A New Fair E-Payment Protocol Based on Certificateless Concurrent Signature

Zhang Minqing[1], Xiao Haiyan[1], Yang Xiaoyuan[1], Li Hua[2]

1. Key Laboratory of Network and Information Security under the Chinese Armed Police Force, Electronic Department, Engineering Institute of the Armed Police, Xi'an, China, 710086

2. School of software, university of electronic science and technology of china, Chengdu, China, 610000

spritexiao@163.com

Abstract: Several protocols for fair exchange have been proposed in recent years，but some of them cannot achieve the fairness in payment. In these traditional ways, seller gets the payment after having sent e-good or buyer gets the e-good after having paid for it, so that is unfair between seller and buyer. And this paper proposed a new fair e-payment protocol based on provably secure certificateless concurrent signature scheme, it guarantees the fairness of the transaction between buyer and seller. At the end of the transaction, e-good and e-check are valid concurrently. It is more convenient and secure because it doesn't need involvement of TTP, so this protocol can be applied to the trade of digital products in network environment.

Keywords: electronic commerce; e-payment protocol; certificateless concurrent signature; fair exchange

1. Introduction

Due to the rapid growth of electronic commerce nowadays, a related security issue on the fair exchange of electronic data between two parties over computer networks becomes more and more important. The purpose of a fair exchange protocol is to barter data between two entities, as a result of which either both parties get what they want or they both get nothing [1]. We can find various exchange instances in different types of commercial activity [2]:

–In contract signing, two parties exchange their non-repudiable commitment to the contract text.

–In purchasing, a payment is exchanged for a valuable item.

–In certified mail, a message is exchanged for an acknowledgement of receipt.

Fair exchange protocol can be examined in three categories:

–Gradual exchange protocols: where two parties gradually disclose the expected items by many steps. These protocols have some theoretical value but seem to be too cumbersome for actual implementation because of the high communication overhead [3].

–Third party protocols: which make use of an on-line or off-line (trusted) third party. In these protocols, it is desirable to minimize the TTP's involvement when designing efficient fair exchange protocols in order to avoid the bottleneck problem.[3]

–Fair exchange protocol based on concurrent signature or other digital signatures with additional properties. These protocols turn out to be increasingly important recently. They don't need TTP's involvement, and schemes are not complicate as protocols mentioned above.

Before 2004, most scholars used the second method to structure fair exchange protocols[1,2,3,7], but this approach has some drawbacks. It is very difficult to find TTP which can be trusted totally on internet. And if we use semi-trusted TTP, the protocol maybe very complicate, and caused the bottleneck problem. So how to minimize the TTP's involvement is very efficient to design a fair exchange protocol. In Eurocrypt'04, Chen et al. introduced the notion of concurrent signatures [4], which provides an alternative approach to solving such problem. In their concurrent signature scheme, two parties A and B interact without the help of a third party to sign messages M_A and M_B in such a way that both signatures are ambiguous until an extra piece of information (called keystone) is released by one of A and B , i.e., from a third party's viewpoint, the two signatures may be generated by either of parties before the keystone is released. Upon releasing the keystone, both signatures become binding to their true signers concurrently [5]. So concurrent signature become increasingly important recently in fair exchange protocol, it can realize the concurrentness of the exchange compare to those traditional protocols and without the involvement of TTP. Another side, certificateless public key cryptography removes the necessary of certificate to ensure the authentication of the user's public key in traditional certificate-based public key cryptography and also overcomes the inherent key escrow problem in identity-based public key cryptography [6]. It made the cryptography has higher efficiency. These two approaches are used in fair exchange protocols frequently.

In this paper, a new e-payment protocol for e-goods is presented. The proposed protocol provides a method for fair exchange of e-check for e-goods, and solves the problem of unfairness in traditional way that seller gets the payment after sent e-good or buyer gets the e-good after pay for it. This protocol introduced a provably secure certificateless concurrent signature scheme to realize fair payment on internet, guarantees the fairness and non-repudiation of the whole transaction .And it can be applied to the trade of digital products in network environment.

2. Provably Secu re C ertificateless Concurren t Signature Scheme

The provably secure certificateless concurrent signature scheme was proposed by Zhenjie Huang, Xuanzhi Lin and Rufen Huang in 2008. And the scheme is as follows [6].

Setup:

–Choose a additive cyclic group G_1 with a prime order q and a multiplicative cyclic group G_2 with the same order, respectively. Let P denotes a generator in G_1. Let $e: G_1 \times G_1 \to G_2$ be a bilinear pairing.

–The Key Generation Center (KGC) selects the system master key $s \in_R Z_q^*$ and sets $P_0 = sP$.

