2013 IEEE International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing

An Optimistic Non-Repudiation Protocol focused on Transparent Trusted Third Party

Qiang Li, Kaigui Wu, 1st Affiliation
College of Computer Science
Chongqing University
Chongqing, China
e-mail: cqulq@sina.com, kaiguiwu@cqu.edu.cn

Fang Li, 2nd Affiliation
College of Information Engineering
Chongqing City Management College
Chongqing, China
e-mail: cqlifang2006@126.com

***Abstract*—Non-repudiation service is widely employed in information exchange and online transaction. It requires that its participants are unable to deny their involvement in specific protocol. At the same time, the service should keep fairness, which means that no one could take any advantages over the other one. Researchers have introduced a trusted third party (TTP) to realize these demands. Protocols are grouped into several categories according to the degree of TTP's involvement in protocols. Optimistic non-repudiation protocol is the most important method undoubtedly owe to its lower TTP's involvement. There are two questions. One is that whether it is possible to further decrease TTP's involvement or not. A computable optimistic non-repudiation protocol seems to be a good improvement. This is also our paper's basic protocol. The other question is about whether two types of finally evidences that produced by participants (here is Alice and Bob, and we denote this type evidences as the first type evidences) and TTP (or with the help of TTP, and we demote this type evidences as the second type evidences) are identical. If both types of evidences are not exactly the same, then it shows that TTP has taken part in process of protocol. Thus, it is certain that either participants behaved dishonest, or network went wrong. Participants' prestige will be negatively affected more or less. On the contrary, if two types of evidences are identical, then such feelings will be eliminated. This kind of TTP is called transparent TTP. In this paper, we have accomplished an optimistic non-repudiation protocol with the help of transparent TTP based on a computable non-repudiation protocol.**

Keywords-optimistic; Trusted Third party; evidences; transparency

I. Introduction

The development of internet exchange and online shopping is making an increasing difference to people's lives and becoming an essential part of more and more people's daily lives. However, some increasingly severe secure problems are in accompany with the convenience, such as repudiation in information exchange, cheating in online shopping, the leakage of personal information in unconsciously behaviors. Fortunately, many solutions to corresponding problems are proposed and are getting better over time. Non-repudiation services are one of these solutions.

Non-repudiation services are aimed at preventing participants denying having participated in a part or the whole of specific protocols. What's more, these protocols could ensure that when protocols end, no one that involving in protocols can achieve any advantages over the other one. In short, non-repudiation services could realize two main functions: non-repudiation and fairness. According to paper [1], a trusted third party (TTP) is of the essence to make these two properties come true.

When looking at the TTP, we have to consider how intense the TTP's involvement is. If an TTP takes part in the executing of a protocol frequently and in excess, the TTP may fall into DoS, caused by its limited performance. As a result, many specific non-repudiation protocols are do their efforts to decrease the involvement of TTP during a running protocol under the condition of non-repudiation and fairness. Therefore, five situations that may arrive should be taken into account. The first one is non-repudiation protocols without TTP. The paper [1] shows that a protocol without TTP or other roles that act as TTP cannot keep fairness or non-repudiation. Hence, many protocols without TTP have turned their eyes into gradual exchange of the expected information; papers [2-3] are the ones. With the help of gradual message exchange, the protocols can realize exchanging "the fraction of a bit". However, besides undoubtless fairness, the protocols are limited in two aspects, that is, the same or similar power of computable between two participants and resulting in masses of communications. These protocols are in dilemmas. Some amelioration came with the idea of probabilistic protocols. Paper [4] and [5] are two protocols that provide probabilistic fairness. Similar powers of compute are not required, but communications are the same. Besides, some other protocols without TTP also attempt to keep the two properties in a different place, such as paper [6] containing a system called "pub", using for recording all operations that act as evidences to keep fairness and non-repudiation. The second one is non-repudiation protocols with inline TTP [7-8]. In these kinds of protocols, TTP plays a role as a delivery authority and intervenes in each transmission of message. Obviously, a communication and computation bottleneck can be triggered by the heavy involvement of the TTP. Afterwards, the use of TTP is limited in each protocol run, but not in each transmission. This is called an online TTP. Papers [9] and [10] are the right protocols that using this method to decrease TTP's involvement and eliminate the above bottleneck. A further improvement in reducing the TTP's improvement own to offline TTP (which is also called an optimistic approach because of TTP's independent

978-0-7695-5088-6/13 $31.00 © 2013 IEEE
DOI 10.1109/HPCC.and.EUC.2013.101

of exchange and only dealing with disputes). This approach is regarded as the most efficient solution to decrease TTP's involvement. Only problems occurring, does TTP intervene to finish the protocols, which realizes a successful exchange or ends with no exchange taking place [11-13]. In other words, if the participants are honest and the communication channels function well, TTP is unoccupied. TTP's involvement is reduced largely. The last notion is denoted as transparent TTP that based on offline TTP. By using of a transparent TTP, it is impossible to detect whether TTP has intervened during the protocols running or not at the end of the protocols only with the help of evidences [14-15]. Furthermore, it is impossible to decide whether some problems have happened during the protocol running. Thus, the reputation of two participants can be unrevealed. These advantages are very useful in electronic commerce [16-17].

