SCIS 2002 The 2002 Symposium on Cryptography and Information Security
Shirahama,Japan, Jan.29-Feb.1,2002
The Institute of Electronics,
Information and Communication Engineers

Fair Exchange of Digital Signatures Using Conditional Signature

Byoungcheon Lee *
sultan@icu.ac.kr

Kwangjo Kim *
kkj@icu.ac.kr

Abstract— To implement fair exchange of digital signatures, we need an efficient scheme to commit a digital signature safely to a specific receiver. To satisfy this security requirement in an efficient manner, we introduce a new variant of digital signature called *conditional signature* which is a specially interpreted signature on a message and a condition together. By imposing a signer-chosen condition which describes expected action of a specified receiver, conditional signature can be used as a private negotiation statement in two-party communication. We model negotiation problem using conditional signature and then construct a fair exchange protocol. We show that matching negotiation and real exchange give a fair exchange of digital signatures.

Keywords: Fair exchange, negotiation, conditional signature, conditional commitment, matching negotiation

1 Introduction

Because of the rapid growth of electronic commerce over the Internet, the fair exchange problem is of more and more importance. Let's consider an electronic commerce scenario that a customer A wants to buy a flight ticket from a shop B. A is willing to give an electronic check to B in exchange for an electronic flight ticket, but the exchange protocol should be guaranteed to be fair. Because both the electronic check and the flight ticket can be implemented using digital signatures, this is the problem of fair exchange of digital signatures. An exchange is considered to be fair if either each player receives the expected item from the other party or neither player receives any useful information about the other's item.

In this paper we consider fair exchange of digital signatures in which two players try to exchange digital signatures (valuable items) in a fair way. Each message itself can be typical and not secret information, but its valid signature is a valuable item. To keep the privacy of exchange, two valuable items should not be linked each other such that each item can be used independently later.

1.1 Approaches to Fair Exchange Problem

There have been extensive researches on the fair exchange problem. Here we review them briefly.

In gradual exchange protocols [BGMR90, Dam93], two players exchange signatures on given messages and some secret information together, and then they gradually disclose their secret information to other players in many steps in a verifiable way. The exchanged signature is valid only when the secret information is presented together with the message.

In online TTP (trusted third party) protocols [FR97], an online TTP acts as a mediator in every transactions of exchange. Both players send their items to TTP, and then TTP verifies the correctness of both items and forwards them to other players. This is a rather straightforward approach, but TTP can be a bottleneck in overall performance.

In optimistic off-line TTP protocols [ZDB00, ASW00], an off-line TTP is used only in an optimistic way. In normal transactions TTP does not need to be involved in the exchange protocol at all, but if one player attempts to cheat or simply clashes, TTP can resolve the argument. In this approach secure information is encrypted with TTP's public key and committed to the receiver. Specially [ASW00] uses a noble cryptographic primitive called verifiable escrow such that the receiver can be convinced of the fact that the ciphertext was really encrypted with TTP's public key. But the verifiable escrow is a very expensive operation because it uses zero-knowledge proofs in a cut-and-choose way.

In accountable fair exchange protocols [ASW00], the concept of pre-contract, a special interpretation of digital signature, was introduced. Two parties firstly exchange pre-contracts and then exchange real contracts. If any argument happens, TTP can resolve it by providing abort token or an alternative contract. It is accountable in the sense that TTP's misbehavior can be proven.

In abuse-free optimistic TTP protocol [GJM99], no party ever can prove to a third party that he is capable of choosing whether to validate or invalidate the contract.

In this paper we are interested in providing efficient protocols for fair exchange of digital signatures between two players without using any expensive cryptographic primitives such as verifiable escrow.

* International Research center for Information Security (IRIS), Information and Communications Univ. (ICU), 58-4 Hwaam-dong, Yusong-gu, Daejeon, 305-732, Korea

1.2 Our Approach

In the scenario of the fair exchange of digital signatures, the message itself is not secret information and can be exposed to other participants, but its valid signature is a valuable item. For the purpose of committing a digital signature, verifiable escrow is very expensive and the proposal of pre-contract seems to be more reasonable and efficient.

