Generalization of Multiple Key Agreement Protocol based on Bilinear Pairings *

Cheng-Chi Lee[1], Ting-Yao Cheng[2], Te-Yu Chen[3]
[1]Department of Library and Information Science
Fu Jen Catholic University, 510 Jhongjheng Road
Sinjhuang City, Taipei County 24205
Taiwan, R.O.C
cclee@mail.fju.edu.tw

[2]Department of Foreign Languages and Literature Asia University
No. 500, Lioufeng Road
Wufeng Shiang, Taichung
Taiwan, R.O.C

[3]Department of Information Management
Hsiuping Institute of Technology No.11, Gongye Road
Dali City, Taichung County 412-80
Taiwan, R.O.C
chendy@mail.hit.edu.tw

Journal of Digital Information Management

ABSTRACT: *In practice, we usually require a key agreement protocol to establish a common secret key for enciphering/deciphering messages. Now, most protocols can establish n^2 common secret keys in a single round of messages exchange. In this paper, we shall propose an extended protocol to establish $2n^2$ common secret keys in a single round. The proposed protocol is based on bilinear pairings. Moreover, the correctness, security, and efficiency of our proposed protocol are presented.*

Categories and Subject Descriptors
E 4. [Coding and Information Theory]: Data compaction and compression; **E 3. [Data encryption]:** Public key cryptosystems

General Terms:
Data security, Data protocols, Data coding

Keywords: Bilinear pairing, Data encryption, Key agreement, Key exchange, MQV

Received: 27 December 2009; Revised 29 March 2010; Accepted 9 April 2010

1. Introduction

Two communicating parties can communicate with each other privately by using conventional symmetric-key cryptosystems such as the AES [1]. The two communicating parties have a common secret key to encrypt and decrypt their communicated messages by using symmetric-key cryptosystem. However, how do two parties securely obtain the common secret key between them in Internet? This can be solved by using Diffie-Hellman key exchange protocol [2]. However, the original Diffie-Hellman key exchange protocol does not support the authentication between the two parties as a result of the man-in-the-middle attack [3, 4]. Since then, Menezes, Qu, and Vanstone [5] proposed the first important key agreement protocol, which is also called the MQV protocol, to sign a signature for the Diffie-Hellman public keys without using a one-way hash function. And the MQV key agreement protocol has become a standard adopted by IEEE P1363 [6].

1.1 Related Work

Based on the MQV protocol, Harn and Lin proposed a generalized multiple key agreement protocol [7]. It enables two communicating parties to establish n^2 common secret keys if they send n Diffie-Hellman public keys in a single round of message exchange. To avoid the known key attack [8], the Harn-Lin protocol adopted no more than $n^2 - 1$ common secret keys. However, Yen and Joye showed that the Harn-Lin protocol cannot resist a forgery attack and proposed an improved protocol [9]. Later, Wu et al. showed that the Yen-Joye protocol still cannot withstand a forgery attack [10]. Therefore, Hwang et al. proposed an improved Yen-Joye protocol to overcome its weakness [11]. In 2001, Harn and Lin [12] modified their protocol to withstand the forgery attack, but their protocol still limits that only $n^2 - 1$ common secret keys to be adopted between two communicating parties.

In 2002, Tseng [13] proposed a robust generalized multiple key agreement protocol. It enables two communicating parties to establish n^2 common secret keys if they send n Diffie-Hellman public keys in a single round of message exchange. In addition, his protocol can avoid the known key attack if two parties use all shared n^2 common secret keys. Tzeng claimed that his protocol is robust against the forgery attack and the known key attack. Unfortunately, Shao showed that Tzeng's protocol is insecure against the forgery attack and then proposed an improved authenticated multiple key agreement protocol to resist the attack [14]. However, in 2007, Shim showed that Shao's improvement is also insecure [15]. In 2008, Lee et al. proposed two new multiple key agreement protocols based on elliptic curves and bilinear pairings [16]. The first protocol is based on elliptic curves. It is more efficient than [12, 17, 18] and it can achieve the same security with smaller key size. And the second protocol is based on bilinear pairings. It can keep the same properties of previous protocols. The available number of shared common secret keys is more than that in [12, 17, 18]. In other words, it allows two parties to establish n^2 common secret keys if they send n Diffie-Hellman public keys.

