An Evenhanded Certified Email System for Contract Signing

Kenji Imamoto[1], Jianying Zhou[2], and Kouichi Sakurai[1]

[1] Information Science and Electrical Engineering, Kyushu University,
6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812-8581, Japan
imamoto@itslab.csce.kyushu-u.ac.jp, sakurai@csce.kyushu-u.ac.jp
[2] Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613
jyzhou@i2r.a-star.edu.sg

Abstract. Certified email is a system which enables a sender to prove a receiver's receipt of email. Such a system can be used for applications related to electronic commerce on the Internet. This paper considers a situation where a sender or a receiver wants to change his/her mind due to the change of mail content value (e.g., stock, auction, gambling) during the transaction. We point out that no traditional certified email systems have been designed for such a case, thus one of the participants can be at a disadvantage. To avoid this problem, we propose an evenhanded certified email system in which each participant can change his/her choice, either cancel or finish the transaction, at any time during the transaction.

1 Introduction

As the Internet has become more and more popular, many contracts are being signed online as well. A variant of the contract signing problem is *certified email* in which, Alice sends a mail to Bob and wants some evidence (i.e., a receipt) that Bob received her mail. *Fairness* is an important requirement for a certified email protocol that guarantees when the protocol terminates, either both parties have obtained their desired items, or neither acquired any useful information.

Many certified email systems have been proposed [1, 2, 3, 4, 5, 6], and some systems are commercialized. For efficiency, most of certified email systems include a *trusted third party* (TTP) as a mediator. This mediator is involved to ensure the fairness of a transaction. Although protocols without TTP were also proposed [7], they are not practical in terms of computation and communication overheads. Hence, this paper only focuses on certified email systems that use a TTP.

Some of existing systems introduced the concept of *timeliness* (i.e., any participant can terminate a session in finite time without loss of fairness) to avoid waiting the other's response forever [1, 2]. In order to provide timeliness, a system proposed in [1] has two sub-protocols: *cancel protocol* and *help protocol*. By using the cancel protocol, a sender can cancel a session if the intended receiver ignores the sender's request. On the other hand, by using the help protocol, a

S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, pp. 1–13, 2005.

receiver can obtain the mail even if the sender does nothing after the receiver responded to the sender's request.

In addition to termination of a session in finite time, the cancel protocol can also be used to change a sender's decision before completing the session (i.e., she can stop transmission of the mail). However, since the receiver cannot execute the cancel protocol, he cannot change his mind after a particular point even before completing the session (i.e., he cannot deny the receipt of the mail). This difference leads to the following disadvantageous situation.

> Suppose Alice sells gold to Bob at 100 dollars. The receipt of gold means that "Alice must sell gold at 100 dollars and Bob must pay 100 dollars". Then, they execute the proposed certified email protocol to exchange gold and the receipt. After Bob's response to Alice's request (but before completing the protocol), someone who says "I want to buy gold at 120 dollars" might appear. In this case, she will cancel the protocol, and sell gold to the new buyer. On the other hand, someone who says "I want to sell gold at 80 dollars" might appear. In this case, Bob wants to cancel the protocol and buy gold from the new seller, but he cannot.

The above situation can happen frequently in the case of a contract that the item's value is changeable, for example a soccer pool where the news of a player's sudden injury may change the betters' choices. Since no traditional certified email systems have been designed for such a case, one of the participants can be at a disadvantage. In this paper, we propose an *evenhanded system* in which each participant can change his/her choice anytime before a termination without loss of fairness. We call this property "*Change of Choice*". In our proposed system, each participant has sub-processes to cope with any event in order to enjoy the best benefit. Note that each participant always plays in a way that increases his/her own benefit, and a participant who first acts will obtain the better benefit (i.e., first come, first served).

2 Model and Requirements

This paper considers contracts of changeable values, where a party owning a variable item wants to sell her item to another party. Suppose the digital item is delivered with a certified email system, and the sender can claim the payment (as indicated in the receipt) from the receiver by proving the item has been received by the receiver. To simplify our analysis, we assume the price offered at the starting point of a session is the one negotiated by both participants.

