Analysis and Improvement on a Single Unit Cyclic Fair Exchange Protocol for Multi-party

Nay Chi Htun, Khin Khat Khat Kyaw

Abstract—With the widespread utilization of e-commerce, improving fair exchange service becomes an important role in research area. A cyclic fair exchange protocol for multi party was proposed by Feng Bao, Robert Deng, Khanh Quoc Nguyen and Vijay Varadharajan in 1999. According to this protocol, the user must trust in not only the Trusted Third Party (TTP) but also the initiator. This paper proposes a modified multi-party fair exchange protocol that does not depend on the initiator in order to provide fairness.

Index Terms—e-commerce, cyclic-exchange, multi-party, fairness, initiator

I. INTRODUCTION

Fairness is one popular research topic in Network Security area. A protocol can be said as “fair” on condition that after performing that protocol, all members must have the same chance and same authority to influence another member. Assume that two members have their own important information and each member wants the information from the other. So each member agrees on one exchange protocol that he expects to get fairness. After performing this protocol, receiving respective information and same authority satisfied each member. Such a protocol is fair exchange protocol.

Fair exchange is usually performed with Trusted Third Party (TTP). TTP is the third party whom both parties. Without TTP, the actual fairness cannot be obtained. Depending on the level of TTP involvement, a protocol can be described in two types: online-TTP and offline-TTP. With online-TTP, every exchange step of the protocol must be noticed by TTP. With offline-TTP, TTP is necessary only when some unnatural behaviors occur. A protocol can be idealized to be optimistic if it can delete the involvement of TTP in the whole exchange process.

Later, researchers tried to gain fairness for more than two parties. These protocols are named as multi-party fair exchange (MPFE) protocols. MPFE based on two exchange methods: Cyclic and General. In Cyclic, the exchange’s topology is a ring in which each participant Pi offers to participant Pi+1 message mi in exchange of message mi-1 offered by participant Pi-1 [1]–[4]. In general, each entity can communicate with the set of entities of his choice [5]. Moreover, the exchanged item may be defined as single unit or multi units. So there are four classes of MPFE: single-unit cyclic exchange, single-unit general exchange, multi-unit cyclic exchange and multi-unit general exchange [6].

In this paper, a cyclic fair exchange protocol for multi-party will be modified and analyzed. In [1], the protocol is applicable for multi-unit or single-unit cyclic exchange. Multi-unit general model could also be achieved with some restrictions. However, participants who agree on the protocol must trust in both TTP and the initiator to achieve the fairness. In 2001, N. Gonzalez Deleito and O. Markowitch proposed a method to cancel the initiator dependence [2]. However, the communication overhead is dramatically increased because of many broadcasted messages [3]. In 2011, Yi Liu and Hongli Hu modified the same protocol to cancel the initiator dependence with low communication overhead. They use the equation: f(a). f(b) = f(ab), f(y) = y^2 mod N where N is the product of two prime numbers. There are many possible number pairs that satisfy above these equations. For example, f(15).f(16) = f(15.16) where N= 13*17. So authentication may be broken down by Dictionary attack. In the proposed protocol, the participants need to trust only in TTP. The communication overhead is not high because no message is broadcasted many times.

The paper is organized as follow. The original protocol proposed by Feng Bao, Robert Deng, Khanh Quoc Nguyen and Vijay Varadharajan is described and analyzed in section 2. The proposed protocol is presented and analyzed in section3. In section 4, the paper is concluded.

II. CYCLIC FAIR EXCHANGE PROTOCOL WITH OFF-LINE TTP

In this section, the single unit cyclic fair exchange protocol with off-line TTP introduced in [1] is briefly described. The notations are described as follow. For i=0, 1, 2…n-1

P_0 : the initiator of the protocol
m_i : the secret item owned by P_i and wanted by P_{i+1}
M_i : h (m_i) and at least known by P_{i+1} and TTP
C_i : the encrypted value of m_i under the public key e, cert_i the proof to convinced that m_i is truly encrypted under the key e_i

Here, TTP knows the status of P_0, public key (e) and private key (d) of the cryptosystem. The channel is resilient. The protocol use the verifiable encryption schemes to convince that m_i is actually encrypted under the key e_i.

