Two Fair Payment Protocols for E-Commerce Transaction

Wei Fan, Huaying Shu, Qiang Yan and Xin Liu

School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China
fanwei@buptinfo.com shuhy@bupt.edu.cn yq_10@sohu.com liuxin1919@gmail.com

Abstract. Fairness is an important property of secure electronic commerce. Most of E-payment protocols guarantee fairness by using a Trusted Third Party (TTP) or semi-trusted third party. There are few protocols without a trusted third party. Based on concurrent signature, we propose two protocols suitable for digital products transaction, one for mobile payment and the other for the transaction which has higher security requirements by customers. The most prominent character of our schemes is that there is no traditional TTP, so the network congestion can be avoided. Furthermore, the protocols satisfy fairness and some other characteristics.

Keywords: *E-commerce, E-payment protocol, Fairness, Non-repudiation, Security, TTP, Concurrent signatures*

1. INTRODUCTION

In practical E-commerce, the participants of a transaction usually do not trust each other. Under this condition, how to protect the merchant and customer's profits is the basis of the secure E-payment. Fair E-payment protocol is the key to solve this problem. That is, we should guarantee both of the two parties' fairness during the transaction.

Fairness is a fundamental problem in exchange protocols [1]. It has a lot of definitions by different researchers. J. Y. Zhou [2] defines that if each participant in a transaction has sufficient evidence to solve the possible dissension, or neither of them predominates during the transaction, the transaction is fair. In Asokan's paper, fairness is defined as each honest participant will not stay in a disadvantage position [3]. In other words a fair exchange protocol ensures that no player can gain an advantage over the other player by misbehaving, misrepresenting or by prematurely aborting the protocol [4]. In summary, fairness means that at the end of exchange, either each party receives the expected item or neither party receives any useful information about the other's item [1].

The existing E-payment protocols mainly use an on-line or off-line Trusted Third Party (TTP) [5-11] or semi-trusted third party [12] to assure the fairness of a transaction. But there are many disadvantages: firstly, the third party is apt to become the bottle-neck of a transaction and lead to network congestion; secondly, TTP

Please use the following format when citing this chapter:

Fan, W., Shu, H., Yan, Q., Liu, X., 2007, in IFIP International Federation for Information Processing, Volume 255, Research and Practical Issues of Enterprise Information Systems II Volume 2, eds. L. Xu, Tjoa A., Chaudhry S. (Boston: Springer), pp. 1037-1046.

decreases the protocol's working efficiency and increases the cost of a transaction; last but no least, how to search a third party to serve as the TTP and to make sure that we can find it at anytime is a difficult problem. Now there are mainly two approaches to solve the problem of bottle-neck caused by the third party: the first is to reduce the traffic passing through the third party, the second is to lower the requirements of protocols to the reliability of the third party. But there is rarely a protocol that does not use the third party completely. If we can design a protocol without TTP, all these problems can be solved.

In an electronic commerce transaction, it is essential to guarantee fairness as well as some other characteristics, such as non-repudiation for the actions and security for the exchanged items. In this paper, by using concurrent signatures [13], we present two protocols. Both of them possess characters of fairness, non-repudiation and security.

Concurrent signature algorithm is first introduced by Chen *et al.* in 2004. It exploits the ambiguity property enjoyed by ring signatures [14] and designated verifier signatures [15]. The most prominent characteristic of concurrent signatures is that the two signers create ambiguous signature to exchange first. Only after an extra piece of information is published (that is keystone), both the ambiguous signatures will bind with their true signers and become valid concurrently. That is, before keystone is published, neither of the signers can convince others the validity of the signatures. This algorithm eliminates the need for a third party thoroughly, and satisfies the property that either the two signatures become valid concurrently, or both of them invalid. Therefore, the fairness is achieved. In this paper, we exploit the excellent characteristics of concurrent signatures, and design a new two times concurrent signatures. Based on this, we present our two fair payment protocols for digital products transaction.

The rest of this paper is organized as follows. We make a basic introduction about concurrent signature first in section 2, then propose two schemes for digital products and analyze their fairness and other characteristics in section 3. Finally, the paper's conclusion and future work is given in section 4.

2. CONCURRENT SIGNATURES

Concurrent signature is a digital signature scheme comprised of four algorithms.

SETUP: It is a probabilistic algorithm that sets up the scheme parameters including keys. It selects two large primes p , q for $q \mid p-1$ and a generator $g \in Z_p^*$ of order q . It also generates two cryptographic hash functions: $H_1, H_2 : \{0,1\}^* \to Z_q$, a private key $x \in Z_q$ and the corresponding public key $y = g^x \bmod p$. And we suppose that Alice and Bob are the two parties in the signing process.

