Undeniable Fair Exchange

Montri Apiromvorakarn[1] and Yongyuth Permpoontanalarp[2]
Logic and Security Laboratory, Department of Computing Engineering Department
King Mongkut's University of Technology Thonburi
126 Pracha-Utid Road, BangMod, Tung-Kru, Bangkok 10140
THAILAND

Abstract: - Fair exchange is an electronic data exchange protocol that allows both sender and receiver to exchange information such that either all of parties have the exchanged information or none of them do. All existing approaches to fair exchange achieve the fair exchange property between a single sender and a receiver. However, they are unable to provide the fair exchange for all senders. A malicious receiver can deny receiving messages from some legitimate senders but do receive messages from conspired senders. Such unfair exchange is devastating in many applications such as electronic auction where the conspired sender can be the bid winner easily. We propose a fair exchange protocol which offers a new property, namely undeniable fairness for all senders. We show that our protocol is practical and secure for the fair exchange system.

Key-Words: - Fair Exchange Protocol, Certified Email, Cryptographic Protocols and Internet Security.

1 Introduction

Nowadays, many internet-based applications involve the use of electronic exchange of information between senders and receivers, for example, email system, e-commerce system and electronic auctions. Normally, in such exchange, each party receives some information after sending out some other information. For example, in electronic auctions, after a bidder submits a bid to an auctioneer, the auctioneer will send back a bid receipt to the bidder.

However, if one party behaves dishonestly, then the exchange becomes unfair to the other party. For example, suppose that a malicious auctioneer receives a bid from a legitimate bidder but the auctioneer does not send back the bid receipt. The auctioneer would learn the bid information and can reveal it to conspired bidders who will then gain the advantage for the bidding. Also, the auctioneer can deny bids based on the bidders and amount, but receive bids from conspired bidders only. Thus, the conspired bidder can be the bid winner.

Fair exchange (eg. [1,2,3,4,5]) is an electronic data exchange protocol that allows both sender and receiver to exchange information such that either all of parties have the exchanged information or none of them do. Fair exchange has several applications, for example, certified email delivery and fair electronic auctions. In order to achieve the fair exchange, all approaches assume an anonymous communication system which hides the identity of the sender of a message from receivers. Also, all interesting approaches employ cryptography technique to encrypt messages when they are being exchanged. These two mechanisms can prevent a receiver from denying messages based on the senders and the bid content during the exchange.

There are three main approaches to fair exchange, depending on the use of a trusted third party (TTP). TTP is an additional infrastructure to mediate the exchange between a sender and a receiver. The first approach (eg. [1]) does not use TTP at all. Both sender and receiver exchange information bit by bit. If a party misbehaves, the party does not have much advantage over the other to cheat since the malicious party knows just one more bit than the other. However, this approach requires too many rounds of message exchange and it is not practical for business use due to the lack of a center to account for exchange expenses.

The second approach (eg. [2,3]) is to use an online TTP to mediate the exchange. This approach is suitable for commercially practical use, but it has an additional cost on TTP. The third approach (eg. [4,5]) is to use an offline TTP, and is also called optimistic fair exchange. TTP is involved only when there is a dispute between sender and receiver. Otherwise, the exchange takes place between sender and receiver only. However, this approach is not practical for business use. Moreover, most works in this approach with few exceptions [5] are too complicated to be used in practice due to the complex underlying cryptography.

All existing approaches to fair exchange achieve the fair exchange property between a single sender and a receiver. They are unable to provide the fair exchange for all senders.

In general, when there are multiple senders exchanging messages with a malicious receiver, the receiver can deny receiving messages from some senders but do receive messages from conspired senders. It is possible that the conspired senders intentionally provide information on the identity of the senders to the receiver in some way when their bids are submitted. Thus, receivers can selectively receive messages from conspired senders easily. To deny receiving messages from a sender, the receiver can just ignore to participate into an agreed message-exchange protocol with the sender. As a result, the exchange is not fair for all senders even though it is fair between each individual sender and the receiver.

In this paper, we propose a novel fair exchange protocol which solves the problem mentioned above. It ensures an undeniable fair exchange property. The undeniable fair exchange means in addition to the usual fair exchange that a receiver cannot refuse to receive messages sent by any sender. To the best of our knowledge, our protocol is the first which offers the undeniable fair exchange.

