Security Flaws in Zhang and Xu Improved Concurrent Signature Scheme

Abstract. Nguyen first introduces the unlinkability property for concurrent signature schemes to protect signers' privacy. Huang and Wang proposed a fair concurrent signature scheme based on identity recently. However, Zhang and Xu show that Huang and Wang scheme does not satisfy the unforgeability property, because Huang and Wang scheme is vulnerable to forgery. To overcome the forgery flaws, they proposed an improved scheme. Unfortunately, a new cheating attack is proposed to show that Zhang and Xu scheme is still unfair. Zhang and Xu scheme does not provide unlinkability property, so the privacy protection is weak.

Keywords: Unlinkabiliy, fairness, concurrent signatures, fair exchange.

I. INTRODUCTION

Chen et al. [2] introduces concurrent signatures as a new solution for fair signature exchange problem between two signers. The fairness of the concurrent signature scheme is based on the idea of ambiguous signatures that maybe generated only by two possible signers. In a concurrent signature scheme, an initial signer sends the matching signer an ambiguous signature which is bound by a secret value, keystone. After verifying the ambiguous signature from the initial signer, the matching signer sends the initial signer his/her ambiguous signature. After verifying the matcher signer's ambiguous signature, the initial signer can transform the matching signer's ambiguous signature into the signature binding to the matching signer by releasing the keystone. After the release of the keystone, the matching signer can transform the initial signer's ambiguous signature into the signature binding to the initial signer. Finally, both the signers fairly exchange their signatures.

Nguyen [3] stressed the importance of unlinkability of concurrent signature for the privacy protection in real applications. By unlinkability, no one can find out the link between the two exchanged signatures in the concurrent signature scheme. For example, the concurrent signature scheme is used in the electronic transaction on Internet. The customer and the merchant have to fairly exchange the signature of the customer's payment instruction and the signature of the merchant's agreement on customers' order. However, the customer does not hope anyone finds out the link between the payment and the order agreement, due to the privacy protection. Therefore, the unlinkability is important in the real applications.

Huang and Wang [4] proposed a fair ID-based concurrent signature scheme. Unfortunately, Zhang and Xu[5] show that Huang and Wang scheme does not satisfy the unforgeability property. To fix the forgery problem of Huang and Wang scheme, they proposed an improved scheme. However, a new cheating attack is proposed to show that Zhang and Xu scheme does not satisfy the fairness and the unlinkability properties.

The security requirements of a concurrent signature scheme are stated in the next section.

Then Zhang and Xu scheme will be reviewed in Section III. The new cheating attack and some comments on Zhang and Xu scheme are presented in Section IV. Finally, Section V is a brief conclusion.

II. SECURITY REQUIREMENTS

A secure concurrent signature scheme should satisfy the following security requirements:

Correctness: If a signature σ is generated correctly by **Asign** algorithm on a message m, **Averify** algorithm returns "accept" with a non-negligible probability. Given a signature σ on m and a security parameter l, after the keystone k is released, the output of **Verify** algorithm is "accept" with a non-negligible probability

Unforgeability: Except the cooperation of the initial signer and the matching signer, no one can generate a valid concurrent signature with a non-negligible probability.

Ambiguity: Before the keystone *k* is released, no verifier is able to identify the actual signer with the probability that is greater than 1/2.

Fairness: Both signatures of the initial signer and the matching signer can be bound with their signers' identities simultaneously after the keystone *k* is released.

Unlinkability: The relationship between two exchanged concurrent signatures cannot be found, as long as the keystone is released

III. REVIEW OF ZHANG AND XU SCHEME

The bilinear pairing and the underlying security assumption are stated before the review of Zhang and Xu scheme.

A. Bilinear Pairings and Security Assumptions

Let G_1 be a cyclic additive group generated by the element P with order q, and G_2 be a cyclic multiplicative group with the same order q, where q is a prime number. A bilinear pairing is a map e: $G_1 \times G_1 \rightarrow G_2$ satisfying three properties:

Bilinear property: For any elements P, Q, $R \in G_1$, $e(P+Q, R)=e(P, R)e(Q, R)$ and $e(P, Q+R)= e(P, Q)e(P, R)$.

