

Short Paper

Efficient Three-Party Authentication and Key Agreement Protocols Resistant to Password Guessing Attacks

HER-TYAN YEH, HUNG-MIN SUN* AND TZONELIH HWANG**

Department of Information and Communication

Southern Taiwan University of Technology

Tainan, 710 Taiwan

E-mail: htyeh@mail.stut.edu.tw

**Department of Computer Science*

National Tsing Hua University

Hsinchu, 300 Taiwan

E-mail: hmsun@cs.nthu.edu.tw

***Department of Computer Science and Information Engineering*

National Cheng Kung University

Tainan, 701 Taiwan

Three-party EKE was proposed to establish a session key between two clients through a server. However, three-party EKE is insecure against undetectable on-line and off-line password guessing attacks. In this paper, we first propose an enhanced three-party EKE to withstand the security risk in three-party EKE. We also propose a verifier-based three-party EKE that is more secure than a plaintext-equivalent mechanism in which a compromise of the server's database will not result in success in directly impersonating clients.

Keywords: network protocol, authentication, key agreement, password guessing attack, perfect forward secrecy

1. INTRODUCTION

Password authentication is regarded as one of the simplest and most convenient authentication mechanisms. However, people usually choose easy-to-remember passwords making them vulnerable to password guessing attacks. Protocols designed to provide mutual authentication and key exchange, which are secure against password guessing attacks, are called Password Authenticated Key Exchange protocols.

Password guessing attacks can be classified into three types:

- Detectable on-line password guessing attacks: This attack is unavoidable. It requires participation of the authentication server. A failed guess will be detected and logged

by the server. All of the password-based protocols accept this kind of attack; it can be prevented by letting the server take appropriate intervals between invalid trials.

- Undetectable on-line password guessing attacks: An attacker attempts to use a guessed password in an on-line transaction. He verifies the correctness of his guess using responses of the server. If his guess fails, he must start a new transaction with the server using another guessed password. A failed guess cannot be detected and logged by the server and the server is not able to distinguish an honest request from a malicious request.
- Off-line password guessing attacks: An attacker eavesdrops on authentication messages and stores them locally. He or she tries to find weak secret passwords and the verification is processed off-line, so that the server cannot detect the off-line password guessing attacks.

Obviously undetectable on-line password guessing attacks and off-line password guessing attacks are the most important considerations in designing a password-based authentication scheme.

In addition to password guessing attacks, replay attacks and perfect forward secrecy are also taken into consideration in designing such a protocol. In replay attack, an adversary tries to replay messages, partial or complete, obtained in previous communications. If he can impersonate another user or expose secrets that are sensitive and useful for further deceptions by using guessing attacks, known-plaintext attacks or other cryptographic analysis methods, then the protocol is said to be vulnerable to replay attacks. Furthermore, the meaning of perfect forward secrecy is that revealing the password to an attacker doesn't help him obtain the session keys of previous sessions. It also means that a stolen session key does not help an attacker carry out a brute-force guessing attack on the password. We consider a more secure level and add the condition that revealing the private key of the trusted server to an attacker still does not help him obtain the session keys of previous sessions. We call our method extended perfect forward secrecy.

In the most general form, two communicating parties, usually a client and a server, authenticate each other through their shared secret password. Such a scheme is called a plaintext-equivalent mechanism. A system that uses a plaintext-equivalent mechanism becomes instantly compromised once the password database in the server is revealed, since every user's password is stored there. Recently, additional work, referred to as verifier-based protocol, has been done to extend the protocols to address the issue of holding plaintext-equivalent data in the password file. These protocols only require a verifier to be stored in a server's database and hence add another method to verify the client's possession of the actual password, as opposed to a stolen verifier from the password file.

From the viewpoint of the session key creation, these protocols can be classified into two flavors: key transfer protocols and key agreement protocols. In a three-party (a server and two clients) setting, key transfer protocol means that the session key is created by the server and securely transmitted to these two clients, while key agreement protocol means that both clients contribute information to derive the common session key. Compared with key-transfer protocols, the latter (key agreement protocols) are fairer and more secure.