Until to now, all researches study on how to establish n^2 com-

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC99-2221-E-030-022.

mon secret keys by sending n Diffie-Hellman public keys in a single round of message exchange. In this paper, the authors shall propose a generalized multiple key agreement protocol based on bilinear pairings. It allows two communicating parties to establish $2n^2$ common secret keys in a single round.

1.2 Organization of This Paper

The organization of this paper is as follows: In Section 2, the bilinear pairings is first review. Then, we propose our protocol based on bilinear pairings in Section 3. In Section 4, the correctness, security, and efficiency of our proposed protocol are discussed. Finally, our brief conclusion is in Section 5.

2. Bilinear Pairings

Let G_1 be a cyclic additive group of prime order q, and G_2 be a cyclic multiplicative group of the same order q, and e : $G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following properties [19, 20]:

1. Bilinear:

$e(P_1 + P_2,Q)= e(P_1,Q)e(P_2,Q)$,

$e(P, Q_1 + Q_2)= e(P, Q_1)e(P, Q_2)$,

$e(aP, b_Q)= e(P, Q)^{ab}$,

where for all $P, P_1,P_2, Q, Q_1,Q_2 \in G_1$ and $a, b \in Z_q^*$.

2. Non-degenerate: If P is generator of $G1$, then $e(P, P) \neq 1$.

3. Computable: There is an $e$fficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

The security of bilinear parings is based on diffculty of the computational Diffie-Hellman problem and bilinear Diffie-Hellman problem which are defined as follows [19, 20]:

1. Discrete Logarithm Problem (DLP): Given P and $Q \in G$, to find an integer $n \in Z_q*$, such that $Q = nP$.

2. Decision Diffie-Hellman Problem (DDHP): Given P, aP, bP, and cP, to decide whether $c = ab$ mod q, where a, b, and $c \in Z_q^*$.

3. Computational Diffie-Hellman Problem (CDHP): Given aP , and bP , to compute abP , where a, and $b \in Z_q^*$.

3. The Proposed Protocol

In this section, we shall propose a generalized multiple key agreement protocol, which is based on bilinear pairings, to enable two communicating parties to establish multiple common secret keys in a single round of message exchange. We divide our generalized protocol into two subsection to explain our generalized protocol, respectively. In Section 3.1, we shall show that eight $(2\times2^2= 8)$ common secret keys are generated by sending two Diffie-Hellman public keys. In Section 3.2, we shall show that eighteen $(2 \times 3^2 = 18)$ common secret keys are generated by sending three Diffie-Hellman public keys. From these presentations, it is easily seen that our generalized protocol can share $2n^2$ common secret keys by sending n Diffie-Hellman public keys.

3.1 In Sending Two Diffie-Hellman Public Keys

In this subsection, we shall propose a multiple key agreement protocol to establish eight common secret keys between two communicating parties. Now, we suppose that Bob and Alice want to establish eight common secret keys by sending two Diffie-Hellman public keys. Let $X_A/X_B \in Z_q^*$ and $Y_A/Y_B(= X_AP/X_BP)$ be Alice's/Bob's long-term private key and long-term public key. Alice and Bob can perform the following steps to establish eight common secret keys.

1. Alice selects two short-term random numbers a_1 and $a_2 \in Zq^*$, and computes $T_{A1} = a_1P$ and $T_{A2} = a_2P$. Let K_{A1} and K_{A2} be the x-coordinate values of T_{A1} and T_{A2}, respectively. Then, Alice can generate her signature S_A as follows:

$$S_A =(a_1K_{A1} + a_2K_{A2})T_{A1} + X_AT_{A2}. \quad (1)$$

Alice then sends the messages $(T_{A1},T_{A2},S_A, \text{Cert}(Y_A))$ to Bob.

2. In the similar way, Bob also selects two short-term random numbers b_1 and $b_2 \in Z_q^*$, and computes $T_{B1} = b_1P$ and $T_{B2} = b_2P$. Let K_{B1} and K_{B2} be the x-coordinate values of T_{B1} and $TB2$, respectively. Then, Bob can generate his signature S_B as follows:

$$S_B =(b_1K_{B1} + b_2K_{B2})T_{B1} + X_BT_{B2}. \quad (2)$$

Bob then sends the messages $(T_{B1},T_{B2},S_B, \text{Cert}(Y_B))$ to Alice.