Non-degenerated property: There exist P and $Q \in G_1$ such that $e(P, Q) \neq 1$, where 1 is the identity of the group G_2.

Computable property: A polynomial-time algorithm exists to compute e(P, Q) for two elements P and $Q \in G_1$.

The security of Zhang and Xu scheme is based on the Computational Co-Diffie-Hellman (Co-CDH for short) assumption. The Co-CDH problem and assumption are defined below.

Co-CDH Problem: Given a randomly chosen (P_1, P_2, aP_1, bP_2), compute $abP_2 \in G_2$, where P_1 and $P_2 \in G_1$, and $a, b \in Z_q^*$ are two unknown integers.

Co-CDH Assumption: For every probabilistic polynomial-time algorithm A, the probability of A solving Co-CDH-Problem is negligible.

B. Zhang and Xu Scheme

Zhang and Xu's scheme [1] consists of the five algorithms and one protocol. The five algorithms are **Setup**, **Key generation**, **Asign**, **Averify,** and **Verify** algorithms. In the following, **Setup**, **Key generation**, **Asign**, and **Averify** algorithms are describe first. Then the protocol is stated. Finally the **Verify** algorithm is stated.

Setup Algorithm

The system parameters and functions are generated by this algorithm. The private key generator (PKG for short) randomly chooses its master private key $s \in Z_q^*$, and sets its public key $P_{pub}= sP$. Then PKG publishes two cryptographic hash functions $H_1:\{0, 1\}^* \rightarrow G_1$ and $H_2:\{0, 1\}^* \rightarrow$

Z_q^*. Then the system parameters are $\{G_1, G_2, e, q, P, P_{pub}, H_1, H_2\}$. All the message space M, keystone space K and keystone fix space F are Z_q^*.

Key Generation Algorithm

The signer U_i submits PKG his/her identity ID_i, then PKG sets U_i's public key $P_i = H_1(ID_i)$ and computes the signer's private key $s_i = sP_i$.

Asign Algorithm

By using this algorithm, the user generates the ambiguous signature on some message. Suppose that Signer U_i generates his/her ambiguous signature on the message m_i for U_j. On the given input (P_i, P_j, s_i, f, m_i), U_i randomly chooses $\alpha \in Z_q^*$, sets $C_1 = f$, and computes $h = H_2(m_i||P_i||P_j||C_1)$, $C_2 = \alpha P_i - C_1 - hP_j$, and $V = (h+\alpha)s_i$. The output ambiguous signature is $\sigma_i = (C_1, C_2, V)$.

Averify Algorithm

To validate the ambiguous signature $\sigma_i = (C_1, C_2, V)$ generated by U_i for U_j, the input of this algorithm is (m_i, σ_i, P_i, P_j, P_{pub}). On this input, check whether or not $e(P, V) = e(P_{pub}, C_1 + C_2 + hP_i + hP_j)$. If the equation holds, output "accept"; otherwise, output "reject".

After the description of those four algorithms, the protocol for the exchange of concurrent signatures is stated below. Without losing generality, suppose that U_A is the initial signer and U_B is the matching signer.

Concurrent Signature Protocol

Step 1: The initial signer U_A chooses a random number $\alpha \in Z_q^*$, and computes $\beta_1 = e(P, P_{pub})^{\alpha}$ and $\beta_2 = H_2(e(P_{pub}, P_B)^{\alpha})$. U_A also picks a random keystone $k \in Z_q^*$, and computes $c = m_A \oplus \beta_2$, $f = H_2(k||\beta_1||c)$, and $S = \alpha P_{pub} - fs_A$, where m_A is the exchanging message of U_A. Then U_A generates his/her ambiguous signature σ_A = **Asign**(P_A, P_B, s_A, f, m_A) = (C_1, C_2, V). Finally send (σ_A, c, S) to B.

