International Journal of Applied Science and Engineering
2005. 3, 1: 13-18

An Improved Tripartite Authenticated Key Agreement Protocol Based on Weil Pairing

Hung-Yu Chien* and Ru-Yu Lin

Department of Information Management,
Chaoyang University of Technology,
Wufong, Taichung County 41349, Taiwan, R.O.C.

Abstract: This paper examines the security weaknesses of Lin, Huang and Lin's tripartite authenticated key agreement scheme, and proposes our new scheme. The proposed scheme resists all the security threats and further provides key confirmation. The scheme is very simple and efficient.

Keywords: Weil pairing; elliptic curve; authenticated key agreement.

1. Introduction

Recently, several new cryptosystems based on bilinear pairings have been proposed. One of the important pairing-based cryptographic schemes is the tripartite key agreement protocol, which allows three entities establish session keys [1, 2, 4, 5]. 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 auditing, or data recovery purposes. It can also facilitate the job of group communication. Joux's pairing-based tripartite key agreement protocol [6] is efficient. However, the protocol does not authenticate the messages, and, therefore, Shim [4] proposed an authenticated version to the basic man-in-the-middle attack. However, Shim's scheme requires two operations to compute the session key. Recently, Lin, Huang, and Lin proposed an efficient tripartite authenticated key agreement protocol that requires only one pairing operation [1]. The schemes are named as the LHL schemes for short in this article.

However, the LHL tripartite key scheme is vulnerable to insider attack, outsider impersonation attack, and key-compromise impersonation attack. This paper examines the security weakness of the scheme and proposes our new scheme. The rest of the paper is organized as follows. Section 2 gives some preliminaries of bilinear pairing, and discusses the security properties of a secure tripartite key agreement scheme. Section 3 reviews the LHL tripartite key agreement protocol. Section 4 shows the security weaknesses of the LHL tripartite key agreement schemes and proposes our improved scheme, which is followed by the conclusions in Section 5.

* Corresponding author: e-mail: Hychien@mail.cyut.edu.tw *Accepted for Publication: March 08, 2005*

2. Preliminaries

2.1. Modified Weil pairing

Let p be a prime such that $p=2 \pmod 3$ and $p=6q-1$ for some prime $q>3$. Let E be the elliptic curve defined by $y^2 = x^3 + 1$ over F_p. Let P be a generator of the group of points with order $q = (p+1)/6$ from the elliptic curve E. Let G be the subgroup of $F_{p^2}^*$ of order q. The Weil pairing on the two elliptic curve discrete logarithm problem, is a mapping $e: G_q \times G_q \to G$. The point G_q is the group of points with order q. The modified Weil pairing is defined as [2][3].

$\hat{e}: G_q \times G_q \to G$, $\hat{e}(P,Q) = e(P, \phi(Q))$

Then $\phi(Q) \in E / F_{P^2}$, and $Q \in E / F_P$ .The map $\phi(x, y) = (\zeta x, y)$ of points on the curve E, $1 \neq \zeta \in F_{P^2}^*$ is a solution of $x^3 - 1 = 0$ in F_{P^2}. The modified Weil pairing then satisfies the following properties:

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

2.2. Security properties of key agreement scheme

A secure tripartite authenticated key agreement protocol should have the following properties:

1. Resistance to known-key security ： Knowledge of past session keys does not allow the attacker to deduce the session keys afterward.
2. Forward secrecy：Even though an adversary has compromised the long-term secret keys of one or more entities, the secrecy of previously established session keys should not be affected.
3. Resistance to key-compromise impersonation attack：The compromise of an entity *A*'s long-term private key will allow an adversary to impersonate *A*, but it should not enable the adversary to impersonate other entities to *A*.
4. No insider attack：The insider attack in a tripartite key agreement protocol means that one of the three entities try to impersonate another one of the three entities. For example, *B* might try to fool *A* that they and *C* have participated in a protocol run, while in fact *C* does not. This attack could have serious consequences: for example, if *C* acts as an on-line escrow agent or a referee. A secure tripartite key agreement protocol should resist this attack.
5. No man-in-the-middle attack：A secure key agreement protocol should resist this attack.

