I.J. Wireless and Microwave Technologies, **2011, 5, 16-22**
Published Online October 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2011.05.03

Available online at http://www.mecs-press.net/ijwmt

A Three-Party Password Authenticated Key Exchange Protocol with Key Confirmation

Gang Yao

State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing 100049, P.R.China

Abstract

Three-party authenticated key exchange protocol is an important cryptographic technique in the secure communication areas, by which any two clients can verify the ability to use a server to establish communication. Recently, researchers have begun proposing new key exchange protocols that would not require the use of server public keys, but a human-memorable password. In this paper, we propose a new three-party password authenticated key exchange protocol with key confirmation. The security of our proposed protocol relies on the hardness of the bilinear Diffie-Hellman problem and Diffie-Hellman problem in the random oracle model, and the proposed protocol achieves the security attributes: dictionary attack resilience, known session key security, perfect forward secrecy, no key compromise impersonation, no unknown key share and no key control.

Index Terms: Three-party password authenticated key exchange; key confirmation; pairing; security requirements.

1. Introduction

It is not secure to transfer a message over the channel directly because an adversary might control the channel. How to communicate securely over a channel is a fundamental problem in cryptography. There are two common methods for the persons to encrypt and authenticate their messages in order to protect the privacy and authenticity of these messages. One is by using public-key encryption and signature, the other is by means of a key exchange protocol. In practice, a kind of key exchange protocols which has received significant attention recently are those based on passwords.

Password-authenticated key exchange (PAKE) protocols enable two entities who share a small password to authenticate each other and agree on a large session key between them. Such protocols are attractive for their simplicity and convenience and have received much interest in the research community. Unlike a cryptographic-

* Corresponding author.
E-mail address: gyao@is.ac.cn

key authenticated key exchange protocol, the two communicating parties do not have any pre-shared cryptographic symmetric key, certificate or support from a trusted third party. Instead they only share a password.

The concept of PAKE was first introduced by Bellovin and Merritt [2] in 1992 known as Encrypted Key Exchange, and various protocols have been proposed to achieve secure password authentication key exchange, such as [4], [5], [8], [9]. Most of these protocols assume that the two users have a common, pre-shared password. However, this assumption is hard to satisfy in some applications. It would be more plausible to assume that a user wants to communicate securely with another user with the two different passwords. In such a case, a two-pary PAKE protocol is hard to implement since the number of passwords that a user has to memorize linearly increases with the number of possible partners. PAKE with different passwords in the three-party setting surmounts all the above problems. In this setting, each user shares a password only with a trusted server. The trusted server authenticates two users and helps the users with different passwords share a common session-key. It thus requires each user to only remember a single password with the trusted server. Consequently, three-party PAKE protocols can limit the number of passwords that each user must memorize. However, the server has to participate in the protocol run to help two users share a session-key.

Although three-party authenticated key exchange protocol can be composed of two password-based authentication key exchanges for two parties, but the efficiency and performance of the combination will be worse than those of an individual three-party authenticated key exchange protocol [11]. Many papers have considered password-based authenticated key exchange protocols in the 3-party setting, such as [1], [6], [10], [11], [12], [16]. Unfortunately, some of these proposed protocols are secure. Chung and Ku [7] pointed out that scheme in [12] cannot resist three variants of the man-in-the-middle attack. Wang and Mo [15] pointed out that scheme in [10] suffered from the impersonation attack. Nam et al. [13] pointed out that the protocol in [16] is vulnerable to unknown key-share attack, and proposed a way to prevent their attack. Phan et al. [14] pointed out that the protocol in [13], [16] are susceptible to key compromise impersonation attacks.

In this paper, we propose a new three-party password authenticated key exchange protocol with key confirmation. The security of our proposed protocol relies on the hardness of the bilinear Diffie-Hellman problem and Diffie-Hellman problem in the random oracle model and the proposed protocol achieves the security attributes.

2. Preliminaries

Here, we briefly recall some basic definitions.

A. Security Requirements

The basic requirements of PAKEs can be found in literature, e.g. [3]. Here, we highlight the security attributes.

