A Fair Non-Repudiation Security Protocol with off-line TTP

Xi Wang[1], Xiangming Wen[1], Ye Liu[2], Xinqi Lin[1,3], Ying Wang[4]
1. School of Information and Telecommunication Engineering, BUPT, Beijing 100876, China
2. School of Economy and Management, BUPT, Beijing 100876, China
3. Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China
4. Yanzhou Coal Mining Company Limited, Yanzhou 252000, China

***Abstract*—A fair non-repudiation security protocol can provide non-repudiation proof for two parts of the exchange at the same time, or both get nothing. But many protocols performed weak in logical analysis. In this paper, a group of new protocols(with 3 sub-protocols) with off-line The Third Party(TTP) are presented. We analysis them with a new Kailar logic analytical method. Finally, the proposed protocols are proved to satisfy the fairness, non-repudiation and efficiency from the theory. And they propose some protections for the networks which have merely physical securities such as Internet, and provide some realistic reference values to the establishment of security mobile Ad-Hoc networks.**

Keywords-Fairness; Non-repudiation; New Kailar logic analytical method; Security

I. INTRODUCTION

With the rapid expansion of electronic commercial, cheats on Internet are more and more often. This has provide a new demand of the design of fair non-repudiation protocols. In the early years, people drawn their attentions on two-party computation protocol, but now only a few study this, Mehmet S. Kiraz and Berry Schoenmakers improved the Yao's Secure Two-party Computation Protocol[1], but it is still semi-honest and couldn't suit the demand of today's electronic commercial affairs.

In recent years the recommend of TTP (the 3rd party) has been a hot researching issue because researchers have reached an agreement that this is the best way to protect the fairness. The famous CMP1 Protocol, with transparent on-line TTP, TTP performs every time when the exchange parts need [2], which means it is required to be actively involved in every exchange transactions[3]. Asokan-Shoup-Waidner Protocol, carried out in 1998, because of supposing that the exchange parts are more likely honest, only a few are wicked, they use the system that TTP performs when it must(off-line TTP), unlike the CMP1, so it cannot prevent the fairness and security[4], the same property we can find in Chameleon-Based Optimistic Fair Exchange Protocol[5], we call them the optimistic fair exchange protocols[6]. But the protocols we propose bellow are not the optimistic fair exchange protocols, instead, we need a strong TTP(off-line) because we suppose that both parts may cheat in the exchange.

Researchers provide lots of protocols, but somehow these protocols cannot protect the fairness and non-repudiation, because normally, people analysis the security protocol in theory, not with strict logic analytical method, so they can only analysis their known weakness, but the latent weakness and disadvantages in the protocols have been missed[7], therefore, we should use logic analytical method to deal with them. There are numbers of methods proposed by scholars. In this paper, we use the new Kailar logic analytical method proposed by Qing Sihan[8].

The former Kailar logic analytical method can be seen in the reference[7], it cannot satisfy the fairness, but Qing Sihan improved it and give a congregation to each part of the transaction before the carry out of the protocol, and with the progress of the protocol, the congregation has expanded, and at the end of the protocol, every part of the transaction have a last congregation. The detailed new Kailar logic analytical method(Speculative Rules and process of analysis) proposed by Qing Sihan can been seen in the reference[9].

The rest of the paper is organized as follows. In next section we analysis the shortcomings of 2 famous protocols with a new Kailar logic analytical method proposed by Qing Sihan. In section 3 and 4, we present our new group of protocols and analysis them. Finally, we give the concluding remarks in section 5.

II. ANALYSIS OF CMP1 AND GFNR PROTOCOLS

We analysis of the CMP1 Protocol[5] and the Generic Fair Non-Repudiation Protocols with Transparent Off-line TTP[10] (GFNR) with a new Kailar logic analytical method proposed by Qing Sihan, some problems are discovered:

A. *Problems*

CMP1 protocol:

1. CMP1 protocol cannot directly prove $A \succ B \rightarrow m$ and $B \succ A \rightarrow m$;

2. In the 4th analysis, CMP1 protocol can not satisfy the demand that EOO and EOR belong to respective part of the transaction. This is hard to achieve and even though we achieve this, the efficiency of the protocol will be pathetic;

978-1-4244-3541-8/09/$25.00 ©2009 IEEE

The GFNP protocols:

The author did not encrypt the $^{(L,confirm)}$ in the Abort Protocol, which is the abort symbol of the protocol from TTP to A, it is dangerous to both part if the cheater steal the $^{(L,confirm)}$, the cheater can catch the important identifying information L and the abort symbol of the protocol AT from TTP to B, so he can pretend to be A or B, but at the same time, B thinks the protocol has been recovered, it is dangerous to B.