3. The LHL tripartite key agreement scheme

Setup：The system set up the following parameters for the users. The public domain parameters $(p, q, E, P, \hat{e})$, where E is a elliptic curve defined over F_p, P be a generator of the group of points with order q from the elliptic curve E, and $\hat{e}(,)$ is the modified Weil pairing that satisfies the bilinear properties. Each entities static public keys are exchanged via certificates. $Cert_A$ denotes *A*'s public-key certificate, which includes her static public key $Y_A = a \cdot P$, an unique identifier string of *A* such as *A*'s name and address, and a certification authority *CA*'s signature over this information. Similarly, $Cert_B$ and $Cert_C$ are the certificates for *B* and *C*, with $Y_B = b \cdot P$ and $Y_C = c \cdot P$ as their static pub-

lic keys, respectively, where *a*, *b* and *c*, are random numbers selected by *A*, *B* and *C*, respectively. *a*, *b* and *c*, are used as the long-term private keys.

The LHL protocol： Let *A*, *B* and *C* are the three entities running the protocol. In each communication, *A* (*B* and *C*) chooses a random number *x* (*y* and *z*), computes $T_A = x \cdot Y_A$ ($T_B = y \cdot Y_B$ and $T_C = z \cdot Y_C$), and broadcasts the value with his certificates, respectively. Note that *x*, *y* and *z* are used as the ephemeral private keys ($x, y, z \in Z$).
Protocol messages:

$$A: T_A = x \cdot (aP), \ Cert_A$$
$$B: T_B = y \cdot (bP), \ Cert_B$$
$$C: T_C = z \cdot (cP), \ Cert_C$$

Than *A*, *B* and *C* compute the session keys K_A, K_B, and K_C as follows:

$$K_A = \hat{e}(Y_B + T_B, Y_C + T_C)^{a+ax} = \hat{e}(P,P)^{(a+ax)(b+by)(c+cz)} \quad (1)$$

$$K_B = \hat{e}(Y_A + T_A, Y_C + T_C)^{b+by} = \hat{e}(P,P)^{(a+ax)(b+by)(c+cz)} \quad (2)$$

$$K_C = \hat{e}(Y_A + T_A, Y_B + T_B)^{c+cz} = \hat{e}(P,P)^{(a+ax)(b+by)(c+cz)} \quad (3)$$

$$\begin{aligned} K &= kdf(K_A \| A \| B \| C) \\ &= kdf(K_B \| A \| B \| C) \\ &= kdf(K_C \| A \| B \| C) \end{aligned} \quad (4)$$

The shared secret key is the K where *kdf()* is a key derivation function.

4. Weaknesses of the LHL scheme and our improvements

4.1. Impersonation attack on the LHL scheme

Insider attack: The insider attack in a tripartite key agreement protocol means that one of the three entities try to impersonate another one of the three entities. Assume *B* is the inside attacker, tries to fool *A* that *C* participates in the protocol run, *B* randomly chooses number z' and computes $T_C' = -Y_C + z' \cdot P$, and sends T_C' to *A*. *A* will accept the messages and will compute the session key. The keys computed by the entities are as follows:

$$K_A' = \hat{e}(Y_B + T_B, Y_C + T_C')^{a+ax} = \hat{e}(P,P)^{(a+ax)(b+by)z'}$$
$$K_B' = \hat{e}(Y_A + T_A, Y_C + T_C')^{b+by} = \hat{e}(P,P)^{(a+ax)(b+by)z'}$$
$$K_C' = \hat{e}(Y_A + T_A, Y_B + T_B)^{z'} = \hat{e}(P,P)^{(a+ax)(b+by)z'}$$

According to the above discussion, the new session key is the
$K_{ABC} = K_A' = K_B' = K_C' = \hat{e}(P,P)^{(a+ax)(b+by)z'}$, that is, *B* fools *A* that they and *C* have participated in a protocol run, while in fact *C* does not. And, *B* has shared a session key with *A*.

