Security weakness of two authenticated key exchange protocols from pairings

Qingfeng Cheng, Chuangui Ma

Zhengzhou Information Science and Technology Institute
Zhengzhou 450002, P. R. China
qingfengc2008@sina.com

Abstract. Recently, Liu proposed two authenticated multiple key exchange protocols using pairings, and claimed two protocols featured many security attributes. In this paper, we show that Liu's protocols are insecure. Both of Liu's protocols cannot provide perfect forward secrecy.

Key words: Key compromise impersonation attack; Authenticated key exchange; Multiple key; Perfect forward secrecy.

1 Introduction

Authenticated key exchange (AKE) plays an important role in secure communications. An AKE protocol allows two or more parties to agree upon a secret common session key over a public network. But the design of secure AKE protocols has always been a notorious hard problem. Many AKE protocols that have appeared in the literature subsequently were proved to be flawed.

Recently, Liu proposed two AKE protocols. One is a three-party multiple key exchange protocol [1], which is based on the Lee's protocol [2] and Hölbl's protocol [3]. The other is a two-party authenticated multiple key exchange protocol [4], which is based on the Lee's protocol [2]. In this paper, we will show that both of them cannot provide perfect forward secrecy. In additional, the former cannot resist key compromise impersonation (KCI) attack.

The rest of this paper is organized as follows. In section 2 we introduce preliminaries used in this paper. In section 3 we review Liu's three-party protocol. In section 4 we present analysis of Liu's three-party protocol. In section 5, we review Liu's two-party protocol. In section 6, we propose analysis of Liu's two-party protocol. In the final section, we conclude this paper.

2 Preliminaries

In this section, we introduce several Diffie-Hellman problems. Let G_1 be an additive group of order q, and G_2 be a multiplicative group of order q. Let $Q, W \in G_1$ and $e : G_1 \times G_1 \longrightarrow G_2$ be a bilinear pairing that has the following properties:

– **Bilinearity**: For any $Q, W \in G_1$ and $a, b \in Z_q^*$, we have $e(aQ, bW) = e(Q, W)^{ab}$.
– **Non-degeneracy**: There exists $Q, W \in G_1$ such that $e(Q, W) \neq 1$.
– **Computability**: For any $Q, W \in G_1$, there exists an efficient algorithm to compute $e(Q, W)$.

Next, we describe DL and BDH problems:

– **Discrete Logarithm (DL) Problem**: Given two elements $Q, W \in G_1$. Find the integer n whenever such an integer exists, such that $Q = nW$.
– **Bilinear Diffie-Hellman (BDH) Problem**: Let P is a generator of G_1. Given (P, aP, bP, cP) with $a, b, c \in Z_q^*$, computes $e(P, P)^{abc} \in G_2$.

We say that G_2 satisfies the DL and BDH assumptions if no feasible adversary can solve the DL and BDH problems with non-negligible probability.

3 Review of Liu's Three-Party Protocol

In this section, we briefly review Liu's three-party protocol proposed by Liu in 2010. Let P be a generator of a cyclic additive group G_1 of the prime order q, and G_2 be a cyclic multiplicative group of the prime order q. $e : G_1 \times G_1 \longrightarrow G_2$ is a bilinear pairing. Each party i has a static private key $X_i \in Z_q^*$ and a static public key $Y_i (= X_i P)$. For more details about the protocol, refer to [1].

In the following description we suppose that three communications parties, A, B and C wish to communicate with each other.

1. Party A chooses $a_1, a_2 \in Z_q^*$ randomly and computes $T_{A1} = a_1 P$ and $T_{A2} = a_2 P$. Then party A computes

   $S_{A1} = a_1 X_A + a_2$, $S_{A2} = a_2 X_A + a_1$.

   Finally, party A sends the message $(T_{A1}, T_{A2}, S_{A1}, S_{A2}, Cert(Y_A))$ to party B and party C.
2. Similarly, party B chooses $b_1, b_2 \in Z_q^*$ randomly and computes $T_{B1} = b_1 P$ and $T_{B2} = b_2 P$. Then party B computes

   $S_{B1} = b_1 X_B + b_2$, $S_{B2} = b_2 X_B + b_1$.

   Finally, party B sends the message $(T_{B1}, T_{B2}, S_{B1}, S_{B2}, Cert(Y_B))$ to party A and party C.
   In the same way, party C chooses $c_1, c_2 \in Z_q^*$ randomly and computes $T_{C1} = c_1 P$ and $T_{C2} = c_2 P$. Then party C computes

   $S_{C1} = c_1 X_C + c_2$, $S_{C2} = c_2 X_C + c_1$.

   Finally, party C sends the message $(T_{C1}, T_{C2}, S_{C1}, S_{C2}, Cert(Y_C))$ to party A and party B.

