An Improved One-Round ID-Based Tripartite Authenticated Key Agreement Protocol

Meng-Hui Lim[1], Sanggon Lee[2]

[1] Department of Ubiquitous IT, Graduate school of Design & IT,
Dongseo University, Busan 617-716, Korea
meng17121983@yahoo.com
[2] Department of Information & Communication,
Dongseo University, Busan 617-716, Korea
nok60@gdsu.dongseo.ac.kr

Abstract. A tripartite authenticated key agreement protocol is generally designed to accommodate the need of three specific entities in communicating over an open network with a shared secret key, which is used to preserve confidentiality and data integrity. Since Joux initiates the development of tripartite key agreement protocol, many prominent tripartite schemes have been proposed subsequently. In 2005, Tso et al. have proposed an ID-based non-interactive tripartite key agreement scheme with k-resilience. Based on this scheme, they have further proposed another one-round tripartite application scheme. Although they claimed that both schemes are efficient and secure, we discover that both schemes are in fact breakable. In this paper, we impose several impersonation attacks on Tso et al.s schemes in order to highlight their flaws. Subsequently, we propose an enhanced scheme which will not only conquer their defects, but also preserve the desired security attributes of a key agreement protocol.

1 Introduction

A *key agreement protocol* is the mechanism in which a shared secret key is derived by two or more protocol entities as a function of information contributed by each of these parties such that no single entity can predetermine the resulting value. Usually, this session key is established over a public network controlled by the adversaries and it would vary with every execution round (session) of the protocol. This secret key can subsequently be used to create a confidential communication channel among the entities.

The situation where three or more parties share a key is often called *conference keying*. The tripartite case is of the 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 communicating parties. For example, a third party can be added to chair, or referee a conversation for ad hoc auditing, or data recovery purposes. Besides, it can also facilitate the job of group communication.

Wilson and Menezes [17, 18] have defined a number of desirable security attributes which can be used to analyze a tripartite key agreement protocol. These security attributes are described as follows:

Known session key security. A protocol is considered to be *known session key secure* if it remains achieving its goal in the face of an adversary who has learned some previous session keys.

(Perfect) forward secrecy. A protocol enjoys *forward secrecy* if the secrecy of the previous session keys is not affected when the long term private keys of one or more entities are compromised. *Perfect forward secrecy* refers to the scenario when the long term private keys of all the participating entities are compromised.

Key-Compromise Impersonation Resilience. Suppose that A's long term private key has been disclosed. Obviously an adversary who knows this value can now impersonate A since it is precisely the value which identifies A. We say that a protocol is *key-compromise impersonation resilient* if this loss will not enable an adversary to masquerade as other legitimate entities to A as well or obtain other entities secret key.

Unknown Key-Share Resilience. In an unknown key-share attack, an adversary convinces a group of entities that they share a key with the adversary whereas in fact, the key is shared between the group and another party. This situation can be exploited in a number of ways by the adversary when the key is subsequently used to provide encryption of integrity.

Key Control Resilience. It should not be possible for any of the participants (or an adversary) to compel the session key to a preselected value or predict the value of the session key.

Over the years, numerous tripartite key agreement protocols have been proposed. However, most of them have been proven to be insecure [1, 2, 6, 8–10, 12, 13]. In 2000, Joux [6] had proposed the first one-round pairing-based tripartite Diffie-Hellman key agreement protocol. However, Shim [13] had pointed out that Joux's protocol does not authenticate the communicating entities and therefore, it is susceptible to the man-in-the-middle attack. To overcome this, Shim had proposed an improved scheme which employs the public key infrastructure to overcome the security flaw in Joux's protocol and she claimed that the improved protocol is able to withstand the man-in-the-middle attack. However, Shim's attempt has also turned out to be insecure eventually [2, 8, 14]. In 2005, Tso et al. [15] have proposed an *ID-based non-interactive key agreement scheme* (IDNIKS) with k-resilience for three parties. They have claimed that their protocol is the first secure non-interactive tripartite protocol which provides ID-based authenticity with no employment of hash functions. Based on this scheme, they have further proposed a tripartite application scheme which requires only one round of message transmission. Although they claimed that both schemes are efficient and secure, we discover that both schemes are in fact susceptible to various impersonation attacks.