3. Alice verifies the authenticated messages $(T_{B1},T_{B2},S_B, \text{Cert}(Y_B))$ from Bob. Firstly, Alice takes out the x-coordinate values K_{B1} and K_{B2} from T_{B1} and T_{B2}, respectively. Then Alice verifies the equation

$$e(S_B,P) =?e(K_{B1}T_{B1} + K_{B2}T_{B2},T_{B1})e(T_{B2},Y_B). \quad (3)$$

If it is correct, Alice can generate the eight common secret keys as follows:

$$K_1 = e(a_1T_{B1},Y_A + Y_B),$$
$$K_2 = e(a_1T_{B2},Y_A + Y_B),$$
$$K_3 = e(a_2T_{B1},Y_A + Y_B),$$
$$K_4 = e(a_2T_{B2},Y_A + Y_B),$$
$$K_5 = e(a_1T_{B1},X_AY_B),$$
$$K_6 = e(a_1T_{B2},X_AY_B),$$
$$K_7 = e(a_2T_{B1},X_AY_B),$$
$$K_8 = e(a_2T_{B2},X_AY_B).$$

4. Bob also verifies the authenticated messages $(T_{A1},T_{A2},S_A, Cert(Y_A))$ from Alice. Firstly, Bob takes out the x-coordinate values K_{A1} and K_{A2} from T_{A1} and T_{A2}, respectively. Then Bob verifies the equation

$$e(S_A,P) =?e(K_{A1}T_{A1} + K_{A2}T_{A2},T_{A1})e(T_{A2},Y_A). \quad (4)$$

If it is correct, Bob can generate the eight common secret keys as follows:

$$K_1 = e(b_1T_{A1},Y_A + Y_B),$$
$$K_2 = e(b_2T_{A1},Y_A + Y_B),$$
$$K_3 = e(b_1T_{A2},Y_A + Y_B),$$
$$K_4 = e(b_2T_{A2},Y_A + Y_B),$$
$$K_5 = e(b_1T_{A1},X_BY_A),$$
$$K_6 = e(b_2T_{A1},X_BY_A),$$
$$K_7 = e(b_1T_{A2},X_BY_A),$$
$$K_8 = e(b_2T_{A2},X_BY_A).$$

Finally, Alice and Bob can share the eight common secret keys.

3.2 In Sending Three Diffie-Helmman Public Keys

In this subsection, we shall propose a multiple key agreement protocol to establish eighteen common secret keys between two communicating parties. Now, we suppose that Bob and Alice want to establish eighteen common secret keys by sending three Diffie-Hellman public keys. Let $X_A/X_B \in Z_q^*$ and $Y_A/Y_B(= X_AP/X_BP)$ be Alice's/Bob's long-term private key and long-term public key. Alice and Bob can perform the following steps to establish eighteen common secret keys.

1. Alice selects three short-term random numbers a_1, a_2 and $a_3 \in Z_q^*$, and computes $T_{A1} = a_1P$, $T_{A2} = a_2P$, and $T_{A3} = a_3P$. Let $KA1$, $KA2$, and $KA3$ be the x-coordinate values of T_{A1}, T_{A2}, and T_{A3}, respectively. Then, Alice can generate her signature S_A as follows:

$$S_A = (a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2} + X_AT_{A3}. \quad (5)$$

Alice then sends the messages $(T_{A1},T_{A2},T_{A3},S_A, Cert(Y_A))$ to Bob.

2. In the similar way, Bob also selects three short-term random numbers b_1, b_2, and $b_3 \in Z_q^*$, and computes $T_{B1} = b_1P$, $T_{B2} = b_2P$, and $T_{B3} = b_3P$. Let K_{B1}, K_{B2}, and K_{B3} be the x-coordinate values of T_{B1}, T_{B2}, and T_{B3}, respectively. Then, Bob can generate his signature S_B as follows:

$$S_B = (b_1K_{B1} + b_2K_{B2} + b_3K_{B3})T_{B1}T_{B2} + X_BT_{B3}. \quad (6)$$

Bob then sends the messages $(T_{B1},T_{B2},T_{B3},S_B, Cert(Y_B))$ to Alice.

