Computers and Mathematics with Applications 58 (2009) 641–648

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Communication-efficient three-party protocols for authentication and key agreement

Tian-Fu Lee [a], Jenn-Long Liu [b,*], Mei-Jiun Sung [c], Shiueng-Bien Yang [d], Chia-Mei Chen [b]

[a] Department of Information Management, Leader University, Tainan 709, Taiwan, ROC
[b] Department of Information Management, National Sun Yat-Sen University, Kaohsiung 804, Taiwan, ROC
[c] Institute of Applied Information, Leader University, Tainan 709, Taiwan, ROC
[d] Department of Computer Science and Information Engineering, Leader University, Tainan 709, Taiwan, ROC

ARTICLE INFO

Article history:
Received 22 November 2007
Received in revised form 16 February 2009
Accepted 27 February 2009

Keywords:
Authentication
Key agreement
EKE
Network communication
Diffie–Hellman

ABSTRACT

Encrypted key exchange (EKE) authentication approaches are very important for secure communicating over public networks. In order to solve the security weaknesses three-party EKE, Yeh et al. [H.T. Yeh, H.M. Sun, T. Hwang, Efficient three-party authentication and key agreement protocols resistant to password guessing attacks, Information Science and Engineering 19 (6) (2003) 1059–1070.] proposed two secure and efficient three-party EKE protocols. Based on the protocol developed by Yeh et al., two improved EKE protocols for authentication and key agreement are proposed in this study. The computational costs of the proposed protocols are the same as those of the protocols of Yeh et al. However, the numbers of messages in the communication are fewer than those of the protocols of Yeh et al. Furthermore, the round efficient versions of our proposed protocols are also described.

1. Introduction

Encrypted key exchange authentication approaches are very important and widely applied in network communication [1–6]. In these authentication approaches, the communicating parties share a secrete key with a trusted server and can exchange confidential and authenticated information over an insecure network with the help of a server.

Secure communication protocols should provide security requirements, which are described as follows.

1. Mutual authentication: Mutual authentication protocols enable participants mutually to authenticate each other's identity.
2. Session key security: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.
3. Off-line guessing attacks: In an off-line password guessing attack, an attacker guesses a password and verifies his guess, but he does not need to participate in any communication during the guessing phase [4,5,7–11].
4. Undetectable on-line guessing attacks: In an on-line guessing attack, an attacker searches to verify a guessed password in an on-line transaction and a failed guess cannot be detected and logged by the server [12,13].
5. Perfect forward secrecy: A protocol has perfect forward secrecy if the compromise of long-life keys does not compromise previous session keys.

* Corresponding author.
E-mail address: linglong@ms23.hinet.net (J.-L. Liu).

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.02.039

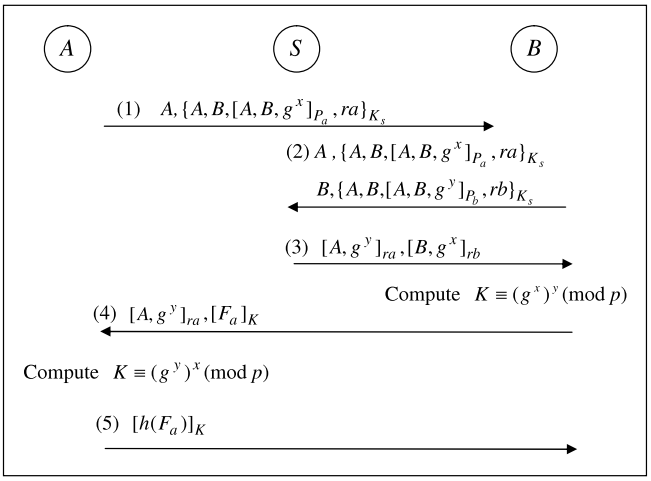

Fig. 1. The three-party EKE protocol of Yeh et al..