Hence, in this paper, we highlight the weaknesses of Tso et al.'s tripartite IDNIKS and their application scheme. In order to conquer these defects, we propose our enhanced scheme based on their application scheme, and subsequently carry out a thorough security analysis to ensure that our enhanced scheme has satisfied all the required security attributes of a desired key agreement protocol. The structure of this paper is organized as follows. In Section 2, we illustrate some basic properties of bilinear pairings and several Diffie-Hellman assumptions. In Section 3, we review Tso et al's tripartite IDNIKS and their subsequent application scheme. In Section 4, we present our impersonation attacks on both schemes and then in Section 5, we propose our enhanced scheme as well as the associated security proofs. Lastly, we conclude this paper in Section 6.

2 Preliminaries

Let $\mathbf{G}_1$ be an additive group of a large prime order, q and $\mathbf{G}_2$ be a multiplicative group of the same order, q. Let $P, Q \in \mathbf{G}_1$ and $\hat{e} : \mathbf{G}_1 \times \mathbf{G}_1 \longrightarrow \mathbf{G}_2$ be a bilinear pairing with the following properties:

- **Bilinearity**: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(abP, Q)$ for any $a, b \in Z_q^*$.
- **Non-degeneracy**: $\hat{e}(P, Q) \neq 1$.
- **Computability**: There exists an efficient algorithm to compute $\hat{e}(P, Q)$.

A bilinear map which satisfies all three properties above is considered as *admissible bilinear*. It is noted that the Weil and Tate pairings associated with the supersingular elliptic curves or abelian varieties, can be modified to create such bilinear maps. Now, we describe some cryptographic problems:

Bilinear Diffie-Hellman Problem (BDHP). Let $\mathbf{G}_1$, $\mathbf{G}_2$, P and $\hat{e}$ be as above with the order q being prime. Given (P, aP, bP, cP) with $a, b, c \in Z_q^*$, compute $\hat{e}(P, P)^{abc} \in \mathbf{G}_2$.

Discrete Logarithm Problem (DLP). Given two groups of elements P and Q, such that $Q = nP$. Find the integer n whenever such an integer exists.

Throughout this paper, we assume that BDHP and DLP are hard such that there is no polynomial time algorithm to solve BDHP and DLP with non-negligible probability.

3 Review of Tso et al.'s Schemes

3.1 *k*-Resilient Tripartite IDNIKS

System Setting:
As described in Sect. 2, assume that $\mathbf{G}_1$ is an additive group and $\mathbf{G}_2$ is a multiplicative group, both with prime order q. Let P be a generator of $\mathbf{G}_1$, $\hat{e} : \mathbf{G}_1 \times \mathbf{G}_1 \longrightarrow \mathbf{G}_2$ be a bilinear pairing and $k \ll q$ be the resilience parameter. These settings are assumed to be generated by the key generation center (KGC).

Key Generation:
KGC picks $k+1$ random numbers $d_0, d_1, \cdots, d_k \in Z_q^*$, and generates a polynomial $f(x)$ of degree k, where

$$f(x) = d_0 + d_1 x + \cdots + d_k x^k \in Z_q[x]. \tag{1}$$

KGC then computes

$$V_0 = d_0 P, V_1 = d_1 P, \cdots, V_k = d_k P. \tag{2}$$

The system public parameters published by KGC are $\{P, V_0, \cdots, V_k\}$ and the KGC's private keys are $\{d_0, d_1, \cdots, d_k\}$. In addition, KGC computes

$$s_i = f(ID_i) = d_0 + d_1 ID_i + \cdots + d_k (ID_i)^k \text{mod } q. \tag{3}$$

for the entity i with identity $ID_i \in Z_q^*$ and sends s_i to i through a private secure channel. For an IDNIKS which involves three protocol entities A, B, and C, the corresponding public / private key pairs are computed as follows:

