Weakness of Several Identity-based Tripartite Authenticated Key Agreement Protocols

Xi-Jun Lin *and Lin Sun †

December 15, 2013

Abstract: Key agreement allows multi-parties exchanging public information to create a common secret key that is known only to those entities over an insecure network. In recent years, several identity-based authenticated key agreement protocols have been proposed. In this study, we analyze three identity-based tripartite authenticated key agreement protocols. After the analysis, we found that these protocols do not possess the desirable security attributes.

Key words: Information security; Authentication; Key agreement; Tripartite; Identity-based

1 Introduction

It is necessary to guarantee confidentiality, integrity and other security services for the communication parties over the open network. In order to achieve this goal, the communication parties always need to agree upon a session key. Key agreement is one of the fundamental cryptographic primitives which allows two or more parties to exchange information over an adversatively controlled insecure network and agree upon a common session key. After that, this shared session key may be used for later secure communication among these parties.

One research line of key agreement is to generalize the two-party key agreement into multi-party setting, amongst which the tripartite key agreement receives much interest. Joux [1] presented a tripartite key agreement protocol employing pairings. The protocol is round-efficient since only one broadcast is required for each entity. However, his protocol does not provide authentication of the three communicating entities. To remove the weaknesses in Joux's protocol, many public key-based tripartite protocols were proposed, however they require a large amount of computation time and storage. Recently, identity-based tripartite authenticated key agreement protocol has rapidly emerged and been well-studied as well, but many of them have turned out to be flawed. Some flaws have taken years to discover. The desirable security attributes of a key agreement protocol are as follows:

- *Known-Key Security* : The protocol achieves its goal despite the fact that an adversary learned some previous session keys.

*X.J.Lin is with the Department of Computer Sciences and Technology, Ocean University of China. Qingdao 266100, P.R.China. email: linxj77@163.com

†L.Sun is with the College of Liberal Arts, Qingdao University. Qingdao 266071, P.R.China. email: sunlin9@126.com

- *Forward Secrecy* : The secrecy of the previous session keys is not affected, if the long-term private keys of one or more parties are compromised.
- *Key-Compromise Impersonation* : An adversary, who has gained the long-term private key of one party (e.g. A), cannot impersonate other parties to the party, whose private key has been disclosed (e.g. A).
- *Unknown Key-Share* : An adversary cannot convince a group of parties that they share a key with the adversary, whereas in fact they share a key with another party.
- *Key Control* : The session key should be determined by all intended parties.

Note that an authenticated key agreement protocol should also be secure against message replay attack, reflect attack and Parallel session attack, etc.

In this study, we examine three tripartite authenticated key agreement protocols. We show that these protocols do not possess the desirable security attributes. We remark that the purpose of this paper is not to investigate the design of the protocols neither to repair the security flaws, but simply to show by example how difficult is to design a secure cryptographic protocol.

2 Preliminaries

2.1 Definition of identity-based tripartite authenticated key agreement protocol

An identity-based tripartite authenticated key-agreement protocol consists of three polynomial-time algorithms: *Setup*, *Extract* and *Key Agreement*. These algorithms are defined as follows.

- *Setup* : This algorithm is run by PKG. It takes as input a security parameter and returns a master key and a list of system parameters.
- *Extract* : This algorithm is also run by PKG. It takes as input the parameters, master key and an entity's identity ID_i, to produce and issue the entity's private key SID_i to the entity ID_i secretly.
- *Key Agreement* : This is a probabilistic polynomial-time interactive algorithm which involves three entities A, B and C. The inputs are the system parameters, the private keys and identities of A, B and C. Eventually, if the protocol does not fail, A, B and C obtain a secret session key K.