6. Known-key security: If a protocol has known-key security, the compromise of one session key does not reveal other session keys.
7. Intruder-in-the-middle attacks: The intruder-in-the-middle attack is that an intruder interrupts a message and substitutes it for his own message, so that communication parties compute the wrong session key without detecting the intruder.
8. Replay attacks: A replay attack is that an unauthorized party records previous messages and uses them to cheat the receiver in later processes [14].
9. Stolen-verifier attacks: The stolen-verifier attack is that the adversary who has stolen the verifier of a user's password can impersonate as that user.

Recently, many encrypted key exchange authentication approaches were presented for secure communicating over public networks. Bellovin and Merritt [1] proposed a key exchange protocol based on memorable passwords between two communicating parties known as the Encrypted Key Exchange (EKE) in 1992. From then on, numbers of password-based authentication and key agreement approaches have been proposed and developed in many communication systems. Gong et al. [9] proposed a three party authentication protocol, in which the server is responsible for user authentication and distributes the common session key shared between two users. Later, Steiner et al. proposed a three-party EKE [2] 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. However, Lin et al. [13] and Yeh et al. [15] stated the weaknesses of three-party EKE proposed by Steiner et al. Besides, Yeh et al. [15] proposed two new secure three-party EKE protocols for authentication and key agreement. One is a plaintext-equivalent authentication protocol and the other is a verifier-based authentication protocol. In a plaintext-equivalent protocol, a client and a server usually authenticate each other through their shared secret password. A system using a plaintext-equivalent protocol becomes instantly compromised if the server's database containing clients' passwords is revealed. Comparatively, a verified-based protocol only requires a verifier to be stored in server's database and hence add another method to verify the client's possession of actual password, as opposed to a stolen verifier from the password file [15].

In a Diffie-Hellman-based authentication protocol, only revealing the information g^x and g^y for generating the session key (g^{xy}) cannot compromise the session key. The clients can directly exchange the information g^x and g^y for generating the session key without the help of the server, and thus the improved protocols can decrease the messages in communications. Using this technique, this investigation proposes communication-efficient three-party EKE protocols based on the protocols of Yeh et al. The computation costs of the protocols developed by Yeh et al. and the proposed protocols herein are equal. However, the numbers of messages in the proposed protocols are reduced. Moreover, the round efficient versions of these protocols are presented.

The rest of this paper is organized as follows. Section 2 reviews the three-party EKE protocols of Yeh et al. Section 3 introduces the improved three-party EKE protocols. Section 4 provides security and performance analyses of the improved protocols. Finally, Section 5 gives the conclusions.

2. Preliminaries

This section describes the underlying primitives used in this paper and briefly reviews the three-party EKE protocols of Yeh et al. [15]. The underlying primitives include the Discrete Logarithms problem and the Diffie-Hellman assumptions [16]. Table 1 details the notation used, and Fig. 1 depicts the three-party EKE protocol of Yeh et al., which works as follows.

Table 1
Notation.

A, B, S	A and B are two communication parties and S is a trusted server
P_a, P_b	Passwords of A and B shared with S
(S_{pa}, V_{pa})	(private, public) key derived from P_a and shared with S
(S_{pb}, V_{pb})	(private, public) key derived from P_b and shared with S
K_s	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)
F_a	The first 64 bits of message 1 in proposed protocol

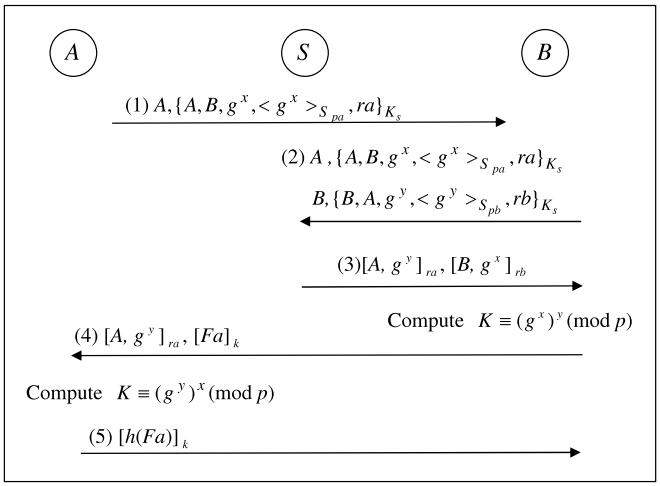

Fig. 2. The verifier-based three-party EKE of Yeh et al..