A: Public key: ID_A, Private key: $s_A = f(ID_A)$
B: Public key: ID_B, Private key: $s_B = f(ID_B)$
C: Public key: ID_C, Private key: $s_C = f(ID_C)$

Key Agreement:
In this non-interactive key establishment scheme, each A, B and C uses the system's public information, peer's public key as well as his own secret key to derive the shared secret with the other protocol entities.

$$\Omega_A = \sum_{i=0}^{k} (ID_A)^i V_i = s_A P. \tag{4}$$

$$\Omega_B = \sum_{i=0}^{k} (ID_B)^i V_i = s_B P. \tag{5}$$

$$\Omega_C = \sum_{i=0}^{k} (ID_C)^i V_i = s_C P. \tag{6}$$

A computes Eqs. (5) and (6), and the tripartite key

$$K_A = \hat{e}(\Omega_B, \Omega_C)^{s_A}. \tag{7}$$

B computes Eqs. (4) and (6), and the tripartite key

$$K_B = \hat{e}(\Omega_A, \Omega_C)^{s_B}. \tag{8}$$

C computes Eqs. (4) and (5), and the tripartite key

$$K_C = \hat{e}(\Omega_A, \Omega_B)^{s_C}. \tag{9}$$

Consistency:

$$\begin{aligned} K_A &= \hat{e}(\Omega_B, \Omega_C)^{s_A} \\ &= \hat{e}(\sum_{i=0}^{k}(ID_B)^i V_i, \sum_{i=0}^{k}(ID_C)^i V_i)^{s_A} \\ &= \hat{e}(s_B P, s_C P)^{s_A} \\ &= \hat{e}(P, P)^{s_A s_B s_C} \\ &= K_B = K_C \end{aligned} \tag{10}$$

3.2 One-round IDNIKS-based Application

Tso et al.'s application scheme has the same system setting and key generation as the previous scheme.

Key Agreement:
A chooses a random number $r_A \in Z_q^*$ and computes

$$X_A = r_A P, \tag{11}$$

B chooses a random number $r_B \in Z_q^*$ and computes

$$X_B = r_B P, \tag{12}$$

C chooses a random number $r_C \in Z_q^*$ and computes

$$X_C = r_C P. \tag{13}$$

Assume that $Sig_i(\cdot)$ denotes the signature of an entity i. Then, over a public channel,

$$A \rightarrow B, C : X_A, Sig_A(X_A). \tag{14}$$

$$B \rightarrow A, C : X_B, Sig_B(X_B). \tag{15}$$

$$C \rightarrow A, B : X_C, Sig_C(X_C). \tag{16}$$

From Eqs. (5), (6), (12) and (13), A computes the tripartite key

$$K_A = \hat{e}(\Omega_B + X_B, \Omega_C + X_C)^{s_A + r_A}. \tag{17}$$

From Eqs. (4), (6), (11) and (13), B computes the tripartite key

$$K_B = \hat{e}(\Omega_A + X_A, \Omega_C + X_C)^{s_B + r_B}. \tag{18}$$

From Eqs. (4), (5), (11) and (12), C computes the tripartite key

$$K_C = \hat{e}(\Omega_A + X_A, \Omega_B + X_B)^{s_C + r_C}. \tag{19}$$

Consistency:

$$\begin{aligned} K_A &= \hat{e}(\Omega_B + X_B, \Omega_C + X_C)^{s_A + r_A} \\ &= \hat{e}(\sum_{i=0}^{k}(ID_B)^i V_i + X_B, \sum_{i=0}^{k}(ID_C)^i V_i + X_C)^{s_A + r_A} \\ &= \hat{e}(s_B P + r_B P, s_C P + r_C P)^{s_A + r_A} \\ &= \hat{e}(P, P)^{(s_A + r_A)(s_B + r_B)(s_C + r_C)} \\ &= K_B = K_C \end{aligned} \tag{20}$$