ASIGN: It is a probabilistic algorithm that takes as input (y_j, x_i, t, f). If Alice wants to sign message M_A, she will choose some random bits k (hold as a secret and we call it keystone), and this algorithm will output an ambiguous signature on M_A: $S_A = (w_A, h_A, f, y_A, y_B, M_A)$, where t is a random number and $t \in Z_q$, $f = H_2(k)$, $h = H_1\left(\left(g^t y_B^f \bmod p\right) \| M_A\right)$, $h_A \equiv (h - f) \bmod q$, $w_A \equiv (t - h_A x_A) \bmod q$.

AVERIFY: This algorithm takes as input $S_A = (w_A, h_A, f, y_A, y_B, M_A)$, and outputs *accept* or *reject*. In detail, that is after Bob receive S_A, he will compute the following equation.

$$h_A + f \equiv \left(H_1\left(\left(g^{w_A} y_A^{h_A} y_B^f \bmod p\right) \| M_A\right)\right) \bmod q \quad (1)$$

If equation 1 is true, the algorithm outputs *accept*, otherwise outputs *reject*. If true, Bob will use f to the compute $S_B = (w_B, h_B, f, y_B, y_A, M_B)$ and send S_B to Alice. Upon receiving S_B, Alice will also use the averifying algorithm to validate S_B. If the signature is correct, Alice will publish keystone k or send k to Bob.

VERIFY: This algorithm takes as input (k, S_i) and output the receiver of S_i and the signer of S_i. That is, after k is published, we can distinguish the receiver from signer. Otherwise, nobody can prove he or she is the signature's real receiver or signer in theory. That is, upon receiving k and $S = (w, a_1, a_2, y_B, y_A, M)$, by computing

$$a_1 + a_2 \equiv H_1\left(g^w y_A^{a_1} y_B^{a_2} (\bmod p) \| M\right)(\bmod q) \quad (2)$$

If the equation above is true, the validation passes, otherwise it fails. And if $a_1 = H_2(k)$, then Bob is the signer of the signature, Alice is the receiver of the signature. In the same way, if $a_2 = H_2(k)$, then Alice is signer and Bob is receiver.

From the introduction above, we can see that signers exchange ambiguous signatures first, and when the keystone is published, the ambiguous signatures come into effect. That is either the two signatures come into effect at the same time, or they are both invalid.

3. NEW SCHEMES FOR E-COMMERCE TRANSACTION

In this part, we propose two fair E-payment protocols without TTP, the scheme 1 and scheme 2, and make the protocol analysis of both schemes. The two protocols are

suitable for digital products transactions in different situation. The transaction flow of scheme 1 is sententious and this scheme can be used in mobile payment. Scheme 2 enhances security of the customers, and it's suitable for the transaction which has higher security requirements by the customers.

3.1 Scheme 1

Before the protocol starts, customer Alice opens an electronic check operation by GPRS or other means, and downloads the signature programme to the hand equipment. Then she can choose and buy products anytime. The protocol follows as below:

1. Alice chooses the digital product (such as a piece of digital image) she wants to buy after browses merchant Bob's website by wireless network, and then downloads an order form. After signing the order form with RSA signature, Alice sends it to Bob with an electronic check (e-check) C that is not signed and whose par value is equal to the digital products' price.
2. Bob first validates the identity and e-check's balance information after receiving the order form and check from Alice. If the par value of the check is not enough, the transaction fail and the system will send prompt information to Alice.

   After validating Alice's identity, Bob will choose a symmetrical key k (as a secret to hold), and encrypt the digital product with k to get cryptograph M. Then Bob uses Hash function to compute $f = Hash(k)$, and encrypts the document package made by cryptograph M, service guaranty and f symmetrically with session key k'. Bob uses Alice's public key y_A to encrypt session key k' to get a digital envelope, in the end Bob packs the encrypted document package and digital envelope together and sends it to Alice.
3. Upon Alice receive the document package from Bob, she first decrypts the digital envelope with her private key and gets the session key k', then uses k' to decrypt the encrypted document package to get cryptograph M, service guaranty and f. Lastly if Alice is satisfied with the service guaranty, she will use f to compute the concurrent signature S_A of e-check C, and send S_A to Bob.

$$S_A = (w_A, h_A, f, y_A, y_B, C) \tag{3}$$

where $f = Hash(k)$, $h = H_1\left(\left(g^t y_B^f \bmod p\right) \| C\right)$, $h_A \equiv (h - f) \bmod q$, $w_A \equiv (t - h_A x_A) \bmod q$, t is a random number. And if Alice is dissatisfied with the service guaranty, she can quit the transaction.

