A Password Authentication Scheme Over Insecure Networks*

I-En Liao† Cheng-Chi Lee†‡ Min-Shiang Hwang§

Department of Management Information Systems§
National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan
Email: mshwang@nchu.edu.tw

Department of Computer Science†
National Chung Hsing University
250 Kuo Kuang Road, 402 Taichung, Taiwan

Department of Computer & Communication Engineering‡
Asia University
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung, Taiwan

September 29, 2005

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC92-2213-E-324-023.

§Responsible for correspondence: Prof. Min-Shiang Hwang

A Password Authentication Scheme Over Insecure Networks

Abstract

Authentication ensures that system's resources are not obtained fraudulently by illegal users. Password authentication is one of the simplest and the most convenient authentication mechanisms over insecure networks. The problem of password authentication in an insecure networks is present in many application areas. Since computing resources have grown tremendously, password authentication is more frequently required in areas such as computer networks, wireless networks, remote login, operation systems, and database management systems. Many schemes based on cryptography have been proposed to solve the problem. However, previous schemes are vulnerable to various attacks and are not efficient, nor user friendly. Users cannot choose and change their passwords at will. In this paper, we propose a new password authentication scheme to achieve the all proposed requirements. Furthermore, our scheme can support the Diffie-Hellman key agreement protocol over insecure networks. Users and the system can use the agreed session key to encrypt/decrypt their communicated messages using the symmetric cryptosystem.

Keywords: Cryptography, Diffie-Hellman key agreement, network security, one-time password, password authentication.

1 Introduction

Password authentication is one of the simplest and the most convenient authentication mechanisms over insecure networks. It provides the legal users to use the resources of the remote systems. Many Internet applications are based on password authentication, for example, remote login, government organizations, private corporations, database management systems, and school systems. However, the current Internet environment is vulnerable to various attacks such as replay attack, guessing attack, modification attack, and stolen-verifier attack. Therefore, a number of researchers have proposed several password authentication schemes for secure login of legal users.

In traditional password authentication scheme, each user has an identifier (ID) and a password (PW). If a user wants to login to a remote server, he/she submit his/her ID and PW to the server. A simplest authentication approach is to store and maintain a password table including users' IDs and PWs in the remote server. Upon receipt of user's ID and PW, the remote server searches the password table to check whether or not the submitted ID and PW match with those stored in the password table. Once the ID and PW match the corresponding pair stored in the server's password table, the user will be granted access to the server's facilities. Since the user's password is stored in plain-text form in password table, this approach is vulnerable to the revelation of the passwords. An intruder can impersonate a legal user by stealing the user's ID and PW from the password table. This attack is called stolen-verifier attack. Besides, two disadvantages are found in this approach. One is that the system load is very high. If a lot of users register with the system, the password table will become big and hard to maintain. The other is an intruder can intercept a user's ID and PW from the Internet and then replay it later to login. This attack is called the replay attack.

To prevent the password table from stealing by others, password is usually hashed or encrypted inside the computer [8, 26]. However, the transmission of unencrypted password could be stolen by wire tap. Therefore, this scheme is still vulnerable to the replay attack. To overcome this problem, Lamport proposed one-time password using one-way hash function [19]. But there are three shortcomings in Lamport's method. The first one is the high hash overhead. The second one is the necessity for password resetting. Finally, a verification table should be stored in the server to verify the legitimacy of a user. If an intruder can somehow break into the server, the contents of the verification table can be modified easily. This is called modification attack. The attack also means that an attacker can impersonate other legal users by constructing a valid login request from an intercepted login request. Researchers have recognized this problem and proposed solutions in which the verification table is no longer required in the server [15]. From the Lamport's method, Haller and Yeh derived S/KEY one-time password scheme [9, 10] and Yeh's scheme,respectively [41]. Some researchers further pointed out that S/KEY scheme is not secure [22]. To solve the above high hash overhead and password resetting problems, Shimizu proposed the CINON protocol [32]. In 2000 and 2001, Sandirigama et al. and Lin et al. proposed SAS [29] and OSPA protocol [21],respectively, which were intended to be superior to the Lamport's protocol, CINON protocol, and the PERM protocol [33], in terms of storage utilization, computing time, and transmission overhead. In 2002, Chen and Ku proposed two stolen-verifier attacks on SAS and OSPA prtocol [3].

Generally speaking, there are three types of identity authentication methods [16, 17]:

1. identity authentication of something known, such as password;
2. identity authentication of something possessed, such as smart cards;
3. identity authentication of some personal characteristics, such as fingerprint.