Definition 2.1 (*The Discrete Logarithms Problem (DLP)*). Let $G = \langle g\rangle$ be a cyclic group of prime order q and x is randomly chosen in Z_q. Consider the equation $y = g^x \bmod p$. Given g, x and p, it is a straightforward matter to calculate y. However, given y, g and p, it is computationally infeasible to calculate x.

Definition 2.2 (*The Diffie-Hellman Assumptions*). Let $G = \langle g\rangle$ be a cyclic group of prime order q and x, y are randomly chosen in Z_q. Given g, g^x, g^y and p, if the equation $z = g^{xy} \bmod p$ holds, then it is computationally infeasible to calculate z without the value of x or y.

2.1. The three-party EKE protocol of Yeh et al.

Step 1. $A \rightarrow B$: $A, \{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$
User A selects two random numbers ra, x, and computes $g^x \pmod p$ which is encrypted by password P_a of A. Then A encrypts $(A, B, [A, B, g^x]_{P_a}, ra)$ by the public key K_s of server and sends the ciphertext to B.

Step 2. $B \rightarrow S$: $A, \{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}, B, \{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$
On receiving the message from A, B composes a similar message to A and sends them to S.

Step 3. $S \rightarrow B : [A, g^y]_{ra}, [B, g^x]_{rb}$
Server S decrypts $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$, $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$ 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.

Step 4. $B \rightarrow A : [A, g^y]_{ra}, [F_a]_K$
B decrypts $[B, g^x]_{rb}$ with rb and computes the session key $K \equiv (g^x)^y \pmod p$. Then he sends a key confirmation message $[F_a]_K$ and forwards $[A, g^y]_{ra}$ to A.

Step 5. $A \rightarrow B : [h(F_a)]_K$
A decrypts $[A, g^y]_{ra}$ withra and computes the session key $K \equiv (g^y)^x \pmod p$. Then he decrypts $[F_a]_K$ with the session key K, checks its validity, and responds with $[h(F_a)]_K$ for key confirmation.

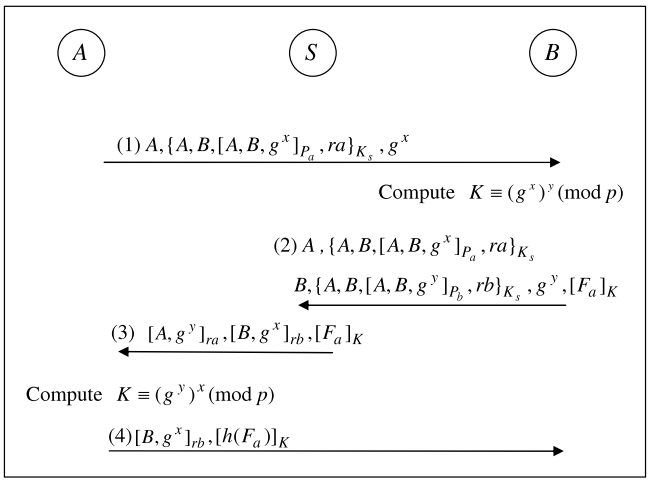

Fig. 3. Proposed efficient three-party EKE protocol.