4 Our Attacks

4.1 Impersonation Attacks on k-Resilient Tripartite IDNIKS

Key-Compromise Impersonation Attack:

The Key-Compromise Impersonation (KCI) attack is deemed successful only if the adversary manages to masquerade as another protocol principal to communicate with the victim after the victim's private key has been compromised. Suppose that an adversary, E_A has the knowledge of A's private key s_A and he intends to launch the KCI attack against A by pretending B in a communication run. E_A then initiates a communication session with A and C. By computing Eqs. (5) and (6), E_A is then able to compute the tripartite key K_B by using Eq. (7). Similarly after compromising a legitimate entity's private key, the adversary can simply impersonate anyone from the other $(k-1)$ legitimate entities to communicate with the victim, with the aim to capture valuable information (e.g. credit card number) about him.

In this key agreement protocol, each of the protocol entities merely employs his static private key and the other entities' public keys to derive a shared secret. Since this protocol is non-interactive, no ephemeral keys are involved in computing the tripartite key. Hence, it seems difficult for IDNIKS to resist the KCI attack.

Insider Impersonation Attack:

In a two-party's authentication protocol, the adversary who impersonates the communicating parties would probably be an *outsider*. However, in the k-party's case where $k \geq 3$, the adversary who impersonates the communicating parties might be a legal entity of the communicating group, known as an *insider* and this kind of impersonation attack is the *insider impersonation attack* [3]. The consequence of this attack would be disastrous if the impersonated party is a referee or an auditor.

In this tripartite IDNIKS, a malicious insider can easily impersonate any legitimate entity during a protocol run. For instance, suppose that B is the insider impersonation attacker who wishes to fool A by masquerading as C in a communication run. B initiates IDNIKS with A while at the same time, B also plays another role as B_C (B masquerading as C). By computing Eqs. (4) and (6), B can then calculate the tripartite key K_B and K_C by using Eq. (8). Since IDNIKS is non-interactive and no ephemeral values are employed, A can never find out that C is in fact absent in that communication run.

Generally, the insider impersonation attack can be launched against any legal entity in this protocol as the malicious insider can impersonate anyone from the other $(k-2)$ entities at the snerally, the insider impersonation attack can be launched against any legal entity in this protocol as the malicious insider can impersonate anyone from the other $(k-2)$ entities at the same time. Hence, we argue that key agreement protocol for three or more parties' should not be designed to be non-interactive as it would be vulnerable to the insider impersonation attack under any circumstances.

4.2 Impersonation Attacks on One-round IDNIKS-based Application

Insider Impersonation Attack:

In the tripartite application scheme, Tso et al. have emphasized that each protocol participant P_i must append a signature to the random parameter X_{P_i} in order to avoid the insider impersonation attack. However, we discover that their application scheme is still insecure since a malicious insider can easily replay any message together with the signature obtained from the previous session to launch the insider impersonation attack. For example, suppose that a malicious legal entity, B has obtained X_A as shown in Eq. (11) in a previous session involving A, B and C. B is now able to victimize D by replaying X_A in another communication session involving B_A (B impersonating A), B and D. The insider impersonation attack can be carried out as follows:

$$\Omega_D = \sum_{i=0}^{k} (ID_D)^i V_i \tag{21}$$

$$B_A \rightarrow B, D : X_A, Sig_A(X_A), \text{where } X_A = r_A P, \tag{22}$$

$$B \rightarrow B_A, D : X'_B, Sig_B(X'_B), \text{where } X'_B = r'_B P, \tag{23}$$

$$D \rightarrow B_A, B : X'_D, Sig_D(X'_D), \text{where } X'_D = r'_D P. \tag{24}$$