–Selects cryptographic Hash functions $H_1: \{0,1\}^* \to G_1$ and $H_{2:}: \{0.1\}^* \to Z_q$.

–Sets the initial-keystone-fix function $F_I: Z_q \to Z_q$ be a one-way permutation, and the matching keystone-fix function $F_M(x, y) = F_I(x) + y(\bmod q)$.

KeyGen: We assume here that the signer's identity is denoted by ID_i, then his public and private keys are generated through the following steps:

–Partial Private Key Extract: Computes $Q_i = H_1(ID_i \| P_0)$ and transports the partial private key $D_i = sQ_i$ to the signer ID_i over a confidential and authentic channel.

–Private Key Generate:

1) Checks whether $e(D_i, P) = e(Q_i, P_0)$ holds. If not, returns to the Partial Private Key Extract phase.

2) Chooses a secret value $x_i \in_R Z_q^*$, sets $S_i = x_i D_i$ as his private key.

–Public Key Generate: Sets $P_i = (X_i, Y_i) = (x, P, x_i P_0)$ as his public key.

Sign: Accepts the input $(ID_i, ID_j, P_i, P_j, S_i, f_i, m_i)$, the algorithm performs the following.

–If $e(X_j, P_0) = e(Y_j, P)$, selects $r_i \in_R Z_q^*$ and computes $R_i = e(P,P)^{r_i} e(Q_j, Y_j)^{f_i}$, $v_i = H_2(m_i \| R_i \| H_2(ID_i \| Y_i) \oplus H_2(ID_j \| Y_j)) - f_i$, $U_i = r_i P - v_i S_i$.

The signature is $\sigma_i = (U_i, v_i)$.

Verify: Accepts the input $((U_i, v_i), ID_i, ID_j, P_i, P_j, f_i, m_i)$, the algorithm performs the following.

–Computes $R_i = e(U_i, P) e(Q_i, Y_i)^{v_i} e(Q_j, Y_j)^{f_i}$.

–Checks whether $v_i + f_i = H_2(m_i \| R_i \| H_2(ID_j \| Y_j))$, $e(X_i, P_0) = e(Y_i, P)$, and $e(X_j, P_0) = e(Y_j, P)$ are held. If they are, accepts; Otherwise, rejects.

Sign-Protocol

1) The initial signer performs the following.

–Picks a random keystone $k_I \in_R Z_q$, and computes the keystone fix $f_I = F_I(k_I)$.

–Picks a message m_I and computes her ambiguous signature $\sigma_I = ASign(ID_I, ID_M, P_I, P_M, S_I, f_I, m_I)$

–Sends σ_I, m_I and f_I to the matching signer.

2) The matching signer performs the following.

–Verifies σ_I by checking whether $AVerify(\sigma_I, ID_I, ID_M, P_I, P_M, f_I, m_I) = accept$,

If not, he aborts.

–Picks $k \in_R Z_q$, computes the keystone $k_M = H_2(e(Q_I, Y_I)^k)$ and the keystone fix $f_M = F_I(k_M) + f_I(\bmod q)$.

–Picks a message m_M and computes his ambiguous signature $\sigma_M = ASign(ID_M, ID_I, P_M, P_I, S_M, f_M, m_M)$,

–Computes the encrypted matching-keystone $K'_M = kP$.

–Sends σ_M, m_M, and K'_M back to the initial signer.

3) The initial signer performs the following.

–Computes $k_M = H_2(e(K'_m, S_I))$ and $f_M = F_I(k_M) + f_I(\bmod q)$.

–Verifies the signature σ_M by checking whether $AVerify(\sigma_M, ID_M, ID_I, P_M, P_I, f_M, m_M) = accept$

If not, aborts. Otherwise, releases the keystone pair (k_I, k_M).

Verify: The algorithm accepts $(k_i, k_j, \sigma_i, \sigma_j, ID_i, ID_j, P_i, P_j, m_i, m_j)$, where $\sigma_i = (U_i, v_i)$, $\sigma_j = (U_j, v_j)$, computes $f_i = F_I(k_i)$, $f_j = F_I(k_j) + f_i (\mathrm{mod}\, q)$, $R_i = e(U_i, P) e(Q_i, Y_i)^{v_i} e(Q_j, Y_j)^{f_i}$, $R_j = e(U_j, P) e(Q_j, Y_j)^{v_j} e(Q_i, Y_i)^{f_j}$, then checks whether $v_i + f_i = H_2(m_i \| R_i \| H_2(ID_i \| Y_i) \oplus H_2(ID_j \| Y_j))$, $v_j + f_j = H_2(m_j \| R_j \| H_2(ID_j \| Y_j) \oplus H_2(ID_i \| Y_i))$, $e(X_i, P_0) = e(Y_i, P)$, $e(X_j, P_0) = e(Y_j, P)$. If all equations are held, it outputs accept. Otherwise, it returns reject.

