Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols

Kyungah Shim

KISA (Korea Information Security Agency),
78, Garak-Dong, Songpa-Gu, Seoul 138-803, Korea
kashim@kisa.or.kr

Abstract. Recently, Al-Riyami and Paterson [1] proposed four authenticated tripartite key agreement protocols which make use of Weil pairing. In this paper, we show that the protocols are insecure against the man-in-the middle attack, key compromise impersonation attack and several known-key attacks.

Key words : Weil pairing, Tate pairing, tripartite authenticated key agreement protocol.

1 Introduction

Authenticated key agreement protocols are cryptographic protocols by which two or more entities that communicate over an adversarially controlled network can generate a common secret key. These protocols are essential for enabling the use of symmetric-key cryptography to protect transmitted data. As such they are a central piece for building secure communications, and perhaps the most commonly used cryptographic protocols.

Recently, there have been proposed several new cryptosystems based on bilinear pairings. In fact, the existence of Weil pairing and Tate pairing was thought to be a bad thing in cryptography. It was shown that the the discrete logarithm problem in supersingular curves was reducible to that in an extension of underlying field via Weil pairing [8]. This led supersingular curves to be avoided from cryptographic use. This situation changed with the work of Boneh-Franclin's ID-based encryption scheme [3] and Joux's one round tripartite Diffie-Hellman protocol [4]. However, like the basic Diffie-Hellman key agreement protocol, Joux's protocol also suffers from the man-in-the-middle attack because it does not attempt to authenticate the communicating entities. Al-Riyami and Paterson [1] proposed four tripartite authenticated key agreement protocols to provide implicit key authentication with Joux's protocol by incorporating certified public keys. The protocols use ideas from Joux's protocol and the MTI and MQV protocols. And they analyzed a number of ad hoc attacks on the protocols and compare the computational and communication efficiency of their protocols. In this paper, we show that Al-Riyami-Paterson's protocols are vulnerable to the man-in-the middle attack, key compromise impersonation attack and several known-key attacks.

2 Desirable security attributes

To clarify the threats protocols may be subject to and to motivate the need for specific security attributes, we describe the types of adversaries which is based on the adversaries' roles;

- a passive adversary: an adversary who is capable only of recording (eavesdropping) the protocol runs,
- an active adversary: an adversary who may also transmit, alter, delete, inject data and interleave multiple instantiations of the same protocol.

A secure protocol should be able to withstand both passive adversary and active adversary. A robust protocol should prevent an insider as well as an outsider. Such adversaries are based on the type of information available to them;

- an outsider: an adversary with no special knowledge beyond that generally available, e.g., by eavesdropping on protocol messages over open channels,
- an insider: an adversary with access to additional information (e.g., past session keys, long-term private key or secret partial information) obtained by some privileged means (e.g., physical access to private computer, conspiracy, etc.).

Let A, B and C be three honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. *Key authentication* is the property whereby one entity is assured that no other entity aside form specifically identified other entities may gain access to a particular secret key. Key authentication is independent of the actual possession of such key by the other entities, or knowledge of such actual possession by the first entity; in fact, it need not involve any action whatsoever by the other entities. For this reason, it is sometimes referred to more precisely as *implicit key authentication*. A key agreement protocol which provides implicit key authentication to all the participating entities is called an *authenticated key agreement* (AK) protocol. In addition to implicit key authentication, a number of desirable *security attributes* of AK protocols have been identified.

1. **Known key security.** Each run of a key agreement between A, B and C should produce a unique secret key; such keys are called *session* keys. A protocol should still achieve its goal in the face of an adversary who has learned some other session keys.
2. **Forward secrecy.** If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities is not affected.
3. **Key-compromise impersonation resilience.** Suppose A's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A, since it is precisely this value that identifies A. However, it may be desirable in some circumstances that this loss does not enable the adversary to impersonate other entities to A.

4. **Unknown key-share resilience.** Entity B cannot be coerced into sharing a key with entity A without B's knowledge, i.e., when B believes the key is shared with some entity E, and A believes the key is shared with B.

