Analysis of Offline Fair Exchange Protocols in Strand Spaces*

Xiangdong Li[1] , Qingpu Guo[2] and Qingxian Wang[1]
1. Department of Network Engineering, Information Engineering University
2. Computer Center, Henan University of Finance and Economics
Zhengzhou, Henan 450002, China
lixiangdong2510@126.com

Abstract

Formal and efficient protocol analysis by pen and paper is highly desired when informal and model-checking methods are not trusted in proving correctness of complicated e-commerce protocols. Based on strand spaces, this paper presents a method for fairness analysis of offline fair exchange protocols. With the new method, this paper formally analyzes an important offline fair exchange protocol-- the ASW Certified Email Protocol, detects two flaws, and makes an improvement. The security analysis shows that such protocols as complicated as fair exchanges can be formally analyzed by pen and paper.

1. Introduction

With the rapid development of e-commerce, the research on e-commerce protocols is drawing more and more attention. Fairness, the core property of e-commerce protocols, guarantees that participants in protocols do not obtain any advantages than honest parties. Fair exchange protocols are the protocols designed for protecting fairness in e-commerce. Offline fair exchange protocols, using offline-TTPs, are also called optimistic [1].

The security of fair exchange protocols, being the basis of the security of e-commerce, mainly refers to viability, fairness, non-repudiation, timeliness, abuse-freeness, accountability and other properties the protocols must have. Although there have been several formal methods for security analysis of fair exchange protocols, there are still many problems unsolved.

* Research was partially supported by the National High Technology Research and Development Program of China (No.2007AA01Z471), and the Key Technologies R&D Program of Henan Province of China (No.0624260017, 072102210029).

Based on BAN Logic [2], Kailar Logic [3,4] is presented for formal analysis of accountability of e-commerce protocols, but it has the similar disadvantages as the other BAN Logics, such as the complicated and inaccurate idealization process.

Kremer et al. proposed the Game Logic method [5,6] for analysis of fair exchange protocols, and got some positive results [7]. The problems of this method include the error-prone translation of protocols into the language used by the model-checker Mocha [8], and the state explosion problem.

Thayer Fabrega et al. proposed the theory of strand spaces [9]. The method is concise and straight, and analysis can be done by pen and paper when the strand space model is used for protocol analyzing. However, the strand space model is mainly used for analyzing secrecy and authentication [10,11]. Paper [12] proposes a method for non-repudiation analysis in strand spaces, which is also based on authentication. Paper [13] proposes a method to analyze fairness in the strand space model, but it is only for online-TTP protocols.

This paper presents a pen and paper method for fairness analysis in strand spaces. Using the method, an offline fair exchange protocol is formally analyzed. Two flaws are detected and an improvement is made.

2. Fairness analysis in strand spaces

In strand spaces, a strand is a sequence of events, which are the behaviors of a party or an intruder of a protocol. A protocol is depicted as a directed graph which includes a group of strands and edges. For more information about the strand space theory, please refer to paper [9,10]. Before we describe the new method, some basic definitions are given below.

Basic definitions

Definition 1 *Fairness-data* of a party at state *i*, denoted by f_i, is a set of terms that are influential to fairness, and attained by the party from exchanges up

978-0-7695-3508-1/08 $25.00 © 2008 IEEE
DOI 10.1109/CIS.2008.167

to the state *i*. Since a party's knowledge always increases, the fairness-data of the party at state *n* is $f_n=\bigcup_{i=1}^{n} f_i$, if the party has changed from state 1, 2 … to state *n*. That is, if a sequence of states are in the partial order of their strand $\prec_s$, their fairness-data is also in a partial order. We denote this partial order of fairness-data as $\prec_f$.

A node in strand space model is also regarded as a state in this paper.

There exists no partial order relationship $\prec_s$ between some states, but there may be partial order relationship $\prec_f$ between them. It is supposed in this paper that, for a given principal, there is a partial order $\prec_f$ between any two states.

Definition 2 Consider that party *A* has fairness-data f_p at state p, and party *B* has fairness-data f_q at state q. If f_p and f_q are equal under the partial order $\prec_f$, f_p and f_q are called *matching-data*; state p and q are called *matching-states*.

Definition 3 The states that a principal can reach in a protocol execution under given conditions are called its *conditional reachable states*, or simply, reachable states. If the fairness-data of all possible reachable states are comparable under the partial order $\prec_f$, the state at which the principal has the maximum fairness-data, is called the maximum reachable state.

Definition 4 If one state p is a reachable state under the condition that the other state q is not yet reached, these two states are called *mutually exclusive*.