From Eqs. (4), (21), (22) and (24), B and B_A computes the tripartite key

$$\begin{aligned} K_A = K_B &= \hat{e}(\Omega_A + X_A, \Omega_D + X'_D)^{s_B + r'_B} \\ &= \hat{e}(P,P)^{(s_A + r_A)(s_B + r'_B)(s_D + r'_D)}. \end{aligned} \tag{25}$$

From Eqs. (4), (5), (22) and (23), D computes the tripartite key

$$\begin{aligned} K_D &= \hat{e}(\Omega_A + X_A, \Omega_B + X'_B)^{s_D + r'_D} \\ &= \hat{e}(P,P)^{(s_A + r_A)(s_B + r'_B)(s_D + r'_D)}. \end{aligned} \tag{26}$$

Outsider Impersonation Attack:

A secure protocol should not allow an outsider attacker to impersonate any protocol entity P_i in establishing a session with the other legal entities without knowing P_i's secret key s_{P_i} even if the other secret information (such as signature) has been exposed. Assume that A's signature has been compromised by some means. An outsider adversary, E_A is then able to impersonate A and carry out his attack as follows:
E_A initiates a protocol run with B and C, and selects a random number $m \in Z_q^*$.
Message Broadcast:

$$E_A \rightarrow B, C : X_A'', Sig_A(X_A''), \text{where } X_A'' = -\Omega_A + mP. \tag{27}$$

$$B \rightarrow E_A, C : X_B, Sig_B(X_B), \text{where } X_B = r_B P. \tag{28}$$

$$C \rightarrow E_A, B : X_C, Sig_C(X_C), \text{where } X_C = r_C P. \tag{29}$$

From Eqs. (5), (6), (28) and (29), E_A computes the tripartite key

$$\begin{aligned} K_{E_A} &= \hat{e}(\Omega_B + X_B, \Omega_C + X_C)^m \\ &= \hat{e}(P, P)^{(s_B + r_B)(s_C + r_C)m}. \end{aligned} \tag{30}$$

From Eqs. (4), (6), (27) and (29), B computes the tripartite key

$$\begin{aligned} K_B &= \hat{e}(\Omega_A + X_A'', \Omega_C + X_C)^{s_B + r_B} \\ &= \hat{e}(P, P)^{(s_B + r_B)(s_C + r_C)m}. \end{aligned} \tag{31}$$

From Eqs. (4), (5), (27) and (28), C computes the tripartite key

$$\begin{aligned} K_C &= \hat{e}(\Omega_A + X_A'', \Omega_B + X_B)^{s_C + r_C} \\ &= \hat{e}(P, P)^{(s_B + r_B)(s_C + r_C)m}. \end{aligned} \tag{32}$$

Hence, without knowing A's secret key, E_A is able to establish a communication session and subsequently agree on a session key with the legal entities by just forging A's signature.

5 Our Enhanced Scheme

In this section, we propose an improved one-round ID-based tripartite authenticated key agreement protocol based on the application scheme described in Sect. 3.2.

5.1 Protocol Improvement Description

Our improved scheme has the same system setting and key generation as the IDNIKS defined in Sect. 3.1.

Key Exchange:
Assume that $T_A, T_B, T_C \in Z_q^*$ are denoted as the timestamp generated by A, B and C respectively. A chooses random $r_A \in Z_q^*$, computes X_A from Eq. (11) and

$$Y_A = s_A(r_A P). \tag{33}$$

B chooses random $r_B \in Z_q^*$, computes X_B from Eq. (12) and

$$Y_B = s_B(r_B P). \tag{34}$$

C chooses random $r_C \in Z_q^*$, computes X_C from Eq. (13) and

$$Y_C = s_C(r_C P). \tag{35}$$

Assume that $Sig_i(\cdot)$ is denoted as the signature of an entity i. Then, over a public channel,

$$A \rightarrow B, C : M_A, Sig_A(M_A), \text{where } M_A = (ID_B, ID_C, X_A, Y_A, T_A). \tag{36}$$

$$B \rightarrow A, C : M_B, Sig_B(M_B), \text{where } M_B = (ID_A, ID_C, X_B, Y_B, T_B). \tag{37}$$

$$C \rightarrow A, B : M_C, Sig_C(M_C), \text{where } M_C = (ID_A, ID_B, X_C, Y_C, T_C). \tag{38}$$

Notice that the same private keys can be used as the entities' long term private keys s_A, s_B and s_C, and to support their corresponding signature schemes Sig_A, Sig_B and Sig_C. However, it is advisable to use different keys for the entities' static private keys, as well as for the computation of their respective signatures.

