Further Observations on Optimistic Fair Exchange Protocols in the Multi-user Setting

Xinyi Huang[1], Yi Mu[2], Willy Susilo[2], Wei Wu[2], and Yang Xiang[3]

[1] School of Information Systems,
Singapore Management University, Singapore
xyhuang@smu.edu.sg

[2] Centre for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Australia
{ymu,wsusilo,ww986}@uow.edu.au

[3] School of Information Technology, Deakin University, Australia
yxiang2@gmail.com

Abstract. Recent research has shown that the single-user security of optimistic fair exchange cannot guarantee the multi-user security. This paper investigates the conditions under which the security of optimistic fair exchange in the single-user setting is preserved in the multi-user setting. We first introduce and define a property called "Strong Resolution-Ambiguity". Then we prove that in the certified-key model, an optimistic fair exchange protocol is secure in the multi-user setting if it is secure in the single-user setting and has the property of strong resolution-ambiguity. Finally we provide a new construction of optimistic fair exchange with strong resolution-ambiguity. The new protocol is setup-free, stand-alone and multi-user secure without random oracles.

1 Introduction

In a fair exchange protocol, two parties can exchange their items in a fair way so that no one can gain any advantage in the process. A simple way to realize fair exchange is to introduce an *online* trusted third party who acts as a mediator: earth party sends the item to the trusted third party, who upon verifying the correctness of both items, forwards each item to the other party. A drawback of this approach is that the trusted third party is always involved in the exchange even if both parties are honest and no fault occurs. In practice, the trusted third party could become a bottleneck of the system and is vulnerable to the denial-of-service attack.

Optimistic Fair Exchange (also known as off-line fair exchange) was introduced by Asokan *et al.* [1]. An optimistic fair exchange protocol also needs a third party called "arbitrator", who is not required to be online all the time. Instead, the arbitrator only gets invoked when something goes wrong (e.g., one party attempts to cheat or other faults occur). An optimistic fair exchange protocol involves three participants, namely the signer, the verifier and the arbitrator.

P.Q. Nguyen and D. Pointcheval (Eds.): PKC 2010, LNCS 6056, pp. 124–141, 2010.
© International Association for Cryptologic Research 2010

The signer (say, Alice) first issues a verifiable "partial signature" σ' to the verifier (say, Bob). Bob verifies the validity of σ' and fulfills his obligation if σ' is valid. After that, Alice sends Bob a "full signature" σ to complete the transaction. Thus, if no problem occurs, the arbitrator does not participate in the exchange. However, if Bob does not receive the full signature σ from Alice, Bob can send σ' (and the proof of fulfilling his obligation) to the arbitrator, who will convert σ' to σ for Bob.

An optimistic fair exchange protocol can be *setup-driven* or *setup-free* [23]. An optimistic fair exchange protocol is called setup-driven if an initial-key-setup procedure between a signer and the arbitrator is involved. On the other hand, an optimistic fair exchange protocol is called setup-free if the signer does not need to contact the arbitrator, except that the signer can obtain and verify the arbitrator's public key certificate and vice versa. As shown in [10], setup-free is more desirable for the realization of optimistic fair exchange in the multi-user setting. Another notion of optimistic fair exchange is *stand-alone* [23], which requires that the full signature be an ordinary signature.

1.1 Previous Work

As one of the fundamental problems in secure electronic transactions and digital rights management, fair exchange has been studied intensively since its introduction. It is known that optimistic fair exchange can be constructed (in a generic way) using "two signatures" construction [11], verifiably encrypted signature [2,3,8,9,15,20,18], the sequential two-party multisignature (first introduced by Park *et al.* [17], and then broken and repaired by Dodis and Reyzin [11]), the OR-proof [10], and conventional signature and ring signature [14]. In the following, we only review some results which are most relevant to this paper.