2.2. The verifier-based three-party EKE protocol of Yeh et al. (shown on Fig. 2)

Step 1. $A \rightarrow B : A, \{A, B, g^x, \langle g^x \rangle_{S_{\text{pa}}}, ra\}_{K_s}$
A generates the message $A,\{A, B, g^x, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ and sends it to S via B. B is a similar legitimate user to A.
Step 2. $B \rightarrow S : A, \{A, B, g^x, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}, B, \{B, A, g^y, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$
B composes a similar message and sends it to S. Server S checks the authenticity of both A and B.
Step 3. $S \rightarrow B : [A, g^y]_{ra}, [B, g^x]_{rb}$
Server S uses the ra and rb generated by A and B, respectively, as encrypting keys replies with message $[A, g^y]_{ra}$,$[B, g^x]_{rb}$.
Step 4. $B \rightarrow A : [A, g^y]_{ra}, [F_a]_K$
B gets the messages and computes the session key $K \equiv (g^x)^y \pmod p$. He then sends $[A, g^y]_{ra}$, $[F_a]_K$ to A, where F_a is produced by the partial bit stream of $(A, \{A, B, g^x, \langle g^x \rangle_{S_{Pa}}, ra\}_{K_s})$.
Step 5. $A \rightarrow B : [h(F_a)]_K$
Similarly, A gets the messages and computes the session key $K \equiv (g^y)^x \pmod p$. If A successfully confirms $[F_a]_K$, then he uses a hash value as a response value to B. Finally, B verifies the key confirmation message $[h(F_a)]_K$.

3. Communication-efficient three-party EKE protocols

This section presents two communication-efficient authentication and key agreement protocols based on the protocol of Yeh et al. [15]. One is a plaintext-equivalent authentication protocol and the other is a verifier-based authentication protocol. The computation costs of the protocols developed by Yeh et al. are equal to those of the protocols proposed herein. However, the number of transmissions is fewer than those of the protocols of Yeh et al. Figs. 3 and 4 illustrate the proposed authentication and key agreement protocol and verifier-based three-party EKE protocols, respectively. The protocols are described in detail as follows.

3.1. Proposed efficient three-party EKE protocol

Step 1. $A \rightarrow B : A, \{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}, g^x$
A selects two random numbers ra, x, and computes $g^x \pmod p$ which is encrypted by password P_a of A. Then he encrypts $(A, B, [A, B, g^x]_{P_a}, ra)$ by the public key K_s of the server and sends it with g^x to B.
Step 2. $B \rightarrow S : A, \{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}, B, \{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}, g^y, [F_a]_K$
On receiving the message from A, B composes a similar message to A and gets g^x to produce the session key $K \equiv (g^x)^y \pmod p$. Then B sends the message to S.
Step 3. $S \rightarrow A : [A, g^y]_{ra}, [B, g^x]_{rb}, [F_a]_K$
Server S decrypts $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$ and $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$ 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 with a key confirmation message $[F_a]_K$ to A.
Step 4. $A \rightarrow B : [B, g^x]_{rb}, [h(F_a)]_K$
A decrypts $[A, g^y]_{ra}$ with ra and computes the session key $K = (g^y)^x$. Then he sends $[h(F_a)]_K$ and forwards $[B, g^x]_{rb}$ to B. Finally, B validates g^x by checking and decrypting $[B, g^x]_{rb}$ with rb.

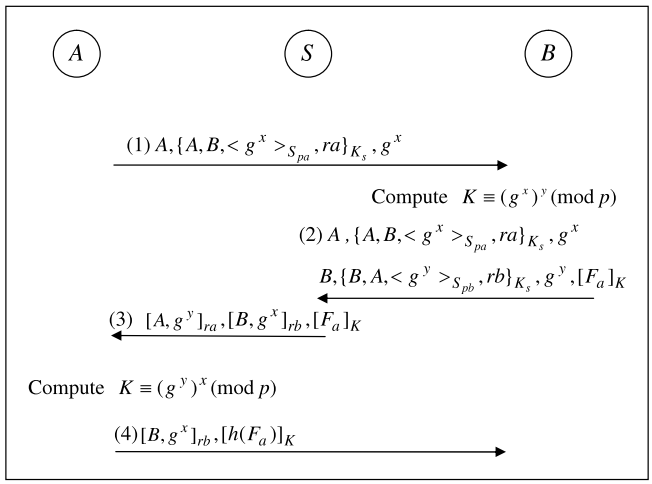

Fig. 4. Proposed efficient verifier-based three-party EKE protocol.

By rearranging and sending the messages in parallel, the proposed efficient three-party EKE protocol can be implemented in less time and obtain a round-efficient version. The round-efficient protocol can be executed in three rounds and is described below.

