An Improved Certificateless Authenticated Key Agreement Protocol

Haomin Yang, Yaoxue Zhang, Yuezhi Zhou
Department of Computer Science and Technology
Tsinghua University
Beijing, China
yanghm07@mails.tsinghua.edu.cn

Abstract—**Recently, Mokhtarnameh, Ho, Muthuvelu proposed a certificateless key agreement protocol. In this paper, we show that their protocol is insecure against a man-in-the-middle attack which is a severe disaster for a key agreement protocol. In addition, the authors claimed that their scheme provides a binding a long-term public key with a corresponding partial private key. In fact, their protocol does not realize the binding.**

We propose an improved key agreement protocol based on the protocol proposed by Mokhtarnameh, Ho and Muthuvelu. The improved protocol can resist a man-in-the-middle attack as well as satisfy the desired security properties for key agreement. It truly realizes the one-to-one correspondence between the long-term public key and the partial private key of a user. If there are two different, working long-term public keys for the same identity, the key generation center will be identified as having misbehaved in issuing both corresponding partial private keys.

Keywords-certificateless public key cryptography; key agreement; man-in-the-middle attack; bilinear pairing

I. INTRODUCTION

Key agreement is a cryptographic primitive for building secure communication channels over a non-secure public network. Two or more parties authenticate each other and agree on a shared key for future communication. Using symmetric cryptosystems for authentication requires an out-of-band security mechanism to bootstrap a pre-shared secret key. Thus, key agreement usually depends on public key cryptography (PKC) in which each user has a unique long-term public key/private key pair [1, 2].

A traditional PKC depends on public key certificates and a public key infrastructure (PKI). It requires heavy certificate transmission, storage and verification overhead. Moreover, a PKI is complex and difficult to deploy. To eliminate the requirement of a PKI, certificates and much of the overhead associated with key management, Shamir [3] proposed the notion of identity-based cryptography (IBC). In IBC, the public key of a user is easily derived from his identity information, i.e., simple email addresses or other online identifiers, and therefore there is no necessity to verify the authenticity of the public key of a user. However, IBC requires that a user's private key must be calculated for him by a trusted authority, called a key generation center (KGC). Thus, there is an inherent private key escrow problem in IBC because the KGC is able to compute all users' private keys. Therefore, users must necessarily place a high level of trust in the KGC. To solve the private key escrow problem in IBC, AI-

Riyami and Paterson [4] introduced the notion of certificateless public key cryptography (CL-PKC). CL-PKC combines the best features of PKI and IBC, such as lack of certificates, no key escrow property, reasonable trust to trusted authority and lightweight infrastructure. In CL-PKC, a user generates his long-term private key by combining the partial private key provided by the KGC with the secret value generated by the user himself. In this way, the KGC has no access to user's long-term private key. Thus, there is no long-term private key escrow problem in CL-PKC.

Recently, Mokhtarnameh, Ho and Muthuvelu [5] proposed a key agreement protocol in the CL-PKC setting (denoted here as MHM protocol). However, we found that their protocol is insecure because it suffers from a man-in-the-middle (MITM) attack. A MITM attack is a form of active attack in which an attacker intercept the exchanged data and inject false information between the two parties, making them believe that they are communicating directly to each other, when in fact the entire conversation is controlled by the attacker. A MITM attack is severe disaster for a key agreement protocol [6]. In addition, Mokhtarnameh, Ho and Muthuvelu claimed their protocol provides a binding a long-term public key with a corresponding partial private key and ensures that users can only create one long-term public key for the corresponding private key. In fact, their protocol does not realize the binding.

In this paper, we improve the MHM protocol. The improved protocol can resist man-in-the-middle attack and truly realizes the one-to-one correspondence between the long-term public key and the partial private key of a user. If there are two different, working public keys for the same identity, the KGC will be identified as having misbehaved in issuing both corresponding partial private keys. The protocol preserves the desired security properties for key agreement. In addition, the improved protocol is secure as long as each user has at least one uncompromised secret key in each protocol run. (There are the following three types of independent, unrelated secret keys of a user in a certificateless key agreement protocol: a partial private key, a secret value and an ephemeral private key. Notice that a long-term private key is not included because it can be derived from the partial private key and the secret value).