Message Verification:

$$\hat{e}(Y_A, P) \stackrel{?}{=} \hat{e}(X_A, \Omega_A) \tag{39}$$

$$\hat{e}(Y_B, P) \stackrel{?}{=} \hat{e}(X_B, \Omega_B) \tag{40}$$

$$\hat{e}(Y_C, P) \stackrel{?}{=} \hat{e}(X_C, \Omega_C) \tag{41}$$

After receiving M_B and M_C, A checks whether T_B and T_C lie within the specific acceptable time interval. Then, A verifies whether Eqs. (40) and (41) hold.
After receiving M_A and M_C, B checks whether T_A and T_C lie within the specific acceptable time interval. Then, B verifies whether Eqs. (39) and (41) hold.
After receiving M_A and M_B, C checks whether T_A and T_B lie within the specific acceptable time interval. Then, C verifies whether Eqs. (39) and (40) hold.

Consistency of the Verification Process:

$$\hat{e}(Y_A, P) = \hat{e}(s_A r_A P, P) = \hat{e}(r_A P, s_A P) = \hat{e}(X_A, \Omega_A) \tag{42}$$

$$\hat{e}(Y_B, P) = \hat{e}(s_B r_B P, P) = \hat{e}(r_B P, s_B P) = \hat{e}(X_B, \Omega_B) \tag{43}$$

$$\hat{e}(Y_C, P) = \hat{e}(s_C r_C P, P) = \hat{e}(r_C P, s_C P) = \hat{e}(X_C, \Omega_C) \tag{44}$$

Session key Generation:
If both the verification processes succeed, A, B and C computes the shared secret, Z_A, Z_B and Z_C respectively, where

$$\begin{aligned} Z_A &= \hat{e}(\Omega_B + X_B, \Omega_C + X_C)^{s_A + r_A} \\ &= \hat{e}(P, P)^{(s_A+r_A)(s_B+r_B)(s_C+r_C)}, \end{aligned} \tag{45}$$

$$\begin{aligned} Z_B &= \hat{e}(\Omega_A + X_A, \Omega_C + X_C)^{s_B + r_B} \\ &= \hat{e}(P, P)^{(s_A+r_A)(s_B+r_B)(s_C+r_C)}, \end{aligned} \tag{46}$$

$$\begin{aligned} Z_C &= \hat{e}(\Omega_A + X_A, \Omega_B + X_B)^{s_C + r_C} \\ &= \hat{e}(P, P)^{(s_A+r_A)(s_B+r_B)(s_C+r_C)}. \end{aligned} \tag{47}$$

Based on this common shared secret, A, B and C then calculate the tripartite session key K_A, K_B and K_C respectively, where

$$K_A = H(Z_A \parallel Y_A \parallel Y_B \parallel Y_C \parallel T_A \parallel T_B \parallel T_C) \tag{48}$$

$$K_B = H(Z_B \parallel Y_A \parallel Y_B \parallel Y_C \parallel T_A \parallel T_B \parallel T_C) \tag{49}$$

$$K_C = H(Z_C \parallel Y_A \parallel Y_B \parallel Y_C \parallel T_A \parallel T_B \parallel T_C) \tag{50}$$

5.2 Protocol Security Analysis

Known session key security. The session key of our protocol varies with every protocol run since it is established according to the values of the protocol entities' ephemeral private keys (r_A, r_B and r_C) in the specific session. Hence, the knowledge of previous session keys do not allow the adversary to derive any future session keys.