3. Alice verifies the authenticated messages $(T_{B1},T_{B2},T_{B3},S_B, Cert(Y_B))$ from Bob. Firstly, Alice takes out the x-coordinate values K_{B1}, K_{B2}, and K_{B3} from T_{B1}, T_{B2}, and T_{B3}, respectively. Then Alice verifies the equation

$$e(S_B,P) = ?e(K_{B1}T_{B1} + K_{B2}T_{B2} + K_{B3}T_{B3},T_{B1}T_{B2})e(T_{B3},Y_B). \quad (7)$$

If it is correct, Alice can generate the eighteen common secret keys as follows:

$$\begin{aligned}
K_1 &= e(a_1T_{B1},Y_A + Y_B), & K_{10} &= e(a_1T_{B1},X_AY_B),\\
K_2 &= e(a_1T_{B2},Y_A + Y_B), & K_{11} &= e(a_1T_{B2},X_AY_B),\\
K_3 &= e(a_1T_{B3},Y_A + Y_B), & K_{12} &= e(a_1T_{B3},X_AY_B),\\
K_4 &= e(a_2T_{B1},Y_A + Y_B), & K_{13} &= e(a_2T_{B1},X_AY_B),\\
K_5 &= e(a_2T_{B2},Y_A + Y_B), & K_{14} &= e(a_2T_{B2},X_AY_B),\\
K_6 &= e(a_2T_{B3},Y_A + Y_B), & K_{15} &= e(a_2T_{B3},X_AY_B),\\
K_7 &= e(a_3T_{B1},Y_A + Y_B), & K_{16} &= e(a_3T_{B1},X_AY_B),\\
K_8 &= e(a_3T_{B2},Y_A + Y_B), & K_{17} &= e(a_3T_{B2},X_AY_B),\\
K_9 &= e(a_3T_{B3},Y_A + Y_B), & K_{18} &= e(a_3T_{B3},X_AY_B).
\end{aligned}$$

4. Bob also verifies the authenticated messages $(T_{A1},T_{A2},T_{A3},S_A, Cert(Y_A))$ from Alice. Firstly, Bob takes out the x-coordinate values K_A1, K_{A2}, and K_{A3} from T_{A1}, T_{A2}, and T_{A3}, respectively. Then Bob verifies the equation

$$e(S_A,P) = ?e(K_{A1}T_{A1} + K_{A2}T_{A2} + K_{A3}T_{A3},T_{A1}T_{A2})e(T_{A3},Y_A). \quad (8)$$

If it is correct, Bob can generate the eighteen common secret keys as follows:

$$\begin{aligned}
K1 &= e(b_1T_{A1},Y_A + Y_B), & K_{10} &= e(b_1T_{A1},X_BY_A),\\
K_2 &= e(b_2T_{A1},Y_A + Y_B), & K_{11} &= e(b_2T_{A1},X_BY_A),\\
K_3 &= e(b_3T_{A1},Y_A + Y_B), & K_{12} &= e(b_3T_{A1},X_BY_A),\\
K_4 &= e(b_1T_{A2},Y_A + Y_B), & K_{13} &= e(b_1T_{A2},X_BY_A),\\
\mathrm{K}_5 &= e(b_2T_{A2},Y_A + Y_B), & K_{14} &= e(b_2T_{A2},X_BY_A),\\
K_6 &= e(b_3T_{A2},Y_A + Y_B), & K_{15} &= e(b_3T_{A2},X_BY_A),\\
K_7 &= e(b_1T_{A3},Y_A + Y_B), & K_{16} &= e(b_1T_{A3},X_BY_A),\\
K_8 &= e(b_2T_{A3},Y_A + Y_B), & K_{17} &= e(b_2T_{A3},X_BY_A),\\
K_9 &= e(b_3T_{A3},Y_A + Y_B), & K_{18} &= e(b_3T_{A3},X_BY_A).
\end{aligned}$$

Finally, Alice and Bob can share the eighteen common secret keys.

4. Discussions

In this section, the correctness of our proposed protocol is shown firstly. Secondly, the security analysis of our proposed protocol will be given. Finally, the efficiency analysis of our proposed protocol will be also given.

4.1 Correctness

We proof the correctness of this paper in the following theorems. Here, we just proof $n = 2$ and n = 3 for example. In the same way, we can apply similar proofs to prove other n.