4. Upon receiving the check that is signed with concurrent signature, Bob will use the verifying algorithm of concurrent signatures to validate the check. That is validate equation 4 is whether true or not.

$$h_A + f \equiv \left(H_1\left(\left(g^{w_A} y_A^{h_A} y_B^{f} \bmod p\right) \| C\right)\right) \bmod q \quad (4)$$

And we must ensure $f = Hash(k)$, if not, the transaction terminates. If equation 4 is true, the validation is passed. Bob will send key k to Alice and transfer the electronic check to his account by using k in the bank. If equation 4 is false, the validation fails. Bob won't send k to Alice. After receiving k, the bank can verify the validity of e-check C. Only if Bob is a legal accepter of C, the bank will agree the transference.

5. Upon receiving k from Bob, Alice use k to decrypt cryptograph M and get the digital products she wants to buy.

3.2 Protocol Analysis of Scheme 1

The analysis of scheme 1 is as below.

1. Fairness: The protocol can guarantee fairness between customer and merchant during transaction.

   Alice can decrypt the encrypted product only when Bob publish keystone to take the e-check into effect. Neither of them will suffer from any party's quit from the transaction.

   As for Alice, if she dissatisfies with the products' quality or the product is not the same as what the merchant promises, she can charge against Bob with the service guaranty. As the e-check is signed with concurrent signatures, its signer and accepter is one and only, other people can not transfer the check to his or her account without the keystone.

   As for Bob, since Alice will send the e-check's information to Bob, it can decrease the vicious purchase situation, in which the par value is not enough for the transaction. And if Bob send the cryptograph M to Alice, but Alice deny that she does not receive it, or because of the network Alice does not receive it, Bob will also not suffer from it. Because only after keystone is published can Alice decrypt M to get the right digital products.
2. Non-repudiation: Alice can't deny she signed the check with k, as the concurrent signature's signer is one and only. After the transaction Alice can not deny she get the right digital products, because cryptograph M is signed by Bob, it can only be decrypted by the right keystone. Bob can't deny he take the check, as only when somebody offers the right key can he transfer the check to his account, and he is the only receiver of the e-check.
3. Security: If an attacker intercepts Alice's e-check at step 1, as the check is not signed, the attacker can not transfer the check. At other step, if the signed check is intercepted, as the keystone is not published to others except Alice, nobody can transfer it to his or her account without keystone. Since the digital envelope can be decrypted only by Alice, nobody else can imitate Alice to sign the check by f. Last but not least, there are just Alice and Bob taking part in the transaction,

so the conspiracy problem can be avoided.

4. The transaction flow of this scheme is sententious. During the transaction, there just two participants taking part in it. This scheme can be used in mobile payment.

3.3 Scheme 2

This scheme uses a new two times concurrent signatures, and the signature's detailed algorithm is included in the following protocol. Before the transaction, Alice should open an electronic check operation. The protocol follows as below:

1. This step is nearly the same as scheme 1. Alice chooses the product she wants to buy after browses merchant Bob's website, and then downloads an order form. After signing the order form with RSA signature, Alice sends it to Bob with an e-check C that is not signed and whose par value is equal to the digital products' price.
2. Upon receiving the signed order and electronic check from Alice, Bob validates the identity and e-check's balance information. If the par value of the check is not enough, the transaction fail and the system will send prompt information to Alice.

   After validating Alice's identity, Bob will choose a symmetrical key k_1 (as a secret to hold), and encrypt the digital products with k to get cryptograph M. Then Bob uses Hash function to compute $f_1 = Hash(k_1)$, and compute the service guaranty's concurrent signature S_B by f_1. Afterwards he encrypts the document package made by cryptograph M and S_B symmetrically with session key k' (made by Bob randomly). He uses Alice's public key y_A to encrypt session key k' to get a digital envelope. In the end Bob packs the package M' and digital envelope together and sends it to Alice.
3. Upon receiving the encrypted package from Bob, Alice will use her private key to decrypt the digital envelop to get session key k' firstly, then decrypt the encrypted package M' with k' to get cryptograph M and S_B. After that she will use the averifying algorithm of concurrent signatures to validate S_B is whether right. If S_B is right, and Alice is satisfied with the service guaranty, then she will choose another key k_2, computing $f_2 = Hash(k_2)$, and use f_2 to compute e-check C's concurrent signature $S_A{}'$. In the end, she will use f_1 to get concurrent signature S_A of $S_A{}'$, and send S_A to Bob, as the figure 1.