The remainder of this paper is organized as follows. Section II presents the preliminaries, including the definition of bilinear pairing and the related computational hardness assumptions. Section III briefly reviews the MHM protocol. In Section IV, we show a man-in-the-middle attack on the MHM protocol. Section V presents the improved protocol. Section VI analyzes the modification to the MHM protocol and the security properties of the improved protocol, and compares with the MHM protocol in terms of security and efficiency. Section VII concludes the paper.

II. PRELIMINARIES

A. Bilinear pairing

Let G_1 be a cyclic additive group of prime order q and G_2 be a cyclic multiplicative group of the same order. We assume that the discrete logarithm problem is hard in both G_1 and G_2.

A cryptographic pairing is a bilinear map $e{:}G_1{\times}G_1{\rightarrow}G_2$, which satisfies the following three properties:

(1) **Bilinear**. For all P, $Q \in G_1$, we have $e(P+Q, R)=e(P, R)\mathrm{e}(Q, R)$ and $e(P, Q+R)=e(P,Q)e(P,R)$.

(2) **Non-degenerate**. For all $P \neq 1_{\mathrm{G1}}$, we have $e(P, P) \neq 1_{\mathrm{G2}}$.

(3) **Computable**. There exists an efficient algorithm to compute $e(P,Q)$ for P, $Q \in G_1$.

The bilinearity property implies we have $e(aP, bQ) = e(P,Q)^{ab}$ for any a, $b \in Z_q^*$ and P, $Q \in G_1$ (aP denotes P added to itself a times). The map e may be computed using a modified Weil Pairing [7] or Tate Pairing [8] on an elliptic curve over F_q.

B. Related computational hardness assumptions

The computation of the following computational hardness assumptions is infeasible in polynomial time [6, 9].

(1) **Discrete Logarithm Problem (DLP)**. Given P, $Q \in G_1$, find an element $a \in Z_q^*$ such that $P = aQ$.

(2) **Computational Diffie–Hellman Problem (CDHP)**: Given (P, aP, bP) in G_1 where a, $b \in Z_q^*$, compute abP.

(3) **Bilinear Diffie–Hellman Problem (BDHP)**: Given (P, aP, bP, cP) in G_1 where a, b, $c \in Z_q^*$, compute $e(P, P)^{abc} \in G_2$.

III. Review of the MHM Protocol

In this section, we review briefly the certificateless key agreement protocol proposed by Mokhtarnameh, Ho and Muthuvelu [MHM]. The protocol is presented as follows:

(1) KGC selects the system parameters (G_1, G_2, e, P, P_0, H_1, H_2, n) where G_1 is a cyclic additive group of prime order q, G_2 is a cyclic multiplicative group of the same order, P is a generator of G_1, $P_0 = sP$ is the system public key where s is the system master key, e is a cryptographic bilinear map e: $G_1{\times}G_1{\rightarrow}G_2$, as well as H_1 and H_2 are two cryptographic hash functions where $H_1{:}\{0,1\}^*{\rightarrow}G_1$ and $H_2{:}\{0,1\}^*{\times}\{0,1\}^*{\times}G_1{\times}G_2{\rightarrow}\{0,1\}^n$.

(2) For a given user i with identity ID_i, KGC generates $D_i = sQ_i$ as the partial private key of user i where $Q_i = H_1(ID_i)$.

(3) User i chooses a random value $x_i \in Z_q^*$ as his secret value, and then generates the long-term private key $S_i = x_iD_i$ and the long-term public key $P_i = x_iQ_i$.

(4) Users A and B execute the key agreement process (Fig. 1) to establish the shared session key $K = H_2(Q_A, Q_B, abP, K_{\mathrm{AB}})$ where $K_{AB} = e(aP + x_AQ_A, bP + x_BQ_B)^{\mathrm{s}}$.

