Breaking a certificateless key agreement protocol withour bilinear pairing

Weiwei Han
School of Mathematics & Computer Science, Guangdong University of Business Studies, Guangzhou, China
Email: hww_2006@163.com

Abstract*:* Certificateless public key cryptography simplifies the complex certificate management in the traditional public key cryptography and resolves the key escrow problem in identity-based cryptography. Many certificateless designated verifier signature protocols using bilinear pairings have been proposed. But the relative computation cost of the pairing is approximately twenty times higher than that of the scalar multiplication over elliptic curve group. Recently, He et al. proposed a certificateless authenticated key agreement protocol without pairings and presented that their protocol is secure in the random oracle model. In this paper, we show that their protocol is insecure against the Type I adversary.

Key words*: Certificateless cryptography; Authenticated key agreement; Provable security; Bilinear pairings; Elliptic curve*

1. Introduction

Public key cryptography is an important technique to realize network and information security. Traditional public key infrastructure requires a trusted certification authority to issue a certificate binding the identity and the public key of an entity. Hence, the problem of certificate management arises. To solve the problem, Shamir defined a new public key paradigm called identity-based public key cryptography [1]. However, identity-based public key cryptography needs a trusted KGC to generate a private key for an entity according to his identity. So we are confronted with the key escrow problem. Fortunately, the two problems in traditional public key infrastructure and identity-based public key cryptography can be prohibited by introducing certificateless public key cryptography (CLPKC) [2], which can be conceived as an intermediate between traditional public key infrastructure and identity-based cryptography.

Following the pioneering work due to Al-Riyami and Paterson [2], several certificateless two-party authenticated key agreement(CTAKA) protocols [3-8] have been proposed. All the above CTAKA protocols may be practical, but they are from bilinear pairings and the pairing is regarded as the most expensive cryptography primitive. The relative computation cost of a pairing is approximately twenty times higher than that of the scalar multiplication over elliptic curve group [9]. Therefore, CTAKA protocols without bilinear pairings would be more appealing in terms of efficiency. Recently, He et al. [10] proposed a pairing-free CTAKA protocol based on Elliptic Curve Cryptography(ECC). He et al. also demonstrated their protocol is secure under the random oracle mode. Unfortunately, we will show their protocol is not secure against the Type I adversary.

The remainder of this paper is organized as follows. Section 2 describes some preliminaries. In Section 3, we review He et al.'s protocol and show that He et al.'s protocol is insecure against the Type I adversary in Section 4. In Section 5, we give a countermeasure to withstand the attack. Conclusions are given in Section 5.

2. Preliminaries

2.1 Background of elliptic curve group

Let the symbol E/F_p denote an elliptic curve E over a prime finite field F_p, defined by an equation

$$y^2 = x^3 + ax + b\ ,\quad a,b \in F_p \tag{1}$$

and with the discriminant

$$\Delta = 4a^3 + 27b^2 \neq 0\ . \tag{2}$$

The points on E/F_p together with an extra point O called the point at infinity form a group

$$G = \{(x,y) : x,y \in F_p, E(x,y) = 0\} \bigcup \{O\}\ . \tag{3}$$

Let the order of G be n. G is a cyclic additive group under the point addition "+" defined as follows: Let $P,Q \in G$, l be the line containing P and Q (tangent line to E/F_p if $P = Q$), and R, the third point of intersection of l with E/F_p. Let l' be the line connecting R and O. Then P "+" Q is the point such that l' intersects E/F_p at R and O and P "+" Q. Scalar multiplication over E/F_p can be computed as follows:

$$tP = P + P + \cdots + P(t\ times) \qquad (4).$$

The following problems defined over G are assumed to be intractable within polynomial time.

Computational Diffie-Hellman (CDH) problem: Given a generator P of G and (aP, bP) for unknown $a, b \in_R Z_n^*$, compute abP. The CDH assumption states that the probability of any polynomial-time algorithm to solve the CDH problem is negligible.

2.2 CTAKA protocol

A CTAKA protocol consists of six polynomial-time algorithms[2, 8]: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key* and *Key-Agreement*. These algorithms are defined as follows.