According to the development of non-repudiation services, we can sum up that optimistic non-repudiation protocols have been the focus of research for its lower involvement of TTP under the condition of fairness and non-repudiation.

The main idea of this paper is to accomplish a non-repudiation protocol based on transparent TTP. We attempt to realize some transparent evidences by means of an combination protocol, which is the fruit of effort of decreasing TTP's involvement, under the condition of non-repudiation and fairness. The rest of the paper is organized as follows. We introduce some basic definitions together with some necessary and rational assumptions in order to express clearly firstly. Then, the simple basic protocol we employed is described in section 3. In section 4, we construct a specific optimistic transparent non-repudiation protocol. This part is our key content and will be described in detail. The next consists of some conclusions and further work of this paper. The last part is an acknowledgement for the supports.

II. Preliminary Definitions and Properties

In this section, we introduce some basic definitions of the non-repudiation service. We consider two aspects, communication channels and properties. They are depicted in the following.

A. Communication Channels

The status of communication channels can be divided into three types: unreliable channels, resilient channels and operational channels. Paper [18] has done a more detailed explanation of them. When using an unreliable channel, data may be lost and the time we aware of this phenomenon are uncertain. A resilient channel delivers correct data in a finite but unknown amount of time. That is, the data can be correctly sent to receiver eventually, no matter what time it delays. An operational channel is the most rigorous one which admits correct data arriving in a known, constant amount of time. In reality, obviously, the first and second channels, namely unreliable and resilient channels, are pervasive. Here, we consider communication channels between two participants is unreliable channel while between each participant and TTP are operational due to TTP's status. This is reasonable.

B. Properties

Obviously, we always avoid something that is harmful to ourselves. Hence, the assumption that participants will not acts against its own interests. This is a normal assumption. We will ignore such problems when dealing with situations where a dishonest involved.

The basic properties of non-repudiation service are the basic requirements at the same time. They contain non-repudiation, fairness, transparency, timeliness [1,2,5,6], etc.. Before introduction of these notions, we define two non-repudiation evidences [14,18]. The participants referred in protocol are denoted as Alice and Bob.

Definition 1 (Non-repudiation of origin). A protocol provides non-repudiation of origin, if and only if it generates a non-repudiation of origin evidence, destined to Bob, that can be presented to an adjudicator, who can unambiguously decide whether Alice is the author of a given message or not.

Definition 2 (Non-repudiation of receipt). A protocol provides non-repudiation of receipt, if and only if it generates a non-repudiation of receipt evidence, destined to Alice, that can be presented to an adjudicator, who can unambiguously decide whether Bob received a given message or not.

Non-repudiation consists of the above two definitions. If the protocol fulfills these definitions properly, corresponding evidences can be realized correctly and further the protocol makes non-repudiation come true.

Definition 3 (Fairness). A non-repudiation protocol provides fairness if and only if at the end of a protocol execution either Alice got the non-repudiation of receipt evidence for the message m, and Bob got the corresponding message m as well as the non-repudiation of origin evidence for this message, or neither of them got any valuable information.

From definition 3, we can see that, at the end of protocol, two situations should be considered. One is that protocol is end with correct message exchanged and corresponding evidences distributed. This is the best status. The other is no participant has had any advantage over the other one. The protocol stays fairness.

Definition 4 (Transparency). By only looking at the generated evidences, it is impossible to decide whether the TTP did intervene or not.

In other words, the generated evidences are independent of the fact whether the TTP did intervene in the protocol or not. As we know that, transparent TTP derives from offline TTP. The difference between them is the type of evidences. In reality, TTP intervenes in protocol mainly caused by network failures, rather than a dishonest behavior of a participant. So using of transparent TTP can be very meaningful.

Definition 5 (Timeliness). A non-repudiation protocol provides timeliness if and only if all honest participants always have the ability to reach, in a finite amount of time, a point in the protocol where they can stop the protocol while preserving fairness.