Round 1. $A \to B : A, g^x$
$A \to S : A, \{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$.
Round 2. $B \to A : B, g^y, [F_a]_K$
$B \to S : B, \{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$
Round 3. $S \to A : [A, g^y]_{ra}$
$S \to B : [B, g^x]_{rb}$
$A \to B : [h(F_a)]_K$.

3.2. Proposed efficient verifier-based three-party EKE protocol

Step 1. $A \to B : A, \{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}, g^x$
A generates *ra*, *x*; computes $g^x \pmod p$, $\langle g^x \rangle_{S_{pa}}$ and sends the message $A, \{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}, g^x$ to *S* via *B*. *B* is a similar legitimate user to *A*.
Step 2. $B \to S : A, \{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}, g^x, B, \{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}, g^y, [F_a]_K$
B gets g^x to compute the session key $K \equiv (g^x)^y \pmod p$. Then he composes a similar message and sends it to *S*. Server *S* checks the authenticity of both *A* and *B*.
Step 3. $S \to A : [A, g^y]_{ra}, [B, g^x]_{rb}, [F_a]_K$
Server *S* decrypts $\{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ and $\{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$ with his private key for checking the authenticities of both *A* and *B*. Then *S* encrypts (A, g^y) with *ra*; encrypts (B, g^x) with *rb*; and sends them with a key confirmation message $[F_a]_K$ to *A*.
Step 4. $A \to B : [B, g^x]_{rb}, [h(F_a)]_K$
A decrypts $[A, g^y]_{ra}$ with *ra* and computes the session key $K \equiv (g^y)^x \pmod p$. Then he sends $[h(F_a)]_K$ and forwards $[B, g^x]_{rb}$ to *B*. Finally, *B* validates g^x by checking and decrypting $[B, g^x]_{rb}$ with *rb*.

Similarly, a round-efficient version of the proposed verifier-based protocol can be obtained by rearranging and sending the messages in parallel. The round-efficient protocol can also be executed in three rounds and is described below.

Round 1. $A \to B : A, g^x$
$A \to S : A, \{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$
Round 2. $B \to A : B, g^y, [F_a]_K$
$B \to S : B, \{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$
Round 3. $S \to A : [A, g^y]_{ra}$
$S \to B : [B, g^x]_{rb}$
$A \to B : [h(F_a)]_K$.

4. Security and performance analyses

This section describes the security analyses of the proposed three-party EKE protocols and compares their performance with that of the three-party EKE protocols of Yeh et al.

4.1. Security analyses

4.1.1. Security analyses of the improved three-party EKE protocol

1. Mutual authentication
   On receiving $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$ and $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$ in Step 2, S can authenticate A and B by verifying P_a and P_b, respectively. Additionally, A and B can validate g^y and g^x, and authenticate S simultaneously by decrypting $[A, g^y]_{ra}$ with ra and $[B, g^x]_{rb}$ with rb in Step 3 and Step 4, respectively. Hence, the proposed protocol exhibits mutual authentication.
2. Session key security
   Given g^x (or g^y), the value x (or y) is hard to obtain, because of the Discrete Logarithm problem. Given g^x and g^y, the session key $K \equiv (g^y)^x \pmod p$ or $K \equiv (g^x)^y \pmod p$ cannot be determined without knowledge of x or y, because of the Diffie-Hellman problem.
3. Off-line guessing attacks
   In the proposed protocol, ra, g^x, rb and g^y are random numbers and K_S is the server public key, so no information can be used to verify directly the correctness of the guessed passwords from $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$ in Step 1 and $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$ in Step 2. Hence, the proposed protocol resists off-line guessing attacks.
4. Undetectable on-line guessing attacks
   In the proposed protocol, on receiving messages from B in Step 2, server S authenticates A and B by verifying P_a and P_b in $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$ and $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$, respectively. Therefore, a failed guess will be detected by S. Hence, the proposed protocol can resist undetectable on-line guessing attacks.
5. Perfect forward secrecy
   In the improved protocol, the ephemeral exponents x and y are randomly selected and independent among each protocol execution. Therefore, the compromised passwords P_a and P_b cannot reveal any previous session keys $K \equiv g^{xy} \pmod p$. Therefore, the proposed protocol has perfect forward secrecy.
6. Known-key security
   The session keys $K \equiv g^{xy} \pmod p$ generated in different runs are independent since A and B randomly selects x and y, respectively, and these random numbers are independent among protocol executions. Hence, the proposed protocol exhibits known-key security.
7. Intruder-in-the-middle attacks
   Since the improved protocol has mutual authentication, the intruder-in-the-middle attacks are necessarily unsuccessful.
8. Replay attacks
   An attacker tries to deceive B (or A) and S by sending out a previous message $\{A, B, [A, B, g^x]_{P_a}, ra\}_{K_s}$ (or $\{A, B, [A, B, g^y]_{P_b}, rb\}_{K_s}$). The attacker cannot compute the session key $K \equiv g^{x \cdot y} \pmod p$ since he does not know the values of x and y. Accordingly, he cannot correctly send out a key confirmation message and the replay attack will fail.

