A Fair-Exchange Protocol
Based on Off-line Semi-Trusted Third Party

Wu Qingtao, zhang Hongyi, Pu Jiexin

Electronic Information Engineering College, Henan University of Science and Technology, Luoyang, China 471003

Summary
The fairness of data exchange becomes a key factor for the transaction over the Internet. To ensure the fairness of data exchange, the protocol needs a trusted third party be a judge when the dispute occurs, and has to give a guarantee to main parties during the data exchange. To eliminate the third party's connectional time and enhance its security, a fair-exchange protocol based on off-line semi-trusted third party has presented in this paper. The protocol employs an off-line semi-trusted third party, where a flexible method for selecting a semi-trusted third party is proposed. Also, the proposed protocol is extended to Web services so that it can be used for main parties in different platforms.

Key words:
Fair data Exchange, Off-line semi-trusted third party, Web Services

1. Introduction

Electronic commerce transactions, especially those that involve the exchange of digital products between the transacting parties, have additional requirements as compared to classical brick-and-mortar transactions. In the classical business environment, a transaction essentially involves fulfillment of some obligation by two parties; a contract describes the penalties if either party fails to meet its obligation. For example, a purchase of products involves the merchant delivering the product and, simultaneously, a customer paying for it. Since each transacting party has an identifiable place of doing business, if any party behaves unfairly in the transaction, that party can be physically approached and held accountable for its unfair behavior, according to the terms of the contract. In an electronic commerce environment, on the other hand, a party does not always have a physically identifiable place of doing business. After behaving unfairly in the electronic commerce transaction, a party can simply vanish without trace. In such a case, it may be next to impossible to enforce the penalties of the contract. Consequently, in an electronic commerce environment the two parties are reluctant to trust each other.

Owing to this lack of trust, electronic commerce protocols need to be carefully designed to prevent unfair business dealings by any player involved. Fairness is thus often a stronger requirement in secure electronic commerce protocols. Fairness is achieved in the transaction if at the end of it, either each player fulfills its obligation and receives the item it expects, or neither receives any portion of the other's item. A fair exchange protocol can then be defined as a protocol that ensures that no player in an electronic commerce transaction can gain an advantage over the other player by misbehaving, misrepresenting or by prematurely aborting the protocol.

Note that the problem of fair exchange is not just limited to information goods. We always assume that fairness is ensured in any business transaction. In an electronic commerce transaction where the product is not a piece of information, but rather something more tangible, we automatically have the same set of safeguards that ensure fair exchange in conventional transactions. However, if the product is a piece of information that is transmitted electronically over an inherently insecure medium such as the Internet, with the destination address possibly not bound to any physical address, fair exchange is more difficult to achieve. Thus, fair-exchange protocol for Web services has received the widest attention lately and the term is now mostly used to denote such protocols.

In this paper, a secure and efficient protocol for fair signature with off-line semi-trusted third party has presented. The protocol employs an off-line semi-trusted third party, where a flexible method for selecting a semi-trusted third party is proposed. Also, the proposed protocol is extended to Web services so that it can be used for main parties in different platforms.

2. Model and Problem Description

2.1 Model Description

We consider two mutually untrusting users, who have data items I_A and I_B respectively, which the other user cannot generate autonomously. User Ux, $X \in \{A, B\}$, advertises that I_X meets specification $\sum_X$ and offers to send Ix in return for receiving I_Y, where $Y \in \{A, B\}$ and $Y \neq X$. P_x denotes the process that executes an exchange protocol on behalf of user U_x on node N_x. In our model we assume that

Manuscript received July 5, 2007.
Manuscript revised July 27, 2007.

U_A and U_B are able to communicate through two secure channels (one from U_A to U_B, the other from U_B to U_A) providing confidentiality, integrity, authentication, and sequential, as shown in Fig.1.