Combining these methods can enhance the security level of a system. Usually, most systems use the first two methods to identify a user. Recently, many password authentication schemes using smart cards have been proposed by some researchers [1, 2, 4, 5, 11, 13, 14, 15, 18, 20, 30, 31, 34, 35, 36, 37, 39, 40]. In these schemes, the smart-card-oriented remote login authentication scheme is used to authenticate a legitimate user. The smart card contains a microprocessor, which can perform arithmetic operations quickly, an I/O port, a RAM, and a ROM in which some messages are stored. Therefore, there is no need to store a password table or verification table in the server.

In [2], Chang and Wu proposed a remote password authentication scheme with a smart card based on the Chinese Remainder Theorem (CRT). The scheme does not need to store verification table and is secure against attacks of replaying previously intercepted requests. However, the user's password of this scheme can not be chosen and changed freely by the owner. A similar problem also occurs in some schemes proposed by [14, 30, 34]. Chan et al. [1] and Shen et al. [30] respectively further pointed out that Hwang et al.'s scheme [14] is insecure. In [38], Yamaguchi et al. proposed a simple but efficient authentication system, SPLICE/AS. Later, Hwang et al. pointed out that SPLICE/AS system is vulnerable to the guessing attack [12]. In [37], Wu proposed an efficient scheme based on the geometric Euclidean plane. The merits of this scheme are its simplicity of geometry and the property that users can freely choose their own passwords. However, the scheme is insecure as indicated in [11]. In [15], Jan and Chen proposed a new scheme without verification table. Users can freely choose and change their own passwords in the scheme. However, the scheme is not efficient because it uses the public key cryptosystem. The computational cost is very high. In [39], Yang and Shieh proposed two methods to prevent replay attack. Their schemes do not store passwords or verification tables in the server, and let users freely change their own passwords. However, two papers [31, 35] pointed out that Yang and Shieh's schemes have a drawback in that an intruder is able to impersonate a legal user by constructing a valid login request from an intercepted login request. Therefore, Yang and Shieh's schemes cannot prevent modification attack. In 2002, Hwang et al. [13] and Chien et al. [4] proposed an efficient and practical smart-card-based schemes based on secure one-way hash function. In those schemes, the authors claimed that their schemes can achieve the following characteristics: (1) the verification or password tables are not required in the server; (2) the communication cost and the computational cost is very low; (3) the replay attack problem is completely solved; and (4) users can freely choose their passwords. However, in [13], their scheme can not achieve mutual authentication. And in [4], their scheme did not let users freely change their passwords.

Although there are many smart-card-oriented schemes proposed to authenticate a legitimate user, none of them can solve all problems. In these schemes, another important problem is the lost smart card problem. If an intruder picks up a smart card lost by some legal user, then he/she can impersonate the user to login the system by off-line guessing the password of the card owner.

In this paper, we first propose the following ten requirements for evaluating a new password authentication scheme. The ten requirements solve all problems in smart-card-oriented schemes. Each requirement is an important and independent requirement for a new password authentication scheme.

R1. The passwords or verification tables are not stored inside the computer.

R2. The password can be chosen and changed freely by the owner.

R3. The password cannot be revealed by the administrator of the server.

R4. The passwords are not transmitted in plain text on network.

R5. No one can impersonate a legal user to login the server.

R6. The scheme must resist the replay attack, guessing attack, modification attack, and stolen-verifier attack.

R7. The length of a password must be appropriate for memorization.

R8. The scheme must be efficient and practical.

R9. The scheme can achieve mutual authentication. Not only can server verify the legal users, but users can verify the legal server.

R10. The password cannot be broken by guessing attack even if the smart card is lost.

The purpose of this paper is to propose a new password authentication scheme to meet the aforementioned ten requirements. In addition, our scheme is intended to be superior to the SAS protocol and OSPA protocol in terms of storage utilization, computing time, and transmission overhead. This paper is organized as follows. In next section, we present a new password authentication scheme. In Section 3, our scheme is compared with other methods. Finally, Section 4 presents our conclusions.

2 The New Password Authentication Scheme

In this section, we present a new password authentication scheme using smart card and show that our scheme can achieve all of the ten requirements described in Section 1.

2.1 Basic Concepts

Our scheme employs some basic concepts, such as one-way hash function,e.g., MD5 [27], or SHA-1 [24], discrete logarithm problem [7], and Diffie-Hellman key agreement protocol [6]. We briefly describe these basic concepts in this subsection.

2.1.1 One-way Hash Function

A one-way hash function $h : a \rightarrow b$ is a function with the following properties:

- The function h takes a message of arbitrary length as the input and produces a message digest of fixed-length as the output.

- The function h is one-way in the sense that given a, it is easy to compute $h(a) = b$. However, given b, it is hard to compute $h^{-1}(b) = a$.

- Given a, it is computationally infeasible to find a' such that $a' \neq a$, but $h(a') = h(a)$.