4.1.2. Security analyses of the improved verifier-based three-party EKE protocol

1. Mutual authentication
   On receiving $\{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ and $\{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$ in Step 2, S can authenticate A and B by verifying S_{pa} and S_{pb}, respectively. Additionally, A and B can validate g^y and g^x, and authenticate S simultaneously by decrypting $[A, g^y]_{ra}$ with ra and $[B, g^x]_{rb}$ with rb in Step 3 and Step 4, respectively. Hence, the proposed protocol exhibits mutual authentication.
2. Session key security
   Given g^x (or g^y), the value x (or y) is hard to obtain, because of the Discrete Logarithm problem. Given g^x and g^y, the session key $K \equiv (g^y)^x \pmod p$ or $K \equiv (g^x)^y \pmod p$ cannot be determined without knowledge of x or y, because of the Diffie-Hellman problem.
3. Off-line guessing attacks
   In the proposed protocol, ra, g^x, rb and g^y are random numbers and K_S is the server public key, so no information can be used to verify directly the correctness of the guessed passwords from $\{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ and $\{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$. Hence, the proposed protocol resists off-line guessing attacks.
4. Undetectable on-line guessing attacks
   In this protocol, server S authenticates A and B by verifying S_{pa} and S_{pb} in $\{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ and $\{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$, respectively. Therefore, a failed guess will be detected by S. Hence, the proposed protocol can resist undetectable on-line guessing attacks.
5. Perfect forward secrecy
   In the improved protocol, the ephemeral exponents x and y are randomly selected and independent among each protocol execution. Therefore, the compromised password cannot reveal any previous session keys. Therefore, the proposed protocol has perfect forward secrecy.
6. Known-key security
   The session keys generated in different runs are independent since A and B randomly selects x and y, respectively, and these random numbers are independent among protocol executions. Hence, the proposed protocol exhibits known-key security.

Table 2
Performance comparisons with related protocols.

Protocols	Random numbers	Computational cost		Steps	Rounds
		Exponential	Asymmetric encryption		
Yeh et al.'s 3PEKE	4	4	2	5	–
Proposed 3PEKE	4	4	2	4	3
Yeh et al.'s VB-3PEKE	4	4	4	5	–
Proposed VB-3PEKE	4	4	4	4	3

7. Intruder-in-the-middle attacks
   Since the improved protocol has mutual authentication, the intruder-in-the-middle attacks are necessarily unsuccessful.
8. Replay attacks
   An attacker tries to deceive B (or A) and S by sending out a previous message $\{A, B, \langle g^x \rangle_{S_{pa}}, ra\}_{K_s}$ (or $\{B, A, \langle g^y \rangle_{S_{pb}}, rb\}_{K_s}$). The attacker cannot compute the session key $K \equiv g^{x \cdot y} (\text{mod } p)$ since he does not know the values of x and y. Accordingly, he cannot correctly send out a key confirmation message and the replay attack will fail.
9. Stolen-verifier attacks
   The adversary who has stolen the verifier V_{pa} (or V_{pb}) of user A's password P_a (or user B's password P_b) cannot impersonate as that user A (or B) since $\langle g^x \rangle_{S_{pa}}$ (or $\langle g^y \rangle_{S_{pb}}$) cannot be determined without knowledge of the private key S_{pa} (or S_{pb}). Accordingly, the proposed protocol can resist stolen-verifier attacks.