Definition 5 An offline protocol is said to be *fair*, if wherever the main protocol stops, the maximum reachable states of all parties are matching-states.

Two-stage method for protocol analysis

This paper mainly considers the property of fairness of offline fair exchange protocols. And the discussion is limited to protocols with two parties, Alice and Bob, exchanging for digital targets from each other.

A typical offline fair exchange protocol consists of a main exchange, and one or more recovery sub-protocols. We divide protocol executions into two categories, and accordingly divide the analysis of a protocol into two stages.

In the first stage, principals are supposed to be able to receive successfully every message in the main protocol. That is, the protocol runs under the condition that all parties are honest, and all communication channels are operational. This is also the reason for the name "optimistic". We take a virtual protocol for an example. This virtual protocol, having a main exchange protocol and a Recovery-B sub-protocol, is as follows:

Main Protocol:	Recovery-B:
①$A\rightarrow B$: m_1 ②$B\rightarrow A$: m_2 ③$A\rightarrow B$: m_3 If B times out, then Recovery-B	①$B\rightarrow$TTP: m_1, m_2 ②TTP$\rightarrow B$: m_3'

Usually, the designer claims that message m_1,m_3 matches message m_2 when the protocol stops after step 3. However, we need to prove the matching-data relationship between data $\{m_1,m_3\}$ and data $\{m_2\}$ holds.

In this stage, it is important to list the matching-data relations of the main protocol. For this example, data $\{m_1,m_3\}$ matches data $\{m_2\}$. Proving rigorously the matching-data relations usually involves the analysis of whether the cryptographic algorithms and mechanisms are properly used or not, which is based on the theory of computational complexity. In this paper, we just list the matching-data relations of the protocol, and assume that all cryptographic algorithms and mechanisms in the protocol are perfect.

In the second stage, principals are supposed not to be able to successfully receive messages in the main protocol. That is, the protocol runs under the condition that not all parties are honest, and not all communication channels are reliable. Take the virtual fair exchange protocol for example. If *B* is not able to receive message m_3 as expected after sending m_2 to *A*, the fairness for *B* is in danger. To prove the fairness of the protocol is to prove that *B* can, in other ways (e.g. through TTP), reach a state and get message m_3, or m_3' which is matching-data of $\{m_2\}$.

In this stage, the sub-protocols have to be examined. It is important to list matching-states and prove that they are matching.

This paper proposes a procedure to examine the matching relations in strand spaces. The procedure is as follows:

① Illustrate the protocol, including its main protocol and sub-protocols, in strand space with a directed graph. Check and record the mutually exclusive states according to the protocol.

② Check whether the maximum reachable states of both principals match or not, after every message transmission of the main protocol.

Claim If the maximum reachable states of both principals match after every message transmission of the main protocol, the protocol is fair.

Proof: Since the principals can behave honestly or not at all, and the communication channels are not reliable, the main protocol may stop after any message transmission of the main protocol.

When the main protocol stops, and an honest principal, say Bob, is not able to receive his next message in the main protocol, he will initiate any sub-protocol possible to guard his fairness. The other principal acts similarly so that both parties reach their conditional maximum states.

If the maximum reachable states of both principals match after every message transmission of the main protocol, according to Definition 5**,** the protocol is fair. □

3. Analysis of the ASW protocol

We apply the two-stage method to the analysis of the Asokan-Shoup-Waidner certified e-mail protocol (ASW protocol) [14]. The ASW protocol uses a main protocol, an abort protocol and two recovery protocols, to fairly exchange an e-mail, non-repudiation of origin evidence (NRO), and non-repudiation of receipt evidence (NRR).

First stage analysis of the ASW protocol

The main protocol of ASW consists of four message transmissions (since N_A is not necessary, it is omitted in the following description):

① $A \rightarrow B$: $m_1 = A,B,\mathrm{TTP},c,\mathrm{h}(m),S_A(A,B,\mathrm{TTP},c,\mathrm{h}(m))$

② $B \rightarrow A$: $m_2 = \mathrm{h}(N_B), S_B(m_1, \mathrm{h}(N_B))$

if A times out then Abort-A

③ $A \rightarrow B$: $m_3 = m$

if B times out then Recovery-B

④ $B \rightarrow A$: $m_4 = N_B$

if A times out then Recovery-A

In the messages, N_A (omitted) and N_B are nonces generated by A and B, respectively. $c = E_{TTP}(m,A,B)$, is for TTP to decrypt m in recovery. The intentions of two parties are:

A: sends out e-mail m, NRO, and receives NRR.

B: receives e-mail m, NRO, and sends out NRR.