- It is computationally infeasible to find any pair a and a' such that $a' \neq a$, but $h(a') = h(a)$.

2.1.2 Discrete Logarithm Problem

Until now, solving discrete logarithm problem is still a hard problem. We describe this problem as follows. Assume that g is a generator of Z_p^* and p is a large prime number. Consider the following equation:

$$J = g^j \bmod p. \tag{1}$$

If we know g, j, and p, it is very easy to compute J. However, if we know g, J, and p, it is very difficult to solve the equation for j. The difficulty is due to factoring prime numbers as that required for RSA [28]. The problem of solving equation 1 for j is called discrete logarithm problem.

2.1.3 Diffie-Hellman Key Agreement

In 1976, Diffie and Hellman proposed a key agreement scheme for making agreement on a session key over insecure networks. The scheme allows two parties communicate each other in a secure communication with the agreed session key. Its security is based on solving discrete logarithm problem. Assume that Alice and Bob are to agree on a session key over insecure networks. The parameters g and p are public. Then, they do the following steps to agree on a session key.

- Alice randomly chooses a large number a and sends Bob $A = g^a \bmod p$.
- In the meantime, Bob also randomly chooses a large number b and sends Alice $B = g^b \bmod p$.
- After that, Alice and Bob can calculate their session key as $K = B^a \bmod p = A^b \bmod p = g^{ab} \bmod p$.

Without knowing a and b, no one can listen on the Alice-Bob channel. To derive a and b, it is discrete logarithm problem.

2.1.4 The Security of Our Proposed Scheme

The security of our proposed scheme is based on having both the properties of discrete logarithm problem and secure one-way hash function. Most password authentication schemes are only based on one of these two properties. Recently, many schemes proposed are based on the one-way hash function [4, 21, 29, 34, 41]. The computational complexity of their schemes is superior to the discrete-logarithm-problem-based schemes. However, our scheme is more secure than those based on one of the two problems. And it does not add too much computational complexity. Furthermore, our scheme can achieve all of the ten requirements discussed in Section 1.

Comparing with other schemes that have a verification table for storing encrypted user passwords, our scheme let the server secretly keep only one value. The value is maintained by the server. If the value is known to others, our scheme is destroyed. Therefore, the server should maintain and store this value very securely.

2.2 Notations

The following notations are used throughout this paper.

- A denotes the user. A user of a computer system uses the authentication protocol to login the host.

- S denotes the authentication server. A server takes charge of verifying legal users and providing services to users.
- ID denotes the user identity.
- PW denotes the password of user A.
- x denotes the long secret key of S.
- p denotes a large prime number.
- g denotes the primitive element in Galois field GF(p).
- SR denotes the service request. It means a request by the user to login the server.
- R denotes a random number.
- T denotes the time stamp.
- h denotes a public one-way hash function with fixed-length output, e.g. MD5 or SHA-1.
- $\|$ denotes the concatenation.
- The expression "$X \rightarrow Y : M$" represents the message M sent from X to Y through an open channel.
- The expression "$X \Rightarrow Y : M$" represents the message M sent from X to Y through a secure channel.

2.3 The Scheme

Our scheme consists of three phases, namely, registration phase, login phase, and authentication phase. In the registration phase, the server issues smart cards to the users who request registration. Once the user registers with the server successfully, he/she can use the resources of server through the login phase and authentication phase. In the login phase, a user inserts his/her smart card to a terminal and keys in his/her identifier (ID) and password (PW). Then the terminal computes and sends a login request message to the remote server. In the authentication phase, the remote server verifies the validity of the submitted message and determines whether the login request should be accepted or not. The registration phase is done only once by the user when a new user wants to join the system. The login phase and authentication phase are performed every time the user wants to login. Now, we describe these three phases in turn.

2.3.1 Registration Phase

In this phase, everyone who wants to register at the server would have a smart card. The smart card is issued by the server or a United Card Issue Center (UCIC). Usually, the UCIC is the server. User information stored on the card for certification includes id number, name, parent's name, address, birthdate, sex, and blood type, etc [15]. As in [15], the server which provides resources to the users would certify the validity of the card with his signature. As to how to identify server signature, we will not discuss these functions here.

Figure 1 shows the registration phase of the scheme. When a new user A wants to use the resources of the system, he/she must do this phase first. The system is not responsible for authenticating users except issuing smart cards to new users. Initially, the server first selects a large prime number p and primitive number g in GF(p). Without loss of generality, p is recommended to be 1024 bits at least. Besides, the server selects a secure one-way hash function $h(\cdot)$ and a long secret key x maintained by himself/herself. Then the server keeps x securely. The details of this phase is described in the following steps.