In recent years, a variety of protocols for authentication and key distribution have been proposed. In 1992, Bellare and Merritt [1] presented a key exchange protocol based on weakly chosen passwords between two communication parties known as the *Encrypted Key Exchange*, or EKE in short. Since then, a variety of authentication and key distribution protocols [2-16] based on weakly chosen passwords have been proposed and applied to many communication systems.

On the other hand, Gong, Lomas, Needham, and Saltzer [7] propose a third-party protocol, called the GLNS protocol. There are three principals (a server and two users) involved in the protocol. The server is responsible for user authentication and distributes the common session key shared between two users. Later, Steiner, Tsudik and Waidner proposed a three-party EKE [8] which extended the EKE to a three-party model where two clients registering in the same server can share a secret session key through the protocol. But the protocol cannot resist undetectable on-line [9] and off-line [10] password guessing attacks, i.e., one client can guess the other user's password on-line or off-line and the server cannot detect it. The main difference between GLNS protocol and three-party EKE is that GLNS protocol uses server's public key, but three-party EKE does not, and GLNS protocol belongs to key transfer protocols, but three-party EKE belongs to key agreement protocols.

All of the above protocols satisfy the property of perfect forward secrecy. But in the series of GLNS [7, 11-16], the users use the server's public key to encrypt some messages such as confounders or passwords and send it to the server. After decrypting the message and authenticating the users, the server then chooses the common session key and transfers it to the users encrypted by password or confounder. If the server's private key is revealed, the attacker can get the password directly or by using a guessing attack to acquire the password and gain the common session key. This breaks the extended perfect forward secrecy.

The aim of this paper is to propose two new secure authentication and key agreement protocols (plaintext-equivalent and verifier-based) to solve the above problems. First, a plaintext-equivalent authentication protocol is proposed to solve the security weakness (undetectable on-line and off-line password guessing attacks) in three-party EKE. Then, a verifier-based authentication protocol that is more secure than a plaintext-equivalent mechanism is proposed. Of course, they all also resist various attacks such as password guessing and replay attacks and provide extended perfect forward secrecy.

The remainder of this paper is organized as follows. In section 2, we briefly describe the notations used in this paper. In section 3, we review three-party EKE and analyze its security properties. In section 4, we propose an improved version of three-party EKE called enhanced three-party EKE. In section 5, we propose another practical authentication and key agreement protocol and then briefly analyze its security. In section 6, we compare our protocols with previous related protocols. In section 7, we give the conclusions.

2. NOTATIONS AND DEFINITIONS

For convenience, the notations and definitions used to describe the protocols in this paper are shown in Table 1.

Table 1. Notations and Definitions.

A, B	Clients
S	Trusted center or a server.
Pa, Pb	Passwords of A and B shared with S
(Spa, Vpa)	(private, public) key derived from Pa and shared with S
(Spb, Vpb)	(private, public) key derived from Pb and shared with S
Ks	Public key of the Trust Server
x, y, z, a, b, ra, rb, X	Random numbers.
K	Session key between A and B
$[info]_K$	Symmetric-key encryption of "info" with key K .
$\{info\}_K$	Asymmetric-key encryption of "info" with key K .
$\langle info \rangle_K$	Digital signature of "info" with key K
$h()$	One-way hash function.
$A \rightarrow B : M$	A sends a message M to B .
g	Base generator
P	The Modulus (all exponentiation in modulo P)
Fa	The first 64 bits of message 1 in proposed protocol

3. REVIEW OF THREE-PARTY EKE

3.1 Protocol of Three-party EKE [8]

The detailed steps are as follows:

- (1) A chooses a random number x and computes g^x . Then he transmits message 1 to B , where Pa is the password of A .
- (2) After receipt of message 1, B composes a similar message and sends it to S .
- (3) The trust center, S , checks the authenticity of both A and B , and then chooses a random number z . He computes g^{yz} , g^{xz} , and transmits message 3 to B . B gets the message and computes the session key $K = g^{xyz}$.
- (4) B produces a challenge value using the partial bit stream of message 1.
- (5) A produces and sends a response value.

