is also supervising several Ph.D. students all of whom are in the field of information and network security.