Cryptanalysis of ID-Based Authenticated Key Agreement Protocols from Bilinear Pairings (Short Paper)*

Kyung-Ah Shim[1] and Seung-Hyun Seo[2]

[1] Department of Mathematics,
Ewha Womans University, Seoul, Korea
kashim@ewha.ac.kr
[2] Graduate School of Information Securities,
Center for Information Security Technologies (CIST),
Korea University, Seoul, Korea
seosh@korea.ac.kr

Abstract. Recently, a number of ID-based authenticated key agreement protocols from bilinear pairings have been proposed. In this paper we present security analysis of four ID-based authenticated key agreement protocols from pairings proposed in [11, 12, 7, 18]. These results demonstrate that no more ID-based authenticated key agreement protocols should be constructed with such ad-hoc methods, i.e, the formal design methodology as in [1, 2, 3, 10] should be employed in future design.

1 Introduction

In ID-based cryptography [14], the main idea is to simplify public-key and certificate management by using a user's identity (e.g., its email address) as its public key. For this to be possible, the ID-based system requires a trusted third party, typically called a Private Key Generator, to generate user private keys from its master secret and the user's identity. Such cryptosystems alleviate the certificate overhead and solve the problems of PKI technology: certificate management including storage, distribution and the computational cost of certificate verification. Since Boneh and Franklin's ID-based encryption scheme based on Weil pairing [6], bilinear pairings of algebraic curves have initiated some completely new fields in cryptography, making it possible to realize cryptographic primitives that were previously unknown or impractical.

At first, Joux [9] proposed a one round tripartite Diffie-Hellman key agreement protocol based on Weil pairings. However, like the basic Diffie-Hellman key agreement protocol [8], Joux's protocol also suffers from man-in-the-middle attacks because it does not attempt to authenticate the communicating entities. Smart [15] proposed an ID-based two-party authenticated key agreement (AK)

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2005-217-C00002), and by the second Brain Korea 21 Project.

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 410–419, 2006.

protocol which combines the idea of Boneh and Franklin with that of Joux. But, Shim [16] pointed out that Smart's protocol does not provide full forward secrecy and proposed a new protocol which provides full forward secrecy. However, it turns out the protocol is insecure against man-in-the-middle attacks [17]. Recently, Kim *et al* [11], Kim *et al* [12], Choi *et al* [7] and Xie [18] proposed two-party or three-party ID-based authenticated key agreement protocols from pairings. The authors argued that the protocols satisfy all the required security attributes for authenticated key agreement protocols described in [5]. In this paper we show that the four protocols do not achieve some attributes of them.

The rest of this paper is organized as follows. In the following Section, we introduce admissible pairings and ID-based public key infrastructures. In Section 3, we point out that Kim *et al* [11], Kim *et al* [12], Choi *et al* [7] and Xie [18] protocol are vulnerable to key-compromise impersonation attacks, unknown key-share attacks, signature forgery attacks and impersonation attacks, respectively. A concluding remark is given in Section 4.

2 Preliminaries

Admissible Pairings. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of a large prime order q. We write $\mathbb{G}_1$ additively and $\mathbb{G}_2$ multiplicatively. We assume that the discrete logarithm problems in both $\mathbb{G}_1$ and $\mathbb{G}_2$ are hard. We call $\hat{e}$ an *admissible pairing* if $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a map with the following properties:

1. Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and for all $a, b \in \mathbb{Z}$.
2. Non-degeneracy: There exists $P \in \mathbb{G}_1$ such that $\hat{e}(P,\ P) \neq 1$. In other words, the map does not send all pair $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$.
3. Computability: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}_1$.

The Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such admissible pairing, as in [6].

ID-based Public Key Infrastructures. An ID-based public key infrastructure involves a Private Key Generator (PKG) and users. It consists of **Setup** and **Private Key Extraction** algorithms. Let P be a generator of $\mathbb{G}_1$. Let $H : \{0,1\}^* \to \mathbb{Z}_q$ and $H_1 : \{0,1\}^* \to \mathbb{G}_1$ be two cryptographic hash functions.

[Setup]: PKG chooses a random $s \in \mathbb{Z}_q^*$ and set $P_{Pub} = sP$. PKG publishes the system parameters $\langle \mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{Pub}, H \text{ or } H_1 \rangle$ and keep s as a master secret.

[Private Key Extraction I]: For a given string $ID \in \{0,1\}^*$, compute the user's public key as $Q_{ID} = H_1(ID) \in \mathbb{G}_1$ and set the private key S_{ID} to be sQ_{ID}, where s is a master secret.