We observe that fair exchange protocol can be modeled as two subprotocols, negotiation and exchange protocols. Negotiation is common experience in our everyday life and is an important issue for electronic commerce. We introduce the concept of conditional signature as a new variant of digital signature in which the signer can commit his signature safely and efficiently under a condition of his choice. Then we implement the negotiation problem using conditional signature. Finally we show that two matching negotiations and real exchange of digital signatures according to negotiations can give fair exchange of digital signatures.

This paper is organized as follows: The concept of conditional signature is introduced in Section 2 and negotiation protocol is modeled using conditional signature in Section 3. In Section 4 we propose a new fair exchange protocol which is a combination of matching negotiation and real exchange. We compare fair exchange protocols in Section 5 and finally conclude in Section 6.

2 Conditional Signatures

2.1 Negotiation in Distributed Environment

Negotiation is common experience in our everyday life. Typical negotiation scenario between an initiator A and a responder B can be as follows: The initiator A first proposes, "If you give me something (condition), I will give you something (promise)". If the responder B accepts A's proposal, then negotiation is finished and real exchange can occur. To accept A's proposal, B needs to give an acceptance message to A which satisfies A's condition and include B's promise. Then A and B can exchange valuable items which correspond to their promises. Note that the overall protocol can be clearly divided into the negotiation stage and the exchange stage. Negotiation can be considered as a kind of commitment for real exchange.

To implement the negotiation statement in electronic form, a digital signature should include both a promise and a condition. If a general signature scheme is used for negotiation, the message should be prepared very carefully to include the promise and the condition together, which will result a delicate sentence including "if – then"statement. Possibly, the interpretation of the message can be different according to entities.

We need to consider how the delicate negotiation statement can be constructed with condition and promise together. To avoid any confusion in negotiation process, we propose to use a specially interpreted digital signature variant called conditional signature.

2.2 Definition of Conditional Signature

Conditional signature is a new variant of digital signature in which a signer can commit his signature safely and efficiently to a specific receiver under a condition of his choice.

Definition 2.1 (Conditional signature) *Let A be a signer and B be a specified receiver (verifier). Let $m \in \mathcal{M}$ be a promise message and $c \in \mathcal{C}$ be a condition chosen by A. Conditional signature scheme consists of a signing algorithm and a verification algorithm.*

- *Signing algorithm CS takes m, c and the private key of the signer A as input and outputs a conditional signature σ_{AB}.*

$$\sigma_{AB} = CS_A(A, B, T, m; c).$$

- *Verification algorithm CV takes σ_{AB}, m, c and the public key of the signer A as input and outputs binary value, accept or reject.*

$$CV_{AB}(A, B, T, \sigma_{AB}, m; c) \stackrel{?}{=} true.$$

The interpretation of σ_{AB} is that the signer A promises to give a regular signature on m to the receiver B if B also promises to give a regular signature for a message which conforms to c.

The conditional signature can be used for the signer to commit a regular signature on m under a condition c. It is opened in a latter stage such that the signer gives the promised regular signature. A commitment scheme should satisfy secrecy and unambiguity. We show that conditional signature is a commitment scheme for a regular signature.

Theorem 2.2 *If the conditional signature scheme is a secure signature scheme, $\sigma_{AB} = CS_A(A, B, T, m; c)$ is A's commitment for his regular signature $s_A = S_A(m)$ under a condition c.*

Proof: To be a commitment scheme, the conditional signature scheme should satisfy secrecy and unambiguity. Let's assume that the conditional signature scheme is a secure signature scheme.

