An Improvement of Certification-based One-Round Tripartite Key Agreement Protocols

Kambombo Mtong[1] and Eun-Jun Yoon[2]

[1] Department of IT Convergence, Kyungil University / Gyeongsan, South Korea kambombomtonga@yahoo.com
[2] Department of Cyber Security, Kyungil University / Gyeongsan, South Korea ejyoon@kiu.ac.kr

***** Corresponding Author: Eun-Jun Yoon

Received July 14, 2013; Revised July 29, 2013; Accepted August 12, 2013; Published October 31, 2013

** Short Paper*

***Abstract*:** Key agreement protocols allow multi-parties exchanging public information to create a common secret key that is known only to those entities over an insecure network. Since Joux first published the pairing-based one round tripartite key agreement protocol, many authenticated protocols have been proposed. Unfortunately, many of them have been broken while others have been shown to be deficient in some desirable security attributes. In 2004, Cheng et al. presented two protocols aimed at strengthening Shim's certificate-based and Zhang et al.'s tripartite identity-based protocols. This paper reports that 1) In Cheng et al.'s identity-based protocol, an adversary can extract long-term private keys of all the parties involved; and 2) Cheng et al.'s certification-based protocol is weak against key integrity attacks. This paper suggests possible remedies for the security flaws in both protocols and then presents a modified Cheng et al.'s identity-based, one-round tripartite protocol that is more secure than the original protocol.

***Keywords*:** Key agreement protocol, Key integrity attack, Session key exposure, Weil pairing

1. Introduction

Key agreement remains one of the fundamental cryptographic attributes that enables two or more parties to exchange information over an adversary controlled insecure network and agree upon a common session key. Normally, the established key varies on each execution (session) of the protocol. In a key agreement protocol, if one party is assured that no other party (or parties) can gain access to the particular established key, aside from the specifically identified party (or parties), then the protocol can be said to provide key authentication. A key agreement protocol is called an authenticated key agreement protocol if it achieves mutual key authentication between (or among) the involved parties. On the other hand, in authenticated key agreement protocols, one party is not entirely sure that the other party is in possession of the established key. The property of key confirmation in key agreement protocols is that one party is sure that the other party is actually in possession of the established key. A key agreement protocol that achieves both key authentication and key confirmation is called an authenticated key agreement with key confirmation [2].

A number of security attributes that can be used to analyze the security of key agreement protocols have been identified [3, 4]. The following discusses these properties:

- ***Known Session Key Security:*** This property requires that the compromise of one session key should not compromise the keys of the other sessions, whether in parallel, previous or future sessions.
- ***Forward Secrecy:*** This property requires that if the long-term private keys of one or more entities are compromised, the secrecy of the previously established session keys should not be a33ected. A protocol is said to achieve partial forward secrecy if one or more but not all the entities' long-term keys can be corrupted without compromising the previously established session keys, whereas a protocol achieves perfect forward secrecy (PFS) if the

Preliminary results of this paper were presented at the KICS 2013 winter conference. This present paper is an improved version which has been extended to include a modified algorithm and comprehensive simulation results. This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(No. 2010-0010106).

long-term keys of all the entities involved may be corrupted without compromising the session key established previously by these entities.

- ***Key-Compromise Impersonation Resilience:*** If party *A*'s long-term private key is compromised, an adversary who knows the *A*'s long-term private key can impersonate *A*. On the other hand, the key-compromise impersonation resilience property entails that any compromise of party *A*'s key should not enable an adversary to masquerade as other legitimate entities to *A*.
- ***Unknown Key-Share Resilience:*** A protocol achieves unknown key-share resilience if entity *A* is never coerced into sharing a key with some entity, *C*, when in fact *A* thinks he/she shares the key with party *B*.
- ***Key Control Resilience:*** This property entails that it should not be possible for any of the participants (or adversary) to compel the session key to a preselected value or predict the value of the session key.

This paper is structured as follows. Section 2 reviews some related work, and section 3 introduces some preliminaries outlining the mathematical background and complexity assumption definitions. Section 4 briefly reviews Cheng et al.'s protocols, and section 5 presents the results of cryptanalysis and improvement of Cheng et al.'s protocols. Section 6 reports the conclusions.

2. Related works

The first practical solution to the key agreement problem was provided by the Di³³æ-Hellman key agreement protocol [5]. The Diffie-Hellman key agreement protocol enables two entities to establish a session key that can be used to provide security or data integrity for later communications between the two entities. On the other hand, the basic Di³³æ-Hellman protocol does not authenticate the two communication entities. Therefore, it can su³³er from a "man-in-the-middle" attack. Di³³erent approaches have been developed to solve this security problem [6, 7]. Basically these protocols can be divided into two broad categories certification-based protocols, e.g. [8, 9], and identity-based protocols, e.g. [10].