Each party communicates through a network where no message is lost or delayed, and the value of an exchanged item can change anytime. Each party decides his/her action to make own benefit as high as possible.

A standard certified email system has the following requirements.

- *Fairness*: Both participants can either obtain the result each one desires, or neither of them does.

- *Authentication*: Each participant can identify his/her partner.
- *Non-repudiation*: Both participants cannot repudiate their own action after the session is over.
- *Timeliness*: Either participant can terminate a session at any time without loss of fairness.

Non-repudiation has two variants. *Non-repudiation of receipt* guarantees that a receiver cannot deny the receipt of mail after the receiver actually received it. *Non-repudiation of origin* guarantees that a sender cannot deny the transmission of mail if the sender actually sent it.

Besides the above standard requirements, there is an extra requirement of "*Change of Choice*" for an evenhanded certified email system as described earlier.

3 Game Tree and Evenhanded System

To concretely define an evenhanded system that can avoid any one-sided disadvantageous scenario as described in Section 1, we define evenhanded situation and stage by using game theory in an extensive form [1]. In this section, we first introduce the notion of game theory, and next define the evenhanded system.

3.1 System Expression Using Game Theory

Games, such as chess, can be represented by a labeled directed tree graph. Each vertex of this tree except for the leaves is labeled with the name of a player and represents a decision point for this player in the course of the game. The choices or possible moves at this point are represented by the edges starting from this vertex. The leaves of the tree correspond to possible ends of the game. Each leaf is labeled with a tuple of real numbers, which represent the payoffs for the players if the game ends in that leaf. The payoff may be negative, in which case it is interpreted as a loss. The starting point of the game is represented by the root of the tree. The goal of the players in a game is to maximize their payoffs.

There are some striking similarities between exchanges and games. Indeed, in both cases, we have two (or more) parties/players who interact with each other according to some rules, and whose actions influence the future actions of the other. From this reason, we try to express certified email systems by game tree using the following rules.

A system is advanced by one of the participants' available choices. Generally, each participant may have one or more choices as follows: (1) execute faithfully, (2) make the session valid with the TTP's help, (3) cancel the session, or (4) stay (i.e., do nothing). If both parties select the choice of stay, then the session is equivalent to being cancelled. A system consists of one or more "stages", which are periods between each choice (except for stay) decided by one of the participants. Moreover, each stage has three "situations" divided by events related to

[1] A similar work in [8] defines fairness by game tree.

the changes of the exchanged item's value, that is, higher/lower than the initial value, or no change. No party can predict these events beforehand.

Each participant can execute anytime one of any available choices at each stage. This means it is not decided which party selects the choice first at each stage if both parties have choices. To express such a chance, we add a chance move (either a sender selects first or a receiver selects first) at the first node of each stage.

In the case of contract with changeable values, each participant's payoff depends on the change of value. For example, if the value becomes higher than the initial value, since the sender of the exchanged item suffers a loss by sending it at the initial value, the choice of cancelling the session can make a higher profit for the sender than another choice. On the other hand, receiving the item at the initial value makes a higher profit for the receiver. Contrary to the above situation, if the value becomes lower than the initial value, sending the item at the initial value makes a higher profit for the sender while cancelling the session makes a higher profit for the receiver. Therefore, we can say that payoff for a party is 1 if the end of a session is desired for the party, and payoff is -1 if the end is not desired. However, completing a session by faithful execution can increase slightly both participants' payoffs $(0 < p_f < 1)$ [2].

Meaning of Each Object in Game Tree
In the following figures, black circle denotes a chance move, white circle denotes a sender's turn, and gray circle denotes a receiver's turn. Each edge represents available choice (i.e., F_i is ith procedure of the faithful execution, C is cancel, S is stay, H is TTP's help), and possible move (i.e., S_F means that the sender can select a choice before the receiver, and R_F means the contrary) at each point. Also each square denotes a termination of the session, at which payoffs for both participants are determined. In appendix A, we show an example system to explain how to express and analyze a system by game tree.