3. A New Fair E-Payment Protocol

Here we present a new fair e-payment protocol based on provably secure certificateless concurrent signature scheme. In this protocol, when seller receives the list which including e-good that buyer needs, he creats the first keystone fix and encrypts the e-good with this keystone, then sends his ambiguous signature and encrypted e-good while the seller responds to his ambiguous signature by creating another ambiguous signature with a matching keystone fix and sends her ambiguous signature and a e-check which is not valid yet. Each party can verify the correctness and validity of the transaction information and if both of them are honest and behave correctly, at the last, when the keystone pair released by seller, both signatures become binding to their respective signers concurrently, so the buyer gets the e-good and the seller gets the valid e-check.

The notations below are used in the description of our protocol.

$Alice$: buyer.

Bob: seller.

P_A: the private key of $Alice$.

P_B: the private key of Bob.

$\{\}_k$: encryption of message with key k.

$List$: The list which including the $e-good$ that buyer need.

$A \to B: M$: principal A dispatches message M addressed to principal B.

$F_B: Z_q \to Z_q$ initial-keystone-fix function, it is a one-way permutation, and the matching keystone-fix function $F_A(x, y) = F_B(x) + y (\mathrm{mod}\, q)$.

m: the description of $e-good$, including introduction of the digital product and the acceptance of service.

The whole protocol is as follows.

Buyer $Alice$ wants to buy a digital product $e-good$ from seller Bob, she must do as follows.

Step1. $Alice$ writes the $List$ which including the e-good she need.

$Alice \to Bob: \{ID_A, ID_B, List\}_{P_A}$

Step2. Bob chooses a radon keystone $k_B \in_R Z_q$, and computes the keystone fix $f_B = F_B(k_B)$, his ambiguous signature is $\sigma_B = ASign(ID_B, ID_A, P_B, P_A, S_B, f_B, \{e-good\}_{k_B})$,

$Bob \to Alice$: $\{ID_B, ID_A, m, f_B, \sigma_B, List, \{e-good\}_{k_B}\}_{P_B}$

Step3. $Alice$ checks whether $AVerify(\sigma_B, ID_B, ID_A, P_B, P_A, f_B, \{e-good\}_{k_B}) = accept$, If not, she aborts. Otherwise, she picks $k \in_R Z_q$, computes the keystone $k_A = H_2(e(Q_B, Y_B)^k)$ and the keystone fix $f_A = F_B(k_A) + f_B (\mathrm{mod}\, q)$. Then $Alice$ signs a check and computes her ambiguous signature $\sigma_A = ASign(ID_A, ID_B, P_A, P_B, S_A, f_A, e-check)$, $K'_A = kP$.

$Alice \to Bob: \{ID_A, ID_B, \sigma_A, K'_A, e-check\}_{P_A}$

Step4. Bob computes $k_A = H_2(e(K'_A, S_B))$ and $f_A = F_B(k_A) + f_B (\mathrm{mod}\, q)$. And verifies the signature σ_A by checking whether $AVerify(\sigma_A, ID_A, ID_B, P_A, P_B, f_A, e-check) = accept$, If not, aborts. Otherwise, releases the keystone pair (k_B, k_A).

While the keystone pair (k_B, k_A) released, $Alice$ decrypts $\{M\}_{k_B}$ with k_B and gets the digital product $e-good$. At the same time, $Alice's$ signature becomes binding to her $e-check$ and valid. Bob can get payment from bank by offering the $e-check$.

While Bob gets $e-check$ which is valid, he need to offering

$$(k_A, k_B, \sigma_A, \sigma_B, ID_A, ID_B, P_A, P_B, \{e-good\}_{k_B}, e-check)$$

to bank in order to get the payment.

Bank verifies as follows:

where $\sigma_A = (U_A, v_A)$, $\sigma_B = (U_B, v_B)$, computes

$f_A = F_B(k_A)$,

$f_B = F_B(k_B) + f_A (\bmod q)$,

$R_A = e(U_A, P)e(Q_A, Y_A)^{v_A} e(Q_B, Y_B)^{f_A}$,

$R_B = e(U_B, P)e(Q_B, Y_B)^{v_B} e(Q_A, Y_A)^{f_B}$,

then checks whether

$$v_A + f_A = H_2(e-check \,\|\, R_A \,\| \, H_2(ID_A \,\|\, Y_A) \oplus H_2(ID_B \,\|\, Y_B)),$$

$$v_B + f_B = H_2(\{e-good\}_{k_B} \,\|\, R_B \,\| \, H_2(ID_B \,\|\, Y_B) \oplus H_2(ID_A \,\|\, Y_A)),$$

$e(X_A, P_0) = e(Y_A, P)$,

$e(X_B, P_0) = e(Y_B, P)$.