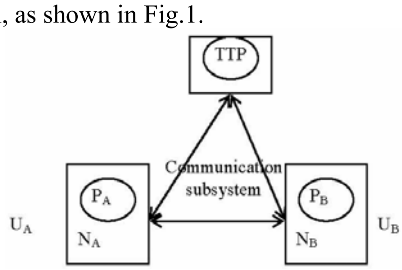

Fig. 1 System model for fair-exchange

Note that one important security property is missing in the channel: timeliness. Actually, some sent messages can never reach their final destination. Therefore, the only way for a malicious man-in-the-middle Nx to make the protocol fail is to stop transmitting messages in one direction or another by cutting the channel. Hence our adversarial model for N_x is a malicious algorithm which decides to cut one channel at some time or the two channels at the same or at different times. Due to the confidentiality property, the choice on when to cut channels cannot depend on the content of the messages, but only on the number of exchanged messages. Here is an example of a secure communication channel from UA to UB. Let *m* be the message to send and seq a sequence number which is incremented each time after a message is sent.

2.2 Problem Description

Several (different) definitions for the fair exchange are available in the literature. Most of them are context-dependent. For completeness we provide an informal one for our purpose.

Definition An exchange protocol between U_A and U_B is a protocol in which U_A and U_B own some items I_A and I_B respectively and aim at exchanging them. We say that the protocol is

(1). **Complete** if U_A gets I_B and U_B gets I_A at the end of the protocol when there is no malicious misbehavior;

(2). **Fair** if its terminates so that either U_A gets I_B and U_B gets I_A (success termination), or U_A gets no information about I_B and U_B gets no information about I_A (failure termination) even in case of misbehavior;

(3). **Timely** if U_A and U_B eventually end.

We say that the protocol is perfectly fair when it follows all these properties. When the protocol is not perfectly fair, we define two measures of unfairness.

– P_a (probability of unfair termination) is the maximum of the probability that the protocol ends on an unfair state over all possible misbehaviors.

– P_c (probability that crime pays) is the maximum of the conditional probability that the protocol ends on an unfair state conditioned on someone deviating from the protocol over all possible misbehaviors.

The fair exchange problem looks trivial when U_A and U_B are honest: they can just exchange their items one after the other and commit to discard them if the protocol fails. However, if timeliness is not guaranteed for the communication channel, *N* can just discard the last message and the protocol becomes insecure despite U_A and U_B being honest. We solve this here by using the synchronization protocol.

Generally, the problem of fair exchange is solved in a context where a dishonest user U_x totally controls the behavior of P_X to undermine every attempt to ensure fairness and non-repudiation. We give some notations for proposed protocol defined below:

(x_A, y_A)	a pair of U_A's asymmetric key
(x_B, y_B)	a pair of U_B's asymmetric key
$Sign_x$	a digital signature with a private key *x*
EKs	a cipher-text with a session key
g and *q*	public parameters
m	the contents of the contract
h	one-way hash function

3. The Proposed Protocol

There are three phases in the proposed protocol, including the semi-trusted third party selective phase, the normal phase, and the dispute phase.

3.1 Semi-Trusted Third Party Selection Phase

In accordance with system model and fair-exchange problem, a semi-trusted third party should be chosen randomly by U_A and/or U_B. All procedure displayed as Fig.2.

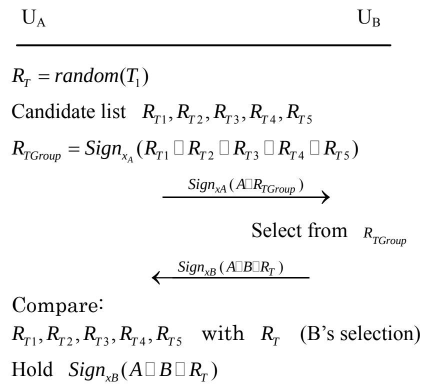

Fig. 2 Semi-trusted third party selective phase

Step 1. To make this phase more flexible, U_A is allowed to select five random third parties which U_A trusts by using an equation $R_T = random(Ti)$, where $i = \{0 < i \leq 100\}$.

Note that amount of computer which chosen be a random third party will be limited within 100 computers in the proposed protocol.

Step 2. A selects five random third parties, as possible random third parties, R_{T_i} $(i = 1, 2...5)$ as possible random third parties.

Note that these third parties are chosen at random, so their IDs are out of sequence.