To reach this property, a time factor should be introduced in specific protocol.

Up to this point, we have defined several basic properties, among which we will focus on the use of transparent TTP under the condition of non-repudiation and fairness.

III. A combination Optimistic Non-Repudiation Protocol

The development process of non-repudiation protocol is on the way to reduce TTP's involvement while keeping fairness and non-repudiation. The degree of TTP interference is decreased from inline TTP to offline TTP (without TTP, the protocol cannot stay fairness). On the foundation of offline TTP and without TTP, another idea on the combination of [5] and [11] to reduce this involvement is effective. We call it computable optimistic non-repudiation protocol.

The word "computable" refers that the involvement of TTP can be computed. In other words, we can locate a specific position where TTP may be introduced with the help of a selected random positive integer. If something happens in and only in the position, the TTP will come into play. Given the circumstances, we can calculate the probability of TTP's involvement based on random integer and can keep protocol reaching fairness and non-repudiation at the same time. At this, the specific protocol, consisting of two sub-protocols (main protocol and recovery protocol), is to achieve such exchanges: Alice expects to send Bob a message m and its corresponding non-repudiation of origin evidence against a non-repudiation of receipt evidence, which is declared by Bob. The main protocol is the basic one that could complete message exchange in a faultless situation while the other one handle the exceptions such as dishonest behaviors and network faults. Considering the paper's emphasis, we put main protocol in stage and leave recovery protocol out. Of course, we will describe recovery protocol in detail in section 4. Main protocol of this idea can be stated as follows.

Message 1:

$$A \rightarrow B : f_{EOO}, B, TTP, \ell, c, t_1, EOO$$

Message 2:

$$B \rightarrow A : f_{EOR}, A, TTP, \ell, t_2, EOR$$

Message 3:

$$A \rightarrow B : f_{EOO_{k,1}}, B, \ell, Round_1, k_1, t_3, EOO_{k,1}$$

Message 4 :

$$B \rightarrow A : f_{EOR_{k,1}}, A, \ell, t_4, EOR_{k,1}$$

……

Message 2n-1:

$$A \rightarrow B : f_{EOO_{k,n-1}}, B, \ell, Round_{n-1}, k_{n-1}, t_{2n-1}, EOO_{k,n-1}$$

Message 2n:

$$B \rightarrow A : f_{EOR_{k,n-1}}, A, \ell, t_{2n}, EOR_{k,n-1}$$

Message 2n+1:

$$A \rightarrow B : f_{EOO_{k,n}}, B, \ell, Round_n, k, t_{2n+1}, EOO_{k,n}$$

Message 2n+2:

$$B \rightarrow A : f_{EOR_{k,n}}, A, \ell, t_{2n+2}, EOR_{k,n}$$

We first need to choice a random positive integer n, which is absolutely unknown to Bob, to decide how many rounds the protocol should carry out. Obviously, the larger n is, the more the rounds are. In the final analysis, the value of n is a matter of how about the importance of exchanging message and network congestion situation. Next, message 1 consists of receiver B, TTP, the label of this exchanges ℓ, ciphertext c of m encrypted by secret key k, timestamp t_1 and evidence of origin EOO. Alice sends encrypted message to Bob all at once to initiate a session, indicating the protocol has begun. Bob accepts the message 1 and checks its format and validity to decide whether to respond to the message or not. If both are correct, he replies message 2 containing corresponding evidence of receipt. If Alice gets the message 2 within a proper time, she sends message 3 to Bob. If the value of n is 1, the key k_1 in message 3 is the right key that can be used to exactly decrypt ciphertext c. Otherwise, k_1 is exactly the same format to k except contents and effectiveness to c. So are message 4, message 5, $\cdots$, message 2n, and $k_2, k_3, \ldots, k_{n-1}$. Alice releases the true key k to Bob until step $2n+1$. Because of ignorant of n and key k, Bob will find it is difficult to make a right choice to stop replying corresponding evidence and end with some advantages. When looking at the protocol, if Bob refuses to reply the evidences from the rounds of step 2 to step $2n$, Alice can detect that Bob wants to deceive and stops the next exchange. The protocol remains fair. If Bob is too sagacity to stop exchange after step $2n+1$, then Alice will launch recovery protocol. The protocol will also end in a fair way with message exchanging successfully with the help of TTP. Thus, TTP will take part in protocol only after step $2n+1$. The probability of TTP's involvement is $1/n$. If some situation causes unfairness and repudiation, but not in step $2n+1$, it is unnecessary for TTP to intervene protocol execution. It has decreased the degree of TTP's involvement to a certain degree.