[Private Key Extraction II]: For a given string $ID \in \{0,1\}^*$, compute $\alpha = H(ID) \in \mathbb{Z}_q$ and set the private key d_{ID} to be $\frac{1}{\alpha+s}P$, where $\alpha P + sP$ is the public key corresponding to ID.

In the following section, Kim *et al*'s protocol [11], Kim *et al*'s protocol [12], Choi *et al*'s protocol [7] use the **Private Key Extraction I** algorithm, while Xie's protocol [18] adapts the **Private Key Extraction II** algorithm.

3 Cryptanalysis of Four ID-Based AK Protocols

3.1 Kim *et al*'s Tripartite AK Protocol with Multiple PKGs

Recently, Kim *et al* [11] proposed ID-based AK protocols among entities whose private keys were issued by different PKGs. We show that the 3PAK-MPE protocol for tripartite key agreement of their protocols is insecure against key-compromise impersonation (K-CI) attacks.

[Different PKGs Setup]. Let A, B and C be legitimate entities who have gotten their private keys from PKG_1, PKG_2 and PKG_3, respectively. The three different PKGs do not share the system parameters;

- PKG_i $(1 \le i \le 3)$ chooses its system parameters $\langle \mathbb{G}_1^i, \mathbb{G}_2^i, q^i, \hat{e}^i, P^i, P_{Pub}^i, H^i \rangle$, where $\mathbb{G}_1^i$ and $\mathbb{G}_2^i$ are groups with prime order q^i, P^i is a generator of $\mathbb{G}_1^i$, $\hat{e}^i : \mathbb{G}_1^i \times \mathbb{G}_1^i \to \mathbb{G}_2^i$ is the bilinear pairing and $H^i : \{0,1\}^* \to \mathbb{G}_1^i$ is a cryptographic hash function.
- PKG_i chooses a random $s^i \in \mathbb{Z}_{q^i}^*$ and set $P_{Pub}^i = s^i P^i$.
- Assume that all users agree on the hash function $H_3 : \{0,1\}^* \to \{0,1\}^k$ used to compute the resulting session key, where k is the length of the session key.

Consequently, the public/private key pairs of A, B and C are $(Q_A^1 = H^1(ID_A),\ S_A^1 = s^1 Q_A^1)$, $(Q_B^2 = H^2(ID_B),\ S_B^2 = s^2 Q_B^2)$, and $(Q_C^3 = H^3(ID_C),\ S_C^3 = s^3 Q_C^3)$, respectively.

■ **3PAK-MPE Protocol**

[The First Round]. Users A, B and C choose ephemeral private keys $\{a^i\}_{i=1}^3$, $\{b^i\}_{i=1}^3$ and $\{c^i\}_{i=1}^3$, respectively, where $a_i, b_i, c_i \in \mathbb{Z}_{q^i}^*, 1 \le i \le 3$. Then they compute $\{W_A^i = a^i P^i\}_{i=1}^3, \{W_B^i = b^i P^i\}_{i=1}^3$ and $\{W_C^i = c^i P^i\}_{i=1}^3$ and broadcast these values.

$$
\begin{array}{lll}
(1)\ A \longrightarrow B,\ C: & W_A^1 = a^1 P^1, & W_A^2 = a^2 P^2,\ \ W_A^3 = a^3 P^3, \\
(2)\ B \longrightarrow A,\ C: & W_B^1 = b^1 P^1, & W_B^2 = b^2 P^2,\ \ W_B^3 = b^3 P^3, \\
(3)\ C \longrightarrow A,\ B: & W_C^1 = c^1 P^1, & W_C^2 = c^2 P^2,\ \ W_C^3 = c^3 P^3.
\end{array}
$$

After receiving the messages from the other entities, each entity computes the partial session keys. In detail, A computes partial keys K_{AB} and K_{AC} as follows;

$$K_{AB} = H_3(\hat{e}^1(S_A^1, W_B^1) || a^1 W_B^1 || \hat{e}^2(Q_B^2, a^2 P_{Pub}^2) || a^2 W_B^2)),$$

$$K_{AC} = H_3(\hat{e}^1(S_A^1, W_B^1) || a^1 W_C^1 || \hat{e}^3(Q_C^3, a^3 P_{Pub}^3) || a^3 W_C^3)).$$