Step 2: U_B computes $\beta_1 = e(P, S)e(P_{pub}, P_A)^{C_1}$, $\beta_2 = H_2(e(S, P_B)e(P_A, s_B)^{C_1})$, and $m_A = c \oplus \beta_2$. Then verify the ambiguous signature σ_A by **Averify**(m_A, σ_A, P_A, P_B, P_{pub}). If **Averify**(m_A, σ_A, P_A, P_B, P_{pub}) = "reject", abort.

Step 3: U_B chooses a random number $t \in Z_q^*$, and computes $\beta_3 = e(P, P_{pub})^t$ and $\beta_4 = H_2(e(P_{pub}, P_A)^t)$. On the exchanged message m_B, U_B computes $c' = m_B \oplus \beta_4$, $f = H_2(C_1||\beta_3||c')$, and $S' = tP_{pub} - fs_B$. Then generate the ambiguous signature σ_B = **Asign**(P_B, P_A, s_B, f, m_B) = (C_1', C_2', V'). Finally send (σ_B, c', S') to U_A.

Step 4: U_A computes $\beta_3 = e(P, S')e(P_{pub}, P_B)^{C_1'}$, $\beta_4 = H_2(e(S', P_A)e(P_B, s_A)^{C_1'})$ and $m_B = c' \oplus \beta_4$. Then verify the ambiguous σ_B by performing **Averify**(m_B, σ_B, P_B, P_A, P_{pub}). If the result is "reject", abort. Check whether or not $C_1' = H_2(C_1||\beta_3||c')$. If $C_1' \neq H_2(C_1||\beta_3||c')$, then A aborts; otherwise, A releases the keystone k to B and both signatures are binding concurrently.

Finally, (m_A, σ_A, c, S, m_B, σ_B, c', S', k) become the concurrent signature of two parties.

The concurrent signature verification algorithm is stated below.

Concurrent Signature Verification

On the concurrent signature (m_A, σ_A, c, S, m_B, σ_B, c', S', k), compute $\beta_1 = e(P, S)e(P_{pub}, P_A)^{C_1}$, and check whether or not $C_1 = H_2(k||\beta_1||c)$ and **Averify**(m_A, σ_A, P_A, P_B, P_{pub}) = "accept". Return invalid if neither $C_1 = H_2(k||\beta_1||c)$ nor **Averify**(m_A, σ_A, P_A, P_B, P_{pub}) = "accept" holds. Then compute $\beta_3 = e(P, S')e(P_{pub}, P_B)^{C_1'}$, and check whether or not

$C_1'= H_2(C_1||\beta_3||c')$ and **Averify**(m_B, σ_B, P_A, P_B, P_{pub})= "accept". Return invalid if any equation does not hold; otherwise, return valid to means that **(m_A, σ_A, c, S, m_B, σ_B, c', S', k)** is a valid concurrent signature of A and B on m_A and m_B.

IV. CRYPTANALYSIS OF ZHANG AND XU SCHEME

The new cheating attack on Zhang and Xu scheme is first proposed. By our cheating attack, the initial signer obtains the concurrent signature on some message without the agreement of the matching signers. The matching signer cannot obtain the concurrent signature after the concurrent signature protocol.

Cheating attack

Our cheating attack on Zhang and Xu scheme is described below.

Step 1: Initial signer U_A chooses a random number $\alpha \in Z_q^*$, and computes $\beta_1= e(P, P_{pub})^{\alpha}$ and $\beta_2= H_2(e(P_{pub},P_B)^{\alpha})$. U_A picks a random keystone $k \in Z_q^*$, and computes $c= m_A \oplus \beta_2$, $f= H_2(k||\beta_1||c)$, and $S= \alpha P_{pub}-fs_A$, where m_A is the exchanging message of U_A. Then U_A generates his/her ambiguous signature σ_A= **Asign**(P_A, P_B, s_A, f, m_A)= (C_1, C_2, V). To cheating the matching signer, U_A computes his/her cheating **ambiguous** signature σ_A''= **Asign**(P_A, P_B, s_A, f , m_A'')= (C_1, C_2'', V''), where m_A'' is a cheating message. Choose a random S'' and compute $\beta_2''= H_2(e(S'', P_B)e(s_A, P_B)^{C_1})$ and $c''= m_A'' \oplus \beta_2''$. Finally send ($\sigma_A''$, c'', S'') to U_B.