If all equations are held, it outputs accept. Otherwise, it returns reject.

4. Protocol Analysis

The requirements for fair exchange were formulated in [2]

–Effectiveness. If two parties behave correctly, they will receive the expected items without any involvement from any arbitrator.

–Fairness. After completion of a protocol run, either each party receives the expected item or neither party receives any useful information about the other's item.

–Timeliness. At any time during a protocol run, each party can unilaterally choose to terminate the protocol without losing fairness.

–Non-repudiation. If an item has been sent from party O to party R, O cannot deny origin of the item and R cannot deny receipt of the item.

–Verifiability. If one party misbehaves, resulting in the loss of fairness for the other party, the victim can verifies correctness and validity of the transaction information.

We analyse our protocol with respect to the requirements listed above.

Claim 1. If the communication channel between O and R is resilient, the protocol satisfies the effectiveness requirement.

The correctness and unforgeability of the provably secure certificateless concurrent signature scheme have proved in [4]. And the correctness of our protocol is based on those proofs, here we needn't prove it again.

In this protocol, if $Alice$ and Bob behave correctly, they will receive the expected items without any involvement of other parties. $Alice$ gets the digital product $e-good$, and Bob gets the valid $e-check$.

First, Bob receives $List$ from $Alice$, and he encrypts the digital product $e-good$ which according to her request, then sends his ambiguous signature $\sigma_B = ASign(ID_B, ID_A, P_B, P_A, S_B, f_B, \{e-good\}_{k_B})$ and other information to $Alice$. $Alice$ verifies σ_B, if it is correct, she sends her ambiguous signature $\sigma_A = ASign(ID_A, ID_B, P_A, P_B, S_A, f_A, e-check)$ to Bob. If Bob verifies σ_A is correct, then releases the keystone pair (k_B, k_A).

At this time, $Alice$ decrypts $\{e-good\}_{k_B}$ with keystone k_B, and gets the digital product $e-good$ that she wants. And Bob gets the e-check and $Alice's$ valid signature which guarantee that Bob can get payment from bank. At the last, each party receives the expected item. So the protocol satisfies the effectiveness requirement.

Claim 2. The protocol satisfies the fairness requirement.

As we mentioned above, if both $Alice$ and Bob are honest, they will send their messages according to the protocol description, and at last, they will receive the expected items without any involvement of any arbitrator. But if anyone is dishonest in the exchange, neither party receives any useful information about the other's item, as for each side, the exchange is fair.

Proof: We first consider the possible unfair ituations that $Alice$ may face.

–If the $e-good$ which $Alice$ received is not what she needs or it doesn't consistent with the description that Bob offered. In this case, When $Alice$ received the encrypted $e-good$ from Bob in step2, she also received the description of $e-good$, including introduction of the digital product and the acceptance of service which signed by Bob. So if Bob is dishonest, $Alice$ will appeal to arbitrators of juristic department by offering these proofs which Bob cannot deny.

978-1-935068-01-3 © 2009 SciRes.

–If Bob doesn't release the keystone pair (k_B, k_A) after he gets $e-check$ which signed by $Alice$, $Alice$ cannot decrypt and get $e-good$. In this case, when Bob wants to get payment from bank, he must offer $(k_A, k_B, \sigma_A, \sigma_B, ID_A, ID_B, P_A, P_B, \{e-good\}_{k_B}, e-check)$ to bank, and bank verifies these information, if it is correct, Bob can get his payment, and at the same time, $Alice$ can offer $\{ID_B, ID_A, m, f_I, \sigma, List\}_{P_B}$ to bank and get the keystone pair (k_B, k_A) if she passed the verification.

And the possible unfair situation that Bob may face is just he sends the encrypted $e-good$ to $Alice$ but not receives $e-check$. In this case, Bob just deny to release the keystone pair (k_B, k_A), so $Alice$ cannot decrypt to get $e-good$. Neither party receives useful item.

Claim 3. The protocol satisfies the timeliness requirement.

–Step1. $Alice$ can simply quit the transaction without losing fairness after she sent $\{ID_A, ID_B, List\}_{P_A}$.