The fairness and non-repudiation of protocol will be guaranteed by some other sub-protocols, such as recovery protocol, abort protocol, error protocol. We will make use of this effective idea to develop our own protocol in section 4.

IV. A Optimistic Non-Repudiation Protocol with Transparent TTP

A. Introduction

The protocol we will propose is transparent based on optimistic non-repudiation protocol, having aimed at protecting participants' reputation and hiding their personal information. These will be reached by produced finally evidences. Specifically, finally evidences produced by participants purely or with the help of TTP are unconsciousness to outsiders which free from participants and TTP. Hence, outsiders cannot make a correct judgment whether TTP has involved in protocol or not or further uncertain of the correct of protocol running. The processes of executing protocol, including participants' dishonest behaviors and network failures, are concealed.

For other properties, such as non-repudiation and fairness referred above, we will illustrate in detail when describing the specific protocol. Actually, if protocol realizes transparency, fairness and non-repudiation are also on the way. Transparency is built upon effective evidences, and the evidences ensure fairness and non-repudiation.

The reason why we use offline TTP is that the protocol has a great advantage in reducing TTP's involvement. With the help of idea proposed in section 3, we could reduce TTP's involvement under the condition of the properties.

B. Notations and Evidences

This part introduces referred evidences. The evidences are produced in each protocol and used to proof participants have taken part in corresponding protocol and have practiced some behaviors. The goal of the protocol is to realize information exchanges, and the correct exchanges are based on evidences. In order to introduce evidences clearly, we need to denote some symbols.

A *: Alice, here, the paper denotes the sender;*

B *: Bob, here, the paper denotes the receiver;*

c *: the ciphertext that will be exchange;*

m *: message that expects to be exchanged;*

k_n *: the key that used to encrypt messages and is only known to Alice;*

k_i *: the* i *th meaningless string; these strings' type are the same as* k_n*;*

TTP*: the Trusted Third Party;*

Sig_X *: the signature of principal* X *;*

k_X *: the private key of* X *,which is used to sign message.*

$E_X(m)$ *:* m *is encrypted with* X *'s secret key;*

f *: indicating the role of item;*

ℓ *: indicating the specific running of protocol;*

$Round_i$ *: a counting that sticks to random number* n *;*

Then, we can define the items.

1) *The origin evidence of* c : $EOO = Sig_A(f_{EOO}, B, TTP, E_{TTP}(k_A), \ell, c)$

2) *The receipt evidence of* c : $EOR = Sig_B(f_{EOR}, A, TTP, E_{TTP}(k_B), \ell, c)$

3) *The origin evidence of* k_i : $EOO_{k,i} = Sig_A(f_{EOO_{k,i}}, B, TTP, \ell, i, k_i)$

4) *The receipt of* k_i : $EOR_{k,i} = Sig_B(f_{EOR_{k,i}}, A, TTP, \ell, i, k_i)$

5) *The submission of* k_n : $Sub = S_A(f_{Sub}, B, \ell, E_{TTP}(k_n))$

6) *The request of recovery:* $\mathrm{Re}c_A = S_A(f_{\mathrm{Re}c}, A, B, \ell)$

7) *The request of abort:* $Abort = S_A(f_{Abort}, B, \ell)$

8) *The confirmation of key* k_n : $Con_k = S_{TTP}(f_{Con_k}, A, B, \ell, k_n)$

9) *The confirmation of abort:* $Con_a = S_{TTP}(f_{Con_a}, A, B, \ell)$

10) *The confirmation of error:* $Con_e = S_{TTP}(f_{Con_e}, A, B, \ell)$

11) *The non-repudiation evidence of origin:* $NRO = \{EOO, EOO_{k_n,n}\}$

12) *The non-repudiation evidence of receipt:* $NRR = \{EOR, EOR_{k_n,n}\}$

The first nine evidences are generated during the protocol and belong to different sub-protocols. The last two, which are so called final evidences, are formatted by evidences 1), 3) and 2), 4). In other words, if we want to achieve the final evidences, we have to possess the above four exactly.

C. the Protocol

The protocol we proposed consists of four sub-protocols: main protocol, recovery protocol, abort protocol and error protocol. With the aid of these protocols, the integrated protocol can achieve non-repudiation, fairness, lower involvement of TTP and the key point of the paper, transparence. Main protocol is able to finish information exchange correctly in the right condition that participants behave honest and network works well. Recovery protocol is launched by one of participants while something have happened falsely to main protocol and efforts to complete exchange with TTP's help or results in other protocol running. Participants can also initiate abort protocol. Its purpose is to abort exchange under the condition of relevant properties. Error protocol is triggered by recovery protocol. When a participant attempts to deceive by launching recovery protocol, the error protocol can stop it. The details of four protocols are presented underneath.

a) Main Protocol

Let's start with the execution of main protocol and then give the detailed explanation.