Optimistic Fair Exchange in the Single-user Setting
There are three parties involved in an optimistic fair exchange protocol, which are signer(s), verifier(s) and arbitrator(s). Most work about optimistic fair exchange was considered only in the single-user setting, namely there is only one signer. The first formal security model of optimistic fair exchange was proposed in [2,3]. Dodis and Reyzin [11] defined a more generalized and unified model for non-interactive optimistic fair exchange, by introducing a new cryptographic primitive called *verifiably committed signature.* In [11], the security of a verifiably committed signature scheme (equivalently, an optimistic fair exchange protocol) in the single-user setting consists of three aspects: security against the signer, security against the verifier and security against the arbitrator. While the arbitrator is not fully trusted, it is still assumed to be semi-trusted in the sense that the arbitrator will not collude with the signer or the verifier. *In the remainder of this paper, an optimistic fair exchange protocol is single-user secure (or, secure in the single-user setting) means that it is secure in the single-user setting defined in [11].* Notice that their definition does not include all security notions of optimistic fair exchange (e.g., abuse-free [12], non-repudiation [16,21], timely-termination [2,3] and signer-ambiguity [13]), but it does not affect the point we

want to make in this paper. Dodis and Reyzin [11] proposed a stand-alone but setup-driven verifiably committed signature scheme from Gap Diffie-Hellman problem. Constructions of stand-alone and setup-free verifiably committed signature were proposed in [22,23].

Optimistic Fair Exchange in the Multi-user Setting
Recently the security of non-interactive optimistic fair exchange in the multi-user setting was independently studied in [10] and [24]. Optimistic fair exchange in the multi-user setting refers to the scenario where there are two or more signers in the system, but items are still exchanged between two parties. This is different from the multi-party exchange which considers the exchange among three or more parties.

In [10], Dodis, Lee and Yum pointed out that the single-user security of optimistic fair exchange cannot guarantee the multi-user security. They presented a simple counterexample which is secure in the single-user setting but is insecure in a multi-user setting. (In the counterexample, a dishonest verifier in the multi-user setting can obtain a full signature without fulfilling the obligation.) Dodis, Lee and Yum defined the multi-user security model of optimistic fair exchange and provided a generic setup-free construction of optimistic fair exchange secure in the multi-user setting [10]. The security of their construction relies on one-way functions in the random oracle model and trapdoor one-way permutations in the standard model. The analysis in [10] shows that two well-known techniques of optimistic fair exchange (namely, constructions based on verifiably encrypted signatures and sequential two-party signatures) remain secure in the multi-user setting if the underlying primitives satisfy some security notions. Independently, Zhu, Susilo and Mu [24] also demonstrated a verifiably committed signature scheme which is secure in the model defined in [11] but is insecure in the multi-user setting. They defined the security notions of verifiably committed signature in the multi-user setting and proposed a concrete construction of multi-user secure stand-alone and setup-free verifiably committed signature [24]. The non-interactive version of their scheme uses the Fiat-Shamir technique and requires a hash function, which is viewed as the random oracle in security analysis. Due to [10], multi-user secure stand-alone and setup-free optimistic fair exchange protocols without random oracles can be constructed from verifiably encrypted signature schemes without random oracles [15,20,18].

Certified-Key Model and Chosen-Key Model
Most optimistic fair exchange protocols are considered in the *certified-key model* where the user must prove the knowledge of the private key at the key registration phase. Therefore, the adversary is only allowed to make queries about certified public keys. Huang *et al.* [14,13] considered the multi-user security of optimistic fair exchange in the *chosen-key model*, where the adversary can make queries about public keys arbitrarily without requiring to show its knowledge of the corresponding private keys. Optimistic fair exchange protocols secure in the certified-key model may not be secure in the chosen-key model [14].

Huang *et al.* [14] proposed another generic construction for optimistic fair exchange. Their construction can lead to efficient setup-free optimistic fair exchange protocols secure in the standard model and the chosen-key model. Very recently, the first efficient ambiguous optimistic fair exchange protocol was proposed in [13]. The new protocol is proven secure in the multi-user setting and chosen-key model without relying on the random oracle assumption. Without any doubt, it is more desirable if cryptographic protocols can be proven secure in the chosen-key model. However, in this paper, the security of optimistic fair exchange is considered in the certified-key model (as defined in [10]), since certified-key model is reasonable and has been widely used in the research of public key cryptography. *In the remainder of this paper, when we say an optimistic fair exchange protocol is multi-user secure (or, secure in the multi-user setting), it refers that the protocol is secure in the multi-user setting defined in [10] (which is in the certified-key model).*

1.2 Motivation

The research on optimistic fair exchange has shown that:

- The single-user security of optimistic fair exchange does not guarantee the multi-user security [10,24].
- Not all single-user secure optimistic fair exchange protocols are insecure in the multi-user setting [10]. Several single-user secure protocols can be proven secure in the multi-user setting [10].