1. Secrecy: At the end of the commitment stage, the receiver B gets σ_{AB} with m and c together. But he cannot get any partial information on s_A, because $CS()$ is a secure signature scheme. Knowing σ_{AB} is of no help for the receiver to get s_A.
2. Unambiguity: Given the commitment σ_{AB}, the signer A is responsible to give a valid regular signature s_A on message m if the receiver B also commits a regular signature on a message which conforms to c.

Therefore conditional signature σ_{AB} is a commitment for a regular signature s_A. □

Conditional signature σ_{AB} represents A's commitment of a regular signature on message m to B. He is responsible for his promise only when B also commits a signature on a message which confirms to c which was specified in σ_{AB}. To remove any possibility of argument, message and condition should be stated explicitly. If any argument occurs, it is the responsibility of the signer. Message should be specific, but condition can be a set of values.

Since a conditional signature is a private negotiation statement between A and B, other third parties except T will not accept it as a valid signature because it is not their business. If there is an argument, T can participate in the protocol to resolve it.

2.3 Secure Implementation of Conditional Signature

Basically conditional signature is defined as a signature on a message m and a condition c together. If we use a specific syntax such that m and c are distinguished explicitly by anyone, we can implement the conditional signature just signing $m||c$ with general signature schemes. But to remove any possibility of confusion or argument, the conditional signature scheme should be distinguished from general signature schemes. But if we use a specially designed signature scheme, many social infrastructure such as banks or shops should be changed. The best solution is using general signature schemes in a distinguished way.

For this purpose we propose to use a conditional signature scheme in which the signer signs on $m||h(c)$ using general signature schemes. Anyone can differentiate it as a conditional signature on m under c rather than a general signature on $m||c$. Therefore the conditional signature is computed as

$$\sigma_{AB} = CS_A(A, B, T, m; c) \equiv S_A(A, B, T, m, h(c))$$

where $S()$ is a general signature scheme which is existentially unforgeable against adaptively chosen message attacks [PS00]. Then the security of the conditional signature scheme can be reduced to that of the based general signature scheme. Knowing a conditional signature $\sigma_{AB} = S_A(A, B, T, m, h(c))$ is of no help for getting a regular signature $s_A = S_A(m)$.

3 Negotiation Protocol using Conditional Signature

Consider an electronic commerce scenario that a customer A wants to buy a flight ticket from a shop B. A will have her specification for the flight ticket, for example, she wants to travel from Seoul to Jeju island on 10th of this month, but the departure time can be flexible for her. She expects (or knows) that it will cost about 100,000 won and hopes that the shop proposes an available flight schedule. Then A can send a negotiation proposal to B as an initiator as

(1) Proposal : $\sigma_{AB} = CS_A(A, B, T, m_A; c_A)$ with
m_A="100,000 won",
c_A="Flight ticket from Seoul to Jeju on 10th of this month".

Here A, B, T represents that A is the initiator, B is the responder, and T is the TTP. Upon receiving A's proposal, B can decide whether to accept it or give another proposal. If B sends another independent proposal which does not match with A's proposal, he will be an initiator of another negotiation. If B accepts A's proposal, he has to commit an acceptance message with a specified message m_B which confirms to c_A and a condition c_B which equals to m_A.

(2) Acceptance : $\sigma_{BA} = CS_B(A, B, T, m_B; c_B)$ with
m_B="Description of a conforming flight ticket",
$c_B = m_A$="100,000 won".

Two negotiation messages, σ_{AB} and σ_{BA}, with $m_B \in \{c_A\}$ and $m_A = c_B$, is called a matching negotiation. If a matching negotiation is obtained, A and B have the responsibility to keep their promises. If there is an argument, T can solve it by checking the match of two negotiation messages.

Definition 3.1 (Matching negotiation) *Let A and B be the initiator and the responder of the negotiation protocol. Two negotiation messages σ_{AB} and σ_{BA} defined above are a matching negotiation if $m_B \in \{c_A\}$ and $m_A = c_B$ hold.*

4 Fair Exchange of Digital Signatures

If we use the conditional signature scheme and the negotiation protocol, we can construct an optimistic fair exchange protocol in very efficient way.