Similarly, B computes partial keys K_{BA} and K_{BC} as follows;

$$K_{BA} = H_3(\hat{e}^1(Q_A^1, b^1 P_{Pub}^1) || b^1 W_A^1 || \hat{e}^2(S_B^2, W_A^2) || b^2 W_A^2)),$$

$$K_{BC} = H_3(\hat{e}^2(S_B^2, W_C^2)||b^2 W_C^2||\hat{e}^3(Q_C^3, b^3 P_{Pub}^3)||b^3 W_C^3)).$$

C also computes partial keys K_{CA} and K_{CB} as follows;

$$K_{CA} = H_3(\hat{e}^1(Q_A^1, c^1 P_{Pub}^1)||c^1 W_A^1||\hat{e}^3(S_C^3, W_A^3)||c^3 W_A^3),$$

$$K_{CB} = H_3(\hat{e}^2(Q_B^2, c^2 P_{Pub}^2)||c^2 W_B^2||\hat{e}^3(S_C^3, W_B^3)||c^3 W_B^3).$$

Then $K_{AB} = K_{BA}$, $K_{AC} = K_{CA}$ and $K_{BC} = K_{CB}$.

[The Second Round]. A, B, and C choose random numbers R_A, R_B, R_C and broadcast $\langle\{R_A\}_{K_{AB}}, \{R_A\}_{K_{AC}}\rangle$, $\langle\{R_B\}_{K_{BA}}, \{R_B\}_{K_{BC}}\rangle$, and $\langle\{R_C\}_{K_{CA}}, \{R_C\}_{K_{CB}}\rangle$, respectively, where $\{M\}_K$ denotes a symmetric encryption under the key K.

$$\begin{array}{ll} (1) & A \longrightarrow B,\ C : \{R_A\}_{K_{AB}},\ \ \{R_A\}_{K_{AC}}, \\ (2) & B \longrightarrow A,\ C : \{R_B\}_{K_{BA}},\ \ \{R_B\}_{K_{BC}}, \\ (3) & C \longrightarrow A,\ B : \{R_C\}_{K_{CA}},\ \ \{R_C\}_{K_{CB}}. \end{array}$$

The definition of key-compromise impersonation resilience attribute described in [5] 2 is originally defined on a two-party setting. But, the definition is easily extended to a multi-party setting as follows; Let $\{A_1, \cdots, A_n\}$ ba a set of communicating entities. Suppose that m $(m < n)$ long-term private keys of A_i $(i = 1, \cdots, m)$ are compromised to an adversary. Then the K-CI resilience implies that the adversary can neither impersonate the other entities A_j $(j = m+1, \cdots, n)$ to A_i $(i = 1, \cdots, m)$ nor obtain the session keys computed by A_i $(i = 1, \cdots, m)$. Now, we show that the 3PAK-MPE protocol is insecure against a K-CI attack in the three-party setting.

■ K-CI Attacks on the 3PAK-MPE Protocol

Suppose that long-term private keys S_A^1 and S_B^2 of A and B, respectively, are compromised to an adversary E and E wants to impersonate C to A and B.

1. First, E chooses random numbers $c^i, u^i, v^i \in \mathbb{Z}_{q^i}^*$, $i = 1, 2, 3$ and computes $W_C^i = c^i P^i, U_A^i = u^i P^i, V_B^i = v^i P^i$, $i = 1, 2, 3$.
2. When A and B broadcast $\{W_A^i\}_{i=1}^3$ and $\{W_B^i\}_{i=1}^3$, respectively, E replaces them with $\{U_A^i\}_{i=1}^3$ and $\{V_B^i\}_{i=1}^3$, respectively, and simultaneously broadcast $\{W_C^i\}_{i=1}^3$ impersonating C. $E(C)$ denotes E masquerades as C.

$$\begin{array}{lll} (1) & A \longrightarrow B,\ C & : W_A^1,\ W_A^2,\ W_A^3 \implies U_A^1,\ U_A^2,\ U_A^3, \\ (2) & B \longrightarrow A,\ C & : W_B^1,\ W_B^2,\ W_B^3 \implies V_B^1,\ V_B^2,\ V_B^3, \\ (3) & E(C) \longrightarrow A,\ B & : \quad W_C^1,\ W_C^2,\ W_C^3. \end{array}$$

After receiving the messages, A computes the partial session keys K_{AB} and K_{AC} from $\{V_B^i\}_{i=1}^3$ and $\{W_C^i\}_{i=1}^3$ as follows;

$$K_{AB} = H_3(e^1(S_A^1, V_B^1)||a^1 V_B^1||e^2(Q_B^2, a^2 P_{Pub}^2)||a^2 V_B^2),$$

$$K_{AC} = H_3(e^1(S_A^1, W_B^1)||a^1 W_C^1||e^3(Q_C^3, a^3 P_{Pub}^3)||a^3 W_C^3)).$$