However, it remains unknown *under which conditions single-user secure optimistic fair exchange protocols will be secure in the multi-user setting*? We believe the investigation of this question not only will provide a further understanding on the security of optimistic fair exchange in the multi-user setting, but also can introduce new constructions of multi-user secure optimistic fair exchange.

1.3 Our Contributions

This paper focuses on both theory investigations and new construction of optimistic fair exchange in the multi-user setting.

1. In Section 3, we introduce and define a new property of optimistic fair exchange, which we call *Strong Resolution-Ambiguity.* Briefly speaking, an optimistic fair exchange protocol has the property of strong resolution-ambiguity if one can transform a partial signature σ' into a full signature σ using signer's private key or arbitrator's private key, and given such a pair (σ', σ), it is infeasible to tell which key is used in the conversion. While there are some optimistic fair exchange protocols satisfying strong resolution-ambiguity, it is the first time this notion is addressed and formally defined.
2. *For an optimistic fair exchange protocol with strong resolution-ambiguity, we prove that its security in the single-user setting is preserved in the multi-user setting.* More precisely, we show that: (1) the security against the signer and the security against the verifier in the single-user setting are preserved in the multi-user setting for optimistic fair exchange protocols with strong

resolution-ambiguity, and (2) the security against the arbitrator in the single-user setting is preserved in the multi-user setting (for optimistic fair exchange protocols either with or without strong resolution-ambiguity).

While strong resolution-ambiguity is not a necessary property for (multi-user secure) optimistic fair exchange protocols, our result provides a new approach for the security analysis of optimistic fair exchange protocols in the multi-user setting: One only needs to analyze the security in the single-user setting (rather than the more complex multi-user setting) for optimistic fair exchange protocols with strong resolution-ambiguity.

3. *In Section 4, we provide a new construction of optimistic fair exchange with strong resolution-ambiguity.* Our construction is a variant of the optimistic fair exchange protocol from the verifiably encrypted signature scheme proposed in [15]. The protocol in [15] has several desirable properties, e.g., setup-free, stand-alone and multi-user secure without random oracles under computational Diffie-Hellman assumption. Our protocol retains all these properties and is more efficient in generating, transmitting and verifying partial signatures. This however is achieved at the cost of larger key size.

2 Definitions of Optimistic Fair Exchange in the Multi-user Setting

This section reviews the syntax and security definitions of optimistic fair exchange in the multi-user setting [10].

2.1 Syntax of Optimistic Fair Exchange

A setup-free non-interactive optimistic fair exchange protocol involves three parties: the signer, the verifier and the arbitrator. It is defined by the following efficient algorithms. An algorithm is called efficient if it is a probabilistic polynomial-time Turing machine.

- $\mathsf{Setup}^{\mathsf{TTP}}$. The arbitrator setup algorithm takes as input a parameter Param, and gives as output a secret arbitration key ASK and a public partial verification key APK.
- $\mathsf{Setup}^{\mathsf{User}}$. The user setup algorithm takes as input Param and (optionally) APK, and gives as output a private signing key SK and a public verification key PK.
- Sig and Ver. These are similar to signing and verification algorithms in an ordinary digital signature scheme.
  - The signing algorithm Sig, run by a signer U_i, takes as input $(m, \mathsf{SK}_{U_i}, \mathsf{APK})$ and gives as output a signature σ_{U_i} on the message m. In fair exchange protocols, signatures generated by Sig are called as *full signatures.*
  - The verification algorithm Ver, run by a verifier, takes as input $(m, \sigma_{U_i}, \mathsf{PK}_{U_i}, \mathsf{APK})$ and returns `valid` or `invalid`. A signature σ_{U_i} is said to be a valid full signature of m under PK_{U_i} if $\mathsf{Ver}(m, \sigma_{U_i}, \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$.

- PSig and PVer. These are partial signing and verification algorithms, where PSig together with Res (which will be defined soon) are functionally equivalent to Sig.
  - The partial signing algorithm PSig, run by a signer U_i, takes as input $(m, \mathsf{SK}_{U_i}, \mathsf{APK})$ and gives as output a signature σ'_{U_i} on m. To distinguish from those produced by Sig, signatures generated by PSig are called as *partial signatures.*
  - The partial verification algorithm PVer, run by a verifier, takes as input $(m, \sigma'_{U_i}, \mathsf{PK}_{U_i}, \mathsf{APK})$ and returns `valid` or `invalid`. A signature σ'_{U_i} is said to be a valid partial signature of m under PK_{U_i} if $\mathsf{PVer}(m, \sigma'_{U_i}, \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$.
- Res. The resolution algorithm Res takes as input a valid partial signature σ'_{U_i} of m under PK_{U_i} and the secret arbitration key ASK, and gives as output a signature σ_{U_i}. This algorithm is run by the arbitrator for a party U_j, who does not receive the full signature from U_i, but possesses a valid partial signature of U_i and a proof that he/she has fulfilled the obligation to U_i.