A. Main Protocol

- \( P_i \rightarrow P_{i+1}: c_i, \text{cert}_i \) for \( i = 0, \ldots, \text{n-1} \)
  where \( c_i = e \) (m_i)\n- \( P_i \rightarrow P_{i+1}: m_i \) for \( i = 0, \ldots, \text{n-1} \).

Manuscript received May, 2013.

Ms. Nay Chi Htun. Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar

Mrs. Khin Khat Khat Kyaw Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

www.ijarcet.org
In the first round, $P_0$ sends $c_0$ and $\text{cert}_0$ to $P_1$. Then $P_1$ checks that verify $(c_0, \text{cert}_0, \text{SID}_0, \text{DID}_0)$ = yes to convince that $c_0$ is $c_0$ ($m_0$). After checking, $P_1$ sends $c_1$ and $\text{cert}_1$ to $P_2$ and so on till $P_{n-1}$. In the second round, $P_0$ sends $m_0$ to $P_1$. $P_1$ sends $m_1$ to $P_2$ and so on till $P_{n-1}$.

B. Recovery Protocol

- $P_1 \rightarrow \text{TPP}: c_{i+1}, \text{cert}_{i+1}$.
- $\text{TPP} \rightarrow P_{0i}: \text{call}$.
- $P_0 \rightarrow \text{TPP}: \text{yes or abort}$.
- $\text{TPP} \rightarrow P_1: m_1$ or abort.

In the second round, if $P_1$ does not receive the $m_1$, it can run the recovery protocol with $c_1$ and $\text{cert}_1$. Calling the recovery protocol may be two different ways. If $P_1$ is the first participant who runs the recovery protocol, then $\text{TPP}$ checks $\text{cert}_{i+1}$, ask $P_0$ whether $P_0$ receives $c_{n-1}$ and $\text{cert}_{n-1}$. If $P_0$ answers “yes”, $\text{TPP}$ replies “abort” message to $P_0$. If $P_1$ is not the first participant who calls the recovery protocol, $\text{TPP}$ makes the decision according to the first time.

C. Analysis of the Protocol

In the protocol, every participant must trust in $P_0$. Here $P_0$ can say false “yes” or “abort” to fool other parties and the honest parties can lose fairness. Assume that $P_1$ colludes with $P_0$ to defeat other parties.

Example 1: $P_0$ gets $m_1$ but did not send $m_1$ to $P_4$. When $P_4$ runs recovery protocol with $c_2$ and $\text{cert}_2$, $P_0$ says abort to $\text{TPP}$ even if it receives $c_{n-1}$ and $\text{cert}_{n-1}$. So $\text{TPP}$ replies “abort” message to $P_4$.

Example 2: $P_1$ gets $c_2$ and $\text{cert}_2$ but did not send $c_2$ and $\text{cert}_2$ to $P_4$. Then $P_4$ runs the recovery protocol to get $m_2$. When $\text{TPP}$ call $P_0$, $P_0$ replies “yes” even though it did not receive $c_{n-1}$ and $\text{cert}_{n-1}$. As described in two examples, it can be seen clearly that fairness can break down.

III. AN IMPROVED CYCLIC FAIR EXCHANGE PROTOCOL

In this section, the improved protocol will be described. The notations and assumptions are similar to those of the original protocol.

A. Main Protocol

- $P_i \rightarrow P_{si}: c_i, \text{cert}_i$ for $i = 0, \ldots , n-1$ where $c_i$= (mi, SID, DID)
- $P_i \rightarrow P_{si}: m_i$ for $i = 0, \ldots , n-1$

Main protocol is the same as the original protocol except $c_i$. In the original protocol, $c_i$ is the encrypted message of only $m_i$. So the dishonest party can cheat the information of the other party and run the recovery protocol.

Example: Assume that $P_2$ is dishonest party. Then $P_2$ Recovery Protocol can cheat $c_i$ and $\text{cert}_i$ ($i = 1, 3, 4, \ldots , n-1$) and run recovery protocol.

In the proposed protocol, $c_i$ includes the encrypted message of $m_i$, source ID (SID) and destination ID (DID). Therefore, $\text{TPP}$ can check the intended receiver is whether $P_2$ or not when it decrypts the $c_i$ with the key $d$ even if $P_2$ cheats $c_i$ and $\text{cert}_i$ ($i = 1, 3, 4, \ldots , n-1$) and runs the recovery protocol.