In section 2, we discuss some related work in order to point out the problem in the existing approaches. In section 3, we propose our new concept of undeniable fair exchange. Then, we present our protocol in details in section 4. We analyze our protocol in section 5 and discuss important issues in section 6.

2 Related Work

In the following, we discuss two important works: one for the offline TTP approach and the other for the online TTP approach.

We use the following notations throughout the paper. Let *S*, *R*, *m* mean sender, receiver and message that *S* wants to send to *R*, respectively. "$S \rightarrow R : M$" means that *S* sends *M* to *R*. $A_X(m)$ means public key encryption on message *m* by *X*'s public key, and $S_X(m)$ means *X*'s signature on message *m*. Also, *k(m)* stands for symmetric encryption of *m* by key *k*, and *h(m)* means hash of message *m*.

2.1 Micali's Approach [5]

Micali proposed a simple optimistic fair exchange which uses the generic construction of the underlying public key cryptography. The detail is as follows.

1) $S \rightarrow R: A_{TTP}(S, R, m)$ (called Z)
2) $R \rightarrow S: S_R(Z)$
3) $S \rightarrow R: m$

If *m* in step (3) does not match *m* in *Z*, then do steps 4, 5 and 6 as follows

4) $R \rightarrow TTP: Z, S_R(Z)$

After TTP validates *R*'s signature, do the following two steps.

5) $TTP \rightarrow S : S_R(Z)$, and
6) $TTP \rightarrow R : m$

Steps 4 – 6 are performed only when a dispute between *S* and *R* occurs. TTP can resolve the dispute caused by malicious S who does not issue the correct *M* at step (3).

However, if malicious *R* denies receiving message from legitimate *S* by not performing step (2), then no exchange between *S* and *R* will ever take place. *R* may provide an excuse later that *R* does not receive message in step (1).

2.2 Abadi et. al. 's Approach [3]

Abadi et. al. presents an interesting fair exchange which uses online TTP in a minimal way. In particular, TTP does not keep state information during the fair exchange. The following presents their simplified version.

1) $S \rightarrow R: k(m), A_{TTP}(k, S, R, h(k(m)))$
2) $R \rightarrow TTP: A_{TTP}(k, S, R, h(k(m))), h(k(m))$
3) $TTP \rightarrow R: h(k(m)), k$
4) $TTP \rightarrow S: S_{TTP}(A_{TTP}(k, S, R, h(k(m))))$

In step (2), *R* sends *h(k(m))* computed from *k(m)* in step (1). After step (2), TTP compares *h(k(m))* with *h(k(m))* in $A_{TTP}(k, S, R, h(k(m)))$ to verify that R gets the correct *k(m)* before issuing decryption key *k* in step (3). In step (4), TTP sends a receipt to *S* on behalf of *R*.

Again, malicious *R* can deny receiving message from legitimate *S* by not performing step 2. As a result, no exchange can be taken place.

3 Undeniable Fair Exchange

We provide the definition of undeniable fair exchange in the following.

Definition 1 Undeniable Fair Exchange
The undeniable fair exchange offered by our system means that
1. Either both a sender and a receiver have exchanged information or none of them do, and
2. A receiver cannot deny receiving messages sent by any sender by ignoring to participate into an agreed exchange protocol with the sender. □

The fair exchange property offered by all existing approaches is in the sense of (1) which is an exchange between one sender and one receiver. However, our new undeniable fair exchange property deals with the fair exchange between all senders and a receiver. Throughout the paper,

fairness in our system means the undeniable fairness.

The undeniable fairness provided by our system is not in the absolute sense. The absolute undeniable fair exchange system would be too strict since any receiver has to read messages as soon as they are sent to the receiver. In other words, a receiver has to read messages in the same order as they are sent. However, our system provides the undeniable fairness in a loose sense in that a receiver can read messages sent to her in some flexible and non-sequential order. However, finally the receiver will have to read all of them.

Our system uses TTP to provide the undeniable fair exchange. Before a sender can send a message to a receiver, the sender must submit a sending request to TTP. TTP provides for each receiver a reading queue of messages sent to the receiver by any sender. The messages in the reading queue are ordered according to the sending requests by the senders.

To ensure that a receiver cannot refuse to read messages sent by any sender, we impose a constraint on receiver's reading ability of messages in a reading queue at the receiver. The constraint is that if a receiver refuses to read some messages in the reading queue, then the receiver is not allowed to read many other subsequent messages in the queue too. Thus, after a receiver denies exchanging some messages with some senders, soon the receiver will not be able to exchange any further messages with any senders at all.