3.2 Definition of Evenhanded System

Roughly speaking, an evenhanded situation means a situation where both parties have a strategy to receive a good (positive) payoff. For example, in the situation of rise of the value, the sender should have the choice of cancelling the session, and the receiver should have the choice to make the session valid. Therefore, we define an evenhanded situation as a period in which each participant has a strategy to obtain his/her positive payoff. However, a choice of moving into the next stage cannot be seen as such strategies. On an evenhanded stage, period of stay does not cause any disadvantage for each participant. In other words, all situations on the evenhanded stage are evenhanded situations; or, if one of the situations is disadvantageous for a party, then another situation is advantageous for the party.

[2] This rule is introduced because we assume that, if nothing happens, each participant wants to send/receive a mail by a faithful way.

From the above definitions, we say a system is evenhanded for contract with changeable values if all of the stages in the system are evenhanded.

3.3 Analysis Example

Fig.1 is an example of a typical on-line certified email system expressed by game tree. This system can be divided into two stages, and Fig.2 shows three situations of the second stage. Each square in Fig.2 denotes a termination of the session, and numbers in a tuple under each square denote payoffs for the participants. (Left number is for the sender, and right one is for the receiver.) In Fig.2, v_i means the value of the exchanged item on ith stage, and v_1 is the initial value.

The faithful execution of the system is as follows. First, a sender sends an encrypted message (using the TTP's public key) and a hash value of the message to her intended receiver (denoted as procedure F_1). Next, if the receiver wants to read the message, he sends the encrypted message and a signature of the hash value to the TTP (denoted as procedure F_2). If the signature is valid, the TTP sends the decrypted message to the receiver and the receipt of the message to the sender, and this session can be completed faithfully. Also, the sender can perform a cancel protocol (C) anytime before a termination of the session.

On the second stage of the example system, the sender has choices of C and S while the receiver has choices of F_2 and S.

- In the case of $v_1 > v_2$, the sender's choice is S and the receiver's choice is S. This strategy results in payoffs of $(-1, 1)$.
- In the case of $v_1 = v_2$, the sender's choice is S and the receiver's choice is F_2. This strategy results in payoffs of (p_f, p_f).
- In the case of $v_1 < v_2$, the sender's choice is C and the receiver's choice is F_2. This strategy results in payoffs of either $(1, -1)$ or $(p_f - 1, p_f + 1)$.

From the above analysis, we can see that situations of "$v_1 < v_2$" and "$v_1 = v_2$" on the second stage are evenhanded because both parties have choices to gain a positive payoff. (In the case of $v_1 < v_2$, the payoffs depend on the result of