The two-party key agreement protocols can be generalized to multi-party key agreement protocols, e.g. the three-party (tripartite) key agreement case. Joux [1] reported how to implement an elegant tripartite key agreement protocol using pairings: only one broadcast is needed for each entity. Joux's protocol has been applied in broadcast networks. On the other hand, as in the basic Di³³æ-Hellman protocol, Joux's protocol did not attempt to authenticate the three communicating entities, and it is also vulnerable to "man-in-the-middle" attacks.

In 2004, Cheng et al. proposed one round tripartite key agreement protocols [11]. Basically, Cheng et al.'s protocols are variants based on Shim's certificate-based and Zhang et al.'s identity-based protocols [12, 13]. Unfortunately, despite achieving many desirable security attributes, in Cheng et al.'s identity-based key agreement protocol, an adversary can successfully extract the long-term private keys of parties, which can allow an adversary to impersonate the respective parties [14]. The present paper also shows that Cheng et al.'s certification-based key agreement protocol is weak against key integrity attacks. This paper suggests solutions to fix these security flaws. Furthermore, a modified Cheng et al.'s identity-based one round tripartite protocol that is secure against private key exposure is presented.

3. Preliminaries

This section briefly introduces some relevant background knowledge.

3.1 Weil pairing

Let p be a prime such that $p = 12q - 1$ for some prime q and E be a super-singular elliptic curve defined by the Weierstrass equation $y^2 = x^3 + 1$ over F_p. The set of rational points $E(F_p) = \{(x, y) \in F_p : (x, y) \in E\}$ forms a cyclic group of order $p+1$. Furthermore, because $p+1 = 12q$ for some prime q, the set of points of order q in $E(F_p)$ form a cyclic subgroup, which is denoted as G_1. Let P be a generator of G_1 and G_2 be a subgroup of F_{p^2} containing all the points of order q. A modified Weil pairing is a bilinear map, $\hat{e} : G_1 \times G_1 \to G_2$ that satisfies the following properties [15]:

- *Bilinear*: for all $P, Q, R, S \in G_1$, $\hat{e}(P+Q, R+S) = \hat{e}(P, R)\hat{e}(P, S)\hat{e}(Q, R)\hat{e}(Q, S)$.
- *Non-degenerate:* For $Q \in G_1$, $\hat{e}(P, Q) \neq 1_{G_2}$.
- *Computable: For* $P, Q \in G_1$, there exists an efficient algorithm to compute $\hat{e}(P, Q) \in G_2$.

3.2 Bilinear Diffie-Hellman (BDH) assumption

For two cyclic groups G_1 and G_2 of prime order q, a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$ and a generator P of G_1, the BDH problem on $\{q, G_1, G_2, \hat{e}\}$ is as follows: compute $\hat{e}(P, P)^{abc} \in G_2$ with the known parameters (P, aP, bP, cP), where $a, b, c \in Z_q^*$ are random. Computing such a problem is assumed to be difficult.

3.3 Discreet logarithm (DL) problem

In a cyclic group G_1 with prime order q and a generator $P \in G_1$, the problem given P and $Y = yP$ for random $y \in Z_q^*$ is assumed difficult to calculate.

4. Review of Cheng et al.'s one-round tripartite key agreement protocols

This section reviews Cheng et al.'s identity-based and

certificate-based one-round tripartite key agreement protocols.

4.1 Identity-based one-round tripartite key agreement protocol

The Public Key Generating Center (PKGC) is responsible for setting up the system parameters and generating the parameters for parties. Let s and $P_{pub} = sP$ be private and public key of PKGC. For each registered party I with an identity ID_I, the PKGC calculates $Q_I = H_1(ID_I)$ and $S_I = sH_1(ID_I)$ as the public and private key pair for I. The following are the protocol steps:

$$A \to B, C: T_A^1 = aP_{pub}, T_A^2 = H(T_A^1)S_A \quad (1)$$
$$B \to A, C: T_B^1 = bP_{pub}, T_B^2 = H(T_B^1)S_B \quad (2)$$
$$C \to A, B: T_C^1 = cP_{pub}, T_C^2 = H(T_C^1)S_C \quad (3)$$