A		B
$Q_A = H_1(ID_A)$		$Q_B = H_1(ID_B)$
$D_A = sQ_A$		$D_B = sQ_B$
$P_A = x_A Q_A$		$P_B = x_B Q_B$
$S_A = x_A D_A$		$S_B = x_B D_B$
$a \in_R Z_q^*$		$b \in_R Z_q^*$
$T_A = aP$	$\xrightarrow{P_A, T_A}$	$T_B = bP$
$K_{AB} = e(aP_0 + S_A, T_B + P_B)$	$\xleftarrow{P_B, T_B}$	$K_{AB} = e(T_A + P_A, bP_0 + S_B)$
$h = aT_B$		$h = bT_A$
	$K = H_2(Q_A, Q_B, h, K_{AB})$	

Figure 1. The MHM protocol

IV. A Man-in-the-middle Attack on the MHM Protocol

In this section, we show that an adversary *Adv* can do the following steps to perform a man-in-the-middle attack on the MHM protocol (Fig. 2):

(1) *Adv* intercepts the message <P_A, T_A> sent by A. He then computes $P_A^* = nP$ and $T_A^* = mP\text{-}nP$ where m and n are randomly selected by *Adv* ($m \neq n$). Then, *Adv* substitutes P_A^* for P_A and T_A^* for T_A, and sends P_A^* and T_A^* to B.

(2) *Adv* intercepts the message <P_B, T_B> sent by B. He then computes $P_B^* = vP$ and $T_B^* = uP\text{-}vP$ where u and v are randomly selected by *Adv* ($u \neq v$). Then, *Adv* substitutes P_B^* for P_B and T_B^* for T_B, and sends P_B^* and T_B^* to A.

A		*Adv*		B
$Q_A = H_1(ID_A)$				$Q_B = H_1(ID_B)$
$D_A = sQ_A$				$D_B = sQ_B$
$P_A = x_A Q_A$				$P_B = x_B Q_B$
$S_A = x_A D_A$				$S_B = x_B D_B$
$a \in_R Z_q^*$				$b \in_R Z_q^*$
$T_A = aP$	$\xrightarrow{P_A, T_A}$		$\xrightarrow{P_A^* = nP,\ T_A^* = mP - nP}$	$T_B = bP$
$K_{AB^*} = e(aP_0 + S_A, T_B^* + P_B^*)$	$\xleftarrow{P_B^* = vP,\ T_B^* = uP - vP}$		$\xleftarrow{P_B, T_B}$	$K_{BA^*} = e(T_A^* + P_A, bP_0 + S_B)$
$h_{AB^*} = aT_B^*$				$h_{BA^*} = bT_A^*$
$K_{AB^*} = H_2(Q_A, Q_B, h_{AB^*}, K_{AB^*})$				$K_{A^*B} = H_2(Q_A, Q_B, h_{BA^*}, K_{BA^*})$

Figure 2. A man-in-the-middle attack on the MHM protocol

After the attack shown in Fig. 2, A and B will compute $K_{AB*} = e(aP_0+S_A,\ T_B^*+P_B^*)$ and $K_{BA*} = e(T_A^*+P_A,\ bP_0+S_B)$ respectively. However, *Adv* is also able to compute K_{AB*} and K_{BA*}.

$$K_{AB*} = e(aP_0+S_A,\ T_B^*+P_B^*) = e(aP_0+S_A,\ uP\text{-}vP+vP) = e(aP_0+S_A,\ uP) = e(aP_0,\ uP)\ e(S_A,\ uP)$$

$$= e(uP_0,\ aP)\ e(x_A sQ_A,\ uP) = e(uP_0,\ T_A)\ e(P_A,\ uP_0) = e(uP_0,\ T_A+P_A)$$

$$h_{AB*} = aT_B^* = a(uP - vP) = (u\text{-}v)aP = (u\text{-}v)T_A$$

Adv knows u, v, T_A, P_A and P_0, and is able to compute K_{AB^*} and h_{AB^*}. Thus, the adversary is able to compute $K_{AB^*} = H_2(Q_A, Q_B, h_{AB^*}, K_{AB^*})$.