4.2. Performance analyses and comparisons

Table 2 compares the performances of the proposed three-party EKE (3PEKE) protocols with those of the three-party EKE protocols of Yeh et al. For plaintext-equivalent, both the three-party EKE protocols of Yeh et al. and the proposed three-party EKE protocol use four random numbers computations, four modular exponential computations and two asymmetric encryption computations. However, the proposed protocol has one step less than the three-party EKE protocol of Yeh et al. Besides, for verifier-based, both the protocols (VB-3PEKE) of Yeh et al. and the proposed protocol require four random number computations, four modular exponential computations and four asymmetric encryption computations. However, the proposed protocol also has one step less than the protocol of Yeh et al. Besides, both the round-efficient protocol can be executed in three rounds.

5. Conclusions

This study proposes two communication-efficient three-party EKE protocols based on the three-party EKE protocol of Yeh et al. One is a plaintext-equivalent authentication protocol and the other is a verifier-based authentication protocol. The computation cost of the proposed three-party EKE protocols is equal to that of the three-party EKE protocols of Yeh et al. In addition, the proposed three-party EKE protocols reveal the same information with the three-party EKE protocols of Yeh et al. According the Diffie-Hellman assumptions and security analyses of the three-party EKE protocols of Yeh et al., the proposed three-party EKE protocols also achieve the security requirements of three-party EKE protocols. However, the proposed protocols have one step less than the three-party EKE protocols of Yeh et al. Therefore, the proposed three-party EKE protocols are more efficient than three-party EKE protocols of Yeh et al. Moreover, the proposed protocols can be executed in three rounds by arranging the messages in parallel.

References

[1] S. Bellovin, 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] M. Steiner, G. Tsudik, M. Waidner, Refinement and extension of encrypted key exchange, ACM Operating Systems Review 29 (3) (1995) 22–30.
[3] D. Jablon, Strong password-only authentication key exchange, Computer Communication Review 26 (5) (1996) 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. Kwon, M. Kang, J. Song, An adaptable and reliable authentication protocol for communication networks, in: Proceedings of IEEE INFOCOM 97, 1997, pp. 738–745.
[7] T. Kwon, J. Song, Efficient key exchange and authentication protocol protecting weak secrets, IEICE Transactions on Fundamentals E81-A (1) (1998) 156–163.
[8] T. Kwon, J. Song, Authentication key exchange protocols resistant to password guessing attacks, IEE Communications 145 (5) (1998) 304–308.
[9] L. Gong, M. Lomas, R. Needham, J. Saltzer, Protecting poorly chosen secrets form guessing attacks, IEEE Journal on Selected Areas in Communications 11 (5) (1993) 648–656.
[10] L. Gong, Optimal authentication protocols resistant to password guessing attacks, in: Proceedings of the 8th IEEE Computer Security Foundation Workshop, 1995, pp. 24–29.
[11] T. Kwon, M. Kang, S. Jung, J. Song, An improvement of the password-based authentication protocol (K1P) on security against replay attacks, IEICE Transaction on Communications E82-B (7) (1999) 991–997.

[12] Y. Ding, P. Horster, Undetectable on-line password guessing attacks, ACM Operating System Review 29 (4) (1995) 77–86.
[13] C.L. Lin, H.M. Sun, T. Hwang, Three-party encrypted key exchange: Attacks and a solution, ACM Operating System Review 34 (4) (2000) 12–20.
[14] S. Keung, 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.
[15] H.T. Yeh, H.M. Sun, T. Hwang, Efficient three-party authentication and key agreement protocols resistant to password guessing attacks, Information Science and Engineering 19 (6) (2003) 1059–1070.
[16] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory IT-22 (6) (1976) 644–654.