Step 2: U_B computes $\beta_1''= e(P, S'')e(P_{pub}, P_A)^{C_1}$, $\beta_2''= H_2(e(S'', P_B)e(P_A, s_B)^{C_1})$ and $m_A''= c'' \oplus \beta_2''$. Then verify the ambiguous signature σ_A'' by **Averify**(m_A'', σ_A'', P_A, P_B, P_{pub}). If **Averify**(m_A'', σ_A'', P_A, P_B, P_{pub})= "reject", abort.

Step 3: U_B chooses a random number $t \in Z_q^*$, and computes $\beta_3= e(P, P_{pub})^t$ and $\beta_4= H_2(e(P_{pub}, P_A)^t)$. On the exchanged message m_B, U_B computes $c'= m_B \oplus \beta_4$, $f= H_2(C_1||\beta_3||c')$, and $S'=tP_{pub}-fs_B$. Then generate the ambiguous signature σ_B= **Asign**(P_B, P_A , s_B , f, m_B)= (C_1', C_2', V'). Finally send (σ_B , c', S') to U_A.

Step 4: U_A computes $\beta_3= e(P,S')e(P_{pub}, P_B)^{C_1'}$, $\beta_4=H_2(e(S', P_A)e(P_B, s_A)^{C_1'})$ and $m_B=c' \oplus \beta_4$. Then verify the ambiguous σ_B by performing **Averify**(m_B, σ_B, P_B, P_A, P_{pub}) If the result is "reject", abort. Check whether or not $C_1'=H_2(C_1||\beta_3||c')$. If $C_1' \neq H_2(C_1||\beta_3||c')$, then A aborts; otherwise, A obtains the valid concurrent signature (σ_A , c , S , σ_B , c' , S' , k) on the message m_A and m_B without the agreement of U_B by using the keystone k.

On the other hand, the concurrent signature (σ_A'', c'', S'', σ_B , c' , S' , k) is illegal for $C_1 \neq H_2(k||\beta_1''||c'')$. That is the matching signer U_B does not obtain the concurrent signature he/she wants. Notice that **Averify**(m''_A, σ''_A, P_A, P_B, P_{pub}) must be accept for the cheating ambiguous signature σ_A''= **Asign**(P_A, P_B, s_A, f, m_A''). Thus the initial signer U_A obtains the concurrent signature he/she wants. Therefore our cheating attack is successful.

Moreover, Zhang and Xu scheme does not satisfy unlinkability property that is an important security requirement in real applications of concurrent signatures.

V. CONCLUSIONS

A cheating attack is proposed to show that Huang et al. concurrent signature scheme is not fair.

Huang et al. scheme does not provide unlinkability but it is important properties for the concurrent signature scheme [3].

REFERENCES

[2] L. Chen, C. Kudla, and K. Paterson, "Concurrent Signature," *Advances in Cryptology- EUROCRYPT 2004*, LNCS 3027, New York: Springer-Verlag, 2004. pp. 287-305.

[3] K. Nguyen, "Asymmetric Concurrent Signatures," *Proceedings of Information and Communications Security Conference, ICICS 2004*, LNCS 3783, New York: Springer-Verlag, 2005, pp. 181-193.

[4] X. F. Huang and L. C. Wang. "A Fair Concurrent Signature Scheme Based on Identity," *Proceedings of the 2nd International Conference on High-performance Computing and Applications (HPCA'09)*, Aug 10-12, 2009, Shanghai, China. LNCS 5938. Berlin: Springer-Verlag, 2010, pp. 198-205.

[5] Z. Zhang and S. Xu. "Cryptanalysis and Improvement of a Concurrent Signature Scheme Based on Identity," *2011 IEEE 2nd International Conference on Proceedings of the Software Engineering and Service Science (ICSESS)*, Beijing, July 15-17, 2011, pp. 453-456.