2.2 Bilinear Pairing

Let $\mathbb{G}_1$ be an additive group of prime order p, $\mathbb{G}_2$ be a multiplicative group of the same order. Bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ which satisfies the following properties:

- Bilinearity: given any $g, h \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab} = \hat{e}(g^{ab}, g), etc.$

- Non-Degeneracy: There exists a $g \in \mathbb{G}_1$ such that $\hat{e}(g, g) \neq 1$.
- Computability: $\hat{e}(g, h)$ can be computed in polynomial time.

3 Xiong et al.'s protocol

In this section, we recall Xiong et al.'s protocol [5] as follows. Then, we give an attack to break their protocol.

- *Setup* : Given a security parameter $k \in \mathbb{Z}$, the algorithm works as follows:
  1. Run the parameter generator on input k to generate a prime q, two groups $\mathbb{G}_1, \mathbb{G}_2$ of prime order q, a generator P of $\mathbb{G}_1$ and pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
  2. Select a master key $x \in_R \mathbb{Z}_q^*$, and compute $P_{pub} = xP$.
  3. Choose cryptographic hash functions $H_1 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \to \{0,1\}^k$.

  Finally the PKG's master key x is kept secret and the system parameters $(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2)$ are published.
- *Extract* : Given a user's identity $ID_U \in \{0,1\}^*$, PKG first chooses at random $r_U \in_R \mathbb{Z}_q^*$, then computes $R_U = r_U P$, $h = H_1(ID_U \| R_U)$ and $s_U = (r_U + hx)^{-1}$. It then sets this user's private key (s_U, R_U) and transmits it to user ID_i secretly.
- *Key Agreement* : The message flows and computations of a protocol run are described below.
  1. A, B, C: choose $a, b, c \in_R \mathbb{Z}_q^*$.
  2. $A \to B, C$: ID_A, R_A
     $B \to A$: $ID_B, R_B, T_{BA} = b(R_A + H_1(ID_A \| R_A)P_{pub})$
     $C \to A$: $ID_C, R_C, T_{CA} = c(R_A + H_1(ID_A \| R_A)P_{pub})$
     $A \to B$: $T_{AB} = a(R_B + H_1(ID_B \| R_B)P_{pub})$
     $A \to C$: $T_{AC} = a(R_C + H_1(ID_C \| R_C)P_{pub})$
     $B \to C$: ID_B, R_B
     $C \to B$: $ID_C, R_C, T_{CB} = c(R_B + H_1(ID_B \| R_B)P_{pub})$
     $B \to C$: $T_{BC} = b(R_C + H_1(ID_C \| R_C)P_{pub})$
  3. A computes:
     $K^1 = aP + s_A T_{BA} + s_A T_{CA} = aP + bP + cP = (a + b + c)P$
     $K^2 = \hat{e}(s_A T_{BA}, s_A T_{CA})^a = \hat{e}(bP, cP)^a = \hat{e}(P, P)^{abc}$
     B computes:
     $K^1 = bP + s_B T_{AB} + s_B T_{CB} = bP + aP + cP = (a + b + c)P$
     $K^2 = \hat{e}(s_B T_{AB}, s_B T_{CB})^b = \hat{e}(aP, cP)^b = \hat{e}(P, P)^{abc}$
     C computes:
     $K^1 = cP + s_C T_{AC} + s_C T_{BC} = cP + aP + bP = (a + b + c)P$
     $K^2 = \hat{e}(s_C T_{AC}, s_C T_{BC})^c = \hat{e}(aP, bP)^c = \hat{e}(P, P)^{abc}$

After the protocol has finished, all three entities share the session key which is computed as $K = H_2(ID_A \| ID_B \| ID_C \| T_{AB} \| T_{AC} \| T_{BA} \| T_{BC} \| T_{CA} \| T_{CB} \| K^1 \| K^2)$.

Cryptanalysis: Let an adversary E, whose identity is ID_E, be a legitimate user in this system and his private key is $(s_E, R_E) = ((r_E + hx)^{-1}, r_E P)$, where $h = H_1(ID_E \| R_E)$. Since R_B is sent in plaintext, it is reasonable to assume that E can obtain R_B when B participants in a protocol run with other entities.