4.1 Model

Assume that A has a message m_A and B has a message m_B. They wants to exchange regular signatures $s_A = S_A(m_A)$ and $s_B = S_B(m_B)$ on these messages in a fair way. Participants of the fair exchange protocol is as follows:

1. Initiator (A): She initiates the fair exchange protocol by giving a negotiation proposal with a conditional signature σ_{AB} on m_A under a condition c_A. If she does not receive any response from B, she can invoke the Abort subprotocol. If she has an argument with the responder B, she can invoke the A-resolve subprotocol.

2. Responder (B): He receives the conditional signature σ_{AB} from A and checks its validity. If he accepts A's proposal, he provides an acceptance message with a conditional signature σ_{BA} on m_B under a condition c_B. To be a matching negotiation $m_B \in \{c_A\}$ and $c_B = m_A$ should hold. If he has an argument with A, he can invoke the B-resolve subprotocol. Note that B does not have Abort subprotocol.

3. Trusted third party (T): He is a trusted and authorized entity to resolve any argument between participants. He has to maintain a secure database DB. He is not involved in the exchange protocol, but if there is an argument between A and B, he can resolve it by providing an alternative regular signature or an Abort token.

In this model we consider an asynchronous network, *i.e.*, we do not assume the existence of synchronous clock. Communication messages can be delayed by arbitrary but finite amount of time. It is assumed that each player can eventually reach the TTP. Both players can force a timely and fair termination of the exchange protocol by contacting T without the cooperation of the other player.

Security requirements for fair exchange protocol can be listed as follows:

1. *Completeness*: If two players behave correctly, they will receive the expected items without any involvement of the TTP. In other words, if neither player is corrupt and no message is lost, then the exchange will be successful eventually.

2. *Fairness*: After completion of the exchange protocol or at any moment during the protocol, either each player receives the expected item or neither player receives any useful information about the other's item. In other words, it is infeasible for an adversary to get honest player's signature without the honest player getting adversary's signature.

3. *Timeliness*: At any time during a protocol run, each player can unilaterally choose to terminate the protocol without losing fairness.

4. *Accountability*: If the TTP misbehaves resulting in the loss of fairness for a player, the victim can prove the fact in a dispute.

5. *Independence*: The exchanged valuable items should be independent each other such that each valuable item can be used alone without exposing the other item.

4.2 Fair Exchange Protocol

An initiator A has a message m_A and a responder B has a message m_B. They wants to exchange their signatures on their messages in a fair way. First, they commit their signatures using the conditional signature in the negotiation stage. Conditional signature is considered as a private negotiation statement and will not be accepted by irrelevant third parties. If their negotiation is matching, they exchange the promised real signatures. If any argument occurs in the middle, TTP resolves it by issuing an alternative regular signature or an abort token. The fair exchange protocol is a 4-pass optimistic protocol as follows.

Exchange protocol

1. Proposal $(A \rightarrow B) : \sigma_{AB} = CS_A(A, B, T, m_A; c_A)$.

   A prepares her message m_A and condition c_A. Then she commits her proposal with a conditional signature σ_{AB}.

2. Acceptance $(A \leftarrow B) : \sigma_{BA} = CS_B(A, B, T, m_B; c_B)$.

   B checks A's proposal σ_{AB}. If he accepts, he prepares matching message $m_B \in \{c_A\}$ and condition $c_B = m_A$ and then commits his acceptance with a conditional signature σ_{BA}. If B does not want to continue the exchange, he can quit.

3. Exchange $(A \rightarrow B) : s_A = S_A(m_A)$

   A checks σ_{BA} whether $m_B \in \{c_A\}$ and $c_B = m_A$ hold. If B has accepted her proposal, she sends $s_A = S_A(m_A)$ to B. If she does not get any response from B or σ_{BA} is not an acceptance, she invokes the Abort subprotocol.