Theorem 4.1 *If the equation* $e(S_A,P) = ?e(K_{A1}T_{A1} + K_{A2}T_{A2},T_{A1})e(T_{A2},Y_A)$ *and* $e(S_A,P) = ?e(K_{A1}T_{A1} + K_{A2}T_{A2} + K_{A3}T_{A3},T_{A1}T_{A2})e(T_{A3},Y_A)$, *respectively, holds, Bob can verify the messages* $(T_{A1},T_{A2},S_A, Cert(Y_A))$ and $(T_{A1},T_{A2},T_{A3}, S_A, Cert(Y_A))$, *respectively, sent from Alice.*

Proof. Since $S_A = (a_1K_{A1} + a_2K_{A2})T_{A1} + X_AT_{A2}$ and the properties of the bilinear pairings, we have

$$\begin{aligned}
e(S_A,P) &= e((a_1K_{A1} + a_2K_{A2})T_{A1} + X_AT_{A2},P)\\
&= e((a_1K_{A1} + a_2K_{A2})T_{A1},P)e(X_AT_{A2},P)\\
&= e((a_1K_{A1} + a_2K_{A2})T_{A1},P)e(T_{A2},P)^{X_A}\\
&= e((a_1K_{A1} + a_2K_{A2})T_{A1},P)e(T_{A2},X_AP)\\
&= e((a_1K_{A1} + a_2K_{A2})T_{A1},P)e(T_{A2},Y_A)\\
&= e((a_1K_{A1} + a_2K_{A2}),P)^{T_{A1}}e(T_{A2},Y_A)\\
&= e((a_1K_{A1} + a_2K_{A2}),T_{A1})_P\, e(T_{A2},Y_A)\\
&= e((a_1PK_{A1} + a_2PK_{A2}),T_{A1})e(T_{A2},Y_A)\\
&= e(K_{A1}T_{A1} + K_{A2}T_{A2},T_{A1})e(T_{A2},Y_A)
\end{aligned}$$

Thus, the equation $e(S_A,P) = e(K_{A1}T_{A1} + K_{A2}T_{A2},T_{A1})e(T_{A2},Y_A)$. Next, since $S_A = (a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2} + X_AT_{A3}$ and the properties of the bilinear pairings, we have

$$\begin{aligned}
e(S_A,P) &= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2} + X_AT_{A3},P)\\
&= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2},P)e(X_AT_{A3},P)\\
&= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2},P)e(T_{A3},P)^{X_A}\\
&= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2},P)e(T_{A3},X_AP)\\
&= e((a_1K_A1 + a_2K_{A2} + a_3K_{A3})T_{A1}T_{A2},P)e(T_{A3},Y_A)\\
&= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3}),P)^{T_{A1}T_{A2}}e(T_{A3},Y_A)\\
&= e((a_1K_{A1} + a_2K_{A2} + a_3K_{A3}),T_{A1}T_{A2})_P\, e(T_{A3},Y_A)\\
&= e((a_1PK_{A1} + a_2PK_{A2} + a_3PK_{A3}),T_{A1}T_{A2})e(T_{A3},Y_A)
\end{aligned}$$

$$= e((K_{A1}T_{A1} + K_{A2}T_{A2} + K_{A3}T_{A3}), T_{A1}T_{A2})e(T_{A3}, Y_A)$$

Thus, the equation $e(S_A,P) = e(K_{A1}T_{A1}+K_{A2}T_{A2}+K_{A3}T_{A3}, T_{A1}T_{A2})\ e(T_{A3},Y_A)$.

In the same way, we can apply similar proofs to prove other signatures S_As holding when n =4, 5, 6, ..., N, where N is an integer.

Q.E.D.

Theorem 4.2 *If the equation* $e(S_B,P\ ) =?e(K_{B1}T_{B1}+K_{B2}T_{B2}, T_{B1})\ e(T_{B2},Y_B)$ *and* $e(S_B,P\ ) =?e(K_{B1}T_{B1}+K_{B2}T_{B2}+K_{B3}T_{B3}, T_{B1}T_{B2})e(T_{B3},Y_B)$, *respectively, holds, Alice can verify the messages* $(T_{B1}, T_{B2}, S_B, Cert(Y_B))$ *and* $(T_{B1}, T_{B2}, T_{B3}, S_B, Cert(Y_B))$, *respectively, sent from Bob.*

Proof. The proof is similar to theorem 4.1.

Q.E.D.