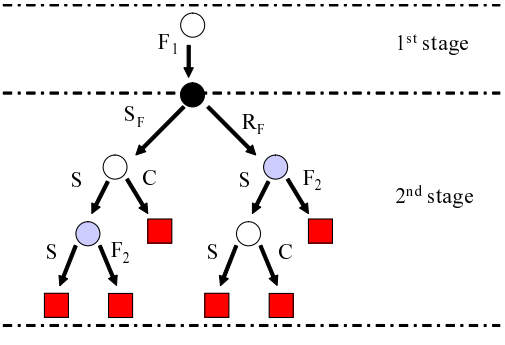

Fig. 1. Example System

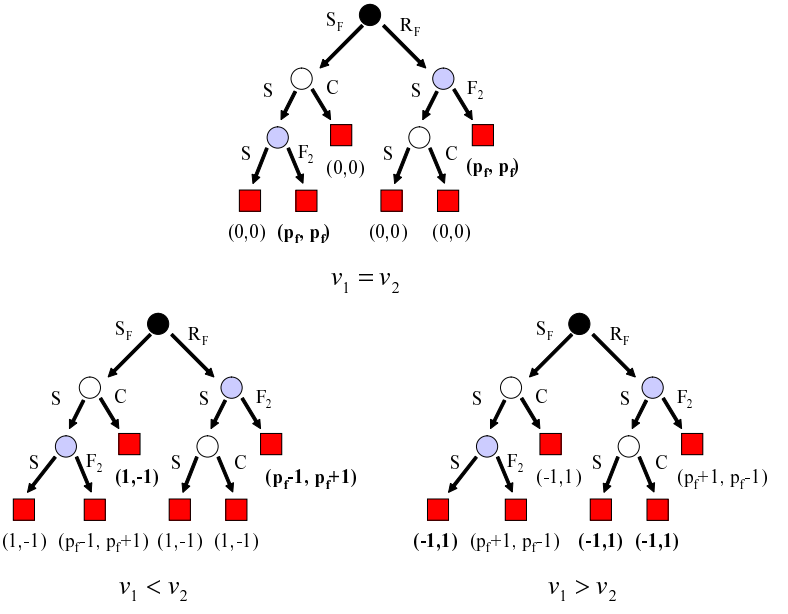

Fig. 2. Situations at 2nd Stage of Example System

the chance move). However, the situation of "$v_1 > v_2$" is advantageous for the receiver because the receiver can always receive a positive payoff if he always selects a rational choice (his payoff must be 1) but the sender cannot (her maximum payoff is -1). Therefore, neither the second stage nor the whole system is evenhanded.

4 Analysis of Existing Systems

Here we express two existing systems [3, 1] by game tree, and demonstrate that they are not evenhanded systems as defined in Section 3. Due to the page limit, we omit the figures of their game trees.

System of Ateniese et al. [3]

On the second stage, the sender has no choice but S while the receiver has choices of F_2, H and S.

- In the case of $v_1 > v_2$, the sender's choice is S and the receiver's choice is S. This strategy results in payoffs of $(-1, 1)$
- In the case of $v_1 < v_2$, the sender's choice is S and the receiver's choice is F_2. This strategy results in payoffs of $(p_f - 1, p_f + 1)$.

From the above analysis, both situations on the second stage are advantageous for the receiver. Therefore, this system is not evenhanded (advantageous for the receiver on the second stage).

System of Onieva et al. [1]

On the second stage, the sender has choices of C and S while the receiver has choices of F_2, H and S.

- In the case of $v_1 > v_2$, the sender's choice is S and the receiver's choice is S. This strategy results in payoffs of $(-1, 1)$.
- In the case of $v_1 < v_2$, the sender's choice is C and the receiver's choice is H. This strategy results in payoffs of $(1, -1)$ or $(-1, 1)$.

From the above analysis, the situation of $v_1 > v_2$ is advantageous for the receiver, and the situation of $v_1 < v_2$ is evenhanded. Therefore, this stage is not evenhanded (advantageous for the receiver in case of $v_1 > v_2$ on the second stage).

On the third stage, the sender has choices of F_3, C and S while the receiver has choices of H and S.

- In the case of $v_1 > v_3$, the sender's choice is F_3 and the receiver's choice is S. This strategy results in payoffs of $(p_f + 1, p_f - 1)$.
- In the case of $v_1 < v_3$, the sender's choice is C and the receiver's choice is H. This strategy results in payoffs of $(1, -1)$ or $(-1, 1)$.

From the above analysis, the situation of $v_1 > v_3$ is advantageous for the sender, and the situation of $v_1 < v_3$ is evenhanded. Therefore, this stage and whole system is not evenhanded (advantageous for the sender in case of $v_1 > v_3$ on the third stage).

In addition to the above analysis, no existing system, as long as we have investigated [2, 4, 5, 6], is evenhanded. Especially, in any optimistic systems, which use an off-line TTP [1, 2, 4, 5], the receiver is advantageous on the second stage where H choice is available, but the sender is advantageous on the following stage(s) because no system prepares the cancel choice for the receiver. Hence, on the second stage, the rational receiver always selects the choice of H or S to prevent his disadvantageous stage because there is no reason for the receiver to execute faithfully. (Compared with the risk of meeting disadvantageous stage, the benefit of p_f might be not very attractive for the receiver.) As a consequence, for contract with changeable values, rational party uses a non-evenhanded optimistic system in the same way as on-line systems, in which the TTP is always used.