Therefore, the maximum fairness-data of the two parties is:

$\{m, \mathrm{h}(m), S_A(\mathrm{h}(m))\}$ and $\{\mathrm{h}(N_B),S_B(\mathrm{h}(N_B)), N_B\}$.

In literature, when a security protocol is designed, the following assumptions are usually made:

① random numbers exist;

② one-way hash functions exist;

③ digital signatures are un-forgeable.

Under these assumptions, Bob can take $\{m, \mathrm{h}(m), S_A(\mathrm{h}(m))\}$ as NRO evidence, and Alice can take $\{\mathrm{h}(N_B),S_B(\mathrm{h}(N_B)), N_B\}$ as NRR evidence.

When the main protocol finishes in the end, both parties get the data they expect. Therefore, at the first stage, fairness is satisfied.

Second stage analysis of the ASW protocol

For the detailed description of sub-protocol Abort-A, Recovery-A, and Recovery-B, please refer to paper [14]. It is assumed that sub-protocols interacting with TTP are atomic. i.e., if they start, they will finish successfully.

According to the procedure given in section 2.2, we have done the following analysis:

① Illustrate the protocol into a directed graph, in Figure1. Sub-protocols are set beside the main protocol in parallel. And there is a double-dotted horizontal line linking a sub-protocol with the main protocol, which indicates that the states of the two nodes linked together are actually identical. And the double-dotted horizontal lines are regarded as ⇒ edges when studying reachable states. We label every node in the main protocol and every sub-protocol in the way in paper [13].

It is found that the state aa4 (the 4th state in Abort-A) and the state rb4 (the 4th state in Recovery-B) are mutually exclusive.

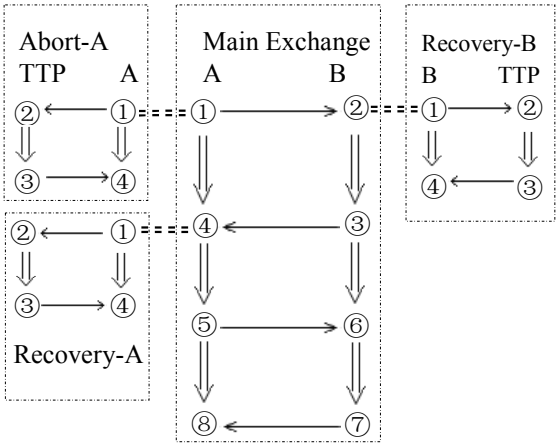

Figure1. ASW main exchange and sub-protocols

② For each message transmission of the main protocol, check and compare the maximum reachable states of both principals. In this step, we find two protocol weaknesses:

Weakness 1 After the first message m_1, Alice is at state main1, and Bob at state main2. The maximum reachable state for Alice is aa4, while for Bob it is rb4.

Their fairness-data does not match, because we have $f_{aa4} \prec_f f_{rb4}$. Considering their mutually exclusive relation, the possible attack is that before Alice reaches aa4, Bob gets to rb4 as soon as he can. If this happens, fairness for Alice is damaged.

Weakness 2 After the second message m_2, Alice is at state main4, and Bob at state main3. The maximum reachable state for Alice is ra4, and for Bob it is rb4. Although ra4 and rb4 are matching-states, the condition for Bob to reach rb4 is that Alice has not reached aa4. The possible attack is for Alice to start Abort-A first, and then start Recovery-A, keeping Bob from reaching rb4. If this happens, fairness for Bob is damaged. These weaknesses were first found in [15].

4. Improvement of the ASW protocol

The reason for the existing weaknesses is the unsuccessful arrangement of the mutually exclusive relations among the three sub-protocols in ASW.

The main idea of the improvement is for TTP to guarantee the three sub-protocols mutually exclusive. That is, no matter who starts whatever sub-protocol, TTP always delivers the result of that sub-protocol to the other principal, terminates the protocol and does not respond to any other request any more.

The improved sub-protocols are as follows.

Abort-A sub-protocol:

① $A \rightarrow$ TTP: a_1

a_1 = Aborted,m_1, S_A (Aborted,m_1)

② if TTP has not terminated the protocol, then

TTP $\rightarrow A$: a_2

TTP $\rightarrow B$: a_2

a_2 = abort-token = a_1, $S_{TTP}(a_1)$

③ TTP terminates this execution of the protocol.

□

Recovery-A sub-protocol:

① $A \rightarrow$ TTP: ra_1

$ra_1 = m_1, m_2, m$

② if TTP has not terminated the protocol, then

TTP $\rightarrow A$: ra_2

TTP $\rightarrow B$: ra_3

$ra_2 = ra_1$, Delivered, S_{TTP}(Delivered, ra_1)

$ra_3 = ra_1$, $S_{TTP}(ra_1)$

③ TTP terminates this execution of the protocol.