1. User A freely chooses his/her ID and PW. PW is a short password that is appropriate for memorization.

2. $A \Longrightarrow S$:$ID, h(PW)$

   A calculates $h(PW)$ with his/her PW using one-way function h, then sends the messages ID and $h(PW)$ to the server S through a secure channel or by hand.

3. After receiving the registration messages, S calculates $B = g^{h(x\|ID)+h(PW)} \bmod p$.

4. $S \Longrightarrow A$:ID, B, p, g

   S issues A a smart card which contains ID, B, p, g, and then sends this card to A through a secure channel.

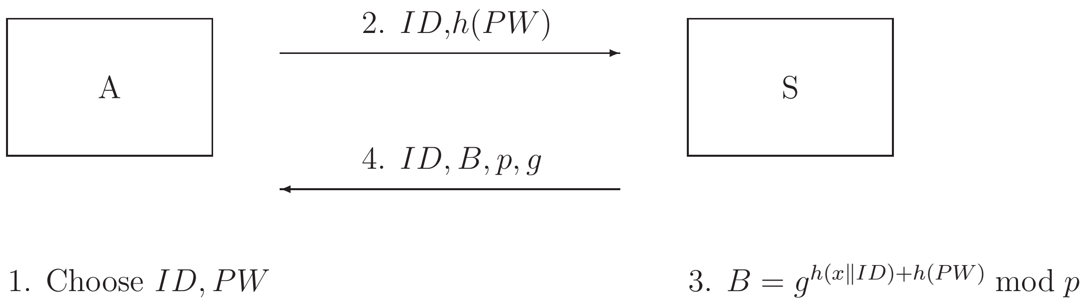

Figure 1: Registration Phase Through Secure Channel

2.3.2 Login Phase

Upon login, user A attaches his/her smart card to his/her input device, and then keys in his/her ID and PW to the device. Afterwards, the server and the smart card will perform the following steps. Figure 2 shows the login phase of the scheme.

1. $A \rightarrow S$:SR

   User A issues a login service request to server S.

2. After receiving the login service request, S calculates $B" = g^{h(x\|ID)R}$ mod p, where x is server's secret key maintained by S, ID is A's identity included in SR, and R is a random number generated by S. Then S calculates $h(B")$ using secure one-way hash function h.

3. $S \rightarrow A$:$h(B"), R$

   Server S then sends the hashed message $h(B")$ and the random number R to A.

4. Upon receiving the messages, the smart card of A calculates $B' = (Bg^{-h(PW)})^R$ mod p. Then A can verify the validity of server S by checking whether the received message $h(B")$ is equal to hashed B'. If it is correct, A calculates $C = h(T\|B')$; otherwise, the server is rejected and repeat the login phase. T is the time stamp of this login. T and R can achieve one-time password and prevent from replaying attack. We will discuss it in Section 2.5.

5. $A \rightarrow S$:ID, C, T

   A sends the login request messages (ID, C, T) to S.

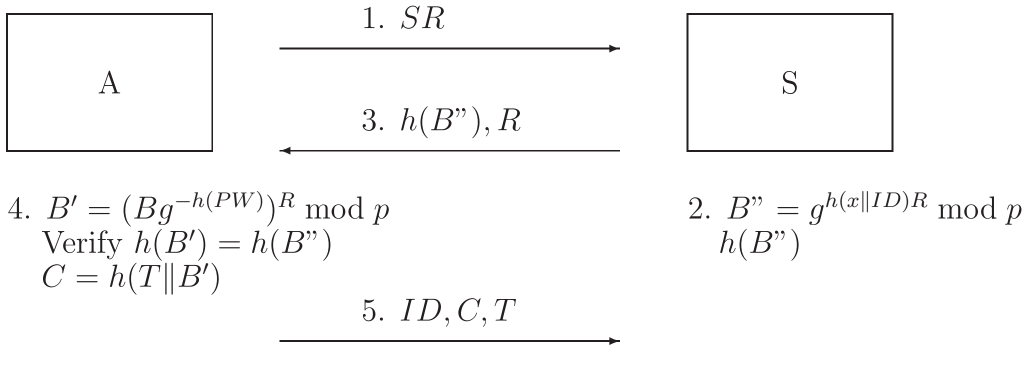

Figure 2: Login Phase

2.3.3 Authentication Phase

This phase is executed by server S to determine whether user A is allowed to login or not. Let T' be the time when S receives the login request messages. Upon receiving the login request message, S executes the following steps to verify the validity of A.

1. Check if the format of ID is correct. If it is incorrect, S rejects the login request.

2. Compare T with T'. If $T' - T \leq \Delta T$, accept A's login request; otherwise, reject it. ΔT is the legal time interval for transmission delay. It can be used to prevent from replaying attack.

