Comments: Insider attack on Cheng et al.'s pairing-based tripartite key agreement protocols

Hung-Yu Chien

Department of Information Management

ChaoYang University of Technology , Taiwan, R.O.C.

hychien@cyut.edu.tw

Abstract

Recently, Cheng et al. proposed two tripartite key agreement protocols from pairings: one is certificate-based and the other is identity-based (ID-based). In this article, we show that the two schemes are vulnerable to the insider impersonation attack and the ID-based scheme even discloses the entities' private keys. Solutions to this problem are discussed.

1. Introduction

Initially, the pairing was used to reduce the DLP problem on some elliptic curves (e.g., the super-singular curves) to the DLP problem on some finite field. It, therefore, hinders cryptographers from building cryptosystems on these curves. The situation changed when Joux [5] proposed the first tripartite key agreement protocol, using the pairing. Since then, many pairing-based tripartite key agreement schemes have been proposed [1-4, 6-12]. A tripartite key agreement protocol allows three parties establish session keys among them. The three-party (or tripartite) case is of most practical importance not only because it is the most common size for electronic conferences but also because it can be used to provide a range of services for two parties communicating. For example, a third party can be added to chair, or referee a conversation for ad hoc auditing, data recovery or escrow purposes [1, 2, 5]. It can

also facilitate the job of group communication. Joux's tripartite key agreement protocol [5] is efficient. However, the protocol does not authenticate the messages, and, therefore, cannot resist the basic man-in-the-middle attack. And, most of previous schemes were shown to be insecure [1-2, 6-12].

These tripartite key agreement protocols can be divided into two broad categories - certificated-based [1-3, 5, 9] and ID-based [4, 6, 7, 8, 10, 12]. Recently, Cheng et al. [3] proposed two tripartite key agreement protocols: one is certificate-based and the other is ID-based. They also analyzed their schemes against a list of attacks. However, some of their attack assumptions are not practical, but their schemes cannot resist the practical insider impersonation attack and their ID-based version even discloses the entities' private keys.

In this article, we will not go through the list of basic attacks, but will focus on the insider impersonation attack and private key disclosure on Cheng et al.'s schemes, because the other attacks have been extensively discussed in many articles but the insider impersonation attack is specific to *n*-party ($n \geq 3$) key agreement protocols. Actually, Al-Riyami and Paterson [1-2] have pointed out this issue, but few researchers noticed this problem.

2. Review of Cheng et al.'s schemes

The two schemes own the same system initialization but different public key/private key issuing processes and different key agreement protocols.

Initialization: CA sets up an additive group G_1 of prime order q and a cyclic multiplicative group G_2 of the same order q. The discrete logarithm problems (DLP) in both G_1 and G_2 are assumed to be hard. Let P be a generator of G_1.

Define one cryptographic hash functions $H:\{0,1\}^* \to Z_q$. CA owns the system's private key $s \in Z_q^*$ and the system's public key $P_{pub} = sP$. CA publishes $\{G_1, G_2, P, q, P_{pub}, H\}$ and a bilinear pairing *e*().

e: $G_1 \text{x} G_1$ → G_2 be a bilinear mapping satisfying the following conditions.

1. Bilinear：Let $a, b \in Z$ and $P, Q \in G_1$, $e(a \cdot P, b \cdot Q) = e(P, Q)^{ab}$.
2. Non-degenerate：There exists $P \in G_1$ such that $e(P, P) \neq 1 \in G_2$.
3. Polynomial-time computable：The mapping function $e(P, Q)$ is computable in polynomial time.

Bilinear Diffie-Hellman Problem (BDHP) for a bilinear pairing e: $G_1 \text{x} G_1$ → G_2 is defined as follows: Given $P, aP, bP, cP \in G_1$, compute $e(P, P)^{abc}$, where a, b, c are random numbers from Z_q^*. It is commonly believed that the BDHP problem is hard.

2.1 The certificate-based version

In addition to the system initialization, CA performs the following certificate issuing process.

Certificate issuing: The certificate for entity *A* will be of the form $Cert_A$ = ($I_A \| Y_A \| P \| SIG_{CA}(I_A \| Y_A \| P)$), where I_A denotes the identity string of *A*, || denotes the concatenation operation, and SIG_{CA} denotes the CA's signature. Entity *A*'s long-term public key is $Y_A = xP$, where $x \in Z_q^*$ is the long-term private key of *A*. Similarly $Cert_B$ and $Cert_C$ are the certificates for entities *B* and *C*, with $Y_B = yP$ and $Y_C = zP$ as their long-term public keys.