Then, E can fool A into believing that they have participated in a protocol run with B, but in fact B has not been active. The attack is as follows.

In *Key Agreement* phase, E initiates a protocol run with A, and at the same time impersonates B (Let $E(B)$ denote the entity B impersonated by E). They perform with the following steps.

1. E: choose $e, b' \in_R \mathbb{Z}_q^*$.
   A: choose $a \in_R \mathbb{Z}_q^*$.
2. $E \to A$: ID_E, R_E
   $A \to E$: $ID_A, R_A, T_{AE} = a(R_E + H_1(ID_E \| R_E)P_{pub})$
   $E(B)$ broadcasts: $ID_B, R_B, T_{E(B)E} = b'(R_E + H_1(ID_E \| R_E)P_{pub})$
   $E \to A$: $T_{EA} = e(R_A + H_1(ID_A \| R_A)P_{pub})$
   E broadcasts: $T_{EE(B)} = e(R_B + H_1(ID_B \| R_B)P_{pub})$
   $A \to E(B)$: ID_A, R_A
   $E(B) \to A$: $ID_B, R_B, T_{E(B)A} = b'(R_A + H_1(ID_A \| R_A)P_{pub})$
   $A \to E(B)$: $T_{AE(B)} = a(R_B + H_1(ID_B \| R_B)P_{pub})$
3. A computes:
   $K^1 = aP + s_A T_{EA} + s_A T_{E(B)A} = aP + eP + b'P = (a + e + b')P$
   $K^2 = \hat{e}(s_A T_{EA}, s_A T_{E(B)A})^a = \hat{e}(eP, b'P)^a = \hat{e}(P, P)^{aeb'}$
   E computes:
   $K^1 = eP + s_E T_{AE} + b'P = eP + aP + b'P = (a + e + b')P$
   $K^2 = \hat{e}(s_E T_{AE}, b'P)^e = \hat{e}(aP, b'P)^c = \hat{e}(P, P)^{aeb'}$

After the protocol has finished, the session key which is computed as

$$K = H_2(ID_A \| ID_B \| ID_E \| T_{AE(B)} \| T_{AE} \| T_{E(B)A} \| T_{E(B)E} \| T_{EA} \| T_{EE(B)} \| K^1 \| K^2).$$

Hence, E can send message to A by impersonating B. □

4 Tan's protocol

In this section, we recall Tan's protocol [4] as follows. Then, we give an attack to break their protocol.

- *Setup* : Given a security parameter k, the algorithm generates the system parameters as follows.

  1. Generate a group $\mathbb{G}_1$ with a generator P of prime order q over the elliptic curve, a group $\mathbb{G}_2$ and pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
  2. PKG chooses $s \in_R \mathbb{Z}_q$ as the master key, and computes $P_{pub} = sP$.
  3. Choose cryptographic hash functions $H_1 : \mathbb{G}_1 \times \{0,1\}^* \to \mathbb{Z}_q$, $H_2 : \{0,1\}^* \times \mathbb{G}_1 \times \{0,1\}^* \to \mathbb{Z}_q$ and $H_3 : \{0,1\}^* \to \{0,1\}^k$.

  The system public parameters are $(\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, H_1, H_2, H_3, P_{pub})$.

- *Extract* : Given a user A's identity $ID_A \in \{0,1\}^*$, PKG chooses $r_1 \in_R \mathbb{Z}_q$ and computes $R_1 = r_1 P$, $x = r_1 + sH_1(ID_A, R_1)$. PKG outputs the secret key (x, R_1) to A. Similarly, the entity B and C obtains the secret key (y, R_2) and (z, R_3), respectively.

- *Key Agreement* : Let $ID = \{ID_A, ID_B, ID_C\}$. A, B and C performs as follows.