3. Compute $C' = h(T \| B")$, and then check whether C is equal to C'. If it is false, then S rejects the login request; otherwise, S accepts the login request.

2.3.4 Change Password

Our scheme allows users to freely change their passwords at will. Users do not need to register the server again to change their password. Compare with some other schemes, our scheme is more friendly. In some schemes, the password of a user has to be generated by the server, and the user cannot change his/her password at will unless he/she re-registers the sever again. On the other hand, for the purpose of security, each user should be allowed to change his/her password periodically without worrying about the possible revelation of the password, or excessive charges from the server for password changing. If a user wants to change his/her password, he/she can insert his/her smart card in a terminal device and enter newly chosen password PW' as in the following steps.

1. Select a new password PW'.

2. Compute $Y = g^{h(PW')} \bmod p$.

3. Compute $Z = Bg^{-h(PW)} \bmod p$, where PW is the old password of the user and B is from the card.

4. Compute $\beta = YZ \bmod p$.

5. Replace B with β in the smart card.

After changing their passwords, Users can get the updated smart card and use the new password PW' to login to the server in the login phase.

2.4 Required Storage and Computational Complexity

In some schemes, their system keeps a verification table in the server for storing encrypted user passwords. Assume that there are n users in the server. Then the amount of space needed to store user identifiers and encrypted user passwords is $O(n)$. In our scheme, the server needs only to keep one secret value. Therefore, the space complexity of our scheme for the server is $O(1)$.

The computational complexity of our scheme is examined in the following. Let T_h be the time for executing the one-way hash function h. In Registration Phase, the time complexity is (1 addition + 1 exponentiation + 1 modular operation). In Login Phase, the time complexity is (5 T_h + 3 exponentiations + 2 modular operations + 1 generated random number + 2 multiplications). In Authentication Phase, the time complexity is (1 T_h + 1 comparison). Besides, our scheme allows users to freely change their passwords at will. The time complexity for changing password is (1 generated new password + 2 exponentiations + 2 multiplications + 3 modular operations).

2.5 Security Analysis and Discussions

In this section, we discuss the security and properties of our scheme.

1. In our scheme, no password or verification table is stored inside computer. There is only one secret key x known and maintained by the server. This

property can resist the stolen-verifier attack and modification attack because no one can steal and modify user passwords. Therefore,our scheme achieves the requirements 1 and 6 proposed in the Introduction.

2. In the registration phase of our scheme, the user password is chosen freely by the user, not assigned by the server. Furthermore, the user can change his/her password at will. This property achieves the requirement 2 proposed in the Introduction.

3. We can see that the administrator of server does not know the password of the user because it is protected by one-way function $h(PW)$. This function has the property that given a,it is easy to compute $h(a) = b$. However, given b, it is hard to compute $h^{-1}(b) = a$. This property achieves the requirement 3 proposed in the Introduction.

4. The password of the user is not transmitted on network. Instead, $h(B")$ and C are transmitted on network. Although these values include x, they are protected by one-way function and discrete logarithms problem. If an attacker knows $B"$ and intercepts R, given $B"$, R, g, and p, it is impossible to deduce x. The security is based on the difficulty of computing discrete logarithm over $GF(p)$. In addition, x is also protected by one-way function. In fact, the attacker does not know $B"$. If an attacker wants to impersonate a user, he/she must know x. Therefore, this property achieves the requirements 4 and 5 proposed in the Introduction.

5. In our scheme, we assume that p is a large prime number. The security of our scheme is based on the difficulty of computing discrete logarithm over $GF(p)$. In [15], they assume that p is about 200 digits (or 640 bits), then it needs 2^{320} (or 10^{100}) exponential operations on average to attack the password. Suppose a computer has computing power of 1 million exponential operations per second, it needs 3.17×10^{86} years to break

the password. To improve the security of our scheme, the length of p can also be set to 200 digits or over.

6. Our scheme also has one-time password property and can prevent the replay attack. The password pattern $(h(B"), C)$ is used only once in Login Phase. These messages cannot be intercepted for reuse because they are generated using R and T with different values for each login. On the other hand, the server can test if $T' - T \leq \Delta T$ to prevent the replay attack. Hence, our scheme meets the requirement 6 proposed in the Introduction.

7. Guessing attacks can be classified as on-line or off-line guessing attacks. The on-line guessing attacks can be prevented easily by limiting the number of failed login. So it will not be discussed here. In our scheme, if an intruder intercepts $h(B"), R, C$, and T, he/she cannot break PW by playing off-line guessing attacks because PW is protected by both properties of one-way function and discrete logarithm problem. Furthermore, even if the smart card is lost, off-line guessing attacks are still impossible to deduce PW. A person, who picks up a lost smart card, may try to derive the password PW from $B = g^{h(x\|ID)+h(PW)} \bmod p$. Although he/she knows $B, g, p, h()$, and ID, he/she cannot derive PW without knowing x. This x is the long secret key maintained by the server. Therefore, our scheme meets the requirements 6 and 10 proposed in the Introduction.