3 Al-Riyami-Paterson's authenticated tripartite key agreement protocols

The basic setting is the same as in [3]. We assume that the public domain parameters $(p, q, E, P, \hat{e})$ are common to all entities. Also, we will assume that static public keys are exchanged via certificates. $Cert_A$ denotes A's public-key certificate, containing a string of information that uniquely identifies A (such as A's name and address), her static public key $Y_A = x \cdot P$, and a certifying authority CA's signature over this information. Similarly, $Cert_B$ and $Cert_C$ are the certificates for entities B and C with $Y_B = y \cdot P$ and $Y_C = z \cdot P$ as their public keys, respectively. Now, we describe Al-Riyami-Paterson's tripartite authenticated key agreement protocols, TAK-1, TAK-2, TAK-3 and TAK-4.

Protocol messages: As usual, in the protocols below, short-term secret keys $a,\ b,\ c \in \mathbb{Z}_q^*$ are selected uniformly at random by A, B and C respectively.

$$\begin{array}{ll} (1) & A:\ P_A = a \cdot P,\ \ Cert_A \\ (2) & B:\ P_B = b \cdot P,\ \ Cert_B \\ (3) & C:\ P_C = c \cdot P,\ \ Cert_C \end{array}$$

• **TAK-1**

$$\begin{array}{rcl} K_A & = & \hat{e}(bP,\ cP)^a \cdot \hat{e}(yP,\ zP)^x \\ K_B & = & \hat{e}(aP,\ cP)^b \cdot \hat{e}(xP,\ zP)^y \\ K_C & = & \hat{e}(aP,\ bP)^c \cdot \hat{e}(xP,\ yP)^z \end{array}$$
$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P,\ P)^{abc+xyz}.$$

• **TAK-2**

$$\begin{array}{rcl} K_A & = & \hat{e}(bP,\ zP)^a \cdot \hat{e}(yP,\ cP)^a \cdot \hat{e}(bP,\ cP)^x \\ K_B & = & \hat{e}(aP,\ zP)^b \cdot \hat{e}(xP,\ cP)^b \cdot \hat{e}(aP,\ cP)^y \\ K_C & = & \hat{e}(aP,\ yP)^c \cdot \hat{e}(xP,\ bP)^c \cdot \hat{e}(aP,\ bP)^z \end{array}$$
$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P,\ P)^{(ab)z+(ac)y+(bc)x}.$$

• **TAK-3**

$$\begin{array}{rcl} K_A & = & \hat{e}(yP,\ cP)^x \cdot \hat{e}(bP,\ zP)^x \cdot \hat{e}(yP,\ zP)^a \\ K_B & = & \hat{e}(aP,\ zP)^y \cdot \hat{e}(xP,\ cP)^y \cdot \hat{e}(xP,\ zP)^b \\ K_C & = & \hat{e}(aP,\ yP)^z \cdot \hat{e}(xP,\ bP)^z \cdot \hat{e}(xP,\ yP)^c \end{array}$$
$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P,\ P)^{(xy)c+(xz)b+(yz)a}.$$

• **TAK-4**

$$\begin{aligned} K_A &= \hat{e}(bP + H(bP||yP)yP,\ cP + H(cP||zP)zP)^{a+H(aP||xP)x} \\ K_B &= \hat{e}(aP + H(aP||xP)xP,\ cP + H(cP||zP)zP)^{b+H(bP||yP)y} \\ K_C &= \hat{e}(aP + H(aP|xP)xP,\ bP + H(bP||yP)yP)^{c+H(cP||zP)z} \\ K_{ABC} = K_A = K_B = K_C &= \hat{e}(P,\ P)^{(a+H(aP|xP)x)(b+H(bP||yP)y)(c+H(cP||zP)z)}. \end{aligned}$$

Protocols TAK-1, TAK-2, and TAK-3 have their roots in the MTI protocols [7]. TAK-4 is modelled on the MQV protocol [6] but avoids that protocol's unknown key-share weakness [5] by using a cryptographic hash function H to combine long-term and short-term secret keys.