The constraint is implemented by a window of messages in a reading queue that are allowed to read by the receiver. If a message is in the window, then it can be read. Otherwise, it cannot. Also, messages in a window can be read in any order. The window can be slid in forward direction across only messages that have been already read. After window is slid, more messages can be read. If message at the leftmost of the window is not read, then the window cannot be slid. So, no more messages can be read. So the window mechanism provides a flexible order of message reading in that within a window there is no order, but there is an order, after a window is slid, between some messages in the slid windows.

The size of windows is important since it indicates the trade-off between the flexibility for message reading and the undeniable fairness. If the window is wide, then the receiver has great flexibility to read any messages in the window. But the undeniable fairness becomes weak since the receiver can still read many other messages while denying reading some message in the window. However, if the window is small, then the flexibility is decreased but the undeniable fairness is strong.

The size of the window should be adjustable dynamically as the exchange of messages proceeds. Conceptually, if the system achieves good result of the undeniable fairness, then the window should be wide to allow for flexible reading. But if the system does not get good result of the undeniable fairness, then the window should be small in order to regain the undeniable fairness in the future. We propose a method to compute the size of window dynamically in section 4.3.

4 Protocol for Undeniable Fair Exchange

Our protocol is based on Abadi et. al.'s approach [3]. The protocol consists of two parts, namely basic protocol and dispute resolution protocol. Basic protocol describes steps used for exchanging messages between sender and receiver. However, dispute resolution protocol is used when there is a dispute occurring between sender and receiver.

4.1 Basic Notations

1) $Q(R)$ = a unique running sequence number (called the message number) generated by TTP for each message that any sender requests to send to R
2) $Min\_Q(R)$ = the smallest message number of unread messages in R's reading queue
3) $W(R) = R$'s window
4) TS = the total number of sent messages by a sender to a receiver
5) TR = the total number of received messages by a receiver
6) $Ratio_{fair}$ (which is a system parameter) = TR/TS where $TR \leq TS$

4.2 Basic Assumptions

We assume that only TTP has a pair of private and public keys. Also, we assume that all senders and receivers know TTP's public key. Moreover, TTP is trusted on all aspects, but both senders and receivers are not trusted.

We assume a secure block cipher for symmetric encryption, for example AES in CBC mode. Also, the hash algorithm required is collision resistant, for instance, SHA-256. Public key encryption needed here must be secure against an adaptive chosen ciphertext attack [6], for example the RSA PKCS standard using OAEP [7]. We also require a signature scheme that is secure against non-existentially forgeable by an adaptive chosen message attack, for example the RSA PKCS standard using PSS [8].

4.3 Basic Protocol

The basic protocol contains six steps. Before a sender can send a message to a receiver, the sender must submit a sending request to TTP to obtain a ticket. The ticket contains necessary information for the receiver in order to read the message. Then, the sender sends both ticket and encrypted message to the receiver. In order to read the encrypted message, the receiver requests the key from TTP. After TTP issues the key to the receiver, TTP sends out the receipt of the message to the sender.

TTP provides for each receiver R a reading queue of messages sent to R by any sender. Messages in the reading queue are ordered according to their sending requests to TTP. When R requests to read a message in the queue, TTP computes a range of messages in the queue that are allowed to read by R.

1) Sender requests a ticket from TTP.

$S \rightarrow TTP : (S, R, h(k(m)))$

2) TTP issues a signed ticket to sender.

$TTP \rightarrow S : S_{TTP}(Q(R), h(k(m)))$

Intuitively, $Q(R)$ represents the message number for message m that S wants to send to R. TTP stores the last $Q(R)$ for each receiver and increments it each time a sender sends a request. The ticket consists of $Q(R)$ and $h(k(m))$.

3) The sender sends the signed ticket, encrypted message and encrypted key to the receiver.

$S \rightarrow R : S_{TTP}(Q(R), h(k(m))), k(m),$
$A_{TTP}(k, S, R, h(k(m)))$

R can verify the validity of TTP's signature in the ticket, and verify that the ticket is for the encrypted message $k(m)$.

4) The receiver requests from TTP the decryption key k to read message m.