8. If a masqueraded server wants to cheat the requesting user, he/she must prepare a valid message $(h(B"), R)$. However, it is impossible to compute $h(B")$ because he/she has no way to derive the value $h(x\|ID)$, due to the one-way property of the one-way hash function and discrete logarithm problem. In addition, x is a long secret key maintained by the server. Without knowing x, no one can forge the server to cheat the user. In our

scheme, not only can the server authenticate the user, but the user can authenticate the server. This property achieves mutual authentication of the requirement 9 proposed in the Introduction. In our scheme, an adversary may impersonate server to cheat users by simply replaying a previously intercepted message $(h(B"), R)$ or $(h(B"||A), R, A)$. That is, the proposed scheme can not achieve mutual authentication as our claimed. To resist replaying attack, we can add time-stamp to it.

9. A user can choose a short password for easy memorization in the Registration Phase. Then the server will generate a long password pattern B using x and ID and place ID, B, p, g,into the smart card issued to the user. The user needs only to remember the short password for using the smart card. The time complexity for the smart card is not very high. It needs only two exponential operations and one hashing operation. Our scheme can be applied to many applications, for examples, remote login, e-government, e-commerce, database management systems, etc. Therefore, our scheme meets the requirements 7 and 8 proposed in the Introduction.

10. In addition, Our scheme can support the Diffie-Hellman key agreement protocol [6]. Figure 3 shows our modified Login Phase to support key agreement protocol. The Registration Phase is not modified. Once a user logins to the server successfully, he/she can use the session key $K = A^w \bmod p = W^a \bmod p = g^{aw} \bmod p$ to encrypt/decrypt the transmitted messages between the user and the server using the symmetrical cryptosystems such as DES [23] or AES [25]. During the Authentication Phase, the server authenticates the user as follows. The Step 1 and Step 2 are the same as those in the Authentication Phase. In Step 3, the server computes $C' = h(T\|B"\|W)$ and then checks whether C is equal to C'. If it is false, then S rejects the login request; otherwise, S accepts

the login request.

11. The proposed scheme is potentially vulnerable to an insider attack. That is, the insider of S can perform an off-line guessing attack on $h(PW)$ to obtain PW. If succeeds, the insider of S can try to use PW to impersonate users to login other servers employing normal password authentication methods. In this case, our proposed scheme cannot prevent it. However, in most schemes, the password of hashing table is encrypted by the administrator of S. The insider of S cannot get $h(PW)$ to obtain PW using an off-line guessing attack.