4 Cryptanalysis of Al-Riyami-Paterson's tripartite authenticated key agreement protocols

In this section, we present several attacks on TAK-1, TAK-2, TAK-3 and TAK-4. In fact, Al-Riyami and Paterson [1] proposed authenticated tripartite key agreement protocols to provide implicit key authentication with Joux's protocol by incorporating certified public keys. However, TAK-2 does not satisfy the implicit key authentication attribute, i.e., the protocol is still insecure the man-in-the-middle attacks. Also, all the protocols are insecure against the key-compromise impersonation attacks.

4.1 Man-in-the-middle attack on the TAK-2 protocol

When A, B and C broadcast P_A, P_B and P_C an adversary E creates ephemeral secret keys a', b' and c' and replaces P_A, P_B and P_C with $P'_A = a' \cdot P$, $P'_B = b' \cdot P$ and $P'_C = c' \cdot P$, respectively. Then A, B and C forms the session keys K_A, K_B and K_C, respectively.

$$\begin{aligned} K_A &= \hat{e}(P'_B,\ zP)^a \cdot \hat{e}(yP,\ P'_C)^a \cdot \hat{e}(P'_B,\ P'_C)^x = \hat{e}(P,\ P)^{ab'z+ac'y+b'c'x}, \\ K_B &= \hat{e}(P'_A,\ zP)^b \cdot \hat{e}(xP,\ P'_C)^b \cdot \hat{e}(P'_A,\ P'_B)^y = \hat{e}(P,\ P)^{a'bz+bc'x+a'c'y}, \\ K_C &= \hat{e}(P'_A,\ yP)^c \cdot \hat{e}(xP,\ P'_B)^c \cdot \hat{e}(P'_A,\ P'_B)^z = \hat{e}(P,\ P)^{a'cy+b'cy+a'b'z}. \end{aligned}$$

Then E who knows the values a', b' and c' is also able to compute these session keys from known values P_A, P_B and P_C as follows ;

$$\begin{aligned} K_A &= \hat{e}(aP,\ zP)^{b'} \cdot \hat{e}(yP,\ aP)^{c'} \cdot \hat{e}(xP,\ P)^{b'c'} = \hat{e}(P,\ P)^{ab'z+ac'y+b'c'x}, \\ K_B &= \hat{e}(bP,\ zP)^{a'} \cdot \hat{e}(xP,\ bP)^{c'} \cdot \hat{e}(yP,\ P)^{a'c'} = \hat{e}(P,\ P)^{a'bz+bc'x+a'c'y}, \\ K_C &= \hat{e}(cP,\ yP)^{a'} \cdot \hat{e}(xP,\ cP)^{b'} \cdot \hat{e}(zP,\ P)^{a'b'} = \hat{e}(P,\ P)^{a'cy+b'cy+a'b'z}. \end{aligned}$$

Thus, TAK-2 is still insecure against the man-in-the-middle attack.

4.2 Key-compromise impersonation attacks on TAK-2, TAK-3, and TAK-4

According to Al-Riyami and Paterson's security analysis on their protocols, only TAK-1 is insecure against the key-compromise impersonation attack. But, we show that the other protocols are still vulnerable to several kinds of the key-compromise impersonation attacks.

• **Key-compromise impersonation attack on TAK-2.** Suppose that the long-term key x of A is compromised to an adversary E. Then E can impersonate C to A and B on TAK-2. When A and B broadcast $P_A = a \cdot P$ and $P_B = b \cdot P$, respectively, E creates ephemeral secret keys a' and b' and replaces P_A and P_B with $P'_A = a' \cdot P$ and $P'_B = b' \cdot P$, respectively. Simultaneously, E chooses a random number c' and broadcasts the value $P_C = c' \cdot P$ masquerading C. We denote E_C the adversary E masquerading C.