As a result, A and B can authenticate each other and confirm that the session key is shared securely.

3.2 Security Analysis

• Replaying the request [9]

Replaying a request of A , attacker B completes with S the following attack. Fig. 2 illustrates the undetectable on-line password guessing attack.

- (1) When B receives message 1 in Fig. 1, he can guess Pa and gain (g^x) .
- (2) B sets $g^y = g^{x'}$, computes $[g^{x'} \oplus A]_{pb}$, and then transmits message 2 to S .
- (3) S responses with $\{g^{xz}, g^{yz}\}$ to B . Attacker B compares the two values. If $g^{x'z} = g^{xz}$, then B gets the correct password of A by this guess.

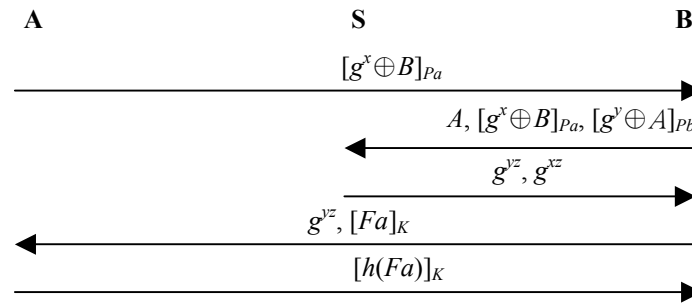


Fig. 1. Three-party EKE.

- | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. B : records $[g^x \oplus B]_{Pa}$ 2. B \rightarrow S : $A, [g^x \oplus B]_{Pa}, [g^y \oplus A]_{Pb}$ 3. S \rightarrow B : g^{xz}, g^{yz} |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fig. 2. Undetectable on-line password guessing attack on three-party EKE.

• **Impersonating the clients [9]**

B can impersonate A to guess his password without the participation of A. Fig. 3 illustrates the above undetectable on-line attack.

- (1) B guesses a password Pa' , computes $g^{x'}$ and $[g^{x'} \oplus B]_{Pa'}$.
- (2) B sends S message $\{A, [g^{x'} \oplus B]_{Pa'}, [g^y \oplus A]_{Pb}\}$
- (3) S responds with $\{g^{xz}, g^{yz}\}$ to B. Attacker B computes and compares the two values $\{(g^{xz})^y, (g^{yz})^{x'}\}$. If $(g^{xz})^y = (g^{yz})^{x'}$, then B gets the correct password of A by this guess.

- | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. B : $[g^{x'} \oplus B]_{Pa'}$ 2. B \rightarrow S : $A, [g^{x'} \oplus B]_{Pa'}, [g^y \oplus A]_{Pb}$ 3. S \rightarrow B : g^{xz}, g^{yz} |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fig. 3. Undetectable on-line password guessing attack on three-party EKE.

• **An Off-line password Guessing Attack [10]**

Fig. 4 illustrates an off-line guessing attack.

- | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 1. A* \rightarrow B : X 2. B \rightarrow S* : A, X, $[g^y \oplus A]_{Pb}$ 3. S* \rightarrow B : $g^{xz'}, Y$ 4. B \rightarrow A* : Y, $[flow1]_K$ |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Fig. 4. Off-line password guessing attack on three-party EKE.

- (1) An arbitrary attacker sends on behalf of A a random number X to B as a request, in which the length of X is the same as the length of $[g^x \oplus B]_{Pa}$. After receiving the re-

quest, B chooses a random number y , keeps it secret, computes g^y , and then encrypts $g^y \oplus A$ with his password Pb . B forwards the request X with the encrypted message to S .

- (2) The attacker intercepts the message being sent from B to S . He chooses three random numbers x' , z' and Y , computes $g^{x'}$ and responds $(g^{x'})^{z'}$, Y to B .
- (3) B computes the session key $K = ((g^{x'})^{z'})^y = g^{x'y'z'}$ and sends Y and a key confirmation message $[flow1]_K$ to A .
- (4) The attacker intercepts the message being sent from B to A . Hereafter, he can do off-line repeatedly guess a password Pb' , get the value $g^{y'}$ from $[g^y \oplus A]_{Pb}$, compute the session key $K' = (g^{y'z'})^{x'} = g^{x'y'z'}$, decrypt $[flow1]_K$ with the session key K' and check $flow1 ? = X$ until it is true.