4. Exchange $(A \leftarrow B) : s_B = S_B(m_B)$

   When B receives s_A, he checks whether it is a valid signature for m_A which was committed in the proposal step. If he accepts s_A as valid, he sends $s_B = S_B(m_B)$ to A. If he does not get any response from A or s_A is not valid, he invokes the B-resolve subprotocol.

5. When A receives s_B, she checks whether it is a valid signature for m_B which was committed in the acceptance step. If she does not get any response from B or s_B is not valid, she invokes the A-resolve subprotocol.

Abort subprotocol

A asks T to abort by sending a signed request message $S_A(A, B, \sigma_{AB}, \mathsf{abort})$. Then T searches his DB and does the following:

1. If σ_{AB} was resolved by B, he gives an alternative regular signature $s_{TB} = S_T(A, B, \sigma_{BA}, \sigma_{AB})$ to A.

2. If σ_{AB} was resolved by A already, he replies with "Resolved" message.

3. If σ_{AB} was not resolved yet, he gives an abort token $S_T(S_A(A, B, \sigma_{AB}, \mathsf{abort}))$ to A and saves it in DB.

B-resolve subprotocol

B sends $(A, B, \sigma_{AB}, \sigma_{BA})$ to T and asks to resolve. Then T searches his DB and does the following:

1. If σ_{AB} was aborted by A, he sends the abort token $S_T(S_A(A, B, \sigma_{AB}, \mathsf{abort}))$ to B.

2. If (σ_A, σ_B) was resolved by A, he computes $s_{TA} = S_T(A, B, \sigma_{AB}, \sigma_{BA})$, an alternative regular signature to s_A, and gives it to B.

3. If $(\sigma_{AB}, \sigma_{BA})$ was neither aborted nor resolved yet, T checks whether they are a matching negotiation. If they are a matching negotiation, T gives $s_{TA} = S_T(A, B, \sigma_{AB}, \sigma_{BA})$ to B and saves

$(A, B, \sigma_{AB}, \sigma_{BA}, s_{TA})$ in DB. s_{TA} is an alternative regular signature to s_A. Anyone can verify the fact that T has issued an alternative regular signature for m_A legally. Therefore s_{TA} is equivalent to s_A in legal sense.

A-resolve subprotocol

A sends $(A, B, \sigma_{AB}, \sigma_{BA})$ to T and asks to resolve. Then T searches his DB and does the following:

1. If σ_{AB} was aborted by A, he replies with "Aborted".
2. If $(\sigma_{AB}, \sigma_{BA})$ was resolved by B, he computes $s_{TB} = S_T(A, B, \sigma_{BA}, \sigma_{AB})$, an alternative regular signature to s_B, and gives it to A.
3. If $(\sigma_{AB}, \sigma_{BA})$ was neither aborted nor resolved yet, he checks whether they are a matching negotiation. If they are a matching negotiation, T gives $s_{TB} = S_T(A, B, \sigma_{BA}, \sigma_{AB})$ to A and saves $(A, B, \sigma_{AB}, \sigma_{BA}, s_{TB})$ in DB. s_{TB} is an alternative regular signature to s_B. Anyone can verify the fact that T has issued an alternative regular signature for m_B legally. Therefore s_{TB} is equivalent to s_B in legal sense.

If the exchange protocol is finished successfully without involvement of TTP, A and B will receive their expected items $s_B(m_B)$ and $s_A(m_A)$, respectively. These two signatures are independent, so one signature does not include any information of the other signature. If the exchange protocol was resolved by TTP, A and B will receive their expected items in the form of alternative regular signatures $s_{TB} = S_T(A, B, \sigma_{BA}, \sigma_{AB})$ and $s_{TA} = S_T(A, B, \sigma_{AB}, \sigma_{BA})$, respectively. These two signatures are highly dependent, so privacy of negotiation and exchange is exposed when it is used to other party.