$$\begin{array}{lll} (1) & A & : \ P_A = a \cdot P \ \rightarrow \ P'_A = a' \cdot P, \ Cert_A \\ (2) & B & : \ P_B = b \cdot P \ \rightarrow \ P'_B = b' \cdot P, \ Cert_B \\ (3) & E_C & : \qquad P_C = c' \cdot P, \quad Cert_C \end{array}$$

Then A and B compute the session keys K_A and K_B as follows:

$$\begin{aligned} K_A &= \hat{e}(b'P,\ zP)^a \cdot \hat{e}(yP,\ c'P)^a \cdot \hat{e}(b'P,\ c'P)^x = \hat{e}(P,\ P)^{ab'z+ac'y+b'c'x} \\ K_B &= \hat{e}(a'P,\ zP)^b \cdot \hat{e}(xP,\ c'P)^b \cdot \hat{e}(a'P,\ c'P)^y = \hat{e}(P,\ P)^{a'bz+b'cx+a'c'y}. \end{aligned}$$

The adversary E can compute K_A and K_B from known values $P_A = a \cdot P$, $P_B = b \cdot P$, and secret values x, a', b' and c' :

$$\begin{aligned} K_A &= \hat{e}(aP,\ zP)^{b'} \cdot \hat{e}(yP,\ aP)^{c'} \cdot \hat{e}(xP,\ P)^{b'c'} = \hat{e}(P,\ P)^{ab'z+ac'y+b'c'x} \\ K_B &= \hat{e}(bP,\ zP)^{a'} \cdot \hat{e}(xP,\ bP)^{c'} \cdot \hat{e}(yP,\ P)^{a'c'} = \hat{e}(P,\ P)^{ab'z+bc'y+a'c'y}. \end{aligned}$$

Thus, E can compute K_A and K_B and succeed to impersonate C to A and B.

• **Key-compromise impersonation attack on TAK-3.** Suppose that the long-term keys x of A is compromised to an adversary E. When A and B broadcast $P_A = a \cdot P$ and $P_B = b \cdot P$, respectively, E generates secret keys a' and b' and replaces P_A and P_B with $P'_A = a' \cdot P$ and $P'_B = b' \cdot P$, respectively. Simultaneously, E chooses a random number c' and broadcasts the value $P_C = c' \cdot P$ masquerading C.

$$\begin{array}{lll} (1) & A & : \ P_A = a \cdot P \ \rightarrow \ P'_A = a' \cdot P, \ Cert_A \\ (2) & B & : \ P_B = b \cdot P \ \rightarrow \ P'_B = b' \cdot P, \ Cert_B \\ (3) & E_C & : \qquad P_C = c' \cdot P, \quad Cert_C \end{array}$$

Then A and B compute the session keys K_A and K_B as follows:

$$\begin{aligned} K_A &= \hat{e}(yP,\ c'P)^x \cdot \hat{e}(b'P,\ zP)^x \cdot \hat{e}(yP,\ zP)^a = \hat{e}(P,\ P)^{xyc'+xzb'+yza} \\ K_B &= \hat{e}(a'P,\ zP)^y \cdot \hat{e}(xP,\ c'P)^y \cdot \hat{e}(xP,\ zP)^b = \hat{e}(P,\ P)^{yza'+xyc'+xzb}. \end{aligned}$$

Then E is able to compute K_B by computing $\hat{e}(zP, yP)^{a'} \cdot \hat{e}(xP, yP)^{c'} \cdot \hat{e}(zP, bP)^{x}$. However, E cannot compute K_A since she does not compute the term $\hat{e}(P, P)^{yza}$ of K_A. Thus, if K_B is used to encrypt subsequent communications, then E can decrypt the encrypted message. But, the protocol participant A cannot decrypt the message. In fact, this attack is a partial key-compromise impersonation attack, i.e., the adversary succeeds to impersonate C to only B and recover the session key K_B.