$$K_{BA^*} = e(T_A^*+P_A, bP_0+S_B) = e(mP\text{-}nP+nP, bP_0+S_B) = e(mP, bP_0+S_B) = e(mP, bP_0)\, e(mP, S_B)$$
$$= e(mP_0, bP)\, e(mP, x_B sQ_B) = e(mP_0, T_B)\, e(P_B, mP_0) = e(mP_0, T_B+P_B)$$

$$h_{BA^*} = bT_A^* = b(mP - nP) = (m\text{-}n)bP=(m\text{-}n)T_B$$

Adv knows m, n, T_B, P_B and P_0, and is able to compute K_{BA^*} and h_{BA^*}. Thus, the adversary is able to compute $K_{BA^*} = H_2(Q_A, Q_B, h_{BA^*}, K_{BA^*})$.

After the above attack, neither A nor B know that any attack was carried out, and both A and B believe that they have established a shared secret key with each other. In fact, each of them has established a shared key with the adversary. Therefore, the MHM protocol is not resilient to a MITM attack.

V. THE IMPROVED PROTOCOL

To overcome the weakness of the MHM protocol, we propose an improved protocol based on it. The improved protocol is specified by six randomized algorithms:

Setup. KGC runs a parameter generator to generate output G_1, G_2, e, where G_1 and G_2 are groups of some prime order q and e: $G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing map. KGC randomly generates the system master key $s \in Z_q^*$ and computes the system public key $P_0 = sP$, where P is a generator of G_1. Then, KGC chooses two cryptographic hash functions H_1 and H_2, where H_1:$\{0,1\}^* \times G_1 \rightarrow G_1$ and H_2: $\{0,1\}^* \times \{0,1\}^* \times G_1 \times G_2 \times G_1 \times G_1 \times G_1 \rightarrow \{0,1\}^n$ which acts as a key derivation function. Here, we assume that the hash functions are modeled as random oracles [10]. Finally, KGC publishes the system parameters params = $<G_1, G_2, e, P, P_0, H_1, H_2, n>$.

Set-Secret-Value. User i with identity $ID_i \in \{0,1\}^*$ picks a random value $x_i \in Z_q^*$ and sets x_i as the user's secret value. Then user i computes $X_i = x_i P$ and sends X_i to KGC.

Partial-Private-Key-Extract. For a given user i with identity ID_i, KGC generates the partial private key of i given as $D_i = sQ_i$ where $Q_i = H_1(ID_i, x_i P)$. Then, KGC sends D_i to user i over a secure channel.

Set-Private-Key. User i computes the private key $S_i = x_i D_i$ by secret value x_i and partial private key D_i.

Set-Public-Key. User i with identity ID_i compute $Y_i = x_i Q_i$ and sets $<X_i, Y_i>$ as the long-term public key P_i.

Key agreement. Suppose two users, A and B, wish to agree on a shared session key. A owns long-term public key $P_A = <X_A, Y_A>$ and private key S_A while B has long-term public key $P_B = <X_B, Y_B>$ and private key S_B. The process of the key agreement is described as follows (Fig. 3):

(1) A picks $a \in Z_q^*$ at random, computes $T_A = aP$ and sends X_A, Y_A and T_A to B.

(2) Upon receiving X_A, Y_A and T_A, B picks $b \in Z_q^*$ at random, computes $R_B = bP$ and sends X_B, Y_B and T_B to A. Then, B computes $K_{AB} = e(T_A+Y_A, bP_0+S_B)$ and $h = bT_A$. Finally, B computes session key $K = H_2(Q_A, Q_B, h, K_{AB}, e(D_B, Q_A), bX_A, x_BT_A)$.