–Step2. Bob can quit the transaction without losing fairness after he sent $\{ID_B, ID_A, m, f_I, \sigma, List\}_{P_B}$ to $Alice$, because $Alice$ just gets the encrypted $e-good$, and the keystone pair (k_B, k_A) is still secret.

Claim 4 . The protocol satisfies the non-repudiation requirement.

Proof: By the protocol description, information send in every step is non-repudiation because of the signatures by two parties.

– $Alice$ cannot deny that she gets $e-good$ in step2, otherwise, Bob cannot receives $e-check$ from $Alice$ without sent encrypted $e-good$.

– Bob cannot deny that he gets $e-check$ in step3, otherwise he cannot get the payment from bank, because he must offering

$$(k_A, k_B, \sigma_A, \sigma_B, ID_A, ID_B, P_A, P_B, \{e-good\}_{k_B}, e-check)$$

to bank to get the payment.

Claim 5. The protocol satisfies the verifiability requirement.

Proof: The verifiability of provably secure certificateless concurrent signature scheme guarantees the verifiability of this protocol.

– $Alice$ verifies the correctness and validity of the transaction information $\{ID_B, ID_A, m, f_B, \sigma_B, List, \{e-good\}_{k_B}\}_{P_B}$ by checking $AVerify(\sigma_B, ID_B, ID_A, P_B, P_A, f_B, \{e-good\}_{k_B}) \overset{?}{=} accept$

– Bob verifies the correctness and validity of the transaction information $\{ID_A, ID_B, \sigma_A, K'_A, e-check\}_{P_A}$ by checking $AVerify(\sigma_A, ID_A, ID_B, P_A, P_B, f_A, e-check) \overset{?}{=} accept$.

–Bank receives $(k_A, k_B, \sigma_A, \sigma_B, ID_A, ID_B, P_A, P_B, \{e-good\}_{k_B}, e-check)$ from Bob, he must verifies correctness and validity of the transaction information by checking whether

$$v_A + f_A = H_2(e-check \,\|\, R_A \,\| \, H_2(ID_A \| Y_A) \oplus H_2(ID_B \| Y_B)),$$

$$v_B + f_B = H_2(\{e-good\}_{k_B} \,\|\, R_B \,\| \, H_2(ID_B \| Y_B) \oplus H_2(ID_A \| Y_A)),$$

$$e(X_A, P_0) = e(Y_A, P),$$

$$e(X_B, P_0) = e(Y_B, P).$$

If equations above are held, the bank outputs accept and give the payment according to $e-check$ to Bob. Otherwise, he returns reject.

5. Conclusions

Fair exchange turns out to be an increasingly important topic due to the rapid growth of electronic commerce. And it is important to guarantee the fairness of the transaction between the buyer and seller. In this paper, we use concurrent signature to ensure the concur- rentness of buying and selling, at the end of the transaction, $e-good$ and $e-check$ are valid concurrently. Our protocol based on provably secure certificateless concurrent signature scheme, it makes the exchange more convenient and secure, the seller and buyer can exchange in a fair way. And this protocol can be applied to the trade of digital products in network environment.

References

[1] Cagil Can Oniz, Erkay Savas, Albert Levi. An Optimistic Fair E-Commerce Protocol for Large E-Goods. Proceedings of the Seventh IEEE International Symposium on Computer Networks,

ISCN'06, 2006, pp.214-219.

[2] N. Asokan, V. Shoup and M. Waidner. Asynchronous protocols for optimistic fair exchange. Proceedings of 1998 IEEE Symposium on Security and Privacy, pages 86-99, Oakland, California, May 1998, pp.46-52.

[3] Jianying Zhou, Robert Deng ,and Feng Bao. Some Remarks on a Fair Exchange Protocol. PKC2000, LNCS 1751, 2000, pp.46-57.

[4] Chen L, Kudla C, Paterson K G. Concurrent Signatures [C] / /LNCS3027: Advances in Cryptology - EUROCRYPT 2004. Berlin /Heidelberg: Springer, 2004, pp. 287- 305.

[5] Hongji Wang, Gang Yao, Xiaoxi Han.A New Concurrent Signature Scheme Based on the FAPKC3's Signature Scheme. State Key Laboratory of Computer Science of China under Grant No. SYSKF0801.IEEE, 2009.

[6] Zhenjie Huang, Xuanzhi Lin, Rufen Huang. Certificateless Concurrent Signature Scheme. The 9th International Conference for Young Computer Scientists, ICYCS, 2008, pp.2102-2107.

[7] Jorge L. Hernandez-Ardieta, Ana I. Gonzalez-Tablas, Benjamin Ramos Alvarez. An optimistic fair exchange protocol based on signature policies. Computers & Security 27(2008), 2008, pp. 309-322.