After exchanging the messages, party A carries out following verification computations: $\hat{e}(P_{pub}, H(T_B^1)Q_B) = (P, T_B^2)$ and $\hat{e}(P_{pub}, H(T_C^1)Q_C) = (P, T_C^2)$. Alternatively, A could also check; $(P, T_B^2 + T_C^2) = \hat{e}(P_{pub}, H(T_B^1)Q_B + H(T_C^1)Q_C)$. Party B and C performs similar verification operations. If the check step is passed, A, B and C respectively computes the session key as follows:

$$K_A = \hat{e}(T_B^1, T_C^1)^a = \hat{e}(P,P)^{abcs^2}$$
$$K_B = \hat{e}(T_A^1, T_C^1)^b = \hat{e}(P,P)^{bacs^2}$$
$$K_C = \hat{e}(T_A^1, T_B^1)^c = \hat{e}(P,P)^{cabs^2}$$

4.2 Certificate-based one-round tripartite key agreement protocol

This subsection presents Cheng et al.'s certification-based protocol. Let I_A be the identity of party A, x_A and $y_A = x_A P$ be his/her private and public key pair. Party A obtains a certification $Cert_A = (I_A \| y_A \| P \| S_{CA}(I_A \| y_A \| P))$ from a Certification Authority (CA). Here, S_{CA} is a signature of CA and P is the system parameter. The following shows the protocol steps for A, B and C, respectively

$$A \to B, C: T_A^1 = aP, T_A^2 = a(xP), Cert_A \quad (1)$$
$$B \to A, C: T_B^1 = bP, T_B^2 = b(yP), Cert_B \quad (2)$$
$$C \to A, B: T_C^1 = cP, T_C^2 = c(zP), Cert_C \quad (3)$$

After exchanging the messages, party A caries out verification process by calculate $\hat{e}(T_B^1, yP) = \hat{e}(T_B^2, P)$ and $\hat{e}(T_C^1, zP) = \hat{e}(T_C^2, P)$. Party B and C performs similar verification operations. After the check step, each party calculates the session key $K = \hat{e}(P, P)^{axbycz}$ using one of the following functions for A, B and C, respectively.

$$K_A = \hat{e}(T_B^2, T_C^2)^{ax} = \hat{e}(P,P)^{axybcz}$$
$$K_B = \hat{e}(T_A^2, T_C^2)^{bx} = \hat{e}(P,P)^{byaxcz}$$
$$K_C = \hat{e}(T_A^2, T_B^2)^{cx} = \hat{e}(P,P)^{czaxyb}$$

5. Cryptanalysis and improvement of Cheng et al.'s protocols

This section discusses security weaknesses of Cheng et al.'s protocols. In addition, possible solutions to fix the flaws are suggested. A modified identity-based protocol is also presented.

5.1 Private key exposure on Cheng et al.'s identity-based protocol

In Cheng et al.'s identity-based protocol, an adversary can successfully extract the long-term private keys of parties. The following shows how an adversary can succeed in this.

Suppose an adversary has succeeded in intercepting $(T_A^1, T_A^2), (T_B^1, T_B^2)$ and (T_C^1, T_C^2). This is quite feasible because A, B and C exchange $(T_A^1, T_A^2), (T_B^1, T_B^2)$ and (T_C^1, T_C^2), respectively, over an insecure channel. Using the intercepted message (T_A^1, T_A^2), the adversary can extract the long-term private key S_A for party A as follows:

- Computes $X = H(T_A^1)$.
- Using X and T_A^2, the adversary then computes:

$$\begin{aligned} X^{-1}T_A^2 &= X^{-1}H(T_A^1)S_A, \text{ since } T_A^2 = H(T_A^1)S_A \\ &= H(T_A^1)^{-1}H(T_A^1)S_A \\ &= S_A \end{aligned}$$

Therefore, an attacker can successfully extract A's long-term private key. Note that using the same reasoning and logic, an adversary can also extract long-term private keys for parties B and C by intercepted messages (T_B^1, T_B^2) and (T_C^1, T_C^2).

To fix this flaw, this paper suggest a modification to the structure of messages (T_A^1, T_A^2), (T_B^1, T_B^2) and (T_C^1, T_C^2). The following presents a modified Cheng et al.'s identity-based one-round tripartite key agreement protocol.

$$A \to B, C: T_A^1 = aP_{pub}, T_A^2 = a \cdot S_A \quad (1)$$
$$B \to A, C: T_B^1 = bP_{pub}, T_B^2 = b \cdot S_B \quad (2)$$
$$C \to A, B: T_C^1 = cP_{pub}, T_C^2 = c \cdot S_C \quad (3)$$

In this case, while verifying the messages, party A calculated $\hat{e}(T_B^1, Q_B) = (P, T_B^2)$ and $\hat{e}(T_C^1, Q_C) = (P, T_C^2)$ (see Table 1 below for correctness). Party B and C performs similar verification operations. If the check step is passed, A, B and C each calculates the session key as

Table 1. Correctness of verification.