Step 3. U_A adds a signature $Sign_{xA}(A \| RT_{Group})$ and sends it to U_B, where $RT_{Group} = Sign_{xA}(R_{T_1} \| R_{T_2} \| R_{T_3} \| R_{T_4} \| R_{T_5})$

Step 4. U_B selects a random party from the candidate lists RT_{Group}, U_B adds a signature $Sign_{xB}(A \| B \| R_T)$ and sends it to U_A

Step 5. U_A compares R_{T_i} $(i = 1, 2...5)$ with U_B's selection. This R_T becomes an off-line semi-trusted third which is mutually agreed by U_A and U_B this time. U_A and U_B hold $Sign_{xB}(A \| B \| R_T)$ until the dispute occurs.

Note that if U_B disagrees with the candidate list RT_{Group}, the protocol will be terminated.

3.2 The Normal Phase

In the proposed protocol, the normal phase consists of two sub-phases including signature verification and contract signing.

(1) Normal phase-signature verification

In the phase, the individual signatures will be generated and verified to complete the signature verification phase.

Step 1 U_A chooses $n_A \in Z_q^*$ at random, let $c_A = g^{n_A}$ mod q, and U_B chooses $n_B \in Z_q^*$ at random, let $c_B = g^{n_B}$ mod q;

Step 2a U_A calculates $d_A = n_A^{-1}(h(m) + x_A c_A)$ mod q, and the U_A's signature (c_A, d_A) is produced.

Step 2b U_B calculates $d_B = n_B^{-1}(h(m) + x_B c_B)$ mod q, and the U_B's signature (c_B, d_B) is produced.

Step 3 To verify the other side's signature, U_A sends $Sign_{xA}(c_A \| j_A)$ to U_B and U_B sends $Sign_{xB}(c_B \| j_B)$ to U_A.

Step 4 When U_A receives U_B's digital signature, U_A calculates $e_B = h(m)(j_B^{-1})'$ mod q and $I_B = c_B'(j_B^{-1})'$ mod q, where let c_B' and j_B' be the received versions of c_B and j_B respectively. And the U_A computes $a_2 = (g^{e_B} y_B^{I_B})$ mod q and compares a_2 with c_B. If a2= c_B, then U_B's signature is verified. U_A will tell U_B the result of verification: the former, the rest of steps will be continued; the latter, the protocol will be terminated.

Step 5 The former situation: when U_B receives the result from U_A, U_B calculates $e_A = h(m)(j_A^{-1})'$ mod q and $I_A = c_A'(j_A^{-1})'$ mod q, where let c_A' and j_A' be the received versions of c_A and j_A respectively. And then U_A computes $a_1 = (g^{e_A} y_A^{I_A})$ mod q and compares a1 with c_A. If $c_A = a_1$, then U_A's signature is verified; otherwise, U_A's signature may have been modified. U_B will tell U_A the result of verification: the former, the contract will start to be signed; the latter, the protocol will be terminated.

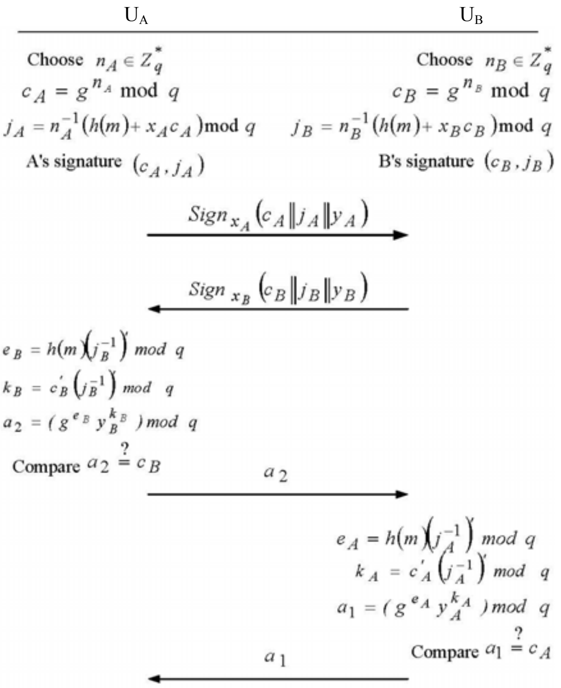

Fig. 3 Normal phase-signature verification

(2) Normal phase-contract signing

When the signature verification phase has completed, parties U_A and U_B prepare to sign the contract in the following phases.