• **Key-compromise impersonation attack on TAK-4.** The same attack can be applied to TAK-4. Suppose that the long-term keys x of A is compromised to an adversary E. The adversary E_C pretending to be C replaces P_A and P_B with P'_A and P'_B, respectively.

$$\begin{array}{lll} (1) & A : P_A = a \cdot P \rightarrow P'_A = a' \cdot P, \ Cert_A \\ (2) & B : P_B = b \cdot P \rightarrow P'_B = b' \cdot P, \ Cert_B \\ (3) & E_C : \qquad P_C = c' \cdot P, \quad Cert_C \end{array}$$

Then A and B compute the session keys K_A and K_B as follows:

$$\begin{array}{l} K_A = \hat{e}(P, \ P)^{(a+H(aP||xP)x)(b'+H(b'P||yP)y)(c'+H(c'P||zP)z)} \\ K_B = \hat{e}(P, \ P)^{(a'+H(a'P||xP)x)(b+H(bP||yP)y)(c'+H(c'P||zP)z)}. \end{array}$$

Then E is able to compute K_B from known values P_A and P_B and secret values a', b', c' and x by computing

$$\begin{array}{c} \hat{e}(bP, \ P)^{a'c'} \cdot \hat{e}(bP, \ H(c'P||zP)zP)^{a'} \cdot \hat{e}(P, \ H(bP||yP)yP)^{a'c'} \\ \cdot \hat{e}(H(bP||yP)yP, \ H(c'P||zP)zP)^{a'} \cdot \hat{e}(bP, \ H(a'P||xP)xP)^{c'} \\ \cdot \hat{e}(H(a'P||xP)bP, \ H(c'P||zP)zP)^{x} \cdot \hat{e}(H(a'P||xP)xP, \ H(bP||yP)yP)^{c'} \\ \cdot \hat{e}(H(bP||yP)yP, \ H(c'P||zP)zP)^{xH(a'P||xP)} \end{array}$$

However, E cannot compute K_A since she does not calculate the term

$$\hat{e}(P, \ P)^{ayzH(b'P||yP)H(c'P||zP)}$$

of K_A. Thus, the adversary succeeds to impersonate C to only B and recover the session key K_B.

4.3 Known-key attacks on TAK-1 and TAK-2

Before description of the known-key attacks on TAK-1 and TAK-2, we present a taxonomy of known-key attacks. The taxonomy splits attacks into Known-key passive attacks and Known-key active attacks, which is based on adversaries' roles. Based on adversaries' goals, the known-key active attacks category is further divided into two subcategories: Known-key active attacks without impersonation (an adversary's goal is only session key retrieval) and Known-key impersonation attacks (an adversary's goal is retrieval of the session key established with legitimate entities as well as impersonation). In particular, the known-key active attacks without impersonation subcategory contains the known-key conspiracy (KKC) attacks in the three or more party setting.

1. **Known-key passive (KKP) attacks**: an adversary obtains some session keys used previously and then uses this information to determine new session keys.
2. **Known-key active (KKA) attacks**: an adversary obtains some keys of the session established between the adversary (as a legitimate entity) and legitimate entities A, B or C and then uses this information to determine new session keys or the earlier session key between A, B and C, i.e., the adversary's goals are impersonation or session key retrieval.

   (1) **Known-key active (KKA) attacks without impersonation** : an adversary obtains some keys of the session established between the adversary and legitimate entities and then uses this information to determine new session keys or the earlier session key between A, B and C.

   - **Known-key Conspiracy (KKC) attacks**: adversaries enters a new session without impersonation and they collude and finally compute the earlier session key between A, B and C, from the keys of new session.

   (2) **Known-key Impersonation (KKI) attacks** : an adversary enters the session and impersonate himself/herself as a valid entity D with the previous session key and present key token, then finally compute the session key between A, B and C, i.e., the adversary's goals are impersonation as well as secret retrieval.

Now, we show how the attacks in the taxonomy can be realized to TAK-1 and TAK-2.