Correctness. If each signature is generated according to the protocol specification, then it should pass the corresponding verification algorithms. Namely,

1. $\mathsf{Ver}(m, \mathsf{Sig}(m, \mathsf{SK}_{U_i}, \mathsf{APK}), \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$.
2. $\mathsf{PVer}(m, \mathsf{PSig}(m, \mathsf{SK}_{U_i}, \mathsf{APK}), \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$.
3. $\mathsf{Ver}(m, \mathsf{Res}(m, \mathsf{PSig}(m, \mathsf{SK}_{U_i}, \mathsf{APK}), \mathsf{ASK}, \mathsf{PK}_{U_i}), \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$.

Resolution-Ambiguity [10,11,14,16,24]. Any "resolved signature" $\mathsf{Res}(m, \mathsf{PSig}(m, \mathsf{SK}_{U_i}, \mathsf{APK}), \mathsf{ASK}, \mathsf{PK}_{U_i})$ is (at least computationally) indistinguishable from the "actual signature" $\mathsf{Sig}(m, \mathsf{SK}_{U_i}, \mathsf{APK})$.

Security of Optimistic Fair Exchange. Intuitively, the fairness of an exchange requires that two parties exchange their items in a fair way so that either each party obtains the other's item or neither party does. This requirement consists of the security against signer(s), the security against verifier(s) and the security against the arbitrator, which will be defined by the game between the adversary and the challenger. During the game, the challenger will maintain three initially empty lists: (1) *PK-List* contains the public keys of created users; (2) *PartialSign-List* contains the partial signing queries made by the adversary; and (3) *Resolve-List* contains the resolution queries made by the adversary.

The definitions in the following sections are inspired by those in [10], with modifications which we believe can demonstrate the difference between the single-user security and the multi-user security of optimistic fair exchange.

2.2 Security against Signer(s)

In an optimistic fair exchange protocol, the signer should not be able to generate a valid partial signature which cannot be converted into a valid full signature by the arbitrator. This property is defined by the following game.

- **Setup**. The challenger generates the parameter Param and the arbitrator's key pair $(\mathsf{APK}, \mathsf{ASK})$ by running $\mathsf{Setup}^{\mathsf{TTP}}$. The adversary $\mathcal{A}$ is given Param and APK.
- **Queries**. Proceeding adaptively, $\mathcal{A}$ can make following queries.
  Creating-User-Queries. $\mathcal{A}$ can create a user U_i by making a creating-user query (U_i, PK_{U_i}). In order to convince the challenger to accept PK_{U_i} (i.e., add PK_{U_i} to the *PK-List*), $\mathcal{A}$ must prove its knowledge of the legitimate private key SK_{U_i}. This can be realized by requiring the adversary to hand over the private key as suggested in [15], or generate a proof of knowledge [4] of the private key[1].
  Resolution-Queries. For a resolution-query $(m, \sigma', \mathsf{PK})$ satisfying $\mathsf{PVer}(m, \sigma', \mathsf{PK}, \mathsf{APK}) = \mathtt{valid}$, the challenger first browses *PK-List*. If $\mathsf{PK} \notin$ *PK-List*, an error symbol "$\top$" will be returned to the adversary. Otherwise, the challenger adds (m, PK) to the *Resolve-List* (if the pair (m, PK) is not there) and responds with an output of $\mathsf{Res}(m, \sigma', \mathsf{ASK}, \mathsf{PK})$.
- **Output**. Eventually, $\mathcal{A}$ outputs a triple $(m_f, \sigma'_f, \mathsf{PK}^*)$ and wins the game if $\mathsf{PK}^* \in$ *PK-List*, $\mathsf{PVer}(m_f, \sigma'_f, \mathsf{PK}^*, \mathsf{APK}) = \mathtt{valid}$, and $\mathsf{Ver}(m_f, \mathsf{Res}(m_f, \sigma'_f, \mathsf{ASK}, \mathsf{PK}^*), \mathsf{PK}^*, \mathsf{APK}) = \mathtt{invalid}$.