5 An Evenhanded Certified Email System

In order to prevent disadvantageous stage, an evenhanded system needs to provide "choice of cancel", for both the sender and the receiver. The reason why no previous system using an optimistic protocol provides the receiver with the choice of cancel is that the TTP does not have a way to check whether the receiver actually received the mail or not. Cancelling a session without checking

the receiver's receipt can break fairness because the sender cannot make the receipt valid after cancelling the session while the receiver might have actually received the mail. To provide a way to check the receiver's receipt, we introduce a *bulletin board* on which anyone can check all posted entries. In this section, we propose an evenhanded certified email system and analyze it in terms of security and actual management. Our proposed system needs some assumptions as follows.

- The bulletin board stores and publishes all authorized entries, and maintains the order of the posted entries.
- The TTP performs nothing except the procedures set by the system.
- The channel between the TTP and the bulletin board is authentic.
- Any party has his/her own public key pair, and knows the other party's public key.

5.1 Proposed System

Our system consists of three protocols: *main protocol*, *help protocol*, and *cancel protocol*. Main protocol is used as a faithful execution to exchange the sender's variable item and the receiver's receipt, and the TTP does not appear. Help protocol can be initiated by the receiver before termination of a session, and used to make the session valid with the TTP's help. Both parties can initiate cancel protocol before termination of a session, and it is used to cancel the session. Table 1 outlines the notation used in the protocol description.

Main Protocol (Faithful Execution)

- Start message ($S \rightarrow R$): $S2R, S2T, H(K), EO$
- Response message ($R \rightarrow B$): $SID, S2R, S2T, H(K), EO, ER, MAC_P(S2R, S2T, H(K), EO, ER)$
- Finish message ($S \rightarrow B$): $SID, K, MAC_P(K)$

Before executing the main protocol, the sender shares SID (Session ID) and P (Password) with the bulletin board. Next, the sender randomly selects a session

Table 1. Notation

S, R, T, B	sender/ receiver/ TTP/ bulletin board
$H(X)$	a one-way cryptographic hash value of X
$PUB_X(Y)$	an encrypted message of Y by X's public key
$SIG_X(Y)$	a signature of Y by X's private key
$cleartext$	the name and price of the item
$S2R$	$PUB_R(SID, P, S, R, E_K(M), cleartext)$
$S2T$	$PUB_T(SID, S, R, K)$
EO	$SIG_S(SID, S2R, S2T, H(K))$
ER	$SIG_R(EO)$
$R2T$	$SIG_R(S2T, help)$

key K, and sends *Start message* to the intended receiver [3]. *cleartext* includes the description of the exchanged item ($=M$) and its initial value ($=v_1$).

After receiving the start message, the receiver verifies the signature of EO. If it is invalid, then abort. Otherwise, the receiver decrypts $S2R$ with his private key and obtains SID, P, and $E_K(M)$. If *Cancel message* has not been published on the bulletin board, the receiver sends *Response message* to the bulletin board. Then, the bulletin board checks the validity of $MAC_P(S2R, S2T, H(K), EO, ER)$ included in the posted message, and if it is valid, $S2R$, $S2T$, $H(K)$, EO, and ER are published on the board. Otherwise, the message is rejected.

Next, the sender verifies the signature of posted ER with the receiver's public key. If it is invalid, then abort. Otherwise, if neither *Finish message* nor *Help message* has been posted, the sender sends *Finish message* to the bulletin board. Then, the bulletin board checks the validity of $MAC_P(K)$ included in the posted message, and if it is valid, K is published on the board. Otherwise, the message is rejected.

Finally, the receiver can receive the message by decrypting $E_K(M)$ with the published K. On the other hand, the sender can insist on the receiver's receipt of M by showing the posted entries on the board.