- Dictionary attack resilience: A dictionary attack is a password guessing technique in which the adversary attempts to determine a user's password by successively trying words from a dictionary in the hope that one of these password guesses will be the user's actual password. Informally, in the scenario of PAKE protocols, we say that a protocol is secure against off-line dictionary attacks if an adversary who obtains all the communication data between the client and the server is unable to carry out the dictionary attack to obtain the client's password.
- Unknown key-share attack resilience: Unknown key-share attack is an attack where a party A believes that he shares a key with another party B upon completion of a protocol run (this is in fact the case), but B falsely believes that the key is instead shared with a party $E \neq A$. A basic PAKE protocol should be resilient to this.
- Perfect forward secrecy: If long-term private keys or secrets of any party are compromised, the secrecy of previously established session keys should not be affected. This is an attempt to still offer some form of security guarantee in spite of the fact that the long-term secret has been leaked.
- Key-compromise impersonation resilience: The compromise of any party's (client or server) long-term key or secret should not enable the adversary to impersonate any other parties.

- Known-key Security: After each execution of a PAKE protocol, the two users can generate a unique secret session key. Each session key is independent of that one generated in another execution of the PAKE protocol. Moreover, the compromise of one session key should not lead to compromise of other session keys.
- Key control. The key should be determined jointly by both the clients. Even the server cannot decide the session key.

It is important for a security protocol, as is a PAKE protocol, to be secure not only against known types of attacks, but also be designed to resist any kind of attack by an adversary of some defined adversarial power.

B. Pairing

Let G1 and G2 be two cyclic groups of order q for some large prime q. G1 is a cyclic additive group and G2 is a cyclic multiplicative group. We assume that the discrete logarithm problems in both G1 and G2 are hard.

A *pairing* is a bilinear map e: $G_1 \times G_1 \rightarrow G_2$ which satisfies the following conditions:

- Bilinear: For all $(P_1, P_2) \in G_1 \times G_1$ and all $(a, b) \in Z_q \times Z_q$, we have $e(aP_1, bP_2) = e(P_1, P_2)^{ab}$.
- Non-degenerate: There exist non-trivial points $P_1, P_2 \in G_1$ such that $e(P_1, P_2) \neq 1$.
- Computable: For all $(P_1, P_2) \in G_1 \times G_1$, $e(P_1, P_2)$ is efficiently computable.

The Bilinear Diffie-Hellman (BDH) Problem for a bilinear pairing e: G1 ×G1 →G2 is defined as follows: Let a, b, c be the random number chosen from Zq*, given (P, aP, bP, cP), compute e(P, P)abc. The Diffie-Hellman (DH) Problem for group G1 is defined as follows: Let a, b be the random number chosen from Zq*, given (P, aP, bP), compute abP.

We assume that both the BDH problem and the DH problem are hard.

3. Three-Party Password Authenticated Key Exchange Protocol

In this section, we describe our new three-party password authenticated key exchange protocol with key confirmation.

C. Notations

Before we present our new three-party password authenticated key exchange protocol, we describe the notations used in the protocol:

A, B	Users in the protocol
S	The trusted server
pw_A, pw_B	The password owned by users
Q_A, Q_B	The shadows of user's password stored by S
H	A cryptographic hash function
H_1	A Map-to-Group hash function
K	The session key

D. System Setup

Let A, B be two participants, and S be the trusted server.

The trusted server S selects two cyclic groups G1 and G2 of order q for some large prime q (160-bit long). G1 is a cyclic additive group and G2 is a cyclic multiplicative group. Let P be an arbitrary generator of G1, and e: G1 ×G1 →G2 be a cryptographic pairing.

The trusted server S chooses H1: {0, 1}* → Zq* to be a Map-to-Group hash function, and H to be a cryptographic hash function. Then the trusted server S publishes system parameters {q, G1, G2, e, P, H1, H}.

The user A chooses a password pwA, then computes QA = H1(A||S||pwA)P and transfers it to the trusted server S through a secure channel. S store (A, QA) in his own database.