Let $\mathsf{Adv\ OFE}_{\mathcal{A}}$ be the probability that $\mathcal{A}$ wins in the above game, taken over the coin tosses made by $\mathcal{A}$ and the challenger. An adversary $\mathcal{A}$ is said to $(t, q_{CU}, q_R, \epsilon)$-break the security against signer(s) if in time t, $\mathcal{A}$ makes at most q_{CU} *Creating-User-Queries*, q_R *Resolution-Queries* and $\mathsf{Adv\ OFE}_{\mathcal{A}}$ is at least ϵ.

Definition 1 (Security against Signer(s)). *An optimistic fair exchange protocol is $(t, q_{CU}, q_R, \epsilon)$-secure against signer(s) if no adversary $(t, q_{CU}, q_R, \epsilon)$-breaks it.*

By setting $q_{CU} = 1$, we can define the security against the signer in the single-user setting, namely an optimistic fair exchange protocol is (t, q_R, ϵ)-secure against the signer in the single-user setting if no adversary $(t, 1, q_R, \epsilon)$-breaks it.

2.3 Security against Verifier(s)

Briefly speaking, the security against verifier(s) requires that the verifier should not be able to generate a valid partial signature of a new message or generate a valid full signature without the assistance from the signer or the arbitrator.

The first requirement is ensured by the security against the arbitrator, namely even the arbitrator (knowing more than the verifier) cannot succeed in that attack. This will be defined shortly in Section 2.4. The second requirement is defined as below.

- **Setup**. The challenger generates the parameter Param and the arbitrator's key pair $(\mathsf{APK}, \mathsf{ASK})$ by running $\mathsf{Setup}^{\mathsf{TTP}}$. The challenger also generates a key pair $(\mathsf{PK}^*, \mathsf{SK}^*)$ by running $\mathsf{Setup}^{\mathsf{User}}$, and adds PK^* to *PK-List*. The adversary $\mathcal{B}$ is given Param, APK and PK^*.

[1] We will use the latter approach in the proof.

- **Queries**. Proceeding adaptively, $\mathcal{B}$ can make all queries defined in Section 2.2 and Partial-Signing-Queries defined as follows.
  Partial-Signing-Queries. For a partial-signing query (m, PK^*), the challenger responds with an output of $\mathsf{PSig}(m, \mathsf{SK}^*, \mathsf{APK})$. After that, (m, PK^*) is added to the *PartialSign-List*. ($\mathcal{B}$ is allowed to make Partial-Signing-Queries only about PK^* as other public keys are created by $\mathcal{B}$.)
- **Output**. Eventually, $\mathcal{B}$ outputs a pair (m_f, σ_f) and wins the game if $(m_f, \mathsf{PK}^*) \notin$ *Resolve-List* and $\mathsf{Ver}(m_f, \sigma_f, \mathsf{PK}^*, \mathsf{APK}) = \mathtt{valid}$.

Let $\mathsf{Adv\ OFE}_{\mathcal{B}}$ be the probability that $\mathcal{B}$ wins in the above game, taken over the coin tosses made by $\mathcal{B}$ and the challenger. An adversary $\mathcal{B}$ is said to $(t, q_{CU}, q_{PS}, q_R, \epsilon)$-break the security against verifier(s) if in time t, $\mathcal{B}$ makes at most q_{CU} *Creating-User-Queries*, q_{PS} *Partial-Signing-Queries*, q_R *Resolution-Queries* and $\mathsf{Adv\ OFE}_{\mathcal{B}}$ is at least ϵ.

Definition 2 (Security against Verifier(s)). *An optimistic fair exchange protocol is $(t, q_{CU}, q_{PS}, q_R, \epsilon)$-secure against verifier(s) if no adversary $(t, q_{CU}, q_{PS}, q_R, \epsilon)$-breaks it.*

Similarly, we can obtain the definition of the security against the verifier in the single-user setting, namely an optimistic fair exchange protocol is $(t, q_{PS}, q_R, \epsilon)$-secure against the verifier in the single-user setting if no adversary $(t, 0, q_{PS}, q_R, \epsilon)$-breaks it.

2.4 Security against the Arbitrator

In this section, we will define the security against the arbitrator and prove that the security against the arbitrator in the single-user setting is preserved in the multi-user setting.

The security against the arbitrator requires that the arbitrator, without the partial signature on a message m, should not be able to produce a valid full signature on m[2]. This notion is defined as follows.