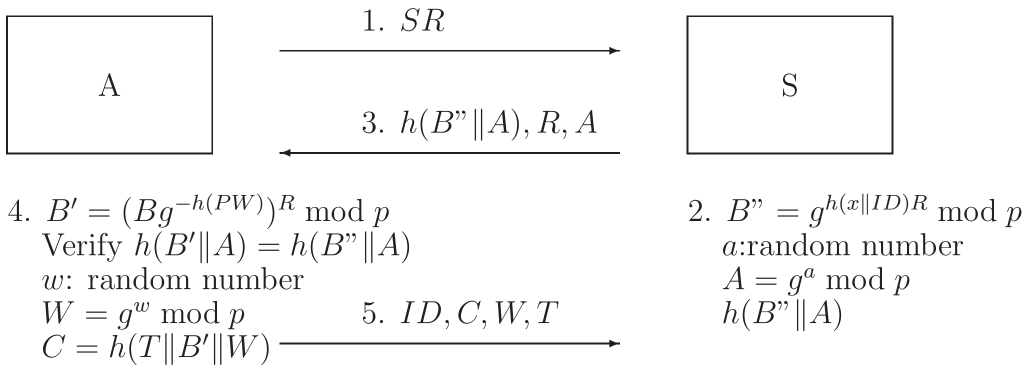

Figure 3: Modified Login Phase & Diffie-Hellman Key Agreement

3 Comparisons

In this section, we compare our scheme with other smart-card-based schemes using the ten requirements for evaluating a new password authentication schemes proposed in Section 1. Table 1 shows comparisons among smart-card-based schemes. It is easy to see that our scheme can achieve all of the ten requirements. Therefore, our scheme is a better password authentication scheme. Besides, our scheme can support the Diffie-Hellman key agreement protocol over insecure networks. Users and the system can use the agreed session key to encrypt/decrypt their communicated messages using the symmetric cryptosystem. Notice that Chieh-Jan-Tseng scheme [4] and Hwang-Lee-Tang scheme [13] are superior to other schemes except ours. Their schemes can achieve eight to nine out of ten requirements. In addition, their schemes are very efficient because they use less one-way hash functions. The computational costs are extremely low. Of course, their schemes can be extended to achieve all requirements. However, the security of their schemes is not very high because their schemes only base on one-way hash function. Hence, we proposed our scheme based on both one-way hash function and discrete logarithm problem. The discrete logarithm problem is still an open problem and is more secure than one-way hash function. Comparing with their schemes, our scheme does not add too much computational complexity. Furthermore, our scheme does not use complex public key cryptosystem. Here, we do not compare and evaluate computational costs of the proposed scheme and other schemes because the proposed scheme is an ideal scheme to achieve all requirements. We just show the scheme is superior to other schemes according to the ten requirements.

Table 1: Comparisons among the smart-card-based schemes

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10
Our Scheme	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Chang-Wu [2]	N	N	N	Y	Y	Y	Y	Y	N	Y
Wu [37]	Y	N	Y	Y	N	N	Y	Y	N	Y
Jan-Chen [15]	Y	Y	Y	Y	Y	Y	Y	N	N	Y
Yang-Shieh [39]	Y	Y	N	Y	N	N	Y	N	N	Y
Sun [34]	N	N	N	Y	Y	Y	N	Y	N	Y
Hwang-Li [14]	N	N	N	Y	N	N	N	N	N	Y
Hwang-Lee-Tang [13]	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
Chieh-Jan-Tseng [4]	Y	N	N	Y	Y	Y	Y	Y	Y	Y
Shen-Lin-Hwang [30]	N	N	N	Y	Y	Y	N	N	N	Y
Yeh-Shen-Hwang [41]	N	N	Y	Y	Y	Y	N	N	Y	Y

Ri: Proposed requirements in Section 1, Y: Supported, N: Not supported.

4 Conclusions and Future Work

In this paper, we have proposed a new password authentication scheme over insecure networks. In terms of ten requirements for evaluating a new password authentication schemes introduced in Section 1, our scheme is a better approach than the other schemes. The primary merit of our scheme is in its simplicity and practicality for implementation under insecure communication links. The advantages of our scheme are as follows:

1. Users can freely choose and change their passwords at will.
2. The verification table or password table is not stored inside the server computer.
3. It has one-time password property, which requires a different password in each transaction.
4. The security is based on having both properties of the discrete logarithm problem and secure one-way hash function. Hence, it is more secure than those schemes based on only one of these two properties.
5. Mutual authentication between the user and the server is provided.
6. It can withstand the guessing attack even though the smart card is lost.

A disadvantage of our scheme is that the server has to keep securely a secret value x. If the value is known to the others, our scheme is destroyed.

In the future, in order to provide a higher security of password authentication scheme, combining three types of identity authentication methods as described in Section 1 is a good approach. We currently research on applying our scheme to identity authentication of some personal characteristics.

References

[1] Chi-Kwong Chan and L. M. Cheng, "Cryptanalysis of a remote user authentication scheme using smart cards," *IEEE Transaction on Consumer Electronics*, vol. 46, pp. 992–993, 2000.

[2] C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings-E*, vol. 138, pp. 165–168, May 1991.

[3] Chien-Ming Chen and Wei-Chi Ku, "Stolen-verifier attack on two new strong-password authentication protocols," *IEICE Transactions on Communications*, vol. E85-B, pp. 2519–2521, November 2002.

[4] Hung-Yu Chien, Jinn-Ke Jan, and Yuh-Min Tseng, "An efficient and practical solution to remote authentication: smart card," *Computers & Security*, vol. 21, pp. 372–375, 2002.

[5] Kim-Kwang Raymond Choo, "Revisit of McCullagh-Barreto two-party id-based authenticated key agreement protocols," *International Journal of Network Security*, vol. 1, no. 3, pp. 154–160, 2005.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[7] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469–472, July 1985.

[8] A. Jr. Evans, W. Kantrowitz, and E. Weiss, "A user authentication scheme not requiring secrecy in the computer," *Communications of the ACM*, vol. 17, pp. 437–442, August 1974.

[9] N. Haller, "The S/KEY (TM) one-time password system," in *Proceedings of Internet Society Symposium on Network and Distributed System Security*, pp. 151–158, 1994.

[10] N. Haller, "The S/KEY one-time password system," *RFC1760*, Feb. 1995.

[11] Min-Shiang Hwang, "Cryptanalysis of remote login authentication scheme," *Computer Communications*, vol. 22, no. 8, pp. 742–744, 1999.

[12] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *International Journal of Informatica*, vol. 12, no. 2, pp. 297–302, 2001.