E. Key Exchange

The three-party password authenticated key exchange protocol may be performed as follows:

1) When the user *A* wants to communicate with the user *B*, he generates a random number *Nonce*, and sends a requirement (*A*, *Require*, *Nonce*) to the user *B*. Then, *A* generates a random number $x \in Z_q^*$, computes

$$R_A = (x + H_1(A||S||pw_A))P,$$

and sends (A, B, R_A, *Nonce*) to the trusted server *S*.

2) After receiving the requirement (*A*, *Require*, *Nonce*) from the user *A*, the user *B* generates a random number $y \in Z_q^*$, computes

$$R_B = (y + H_1(B||S||pw_B))P,$$

and sends (B, A, R_B, *Nonce*) to the trusted server *S*.

3) After receiving messages (A, B, R_A, *Nonce*) and (B, A, R_B, *Nonce*), the trusted server *S* generates a random number $r \in Z_q^*$, computes

$$X_A = R_A - Q_A,$$

$$X_B = R_B - Q_B,$$

$$T_A = rX_B + H_1(A||B||S||Nonce)Q_A,$$

$$T_B = rX_A + H_1(B||A||S||Nonce)Q_B,$$

$$V_S = e(rX_A, X_B),$$

$$V_{SA} = H(A||B||S||T_A||V_S),$$

$$V_{SB} = H(B||A||S||T_B||V_S),$$

and sends (S, T_A, V_{SA}) to the user *A*, (S, T_B, V_{SB}) to the user *B*.

4) After receiving the message (S, T_A, V_{SA}) from the trusted server *S*, the user *A* computes

$$N_A = T_A - H_1(A||B||S||Nonce)\, H_1(A||S||pw_A)P,$$

$$V_{AS} = H(A||B||S||T_A||e(xP, N_A)),$$

and verifies whether $V_{AS} = V_{SA}$ holds or not. If the verification succeeds, the user *A* computes

$$V_{AB} = H(A||B||S||Nonce||xN_A||e(xP, N_A)),$$

and sends (A, B, V_{AB}) to the user *B*. Otherwise, the protocol execution is terminated.

5) After receiving the message (S, T_B, V_{SB}) from the trusted server *S* and the message (A, B, V_{AB}) from the user *A*, the user *B* computes

$$N_B = T_B - H_1(B||A||S||Nonce)H_1(B||S||pw_B)P,$$

$$V_{BS} = H(B||A||S||T_B||e(yP, N_B)),$$

$$V'_{AB} = H(A||B||S||Nonce||yN_B\,||e(yP, N_B)),$$

and verifies whether $V_{BS} = V_{SB}$ and $V_{AB} = V'_{AB}$ hold or not. If the verification succeeds, the user *B* computes

$$V_{BA} = H(B||A||S||Nonce||yN_B||e(yP, N_B)),$$

$$K_{BA} = H(A||B||yN_B||e(yP, N_B)),$$

and sends (B, A, V_{BA}) to the user *A*. Otherwise, the protocol execution is terminated.

6) After receiving the message (B, A, V_{BA}) from the user B, the user A computes

$V'_{BA} = H(B||A||S||Nonce||xN_A||\ e(xP, N_A))$,

and verifies whether $V_{BA} = V'_{BA}$ holds or not. If the verification succeeds, the user A computes

$K_{AB} = H(A||B||\ xN_A||\ e(xP, N_A))$.

Otherwise, the protocol execution is terminated.

Finally, the user A and the user B negotiate a common session key $K = K_{AB} = K_{BA}$.

4. Analysis

In this section, we present a detailed analysis on the security and performance of the protocol.

F. Dictionary Attack

Suppose that the adversary wants to guess A's password. In our protocol, A's password is used to generate RA. If the adversary guess a password pw'A, he can compute Q'A = H1(A||S|| pw'A)P and X'A = RA – Q'A. Now, the adversary cannot verify whether pw'A is right or not using RA, Q'A, X'A and TB, for he does not know the values of r and pwB. For the similar reason, it is easy to see that the adversary cannot mount a dictionary attack on the password pwB.