- **Setup**. The challenger generates the parameter Param, which is given to the adversary $\mathcal{C}$.
- **Output-I**. $\mathcal{C}$ generates the arbitrator's public key APK and sends it to the challenger. ($\mathcal{C}$ is required to prove the knowledge of the legitimate private key ASK.) In response, the challenger generates a key pair $(\mathsf{PK}^*, \mathsf{SK}^*)$ by running $\mathsf{Setup}^{\mathsf{User}}$ and adds PK^* to *PK-List*. The adversary $\mathcal{C}$ is given PK^*.
- **Queries**. Proceeding adaptively, $\mathcal{C}$ can make *Creating-User-Queries* (defined in Section 2.2) and *Partial-Signing-Queries* (defined in Section 2.3).
- **Output-II**. Eventually, $\mathcal{C}$ outputs a pair (m_f, σ_f) and wins the game if $(m_f, \mathsf{PK}^*) \notin$ *PartialSign-List* and $\mathsf{Ver}(m_f, \sigma_f, \mathsf{PK}^*, \mathsf{APK}) = \mathtt{valid}$.

[2] As almost all previous work about optimistic fair exchange, we assume that signer-arbitrator collusion or verifier-arbitrator collusion will not occur. Please refer to [3,11] for discussions of those attacks.

Let $\mathsf{Adv\ OFE}_{\mathcal{C}}$ be the probability that $\mathcal{C}$ wins in the above game, taken over the coin tosses made by $\mathcal{C}$ and the challenger. An adversary $\mathcal{C}$ is said to $(t, q_{CU}, q_{PS}, \epsilon)$-break the security against the arbitrator if in time t, $\mathcal{C}$ makes at most q_{CU} *Creating-User-Queries*, q_{PS} *Partial-Signing-Queries* and $\mathsf{Adv\ OFE}_{\mathcal{C}}$ is at least ϵ.

Remark 1. In the game, the adversary must first generate the arbitrator's public key APK before obtaining PK^* or making other queries. This reflects the definition of optimistic fair exchange as APK could be an input of algorithms $\mathsf{Setup}^{\mathsf{User}}$ and PSig. For concrete protocols where these algorithms do not require APK as the input, the adversary can obtain PK^* and/or make partial-signing-queries of PK^* before generating APK.

Definition 3 (Security against the Arbitrator). *An optimistic fair exchange protocol is* $(t, q_{CU}, q_{PS}, \epsilon)$*-secure against the arbitrator in the multi-user setting if no adversary* $(t, q_{CU}, q_{PS}, \epsilon)$*-breaks it.*

We can obtain the definition of the security against the arbitrator in the single-user setting, namely an optimistic fair exchange protocol is (t, q_{PS}, ϵ)-secure against the arbitrator in the single-user setting if no adversary $(t, 0, q_{PS}, \epsilon)$-breaks it. The following theorem shows that *the security against the arbitrator in the single-user setting is preserved in the multi-user setting.*

Theorem 1. *An optimistic fair exchange protocol is* $(t, q_{CU}, q_{PS}, \epsilon)$*-secure against the arbitrator in the multi-user setting if it is* $(t + t_1 q_{CU}, q_{PS}, \epsilon)$*-secure against the arbitrator in the single-user setting. Here,* t_1 *denotes the time unit to respond to one creating-user query.*

Proof. We denote by $\mathcal{C}_S$ the adversary in the single-user setting and $\mathcal{C}_M$ in the multi-user setting. We will show how to convert a successful $\mathcal{C}_M$ to a successful $\mathcal{C}_S$. At the beginning, $\mathcal{C}_S$ obtains Param from its challenger in the single-user setting.

- **Setup**. Param is given to $\mathcal{C}_M$.
- **Output-I**. Let APK be the arbitrator's public key created by $\mathcal{C}_M$ in the multi-user setting. APK will be sent to $\mathcal{C}_S$'s challenger in the single-user setting. $\mathcal{C}_S$ will make use of $\mathcal{C}_M$ to generate a proof of knowledge, namely $\mathcal{C}_S$ will act as a relay in the proof by forwarding all messages from its challenger to $\mathcal{C}_M$ (or, from $\mathcal{C}_M$ to its challenger). At the end of this phase, $\mathcal{C}_S$ will be given a public key PK^*, which will be forwarded to $\mathcal{C}_M$ as its challenging public key in the multi-user setting.
- **Queries**. We show how $\mathcal{C}_S$ can correctly answer $\mathcal{C}_M$'s queries.