(3) Upon receiving X_B, Y_B and T_B, A computes $K_{AB} = e(aP_0+S_A, T_B+Y_B)$ and $h = aT_B$. Finally, A computes session key $K = H_2(Q_A, Q_B, h, K_{AB}, e(D_A, Q_B), aX_B, x_AT_B)$.

A		B
$Q_A = H_1(ID_A, x_AP)$		$Q_B = H_1(ID_B, x_BP)$
$D_A = sQ_A$		$D_B = sQ_B$
$P_A = < X_A, Y_A > = < x_AP, x_AQ_A >$		$P_B = < X_B, Y_B > = < x_BP, x_BQ_B >$
$S_A = x_AD_A$		$S_B = x_BD_B$
$a \in_R Z_q^*$	$\xrightarrow{X_A, Y_A, T_A}$	$b \in_R Z_q^*$
$T_A = aP$	$\xleftarrow{X_B, Y_B, T_B}$	$T_B = bP$
$K_{AB} = e(aP_0 + S_A, T_B + Y_B)$		$K_{AB} = e(T_A + Y_A, bP_0 + S_B)$
$h = aT_B$		$h = bT_A$
$K = H_2(Q_A, Q_B, h, K_{AB}, e(D_A, Q_B), x_AT_B, aX_B)$		$K = H_2(Q_A, Q_B, h, K_{AB}, e(D_B, Q_A), bX_A, x_BT_A)$

Figure 3. The improved protocol

VI. Analysis and Comparisons

In this section, we analyze the modification to the MHM protocol, security properties of the improved protocol, and compare the improved protocol with the MHM protocol.

A. Modifications to the MHM protocol

In the improved protocol, there are the following two modifications to the MHM protocol. (We take user A as example to analyze since the two protocols are role symmetric)

(1) Embedding $e(D_A, Q_B)$, x_AT_B and aX_B in key derivation function H_2.

R**eason**. The MHM protocol suffers from a man-in-the-middle attack.

Result. This modification makes the improved protocol able to resist a man-in-the-middle attack, because the adversary is not able to compute $e(D_A, Q_B)$ where partial private key D_A is required. Notice that if we only want to resist a man-in-the-middle attack, embedding $e(D_A, Q_B)$ in H_2 is enough. Embedding $e(D_A, Q_B)$, x_AT_B and aX_B in key derivation function H_2 makes the improved protocol be secure as long as each party has at least one uncompromised secret.

(2) Embedding the x_AP in Q_A

Reason. Mokhtarnameh, Ho and Muthuvelu claimed their protocol provides a binding a long-term public key with a corresponding partial private key. In fact, their protocol does not realize the binding. In the MHM protocol, user A's partial private key D_A ($D_A=sH_1(ID_A)$) and long-term public key P_A ($P_A = x_AH_1(ID_A)$) are unrelated because system master key s and secret value x_A are unrelated.

Result. We adopt the binding technique presented in [4], and this modification truly realizes the one-to-one correspondence between the public key and the partial private key of a user, and ensures that users can only create one long-term public key for the corresponding private key. In the improved protocol, there is a one-to-one correspondence between partial private key D_A and secret value x_A because D_A is equal to $sH_1(ID_A, x_AP)$. There is also one-to-one correspondence between partial private key D_A and long-term public key P_A ($P_A = <x_AP, x_AH_1(ID_A, x_AP)>$). Thus, users can only create one long-term public key for the corresponding private key. If there are two different, working public keys for the same identity, the KGC will be identified as having misbehaved in issuing both corresponding partial private keys.

B. Security properties of the improved protocol

The improved protocol satisfies the following desired security properties for key agreement [11]:

Known-key security. Each session key is unique because users A and B choose ephemeral private key a and b, respectively, in each protocol run. Thus, the knowledge of previous session keys does not help the adversary to derive information about the other session keys.

Unknown key-share resilience. Q_A and Q_B are included in key derivation function H_2. Thus, two parties know who they share the key with.