Setup: **This algorithm takes** security parameter k as input and returns the system parameters $params$ and master key.

Partial-Private-Key-Extract: This algorithm takes $params$, master key and a user's identity ID_i as inputs and returns a partial private key D_i.

Set-Secret-Value: This algorithm takes $params$ and a user's identity ID_i as inputs, and generates a secret value x_i.

Set-Private-Key: This algorithm takes $params$, a user's partial private key D_i and his secret value x_i as inputs, and outputs the full private key S_i.

Set-Public-Key: This algorithm takes $params$ and a user's secret value x_i as inputs, and generates a public key P_i for the user.

Key-Agreement: This is a probabilistic polynomial-time interactive algorithm which involves two entities A and B. The inputs are the system parameters $params$ for both A and B, plus (S_A, ID_A, P_A) for A, and (S_B, ID_B, P_B) for B. Here, S_A, S_B are the respective private keys of A and B; ID_A is the identity of A and ID_B is the identity of B; P_A, P_B are the respective public key of A and B. Eventually, if the protocol does not fail, A and B will obtain a secret session key $K_{AB} = K_{BA} = K$.

3. Review of He et al.'s CTAKA protocol

He et al.'s CTAKA protocol consists of the following six algorithms: *Setup*, *Partial-Private-Key-Extract*, *Set-Secret-Value*, *Set-Private-Key*, *Set-Public-Key* and *Key-Agreement*.

Setup: This algorithm takes a security parameter k as inputs and returns system parameters and a master key. Given k, KGC does the following.

1) KGC chooses a k -bit prime p and determines the tuple $\{F_p, E/F_p, G, P\}$.
2) KGC chooses the master private key $x \in Z_n^*$ and computes the master public key $P_{pub} = sP$.
3) KGC chooses two cryptographic secure hash functions $H_1 : \{0,1\}^* \to Z_n^*$ and $H_2 : \{0,1\}^* \to Z_n^*$.
4) KGC publishes $params = \{F_p, E/F_p, G, P, P_{pub}, H_1, H_2\}$ as system parameters and secretly keeps the master key s.

Partial-Private-Key-Extract: This algorithm takes master key, a user's identifier, system parameters as input and returns the user's ID-based private key. With this algorithm, for each user with identifier ID_i, KGC works as follows.

1) KGC chooses at random $r_i \in Z_n^*$, computes $R_i = r_i \cdot P$ and $h_i = H_1(ID_i, R_i)$.
2) KGC computes $s_i = r_i + h_i s \bmod n$.

The user's s partial private key is the tuple $D_i = (s_i, R_i)$ and he can validate her private key by checking whether the equation $s_i \cdot P = R_i + h_i \cdot P_{pub}$ holds. The private key is valid if the equation holds and vice versa.

Set-Secret-Value: The user with identity ID_i picks randomly $x_i \in Z_n^*$ sets x_i as his secret value.

Set-Private-Key: The user with identity ID_i takes the pair $S_i = (x_i, D_i)$ as its private key, where $D_i = (s_i, R_i)$.

Set-Public-Key: The user with identity ID_i takes $params$ and its secret value x_i as inputs, and generates its public key $P_i = x_i \cdot P$.

Key-Agreement: Assume that an entity A with identity ID_A has private key $S_A = (x_A, D_A)$ and public key $P_A = x_A \cdot P$, an entity B with identity ID_B

has private key $S_B = (x_B, D_B)$ and public key $P_B = x_B \cdot P$. A and B run the protocol as follows.

1) A send $M_1 = (ID_A, R_A, P_A)$ to B.

2）After receiving M_1, B chooses at random the ephemeral key $b \in Z_n^*$ and computes $T_B = b \cdot (P_A + R_A + H_1(ID_A, R_A)P_{pub})$, then B send $M_2 = (ID_B, R_B, P_B, T_B)$ to A.

3) After receiving M_2, A chooses at random the ephemeral key $a \in Z_n^*$ and computes $T_A = a \cdot (P_B + R_B + H_1(ID_B, R_B)P_{pub})$, then A send $M_3 = (T_A)$ to B.

Then both A and B can compute the shared secrets as follows.