From S_A^1, S_B^2 and v^i $(i = 1, 2)$, E also computes K'_{AB} as follows;

$$K'_{AB} = H_3(e^1(S_A^1, V_B^1)||v^1 W_A^1||e^2(S_B^2, W_A^2)||v^2 W_A^2)).$$

Then, $K_{AB} = K'_{AB}$. However, E cannot obtain K_{AC} since E, who does not know S_C^3, cannot compute the term $e^3(Q_C^3, a^3 P_{Pub}^3)$ of K_{AC} Also, B computes K_{BA} and K_{BC} from $\{U_A^i\}_{i=1}^3$ and $\{W_C^i\}_{i=1}^3$ as follows;

$$K_{BA} = H_3(e^1(Q_A^1, b^1 P_{Pub}^1)||b^1 U_A^1||e^2(S_B^2, U_A^2)||b^2 U_A^2)),$$

$$K_{BC} = H_3(e^2(S_A^2, W_C^2)||b^2 W_C^2||e^3(Q_C^3, b^3 P_{Pub}^3)||b^3 W_C^3)).$$

Similarly, from S_A^1, S_B^2 and u^i $(i = 1, 2)$, E can compute K'_{BA} as follows;

$$K_{BA} = H_3(e^1(S_A^1, W_B^1)||u^1 W_B^1||e^2(S_B^2, U_A^2)||u^2 W_B^2)).$$

Then, $K_{BA} = K'_{BA}$. However, E cannot obtain K_{BC} since E, who does not know S_C^3, cannot compute the term $e^3(Q_C^3, b^3 P_{Pub}^3)$ of K_{BC}.

3. In the second round, when A and B broadcast $\langle\{R_A\}_{K_{AB}}, \{R_A\}_{K_{AC}}\rangle$ and $\langle\{R_B\}_{K_{BA}}, \{R_B\}_{K_{BC}}\rangle$, E replaces $\langle\{R_A\}_{K_{AB}}, \{R_A\}_{K_{AC}}\rangle$ and $\langle\{R_B\}_{K_{BA}}, \{R_B\}_{K_{BC}}\rangle$ with $\langle\{R_B\}_{K_{BA}}, \{R_A\}_{K_{AC}}\rangle$ and $\langle\{R_A\}_{K_{AB}}, \{R_B\}_{K_{BC}}\rangle$, respectively, and simultaneously broadcast $\langle\{R_A\}_{K_{AC}}, \{R_B\}_{K_{BC}}\rangle$ to A and B, impersonating C.

$$\begin{array}{lll} (1) & A \longrightarrow B,\ C: & \{R_A\}_{K_{AB}},\ \{R_A\}_{K_{AC}} \Longrightarrow \{R_B\}_{K_{BA}}, \{R_A\}_{K_{AC}}, \\ (2) & B \longrightarrow A,\ C: & \{R_B\}_{K_{BA}},\ \{R_B\}_{K_{BC}} \Longrightarrow \{R_A\}_{K_{AB}}, \{R_B\}_{K_{BC}}, \\ (3) & E(C) \longrightarrow A,\ B: & \{R_A\}_{K_{AC}},\ \{R_B\}_{K_{BC}}. \end{array}$$

4. After receiving $\{R_A\}_{K_{AB}}$ and $\{R_A\}_{K_{AC}}$ intended to A, A can obtain R_A by decrypting $\{R_A\}_{K_{AB}}$ and $\{R_A\}_{K_{AC}}$ under K_{AB} and K_{AC}, respectively. Then A computes the session key $SK_A = H_3(R_A||R_A||R_A)$ from the decrypted messages and its own choice R_A. Similarly, B also obtain R_B by decrypting $\{R_B\}_{K_{BA}}$ and $\{R_B\}_{K_{BC}}$ under K_{BA} and K_{BC}, respectively. Then B computes the session key $SK_B = H_3(R_B||R_B||R_B)$ from the decrypted messages and its own choice R_B. E also obtains R_A and R_B by decrypting $\{R_A\}_{K_{AB}}$ and $\{R_B\}_{K_{BA}}$ under K'_{AB} and K'_{BA}, respectively, because $K_{AB} = K'_{AB}$ and $K_{BA} = K'_{BA}$. Therefore, E can compute the session keys SK_A and SK_B calculated by A and B from R_A and R_B. Finally, E succeeds in impersonating C to both A and B as well as in obtaining the session keys SK_A and SK_B.