Perfect forward secrecy. Suppose that the entire long term private keys s_A, s_B and s_C have been disclosed to the adversary. In addition, assume that the adversary has also obtained some previous session keys established by the protocol entities. However, the adversary is unable to derive any other previously established session keys as derived in Eqs. (48) (49) and (50) since he does not possess the ephemeral private keys used in those particular protocol runs.

Key-Compromise Impersonation Resilience. Suppose that the long term private key s_A has been compromised and the adversary wishes to impersonate B in order to establish a session with A. However, he is unable to compute $Sig_B(M_B)$ since he is unable to forge the signature on behalf of B. Even if the adversary is able to counterfeit B's signature and broadcast the message in Eq. (37) to A and C, he still cannot compute the tripartite session key as he does not know s_B, which is required to calculate Z_B in Eq. (46). Now, the adversary wants to make use of s_A to derive the shared secret by computing Z_A (which is equivalent to Z_B) in Eq. (45). However, again, he fails as he does not have the knowledge of r_A. Suppose that the adversary then wishes to guess r_A or s_B in a random manner so as to derive the session key, his probability to succeed is only $\frac{1}{q}$, which is negligible as q is often chosen to be extremely large ($\geq$ 256 bits). Generally, the same situation would result when the long term key s_B or s_C is compromised as our enhanced protocol is symmetric. Hence, our enhanced protocol is able to withstand the KCI attack under any circumstances.

Insider Impersonation Resilience. Although an insider attacker, who wishes to impersonate B, could compute the session key by using the legal method, he could not forge the signature on behalf of B. Even if the malicious insider replays any of B's previous messages, the participated entities would reject the message as the first verification process would fail since the timestamp which the legal entity received would be out of bound of the acceptable time interval. Hence, as long as B's signature is not able to be forged and the timestamp has not been modified by the insider attacker, our protocol is immune to the insider impersonation attack.

Outsider Impersonation Resilience. Suppose that an outsider attacker is able to forge A's signature by some means and he attempts to impersonate A in a communication run with B and C. With the additional verification process introduced in our enhancement scheme (Eqs. (39), (40) and (41)), the adversary can no longer impose his outsider impersonation attack as described in Sect. 4.2. For instance, if the adversary broadcasts the forged value X''_A as computed in Eq. (27), the legitimate entities would have detected the counterfeit by verifying Eq. (39) unsuccessfully. Suppose that the adversary now wishes to guess A's long term private key randomly so as to derive the shared secret, his probability of success is only $\frac{1}{q}$ which is again deemed negligible. Hence, even though forgability of the signature is a strong assumption, as long as A's long term private key is kept secret from the adversary, our protocol is able to withstand the outsider impersonation attack.

Unknown Key-Share Resilience. In our enhancement scheme, the identities of the communicating parties have been included in the signed message of M_A, M_B and M_C. This significantly prevents the attacker from launching the unknown key-share attack in various ways on our improved protocol. With this, a stronger sense of authentication can be achieved explicitly.

Key Control Resilience. Apparently in our protocol, no single protocol participant could force the session key to a predetermined or predicted value since the session key of our protocol is derived by using the long term and ephemeral private keys of all the protocol participants, as well as their corresponding timestamps employed in that particular session.

6 Conclusion

Tso et al's IDNIKS is impractical since a non-interactive scheme for three or more parties cannot resist the KCI attack and the insider impersonation attack under any circumstances. Furthermore, we have also pointed out the demerits of their IDNIKS-based tripartite application scheme by launching several impersonation attacks in this paper. Based on these defects, we have proposed our improved tripartite authenticated key agreement scheme which includes an extra timestamp and the communicating entities' identities in the broadcasted messages during the key exchange stage. In addition, we have also introduced a two-stage verification process before the session key computation stage in order to authenticate the received messages and prevents all kinds of impersonation attacks. More significantly, we have carried out a detailed security analysis to scrutinize our enhanced scheme heuristically. In a nutshell, we have proven our enhanced one-round ID-based tripartite authenticated key agreement protocol to be secure against various cryptographic attacks, while preserving the desired security attributes of a key agreement protocol.