Help Protocol

- Request message ($R \rightarrow T$): $R2T$
- Help message ($T \rightarrow B$): K

This protocol can be initiated by the receiver if neither valid *Finish message* nor *Cancel message* has been posted on the bulletin board. First, the receiver signs $S2T$ included in *Start message*, and sends *Request message* to the TTP.

Next, the TTP checks whether valid *Response message* has been posted on the bulletin board or not, and if it is published, the TTP decrypts $S2T$ with its private key, and publishes K included in it.

After all, the receiver can receive the message by decrypting $E_K(M)$ with the published K and the sender can insist on the receiver's receipt of M by showing the posted entries on the board.

Cancel Protocol

- Cancel message (S *or* $R \rightarrow B$): $SID, cancel, MAC_P(cancel)$

This protocol can be initiated by either the sender or the receiver if neither valid *Finish message* nor *Help message* has been posted on the bulletin board. The bulletin board checks the validity of $MAC_P(cancel)$ included in the posted message, and if it is valid, the cancel message is published on the board. Otherwise, the message is rejected. By completing this protocol, the session can be

[3] In the case that M is a big message to be exchanged, to improve the efficiency, a session key L may be introduced and $S2R$ can be re-defined as $S2R = E_L(E_K(M)), PUB_R(SID, P, S, R, L, cleartext)$.

cancelled. However, if valid *Finish message* or *Help message* has been posted before *Cancel message*, the cancel is invalid.

If a session is cancelled, the bulletin board stops receiving new entries of the session. Without this rule, an unfair situation can happen in a race condition: the sender sends *Finish message* to the bulletin board while the receiver sends *Cancel message* at the same time. In this case, the cancel becomes invalid but the receiver can get the message by decrypting $E_K(M)$ with the published K.

5.2 Analysis

Now, we conduct an informal analysis of the proposed system against the requirements introduced in Section 2. In the proposed system, the sender's authentication and non-repudiation of origin are available by the verification of EO. Similarly, the receiver's authentication and non-repudiation of receipt are available by the verification of ER. The receiver cannot obtain M without knowing K, and the sender cannot make the receipt valid without publishing K. Moreover, a receipt cannot be cancelled after publishing K. These properties result in fairness of the whole system. In addition, timeliness is available by using the help protocol or the cancel protocol.

Since our system uses a bulletin board, actual management of the board should be considered. We especially consider *denial of service* (DoS) attack against the bulletin board as an important problem. In our system, because only parties who know the correct pair of SID/P are allowed to post on the board, plenty of waste posting from DoS attackers can be prevented. Moreover, by introduction of a unique SID, the board can detect DoS attacks that resend previously transferred messages, and the receiver can also detect the replay attack which aims to make him believe the sender sends the message twice.

We also consider a case in which a wrong key K or $S2T$ different from the one included in *Start message* is posted. To confirm the correctness of K in *Finish message*, B will check whether the hash value of K is equal to $H(K)$ posted on the bulletin board. To confirm the correctness of $S2T$ in *Request message*, the TTP will check whether $S2T$ is equal to the one posted on the bulletin board. Moreover, the correctness of the posted $S2T$ and $H(K)$ can be confirmed by verification of EO which is also on the bulletin board.

If SID is generated at random, it is possible that the receiver uses this SID/P pair to initiate another session with another party. This problem can be solved by introducing both participants' identities into each SID. Other parties cannot use a SID with incorrect identities to post messages. As the sender (S) and the bulletin board (B) share SID and P before executing the main protocol, and SID and P will be different for each session, there should be an efficient mechanism to allow S and B to share SID and P before each session. Suppose S and B share a secret (P_{master}) in advance. Then, SID and P for each session can be derived as $SID = S||R||i$, and $P = H(S, R, i, P_{master})$, where $||$ means a concatenation and i is a time-stamp or a counter.

Compared with other systems' public-key operations (i.e., encryption, decryption, signature generation and verification), the computational complexity of a

normal execution in our system is low: the total numbers in $[1, 2, 3, 4, 5]$ and ours are 6, 6, 9, 10, 7 and 6, respectively.