  1. A chooses $a \in_R \mathbb{Z}_q$ and computes $V_A = aP, \sigma_1 = a + xH_2(ID, V_A, T_A) \pmod q$, where T_A is A's timestamp. Then A broadcasts $(ID_A, V_A, T_A, \sigma_1, R_1)$.
  2. B chooses $b \in_R \mathbb{Z}_q$ and computes $V_B = bP$, $\sigma_2 = b + yH_2(ID, V_B, T_B) \pmod q$, where T_B is B's timestamp. Then B broadcasts $(ID_B, V_B, T_B, \sigma_2, R_2)$.
  3. C chooses $c \in_R \mathbb{Z}_q$ and computes $V_C = cP$, $\sigma_3 = c + zH_2(ID, V_C, T_C) \pmod q$, where T_C is C's timestamp. Then C broadcasts $(ID_C, V_C, T_C, \sigma_3, R_3)$.

  (*Remark* : In the original paper, R_1, R_2 and R_3 are not broadcasted. We correct these typos.)

  After receiving the messages broadcasted by the other two entities, A, B and C performs respectively as follows.

  1. A checks if T_B and T_C fall within the valid time period. If either is out of the time expectation, A aborts. Otherwise, A checks if the following equations hold.

  $$\sigma_2 P \stackrel{?}{=} V_B + H_2(ID, V_B, T_B)(R_2 + H_1(ID_B, R_2)P_{pub}),$$

  $$\sigma_3 P \stackrel{?}{=} V_C + H_2(ID, V_C, T_C)(R_3 + H_1(ID_C, R_3)P_{pub}).$$

  If they hold, A computes $K = \hat{e}(V_B, V_C)^a$. Otherwise, A aborts.
  2. B checks if T_A and T_C fall within the valid time period. If either is out of the time expectation, B aborts. Otherwise, B checks if the following equations hold.

  $$\sigma_1 P \stackrel{?}{=} V_A + H_2(ID, V_A, T_A)(R_1 + H_1(ID_A, R_1)P_{pub}),$$

  $$\sigma_3 P \stackrel{?}{=} V_C + H_2(ID, V_C, T_C)(R_3 + H_1(ID_C, R_3)P_{pub}).$$

  If they hold, B computes $K = \hat{e}(V_A, V_C)^b$. Otherwise, B aborts.

3. C checks if T_A and T_B fall within the valid time period. If either is out of the time expectation, C aborts. Otherwise, C checks if the following equations hold.

$$\sigma_1 P \stackrel{?}{=} V_A + H_2(ID, V_A, T_A)(R_1 + H_1(ID_A, R_1)P_{pub}),$$

$$\sigma_2 P \stackrel{?}{=} V_B + H_2(ID, V_B, T_B)(R_2 + H_1(ID_B, R_2)P_{pub}).$$

   If they hold, C computes $K = \hat{e}(V_A, V_B)^c$. Otherwise, C aborts.

4. Finally, they compute a session key $SK = H_3(ID \| V_A \| V_B \| V_C \| K)$.

Cryptanalysis: Let an adversary E, whose identity is ID_E, be a legitimate user in this system and his private key is $(R_4, w) = (r_4 P, r_4 + sH_1(ID_E, R_4))$.

Suppose E, A and B participant in a protocol run *Round1* previously, and $\delta_2 = (ID_B, V_B, T_B, \sigma_2, R_2)$ is broadcasted by B. With δ_2, E can initiate a new protocol run *Round2* with A by impersonating B to fool A into believing that he has participated in *Round2* with E and B, but in fact B has not been active in *Round2*.

Suppose T_B is still fall within the valid time period. Let $ID = \{ID_A, ID_B, ID_E\}$. The attack is as follows.

In the *Key Agreement* phase, A and E perform as follows.