In the attack, E, can compute neither $\{R_A\}_{K_{AC}}$ nor $\{R_B\}_{K_{BC}}$ in the second round because E knows neither K_{AC} nor K_{BC}. But, E can obtain $\{R_A\}_{K_{AC}}$ and $\{R_B\}_{K_{BC}}$ from the messages sent by A and B, respectively and so replay them to A and B impersonating C as C's second message. Since the messages themselves cannot contain any information on the receivers, they can be reused as messages intended to other entities. Its weakness against the K-CI attacks are the lack of explicitness in messages transmitted. Thus, the attacks can be prevented by adding the ordered pair of identities in messages being signed, for example, $\{R_A\}_{K_{AB}}$ is replaced with $\{R_A||Q_B||Q_C\}_{K_{BA}}$ as described in [11]. But, in their paper, it is not mandatory but optional. Such a misused optional condition opens the door to the attacks.

3.2 Kim *et al*'s ID-Based Multiple AK Protocol

Kim *et al* [12] proposed an ID-based authenticated multiple-key agreement protocol (KRY protocol) which allows two entities to establish multiple session keys in a protocol run. We show that the KRY protocol is insecure against an unknown key-share (UK-S) attack and does not achieve forward secrecy in the case of the compromise of additional secret information.

■ **KRY Protocol**

(1) $A \longrightarrow B : P_A = aP,\ P'_A = a'P,\ T_A = H(P_A)H(P'_A)S_A + (a+a')P_{Pub}$,
(2) $B \longrightarrow A : P_B = bP,\ P'_B = b'P,\ T_B = H(P_B)H(P'_B)S_B + (b+b')P_{Pub}$.

Assume that A and B want to agree to four session keys. First, A sends (P_A, P'_A, T_A) to B. On the receipt of the message from A, B verifies

$$\hat{e}(T_A, P) = \hat{e}(H(P_A)H(P'_A)Q_A + P_A + P'_A,\ P_{Pub}).$$

If the equation holds, B sends (P_B, P'_B, T_B) to A and then computes four session keys as $K_B^{(1)} = \hat{e}(P_A,\ P_{Pub})^b, K_B^{(2)} = \hat{e}(P_A,\ P_{Pub})^{b'}, K_B^{(3)} = \hat{e}(P'_A,\ P_{Pub})^b, K_B^{(4)} = \hat{e}(P'_A, P_{Pub})^{b'}$. After receiving the message, A verifies

$$\hat{e}(T_B, P) = \hat{e}(H(P_B)H(P'_B)Q_B + P_B + P'_B,\ P_{Pub}).$$

If the equation holds, A computes the session keys $K_A^{(1)} = \hat{e}(P_B, P_{Pub})^a, K_A^{(2)} = \hat{e}(P'_B, P_{Pub})^a, K_A^{(3)} = \hat{e}(P_B, P_{Pub})^{a'}, K_A^{(4)} = \hat{e}(P'_B, P_{Pub})^{a'}$. Each entity takes the four values K^i $(i = 1, \cdots, 4)$ as the final session keys $K^{(1)} = \hat{e}(P,\ P)^{abs}, K^{(2)} = \hat{e}(P,\ P)^{ab's}, K^{(3)} = \hat{e}(P,\ P)^{a'bs}, K^{(4)} = \hat{e}(P,\ P)^{a'b's}$.

■ **UK-S Attacks on the KRY Protocol**

Suppose that an adversary E, who is a legitimate entity, has gotten her own long-term private key S_E. Then attack on the protocol is mounted as follows;

1. When A sends $\{P_A = aP,\ P'_A = a'P,\ T_A,\ ID_A\}$ to B, an adversary E intercepts it and computes (P_E, P'_E, T_E) as follows;
   - First, E chooses a random $r \in \mathbb{Z}_q^*$ and let $r = a + r'$. Then E can obtain $r'P$ by computing $rP - aP$.
   - Next, E takes P_E and P'_E as aP and $r'P$, respectively and computes her own signature on $\{P_E, P_{E'}\}$ as $T_E = H(P_A)H(P'_E)S_E + rP_{Pub}$. Note that E knows neither a nor r', while she knows $aP,\ r'P$ and r.