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. If you are using A4-sized paper, please close this file and download the file for "MSW A4 format".

B. Solutions

CMP1 protocol:

Due to CMP1 protocol can not directly prove $A \succ B \rightarrow m$ and $B \succ A \rightarrow m$, we can add information m or the encrypted m in the design of the protocol;

It is hard to satisfy that both part have the EOO or EOR in the 4th analysis, so we change our thought, we try to let the both part get nothing, before the end of the protocol, as to keep synchronous, we give both part nothing but at the end of the protocol, we give them their EOO and EOR, it is fair to both part and this can carry out easily;

The GFNP protocols:

1. Encrypt every message with the receiving part's public key, through this, only the receiving part can catch the content of the message;

2. In the abort protocol, the TTP should sign ^{L,confirm} with signature, so ^{L,confirm} can be a part of EOO or EOR from TTP, it is a compensation to A, if B request for arbitration with EOO, A can bring out $^{\{L,confirm\}_{K_{ttp}^{-1}}}$ as the symbol of the abort, it can avoid some unfairness.

III. The Design Of New Protocols

A. Consultations And Explanations

Based on the analysis of the 2 protocols above, we finally propose our new protocols, the new protocols have 3 parts:

Before we propose the protocols, let us give some introductions:

We assume that the TTP is linked with sender A and receiver B by resilient communication channels, i.e. messages inserted into such a channel will be eventually delivered to the recipient after a finite but unknown delay. However, the communication channel between A and B may be unreliable, i.e., messages inserted into such a channel may be lost.

M: message delivered to the receiver B from the sender A;

K : Conversation key that used to encrypt M;

K_a, K_b, K_{ttp} : The public key of A, B, TTP;

K_a^{-1}, K_b^{-1}, K_{ttp}^{-1} : The secret key of A, B, TTP;

f_K, f_{EOO}, f_{EOR}, f_{AT}, f_{Rec} : Publicly known unique flags that indicate distinct purposes of different messages in our protocol. So these can be part of EOO and EOR;

$\{\{abort\ confirm\}_{K_{ttp}^{-1}}\}_{K_a}$: Abort symbol from TTP to A, it can be non-repudiation evidence when protocols have been ended;

$EOO = \{\{f_{EOO}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$: Evidence of Origin, showing that A sent M to B;

$EOR = \{\{f_{EOR}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$: Evidence of Receipt, showing that B received M from A;

AT : Abort symbol issued by A to cancel the protocol;

Rec : Recover request from B to resolve the protocol;

$\leftrightarrow$: Deliver by repeatedly FTP operating system, we recommend that the A/B can contact TTP with repeatedly FTP operating system, the channel will not be broken off, which is first promoted in Zhou-Gollmann protocol[11].

B. The New Protocols

After introduction of some notations, the protocols are given bellow, the flow chart of the protocols is shown in the Fig. 1, and the broken lines are comments:

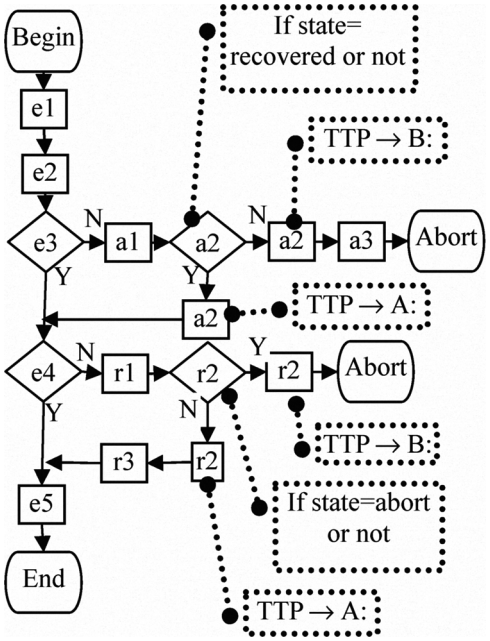

Figure 1. The flow chart of the protocols

The 1st part: the exchange protocol:

(e1). $TTP \rightarrow A:\{K,\{f_K,K\}_{K_{ttp}^{-1}}\}_{K_a}$

(e2). $A \rightarrow B:\{A,B,T,\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_a^{-1}}\}_{K_b}$

if A get no reply, or A get a wrong reply,
A apply for abort, turn to abort protocol