Step 6 U_A encrypts the m with session key, and encrypts the message digest $h(m)$ with U_A's private key. Then U_A sends $EK_s^{(m)}$ and $Sign_{xA}(h(m))$ to U_B;

Step 7 When receiving $EK_s^{(m)}$, U_B reproduces a new message digest $h(m)'$ of the received message, and decrypts the $Sign_{xA}(h(m))$;

Step 8 Compared $h(m)'$ with $h(m)$, if two values are match, the contract is a valid. U_B encrypts the message digest $h(m)$ with U_B's private key, and then sends it to U_A in order to complete the normal thoroughly.

3.3 Dispute phase

In Steps 4-5 in the normal phase, the dispute may occur. Party U_B does not receive the result that party U_A compares $a2$ with c_A (Case I) or party U_A does not receive the result that party U_B compares $a1$ with c_B (Case II). At this time, party U_A or U_B will ask an off-line semi-trusted third party to solve the dispute.

It is assumed that the computer agrees and accepts to be an off-line semi-trusted third party in the proposed protocol when party U_A or U_B sends a request to it.

(1) Case I: dispute in party U_B

It is assumed that party U_B does not receive the result that party U_A compares a2 with c_B in Step 4 in the normal phase, so party U_B will ask the off-line semi-trusted third party to solve the dispute.

① U_B sends a request and $Sign_{xB}(A \| B \| R_T)$ to a computer on the network to tell the fact that it has been chosen as an off-line semi-trusted third party this time.

② The computer sends a response $Sign_{xOSTTP}(Sign_{xB}(A \| B \| R_T))$ to U_A and U_B simultaneously.

③ U_B sends $Sign_{xB}(e_B \| I_B \| y_B)$, c, $h(j)$, $h(m)$, y, g to the semi-trusted third party, where $h(j) = (h(j_A) + h(j_B))$ mod (q-1) and $c = c_A.c_B$ mod p.

④ The semi-trusted third party calculates $a_2 = (g^{e_B} y_B^{I_B})$ mod q and verifies the equation $y^{h(m)} = c^c.g^{h(j)}$ mod p.

⑤ If the equation holds, the semi-trusted third party sends a2 to U_A to continue the rest of steps; otherwise, the protocol will be terminated.

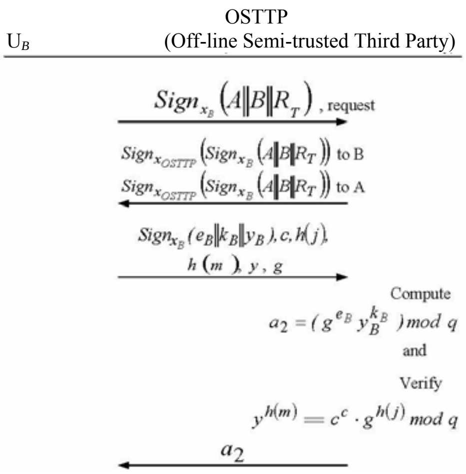

Fig. 4 Dispute phase (Case I)

Note that if the protocol is terminated, the off-line semi-trusted third party's responsibility will also be terminated. Parties U_A and U_B need to select a semi-trusted third party again if they want to re-start the protocol.

(2) Case II: dispute in party U_A

It is assumed that party U_A does not receive the result that party U_B compares a1with c_A in Step 5. In the normal phase, party U_A will ask the off-line semi-trusted third party to solve the dispute.

① U_A sends a request and $Sign_{xA}(A \| B \| R_T)$ to a computer on the network to tell the fact that it has been chosen as an off-line semi-trusted third party this time.

② The computer sends a response $Sign_{xOSTTP}(Sign_{xA}(A \| B \| R_T))$ to U_A and U_B simultaneously.

③ U_B sends $Sign_{xA}(e_A \| I_A \| y_A)$, c, $h(j)$, $h(m)$, y, g to the semi-trusted third party, where $h(j) = (h(j_A) + h(j_B))$ mod (q-1) and $c = c_A.c_B$ mod p.

④ The semi-trusted third party calculates $a_1 = (g^{e_A} y_A^{I_A})$ mod q and verifies the equation $y^{h(m)} = c^c.g^{h(j)}$ mod p.

⑤ If the equation holds, the semi-trusted third party sends a1 to U_B to continue the rest of steps; otherwise, the protocol will be terminated.