Message 1:

$A \rightarrow B : f_{EOO}, f_{Sub}, B, TTP, \ell, h(k_n), c, t_1, E_{TTP}(k_n), EOO, Sub$

Message 2:

$B \rightarrow A : f_{EOR}, A, TTP, \ell, t_2, EOR$

if A *times out, then launching abort protocol*

Message 3:

$A \rightarrow B : f_{EOO_{k,1}}, B, \ell, Round_1, k_1, t_3, EOO_{k,1}$

Message 4:

$B \rightarrow A : f_{EOR_{k,1}}, A, \ell, t_4, EOR_{k,1}$

……

Message 2n-1:

$A \rightarrow B : f_{EOO_{k,n-1}}, B, \ell, Round_{n-1}, k_{n-1}, t_{2n-1}, EOO_{k,n-1}$

Message 2n:

$B \rightarrow A : f_{EOO_{k,n-1}}, A, \ell, t_{2n}, EOR_{k,n-1}$

Message 2n+1:

$A \rightarrow B : f_{EOO_{k,n}}, B, \ell, Round_n, k_n, t_{2n+1}, EOO_{k,n}$

Message 2n+2:

$$B \rightarrow A : f_{EOR_{k\ ,n}}, A, \ell, t_{2n+2}, EOR_{k,n}$$

if A *times out, then launching recovery protocol*

This main protocol is based on the idea of section 3. Before executing protocol, we need to choose a random positive integer n, which is known to Alice. Besides, the protocol has its own behaviors. We can see that if B has not received the next message in a limited time, he cannot decide whether to launch recovery protocol or not to press the protocol running rather than to keep fairness. It is reasonable. If A stops sending items to B, she cannot gain the corresponding evidences and will be detected attempting to deceive. A has not taken any advantages except for divulging ciphertext to B. B has no need to launch recovery protocol for losing nothing.

However, in the step 2n+2, it is essential for A to launch recovery protocol. In this point, B has mastered ciphertext as well as the right key and is able to constitute non-repudiation evidences of origin. If A has not received message 2n+2 in a limited time, she can ensure B plays dishonest or network breaking and launches recovery protocol to rebuild evidences and to keep fairness.

The main protocol can finish exchange under right condition and makes TTP keep silence. This is a character of offline TTP. Otherwise, TTP will do its work in other protocols.

b) Recovery Protocol

Only Alice could initiate recovery protocol. Its processes are described below.

Message 1:

$$A \rightarrow TTP : f_{\mathrm{Rec}_A}, f_{Sub}, B, \ell, h(c), h(k_n), E_{TTP}(k_n), \mathrm{Re}\,c_A, Sub, E_{TTP}(n), EOO, EOR$$

if $h(k_n)$ *is not the same with* $h(D_{TTP}(E_{TTP}(k_n)))$, *then launching error protocol*

if abort protocol or recovery protocol is running, then stop

else recovery protocol is true

Message 2:

$$TTP \rightarrow A : f_{NRR}, A, \ell, NRR$$

Message 3:

$$TTP \rightarrow B : f_{NRO}, B, k_n, \ell, NRO$$

The recovery protocol is mainly with the help of TTP. In the first step, in order to recover fairness and rebuild evidences, Alice sends TTP a request for recovery, which includes relevant evidences. TTP first checks the correctness of them, such as $h(k_n)$, and the status of protocols to decide whether to execute this request or not. If conditions are viable, the recovery protocol becomes true and prevents other requests. Then, message 2 and 3 are on the way. The sequence of message 2 and 3 can be inversed because of the same status. They contain evidences and the key (this key is used to prevent Alice launching a malicious request while without sending Bob message 2n+1). At this point, the information exchange successfully.

c) Abort Protocol

Message 1:

$$A \rightarrow TTP : f_{abort}, \ell, B, abort$$

if abort and recovery protocol is running, then stop

else abort protocol is true

Message 2:

$$TTP \rightarrow A : f_{Con_a}, A, B, \ell, Con_a$$

Message 3:

$$TTP \rightarrow B : f_{Con_a}, A, B, \ell, Con_a$$

When Alice fails to possess message 2 in main protocol, she can initiate abort protocol. The first message is a request issued by Alice aiming at aborting the information exchange with corresponding confirmations dispatched to both sides in last two messages. When the protocol ends, Alice and Bob achieve an item Con_a respectively that declaims the present exchange is abort and all information of the exchange, labeled by ℓ, are invalid. The protocol will also end in a fair way.