□

Recovery-B sub-protocol:

① $B \rightarrow$ TTP: rb_1

$rb_1 = m_1, m_2, N_B$

② if TTP has not terminated the protocol, then

TTP $\rightarrow A$: rb_2

TTP $\rightarrow B$: rb_3

$rb_2 = rb_1$, $S_{TTP}(rb_1)$

$rb_3 = rb_1, m, S_{TTP}(rb_1, m)$

③ TTP terminates this execution of the protocol.

□

5. Conclusion

In this paper we presented a method for the analysis of offline fair exchange protocols in the strand space model. It is easy to use this intuitive method by pen and paper, or to do the searching automatically by machine. Two attacks were found when applying the method to the analysis of the ASW protocol. Our analysis revealed the reason for the weaknesses, and we made an improvement to the ASW protocol.

We also applied the method to ZG offline protocol [16], and did not find any weakness. But in paper [7], it is pointed out that the sub-protocol Recovery may end up with time-out because the communication channels in the ZG offline protocol are resilient. We should study this more in detail.

Our future work also includes how to formally illustrate the concurrent and exclusive relationship of protocol events in directed graphs in the strand space model.

6. References

[1] N. Asokan, M. Schunter, and M. Waidner, "Optimistic Protocols for Fair Exchange", in: 4th ACM Conference on Computer and Communications Security, ACM Press, Zurich, Switzerland, 1997, pp. 6, 8-17.

[2] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication", ACM Transactions on Computer Systems, 1990, 8(1),pp.18-36.

[3] Kailar R., "Reasoning about Accountability in Protocols for Electronic Commerce", Proceedings of the 1995 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Washington D.C. USA, 1995, pp.236-250.

[4] Kailar R., "Accountability in Electronic Commerce Protocols", IEEE Transaction on Software Engineering, Piscataway , 1996, 22(5), pp. 313-328.

[5] S. Kremer and J. Raskin, "Game Analysis of Abuse-free Contract Signing", In Steve A. Schneider, editor, 15th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, Washington, DC, 2002, pp. 206-220.

[6] S. Kremer and J. Raskin, "A Game-based Verification of Non-repudiation and Fair Exchange Protocols", Journal of Computer Security, Amsterdam, 2003, 11(3), pp.399-429.

[7] S. Kremer, "Formal Analysis of Optimistic Fair Exchange Protocols", Ph. D thesis, Universit'e Libre de Bruxelles Facult'e des Sciences, Brussels, 2004

[8] R. Alur, T. Henzinger, F. Mang, et.al. “Mocha: Modularity in Model Checking”, In Alan J. Hu and Moshe Y. Vardi, editors, Tenth International Conference on Computer-aided Verification, volume 1427 of Lecture Notes in Computer Science, Springer-Verlag, Vancouver, 1998, pp.521-525.

[9] F. J. Thayer Fabrega, Herzog J., and Guttman J., “Strand spaces: Proving Security Protocols Correct”, Journal of Computer Security, Amsterdam, 1999, 7(2/3), pp.191-230.

[10] J. Guttman, “Key Compromise, Strand Spaces, and the Authentication Tests”, Electronic Notes in Theoretical Computer Science, Elsevier B.V. ,2001, 45, pp.141-161.

[11] Xiangdong Li, and Qingxian Wang, “An Improvement of Authentication Test for Security Protocol Analysis”, Proceedings of 2007 International Conference on Computational Intelligence and Security – Workshops, IEEE Computer Society Press, Herbin, China, 2007, pp.745-748.

[12] Guttman J., “Security Protocol Design via Authentication Tests”, Proceedings of 15th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, Washington D.C., USA, 2002, pp.92-103

[13] Haifeng Shen, Rui Xu E, and Heyan Huan, “Analyzing Fair Exchange Protocols with Strand Spaces”, Mini-Micro Systems, Vol.27,Shenyang, China, 2006,pp.62-68

[14] N. Asokan, V. Shoup, and M. Waidner, “Asynchronous Protocols for Optimistic Fair Exchange”, In IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Oakland ,USA,1998, pp. 86-99.

[15] Zhou J., R. Deng, and Bao F., “Some remarks on a fair exchange protocol”, In Int.Workshop on Practice and Theory in Public Key Cryptography, vol. 1751 of LNCS, Springer-Verlag, 2000, pp.46-57.

[16] Zhou J., and Gollmann D., “An Efficient Non-repudiation Protocol”, Proceedings of the 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press, Los Alamitos,USA, 1997, pp.126-132.