  Creating-User-Queries. For a creating-user query (U_i, PK_{U_i}), $\mathcal{C}_S$ will add PK_{U_i} to *PK-List* if $\mathcal{C}_M$ can generate a proof of knowledge of the legitimate private key.

  Partial-Signing-Queries. For a partial-signing query (m, PK^*), $\mathcal{C}_S$ forwards it to its own challenger and sends the response to $\mathcal{C}_M$.

– **Output-II**. Eventually, $\mathcal{C}_M$ will output a pair (m_f, σ_f). $\mathcal{C}_S$ will set (m_f, σ_f) as its own output in the single-user setting.

$\mathcal{C}_S$ will win the game in the single-user setting if $\mathcal{C}_M$ wins the game in the multi-user setting. It follows that the success probability of $\mathcal{C}_S$ will be ϵ if $\mathcal{C}_M$ can $(t, q_{CU}, q_{PS}, \epsilon)$-break the security against the arbitrator in the multi-user setting.

It remains to show the time consumption in the proof. $\mathcal{C}_S$'s running time is the same as $\mathcal{C}_M$'s running time plus the time it takes to answer creating-user-queries, which we assume each query takes time at most t_1. Therefore, the total time consumption is $t + t_1 q_{CU}$.

We have shown that for an optimistic fair exchange protocol, if there is an adversary $(t, q_{CU}, q_{PS}, \epsilon)$-breaks the security against the arbitrator in the multi-user setting, then there is an adversary $(t + t_1 q_{CU}, q_{PS}, \epsilon)$-breaks the security against the arbitrator in the single-user setting. This completes the proof of Theorem 1. □

Section 3 will investigate the conditions under which the security against the signer and the security against the verifier in the single-user setting will remain in the multi-user setting.

3 Strong Resolution-Ambiguity

This section investigates a new property of optimistic fair exchange, which we call "Strong Resolution-Ambiguity". We will give the definition of strong resolution-ambiguity and prove that for optimistic fair exchange protocols with that property, the security against the signer and the security against the verifier in the single-user setting are preserved in the multi-user setting. Before giving the formal definition, we first review a generic construction of optimistic fair exchange [11].

Optimistic Fair Exchange from Sequential Two-Party Multisignature A multisignature scheme allows any subgroup of users to jointly sign a document such that a verifier is convinced that each user of the subgroup participated in the signing. To construct an optimistic fair exchange protocol, one can use a simple type of multisignature, which is called sequential two-party multisignature. In this construction, the signer first generates two key pairs (pk, sk) and $(\mathsf{APK}, \mathsf{ASK})$, where $(pk, \mathsf{APK}, \mathsf{ASK})$ are sent to the arbitrator through a secured channel. The signer's private key SK is the pair (sk, ASK) and the arbitrator's private key is ASK. The partial signature σ' of a message m is an ordinary signature generated using sk, and the full signature σ is the multisignature generated using σ' and ASK. Given a valid partial signature, both the arbitrator and the signer can convert it to a full signature using ASK. (Recall that ASK is the arbitrator's private key and part of the signer's private key.) It is thus virtually infeasible to tell who (the signer or the arbitrator) converted the partial signature to the full signature. This is the essential requirement of optimistic fair exchange with strong resolution-ambiguity, which is formally defined as follows.

3.1 Definition of Strong Resolution-Ambiguity

We first introduce a probabilistic polynomial-time algorithm $\mathsf{Convert}$ which allows the signer to convert a partial signature to a full one. The definition of $\mathsf{Convert}$ is given as below.

- $\mathsf{Convert}$. This algorithm takes as input the signer's private key SK_{U_i}, (optionally) arbitrator's public key APK, a message m and its valid partial signature σ'. The output is the signer's full signature σ on m.

In a trivial case, each optimistic fair exchange protocol has an algorithm $\mathsf{Convert} = \mathsf{Sig}$. (In this case the full signature generated by $\mathsf{Convert}$ could be totally independent of the partial signature.) Our interest here is to investigate non-trivial $\mathsf{Convert}$ and compare it with the resolution algorithm Res. Recall that, with the knowledge of ASK, one can also convert a partial signature to a full one using Res. This makes the following question interesting: *Given a valid partial signature σ', what are the differences between full signatures produced by* $\mathsf{Convert}$ *and those produced by* Res? The answer to this question inspires the definition of strong resolution-ambiguity.