4. ENHANCED THREE-PARTY EKE

4.1 The Proposed Protocol

In order to solve the above undetectable on-line and off-line password guessing attacks mentioned in section 3.2, we introduce the server's public key Ks and propose an improved version which is called enhanced three-party EKE. Fig. 5 illustrates the enhanced three-party EKE.

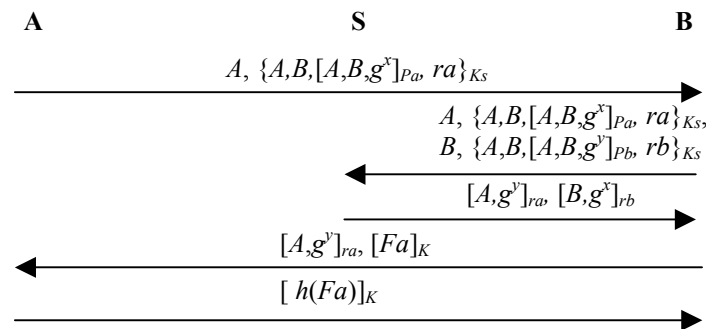


Fig. 5. Enhanced three-party EKE.

The following steps describe the procedure.

- (1) $A \rightarrow B : A, \{A, B, [A, B, g^x]_{Pa}, ra\}_{Ks}$
 A chooses a confounder ra , a random number x and computes g^x which is encrypted by A 's password Pa . Then A encrypts $A, B, [A, B, g^x]_{Pa}$, and ra by the server's public key Ks and sends the ciphertext to B .
- (2) $B \rightarrow S : A, \{A, B, [A, B, g^x]_{Pa}, ra\}_{Ks}, B, \{A, B, [A, B, g^y]_{Pb}, rb\}_{Ks}$
 After receiving A 's message, B also composes a similar message and sends it to S .
- (3) $S \rightarrow B : [A, g^y]_{ra}, [B, g^x]_{rb}$
 The trust center, S , can decrypts $\{A, B, [A, B, g^x]_{Pa}, ra\}_{Ks}, \{A, B, [A, B, g^y]_{Pb}, rb\}_{Ks}$ with his private key and checks the authenticities of both A and B . Then S encrypts A ,

g^y with ra , encrypts B , g^x with rb and sends them to B . Notice that the ra and rb also act as one-time keys.

- (4) $B \rightarrow A : [A, g^y]_{ra}, [Fa]_K$
 B decrypts $[B, g^x]_{rb}$ with rb and computes the session key $K = (g^y)^x$. Then he sends $[A, g^y]_{ra}$ together with a key confirmation message $[Fa]_K$ to A .
- (5) $A \rightarrow B : [h(Fa)]_K$
 A decrypts $[A, g^y]_{ra}$ with ra and computes the session key $K = (g^y)^x$. Then he decrypts $[Fa]_K$ with the session key K , checks the validity, and responds with $[h(Fa)]_K$ for key confirmation.

As a result, A and B can authenticate each other and confirm that the session key K is shared securely.

4.2 Security Analysis of Enhanced Three-Party EKE

• Password Guessing Attacks

The password Pa of A included in message 1 is only used to authenticate A 's status and that there is not any verifiable data included in other messages. If B knows only the private key of server S , he cannot decrypt message 1, so B does not have any chance to guess A 's password on line. Therefore, our scheme is free on-line password guessing attacks.

From Fig. 5, though we directly encrypt some messages with poorly chosen passwords, which appear only in message 1 and message 2, there is not any verifiable data included in other messages. Therefore, our protocol is immune to off-line password guessing attacks.