• **Known-key impersonation attack on TAK-1.** In [1], they showed that TAK-1 was not secure against key-compromise impersonation attack. In fact, their key-compromise impersonation attack on TAK-1 is a known-key attack by an insider who knows A's ephemeral secret key a. But, a certain known-key attack, so-called KKI attack is possible even though the adversary knows neither any long-term secret key nor any ephemeral secret key. The known-key impersonation attack can be launched as follows. The adversary E chooses a random number c and broadcasts $P_C = c \cdot P$ masquerading C.

$$\begin{array}{rrl} (1) & A & : \ P_A = a \cdot P, \ Cert_A \\ (2) & B & : \ P_B = b \cdot P, \ Cert_B \\ (3) & E_C & : \ P_C = c \cdot P, \ Cert_C \end{array}$$

The the shared secret value is

$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P, \ P)^{abc+xyz}.$$

E_C now induces A or B to reveal the key, K_{ABC} established in the session between them. Because A and B believe that the session key should be known

to C this may be reasonable assumption (refer to [2]). Then, E can obtain the value $\hat{e}(P,\ P)^{xyz}$ by computing

$$K_{AB}/\hat{e}(P,\ P)^{abc} = \hat{e}(P,\ P)^{abc+xyz}/\hat{e}(P,\ P)^{abc} \hat{=} e(P,\ P)^{xyz}.$$

Finally, E can impersonate any entity (A, B or C) to the others and computes all the session keys for subsequent protocol sessions. Indeed, E starts the protocol with B and C pretending to be A, chooses a random number a and broadcasts a message $P_A = a \cdot P$. B and C also broadcast $P_B = b \cdot P$ and $P_C = c \cdot P$. Then, although E does not know A's long-term secret key x, E (pretending to be A) can compute the session key K_{E_A} which is equal to K_B and K_C from the known value $\hat{e}(P,\ P)^{xyz}$:

$$\begin{aligned} K_{E_A} &= \hat{e}(bP,\ cP)^a \cdot \hat{e}(P,\ P)^{xyz} \\ K_B &= \hat{e}(aP,\ cP)^b \cdot \hat{e}(xP,\ zP)^y \\ K_C &= \hat{e}(aP,\ bP)^c \cdot \hat{e}(xP,\ yP)^z. \end{aligned}$$

Finally, E can impersonate to B as well as session key retrieval. Thus, the term $\hat{e}(P,P)^{xyz}$ which consists of only long-term secret keys of all related entities should be avoided in the resulting shared secret preventing this kind of attack.

• **Known-key conspiracy attack on TAK-2.** According to Al-Riyami and Paterson, TAK-2 is secure against the known-key attacks. It is due to the lack of the consideration on three-party setting. Unlike two-party setting, in the three-party setting, we should consider some attacks mounted by participant's conspiracy. Now, we show that TAK-2 is vulnerable to the KKC attack which is a combination of a triangle attack and a conspiracy attack. $Cert_D$ and $Cert_E$ are the certificates for entities D and E and $Y_D = v \cdot P$ and $Y_E = w \cdot P$ as their public keys, respectively. Assume that D and E who are dishonest entities. Given certain assumption that release of session keys, if D and E conspire then they can recover the earlier session key established between A, B and C. The known-key conspiracy attack on TAK-2 is executed as follows.

1. D and E eavesdrops on a session between A, B and C.

$$\begin{aligned} &(1)\ \ A:\ P_A = a \cdot P,\ Cert_A \\ &(2)\ \ B:\ P_B = b \cdot P,\ Cert_B \\ &(3)\ \ C:\ P_C = c \cdot P,\ Cert_C \end{aligned}$$

2. And then D and E starts sessions with A, B and C in which use information gained during step 1. As a result, D and E do not obtain the session keys, K_{ADE}, K_{BDE} and K_{CDE}, established in these sessions.