In the protocol below, $a,b,c \in Z_p^*$ are randomly ephemeral values selected by *A*, *B* and *C* respectively.

Key agreement protocol

$A \rightarrow B,C$: $T_A^1 = aP$, $T_A^2 = a(xP)$, $Cert_A$ (1)

$B \rightarrow A,C$: $T_B^1 = bP$, $T_B^2 = b(yP)$, $Cert_B$ (2)

$C \rightarrow B, A$: $T_C^1 = cP$, $T_C^2 = c(zP)$, $Cert_C$ (3)

After exchanging the messages, *A* verifies *B*'s and *C*'s messages by checking whether the equations $e(T_B^1, yP) \stackrel{?}{=} e(T_B^2, P)$ and $e(T_C^1, zP) \stackrel{?}{=} e(T_C^2, P)$ hold. If the verifications succeed, *A* computes $K_A = e(T_B^2, T_C^2)^{ax} = e(P,P)^{axbycz}$. Similarly, *B* and C perform similar checking. After that, *B* computes $K_B = e(T_A^2, T_C^2)^{by} = e(P,P)^{axbycz}$ and *C* computes $K_C = e(T_B^2, T_A^2)^{cz} = e(P,P)^{axbycz}$. To resist some attacks, the final session key is defined as $K = kdf(K_A)$, where *kdf*() is some pre-defined key derivation function.

2.2 The ID-based version

In addition to the system initialization, CA publishes one more hashing function $H_1 : \{0,1\}^* \rightarrow G_1$ and performs the following private key issuing process.

Private key extraction: Let A, B and C be the three entities running the protocol. A has his public key/private key as $Q_A = H_1(ID_A) / S_A = sQ_A$, B has his public key/private key as $Q_B = H_1(ID_B) / S_B = sQ_B$, and C has his public key/private key as $Q_C = H_1(ID_C) / S_C = sQ_C$.

Key agreement protocol

$A \rightarrow B,C$: $T_A^1 = aP_{pub}$, $T_A^2 = H(T_A^1)S_A$ (4)

$B \rightarrow A,C$: $T_B^1 = bP_{pub}$, $T_B^2 = H(T_B^1)S_B$ (5)

$C \rightarrow B,A$: $T_C^1 = cP_{pub}$, $T_C^2 = H(T_C^1)S_C$ (6)

After exchanging the messages, A verifies $e(P_{pub}, H(T_B^1)Q_B) \stackrel{?}{=} e(P, T_B^2)$ and $e(P_{pub}, H(T_C^1)Q_C) \stackrel{?}{=} e(P, T_C^2)$. If the verifications succeed, A computes $K_A = e(T_B^1, T_C^1)^a = e(P,P)^{abcs^2}$. Similarly, B and C perform their checking. After that, B computes $K_B = e(T_A^1, T_C^1)^b = e(P,P)^{abcs^2}$, and C computes $K_C = e(T_A^1, T_B^1)^c = e(P,P)^{abcs^2}$. The final session key is defined as $K = kdf(K_A)$, where *kdf*() is some pre-defined key derivation function.

3. Comments on Cheng et al.'s schemes

Insider impersonation attack: In the two-party case, only an outsider would impersonate the communicating parties. However, in the *n*-party case for $n \geq 3$, any entity of the communicating group might impersonate another entity to the rest entities of the group. This kind of impersonation attack is called the *insider impersonation attack*. This threat could result in serious loss. For example, the impersonated party is the referee or the auditor.

3.1 Insider attack on Cheng et al.'s certificate-based scheme

Cheng et al. have noticed the insider impersonation attack since any entity, for

example *A*, can just relay *B*'s message to cheat *C* and shares a session key with *C*. Cheng et al., therefore, proposed that "the three parties should negotiate a session related unique information, e.g., a session counter, and securely bind the negotiated unique session information with messages".