Theorem 4.3 *If the shared secret keys are generated, all shared secret keys are the same between Alice and Bob. We proof* n =2 *(eight keys) and* n =3 *(eighteen keys) for example.*

Proof. From Section 3.1, we know that the all common secret keys are computed by

$$K_1 = e(b_1T_{A1}, Y_A + Y_B) = e(P, P)^{a_1b_1(X_A+X_B)} = e(a_1T_{B1}, Y_A + Y_B),$$
$$K_2 = e(b_2T_{A1}, Y_A + Y_B) = e(P, P)^{a_1b_2(X_A+X_B)} = e(a_1T_{B2}, Y_A + Y_B),$$
$$K_3 = e(b_1T_{A2}, Y_A + Y_B) = e(P, P)^{a_2b_1(X_A+X_B)} = e(a_2T_{B1}, Y_A + Y_B),$$
$$K_4 = e(b_2T_{A2}, Y_A + Y_B) = e(P, P)^{a_2b_2(X_A+X_B)} = e(a_2T_{B2}, Y_A + Y_B),$$
$$K_5 = e(b_1T_{A1}, X_BY_A) = e(P, P)^{a_1b_1X_AX_B} = e(a_1T_{B1}, X_AY_B),$$
$$K_6 = e(b_2T_{A1}, X_BY_A) = e(P, P)^{a_1b_2X_AX_B} = e(a1T_{B2}, X_AY_B),$$
$$K_7 = e(b_1T_{A2}, X_BY_A) = e(P, P)^{a_2b_1X_AX_B} = e(a_2T_{B1}, X_AY_B),$$
$$K_8 = e(b_2T_{A2}, X_BY_A) = e(P, P)^{a_2b_2X_AX_B} = e(a_2T_{B2}, X_AY_B).$$

From Section 3.2, we know that the all common secret keys are computed by

$$K_1 = e(b_1T_{A1}, Y_A + Y_B) = e(P, P)^{a_1b_1(X_A+X_B)} = e(a_1T_{B1}, Y_A + Y_B),$$
$$K_2 = e(b_2T_{A1}, Y_A + Y_B) = e(P, P)^{a_1b_2(X_A+X_B)} = e(a_1T_{B2}, Y_A + Y_B),$$
$$K_3 = e(b_3T_{A1}, Y_A + Y_B) = e(P, P)^{a_1b_3(X_A+X_B)} = e(a_1T_{B3}, Y_A + Y_B),$$
$$K_4 = e(b_1T_{A2}, Y_A + Y_B) = e(P, P)^{a_2b_1(X_A+X_B)} = e(a_2T_{B1}, Y_A + Y_B),$$
$$K_5 = e(b_2T_{A2}, Y_A + Y_B) = e(P, P)^{a_2b_2(X_A+X_B)} = e(a_2T_{B2}, Y_A + Y_B),$$
$$K_6 = e(b_3T_{A2}, Y_A + Y_B) = e(P, P)^{a_2b_3(X_A+X_B)} = e(a_2T_{B3}, Y_A + Y_B),$$
$$K_7 = e(b_1T_{A3}, Y_A + Y_B) = e(P, P)^{a_3b_1(X_A+X_B)} = e(a_3T_{B1}, Y_A + Y_B),$$
$$K_8 = e(b_2T_{A3}, Y_A + Y_B) = e(P, P)^{a_3b_2(X_A+X_B)} = e(a_3T_{B2}, Y_A + Y_B),$$
$$K_9 = e(b_3T_{A3}, Y_A + Y_B) = e(P, P)^{a_3b_3(X_A+X_B)} = e(a_3T_{B3}, Y_A + Y_B),$$
$$K_{10} = e(b_1T_{A1}, X_BY_A) = e(P, P)^{a_1b_1X_AX_B} = e(a_1T_{B1}, X_AY_B),$$
$$K_{11} = e(b_2T_{A1}, X_BY_A) = e(P, P)^{a_1b_2X_AX_B} = e(a_1T_{B2}, X_AY_B),$$
$$K_{12} = e(b_3T_{A1}, X_BY_A) = e(P, P)^{a_1b_3X_AX_B} = e(a_1T_{B3}, X_AY_B),$$
$$K_{13} = e(b_1T_{A2}, X_BY_A) = e(P, P)^{a_2b_1X_AX_B} = e(a_2T_{B1}, X_AY_B),$$
$$K_{14} = e(b_2T_{A2}, X_BY_A) = e(P, P)^{a_2b_2X_AX_B} = e(a_2T_{B2}, X_AY_B),$$
$$K_{15} = e(b_3T_{A2}, X_BY_A) = e(P, P)^{a_2b_3X_AX_B} = e(a_2T_{B3}, X_AY_B),$$
$$K_{16} = e(b_1T_{A3}, X_BY_A) = e(P, P)^{a_3b_1XAXB} = e(a_3T_{B1}, X_AY_B),$$
$$K_{17} = e(b_2T_{A3}, X_BY_A) = e(P, P)^{a_3b_2X_AX_B} = e(a_3T_{B2}, X_AY_B),$$
$$K_{18} = e(b_3T_{A3}, X_BY_A) = e(P, P)^{a_3b_3X_AX_B} = e(a_3T_{B3}, X_AY_B).$$