Next, we show that the proposed system is evenhanded (see Fig.3).

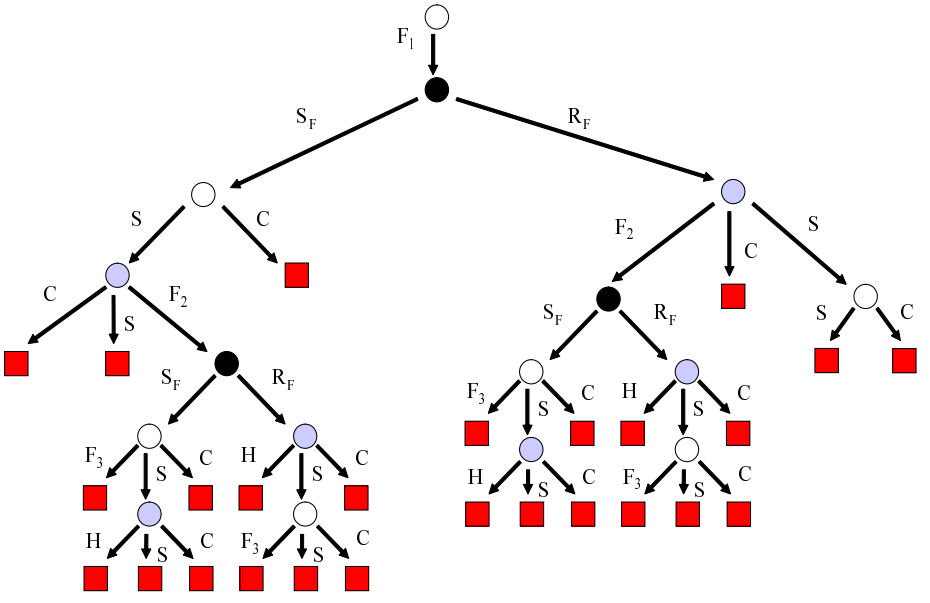

Fig. 3. Proposed System

- On the second stage, the sender has choices of C and S while the receiver has choices of F_2, C and S.
  - In the case of $v_1 > v_2$, the sender's choice is S and the receiver's choice is C. This strategy results in payoffs of $(-1, 1)$.
  - In the case of $v_1 < v_2$, the sender's choice is C and the receiver's choice is F_2. This strategy results in payoffs of $(1, -1)$ or it moves into the next stage.

  From the above analysis, the situation of $v_1 > v_2$ is advantageous for the receiver, and the situation of $v_1 < v_2$ is advantageous for the sender. Therefore, this stage is evenhanded.
- On the third stage, the sender has choices of F_3, C and S while the receiver has choices of H, C and S.
  - In the case of $v_1 > v_3$, the sender's choice is F_3 and the receiver's choice is C. This strategy results in payoffs of $(p_f + 1, p_f - 1)$ or $(-1, 1)$.
  - In the case of $v_1 < v_3$, the sender's choice is C and the receiver's choice is H. This strategy results in payoffs of $(1, -1)$ or $(-1, 1)$.

  From the above analysis, both situations at this stage are evenhanded.

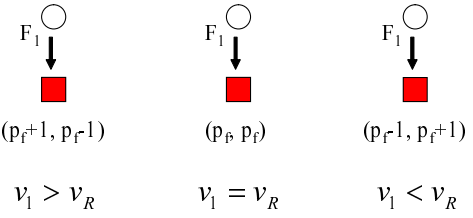

Fig. 4. Situations of Toy Example

Since all stages in this system are evenhanded, it is an evenhanded certified email system.