$$\begin{aligned} &(1)\ A:\ a' \cdot P,\ Cert_A \quad (1')\ B: b' \cdot P \quad Cert_B \quad (1'')\ C: c' \cdot P \quad Cert_C \\ &(2)\ D:\ b \cdot P,\ Cert_D \quad (2')\ D: a \cdot P \quad Cert_D \quad (2'')\ D: a \cdot P \quad Cert_D \\ &(3)\ E:\ c \cdot P,\ Cert_C \quad (3')\ E: c \cdot P \quad Cert_E \quad (3'')\ E: b \cdot P \quad Cert_E. \end{aligned}$$

$$\begin{aligned} K_{ADE} &= \hat{e}(P,\ P)^{a'bw} \cdot \hat{e}(P,\ P)^{a'cv} \cdot \hat{e}(P,\ P)^{bcx} \\ K_{BDE} &= \hat{e}(P,\ P)^{b'aw} \cdot \hat{e}(P,\ P)^{b'cv} \cdot \hat{e}(P,\ P)^{acy} \\ K_{CDE} &= \hat{e}(P,\ P)^{c'aw} \cdot \hat{e}(P,\ P)^{c'bv} \cdot \hat{e}(P,\ P)^{abz}. \end{aligned}$$

3. D and E now induces A, B and C to reveal the keys K_{ADE}, K_{BDE} and K_{CDE} established in the sessions between them. Because A, B and C believe that the session keys should be known to D and E this may be reasonable assumption.
4. With this information, D and E can recover the key K_{ABC} established between A, B and C as follows ;

$$K_{ABC} = K_{ADE} \cdot K_{BDE} \cdot \ K_{CDE} \cdot (\hat{e}(aP,\ P)^{b'w} \cdot \hat{e}(cP,\ P)^{a'w})^{-1}$$
$$\cdot(\hat{e}(aP,\ P)^{b'w} \cdot \hat{e}(cP,\ P)^{b'w} \cdot \hat{e}(aP,\ P)^{c'w} \cdot \hat{e}(bP,\ P)^{c'v})^{-1}.$$

In fact, this attack requires more assumptions than usual. Plausible attack scenario is described in [2]. Of course, the triangle attack can be prevented if a key derivation function is used to derive a session key from the shared secret. But, without using additional function, a robust protocol should be against those kinds of attacks.

4.4 Summary

This section compares the security of the protocols presented in previous sections.

	TAK-1	**TAK-2**	**TAK-3**	**TAK-4**
Implicit Key Authentication	Yes	No	Yes	Yes
Known-Key Security	No	No	Yes	Yes
Key-Compromise Impersonation	No	No	No	No

5 Conclusion

Al-Riyami and Paterson [1] proposed a suite of tripartite authenticated key agreement protocols from Weil pairing. Their heuristic analysis showed that TAK-4 is resistant to all known attacks except the public key substitute unknown key-share attack which is thwarted via robust registration procedure. In this paper, we have shown that TAK-2, TAK-3 and TAK-4 are still insecure against several active attacks including the key-compromise impersonation attacks. Thus, it is fair to say that constructing a usable tripartite authenticated key agreement protocol satisfying the security attributes described in the section 2 is still an open problem.

References

1. S. S. Al-Riyami and K. G. Paterson, Authenticated three party key agreement protocols from pairing, available from eprint.iacr.org.
2. M. Bumester, On the risk of opening distributed keys, Advances in Cryptology; Crypto'94, LNCS 839, 1994, pp. 308-317.

3. D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, Advances in cryptology; Crypto'01, Springer-Verlag, pp. 213-229.
4. A. Joux, A one round protocol for tripartite Diffie-Hellman, Proc. of ANTS IV, LNCS 1838, 2000, pp. 385-394.
5. B. Kaliski, An unknown key-share attack on the MQV key agreemnent protocol, ACM Trans. on Information and System Security, 4(3): 275-288, 2001.
6. L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, An efficient protocol for authenticated key agreement, Technical report CORR 98-05, University of Waterloo, 1998.
7. T. Matsumoto, Y. Takashima, and H. Imai, On seeking smart public-key distribution systems, Trans. IEICE of Japan, E69:99-106, 1986.
8. A. J. Menezes, T. Okamoto and S. A. Vanstone, Reducing elliptic curve logarithms in a finite field, *IEEE Transaction on Information Theory*, vol. 39, no. 5, pp. 1639-1646, 1993.