(e3). $B \rightarrow A:\{\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_b^{-1}}\}_{K_a}$

if B get no reply, or B get a wrong reply,
B apply for recovery, turn to recovery protocol

(e4). $A \rightarrow B:\{K\}_{K_b}$

(e5). $A \leftrightarrow TTP:\{\{f_{EOR}\}_{K_{ttp}^{-1}}\}_{K_a}$

$B \leftrightarrow TTP:\{\{f_{EOO}\}_{K_{ttp}^{-1}}\}_{K_b}$

The 2nd part: the abort protocol:

(a1). $A \rightarrow TTP:\{A,B,T,\{f_K,K\}_{K_{ttp}^{-1}},\{f_{AT}\}_{K_a^{-1}}\}_{K_{ttp}}$

(a2). *TTP* check,

if state=recovered

$TTP \rightarrow A:\{\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_b^{-1}}\}_{K_a}$

back to exchange protocol (e4),
continue the exchange protocol

or

$TTP \rightarrow A:\{\{abort\ confirm\}_{K_{ttp}^{-1}}\}_{K_a}$

(a3). $TTP \rightarrow B:\{A,B,T,\{f_{AT}\}_{K_a^{-1}}\}_{K_b}$

The 3rd part: the recovery protocol:

(*r*1). $B \rightarrow TTP:\{A,B,T,\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_a^{-1}}$,
$\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_b^{-1}},\{f_{\text{Rec}}\}_{K_b^{-1}}\}_{K_{ttp}}$

(*r*2). *TTP* check,

if state=abort

$TTP \rightarrow B:\{\{f_{AT}\}_{K_a^{-1}}\}_{K_b}$

or

$TTP \rightarrow A:\{A,B,T,\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_b^{-1}}\}_{K_a}$

(*r*3). $TTP \rightarrow B:\{K\}_{K_b}$

C. *The emphasis of the protocols*

In the protocols, when sending messages, encrypt the message with the receiver's public key, this prevent the security of the protocol, even the cheater catch the message, he can not decryption it so he can not catch the information M.

At the beginning of the exchange protocol, TTP send A $\{f_K,K\}_{K_{ttp}^{-1}}$ and a conversation key K, the reason why K is generated not by A but by TTP is that if A send a fake K to B, B can ask the TTP to inspect the K in the recovery protocol.

The $\{f_K,K\}_{K_{ttp}^{-1}}$, it is an identifier(ID) for identifying the A/B in the abort/recovery protocol, in case that some hacker imitate as A/B. We can see that if the A wants to cheat, and send B the wrong $\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K$ as to gain the EOR from B, it is impossible because the $\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_b^{-1}}$ from B is made up of the $\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_b^{-1}}$ from A, if A cheated, he can not get the right part of the EOR, so this can avoid A's cheating.

Now let's discuss Abort protocol. If the exchange protocol ends at the 2nd step, in this situation, A get nothing or get the wrong EOR, he can contact the TTP for the application of abort, TTP will respond due to the situations, at the 1st step of the abort protocol, A send TTP with A, B, T, $\{f_K,K\}_{K_{ttp}^{-1}}$ and $\{f_{AT}\}_{K_a^{-1}}$ (the symbol of applying abort), TTP check the 3 side of the exchange, then check if the $\{f_K,K\}_{K_{ttp}^{-1}}$ is valid, if not, drop the application(in case A cheat for abort), if valid, check the exchange system, if the exchange is recovered, then TTP send A the valid $\{\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}_{K_b^{-1}}\}_{K_a}$, which is a part of EOR, or send A $\{abort\ confirm\}_{K_{ttp}^{-1}}$ as the evidence of non-repudiation indicates him that his abort application has been accepted. Due to that $\{abort\ confirm\}_{K_{ttp}^{-1}}$ is signed by TTP, so it can be an evidence of non-repudiation, A can save this in case the controversy. Then TTP send $\{f_{AT}\}_{K_a^{-1}}$ to B as to inform that the exchange has been ended.