However, we comment that Cheng et al.'s solution is not practical, not secure and unwieldy. One possible way to securely bind the session counter with messages is to sign on them, for example $SIG_B(messages \,||\, session\ counter)$. This approach requires each entity to maintain $\binom{n}{3}$ session counters if there are *n* entities in the system. This number $\binom{n}{3}$ is even larger than the number $\binom{n}{2}$ in the two-parties case, for $n>5$. This causes the scheme not practical and unwieldy. Further, one entity can replay the messages and signatures to launch the insider attack, since the signature does not securely bind the messages with the communicating entities' identities. For example, *A* can eavesdrop *B*'s messages and signatures $SIG_B(messages \,||\, session\ counter)$ in the session among *A*, *B*, and *C*. Later *A* replays *B*'s messages and signatures $SIG_B(messages \,||\, session\ counter)$ in another session among *A*, *B*, and *D*, if the session counter is matched. It is easy to launch the attack since *A* can easily eavesdrop *B*'s signatures as many as he/she wishes and it is easy to find the matched session counters. To secure bind a message with its corresponding session, we argue that the entity should sign on the message, the communicating entities' identities, and the session counter (or the timestamp if time synchronization is feasible).

3.2 Insider attack and private key disclosure of Cheng et al.'s ID-based version

Like the certificate-based version, Cheng et al.'s ID-based scheme still cannot resist the insider attack as described above. Further, the ID-based version discloses

one entity's private key.

In the messages (4-6), T_A^2, T_B^2 and T_C^2 act as *A*'s, *B*'s and *C*'s signatures on their ephemeral values. However, any one can easily derive *A*'s (*B*'s and *C*'s) private key by computing $S_A = H(T_A^1)^{-1} \cdot T_A^2$, since $T_A^2 = H(T_A^1)S_A$ and T_A^1 is publicly transmitted. Once the private key is compromised, the whole system is not secure.

4. Conclusions and remarks

In this paper, we have shown the insider attack on Cheng et al.'s tripartite key agreement schemes and the private key disclosure of their ID-based version. We focus on the insider attack because it is specific to *n*-party case (for $n>2$) and this issue was neglected by many researchers. To counter this attack, Al-Riyami and Paterson proposed two approaches: one is to sign the ephemeral values and the timestamp and the other is to design tripartite key confirmation protocols. However, Al-Riyami and Paterson's first approach should be amended to be secure. Assume *A*, *B*, and *C* are initiating a new session. Now the entity *A* eavesdrops *B*'s ephemeral values, timestamp and signatures in one session, and immediately initiates another session with *D* by replaying B's messages and signatures. This implies that the communicating entities' identities should be explicitly included in the signed data. Considering the time synchronization and performance, the key confirmation approach would be more desirable.

References

[1] Al-Riyami, S. and Paterson, K., "Authenticated three party key agreement protocols from pairings", Cryptology ePrint Archive, Report 2002/035.

[2] Al-Riyami, S. S. and Paterson, K. G., “Tripartite Authenticated Key Agreement Protocols from Pairings”, IMA Conference on Cryptography and Coding, LNCS 2898, Springer-Verlag (2003), pp. 332-359.

[3] Cheng, Z., Vasiu, L., and Comley, R., “Pairing-based one-round tripartite key agreement protocols”, Cryptology ePrint Archive, Report 2004/079, available at http://eprint.iacr.org/2004/079/.

[4] Chien, H.Y., “Improved ID-based Tripartite Multiple Key Agreement Protocol from Pairings”, Proceedings of the 14th Information Security Conference, (2004), Taiwan.

[5] Joux, A., “A one round protocol for tripartite Diffie-Hellman”, ANTS IV 2000, LNCS1838, Spring-Verlag, pp. 385-394.

[6] Liu, S., Zhang, F., Chen, K., “ID-based tripartite key agreement protocol with pairing”, In Proc. of IEEE ISIT 2003, Yokohama, Japan, 2003, pp. 136.

[7] Nalla, D., “ID-based tripartite key agreement with signatures”, Cryptology ePrint Archive, Report 2003/144, available at http://eprint.iacr.org/2003/144/.

[8] Nalla, D. and Reddy, K.C., “ID-based tripartite authenticated key agreement protocols from pairings’, Cryptology eprint Archive, Report 2003/004.

[9] Shim, K., ”Efficient one round tripartite authenticated key agreement protocol from Weil pairing”, Electron. Lett., 2003, 39(2), pp.208-209.

[10] Shim, K. “A man-in-the-middle attack on Nalla-Reddy’s ID-based tripartite authenticated key agreement protocol,” Cryptology ePrint Archive, Report 2003/115.

[11] Sun, H.-M., and Hsieh, B.T., “Security analysis of Shim’s authenticated key agreement protocols from pairings,” Cryptology ePrint Archieve, Report 2003/113.

[12] Zhang, F., Liu, S., and Kim, K., “ID-based one-round authenticated tripartite key agreement protocol with pairings”, Cryptology eprint Archive, Report 2002/122.