3. Upon receiving the message from parties B and C, party A checks whether

$e((S_{B1}+S_{B2})P-(T_{B1}+T_{B2}),P)=e(T_{B1}+T_{B2},Y_B)$,
$e((S_{C1}+S_{C2})P-(T_{C1}+T_{C2}),P)=e(T_{C1}+T_{C2},Y_C)$,

if they are equal, then computes the session keys $K_i(i=1,2,...,8)$ as follows:

$$K_1=e(a_1T_{B1},X_A(S_{C1}P-T_{C2})+T_{C1})e(a_1(S_{B1}P-T_{B2}),X_AT_{C1})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_1b_1c_1}$$

$$K_2=e(a_1T_{B1},X_A(S_{C2}P-T_{C1})+T_{C2})e(a_1(S_{B1}P-T_{B2}),X_AT_{C2})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_1b_1c_2}$$

$$K_3=e(a_1T_{B2},X_A(S_{C1}P-T_{C2})+T_{C1})e(a_1(S_{B2}P-T_{B1}),X_AT_{C1})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_1b_2c_1}$$

$$K_4=e(a_1T_{B2},X_A(S_{C2}P-T_{C1})+T_{C2})e(a_1(S_{B2}P-T_{B1}),X_AT_{C2})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_1b_2c_2}$$

$$K_5=e(a_2T_{B1},X_A(S_{C1}P-T_{C2})+T_{C1})e(a_2(S_{B1}P-T_{B2}),X_AT_{C1})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_2b_1c_1}$$

$$K_6=e(a_2T_{B1},X_A(S_{C2}P-T_{C1})+T_{C2})e(a_2(S_{B1}P-T_{B2}),X_AT_{C2})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_2b_1c_2}$$

$$K_7=e(a_2T_{B2},X_A(S_{C1}P-T_{C2})+T_{C1})e(a_2(S_{B2}P-T_{B1}),X_AT_{C1})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_2b_2c_1}$$

$$K_8=e(a_2T_{B2},X_A(S_{C2}P-T_{C1})+T_{C1})e(a_2(S_{B2}P-T_{B1}),X_AT_{C2})$$
$$=e((X_AX_B+X_AX_C+X_BX_C)P,P)^{a_2b_2c_2}$$

Otherwise party A aborts.
4. Party B and party C compute these session keys in the similar way, here we omit the details.

4 Analysis of Liu's Three-Party Protocol

In this section, we show that Liu's protocol cannot provide perfect forward secrecy, and cannot resist key compromise impersonation attack.

4.1 No PFS

In this subsection, we show that Liu's three-party protocol cannot provide perfect forward secrecy. If the adversary learns long-term private keys X_A, X_B and X_C, the adversary can compute a_1, a_2 from S_{A1}, S_{A2} as follows:

$$S_{A1} = a_1 X_A + a_2 \Rightarrow a_2 = S_{A1} - a_1 X_A$$

$$\Downarrow$$

$$S_{A2} = a_2 X_A + a_1, a_2 = S_{A1} - a_1 X_A \Rightarrow S_{A2} = (S_{A1} - a_1 X_A) X_A + a_1$$

$$\Downarrow$$

$$a_1 = (S_{A2} - S_{A1} X_A)((X_A)^2 + 1)^{-1}$$

$$\Downarrow$$

$$a_2 = S_{A1} - (S_{A2} - S_{A1} X_A)((X_A)^2 + 1)^{-1} X_A.$$

Similarly, the adversary also can compute b_1, b_2 from S_{B1}, S_{B2} and c_1, c_2 from S_{C1}, S_{C2}. With these values $(X_A, X_B, X_C, a_1, a_2, b_1, b_2, c_1, c_2)$, the adversary can easily recover session keys $K_i (i = 1, ..., 8)$. It means that Liu's three-party multiple key agreement protocol cannot provide perfect forward secrecy.

4.2 KCI Attack

In this subsection, we assume the adversary learns the long-term key X_A. From subsection 4.1, we know that the adversary can compute the values a_1, a_2. If the adversary has past session transcripts, he can impersonate successfully party B and party C to cheat party A in the new session.

5 Review of Liu's Two-Party Protocol

In this section, we briefly review Liu's two-party protocol [4] proposed by Liu in 2010. Let P be a generator of a cyclic additive group G_1 of the prime order q, and G_2 be a cyclic multiplicative group of the prime order q. $e : G_1 \times G_1 \longrightarrow G_2$ is a bilinear pairing. Each party i has a static private key $X_i \in Z_q^*$ and a static public key $Y_i (= X_i P)$. For more details about the protocol, refer to [4].

In the following description we suppose that three communications parties, A and B wish to communicate with each other.