$\hat{e}(T_B^1, Q_B) = (P, T_B^2)$	$\hat{e}(T_C^1, Q_C) = (P, T_C^2)$
$\hat{e}(T_B^1, Q_B) = \hat{e}(bP_{pub}, Q_B)$ $= \hat{e}(bsP, Q_B)$ $= \hat{e}(P, bsQ_B)$ $= \hat{e}(P, bS_B)$ $= \hat{e}(P, T_B^2)$	$\hat{e}(T_C^1, Q_C) = \hat{e}(cP_{pub}, Q_C)$ $= \hat{e}(csP, Q_C)$ $= \hat{e}(P, csQ_C)$ $= \hat{e}(P, cS_C)$ $= \hat{e}(P, T_C^2)$

follows:

$$K_A = \hat{e}(T_B^1, T_C^1)^a = \hat{e}(P,P)^{abcs^2}$$
$$K_B = \hat{e}(T_A^1, T_C^1)^b = \hat{e}(P,P)^{bacs^2}$$
$$K_C = \hat{e}(T_A^1, T_B^1)^c = \hat{e}(P,P)^{cabs^2}$$

Therefore, the three parties can successfully share the same session key securely.

5.2 Session key integrity attack on Cheng et al.'s certificate-based protocol

This subsection shows that Cheng et al.'s certification-based protocol is insecure against key integrity attacks and how the proposed solution fixed the problem. In key integrity attacks, the goal of an adversary is to cause honest parties to compute different session keys [16]. A key agreement protocol is said to be secure against key integrity attacks if no adversary (either insider or not) succeeds in causing honest parties to compute different session keys from other honest parties in a protocol session. The following shows how an adversary caries out a key integrity attack.

- First the adversary captures *B*'s broadcasted message $(T_B^1 = aP, T_B^2 = a(xP), Cert_B)$
- Computes two new messages:

$$T_B^{1^*} = P \text{ and } T_B^{2^*} = yP, Cert_B$$
$$T_B^{1^{**}} = b'P, T_B^{2^{**}} = b'(yP), Cert_B$$

- Forwards $(T_B^{1^*} = P, T_B^{2^*} = yP, Cert_B)$ to *A* and $(T_B^{1^{**}} = b'P, T_B^{2^{**}} = b'(yP), Cert_B)$ to *C*.

Upon receiving the message, party *A* verifies $\hat{e}(T_B^{1^*}, yP) = \hat{e}(T_B^{2^*}, P)$ and $\hat{e}(T_C^1, zP) = \hat{e}(T_C^2, P)$, While party *C* verifies $\hat{e}(T_B^{1^{**}}, yP) = \hat{e}(T_B^{2^{**}}, P)$ and $\hat{e}(T_A^1, xP) = \hat{e}(T_A^2, P)$.

The verification phase holds because the messages computed and forwarded by the adversary are of the same format as those from the legal parties.

After the protocol run, parties *A*, *B* and *C* will respectively calculate following session keys.

$$K_A = \hat{e}(T_B^{2^*}, T_C^2)^{ax} = \hat{e}(P,P)^{axycz}$$
$$K_B = \hat{e}(T_A^2, T_C^2)^{bx} = \hat{e}(P,P)^{byaxcz}$$
$$K_C = \hat{e}(T_A^2, T_B^{2^{**}})^{cx} = \hat{e}(P,P)^{czaxy\ b'}$$

Clearly, $K_A \neq K_B \neq K_C$. Therefore, an adversary succeeds in compromising the integrity of the session key.

To overcome this vulnerability, an additional protocol step, the key confirmation step, was introduced. The protocol requires that once each party computes the session key, he/she should compute a message authentication code (*MAC*) as follows.

$$A \rightarrow B, C: V_A = MAC_{K_A}(T_A^2, T_B^2, T_C^2)$$
$$B \rightarrow A, C: V_B = MAC_{K_B}(T_A^2, T_B^2, T_C^2)$$
$$C \rightarrow A, B: V_C = MAC_{K_C}(T_A^2, T_B^2, T_C^2)$$

Therefore, each party can confirm whether he/she shares the same key with counterparts by verifying two authentication codes.

6. Conclusion

This study showed that Cheng et al.'s identity-based and certificate-based protocols suffer from private key exposure attack and session key integrity attack, respectively. This paper proposed a solution to strengthen the security of the protocols. Furthermore, an improved Cheng et al.'s one-round tripartite key agreement identity-based protocol was presented, which showed enhanced security upon session key sharing.