Note, it is easy to design an evenhanded fair exchange protocol using the bulletin board without TTP. For example, sending all messages on the bulletin board, the posted message is used as the receiver's receipt of the message. However, considering exchange of items with changeable values, the assumption about adequate price offered at the starting point is not realistic. Then the above protocol without TTP is not evenhanded (see appendix A). For this reason, a contract of changeable values needs both participants' operations to make the contract evenhanded even if the assumption does not exist. In this case, even if using the bulletin board, it is not straightforward to design an evenhanded fair exchange protocol without TTP. This is because if the proposed protocol does not use TTP, the receiver loses the way to decrypt the message without the sender's help (i.e., the help choice to TTP H is not available), and the whole system becomes un-evenhanded. To design an evenhanded fair exchange protocol without TTP is one of our future works.

6 Conclusion

This paper considered a situation where a sender or a receiver can change his/her mind anytime before termination of a session. To cope with such a situation, we defined a notion of an evenhanded system by game tree and showed no previous system is evenhanded. We further proposed an evenhanded certified email system by using a bulletin board.

As far as we know, no existing system is evenhanded. So, it would be interesting to investigate how other non-evenhanded protocols can be turned into evenhanded ones.

References

1. J. A. Onieva, J. Zhou, and J. Lopez, "Enhancing Certified Email Service for Timeliness and Multicasting", INC'04.
2. S. Kremer, and O. Markowitch, "Selective Receipt in Certified email", INDOCRYPT'01.

3. G. Ateniese, B. d. Medeiros, and M. T. Goodrich, "TRICERT: A Distributed Certified email Scheme", NDSS'01.
4. O. Markowitch and S. Kremer, "An Optimistic Non-repudiation Protocol with Transparent Trusted Third Party," ISC'01.
5. G. Ateniese and C. N. Rotaru, "Stateless-Recipient Certified Email System based on Verifiable Encryption," CT-RSA'02.
6. M. Abadi, N. Glew, B. Horne and B. Pinkas, "Certified Email with a Light On-line Trusted Third Party: Design and Implementation," WWW'02.
7. R. Markle, "Secure Communications over Insecure Channels", Communications of the ACM 21:294-299, 1978.
8. L. Buttyan, and J-P. Hubaux, "Toward a Formal Model of Fair Exchange - a Game Theoretic Approach", EPFL SSC Technical Report No. SSC/1999/039.

A Assumption About Adequate Initial Price Offered by the Sender

Under the model introduced in Section 2, we can easily design a simpler evenhanded system by using an on-line TTP. However, in such a system, if the assumption about adequate price offered at the starting point is not realized, some disadvantageous situation might happen.

Now we show a toy example which leads to a disadvantageous situation, and introduce requirements of contract systems against the problem. The procedure of the toy example is as follows. First, the sender sends $S, R, M, cleartext$, $SIG_S(S, R, M, cleartext)$ to the TTP. Then, the TTP sends M to the intended receiver, and its recept, $SIG_T(SIG_S(S, R, M, cleartext))$, to the sender. Because each session is completed soon after the initiation, the value of M does not change from the initial value v_1. Hence, this system is evenhanded, in which the payoffs are always (p_f, p_f).

This system can be seen as an evenhanded system only based on the assumption that v_1 offered by the sender is adequate. Otherwise, a problem will arise. Here, v_S (v_R) denotes the adequate price for the sender (the receiver). At this time, this system has three situations for the receiver: $v_1 > v_R$, $v_1 = v_R$, and $v_1 < v_R$. (The assumption means $v_1 = v_R$.) Regarding the case where the sender (receiver) sells (buys) the item at the lower (higher) price than the adequate price as a negative result, the payoffs in each situation are shown as in Fig.4 (suppose $v_S = v_R$).

In this system, the sender can decide v_1 and complete the exchange without the receiver's operation. As a natural result of this, a rational sender always decides v_1, where $v_1 > v_S = v_R$. This strategy results in the payoffs always being $(p_f + 1, p_f - 1)$. That is, in case the assumption $v_1 = v_R$ is not realized, the sender is advantageous in this system.

For this reason, in a contract of changeable values, any session requires both participants' operations to make the contract valid as our proposed system introduced in Section 5. In addition, the conditions of a contract should be explicit such as the usage of *cleartext* to make both participants agree on the contract. In a system with these properties, each participant can choose to cancel a session if the offered value is not adequate.