• Replay Attack

Now, the attacker pretends to be A or B and tries to get session key K or password Pa or Pb . This attack cannot succeed with our protocol. First, even if the intruder replays message 1 or message 2, because he does not know the server's private key for decrypting message 1 or message 2, he still cannot acquire Pa or Pb . Second, session key K is created by Diffie-Hellman key exchange. Even if the intruder gets g^x and g^y , the attackers still cannot obtain session key $K = g^{xy}$, because the difficulty is similar to solving the Diffie-Hellman problem [17]. Thus, our protocol is secure against message replay attacks.

• Extended Perfect Forward Secrecy

Extended perfect forward secrecy is defined in section 1 and we divide it into three parts. First, when the user's passwords are revealed, the attackers can know Pa and Pb . Because the attackers do not know the server's private key to decrypt message 1 and message 2, they cannot get g^x , g^y . Even if they get g^x and g^y , the attackers still cannot obtain session key $K = g^{xy}$, because the difficulty is similar to solving the Diffie-Hellman problem [17]. Therefore, the session key is still secure. Second, if the server's private key is leaked, an attacker can decrypt message 1, message 2 and get ra , rb directly and Pa , Pb , g^x , g^y by implementing password guessing attacks. If the attacker wants to get session key $K = g^{xy}$, the difficulty is equal to solving the Diffie-Hellman problem [17].

Thus, the session key is still secure. Third, if an attacker learned the previous session key K , he still has no idea what the passwords of A and B are. Because the messages in Fig. 5 do not include any verifiable data, it is impossible to guess the passwords of A and B .

To sum up the above description, our scheme provides the extended perfect forward secrecy.

5. VERIFIER-BASED THREE-PARTY EKE

Although enhanced three-party EKE solves the security problems mentioned above, it is a plaintext-equivalent authenticated and key agreement protocol. In this section, we propose a verifier-based protocol called verifier-based three-party EKE.

5.1 The Proposed Protocol

Fig. 6 illustrates the verifier-based three-party EKE whose detailed steps are as follows:

- (1) In step 1, A generates the message as we introduced in Fig. 6 and sends it to S via B . B is a legitimate counterpart of A .
- (2) In step 2, B composes a similar message and sends it to S . The trust center, S , can check the authenticity of both A and B .
- (3) In step 3, server S uses the ra and rb generated by A and B as encrypting keys, replies with message 3 as we described in Fig. 6. B gets the message and computes the session key $K = g^{xy}$.
- (4) In step 4, B produces a challenge value using the partial bit stream of message 1.
- (5) In step 5, A produces and sends a response value.

As a result, A and B can authenticate each other and confirm that the session key is shared securely.

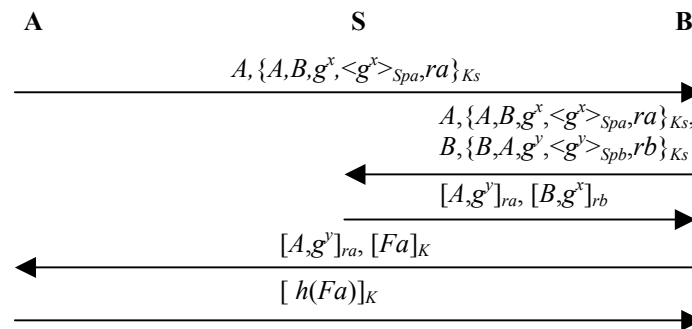


Fig. 6. Verifier-based three-party EKE.

5.2 Security Analysis of Verifier-based Three-Party EKE

• Password Guessing Attacks

The secret key Spa derived from password Pa included in message 1 is used only to authenticate A 's status; there is not any verifiable data included in other messages. If B only knows the private key of server S , he cannot decrypt message 1, and so B does not have chance to guess A 's password Pa on-line. Therefore, our scheme is immune from on-line password guessing attacks.

From Fig. 6, though we directly encrypt some messages with private keys derived from poorly chosen passwords, they only appear in message 1 and message 2, and there is not any verifiable data included in other messages. Therefore, our protocol is immune from off-line password guessing attacks.