References

[1] A. Joux, "A one-round protocol for tripartite Diffie-Hellman," Algorithm Number Theory Symposium-ANTS-IV, Lecture Notes in Computer Science 1838, *Springer-Verlag (2000)*, pp. 385–394. Article (CrossRef Link)

[2] A. Menezes, P. van Oorschot and S. Vanstone, "Handbook of applied cryptography," *CRC Press*, 1996. Article (CrossRef Link)

[3] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," *In Proceedings of the sixth IMA International Conference on Cryptography and Coding*, vol. 1355, pp. 30–45. *Springer-Verlag*, 1997. Article (CrossRef Link)

[4] L. Law, A. Menezes, M. Qu, J. Solinas, S. A. Vanstone, "An efficient protocol for authenticated key agreement," *Technical Report CORR 98-05, Department of C & O, University of Waterloo, 1998.* To appear in Designs, Codes and Cryptography. Article (CrossRef Link)

[5] W. Di³³æ, M. Hellman, "New directions in cryptography," *In IEEE Transactions on Information*

Theory, 1976, No.22, pp.644-654, 1976. Article (CrossRef Link)

[6] S. Blake-Wilson, A. Menezes, "Authenticated Di³³æ-Hellman key agreement protocols," *In Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), Springer-Verlag*, vol. 1556, pp.339-361, 1999. Article (CrossRef Link)

[7] B. Song, K. Kim, "Two-pass authenticated key agreement protocol with key confirmation," *In Proceedings of Indocrypt 2000, Springer-Verlag*, vol. 1977, pp.237-249, 2000. Article (CrossRef Link)

[8] S. S. Al-Riyami, K. G. Paterson, "Tripartite authenticated key agreement protocols from pairings," *IMA Conference on Cryptography and Coding, Lecture Notes in Computer Science 2898, Springer-Verlag (2003)*, pp. 332–359. See also Cryptology ePrint Archive, Report 2002/035. Article (CrossRef Link)

[9] K. Shim, K, "Efficient one-round tripartite authenticated key agreement protocol from the Weil pairing," *Electronics Letters*, vol. 39, pp. 208–209, 2003. Article (CrossRef Link)

[10] D. Nalla, K. C. Reddy, "ID-based tripartite authenticated key agreement protocols from pairings," *Cryptology ePrint Archive, Report 2003/004.* Article (CrossRef Link)

[11] Z. Cheng, L. Vasiu, R. Comley, "Pairing-based one-round tripartite key agreement protocols," *Cryptology ePrint Archive, Report 2004/079.* Article (CrossRef Link)

[12] K. Shim, "A man-in-the-middle attack on Nalla-Reddy's ID-based tripartite authenticated key agreement protocol," *Cryptology ePrint Archive, Report 2003/115.* Article (CrossRef Link)

[13] F. Zhang, S. Liu, K. Kim, "ID-based one round authenticated tripartite key agreement protocol with pairings," *Cryptology ePrint Archive, Report 2002/122.* Article (CrossRef Link)

[14] K. Mtonga, H. Kim, E. Yoon, "Advanced pairing-based one-round tripartite key agreement protocol," KICS 2013 winter conference, pp.110-111, 2013. Article (CrossRef Link)

[15] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," *In Advances in Cryptology—CRYPTO 2001, Springer Berlin Heidelberg*, pp. 213-229, 2001. Article (CrossRef Link)

[16] S. Arita, A. Sakuma, "A group-key agreement protocol secure against the malleability attacks," *Information Theory and Its Applications*, 2008. ISITA 2008. International Symposium on, vol., no., pp.1-6, 7-10, 2008. Article (CrossRef Link)

Kambombo Mtoga received his B.S. degree in Mathematics Education from University of Malawi-Chancellor College, Malawi, in 2010. Currently, he is a graduate student of Department of IT Convergence at the Kyungil University, Republic of Korea. His current research interests are cryptography, authentication technologies, and RFID security.

Eun-Jun Yoon received his MSc degree in computer engineering from Kyungil University in 2002 and the PhD degree in computer science from Kyungpook National University in 2006, Republic of Korea. From 2007 to 2008, he was a full-time lecturer at Faculty of Computer Information, Daegu Polytechnic College, Republic of Korea. From 2009 to 2011, he was a 2nd BK21 contract professor at the School of Electrical Engineering and Computer Science, Kyungpook National University, Republic of Korea. Currently, he is a professor at the Department of Cyber Security, Kyungil University, Republic of Korea. His current research interests are cryptography, authentication, smart card security, network security, mobile security, and steganography.