This sub-protocol could terminate the present exchange. If participants want to launch another completely irrelevant information exchange, it can be independent of the former and keeps non-memory. If Alice fells that Bob is dishonest and decides no longer to exchange information with him, the sub-protocol can be removed.

d) Error Protocol

Error protocol is initiated in abort protocol while Alice plans to launch cheat.

Message 1:

$$TTP \rightarrow A : f_{Con_e}, A, B, \ell, Con_e$$

Message 2:

$$TTP \rightarrow B : f_{Con_e}, A, B, \ell, Con_e$$

The running of error protocol means that Alice is cheating. Therefore, message 1 can be elided. Both sides achieve a confirmation of error Con_e about the protocol and aware of that the protocol has ended with Alice's cheat.

e) Some Other Questions

The protocols referred above depend on two rational assumptions. One is each participant prefers to protect his/her interests rather than to be harmed. Therefore, in main protocol, there is no need to launch recovery protocol for Bob to protect himself. If this happens, Bob will either be refused for wrong request, or gain what he already masters. And that benefits no one. The other one is communication channels. Because of resilient channels between TTP and each participant, the messages exchanged between them can be finally received which allows TTP to take part in protocols and participants to receive the evidences and messages. The recovery protocol and abort protocol are possible triggered by unreliable channel between two participants. Thus, TTP always sends messages to both sides during these two protocols. Error protocol is not the same for the doubtless deceive of Alice.

D. Properties Analysis

The properties of protocol consist of non-repudiation, fairness, timeliness and transparence. Non-repudiation can be realized through final evidences and we will introduce them when explain transparence. Timeliness is satisfied with timestamps included in each message. When limited time runs out, if receiver has received correct message, the protocol goes on. Otherwise, the receiver can stop the protocol or launch other protocol. In short, the process can be completed in a limited time. As for fairness and transparence, we will introduce them in detail follow.

a) *Fairness*

We need to analyze fairness message by message from main protocol to error protocol. In main protocol, after having sent Bob message 1, if Alice can accept message 2 correctly in limited time t_1, then the protocol goes on. Otherwise, if message 2 hasn't come to Alice or its content is incorrect, Alice will initiate abort protocol. Turn to abort protocol. In message 1, Alice sends TTP a request for abort, including the current label of the protocol ℓ. After TTP's validating, the confirmations of aborting the protocol are dispatched to both sides. Because of resilient channels, these messages can arrive at destinations eventually. Now, the protocol ends with nothing exchanged and stays fairness for Bob's unknown to the key. Considering another problem. If Alice still launches abort protocol after accepting message 2 correctly, she can possess the evidence of receipt EOR. However, the final evidence for Alice consists of EOR and $EOR_{k,n}$. That is, it is unavailable to proof something has happened. The protocol stays fairness.

Back to main protocol. From message 3 to message 2n, the protocol keeps fairness for no useful messages being exchanged. If Alice or Bob stops the message exchange, the other party will perceive this interrupt and stops reply to protect personal benefits. Both sides have no need to launch other protocols for already achieving a relative fairness.

The last two messages in main protocol exchange true key and produce two important evidences. If Alice launches recovery protocol before sending message 2n+1, she has to provide TTP message 1 in recovery protocol correctly. TTP is able to construct $EOO_{k,n}$ and $EOR_{k,n}$ by making use of EOO, EOR and $E_{TTP}(k_n)$. Further, TTP could assemble the final evidences NRO and NRR, which will be dispatched to both sides, respectively. The protocol ends with correct information exchange in a fair way. If Alice does not submit a correct request, that is, $h(k_n)$ is different from $h(D_{TTP}(E_{TTP}(k_n)))$, then error protocol sets to work at once. The confirmations of error are dispatched to each participant. The protocol remains fair without information exchange. If Bob dose not reply message 2n+2 after accepting message 2n+1 correctly, Alice will turn into recovery protocol. This is just like the former case.

Seeing from above analyses, we can find that no matter whether information is exchanged successfully or not, the protocol always stays fairness.

b) *Transparence*

Evidences present transparence. If we cannot detect whether TTP has involved in protocol by only looking at final evidences, we treat it as transparency. Thus, we will focus on evidences produced in two different conditions: the protocol has realized information exchange successfully without TTP's involvement and with the help of TTP's involvement.