1. Party A chooses $a_1, a_2 \in Z_q^*$ randomly and computes $T_{A1} = a_1 Y_A$ and $T_{A2} = a_2 P Y_A$, Let K_{A1} and K_{A2} be the x-coordinate values of T_{A1} and T_{A2}. Then party A computes

$S_A = (a_1 K_{A1} + a_2 K_{A2}) T_{A1} + X_A T_{A12}$.

Finally, party A sends the message $(T_{A1}, T_{A2}, S_A, Cert(Y_A))$ to party B.

2. Similarly, party B chooses $b_1, b_2 \in Z_q^*$ randomly and computes $T_{B1} = b_1 Y_B$ and $T_{B2} = b_2 Y_B$, Let K_{B1} and K_{B2} be the x-coordinate values of T_{B1} and T_{B2}. Then party B computes

   $S_B = (b_1 K_{B1} + b_2 K_{B2}) T_{B1} + X_B T_{B2}$.

   Finally, party B sends the message $(T_{B1}, T_{B2}, S_B, Cert(Y_B))$ to party A.
3. Upon receiving the message $(T_{B1}, T_{B2}, S_B, Cert(Y_B))$, party A takes out the x-coordinate values K_{B1} and K_{B2} from T_{B1} and T_{B2}, checks whether

   $e(S_B, Y_B) = e(K_{B1} T_{B1} + K_{B2} T_{B2}, T_{B1}) e(T_{B2}, Y_B)$,

   if $e(S_B, Y_B) = e(K_{B1} T_{B1} + K_{B2} T_{B2}, T_{B1}) e(T_{B2}, Y_B)$, then computes the session keys K_1, K_2, K_3, K_4 as follows:

   $K_1 = e(a_1 X_A T_{B1}, Y_A + Y_B)$
   $K_2 = e(a_1 X_A T_{B2}, Y_A + Y_B)$
   $K_3 = e(a_2 X_A T_{B1}, Y_A + Y_B)$
   $K_4 = e(a_2 X_A T_{B2}, Y_A + Y_B)$

   Otherwise party A aborts.
4. Upon receiving the message $(T_{A1}, T_{A2}, S_A, Cert(Y_A))$, party B takes out the x-coordinate values K_{A1} and K_{A2} from T_{A1} and T_{A2}, checks whether

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

   if $e(S_A, Y_A) = e(K_{A1} T_{A1} + K_{A2} T_{A2}, T_{A1}) e(T_{A2}, Y_A)$, then computes the session keys K_1, K_2, K_3, K_4 as follows:

   $K_1 = e(b_1 X_B T_{A1}, Y_A + Y_B)$
   $K_2 = e(b_1 X_B T_{A2}, Y_A + Y_B)$
   $K_3 = e(b_2 X_B T_{A1}, Y_A + Y_B)$
   $K_4 = e(b_2 X_B T_{A2}, Y_A + Y_B)$

   Otherwise party B aborts.

6 Analysis of Liu's Two-Party Protocol

In this section, we show that Liu's protocol cannot provide perfect forward secrecy. The adversary E can carry out his attack as follows:

$K_1 = e(b_1 X_B T_{A1}, Y_A + Y_B) = e(b_1 X_B T_{A1}, Y_A + Y_B)$

$$= e(b_1 X_B T_{A1}, (X_A + X_B)P)$$

$$= e(b_1 X_B T_{A1}, (X_A + X_B)P)$$

$$= e(X_B T_{A1}, (X_A + X_B) b_1 P)$$

Since the adversary learns X_B and X_A, he can compute X_B^{-1}, then computes $b_1 P = X_B^{-1} T_{B1}$. Finally, he can recover the session key $K_1 = e(X_B T_{A1}, (X_A + X_B) b_1 P)$. In the similar way, the adversary also can recover K_2, K_3, K_4 if he can learns X_B and X_A.

It means that Liu's two-party protocol cannot provide perfect forward secrecy.

7 Conclusion

In this paper, we show Liu's protocols cannot satisfy the security properties as claimed. The three-party protocol cannot provide PFS and resist KCI attack, and the two-party protocol also cannot provide PFS.

References

1. Feng Liu, One-round and authenticated three-party multiple key exchange protocol from parings. Available at http://eprint.iacr.org/2010/239.
2. Lee N.Y., Wu C.N., Wang C.C., Authenticated multiple key exchange protocols based on elliptic curves and bilinear pairings. Computer Electronic Engneering 2008, 34(1): 12-20
3. Holbl M, Welzer T, Brumen B. Two proposed identity-based three-party authenticated key agreement protocols from pairings. Computers & Security 29(2010) 244-252
4. Feng Liu, Two improved authenticated multiple key exchange protocols. Available at http://eprint.iacr.org/2010/267.