4.3 Security Analysis

First of all, we consider the fairness of exchange protocol. We formalize the fairness of exchange protocol as follows: Consider an exchange protocol between an honest player and an attacker. The attacker has full control of protocol, *i.e.*, he can stop the protocol, delay communication arbitrary, and be involved in as many sessions as he wants. The honest player works as a random oracle, *i.e.*, he replies the query of the attacker according to the protocol. If the attacker gets the honest player's signature while the honest player does not get the attacker's signature, the attacker wins the game. Fairness means that the probability for the attacker to win the game is negligible.

Theorem 4.1 *If the honest player can eventually reach the TTP, then the proposed exchange protocol is fair.*

Proof: (sketch)

We consider the cases that the exchange protocol is stopped in the middle by the attacker and consider the advantage of the attacker.

1. Consider the case that attacker B stops the exchange protocol after he receives σ_{AB}. B is potentially more advantageous than A. But the conditional signature σ_{AB} is of no use by itself. If he wants to transfer his advantage to a real signature, he has to prepare conforming σ_{BA} and invoke the B-resolve subprotocol by presenting $(A, B, \sigma_{AB}, \sigma_{BA})$ to T. Assume that B successfully get s_{TA} from T. If the honest player A can eventually reach T, then A will get s_{TB} from T.
2. Consider the case that attacker A stops the exchange protocol after she receives σ_{BA}: A is potentially more advantageous than B because she has two choices either to abort or to resolve while B has only one choice to resolve. But the conditional signature σ_{BA} is of no use by itself. If she wants to get a real signature, she has to invoke the A-resolve subprotocol by presenting $(A, B, \sigma_{AB}, \sigma_{BA})$ to T. Assume that A successfully get s_{TB} from T. If the honest player B can eventually reach T, then B will get s_{TA} from T.
3. Consider the case that attacker B stops the exchange protocol after he receives s_A: B is temporary more advantageous than A because he has received a real signature s_A while A does not have one. But A can invoke A-resolve subprotocol. If the honest player A can eventually reach T, then A will get s_{TB} from T which is equivalent to s_B in legal sense. Note that B cannot abort the exchange protocol.

Therefore, in every possible cases, the proposed exchange protocol is fair for both A and B. □

Next, we consider the accountability of TTP. Although we assume the trustedness of TTP, there is possibility of TTP's misbehavior. For example, TTP can try to give an abort token to A while he has issued an alternative regular signature s_{TA} to B. An exchange protocol is accountable if any misbehavior of T which results in loss of fairness can be proven.

Theorem 4.2 *The proposed exchange protocol is accountable.*

Proof: (sketch) If A or B asks abort or resolve, T has to answer either with an abort token or an alternative regular signature. If T gives both an abort token and an alternative regular signature for the same $(\sigma_{AB}, \sigma_{BA})$, he will be judged to have cheated. If T gives an abort token to a player and gives an alternative regular signature to the other player, then the victim who has the abort token can prove that T has cheated. □

The proposed fair exchange protocol using conditional signature satisfies all the security requirements listed before.

1. *Completeness*: If the exchange was finished successfully without any involvement of TTP, A and B get the expected items s_B and s_A, respectively.

2. *Fairness*: Theorem 4.1.

3. *Timeliness*: At any time during a protocol run, each player can ask T to abort or resolve which will resolve any argument in time. Therefore, each player does not need to wait.

4. *Accountability*: Theorem 4.2.

5. *Independence*: If the exchange was finished successfully without any involvement of TTP, A and B get the expected items s_B and s_A, respectively, which are not related with each other at all. But if the protocol was resolved by TTP, the independence of items cannot be kept. Alternative regular signatures s_{TA} and s_{TB} issued by TTP are highly related each other. For example, s_{TA} is an electronic check paid for a flight ticket and s_{TB} is a flight ticket paid by an electronic check.