$R \rightarrow TTP : S_{TTP}(Q(R), h(k(m))),$
$A_{TTP}(k, S, R, h(k(m))), h(k(m))$

Then, TTP verifies its own signature in the ticket and verifies that the ticket corresponds with the encrypted key. Also, TTP verifies that R received the correct $k(m)$.

5) TTP checks if R is currently allowed to read m.

If $Min\_Q(R) \leq Q(R) \leq Min\_Q(R)+W(R)$, then
$TTP \rightarrow R : h(k(m)), k$
Go to step 6)
Else
$TTP \rightarrow R : S_{TTP}(\text{“Request is refused”})$
Repeat step 4)

TTP allows R to read m if m is in R's window on messages in the reading queue. The range of readable messages in R's reading queue starts from the earliest unread message in the queue, called $Min\_Q(R)$, and the size of the range is defined by $W(R)$. In section 4.4, we discuss a method to compute $W(R)$.

6) TTP sends an evidential receipt of message m by R to the sender S.

$TTP \rightarrow S : S_{TTP}(A_{TTP}(k, S, R, h(k(m))))$

TTP certifies that TTP has released the decryption key for $A_{TTP}(k, S, R, h(k(m)))$. S can use this receipt with $k(m)$ to prove to a third party with the help of TTP that R has received m.

4.4 Calculation of Controlling Parameters

The following gives the definition of fairness for all senders in our system.

Definition 2 The system is fair for all senders iff for every receiver R and every sender S, $TR(S,R)/TS(S,R) \geq Ratio_{fair}$ where $TS(S,R)$ means the total number of sent messages by S to R and $TR(S,R)$ means the total number of read messages sent by S and read by R. □

Suppose that there are sender S and receiver R such that $TR(S,R)/TS(S,R) < Ratio_{fair}$. We call the receiver R who causes the unfairness by not receiving messages sufficiently as an unfair receiver and we call sender S as an unfairly treated sender.

Indeed, $Ratio_{fair}$ defines an acceptable degree of the undeniable fairness. If $Ratio_{fair}$ is large (TR is closed to TS), then it results in a strict kind of message reading in that many messages must be read by receivers soon after they are sent. Moreover, since those sent messages may come from many different senders, the amount of those messages tends to be large. Thus, it is unlikely that the receiver will be able to read all sent messages quickly. However, with a large $Ratio_{fair}$, the system provides a better level of fairness. We will discuss about this fairness later in this section.

Initially, we compute the window size based on the ratio and $TS_C(R)$, which is the total number of currently unread messages in R's reading queue. Later on, if the system becomes undeniably unfair, the window size is decreased. Then, the system will regain the undeniable fairness since unfair receivers will have less number of choices of messages to read. However, if the system regains the fairness, the window size will be increased to provide the flexibility for receivers again.

We provide two alternative methods to reduce the size of windows. The first method is to use a reduction of $TS_C(R)$ to compute R's window size. The reduction is to consider messages in $TS_C(R)$ that are sent from fairly treated senders only. Thus, the receiver will have the flexible message reading for fairly treated senders only. The second method is to reduce the window size according to the degree of the unfairness occurred.

We show an algorithm for computing the size of windows in the following. The algorithm is involved in step (5) of the basic protocol.

Algorithm Window Computation for *R*
Input: *TR(S,R)* and *TS(S,R)* for all senders *S*
Output: *W(R)* : positive integer
If the system is fair for all senders, then

$$W(R) = \lceil Ratio_{fair} \times TS_C(R) \rceil$$

Else choose one the following methods:
a) 1st method of window reduction
If $TS_{CF}(R) > 0$ then

$$W(R) = \lceil Raio_{fair} \times TS_{CF}(R) \rceil$$

Else $W(R) = 1$
Where $TS_{CF}(R)$ is the total number of currently unread messages in *R*'s reading queue that have been sent by fairly treated senders.
b) 2nd method of window reduction
If $UF\%(R) < T(UF\%)$ then

$$W'(R) = \left\lceil W'(R) - (W'(R) \times \frac{UF\%(R)}{T(UF\%)}) \right\rceil$$