Outsider impersonation attack：We now demonstrate how an adversary (*E*) who is an outsider can easily impersonate as *B* to cheat both *A* and *C* as follows. *E* masquerades as *B*, and sends ($T_B = -Y_B + y'P, Cert_B$) to *A* and *C*, the number y' is random select by *E*. After receiving all the data, *A* will compute $K_A = \hat{e}(Y_B + T_B, Y_C + T_C)^{a+ax} = \hat{e}(P,P)^{(a+ax)y'(c+cz)}$, and *C* will compute
$K_C = \hat{e}(Y_A + T_A, Y_B + T_B)^{c+cz} = \hat{e}(P,P)^{(a+ax)y'(c+cz)}$
So, the adversary can derive the same session by computing $K_{B(E)} = \hat{e}(Y_A + T_A, Y_C + T_C)^{y'}$ $= \hat{e}(P,P)^{(a+ax)y'(c+cz)}$. That is, the adversary has impersonated *B* successfully. It is also easy that an adversary can masquerades as *B* and *C* simultaneously by extending above attack.

Key-compromise impersonation attack： It is easy to check that the LHL also cannot resist key-compromise impersonation attack, since an outsider who does not know any private keys of the communicating parties can impersonate these parties.

4.2. The improved scheme

This section proposes our improved scheme to conquer all the weaknesses, and it further provides key confirmation. Our protocol op-

erates in broadcast mode. The proposed scheme has the same first round and session key confirmation (Equations (1) to (4)) as the LHL scheme. But, it requires a second round to confirm the derived key. In the second round, each entity is required to sign on the hashing values on the derived session key, using Boneh, Lynn, and Shacham's short signature scheme [7].

The proposed scheme is described as follows. Define one cryptographic hash function $H_1(,):\{0,1\}^* \to G_q$, where G_q is a subgroup of an elliptic curve.

Protocol run 1:

$$A: T_A = x \cdot (aP), \ Cert_A$$
$$B: T_B = y \cdot (bP), \ Cert_B$$
$$C: T_C = z \cdot (cP), \ Cert_C$$

Protocol run 2:

Than, *A*, *B* and *C* compute the signatures S_A, S_B and S_C as follows：

A： $S_A = aH_1(ID_A, K)$, where ID_A is *A*'s identity, K is the session key as in Equation (4), and a is *A*'s long-term private key.

B： $S_B = bH_1(ID_B, K)$

C： $S_C = cH_1(ID_C, K)$

After receiving the confirmation data from other entities, each entity will verify the signature data to conform the shared key. For example, *A* will perform the following checking, suing Equations 5 and 6. If the verification succeeds, then *A* accepts the session key. Likewise, *B* and *C* will perform similar verifications.

$$\hat{e}(S_B, P) = \hat{e}(H_1(ID_B, K), Y_B) \quad (5)$$

$$\hat{e}(S_C, P) = \hat{e}(H_1(ID_C, K), Y_C) \quad (6)$$

We show the correctness of Equation (5) as follows:

$\hat{e}(S_B, P) = \hat{e}(bH_1(ID_B, K), P) = \hat{e}(H_1(ID_B, K), Y_B)$.

Please note that the proposed scheme can conquer all the weaknesses of the LHL scheme and further provides key confirmation, with two additional pairing operations.

4.3. Security analysis

Known-key security: Each run of the protocol computes a different session key which depends on the ephemeral private keys (*x*, *y* and *z*). Thus, knowledge of past session keys does not allow the attacker to deduce the session keys afterward.

Full forward security: Suppose the long-term private keys of all the entities are compromised. Also allows an adversary to obtain session keys previously established between participator. But he/she cannot compute the previously established session key. In this case, if an adversary has learned that all entities long-term private keys, say *a*, *b* and *c*, at some point in the future, the adversary is not able to compute the previously established session key $K_A = \hat{e}(Y_B + T_B, Y_C + T_C)^{a+ax}$ without ephemeral private key *x*. Similarly, K_B and K_C cannot be computed without y and z, respectively.

Key-compromise impersonation resistance: Key-compromise impersonation means that compromise of an entity's (say *A*) long-term private key *a* will allow an adversary *E* to masquerade as *C* (or *B*) to *A*. In our protocol, even though an adversary who has compromised *A*'s private key could forge the message in the first run and compute the same session key with *A*, he cannot forge the signature on behalf of *C* (or *B*) to *A*. This key confirmation requirement makes our protocol resistant to key compromise impersonation attack.