B. Recovery Protocol

- $P_i \rightarrow \text{TPP}: c_i, \text{cert}_i, c_{i+1}, \text{cert}_{i+1}$.
- $\text{TPP} \rightarrow P_{si}: c_i, \text{cert}_i$ or abort.
- $\text{TPP} \rightarrow P_i : m_i$ or abort.

If $P_i$ does not receive the $m_i$, it must run the recovery protocol with not only $c_i$ and $\text{cert}_i$ but also $c_1$ and $\text{cert}_1$. Then $\text{TPP}$ does not call $P_0$. Instead, $\text{TPP}$ sends $c_i$ and $\text{cert}_i$ to $P_{si}$ and $m_{n-1}$ to $P_i$ after $\text{TPP}$ verifies and satisfies with the information of $P_i$. Otherwise, $\text{TPP}$ sends abort message to $P_i$ and $P_{si}$.

C. Analysis of the Protocol

Proposition 1: After the protocol (including the recovery protocol) has executed, dishonest $P_i$ can never achieve $m_{n-1}$ without sending his secret information ($c_i$, $\text{cert}_i$).

Proof: Fairness means “Give and Take”. To take the one of the other, give the one of mine.

Condition 1: In the first round, for $i = 1, 2, 3, \ldots , n-1$, dishonest $P_i$ gets $(c_{i-1}, \text{cert}_{i-1})$ from $P_{i-1}$. But dishonest $P_i$ doesn’t give $(c_i, \text{cert}_i)$ to $P_{i+1}$. However $P_i$ wants $m_n$ and he cannot get $m_n$ from $c_{i-1}$ because $P_i$ doesn’t know the decryption key. So $P_i$ must call the recovery protocol with $(c_i, \text{cert}_i)$ as well as $(c_{i-1}, \text{cert}_{i-1})$. Consequently, $\text{TPP}$ gets $(c_i, \text{cert}_i)$ and $\text{TPP}$ can forward to other parties. For that reason, $P_i$ cannot call the recovery protocol.

Condition 2: In the second round, if $P_i$ gets $m_{n-1}$ from $P_{i+1}$ and doesn’t give $m_i$ to $P_{si}$, then $P_{si}$ can run recovery protocol and ask $m_i$ from $\text{TPP}$.

Condition 3: $\text{TPP}$ can independently recover the $m_{n-1}$ for every honest party $P_i$ without confirming the honesty of $P_i$ with the initiator.

Proposition 2: During the protocol, any dishonest party (external or internal) cannot run the recovery protocol with the cheating information ($c_i$, $\text{cert}_i$) in order to get the actual message $m_{n-1}$.

Proof: In the proposed protocol, $c_i$= (mi, SID, DID), where SID= source ID and DID= destination ID. So anyone cannot fool $\text{TPP}$ with illegal $c_i$.

IV. CONCLUSION

Fairness plays a vital role in e-commerce applications. The current multi-party cyclic fair exchange protocols still have weak-points such as initiator dependence, high
communication overhead. The proposed protocol can overcome those weak points. It can give the actual fairness for multi-party exchange applications without trusting the initiator. Communication overhead gets lower by deleting broadcasted messages. Moreover, the proposed protocol modifies the message $c_i$. Therefore, it can prevent the dishonest party from running the recovery protocol with the cheated information.

ACKNOWLEDGMENT

We foremost thanks go to Professor Dr. Aung Win, the Principal of the Technology of University in Yatanarpon Cyber City, for welcoming our research and giving a hand for us. Next, we would like to thank Professor Dr. Soe Soe Khaing, the Head of the Department of Information and Communication Technology in our university, for giving a chance to fulfill my goal. Moreover, I also wish to thank to the other members of our department for encouraging us and offering guidelines about our research. Finally, our thanks go to our families and friends for all the love and kindness they give us.

REFERENCES


Ms. Nay Chi Htun is a post-graduate student at University of Technology in Yatanarpon Cyber City, Myanmar.

Mrs. Khin Khat Khat Kyaw is an Assistant Professor at the faculty of Information and Communication Technology in University of Technology, Yatanarpon Cyber City, Myanmar.