1. A chooses $a \in_R \mathbb{Z}_q$ and computes $V_A = aP$, $\sigma_1 = a + xH_2(ID, V_A, T_A) \pmod q$, where T_A is A's timestamp. Then A broadcasts $(ID_A, V_A, T_A, \sigma_1, R_1)$.

2. E chooses $e \in_R \mathbb{Z}_q$ and computes $V_E = eP$, $\sigma_4 = e + wH_2(ID, V_E, T_E) \pmod q$, where T_E is E's timestamp. Then E broadcasts $(ID_E, V_E, T_E, \sigma_4, R_4)$.

   Furthermore, E sends δ_2 to A by impersonating B.

Then, A and E performs as follows.

1. Since δ_2 is a valid message sent by B previously, we have that $V_B = bP$, $\sigma_2 = b + yH_2(ID, V_B, T_B) \pmod q$. Then, the equation $\sigma_2 P = V_B + H_2(ID, V_B, T_B)(R_2 + H_1(ID_B, R_2)P_{pub})$ holds.

   Similarly, the equation $\sigma_4 P = V_E + H_2(ID, V_E, T_E)(R_4 + H_1(ID_E, R_4)P_{pub})$ holds. Hence, A computes $K = \hat{e}(V_B, V_E)^a = \hat{e}(P, P)^{abe}$.

2. Since E knows V_B and e, he can compute the same value $K = \hat{e}(V_A, V_B)^e = \hat{e}(P, P)^{abe}$

3. Finally, they compute the same session key $SK = H_3(ID \| V_A \| V_B \| V_E \| K)$.

Hence, E can send message to A by impersonating B. □

5 Shim's Protocol

Nalla proposed an identity-based tripartite authenticated key agreement protocol with signatures [2], but was broken by Shim. Shim improved Nalla's protocol [3]. However we point out that Shim's improved protocol is still insecure.

Shim's protocol is given below.

1. *Setup* : Choose a large prime p. Let $\mathbb{E}$ be a supersingular curve defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$. Let H_1 and H be a collision resistant hash function $H_1, H : \{0,1\}^* \to \mathbb{F}_p$. Let μ_q be the subgroup of $\mathbb{F}^*_{p^2}$ contains all elements of order q. The modified Weil pairing is defined by

$$\hat{e} : \mathbb{G}_q \times \mathbb{G}_q \to \mu_q, \hat{e}(P, Q) = \hat{e}(P, \phi(Q)).$$

where $\phi(x, y) = (\zeta x, y)$, $1 \neq \zeta \in \mathbb{F}^*_{p^2}$ is a solution of $x^3 - 1 = 0 \pmod{p}$ and $\mathbb{G}_q$ is a group of points with order q. Let P be a generator of $\mathbb{G}_q$. The key generation center (KGC) chooses a random $s \in \mathbb{Z}_q$ and set $P_{KGC} = sP$. The KGC publishes the system parameters $(p, q, \mathbb{E}, P, P_{KGC}, \hat{e}, H_1, H)$ and keep s as the secret master key.

2. *Extract* : A user submits his identity information ID to KGC. KGC computes the user's public key as $Q_{ID} = H(ID)$ and returns $S_{ID} = sQ_{ID}$ to the user as his private key.

3. *Key Agreement* : A, B and C respectively choose random numbers a, b and c, and compute (U_A, V_A), (U_B, V_B) and (U_C, V_C) and broadcast these values as follows.

   (1) $A : U_A = aP, V_A = H(U_A)S_A + aP_{KGC}$

   (2) $B : U_B = bP, V_B = H(U_B)S_B + bP_{KGC}$

   (3) $C : U_C = cP, V_C = H(U_C)S_C + cP_{KGC}$

   After receiving the messages broadcasted from the other two participants, A, B and C respectively perform as follows.

   (1) A verifies $\hat{e}(V_B + V_C, P) \stackrel{?}{=} \hat{e}(P_{KGC}, H(U_B)Q_B + H(U_C)Q_C + U_B + U_C)$. If the equation holds, A computes $k_A = \hat{e}(U_B, U_C)^a = \hat{e}(P, P)^{abc}$.