Insider impersonation resistance: Even though an insider attacker could compute the session key, he could not forge the signature on behalf of other entities. Without knowing their private key, this key confirmation mes-

sage (signature on hashed key) makes the protocol immune to insider attack.

Authenticated key confirmation: After computing the session keys, each entity use their long-term private key to sign on the hashing values and broadcast their signature. Every receiver can validate these signatures, using Equations (5) and (6). Therefore, authenticating key confirmation is achieved.

4.4. Efficiency analysis

This section analyzes the efficiency of our improved scheme and its counterparts. According to the Table 1, both our improved protocol and TAKC use signatures. To make a comparison, we assume the two schemes use the same signature and verification scheme. Both our protocol and TAKC protocol achieve key confirmation, using three pairing operation (one pairing is used to compute secret key, and the others are used to verify the signature). Our protocol needs three Weil pairing, two elliptic curve scalar multiplication and two additions, to combine the long-term public key with the ephemeral public key. Additionally, it requires one modular exponentiation, one signature operation and two signature verifications. Compared with TAKC, our protocol is more computationally efficient. Furthermore, our scheme and TAKC protocol request two rounds of broadcast to satisfy key confirmation.

Table 1. Summaries of certificate-based tripartite key agreement protocols

	SHIM [6]	LHL [2]	TAKC [8]	Ours
Key confirmation	N	N	Y	Y
Outsider impersonation	N	Y	N	N
Inside impersonation	Y	Y	N	N
Key compromise impersonation	Y	Y	N	N
Forward secrecy	Y	Y	Y	Y
Number of round	1	1	2	2
Computation of one party	$2E_{pair}$ $1E_{scalar}$ $2F_{\exp}$	$1E_{pair}$ $1E_{scalar}$ $2E_{add}$ $1F_{\exp}$	$1E_{pair}$ $3E_{scalar}$ $2E_{add}$ $1F_{\exp} + 3Enc$ $*1Sig + 2Ver$ $(2E_{pair} + 1E_{scalar})$	$1E_{pair}$ $1E_{scalar}$ $2E_{add}$ $1F_{\exp}$ $*1Sig + 2Ver$ $(2E_{pair} + 1E_{scalar})$

Notation: E_{pairr} denotes one pairing operation on the elliptic curves (E); E_{scalar} denotes one scalar multiplication on the elliptic curves (E); E_{add} denotes one elliptic curve point addition; $F_{\exp}$ denotes one exponentiation on G; Enc denotes one symmetric encryption; *Sig* denotes one signature operation; *Ver* denotes one signature verification.

* The values in the brackets count the number of operations required, assuming we use the signature and verification operations described in this paper.

5. Conclusions

This paper has showed the security weaknesses of the Lin-Huang-Lin tripartite authenticated key agreement protocol, and has proposed an improved scheme that resists all the security threats and provides key confirmation. Also note that the proposed scheme requires only three pairing operations. It is secure and efficient.

References

[1] Lin, C. H., K. J. Huang, and S. S. Lin. 2004. Improving Shim's tripartite authenticated key agreement protocol based on Weil pairing. *ISC*: 250-255.

[2] Boneh, D. and M. Franklin. 2001. Identity-based encryption from the Weil Pairing. *Advances in Cryptology Crypto'01*, Santa Barbara, CA, USA, August: 213-229.

[3] Silverman, J. H. 1986. *"The arithmetic of elliptic curves"*. GTM 106, Springer-Verlag.

[4] Shim, K. 2003. Efficient one-round tripartite authenticated key agreement protocol from Weil pairing. *Electronics Letters*, 39, 2: 208-209.

[5] Chen, L. and C. Kudla. 2002. *"Identity Based Authenticated Key Agreement Protocols from Pairings"*. IACR, Report.

[6] Joux, A. 2000. A one round protocol for tripartite Diffie-Hellman. *ANTS IV*, LNCS1838, Spring-Verlag: 385-394.

[7] Boneh, D. B. Lynn, and H. Shacham. 2001. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology-Asiacrypt 2001*, LNCS 2248, Springer-Verlag: 514-532.

[8] Al-Riyami, S. S. and K. G. Paterson. 2002. *"Tripartite authenticate key agreement protocols from Paiiring"*. Cryptology ePrint Archive, Report.