Thus, Alice and Bob can share the same common secret keys.

Q.E.D.

4.2 Security Analysis

In this subsection, we shall show the security analysis of our protocol as follows.

4.2.1 Perfect forward secrecy

From Theorem 4.3, we know that the eight or eighteen common secret keys are computed by a_1, a_2, a_3, b_1, b_2, b_3, X_A, and X_B. The lifetime of the short-term random numbers a_1, a_2, a_3, b_1, b_2, and b_3 is only one session long. Hence, although the adversary knows X_A and X_B, he/she still needs to get random numbers a_1, a_2, a_3, $b1$, b_2, and b_3 for computing the eight or eighteen common secret keys. However, it is equivalent to solve the DLP to derive these random numbers a_1, a_2, a_3, b_1, b_2, and b_3 from public values T_{A1}, T_{A2}, T_{A3}, T_{B1}, T_{B2}, and T_{B3}. Therefore, the adversary does not know the forward session keys even though he/she knows X_A and X_B. It keeps the property of perfect forward secrecy.

4.2.2 Known key attack

The common secret keys established between Alice and Bob are not mutually independent. If an adversary obtains all common secret keys (or the short-term keys), he/she can obtain the long-term shared key $K_{AB}(= e(X_AX_BP, Y_A + Y_B) = e(P, P)^{X_AX_B(X_A+X_B)})$. This is called the known key attack [13]. The adversary may compute K_{AB} from Y_A and YB directly. However, it is equivalent to solve CDHP problem to compute X_AX_BP . Moreover, assume that the adversary knows the all common secret keys, and tries to derive the long-term shard key K_AB. He/she may try to derive the random numbers a_1, a_2, a_3, b_1, b_2, b_3, and applies them to derive X_A and X_B. However, it is also DLP problem. Without the knowledge of X_A and X_B, the adversary have no way to compute the long-term shared key. Thus, the proposed protocol can withstand the known key attack.

4.2.3 Forgery attack

Assume that an intruder wants to impersonate Alice to establish the common secret keys with Bob to pass the verification Eq. 4 or Eq. 8. The intruder needs to find S_A to satisfy Eq. 4 or Eq. 8. However, the intruder cannot forge a valid SA without the knowledge of X_A by Eq. 1 or Eq. 5. In addition, to find X_A from Y_A is equivalent to solve DLP problem. Thus, the proposed protocol can withstand the forgery attack.

4.3 Efficiency Analysis

In Table 1, It is seen that our protocol is more efficient than Harn-Lin's protocol [12], Tseng's protocol [13], Shao's protocol [14], and Lee et al.'s protocol [16]. Harn-Lin's protocol establishes four common secret keys in one round between two parties, but only three keys can be used for withstanding the known key attack. In generalization, their protocol can share n^2 common secret keys by sending n Diffie-Hellman public keys in one round and only $n^2 - 1$ keys can be used for withstanding the known key attack. In Tseng's protocol, the n^2 common keys can be used without suffering from the attack. An improvement of Tzeng's protocol proposed by Shao establishes n^2 common secret keys and the n^2 keys can be used. The first protocol of Lee et al.'s protocols establishes n^2 keys and only $n^2 - 1$ keys can be used. The second protocol of Lee et al.'s protocols establishes

	[12]	[13]	[14]	[16]	Our
Numbers of session key	n^2	n^2	n^2	n^2 / n^2	$2n^2$
Numbers of session key can be used	$n^2 - 1$	n^2	n^2	$n^2 - 1/n^2$	$2n^2$

Table 1. The comparison of efficiency

n^2 keys and the n^2 keys can be used. In the proposed protocol, two parties can establish $2n^2$ common secret keys and all the keys can be used. It is seen that our protocol is superior to other protocols.