[13] Min-Shiang Hwang, Cheng-Chi Lee, and Yuan-Liang Tang, "A simple remote user authentication scheme," *Mathematical and Computer Modelling*, vol. 36, pp. 103–107, 2002.

[14] Min-Shiang Hwang and Li-Hua Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[15] J. K. Jan and Y. Y. Chen, "'paramita wisdom' password authentication scheme without verification tables," *The Journal of Systems and Software*, vol. 42, pp. 45–57, 1998.

[16] R. Joyce and G. Gupta, "Identity authentication based on keystroke latencies," *Communications of the ACM*, vol. 33, pp. 168–176, Feb. 1990.

[17] H. J. Kim, "Biometrics, is it a viable proposition for identity authentication and access control," *Computers & Security*, vol. 14, pp. 205–214, 1995.

[18] M. Kim and Ctin Kaya Koc, "A simple attack on a recently introduced hash-based strong-password authentication scheme," *International Journal of Network Security*, vol. 1, no. 2, pp. 77–80, 2005.

[19] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770–772, November 1981.

[20] C. C. Lee, "Two attacks on the Wu-Hsu user identification scheme," *International Journal of Network Security*, vol. 1, no. 3, pp. 147–148, 2005.

[21] C. L. Lin, H. M. Sun, and T. Hwang, "Attacks and solutions on strong-password authentication," *IEICE Transactions on Communications*, vol. E84-B, pp. 2622–2627, September 2001.

[22] C. J. Mitchell and L. Chen, "Comments on the S/KEY user authentication secheme," *ACM Operating Systems Review*, vol. 30, pp. 12–16, Oct. 1996.

[23] National Bureau of Standard, *Data Encryption Standard*. NBS: FIPS, 1977.

[24] NIST. "Secure hash standard,". Tech. Rep. FIPS 180-1, NIST, US Department Commerce, April 1995.

[25] NIST. "Advanced encryption standard,". Tech. Rep. FIPS 197, NIST, US Department Commerce, Nov. 2001.

[26] G. B. Purdy, "A high security log-in procedure," *Communications of the ACM*, vol. 17, pp. 442–445, Aug. 1974.

[27] R. Rivest. "The MD5 message digest algorithm,". Tech. Rep. RFC 1321, IETF, April 1992.

[28] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[29] M. Sandirigama, A. Shimizu, and M. T. Noda, "Simple and secure password authentication protocol (sas)," *IEICE Transactions on Communications*, vol. E83-B, pp. 1363–1365, June 2000.

[30] Jau-Ji Shen, Chih-Wei Lin, and Min-Shiang Hwang, "A modified remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 414–416, 2003.

[31] Jau-Ji Shen, Chih-Wei Lin, and Min-Shiang Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Computers & Security*, vol. 22, no. 7, pp. 591–595, 2003.

[32] A. Shimizu, "A dynamic password authentication method by one-way function," *IEICE Transactions on Information and System*, vol. J73-D-I, pp. 630–636, July 1990.

[33] A. Shimizu, T. Horioka, and H. Inagaki, "A password authentication method for contents communication on the Internet," *IEICE Transactions on Communications*, vol. E81-B, pp. 1666–1763, Aug. 1998.

[34] Hung-Min Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958–961, 2000.

[35] B. Wang, J. H. Li, and Z. P. Tong, "Cryptanalysis of an enhanced timestamp-based password authentication scheme," *Computers & Security*, vol. 22, no. 7, pp. 643–645, 2003.

[36] Hsien-Chu Wu, Chi-Yu Liu, and Shu-Fen Chiou, "Cryptanalysis of a secure one-time password authentication scheme with low-communication for mobile communications," *International Journal of Network Security*, vol. 1, no. 2, pp. 74–76, 2005.

[37] T. C. Wu, "Remote login authentication scheme based on a geometric approash," *Computer Communications*, vol. 18, no. 12, pp. 959–963, 1995.

[38] S. Yamaguchi, K. Okayama, and H. Miyahara, "Design and implementation of an authentication system in WIDE Internet environment," in *Proceedings of IEEE Region Conference on Computer and Communication System*, 1990.

[39] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727–733, 1999.

[40] Cheng-Ying Yang, Cheng-Chi Lee, and Shu-Yin Hsiao, "Man-in-the-middle attack on the authentication of the user from the remote autonomous object," *International Journal of Network Security*, vol. 1, no. 2, pp. 81–83, 2005.

[41] Tzu-Chang Yeh, Hsiao-Yun Shen, and Jing-Jang Hwang, "A secure one-time password authentication scheme using smart cards," *IEICE Trans. on Communications*, vol. E85-B, pp. 2515–2518, Nov. 2002.