The first case, namely, the information is exchanged with fairness under the condition of no dishonest participants or network problem, and produces evidences in an obvious way.

The non-repudiation evidence of origin generated by Alice:

$$NRO = \{EOO, EOO_{k,n}\}$$
$$= \{Sig_A(f_{EOO}, B, TTP, E_{TTP}(k_A), \ell, c), Sig_A(f_{EOO_{k,n}}, B, TTP, \ell, n, k_n)\}$$
$$= \{Sig_A(f_{NRO}, B, TTP, E_{TTP}(k_A), \ell, c, n, k_n)\}$$

The non-repudiation evidence of receipt generated by Bob:

$$NRR = \{EOR, EOR_{k,n}\}$$
$$= \{Sig_B(f_{EOR}, A, TTP, E_{TTP}(k_B), \ell, c), Sig_B(f_{EOR_{k,n}}, A, TTP, \ell, n, k_n)\}$$
$$= \{Sig_B(f_{NRR}, A, TTP, E_{TTP}(k_B), \ell, c, n, k_n)\}$$

The second case, namely, with the help of TTP, by means of all four protocols, the information exchange has completed and the protocol stays fairness. The evidences are generated in recovery protocol by TTP. After accepting the recovery request, TTP possesses $B, \ell, h(c), h(k_n), E_{TTP}(k_n), \mathrm{Re}\,c_A, Sub, E_{TTP}(n), EOO, EOR$ and begins to construct final evidences. The focus point is how to produce $EOO_{k,n}$ and $EOR_{k,n}$. We can get k_A and k_B with TTP's public key. These two private keys can be used to produce participants' signatures. Then, using $E_{TTP}(k_n)$ and $E_{TTP}(n)$, TTP could extract k_n ,which is validated by $h(k_n)$, and n. Thus, TTP is able to construct $EOO_{k,n}$ and $EOR_{k,n}$. Together with EOO and EOR, $EOO_{k,n}$ and $EOR_{k,n}$ could become final evidences NRO and NRR, respectively.

The non-repudiation evidence of receipt generated by TTP for Alice:

$$NRR = \{EOR, EOR_{k,n}\}$$
$$= \{Sig_B(f_{NRR}, A, TTP, E_{TTP}(k_B), \ell, c, n, k_n)\}$$

The non-repudiation evidence of origin generated by TTP for Bob:

$$NRO = \{EOO, EOO_{k,n}\}$$
$$= \{Sig_A(f_{NRO}, B, TTP, E_{TTP}(k_A), \ell, c, n, k_n)\}$$

These two kinds of final evidences generated in different circumstances are the same. After completing information exchange, it is impossible to decide whether TTP has involved in protocol. Thus, the TTP is transparent and the protocol stays transparency.

c) *Non-Repudiation*

The above two final evidences indicate that non-repudiation of origin and receipt comes true. If one participant decides to deny the exchanged information and calls for the revocation of exchange, the other one could submit his/her final evidence to judge to protect the benefits. Judge is independence of TTP; TTP has played a role to ensure the running of protocol while judge is a justice to maintain the fruits after the running of protocol. During the running of the protocol, if one party denies a sent message or has had accepted a message, the other one could submit corresponding origin/receipt evidence to proof it. For example, in main protocol, if Alice denies that she has sent Bob message 3, then, Bob could submit $EOO_{k,1}$ to judge to protect his interests; and if Bob denies that he has accepted message 3, Alice could submit $EOR_{k,1}$ to defense it.

By virtue of all evidences, including final evidences and evidences generated during the running of protocol, no one could launch repudiation. Thus, the protocol realizes non-repudiation.

d) Further Discussion

From the above analyses, we can ensure that the protocol has achieved fairness and transparence. Timeliness is reached by using timestamps in each message. Coming to non-repudiation, we should recur to all evidences. If repudiations happen, participants could present corresponding evidences to protect his/her own benefits.

Besides, the lower involvement of TTP referred above comes after contrastive results. Because the TTP's involvement in an optimistic non-repudiation protocol is lower than in an online or inline non-repudiation protocol, the protocol we proposed is better referring to the involvement of TTP. What's more, the protocol contains a random positive integer n, which makes TTP intervene protocol with a certain probability, $1/n$, when deception occurs in processes from message 3 to message 2n+2. In other words, TTP may not take part in protocol even though some mistakes come to protocol. This further reduces the possibility of TTP's involvement relative to other optimistic protocols that exclude random number referred in references.