5 Comparison of Fair Exchange Protocols

We compare the properties of the proposed fair exchange protocol with those of [ASW00].

First, the verifiable escrow based scheme does not satisfy the accountability. If T opens the verifiable escrow β or the ordinary escrow α to get a real signature and gives it to a player, it is indistinguishable whether the real signature is given by the signer or opened by T. Therefore T's misbehavior cannot be proven. If an argument is resolved by T, the resulting message is a real signature of the other player. Therefore the independence of messages is preserved although T has resolved the argument.

But other two schemes satisfy the accountability property. If T gives both an abort token and an alternative regular signature for the same exchange, he will be judged to have cheated. If an argument of a fair exchange protocol is resolved by T, the resulting signatures are highly related with other's item and independence of message is not provided. Therefore accountability and independence of message cannot be satisfied together.

If we compare the proposed fair exchange protocol with the accountable contract signing protocol of [ASW00], we can see several differences although the overall approach is very similar. In the proposed exchange protocol conditional signature is used rather than pre-contract. Conditional signature is a conditional commitment of the signer for a regular signature on a message under a signer-chosen condition. Using conditional signature the proposed exchange protocol provides additional functionality of negotiation. We apply the conditional signature to the fair exchange of digital signatures while [ASW00] apply pre-contract to the contract signing problem.

6 Concluding Remarks

We have introduced the concept of conditional signature as a tool to implement private negotiation between two entities. Using conditional signature a signer can commit his regular signature on a message to a specific receiver under a signer-chosen condition.

Using the conditional signature we have constructed a negotiation protocol between an initiator and a responder. Negotiation protocol consists of a proposal stage by the initiator and an acceptance stage by the responder. Two negotiation statements in which each message conforms to other's condition is called a matching negotiation.

We model the fair exchange problem as a combination of negotiation stage and real exchange stage. Matching negotiation and real exchange according to the negotiation give a fair exchange. The proposed exchange protocol is an optimistic fair exchange protocol with 4 step exchanges and 3 subprotocols. We prove that the proposed exchange protocol is fair and accountable.

Real exchange after negotiation is a common experience in electronic commerce scenario. The proposed fair exchange protocol is efficient since it is implemented only with digital signature. It is more flexible because it provides negotiation functionality. It is expected that many electronic commerce protocols can be designed by using the proposed fair exchange protocol.

References

[ASW00] N. Asokan, V. Shoup, and M. Waidner,, "Optimistic fair exchange of digital signatures", *IEEE Journal on Selected Areas in Communications*, Volume 18, Issue 4, pages 593–610, April 2000.

[BGMR90] M. Ben-Or, O. Goldreich, S. Micali, and R. Rivest, "A fair protocol for signing contracts", *IEEE Transactions on Information Theory*, IT-36(1), pages 40–46, January, 1990.

[Dam93] I. B. Damgard, "Practical and provably secure release of a secret and exchange of signatures", *Advances in Cryptology–Eurocrypt'93*, LNCS Vol.765, pages 200–217, Springer-Verlag, 1993.

[FR97] M. K. Franklin and M. K. Reiter, "Fair exchange with a semi-trusted third party", *4th ACM Conference on Computer and Communications Security*, pages 1–5, 1997.

[GJM99] J. Garay, M. Jakobsson, and P. MacKenzie, "Abuse-free Optimistic Contract Signing", *Advances in Cryptology – Crypto'99*, LNCS Vol.1666, pages 449–466, Springer-Verlag, 1999.

[PS00] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures", *Journal of Cryptology*, Volume 13, Number 3, Pages 361–396, Springer-Verlag, 2000.

[ZDB00] J. Zhou, R.H. Deng and F. Bao, "Some remarks on a fair exchange protocol", *PKC2000, Public Key Cryptography*, LNCS Vol.1751, pages 45–57, Springer-Verlag, 2000.