Else $W(R) = 1$
Where $W'(R) = \lceil Ratio_{fair} \times TS_C(R) \rceil$, $TS_C(R) = \sum_S TS(S,R) - TR(S,R)$ which is the total number of currently unread messages in *R*'s reading queue, $S_{UF}(R)$ is the set of all unfairly treated senders caused by receiver *R*, *T(UF%)* is the threshold of the degree of the unfairness by a receiver in percentage and *UF%(R)* is the current degree of the unfairness by *R* in percentage which is defined by

$$\text{Let } n = \frac{\sum_{j1 \in S_{UF}(R)} TR(j1,R)}{\sum_{j2 \in S_{UF}(R)} TS(j2,R)} \text{ and}$$

$$UF\%(R) = 100 - (\frac{n}{Ratio_{fair}} \times 100\%) \quad \square$$

The degree of the unfairness *UF%(R)* is calculated by comparing *TR(S,R)/TS(S,R)* for all unfairly treated senders *S* with $Ratio_{fair}$. In general, the window size is reduced according to the closeness of *UF%(R)* to *T(UF%)*. The closer the degree of the unfairness to the threshold, the more reduction the window size is.

Conceptually, a large $Ratio_{fair}$ means a high requirement for fairness due to the large *TR*. Let consider the second method of window reduction. When the system is unfair for all senders, with larger $Ratio_{fair}$ the window size should be reduced faster to limit the unfairness since the system is more sensitive to the unfairness. As a result of the fast window reduction, the severity of the unfairness is low since the amount of messages that can be read by an unfair receiver during the unfairness is small.

When the fairness occurs, a larger $Ratio_{fair}$ produces a larger *UF%(R)* but also gives a larger *W'(R)* due to the ratio itself and a larger range of $TS_C(R)$. Thus, to get faster window reduction for larger $Ratio_{fair}$, we need to set a smaller threshold *T(UF%)*. This is intuitive since a larger $Ratio_{fair}$ should tolerate the unfairness at a lower degree. However, if we set the same value of *T(UF%)* for all ratios $Ratio_{fair}$, then with the same $TS_C(R)$, the window reduction will be the same for all ratios.

While the first method of reduction is simpler, the second method would reduce windows faster.

4.5 Dispute Resolution Protocol

Dispute resolution protocol is used when there is a dispute between sender and receiver. For example, *R* may claim that she does not receive a message in step (3) of the basic protocol for message number *Q(R)*. As a result, *R*'s window cannot be slid over *Q(R)*. There may be some malicious sender who requests for tickets but does not send out messages to *R*. In such case, *R* may engage in a dispute resolution protocol to provide a pre-committed receipt of message number *Q(R)*.
7) R→TTP : $S_R(R, Q(R))$
8) TTP → R : ok
9) TTP → S : "Q(R) is pre-committed"

After (7), TTP will treat message number *Q(R)* as if it has been read by *R*. And *S* can resolve the dispute by contacting TTP. TTP will act as a mediator for the actual message exchange.

Note that here we show the use of receiver's private key, but such key is not necessary. Disputed receivers can use other message authentication methods, for example MAC or passwords.

5 Analysis

Figure 1 shows that the size of windows is in proportion to both $Ratio_{fair}$ and $TS_C(R)$. The $Ratio_{fair}$ axis shows all of its values and the $TS_C(R)$ axis ranges from 0 to 100 messages. If $Ratio_{fair}$ is large, the window size is large also to allow a receiver to read many messages to get the fairness. For example, with $Ratio_{fair}$=0.2, when $TS_C(R)$ is 80, *W(R)* is 16. But with $Ratio_{fair}$=0.7, when $TS_C(R)$ is 80, *W(R)* becomes 56.

Figure 2 shows that in the first method of reduction, the size of window is reduced in proportion to the reduction of $TS_{CF}(R)$. The $\%TS_{CF}(R)$ axis shows the reduction of $TS_{CF}(R)$ in all percentages.

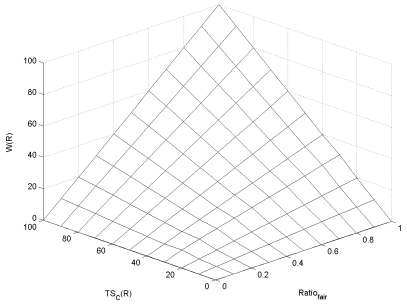

Fig.1 Window computation

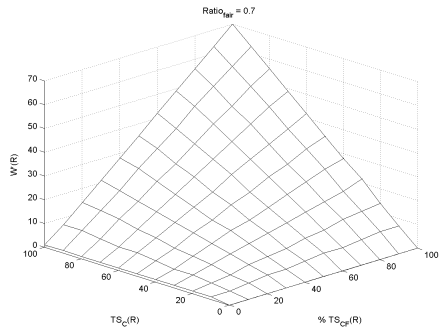

Fig.2 Window reduction (1st method)