• Replay Attack

Now, the attacker wants to pretend to be A or B to get session key K or password Pa or Pb . This attack is not possible with our protocol. First, even if the intruder replays message 1 or message 2, because he does not know the server's private key for decrypting message 1 or message 2, he still cannot acquire the password Pa or Pb . Second, the session key K is created by Diffie-Hellman key exchange. Even if the intruder gets g^x and g^y , the attackers still cannot obtain the session key $K = g^{xy}$, because the difficulty is similar to solving the Diffie-Hellman problem [17]. Thus, our protocol is secure against the message replay attacks.

• Extended Perfect Forward Secrecy

First, when the user's passwords are revealed, an attacker can know two pairs (Spa, Vpa) , (Spb, Vpb) . Because the attacker doesn't know the server's private key to decrypt message 1 and message 2, he cannot get g^x , g^y . Even if the attacker does get g^x and g^y , he still cannot obtain the session key $K = g^{xy}$, because the difficulty is similar to solving the Diffie-Hellman problem [17]. Therefore, the session key is still secure. Second, if the server's private key is leaked, an attacker can get ra , rb by decrypting message 1 and message 2 and Pa , Pb , g^x , g^y by implementing password guessing attacks. If the attacker wants to get session key $K = g^{xy}$, the difficulty is equal to solving the Diffie-Hellman problem [17]. Thus, the session key is still secure. Third, if an attacker learned the previous session key K , he still has no idea how to know the passwords of A and B . Because the messages in Fig. 6 do not include any verifiable data, it is impossible to guess the passwords of A and B .

To sum up the above description, our scheme provides the extended perfect forward secrecy.

6. COMPARISONS

In this section, we compare the enhanced three-party EKE and verifier-based three-party EKE with the related protocols. The comparisons are listed in Table 2.

Optimal GLNS protocol and Improved Three Way KIP do not provide extended perfect forward secrecy and belong to key transfer protocols. three-party EKE is not

Table 2. Comparisons with Related Protocols.

	# of steps	Password /Verifier - based	Key Transfer or agreement	Extended Perfect forward secrecy	Need server's public key	Resist to guessing Attacks	# of Random Numbers	Computational cost	
								# of Exponential	# of Asymmetric Encryption
Optimal GLNS protocol	5	Password	Transfer	No	Yes	Yes	10	0	2
Improved Three-Way KIP	5	Password	Transfer	No	Yes	Yes	5	0	2
three-party EKE	5	Password	Agreement	Yes	No	No	3	6	0
Enhanced three-party EKE	5	Password	Agreement	Yes	Yes	Yes	4	4	2
Verifier-based three-party EKE	5	Verifier	Agreement	Yes	Yes	Yes	4	4	4

secure against password guessing attacks. Compared with other related protocols, our two proposed protocols are fairer (replaced key transfer protocols with key agreement protocols) and more secure (defeat various attacks and provide extended perfect forward secrecy).

7. CONCLUSIONS

Two efficient three-party authentication and key agreement protocols (enhanced three-party EKE and verifier-based three-party EKE) have been proposed in this paper to establish a session key between two users through an authentication server. The enhanced three-party EKE is designed to repair the security weaknesses in three-party EKE. A verifier-based three-party EKE that is more secure than a plaintext-equivalent mechanism is proposed to provide the property that a compromise of the server's database will not result in successfully impersonating clients.

REFERENCES

1. S. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings of IEEE Symposium on Research in Security and Privacy*, 1992, pp. 72-84.
2. S. Bellovin and M. Merritt, "Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise," *AT&T Bell Laboratories*, 1993.
3. D. Jablon, "Strong password-only authentication key exchange," *Computer Communication Review*, Vol. 26, 1996, pp. 5-26.