Let's discuss Recovery protocol, if B didn't get or get the wrong session key K, TTP will respond due to the different situations, at the 1st step of the recovery protocol, B send TTP with A, B, T, $\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_a^{-1}},\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_b^{-1}}$, and $\{f_{\text{Rec}}\}_{K_b^{-1}}$ (the symbol of applying recovery), TTP c check if the $\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_a^{-1}},\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_b^{-1}}$ is valid, drop the application if anyone of the two is not valid, if both are valid, decrypt f_{Rec}, then check the situation of exchange system, if the exchange is aborted, then TTP send B the A's abort application $\{f_{AT}\}_{K_a^{-1}}$, indicates B that the protocol has been aborted. Otherwise, send A $\{\{f_K,K\}_{K_{ttp}^{-1}},\{m\}_K\}\}_{K_b^{-1}}$, which is a part of EOR, then send B conversation key K, so the protocol back to the 5th step of the exchange protocol, both side A and B capture their $\{f_{EOO}\}_{K_{ttp}^{-1}}$ and $\{f_{EOR}\}_{K_{ttp}^{-1}}$ from TTP through FTP operating system, which is a important part of EOO / EOR.

IV. Analysis The Protocols With New Kailar Logic Analytical Method Proposed By Qing Sihan

We analysis the protocol with new Kailar logic analytical method proposed by Qing Sihan, due to the reference[8], the analysis has 4 processes, the 1st, 2nd and 3rd processes are analysis for the accountability, and the 4th process is for fairness:

A. Analysis Exchange protocol:

When Exchange protocol happens, it runs (e1) (e2) (e3) (e4) (e5):

(1)List the initial congregation:

$O_a^0 = \{K_a^{-1}, K_a, K_b, K_{ttp}\}$, $O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$

$A \succ (\xrightarrow{K_b} B, \xrightarrow{K_{ttp}} TTP)$, $B \succ (\xrightarrow{K_a} A, \xrightarrow{K_{ttp}} TTP)$

(2)List EOO and EOR , we have to analysis if $EOO \in O_b$, $EOR \in O_a$ is true, we can achieve the accountability or not:

$EOO = \{\{f_{EOO}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$

$EOR = \{\{f_{EOR}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$

Analysis of EOO :

Supposing that $EOO \in O_b$ is true, so $\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}} \in O_b$, so $\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}} \in B$, and due to $B \succ \xrightarrow{K_a} A$ use Autograph Rule:

$B \succ A \rightarrow \{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K$

use Connection Rule: $B \succ A \rightarrow \{f_K, K\}_{K_{ttp}^{-1}}$, $B \succ A \rightarrow \{m\}_K$

Because of $K_{ttp} \in A$, we can prove $B \succ A \rightarrow K$, that is $B \succ K \in A$,

Now $B \succ A \rightarrow \{m\}_K$ and $B \succ K \in A$, use Cryptograph Understanding Rule we can prove $B \succ A \rightarrow m$, so the receiver B can prove that sender A is accountability for m , so the EOO satisfy accountability.

Analysis of EOR is same as Analysis of EOO .

So both EOO and EOR satisfy accountability.

(3) When the protocol ended natural, we will analysis to see if $EOO \in O_b$ $EOR \in O_a$ are true:

list O_a^i and O_b^i, $i = \{1,2,3,4,5\}$, (e1) (e2) (e3) (e4) (e5):

$O_a^2 = O_a^1 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}\}$

$O_a^3 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$

$O_a^4 = O_a^3 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$

$O_a^5 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}, \{f_{EOR}\}_{K_{ttp}^{-1}}\}\}$

$O_b^1 = O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$

$O_b^3 = O_b^2 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$

$O_b^4 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$

$O_b^5 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}, \{f_{EOO}\}_{K_{ttp}^{-1}}\}\}$

When the protocol ended, we can see that:

By the investigate of O_a^5 , A get 2 parts: $\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}$ and $\{f_{EOR}\}_{K_{ttp}^{-1}}$, combine them: $EOR = \{\{f_{EOR}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$.

By the investigate of O_b^5 , B get 2 parts: $\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}$ and $\{f_{EOO}\}_{K_{ttp}^{-1}}$, combine them: $EOO = \{\{f_{EOO}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$.

Both parts obtain their evidence of non-repudiation(EOO and EOR) at the end of the protocol.

(4) When the channel is unreliable, (e1) (e2) (e3) (e4) may be break off(due to (e5) is operated by repeatedly FTP operating system, the channel will not be broken off:

The protocol satisfies fairness and non-repudiation equals the conjectures are true bellow:

$EOO \in O_b^{i-1}$ if and only if $EOR \in O_a^{i-1}$, $i = 1,2,3,4$, here we find lead both parts obtain their evidence of non-repudiation is hard, we change our angle and try our best to let the both part get nothing at the same time, we give both part nothing but at the end they got their evidence of non-repudiation, it is fair to both part and this can carry out easily.