The protocol suffers from automated online password guessing attacks (similar weaknesses can be found in other published password-based key exchange protocols). Suppose that the adversary mount an online guessing attack on A's password pwA. If the online adversary guess a password pw'A, he can compute R'A = (x + H1(A||S||pw'A))P, send (A, B, R'A, Nonce) to the trusted server S. After receiving the message (S, TA, VSA) from S, the adversary computes N'A = TA – H1(A||B||S||Nonce) H1(A||S||pw'A)P and V'AS = H(A||B||S||TA||e(xP, N'A)). The online adversary verify whether the guessed password pw'A is right or not by verifying whether V'AS = VSA holds or not. After the online adversary try this process t times, he can reduce the size of the possible set.

G. Unknown Key Share

In our protocol, if an adversary wants to share a session key with B, while B believes that he share the session key with A, the adversary needs to send the trusted server S a message containing (A, B, RA, Nonce). Since the adversary does not know the password pwA, he may guess a random string pw'A to take place of pwA. When the trusted server S sends the message (S, TA, VSA) to the adversary, the adversary cannot compute NA for he does not know the right password, so he cannot compute VAB. This is to say that the probability of the adversary successfully impersonate of client A can be negligible.

H. Perfect Forward Secrecy

Suppose that an adversary knows the password pwA belong to A, it is still difficult for him to get the previous session keys between A and B. Since the adversary knows pwA, he may compute xP = RA – H1(A||S||pwA)P and NA. But the adversary still cannot compute x, thus he cannot get K = H(A||B||xNA|| e(xP, NA)). That is, the adversary cannot get the session key between A and B.

Suppose that an adversary knows the passwords pwA and pwB, it is still difficult for him to get the previous session keys between A and B. Since the adversary knows pwA, he may computes xP and NA. Since the adversary knows pwB, he may compute yP and NB. But the adversary still cannot compute x or y, thus he cannot get K.

I. Key-Compromise Impersonation

Although the adversary knows the password pwC belong to C, it is useless for him to guess the password pwA of A. When clients A and B generate a session key, they do not use the information about pwC. Thus, the knowledge of pwC is useless for the adversary to impersonate client A. If the adversary wants to impersonate client A, he chooses a random password pw'A as A's password, and performs the protocol with client B. Since

the adversary does not know the right password, he cannot compute the right values of VAB. Thus, he cannot pass the verification by B. That is to say that the adversary cannot impersonate client A.

J. Known-Key Security

In our protocol, suppose that an adversary knows the session key KCD = H(C||D||x'y'r'P||(e(P, P))x'y'r'), and he wants to guess the session key KAB = H(A||B||xyrP||(e(P, P))xyr). Since the process of computing KAB do not use the session key KCD, and the parameters {x, y, r} and {x', y', r'} are independent, knowing the session key KCD is useless for computing the session key KAB.

K. No Key Control

In our protocols, the secret session key between A and B is KAB = H(A||B||xyrP||(e(P, P))xyr), which is determined by A, B and S. Since each user does not know the value of secret number chosen by the other participants, he cannot determine how to select the random number, such that the private session key is equal the pre-determined value, or fall into a pre-determined interval. Hence, none of the two users can influence the outcome of the secret session key, or enforce the session key to fall into a pre-determined interval.

L. Other Attacks

- *Replay Attack*: The probability of success with regard to a replay attack is trivially negligible. This is because *x, y* and *Nonce* are ephemeral parameters of both participants in a session.
- *Man-in-the-middle Attack*: Suppose that an attacker intends to intercept the communication between *A* and *B*, or users and the server, and intends to acquire the session key between *A* and *B*. Since the attacker cannot compute correct V_{SA}, V_{SB}, V_{AB} and V_{BA}, he cannot compute the session key K_{AB}.
- *Trivial attack*: Suppose that an attacker gets all the messages in the protocol, he may directly try to compute the session. However, due to the intractability of DH problem is hard in group G_1, the trivial attack is not possible in the proposed protocol.

M. Performance

We look at the communication cost of our protocol. In our protocol, there are seven messages. Our seven-message key exchange for three-party communication protocol is not only with authentication, but also contains the key confirmation.