   Next, E sends (P_E, P'_E, T_E) together with her identity ID_E to B.
2. On the receipt of the message, B thinks that the protocol run is initiated by E. Then B verifies E's signature. In fact, the verification always holds, because T_E is E's valid signature on $\{P_E,\ P'_E\}$. B sends $\{P_B, P'_B, T_B, ID_B\}$ to E which forwards to A. Next, B computes four session keys as follows;

$$K_B^{(1)} = \hat{e}(P_E,\ P_{Pub})^b = \hat{e}(P,P)^{abs},\ K_B^{(2)} = \hat{e}(P_E,\ P_{Pub})^{b'} = \hat{e}(P,P)^{ab's},$$
$$K_B^{(3)} = \hat{e}(P'_E,\ P_{Pub})^b = \hat{e}(P,P)^{r'bs},\ K_B^{(4)} = \hat{e}(P'_E, P_{Pub})^{b'} = \hat{e}(P,P)^{r'b's}.$$

3. After receiving the message, A verifies B's signature and computes

$$K_A^{(1)} = \hat{e}(P_B,\ P_{Pub})^a = \hat{e}(P,P)^{abs},\ K_A^{(2)} = \hat{e}(P'_B,\ P_{Pub})^a = \hat{e}(P,P)^{ab's},$$
$$K_A^{(3)} = \hat{e}(P_B,\ P_{Pub})^{a'} = \hat{e}(P,P)^{a'bs},\ K_A^{(4)} = \hat{e}(P'_B, P_{Pub})^{a'} = \hat{e}(P,P)^{a'b's}.$$

4. Finally, A and B share the same two of four session keys, $K^{(1)} = \hat{e}(P,P)^{abs}$, $K^{(2)} = \hat{e}(P,P)^{ab's}$. A thinks that the session keys are shared with B, while B mistakenly believes that he shares the keys with E.

Finally, the UK-S attack on two of four session keys is successfully mounted. If A and B use the former two session keys for a subsequent encryption, serious consequences stated in [4] will be happened. Its weakness against the UK-S attack is due to the fact that an adversary E, who knows neither a and r', can generate its signature on $\{P_E = aP, P_{E'} = r'P\}$. In fact, it is known that all types of UK-S attacks can be prevented by adding identities of the communicating entities in inputs of a key derivation function [4]. However, to avoid the attack without using additional functions such as a key derivation function, the adapted signature should be designed so that only one, who knows both a and a', can generate its signature on $\{aP, a'P\}$.

■ **Forward Secrecy of the KRY Protocol**

Now, we show that the KRY protocol does not satisfy forward secrecy in the case of the compromise of additional secret information. Suppose that the long-term private keys, S_A and S_B of A and B, respectively, are compromised to an adversary E. Then E can obtain some equations related to each user's ephemeral private keys. Indeed, E, who knows S_A, can compute $(a+a')P_{Pub}$ from $T_A = H(P_A)H(P'_A)S_A + (a+a')P_{Pub}$ by computing $T_A - H(P_A)H(P'_A)S_A$. Similarly, E, who knows S_A, can compute $(b+b')P_{Pub}$ from $T_B = H(P_B)H(P'_B)S_B + (b+b')P_{Pub}$ by computing $T_B - H(P_B)H(P'_B)S_B$. Finally, E can compute the following equations;

$$\hat{e}((a+a')P_{Pub}, bP) = \hat{e}(P,P)^{(a+a')bs} \quad (1)$$
$$\hat{e}((a+a')P_{Pub}, b'P) = \hat{e}(P,P)^{(a+a')b's} \quad (2)$$
$$\hat{e}((b+b')P_{Pub}, aP) = \hat{e}(P,P)^{(b+b')as} \quad (3)$$
$$\hat{e}((b+b')P_{Pub}, a'P) = \hat{e}(P,P)^{(b+b')a's} \quad (4).$$

These relationships lead to serious consequences in the case of the compromise of additional secret information. If one session key of the past session, say $K^{(1)} = \hat{e}(P,P)^{abs}$, is compromised then the other three session keys, $K^{(2)}$, $K^{(3)}$ and $K^{(4)}$ are revealed, i.e., E can recover $K^{(2)} = \hat{e}(P,P)^{ab's}$ from the equation (3) by calculating $(3) \times K_1^{-1} = \hat{e}(P,P)^{ab's}$, $K^{(3)} = \hat{e}(P,P)^{a'bs}$ from the equation (1) by calculating $(1) \times K_1^{-1} = \hat{e}(P,P)^{a'bs}$, and $K^{(4)} = \hat{e}(P,P)^{a'b's}$ from the equation (2) by calculating $(2) \times K_2^{-1} = \hat{e}(P,P)^{a'b's}$. Thus, it does not satisfy forward secrecy in the case of the compromise of additional secret information.