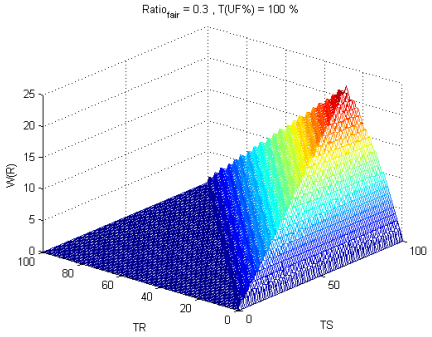

Fig.3 Window Adjustment ($Ratio_{fair}$=0.3)

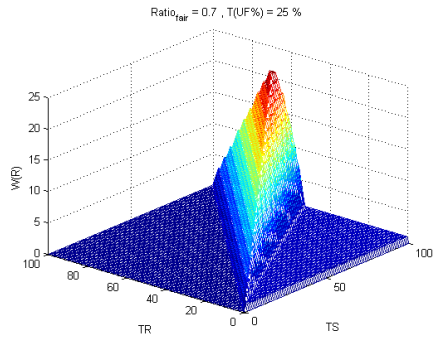

Fig.4 Window Adjustment ($Ratio_{fair}$=0.7)

Figures 3 and 4 show the second method of window reduction for the two ratios where *T(UF%)* is 100 and 25, respectively. The slope on the right shows the window reduction. When the unfairness occurs, a larger $Ratio_{fair}$ gives faster window reduction. Consider *TS(S,R)*=100. In figure 3, the window starts to reduce at *TR(S,R)*=29 and it gets to 1 at *TR(S,R)*=0. But in figure 4, the window starts to reduce at *TR(S,R)*=69 and it gets to 1 at *TR(S,R)*=53. Clearly, the slope of the window reduction in the latter is steeper than that in the former. Thus, the severity of the unfairness is low for a large $Ratio_{fair}$.

6 Discussion

The security of our system in terms of the undeniable fairness is justified by the ratio $Ratio_{fair}$ since the ratio defines the concepts and behaviors of the fairness naturally.

In our system, TTP has an additional cost to store some state information, for example, *Min_Q(R)*, *TR(S,R)*, *TS(S,R)* for each pair of sender and receiver. However, the amount of this information is constant for each pair and thus negligible. Also, this state information can be used for accounting expenses for users in the system. Thus, our approach is practical for business use.

7 Conclusion

In this paper, we present a new fair exchange protocol which offers a new property, namely the undeniable fairness for all senders. The undeniable fairness is achieved by using the ratio $Ratio_{fair}$ and the window mechanism to constraint message receiving. We have shown an analysis of our system and discussed the security and the practicality of it. As a future work, we aim to develop a prototype of this system and apply it to case studies.

References:

[1] S. Even, O. Goldreich and A. Lempel, A randomized protocol for signing contracts, *Communications of the ACM*, 28(6), 1985, 637-647.

[2] R.H. Deng, L. Gong, A.A.Lazar and W. Wang, Practical protocols for certified electronic mail, *J. of Network and Systems Management*, 3(4), 1996, 279-297.

[3] M. Abadi, N. Glew, B. Horne and B. Pinkas, Certified Email with a Light On-line Trusted Third Party: Design and Implementation, In Proc. of World Wide Web Conference, ACM, 2002, 387-395.

[4] N. Asokan, V. Shoup and M. Waidner, Optimistic Fair Exchange of Digital Signatures, *IEEE Journal of Selected Areas in Communications*, 18(4), 2000, 591-606.

[5] S. Micali, Simple and Fast Optimistic Protocols for Fair Electronic Exchange, In Proc. of ACM symposium of Principles of Distributed Computing, ACM, 2003, 12-19.

[6] D. Dolev, C. Dwork and M. Naor, Non-malleable cryptography, In 23rd Symposium on Theory of Computing, ACM, 1991, 542-552.

[7] V. Shoup, OAEP reconsidered, CRYPTO'01, Springer-Verlag, 2001, 239-259.

[8] M. Bellare and P. Rogaway, The exact security of digital signatures – how to sign with RSA and Rabin, EUROCRYPT'96, Springer-Verlag, 1996, 399-416.