References

1. S. S. Al-Riyami and K. G. Paterson, Tripartite Authenticated Key Agreement Protocols from Pairings, *Cryptology ePrint Archive*: Report, (035)(2002).
2. Z. H. Cheng, L. Vasiu, and R. Comley, Pairing-based One-round Tripartite Key Agreement Protocols, *Cryptology ePrint Archive*: Report (079)(2004).
3. H. Y. Chien, Comments: Insider Attack on Cheng et als Pairing-based Tripartite Key Agreement Protocols, *Cryptology ePrint Archive*: Report, (013)(2005).
4. H. Y. Chien and R. Y. Lin, An Improved Tripartite Authenticated Key Agreement Protocol Based on Weil Pairing, *Int. J. Appl. Sci. Eng.*, 2005. 3, 1.
5. J. S. Chou, C. H. Lin, C. H. Chiu, Weakness of Shims New ID-based Tripartite Multiple-key Agreement Protocol, *Cryptology ePrint Archive*: Report, (457)(2005).
6. A. Joux, A One-round Protocol for Tripartite Diffie-Hellman, *Proceedings of the 4th International Algorithmic Number Theory Symposium (ANTS-IV)*, LNCS vol. 1838, July, 2000, pp. 385-394.
7. M. H. Lim, S. G. Lee, Y. H. Park, H. J. Lee, An Enhanced One-round Pairing-based Tripartite Authenticated Key Agreement Protocol, *Cryptology ePrint Archive*: Report, (142)(2007).
8. C. H. Lin, H. H. Li, Secure One-Round Tripartite Authenticated Key Agreement Protocol from Weil Pairing, *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, pp. 135-138.
9. D. Nalla, ID-based Tripartite Key Agreement with Signatures", *Cryptology ePrint Archive*: Report, (144)(2003).
10. D. Nalla and K. C. Reddy, ID-based tripartite Authenticated Key Agreement Protocols from pairings", *Cryptology ePrint Archive*: Report, (004)(2003).
11. K. Shim, Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols", *Cryptology ePrint Archive*: Report, (122)(2003).
12. K. Shim, Efficient ID-based Authenticated Key Agreement Protocol based on Weil Pairing, *Electronics Letters*, vol. 39, no. 8, April, 2003, pp. 653-654.
13. K. Shim, Efficient One-round Tripartite Authenticated Key Agreement Protocol from Weil Pairing, *Electronics Letters*, vol. 39, no. 2, January, 2003, pp. 208-209.
14. H. M. Sun and B. T. Hsieh, Security Analysis of Shims Authenticated Key Agreement Protocols from Pairings, *Cryptology ePrint Archive*: Report, (113)(2003).
15. R. Tso, T. Okamoto, T. Takagi, E. Okamoto, An ID-based Non-Interactive Tripartite Key Agreement Protocol with K-Resilience, *Communications and Computer Networks 2005*, pp. 38-42.
16. Y. Xun, Efficient ID-based Key Agreement from the Weil Pairing, *Electronics Letters*, vol. 39, no. 8, January, 2003, pp. 206-208.

17. S. B. Wilson, and A. Menezes, Authenticated Diffie-Hellman key agreement protocols, *Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC 98)*, LNCS, vol. 1999, pp. 339-361.
18. S. B. Wilson, D. Johnson and A. Menezes, Key Agreement Protocols and their Security Analysis, *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, vol. 1355, LNCS, pp. 339-361. Springer-Verlag, 1998.
19. F. G. Zhang, S. L. Liu, K. J. Kim, ID-based One Round Authenticated Tripartite Key Agreement Protocol with Pairings, *Cryptology ePrint Archive*: Report, (122)(2002).