In general, we note the compromise of long-term secret keys does not necessarily mean that they are obtained via an inversion of the long-term public key. Long-term secrets are in practice vulnerable secrets in the system; in a typical setting, they are stored on disk, perhaps protected by a password. Since users must store their secret keys for use in key computation, the secret keys may also be obtained through lack of suitable physical measures. An adversary is also able to obtain the session key used in any sufficiently old previous run of the protocol. In some environments (e.g., due to implementation and engineering decisions), the probability of compromise of session keys may be greater than that of long-term keys. In particular, when using cryptographic techniques of only moderate strength, the possibility exists that over time extensive cryptanalytic effort may uncover past session keys. These properties may be attractive for the robustness of the security in most commercial applications where customers does not always protect their key sufficiently. Thus, a secure protocol design will minimize the effects of such events.

3.3 Choi *at al*'s ID-Based AK Protocol

Choi *et al* [7] proposed two ID-based AK protocols satisfying the forward secrecy. Their protocol I uses a signature scheme to provide authentication; the authenticity of the ephemeral public keys in the protocol is assured by each user's signature. We show that the protocol I does not achieve authentication as intended, i.e., anyone can forge each user's signature.

■ **Protocol I**

$$(1)\ \ A \longrightarrow B\ :\ U_A = aP_{Pub},\ V_A = aS_A$$
$$(2)\ \ B \longrightarrow A\ :\ U_B = bP_{Pub},\ V_B = bS_B.$$

First, A sends (U_A, V_A) to B. On the receipt of the message from A, B verifies $\hat{e}(V_A, P) = \hat{e}(Q_A, U_A)$. If the equation holds, B sends $(U_B V_B)$ to A and computes $K_B = bU_A$. After receiving the message from B, A verifies $\hat{e}(V_B, P) = \hat{e}(Q_B, U_B)$. If the equation holds, A computes $K_A = aU_B$. The resulting session key is $K = kdf(K_A, Q_A, Q_B) = kdf(K_B, Q_A, Q_B) = kdf(absP, Q_A, Q_B)$, where kdf is a key derivation function.

■ **Signature Forgery Attack on the Protocol I**

In the protocol I, anyone can generate a valid pair (U_A, V_A) satisfying $\hat{e}(V_A, P) = \hat{e}(Q_A, U_A)$ as follows; an adversary chooses a at random and then computes $U_A = aP$ and $V_A = aQ_A$. Then the pair satisfies the verification equation;

$$\hat{e}(V_A, P) = \hat{e}(aQ_A, P) = \hat{e}(Q_A, aP) = \hat{e}(Q_A, U_A).$$

Therefore, an adversary, who does not know the corresponding long-term private key, can forge each user's signature on the ephemeral public key. In fact, the adversary cannot obtain the session key established in this session involved in this forgery attack. However, the signature scheme adapted to the cryptographic protocols should be secure against forgery attacks.

3.4 Xie's ID-Based AK Protocol with Escrow

Recently, Xie [18] showed that McCullagh and Barreto's AK protocol [13] is insecure against impersonation attacks. Then he proposed an improved protocol to defeat the attacks and argued that its protocol satisfies all the security attributes. We show that the protocol satisfies neither the implicit key authentication nor the K-CI resilience.

■ **Xie's Protocol**

This protocol uses the **Private Key Extraction II** algorithm. Let $H_1(ID_A) = a$ and $H_1(ID_B) = b$. First, A and B exchange the ephemeral public keys A_{KA} and B_{KA}.

$$(1)\ A \longrightarrow B\ :\ A_{KA} = x(bP + sP)$$
$$(2)\ B \longrightarrow A\ :\ B_{KA} = y(aP + sP).$$

Then, A and B compute $K_A = \hat{e}(B_{KA}, d_A)^{x+1}\hat{e}(P,P)^x$ and $K_B = \hat{e}(A_{KA}, d_B)^{y+1}\ \hat{e}(P,P)^y$, respectively. The resulting session key is $K = K_A = K_B = e(P,P)^{xy+x+y}$.

Now we show that Xie's protocol is insecure against impersonation attacks, i.e., an adversary can impersonate A to B at any time. The attack on the protocol is mounted as follows;

■ **Impersonation Attacks on Xie's Protocol**

Suppose that an adversary E wants to impersonate A to B. $E(A)$ denotes E masquerade as A. First, $E(A)$ sends $A_{KA} = -(bP + sP)$ to B impersonating A. After receiving the message, B sends $B_{KA} = y(aP + sP)$ and computes the session key

$$K_B = \hat{e}(-(bP + sP), d_B)^{y+1}\hat{e}(P,P)^y = \hat{e}(P,P)^{-y-1}\hat{e}(P,P)^y = \hat{e}(P,P)^{-1}.$$

By bilinearity of $\hat{e}$, the value $\hat{e}(P,P)^y$ disappears in the resulting session key. Thus, E is able to compute $K_B = \hat{e}(P,P)^{-1}$ from known value. Finally, E succeeds in impersonating A to B as well as in obtaining the session key K_B.