A computes

$$K_{AB}^1 = (x_A + s_A)^{-1} \cdot T_B + a \cdot P \quad \text{and} \quad K_{AB}^2 = a \cdot (x_A + s_A)^{-1} \cdot T_B \qquad (5)$$

B computes

$$K_{BA}^1 = (x_B + s_B)^{-1} \cdot T_A + b \cdot P \quad \text{and} \quad K_{AB}^2 = b \cdot (x_B + s_B)^{-1} \cdot T_A \qquad (6)$$

3. Attack

In CTAKA, as defined in [2], there are two types of adversaries with different capabilities, we assume *Type 1 Adversary*, A 1 acts as a dishonest user while *Type 2 Adversary*, A 2 acts as a malicious KGC:

Type 1 Adversary: Adversary A 1 does not have access to the master key, but A 1 can replace the public keys of any entity with a value of his choice, since there is no certificate involved in CLPKC.

Type 2 Adversary: Adversary A 2 has access to the master key, but cannot replace any user's public key.

In this section, we will show that a Type I adversary A 1 is able to impersonate any user to finish the key agreement with any other user. Assume A 1 want to impersonate a user A with the private key $S_A = (x_A, D_A)$ and the public key P_A to finish the key agreement with a user B with the private key $S_B = (x_B, D_B)$ and the public key P_B. A 1 can get the goal through the followings steps.

1) A 1 generates a random number $r'_A, r''_A \in Z_n^*$ and computes $R'_A = r'_A \cdot P$

and $h'_A = H_1(ID'_A, R'_A)$.

2) A 1 replace A's public key P_A with $P'_A = -h'_A \cdot P_{pub}$.
3) A 1 send $M_1 = (ID_A, R'_A, P'_A)$ to B.
4) After receiving M_1, B chooses at random the ephemeral key $b \in Z_n^*$ and computes $T_B = b \cdot (P'_A + R'_A + H_1(ID_A, R'_A)P_{pub})$, then B send $M_2 = (ID_B, R_B, P_B, T_B)$ to A 1.
5) After receiving M_2, A 1 chooses at random the ephemeral key $a \in Z_n^*$ and computes $T'_A = a \cdot (P_B + R_B + H_1(ID_B, R_B)P_{pub})$, then A 1 send $M_3 = (T_A)$ to B.

Then both A 1 and B can compute the shared secrets as follows.

A 1 computes

$$K_{AB}^1 = (r'_A)^{-1} \cdot T_B + a \cdot P \quad \text{and} \quad K_{AB}^2 = a \cdot (r'_A)^{-1} \cdot T_B \tag{7}$$

B computes

$$K_{BA}^1 = (x_B + s_B)^{-1} \cdot T_A + b \cdot P \quad \text{and} \quad K_{AB}^2 = b \cdot (x_B + s_B)^{-1} \cdot T_A \tag{8}$$

Since

$$\begin{aligned}(r'_A)^{-1} \cdot T_B &= (r'_A)^{-1} \cdot b \cdot (P'_A + R'_A + H_1(ID_A, R'_A)P_{pub}) \\ &= (r'_A)^{-1} \cdot b \cdot (-H_1(ID_A, R'_A)P_{pub} + R'_A + H_1(ID_A, R'_A)P_{pub}) \\ &= (r'_A)^{-1} \cdot b \cdot R'_A = (r'_A)^{-1} \cdot b \cdot r'_A \cdot P = b \cdot P\end{aligned} \tag{9}$$

then we can get that

$$K_{AB}^1 = (r'_A)^{-1} \cdot T_B + a \cdot P = b \cdot P + a \cdot P \tag{10}$$

$$K_{BA}^1 = (x_B + s_B)^{-1} \cdot T_A + b \cdot P = a \cdot P + b \cdot P \tag{11}$$

$$K_{AB}^2 = a \cdot (r'_A)^{-1} \cdot T_B = abP \tag{12}$$

and

$$K_{BA}^2 = b \cdot (x_B + s_B)^{-1} \cdot T_A = baP \tag{13}$$

Thus the agreed session key for A 1 and B can be computed as:

$$\begin{aligned}sk &= H_2(ID_A \,||\, ID_B \,||\, T_A \,||\, T_B \,||\, K_{AB}^1 \,||\, K_{AB}^2) \\ &= H_2(ID_A \,||\, ID_B \,||\, T_A \,||\, T_B \,||\, K_{BA}^1 \,||\, K_{BA}^2)\end{aligned} \tag{14}$$