We can see that $EOO \notin O_b^i$ if and only if $EOR \notin O_a^i$, neither A or B obtain their evidence of non-repudiation(EOO and EOR), it means only at the end of the protocol can both parts obtain them, this guarantee the fairness and non-repudiation.

The conjectures are true, so the protocol is fairness and non-repudiation when the channel is unreliable.

B. Abort Protocol

In the Abort protocol, there are 2 situations: 1.the transaction recovered; 2.the transaction finally aborts.

The 1st and 2nd processes of the analysis are same as Exchange protocol. Let's deal with the 3rd and 4th.

1) If the transaction has been recovered:

(3) It runs (e1) (e2) (a1) (a2) (e4) (e5).

$O_a^3 = O_a^2 = O_a^1 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}\}$

$O_a^4 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$

$$O_a^6 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}, \{f_{EOR}\}_{K_{ttp}^{-1}}\}\}$$

$$O_b^1 = O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$$

$$O_b^4 = O_b^3 = O_b^2 \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^5 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^6 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}, \{f_{EOO}\}_{K_{ttp}^{-1}}\}\}$$

We can see the O_b^6 and O_a^6 is the same as the O_b^5 and O_a^5 in the Exchange protocol, so result is the same, Both parts obtain their evidence of non-repudiation(EOO and EOR) at the end of the protocol.

(4) The result is the same as the Exchange protocol, so the conjectures are true, so the protocol is fairness and non-repudiation when the channel is unreliable.

2) If the transaction finally aborts:

(3) It runs (e1) (e2) (a1) (a2) (a3):

$$O_a^3 = O_a^2 = O_a^1 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}\}$$

$$O_a^5 = O_a^4 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{abort\ confirm\}_{K_{ttp}^{-1}}\}$$

$$O_b^1 = O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$$

$$O_b^4 = O_b^3 = O_b^2 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^5 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}, \{f_{AT}\}_{K_a^{-1}}\}$$

By the investigate of O_a^5, A get $\{abort\ confirm\}_{K_{ttp}^{-1}}$, we have discussed this in C part section III. It can be the evidence of non-repudiation sighed by TTP.

By the investigate of O_b^5, B get $\{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}$ and $\{f_{AT}\}_{K_a^{-1}}$, we have discussed this in C part section III. It can be an evidence of non-repudiation sighed by A.

Both parts obtain evidence of non-repudiation($\{f_{AT}\}_{K_a^{-1}}$ and $\{abort\ confirm\}_{K_{ttp}^{-1}}$) at the end of the protocol.

(4) By comparing every O_a^i and O_b^i, the conjectures are true, so the protocol is fairness and non-repudiation when the channel is unreliable.

C. Recovery Protocol

In the recovery protocol, there are also 2 situations: 1.the transaction aborted; 2.the transaction finally recovers.

The 1st and 2nd processes of the analysis are same as Exchange protocol. Let's deal with the 3rd and 4th.

1) If the transaction has been aborted:

(3) It runs (e1) (e2) (e3) (r1) (r2).

$$O_a^2 = O_a^1 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}\}$$

$$O_a^4 = O_a^3 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$$

$$O_a^5 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}, \{abort\ confirm\}_{K_{ttp}^{-1}}\}$$

$$O_b^1 = O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$$

$$O_b^4 = O_b^3 = O_b^2 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^5 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}, \{f_{AT}\}_{K_a^{-1}}\}$$

We can see the O_b^5 / O_a^5 have $\{f_{AT}\}_{K_a^{-1}} / \{abort\ confirm\}_{K_{ttp}^{-1}}$, so it is the same as *b)* of Abort protocol, so the result is the same.

Both parts obtain evidence of non-repudiation($\{f_{AT}\}_{K_a^{-1}}$ and $\{abort\ confirm\}_{K_{ttp}^{-1}}$) at the end of the protocol.

(4) By comparing every O_a^i and O_b^i, the conjectures are true, so the protocol is fairness and non-repudiation when the channel is unreliable.