V.Conclusion

We first give an introduction about non-repudiation protocol and define a series of properties. Then an idea of protocol named computable optimistic non-repudiation protocol is presented by which we complete the paper's protocol. The protocol consists of four parts and could ends with fairness no matter whether exchange is completed or not together with the loyalty of participants and the reliable of network. Besides, according to specific processes and evidences, we have given a detailed analysis of fairness, non-repudiation, timeliness and especially transparency. Evidences are the most important to protocol, and they refer more other mechanisms, such as digital signature, revocation of evidences and time efficiency for evidences. Besides, the properties can also be proved by formal methods, such as SVO logic. These questions need to deepen in further studies.

VI. Acknowledgment

This work is supported by the Major Research Project of the National Natural Science Foundation of China under Grant No.90818028. Natural Science Foundation Project of CQ CSTC: 2011BB2064.

References

[1] H. Pägnia, F. C. Gartner. On the impossibility of fair exchange without a trusted third party[R]. Technical Report TUD-BS-1999-02, Darmstadt University of Technology, Darmstadt, Germany. 1999.

[2] T. Tedrick, How to exchange half a bit, in: D. Chaum (Ed.), Advances in Cryptology: Proceedings of Crypto 83,Plenum Press, New York and London, 1984,1983, pp.147-151.

[3] T. Tedrick, Fair exchange of secrets, in: G. R. Blakley, D.C. Chaum (Eds.), Advances in Cryptology: Proceedings of Crypto 84, Vol. 196 of Lecture Notes in Computer Science, Springer - Verlag, 1985, pp. 434-438.

[4] M. Ben-Or, O. Goldreich, S. Micali, R. Rivest, A fair protocol for signing contracts, IEEE Transaction on Information Theory 36 (1) (1990) 40-46.

[5] O. Markowitch, Y. Roggeman, Probabilistic non-repudiation without trusted third party, in: Second Conference on Security in Communication Networks'99, Amalfi, Italy, 1999.

[6] Y. Han, Investigation of non-repudiation protocols, in: ACISP: Information Security and Privacy: Australasian Conference, Vol. 1172 of Lecture Notes in Computer Science, Springer-Verlag, 1996, pp.38-47.

[7] T. Coffey, P. Saidha, Non-repudiation with mandatory proof of receipt, ACMCCR: Computer Communication Review 26.

[8] A. Bahreman, J. D. Tygar, Certified electronic mail, in: Symposium on Network and Distributed Systems Security, Internet Society, 1994, pp.3-19.

[9] N. Zhang, Q. Shi, Achieving non-repudiation of receipt, The Computer Journal 39 (10) (1996) 844-853.

[10] J. Zhou, D. Gollmann, A fair non-repudiation protocol, in: IEEE Symposium on Security and Privacy, Research in Security and Privacy, IEEE Computer Society, Technical Committee on Security and Privacy, IEEE Computer Security Press, Oakland, CA, 1996, pp.55-61.

[11] S. Kremer, O. Markowitch, Optimistic non-repudiable information exchange, in: J. Biemond (Ed.), 21st Symp. on Information Theory in the Benelux, Werkgemeenschap Informatie- en Communicatietheorie, Enschede (NL), Wassenaar (NL), 2000, pp.139-146.

[12] J. Zhou, D. Gollmann, An efficient non-repudiation protocol, in: Proceedings of The 10th Computer Security Foundations Workshop, IEEE Computer Society Press, 1997, pp.126-132.

[13] J. Zhou, R. Deng, F. Bao, Evolution of fair non-repudiation with TTP, in: ACISP: Information Security and Privacy: Australasian Conference, Vol. 1587 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp.258-269.

[14] O. Markowitch, S. Kremer, An optimistic non-repudiation protocol with transparent trusted third party, in: Information Security Conference 2001 Lecture Notes in ComputerScience, Springer-Verlag, 2001.

[15] O. Markowitch, S. Saeednia, Optimistic fair-exchange with transparent signature recovery, in: 5th International Conference, Financial Cryptography 2001, Lecture Notes in Computer Science, Springer-Verlag,2001.

[16] Alaraj.A, Munro.M,An efficient fair exchange protocol that enforces the merchant to be honest.Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on 12-15 Nov.2007,page(s)196-202.

[17] Su Rui-dan,Fu Shao-feng,Zhou Li-hua, Achieving fair non-repudiation for web services transaction.Education Technology and Computer (ICETC), 2010 2nd International Conference on (Volume:5) 22-24 June 2010.

[18] Kremer S., Markowitch O., zhou J.. An intensive survey of

non-repudiation protocols. Computer Communications, 2002,25(17):1606~1621.