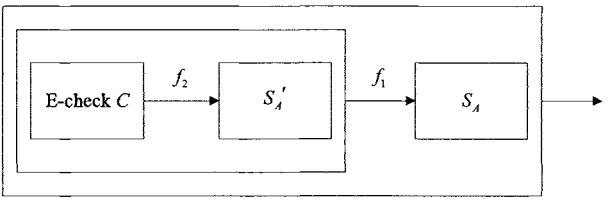

Figure 1. Sketch Map of the Two Times Concurrent Signatures

$$S_A' = \left(w_A', h_A', f_2, y_A, y_B, C\right) \tag{5}$$

$$S_A = \left(w_A, h_A, f_1, y_A, y_B, S_A'\right) \tag{6}$$

In the equations above,

$f_2 = Hash\left(k_2\right)$, $h' = H_1'\left(\left(g^{t'} y_B^{f_2} \bmod p'\right) \| C\right)$, $h_A' \equiv \left(h' - f_2\right) \bmod q'$,

$w_A' \equiv \left(t' - h_A' x_A\right) \bmod q'$;

$f_1 = Hash\left(k_1\right)$, $h = H_1\left(\left(g^t y_B^{f_1} \bmod p\right) \| S_A'\right)$, $h_A \equiv \left(h - f_1\right) \bmod q$,

$w_A \equiv \left(t - h_A x_A\right) \bmod q$; t and t' are random numbers.

And if Alice is dissatisfied with the service guaranty, she can choose to quit the transaction.

4. Upon receiving S_A from Alice, Bob use averifying algorithm of concurrent signatures to validate equation 7 is whether true or not. And we must ensure $f_1 = Hash\left(k_1\right)$, otherwise, the transaction fails.

$$h_A + f_1 \equiv \left(H_1\left(\left(g^{w_A} y_A^{h_A} y_B^{f_1} \bmod p\right) \| S_A'\right)\right) \bmod q \tag{7}$$

If equation 7 is true, the validation is passed and Bob send key k_1 to Alice. After receiving k_1, Alice can decrypt M by using k_1 and get the digital product. At the same time, she will publish k_2 and Bob can transfer the electronic check to his account by using k_2. If equation 7 is false, Bob won't send k_1 to Alice.

Upon receiving k_2, the bank can verify the validity of e-check C. Only if Bob is a legal accepter of C, the bank will agree the transference.

3.4 Protocol Analysis of Scheme 2

The analysis of scheme 2 is similar with scheme 1, and it is as below.

1. Fairness: The protocol also can guarantee fairness between customer and merchant during transaction.

   If Bob has received S_A but refuses publishing k_1, Alice won't suffer from it, as e-check is a valid one only when k_2 is published.

   If Alice has received digital products the same as the service guaranty but refused publishing k_2, then Bob can charge against her with the concurrent signature S_A and S_A'. Because although the e-check is not a valid one, but the arbitration organization can validate that Bob is the legal accepter of the e-check, and since Alice computes concurrent signature S_A of S_A', she must be responsible for the signature. Now Bob does not receive k_2, Alice must publish it.

   As for Alice, if she dissatisfies with the products' quality or the product is not the same as what the merchant promises, she can charge against Bob with the service guaranty. If Bob does not send the products at all, after decrypting cryptograph M Alice will find that. And she won't publish k_2, so Bob can not transfer the check without k_2.

2. Non-repudiation: After transaction, Alice can't deny she has received the digital product, as before she publishes k_2, she has got k_1 and decrypted M. Alice can't deny she has signed the check, as S_A is the concurrent signature of S_A'. She sends S_A to Bob, so the check can just be signed by her.

   Also Bob can't deny he has taken the check, as only when somebody offers the right key can he transfer the check to his account, and Bob is the only accepter of the check. If the check has been transferred, it must be done by Bob.

3. Security: If an attacker intercepts the signed check or cryptograph M during the transaction, there will be no loss for Alice and Bob. As the two keystones are not published to others and the check's accepter is one and only, nobody can transfer it to one's account or decrypt M without keystones. Since the digital envelope can be decrypted only by Alice, nobody else can imitate Alice to sign the check by f_1, that is the check's signature is one and only. There are just two parties in the transaction, so the conspiracy problem can be avoided. In conclusion, the scheme can guarantee transaction's security.