5. Conclusions

In this paper, we have proposed a more efficient and a generalization of multiple key agreement protocol based on bilinear pairings. The protocol can provide perfect forward secrecy, withstand forgery attack, and withstand known key attack even though all common secret keys established between two parties are adopted. Compare with other protocols, our proposed protocol can establish $2n^2$ common secret keys between two parties in a single round of message exchange. Thus, our proposed protocol is superior to other protocols.

References

[1] National Institute of Standards and Technology, Specification for the Advanced Encryption Standard (AES), *Federal Information Processing Standards Publication (FIPS)* 197, Available at: http://csrc.nist. gov/publications/fips/fips197/fips- 197.pdf.

[2] Diffie, W., Hellman, M. E. (1976). New directions in cryptography, *IEEE Transactions on Information Theory*, IT-22, p. 644– 654.

[3] Kim, M. H. , Koc, C. K. (2008). Improving the Novikov and Kiselev user authentication scheme, *International Journal of Network Security*, 6 (3) 241–245.

[4] Yang, C. Y., Lee, C. C.,Hsiao, S. Y. (2005). Man-in-themiddle attack on the authentication of the user from the remote autonomous object, *International Journal of Network Security*, 1 (2) 81–83.

[5] Menezes, A. J. Qu, M.and Vanstone, S. A. (1995). Some key agreement protocols providing implicit authentication, *In*: *Proceedings of 2nd Workshop Selected Areas in Cryptography*, p. 22–32.

[6] IEEE 2000, (2002). IEEE Standard 1363-2000: Standard specifications for public key cryptography, *IEEE*, 2002.

[7] Harn, Lein., Lin, Hung-Yu (1998). An authenticated key agreement protocol without using one-way functions, *In*: *Proceedings of the 8th National Conference on Information Security*, p. 155–160, Kaohsiung, Taiwan, May 1998.

[8] Nyberg, K., Rueppel, R. A (1994). Weakness in some recent key agreement protocol, *IEE Electronics Letters*, 30 (1) 26–27.

[9] Yen, Sung-Ming., Joye, M. (1998). Improved authenticated multiple-key agreement protocol, *Electronics Letters*, 34 (18) 1738–1739.

[10] Wu, Tzong-Sun., He, Wei-Hua.,Hsu, Chien-Lung (1999). Security of authenticated multiple-key, *Electronics Letters*, 35 (5) 391–392.

[11] Hwang, Min-Shiang., Lin, Chih-Wei ., Lee, Cheng-Chi (2002). Improved yenjoye's authenticated multiple-key agreement protocol, *IEE Electronics Letters*, 38 (23) 1429–1431.

[12] Harn, Lein., Lin, Hung-Yu (2001). Authenticated key agreement without using one-way hash functions," *Electronics Letters*, 37 (10) 629–630.

[13] Tseng, Y. M. (2002). Robust generalized MQV key agreement protocol without using one-way hash function, *Computer Standards & Interfaces*, 24 (3) 241–246.

[14] Shao, Z. (2003). Security of robust generalized MQV key agreement protocol without using one-way hash functions," *Computer Standards and Interfaces*, 25 (5) 431-436.

[15] Shim, K. (2007). Vulnerabilities of generalized MQV key agreement protocol without using one-way hash functions, *Computer Standards and Interfaces*, 29 (4) p. 467– 470.

[16] Lee, N. Y., Wu, C. N., Wang, C. C. (2008). Authenticated multiple key exchange protocols based on elliptic curves and bilinear pairings, *Computers and Electrical Engineering*, 34 (1) p. 12–20.

[17] Hwang, R. J., Shiau, S. H., Lai, C. H. (2003). An enhanced authentication key exchange protocol, *In*: *Proceedings of the 17th international conference on Advanced Information Networking and Application*, p. 202–205,

[18] Lee, N. Y., Wu, C. N (2004). Improved authentication key exchange protocol without using one-way hash function," ACM Opera*tion Systems Review*, 38 (2) 85–92.

[19] Awasthi, Amit K., Lal, Sunder (2007). ID-based ring signature and proxy ring signature schemes from bilinear pairings, *International Journal of Network Security*, 4 (2) 187–192.

[20] Boneh, D., Franklin, M. (2001). Identity based encryption from the weil pairing, *In*: *Advances in Cryptology-Crypto'2001, LNCS 2139*, p. 213–229.