2) If the transaction finally recovers:

(3) It runs (e1) (e2) (e3) (r1) (r2) (r3) (e5).

$$O_a^2 = O_a^1 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}\}$$

$$O_a^6 = O_a^5 = O_a^4 = O_a^3 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$$

$$O_a^7 = \{K_a^{-1}, K_a, K_b, K_{ttp}, K, \{f_K, K\}_{K_{ttp}^{-1}}, \{f_{EOR}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_b^{-1}}\}$$

$$O_b^1 = O_b^0 = \{K_b^{-1}, K_b, K_a, K_{ttp}\}$$

$$O_b^5 = O_b^4 = O_b^3 = O_b^2 = \{K_b^{-1}, K_b, K_a, K_{ttp}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^6 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

$$O_b^7 = \{K_b^{-1}, K_b, K_a, K_{ttp}, K, \{f_{EOO}\}_{K_{ttp}^{-1}}, \{\{f_K, K\}_{K_{ttp}^{-1}}, \{m\}_K\}_{K_a^{-1}}\}$$

We can see the O_b^7 and O_a^7 is the same as the O_b^5 and O_a^5 in the Exchange protocol, so result is the same, Both parts obtain their evidence of non-repudiation(EOO and EOR) at the end of the protocol.

(4) By comparing every O_a^i and O_b^i, the conjectures are true, so the protocol is fairness and non-repudiation when the channel is unreliable.

Compare with the 2 protocols we analysis above, it is clear that the protocols are fair under the situation that the channel is unreliable, the protocols satisfied the demand, it not only guarantee fairness, non- repudiation and accountability, but also did some achievements in security.

As we bring the idea that through repeatedly FTP operating system to get the information that guarantee the fairness, so the

efficiency should be influenced, but we believed that in the days of the flourish of network cheating and hacker, the security is the most important thing that we should pay more attention to, because we encrypt every message with the receiver's public key and we use repeatedly FTP operating system, the protocol is fit to the networks that have merely physical securities like Internet[12], and provide some realistic significance to the establish of security mobile Ad-Hoc networks.

CONCLUSION

Protocols for fair exchange have found numerous practical applications in electronic commerce. In this paper, in order to satisfy fairness and non-repudiation, we propose a new group of protocols with repeatedly FTP operating system and off-line TTP, by analysis of logical method, the protocols and provide some realistic significance to the establish of security mobile Ad-Hoc networks.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China Grants No.60743007, and No. 60872050, and Beijing Municipal Commission of Education Disciplines and Graduate Education Projects Grants No.XK100130648.

REFERENCES

[1] Hernandez-Ardieta; Jorge, L "An optimistic fair exchange protocol based on signature policies". Computers and Security, Vol. 27, No. 7-8, December, 2008, pp. 309-322

[2] Wen, Jinghua; Tian, Jianqiang; Li, Xiang "New fair and non-repudiation protocol". Shanghai Computer Society, Shanghai, China, 2006, pp.132-134

[3] N. Asokan "Fairness in Electronic Commerce". Computer Science/Mathematics, 1998.

[4] Michael, Backes; Anupam, Dattab; Ante, Derek; John C, Mitchell "Compositional analysis of contract-signing protocols". Theoretical Computer Science, 2006, Volume 367, Issues 1-2, 24 November 2006, pp.33-56

[5] Xuan Yang, Zhaoping Yu, Bin Kang "Chameleon-Based Optimistic Fair Exchange Protocol". The 2008 International Conference on Embedded Software and Systems (ICESS2008), 2008, pp.298-302

[6] Kiraz, Mehmet S; Schoenmakers, Berry "An efficient protocol for fair secure two-party computation". Lecture Notes in Computer Science, v 4964 LNCS, Topics in Cryptology - CT-RSA 2008 - The Cryptographers' Track at the RSA Conference 2008, Proceedings, 2008, pp. 88-105

[7] Qing, Sihan "Operating System Secrity". Tsinghua University Press, 2004, pp.88-98

[8] Qing, Sihan; Gong, Li "Security Protocol" .Tsinghua University Press, 2005, pp.56-74

[9] Li, Li; Zhang, Huan-Guo; Wang, Li-Na "An improved non-repudiation protocol and its security analysis". Wuhan University Journal of Natural Sciences, 2004, pp.88-98

[10] Guilin Wang "Generic Fair Non-Repudiation Protocols with Transparent Off-line TTP". IOS Press, 2003

[11] Zhou, J; Gollmann, D; "A fair non-repudiation protocol". Proceedings of the 1996 17th IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 1996, pp.55-61

[12] Lei, Zhu; Nyang, DaeHun; Lee, KyungHee; Lim, Hyotaek "Fair reputation evaluating protocol for mobile ad hoc network". 2006 International Conference on Computational Intelligence and Security, ICCIAS 2006, v 1, 2006 International Conference on Computational Intelligence and Security, ICCIAS 2006, 2007, pp. 613-616