Weak Perfect forward secrecy. Suppose that an adversary has compromised long-term secret key S_A, S_B, x_A, x_B, D_A and D_B, he cannot obtain ephemeral private key a or b, because these long-term secret keys are unrelated to ephemeral private keys a and b. Thus, the adversary is unable to determine previously established session keys. In [12], Krawczyk shows that no two-flow authenticated key agreement protocol can archive full perfect forward secrecy.

Key-compromise impersonation resilience. An adversary who has compromised the long-term private key of entity A is unable to compute the session key because D_A, x_A and a are also required in computing the session key. Thus, the adversary has no ability to impersonate entity B to establish a session key with entity A.

Leakage of ephemeral private keys resilience. Suppose that an adversary has obtained two ephemeral private keys of a session (i.e., a and b). He is not able to compute the session key because computing a session key also requires partial private key (i.e., D_A and D_B) and secret value (i.e., x_A and x_B).

C. Comparisons with the MHM protocol

The improved protocol requires two evaluations of bilinear pairing and five scalar multiplications in G_1 on one party. The number of evaluations of bilinear pairing and scalar multiplications on one user in the improved protocol are higher than that required in the MHM protocol. However, the improved protocol has a distinct advantage in terms of security which is more important than efficiency for key agreement (Table I).

TABLE I. COMPARISONS WITH THE MHM PROTOCOL

Protocol	*Keys correspondence*	*Security weakness*	*P*	*M*	*Communication cost (block)*
MHM [5]	No	MITM attack	1	3	2
Improved protocol	Yes		2	5	3

P: evaluation of the bilinear pairing; M: scalar multiplication in G_1;
Keys correspondence: one-to-one correspondence between the public key and the partial private key of a user.

VII. CONCLUSIONS

In this paper, we have shown that MHM protocol is insecure against the man-in-the-middle attack and propose an improved protocol. The improved protocol can resist a man-in-the-middle attack as well as satisfy the desired security properties for key agreement. It truly realizes the one-to-one correspondence between the public key and the partial private key of a user. The efficiency of the improved protocol is lower than that of the MHM protocol. However, the improved protocol has a distinct advantage in terms of security which is more important than efficiency for a key agreement protocol.

REFERENCES

[1] W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and authenticated key exchanges," Designs, Codes and Cryptography. vol. 2, pp.107-125, June 1992.

[2] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman key exchange protocols," Proc. of the 5th Annual Workshop on Selected Areas in Cryptography, pp.339-361, 1998.

[3] A. Shamir, Identity-based cryptosystems and signature schemes, Proc. CRYPTO 1984, Santa Barbara, CA, 1984, pp.47-53.

[4] S. S. Al-riyami, K. G. Paterson, and R. Holloway, "Certificateless public key cryptography," in Proc. Asiacrypt'03. Springer-Verlag, 2003, pp. 452–473.

[5] R. Mokhtarnameh, S. Ho, and N. Muthuvelu, "An enhanced certificateless authenticated key agreement protocol", Proc. of the 13th International Conference on Advanced Communication Technology (ICACT), pp.802-806, 2011.

[6] J. Katz and Y. Lindell. Introduction to modern cryptography. Chapman & Hall/CRC Press, 2007, pp.476-478.

[7] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," Proc. CRYPTO 2001, LNCS 2139, 2001, pp.213-229.

[8] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," Proc. Algorithm Number Theory Symposium - ANTS V, LNCS 2369, 2002, pp. 324-337.

[9] R. Dutta, R. Barua, P. Sarkar, "Pairing-based cryptographic protocols - A survey," Cryptology ePrint Archive: Report 2004/064, 2004.

[10] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," Proc. of the 1st ACM Conference on Computer and Communications Security (CCS'93), pp.62-73, 1993.

[11] C. Swanson, D. Jao, "A Study of Two-Party Certificateless Authenticated Key-Agreement Protocols," INDOCRYPT, 2009, LNCS 5922, Springer Berlin/Heidelberg, 2009, pp. 57-71.

[12] H. Krawczyk: "HMQV: A high-performance secure Diffie-Hellman protocol," Cryptology ePrint Archive: Report 2005/176, 2005.