   (2) B verifies $\hat{e}(V_A + V_C, P) \stackrel{?}{=} \hat{e}(P_{KGC}, H(U_A)Q_A + H(U_C)Q_C + U_A + U_C)$. If the equation holds, B computes $k_B = \hat{e}(U_A, U_C)^b = \hat{e}(P, P)^{abc}$.

   (3) C verifies $\hat{e}(V_A + V_B, P) \stackrel{?}{=} \hat{e}(P_{KGC}, H(U_A)Q_A + H(U_B)Q_B + U_A + U_B)$. If the equation holds, C computes $k_C = \hat{e}(U_A, U_B)^c = \hat{e}(P, P)^{abc}$.

Cryptanalysis: Let an adversary E, whose identity is ID_E, be a legitimate user in this system and his private key is $S_E = sQ_E$, where $Q_E = H(ID_E)$. E can initiate a protocol run with A by impersonating B to fool A into believing that he has participated in a protocol run with E and B, but in fact B has not been active. The attack is given below.

In *Key Agreement* phase, A chooses a random number a, and then computes and broadcasts (U_A, V_A) as follows.

$$U_A = aP, V_A = H(U_A)S_A + aP_{KGC}$$

E chooses a random number b, and then computes (U_B, V_B) and (U_E, V_E) as follows.

$$U_B = bP, V_B = bP_{KGC}, U_E = -H(U_B)Q_B, V_E = H(U_E)S_E$$

E broadcasts (U_E, V_E) with his true identity, and broadcasts (U_B, V_B) by impersonating B.

Then, A verifies whether the equation $\hat{e}(V_B+V_E, P) = \hat{e}(P_{KGC}, H(U_B)Q_B+H(U_E)Q_E+U_B+U_E)$ holds. We have that

$$\begin{aligned}
\hat{e}(V_B+V_E, P) &= \hat{e}(bP_{KGC}+H(U_E)S_E, P) \\
&= \hat{e}(bP+H(U_E)Q_E, P_{KGC}) \\
&= \hat{e}(H(U_B)Q_B+bP+H(U_E)Q_E-H(U_B)Q_B, P_{KGC}) \\
&= \hat{e}(P_{KGC}, H(U_B)Q_B+H(U_E)Q_E+bP+U_E) \\
&= \hat{e}(P_{KGC}, H(U_B)Q_B+H(U_E)Q_E+U_B+U_E)
\end{aligned}$$

It is clear that (U_B, V_B) and (U_E, V_E) are valid protocol messages. Then, A computes the session key $k_A = \hat{e}(U_B, U_E)^a = \hat{e}(P, U_E)^{ab}$. E can also compute the same session key $k_E = \hat{e}(U_A, U_E)^b = \hat{e}(P, U_E)^{ab}$.

Hence, E can send message to A by impersonating B. □

6 Conclusion

In this paper, we have highlighted the security flaws of Xiong et al.'s, Tan's and Shim's protocols. These protocols do not possess the desirable security attributes.

References

[1] A.Joux. A one round protocol for tripartite Diffie-Hellman. Proceedings of the 4th International Symposium on Algorithmic Number Theory. 2000. pp: 385-394.

[2] D.Nalla. ID-based tripartite key agreement with signatures. available at http://eprint.iacr.org/2003/144.

[3] K.Shim. Cryptanalysis of ID-based tripartite authenticated key agreement protocols. available at http://eprint.iacr.org/2003/115.

[4] Z.Tan. An efficient identity-based tripartite authenticated key agreement protocol. Electron Commer Res. 2012(12). pp:505-518.

[5] H.Xiong, Z.Chen and F.Li. New identity-based three-party authenticated key agreement protocol with provable security. Journal of Network and Computer Applications. 2013(36). pp:927-932.