We look at the computation cost of our protocol. In our protocol, the following table gives the computation of the user (MP represents scalar multiplications in group G1, e pairing computation, H hash function evaluation, H1 map-to-group hash operation): For A, 3 MP operations, 1 e operation, 4 H operations and 2 H1 operations; For B, 3 MP operations, 1 e operation, 4 H operations and 2 H1 operations; For S, 4 MP operations, 1 e operation, 2 H operations and 2 H1 operations.

Acknowledgment

This work is supported by the National Basic Research Program of China (973 Program) under Grant No. 2007CB311202, and the President Fund of GUCAS No. 0851012MN00. The authors would like to thanks the anonymous referees for their helpful comments.

References

[1] M. Abdalla, P. Fouque, and D. Pointcheval. "Password-Based Authenticated Key Exchange in the Three-Party Setting". *International Workshop on Theory and Practice in Public Key Cryptography*, LNCS 3386, Springer-Verlag, pp. 65-84, 2005.

[2] S. Bellovin and M. Merritt. "Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks". *Symposium on Security and Privacy*, IEEE Computer Society, pp. 72-84, 1992.

[3] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.

[4] C. Boyd, P. Montague and K. Nguyen. "Elliptic Curve Based Password Authenticated Key Exchange Protocols". *Australasian Conference on Information Security and Privacy*, LNCS 2119, Springer-Verlag, pp. 487-501, 2001.

[5] M. Bellare, D. Pointcheval, and P. Rogaway. "Authenticated Key Exchange Secure Against Dictionary Attacks". *Advances in Cryptology − Proceedings of EUROCRYPT 2000*, LNCS 1807, Springer-Verlag, pp. 139-155, 2000.

[6] Y. Chang. "A Practical Three-party Key Exchange Protocol with Round Efficiency". *International Journal of Innovative Computing*, vol. 4, no. 4, pp. 953-960, 2008.

[7] H. Chung and W. Ku, "Three Weaknesses in a Simple Three-Party Key Exchange Protocol". *Information Science*, vol. 178, no. 1, pp. 220-229, 2008.

[8] R. Gennaro. "Faster and Shorter Password-Authenticated Key Exchange". *Theory of Cryptography Conference*, LNCS 4948, Springer-Verlag, pp. 589-606, 2008.

[9] J. Katz, R. Ostrovsky and M. Yung. "Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords". *Advances in Cryptology − Proceedings of EUROCRYPT 2001*, LNCS 2045, Springer-Verlag, pp. 475-494, 2001.

[10] S. Lee, H. Kim and K. Yoo. "Efficient Verifier-based Key Agreement Protocol for Three Parties without Server's public key". *Applied Mathematics and Computation*, vol. 167, pp. 996-1003, 2005.

[11] C. Lin, H. Sun, and T. Hwang. "Three-Party Encrypted Key Exchange: Attacks and a Solution". *ACM Operating Systems Review*, vol. 34, no. 4, pp. 12-20, 2000.

[12] R. Lu, and Z. Cao. "Simple Three-Party Key Exchange Protocol". *Computers & Security*, vol. 26, no. 1, pp. 94-97, 2007.

[13] J. Nam, Y. Lee, S. Kim, and D. Won. "Security Weakness in a Three-party Pairing-based Protocol for Password Authenticated Key Exchange". *Information Sciences*, vol. 177, pp. 1364-1375, 2007.

[14] R. Phan, W Yau, and B. Goi. "Analysis of Two Pairing-based Three-party Password Authenticated Key Exchange Protocols". *International Conference on Network and System Security*, IEEE Computer Society, pp. 102-8106, 2009.

[15] R. Wang and K. Mo. "Security Enhancement on Efficient Verifier-based Key Agreement Protocol for Three Parties without Server's Public Key". *International Mathematical*, vol. 1, no. 20, pp. 965-972, 2006.

[16] H. Wen, T. Lee, and T. Hwang. "Provably Secure Threeparty Password-based Authenticated Key Exchange Protocol using Weil Pairing". *IEE Proceedings − Communications*, vol. 152, no. 2, pp. 138-143, 2005.