In above attack, an adversary can generate an ephemeral public key to confine the shared secret to a predictable value. Thus, Xie's protocol does not provide implicit key authentication attribute. From the attack, we can easily see that it is insecure against man-in-the-middle attacks and key-compromise impersonation attacks. The same attacks can be applied to Xie's ID-based AK protocol without escrow and AK protocol between members of distinct domains.

4 Conclusion

We have shown that four ID-based AK protocols are insecure against several active attacks including unknown key-share attacks and key-compromise impersonation attacks. Our results demonstrate that no more ID-based AK protocols should be constructed with such ad-hoc methods and the formal design methodology in [1, 2, 3, 10] should be employed in future design.

References

1. M. Bellare, R. Canetti, and H. Krawczyk, A modular approach to the design and analysis of authentication and key exchange protocols, Proc. 30th Annual Symposium on the Theory of Computing, ACM, pp. 419-428, 1998.
2. M. Bellare and P. Rogaway, Provably secure session key distribution; the three party case, Proc. 27th Annual Sym. on the Theory of Computing, ACM, pp. 57-66, 1995.
3. M. Bellare and P. Rogaway, Entity autentication and key distribution, Advances in Cryptology; Crypto'93, LNCS 773, Springer-Verlag, pp. 232-249, 1994.
4. S. Blake-Wilson, D. Johnson and A. Menezes, Unknown key-share attacks on the station-to-station (STS) protocol, PKC'99, LNCS 1560, Springer-Verlag, pp. 154-170, 1999.
5. S. Blake-Wilson and A. Menezes, Authenticated Diffie-Hellman key agreement protocols, Proc. of the 5th Annual Workshop on Selected Areas in Cryptography, SAC'98, LNCS 1556, Springer-Verlag, pp. 339-361, 1999.
6. D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, Advances in cryptology; Crypto'01, LNCS 2139, Springer-Verlag, pp. 213-229, 2001.
7. Y. J. Choie, E. Jeong and E. Lee, Efficient identity-based authenticated key agreement protocol from pairings, Applied Mathematics and Computation, 162(1), pp. 179-188, 2005.
8. W. Diffie, and M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, 22(6), pp. 644-654, 1976.
9. A. Joux, A one round protocol for tripartite Diffie-Hellman, ANTS IV, LNCS 1838, Springer-Verlag, pp. 385-394, 2000.
10. J. Katz and M. Yung, Scalable protocols for authenticated group key exchange, Advances in Cryptology; Crypto'03, LNCS 1807, Springer-Verlag, 2004.
11. K. Kim, H. Lee, and H. Oh, Enhanced ID-based authenticated key agreement protocols for a multiple independent PKG environment, ICICS'05, LNCS 3783, Springer-Verlag, pp. 323-335, 2004.
12. K. Kim, E. Ryu, and K. Yoo, ID-based authenticated multiple-key agreement protocol from pairing, International Conference on Computational Science and Its Applications, ICCSA'04, LNCS 3046, Springer-Verlag, pp. 672-680, 2004.
13. N. McCullagh, P. S. L. M. Barreto, A new two-party identity-based authenticated key agreement, CT-RSA'05, LNCS 3376, Springer-Verlag, pp. 262-274, 2005.
14. A. Shamir, Identity-based cryptosystems and signature schemes, Advances in Cryptology; Crypto'84, LNCS 196, Springer-Verlag, pp. 47-53, 1884.
15. N. Smart, An ID-based authenticated key agreement protocol based on the Weil pairing, Elec. Lett., vol. 38(13), pp. 630-632, 2002.
16. K. Shim, Efficient one round authenticated tripartite key agreement protocol from Weil pairing, Elec. Lett., vol. 39(8), pp. 653-654, 2003.
17. H. Sun and B. Hsieh, Security analysis of Shim's authenticated key agreement protocols from pairings, Cryptogarphy ePrint Archive, Report 2003/113, available at http://eprint.iacr.org/2003/113/.
18. G. Xie, An ID-based key agreement scheme from pairing, Cryptology ePrint Archive: Report 2005/093, available at http://eprint.iacr.org/2005/093, 2005.