4. As before Alice publishes k_2, she can get the product first and check its quality. So this scheme increases requirements for merchants' credit and the quality of products, and it can protect customers' benefit well. Generally speaking, compared with merchants, customers are vulnerable groups in e-commerce. It is

better to solve the probable problems before transferring check than to solve them after the transaction has ended. In conclusion, this scheme enhances security of the customers, and it's suitable for the transaction which has higher security requirements by the customers.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed two protocols based on concurrent signatures. Scheme 1 guarantees that the two parties get the expected item they want to exchange concurrently. Scheme 2 protects customers' benefit well and enhances their security by using a new two times concurrent signatures. The most prominent characteristic of our protocols is that we get rid of the traditional TTP, and the protocols satisfy fairness and some other characteristics by using concurrent signature scheme. Both the benefits of customers and merchants can be protected, and the network congestion brought by a traditional TTP can also be avoided.

In our protocols, whether the keystone is published determines whether electronic checks become valid or electronic products are decrypted. Thus the party who holds keystone has the initiative in the transaction. The future work includes how to obtain absolute fairness as well as efficiency in order to apply it in a more resource-limited environment such as mobile communication.

ACKNOWLEDGEMENTS

Thanks for Yanping Wang and Huaping Li's contribution to this paper. And this work is supported by the National Natural Science Foundation of China (Grant No. 70671010).

REFERENCES

1. J. Zhou, R. Deng, and F. Bao, Some remarks on a fair exchange protocol, in *Proc. of the Third International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography Public Key Cryptography* (Springer-Verlag: Berlin, Heidelberg, 2000), pp.46-57.
2. J.Y. Zhou, *Non-Repudiation*. Ph.D Thesis, Department of Computer Science, Royal Holloway, University of London (1997).
3. N. Asokan, *Fairness in electronic ecommerce*. Ph.D Thesis, Department of Mathematics, University of Waterloo (1998).
4. I. Ray and I. Ray, Fair exchange in e-commerce, *ACM SIGecom Exchange*. Volume 3, Number 2, pp.9-17, (2002).
5. N. Asokan, M. Schunter, and M. Waidner, Optimistic protocols for fair exchange, in *Proc. of the fourh ACM Conference on Computer and Communications Security* (ACM Press, 1997), pp.6-17.

6. N. Asokan, V. Shoup, and M. Waidner, Asynchronous protocols for optimistic fair exchange, in *Proceedings of the IEEE Symposium on Research in Security and Privacy* (IEEE Computer Society Press, 1998), pp.86-99.
7. N. Asokan, V. Shoup, and M. Waidner. Optimistic Fair Exchange of Digital Signatures. *IEEE Journal on Selected Areas in Communication*. Volume 18, Number 4, pp.593-610, (2000).
8. J. Park, E. Chong, and H. Siegel, Constructing Fair-Exchange Protocols for E-Commerce via Distributed Computation of RSA Signatures, *PODC' 03* (ACM Press: New York, 2003), pp.172-181.
9. J. Zhou and D. Gollmann, An Efficient Non-repudiation Protocol, in *Proc. of 1997 IEEE Computer Security Foundations Workshop* (IEEE Computer Society Press, 1997), pp.126-132.
10. F. Bao, R. Deng, and W. Mao, Efficient and practical fair exchange protocols with off-line TTP, in *Proc. of IEEE Symposium on Security and Privacy* (1998), pp.77-85.
11. C. Boyd and E. Foo, Off-Line Fair Payment Protocols Using Convertible Signature, in *Proce. of Asiacrypt'98 (LNCS 1514)* (Springer-Verlag: Berlin, Heidelberg, 1998), pp.271-285.
12. M.K. Franklin and M.K. Reiter, Fair exchange with a semi-trusted third party, in *Proc. of the 4th ACM conference on Computer and Communications Security* (ACM Press, 1997), pp.1-5.
13. L. Chen, C. Kudla, and K.G. Paterson, Concurrent signature, in *Proc. of EUROCRYPT 2004, LNCS 3027* (Springer, 2004), pp.287-305.
14. R. Rivest, A. Shamir and Y. Tauman, How to leak a secret, in *Advances in Cryptology – ASIACRYPT 2001, LNCS 2248* (Springer-Verlag, 2001), pp.552-565.
15. M. Jakobsson, K. Sako, and R. Impagliazzo, Designated verifier proofs and their applications, in *Advances in Cryptology-EUROCRYPT 1996, LNCS Volume 1070* (Springer-Verlag, 1996), pp.143-154.