4. Coutermeasure

In the review of traditional public key cryptography, the user i's public key is $P_i + R_i + H_1(ID_i, R_i)P_{pub}$ in He et al.'s protocol. However, P_i almost has nothing relation with $R_i + H_1(ID_i, R_i)P_{pub}$. Then the type I adversary can remove the role of the KGC's public key P_{pub} through replacing P_i with $-H_1(ID_i, R_i)P_{pub}$. Then He et al.'s protocol is not secure. We can overcome the weakness through revising the **Partial-Private-Key-Extract** algorithm.

The user carries out the **Set-Secret-Value** algorithm and **Set-Public-Key** the algorithm first according to the description in Section 3.1 and gives his public key P_i to KGC. Then KGC generate the partial secret key for the user through the following **Partial-Private-Key-Extract** algorithm.

Partial-Private-Key-Extract: This algorithm takes master key, a user's identifier, a user's public key P_i, system parameters as input and returns the user's ID-based private key. With this algorithm, for each user with identifier ID_i, KGC works as follows.

1) KGC chooses at random $r_i \in Z_n^*$, computes $R_i = r_i \cdot P$ and $h_i = H_1(ID_i, R_i, P_i)$.
2) KGC computes $s_i = r_i + h_i s \bmod n$.

The user also changes the computation of h_i when generating T_i in the **Key-Agreement** algorithm. He et al.'s protocol can withstand the attack described in the above section, since $h_i = H_1(ID_i, R_i, P_i)$ will changes according to the change of the public key P_i.

5. Conclusion

The certificateless public key cryptography is receiving significant attention because it is a new paradigm that simplifies the public key cryptography. Recently, He et al. proposed a CTAKA protocol without pairings. In this paper, we showed that the CTAKA protocol is insecure against a Type I adversary who has no access to the master key but is allowed to replace public keys of users. We then proposed a countermeasure to overcome the weakness.

References

[1]. A. Shamir, Identity-based cryptosystems and signature protocols, Proc. CRYPTO1984, LNCS, vol.196, 1984, pp.47–53.

[2]. S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, Proceedings of ASIACRYPT 2003, LNCS 2894, Springer-Verlag, 2003, pp. 452－473.

[3]. S. Wang, Z. Cao, X. Dong, Certificateless authenticated key agreement based on the MTI/CO protocol, Journal of Information and Computational Science 3 (2006) 575 – 581.

[4]. Y. Shi, J. Li, Two-party authenticated key agreement in certificateless public key cryptography, Wuhan University Journal of Natural Sciences 12 (1) (2007) 71 – 74.

[5]. M. Luo, Y. Wen, H. Zhao, An enhanced authentication and key agreement mechanism for SIP using certificateless public-key cryptography, in:Proceedings of the IEEE ICYCS 2008, IEEE, 2008, pp. 1577 – 1582.

[6]. T. Mandt, C. Tan, Certificateless authenticated two-party key agreement protocols, in: Proceedings of the ASIAN 2006, LNCS, vol. 4435, Springer-Verlag, 2008, pp. 37 – 44.

[7]. F. Wang, Y. Zhang, A new provably secure authentication and key agreement mechanism for SIP using certificateless public-key cryptography, Computer Communications 31 (10) (2008) 2142 – 2149.

[8]. L. Zhang, F. Zhang, Q. Wua, J. Domingo-Ferrer, Simulatable certificateless two-party authenticated key agreement protocol, Information Sciences 180 (2010) 1020 – 1030.

[9]. L. Chen, Z. Cheng, and N.P. Smart, Identity-based key agreement protocols from pairings, Int. J. Inf. Secur., 6(2007) pp.213–241,.

[10]. D. He, J. Chen, J. Hu, A pairing-free certificateless authenticated key agreement protocol, Internal Journal of Communication System, DOI: 10.1002/dac.1265.