Note that if the protocol is terminated, the off-line semi-trusted third party's responsibility will also be terminated. Parties U_A and U_B need to select a semi-trusted third party again if they want to re-start the protocol.

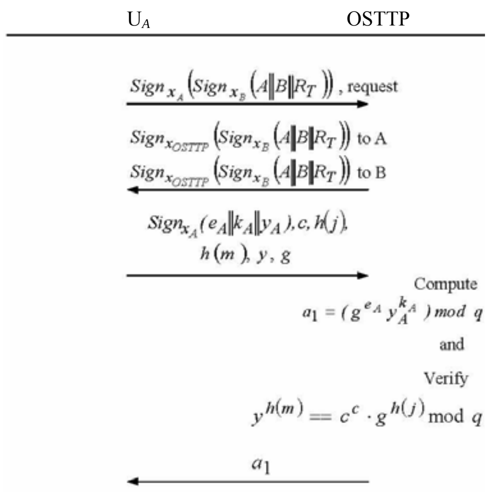

Fig. 5 Dispute phase (case II)

4. Conclusion

In this paper, we present a fair-exchange protocol based on off-line semi-trusted third party. There are three phases in the proposed protocol, including the semi-trusted third party selective phase, the normal phase, and the dispute phase. The proposed protocol has the following properties: 1) It allows exchange of digital goods among groups of participants, and Main parties obtain a flexible method for selecting a random third-party as an off-line semi-trusted third party actually. 2) It uses an off-line third party, 3) the third party is semi-trusted in the sense that the third party can not cause any unfairness to the main parties by misbehaving on its own. Also, this model is well suited to Electronic commerce transactions and may deserve future research.

Acknowledgment

This work is supported by Scientific Research Foundation for Doctoral Scholars, He'nan University of Science and Technology, China (No. 09001136)

References

[1] G. Avoine and S. Vaudenay. Fair Exchange with Guardian Angels. in *Information Security Applications*, LNCS 2908, pp. 188-202, 2004.
[2] G. Avoine, F. Gartner, R. Guerraoui and M. Vuolic. Gracefully Degrading Fair-Exchange with Security Modules, in the Proc. of the 5th European Dependable Computing Conference (EDCC2005), Springer Verlag, LNCS 3463, pp. 55-71.
[3] G. Bella. Inductive Verification of Smart Card Protocols. Journal of Computer Security, 11(1), 2003, pp. 87-132.
[4] R. Chadha, J.C. Mitchell, A. Scedrov, V. Shmatikov, Contract Signing, Optimism, and Advantage, 14th International Conference on Concurrency Theory (CONCUR), LNCS, 2761, 2003, pp. 366-382.
[5] P. D. Ezhilchelvan and S. K. Shrivastava, "Systematic Development of a Family of Fair Exchange Protocols", In the Proc. of the 17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security, 2003, (Eds. Sabrina De Capitani di Vimercati, Indrakshi Ray and Indrajit Ray), Kluwer Academic Press, ISBN 1 4020 8069 7, 2004, pp. 243-258.
[6] K. Hogg, P. Chilcott, M. Nolan, and B. Srinivasn. An Evaluation of Web Services in the Design of a B2B Application. in Proc. of the 27th Conference on Australasian Computer Science, pp. 331-340, 2004.

Wu Qingtao received the M.S. degrees in computer science from Henan University of Science and Technology, in 2003. He received the Ph.D. degrees in computer applications technology from East China University of Science and Technology in 2006. His research interest includes network & information security, software formal methods.

Zhang Hongyi received the B.S., M.S. degrees from Northeast Normal University, Changchun, China, in 1995 and 1998, respectively. Since 1998, she has been faculty of Henan University of Science & Technology, Luoyang, China, where she is currently a assistant professor of Electronic Information Engineering College. Since 2004, she has been to Xidian University for her Ph. D. degree. Her fields of interest are bioinformatics and signal processing.

Pu Jiexin received the B.S. degree in computer science & engineering from Jilin University of Technology, Changchun, China in 1982, M.S. degree from Nanjing University of Science and Technology, China in 1993. He has been a professor in Luoyang Institute of Technology since 1999. He is currently a professor and dean of Electronic Information Engineering College, Henan University of Science and Technology, China. His research interests include computer vision and pattern recognition, image processing, artificial intelligence and cognitive science.