4. D. Jablon, "Extended password key exchange protocols immune to dictionary attack," in *Proceedings of the WETICE Workshop on Enterprise Security*, 1997, pp. 248-255.
5. S. Lucks, "Open key exchange: how to defeat dictionary attacks without encrypting public keys," in *Proceedings of the Security Protocol Workshop '97*, 1997, pp. 79-90.
6. T. Wu, "The secure remote password protocol," *Internet Society Symposium on Network and Distributed System Security*, 1998, pp. 97-111.
7. L. Gong, M. Lomas, R. Needham, and J. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *IEEE Journal on Selected Areas in Communications*, Vol. 11, 1993, pp. 648-656.
8. M. Steiner, G. Tsudik, and M. Waidner, "Refinement and extension of encrypted key exchange," *Operating Systems Review*, Vol. 29, 1995, pp. 22-30.
9. Y. Ding and P. Horster, "Undetectable on-line password guessing attacks," *Operating System Review*, Vol. 29, 1995, pp. 77-86.
10. C. L. Lin, H. M. Sun, and T. Hwang, "Three-party encrypted key exchange: attacks and a solution," *ACM Operating Systems Review*, Vol. 34, 2000, pp. 12-20.
11. L. Gong, "Optimal authentication protocols resistant to password guessing attacks," in *Proceedings of the 8th IEEE Computer Security Foundation Workshop*, 1995, pp. 24-29.
12. T. Kwon, M. Kang, and J. Song, "An adaptable and reliable authentication protocol for communication networks," in *Proceedings of IEEE INFOCOM 97*, 1997, pp. 738-745.
13. T. Kwon and J. Song, "Efficient key exchange and authentication protocol protecting weak secrets," *IEICE Transactions on Fundamentals*, Vol. E81-A, 1998, pp. 156-163.
14. T. Kwon, M. Kang, S. Jung, and J. Song, "An improvement of the password-based authentication protocol (K1P) on security against replay attacks," *IEICE Transactions on Communications*, Vol. E82-B, 1999, pp. 991-997.
15. S. Keung and K. Siu, "Efficient protocols secure against guessing and replay attacks," in *Proceedings of the Fourth International Conference on Computer Communications and Networks*, 1995, pp. 105-112.
16. T. Kwon and J. Song, "Authentication key exchange protocols resistant to password guessing attacks," *IEE Communications*, Vol. 145, 1998, pp. 304-308.
17. W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-11, 1976, pp. 644-654.

Her-Tyan Yeh (葉禾田) was born in Tainan, Taiwan, in 1965. He received his B.S. degree in Information Science from Soochow University in 1987, his M.S. degrees in Computer Science and Information Engineering from National Taiwan University in 1996, and his Ph.D. degree in Computer Science and Information Engineering from National Cheng Kung University in 2003, respectively. Currently, he is an assistant professor with the Department of Information and Communication, Southern Taiwan University of Technology. His research interests include cryptography, network security and computer communication.

Hung-Min Sun (孫宏民) received his B.S. degree in Applied Mathematics from National Chung-Hsing University in 1988, his M.S. degree in Applied Mathematics from National Cheng Kung University in 1990, and his Ph.D. degree in Computer Science and Information Engineering from National Chiao-Tung University in 1995, respectively. He was an associate professor with the Department of Information Management, Chaoyang University of Technology from 1995 to 1999, and the Department of Computer Science and Information Engineering, National Cheng Kung University from 1999 to 2002. Currently, he is an associate professor with the Department of Computer Science, National Tsing Hua University. He has published over seventy papers. He was the program chair of 2001 National Information Security Conference and the program committee member of 1997 Information Security Conference, 2000 Workshop on Internet and Distributed Systems, Workshop on the 21st Century Digital Life and Internet Technologies, 1998 and 1999 National Conference on Information Security. His research interests include cryptography, information theory, network security and image compression.

Tzonelih Hwang (黃宗立) was born in Tainan, Taiwan, R.O.C. in March, 1958. He received his undergraduate education from National Cheng Kung University, Taiwan, R.O.C. in 1980 and the M.S. and Ph.D. degrees in Computer Science from the University of Southwestern Louisiana U.S.A. in 1988. He is presently a professor of the Department of Information Engineering, National Cheng Kung University, Taiwan, R.O.C.. His research interests include cryptology, network security and coding theory. Dr. Hwang is a member of IEEE and also a member of International Association for Cryptographic Research.