To formally define the strong resolution-ambiguity, we assume the arbitrator's key pair satisfies an NP-relation R_{TTP}, and users' key pairs satisfy another NP-relation R_{U}. An NP-relation R is a subset of $\{0,1\}^* \times \{0,1\}^*$ for which there exists a polynomial f such that $|y| \leq f(|x|)$ for all $(x, y) \in R$, and there exists a polynomial-time algorithm for deciding membership in R.

In an optimistic fair exchange protocol defined in Section 2, let $(\mathsf{APK}, \mathsf{ASK})$ be any pair in R_{TTP}, and let $(\mathsf{PK}_{U_i}, \mathsf{SK}_{U_i})$ be any pair in R_{U}. For any pair (m, σ') satisfying $\mathsf{PVer}(m, \sigma', \mathsf{PK}_{U_i}, \mathsf{APK}) = \texttt{valid}$, we define

$\mathbb{D}_{\mathsf{Convert}}^{(m,\sigma')}$: probability distribution of full signatures produced by $\mathsf{Convert}(m, \sigma', \mathsf{SK}_{U_i}, \mathsf{APK})$.

$\mathbb{D}_{\mathsf{Res}}^{(m,\sigma')}$: probability distribution of full signatures produced by $\mathsf{Res}(m, \sigma', \mathsf{PK}_{U_i}, \mathsf{ASK})$.

Definition 4 (Strong Resolution-Ambiguity). *An optimistic fair exchange protocol is said to satisfy strong resolution-ambiguity if there exists an algorithm* $\mathsf{Convert}$ *as defined above such that $\mathbb{D}_{\mathsf{Convert}}^{(m,\sigma')}$ is identical to $\mathbb{D}_{\mathsf{Res}}^{(m,\sigma')}$.*

Strong Resolution-Ambiguity and Resolution-Ambiguity: A Brief Comparison

An optimistic fair exchange protocol with strong resolution-ambiguity will satisfy resolution-ambiguity if Sig is defined as $(\mathsf{PSig} + \mathsf{Convert})$, namely the signer first generates a partial signature and then converts it to a full one using $\mathsf{Convert}$. In this case, actual signatures (generated by Sig) are indistinguishable from resolved signatures (generated by Res). However, resolution-ambiguity cannot ensure strong resolution-ambiguity which requires that one can use the signer's private key to convert a partial signature to a full one and the conversion is indistinguishable from that using the arbitrator's private key.

3.2 Optimistic Fair Exchange Protocols with/without Strong Resolution-Ambiguity

It is evident that the generic construction of optimistic fair exchange from sequential two-party multisignature [11] (reviewed at the beginning of Section 3) has the strong resolution-ambiguity property by defining $\mathsf{Convert} = \mathsf{Res}$. Below are some other concrete examples of optimistic fair exchange with/without strong resolution-ambiguity.

Optimistic Fair Exchange from Verifiably Encrypted Signatures
Let OFE-VES be optimistic fair exchange protocols constructed from verifiably encrypted signatures. If the algorithm Sig is deterministic (e.g., the verifiably encrypted signature scheme in [8]), then OFE-VES will have the strong resolution-ambiguity property. For any valid partial signature of m, there is only one output of the algorithm Res, namely the unique full signature of m. By defining $\mathsf{Convert} = \mathsf{Sig}$, $\mathbb{D}^{(m,\sigma')}_{\mathsf{Convert}}$ and $\mathbb{D}^{(m,\sigma')}_{\mathsf{Res}}$ will be identical and the protocols satisfy strong resolution-ambiguity. OFE-VES with probabilistic Sig algorithms could also have the strong resolution-ambiguity property. One example is the optimistic fair exchange protocol from the verifiably encrypted signature scheme proposed in [15]. In [15], the Sig algorithm is the signing algorithm in Waters signature [19], and the partial signature σ' is the encryption of the full signature σ using APK. After extracting σ from σ', the arbitrator will randomize σ such that the output of Res is a full signature uniformly distributed in the full signature space. This makes the distribution of full signatures produced by Res the same as that of full signatures generated by $\mathsf{Convert} = \mathsf{Sig}$.

A Concrete Instance of the Generic Construction in [14]
The generic construction of optimistic fair exchange in [14] is based on a conventional signature scheme and a ring signature scheme, both of which can be constructed efficiently without random oracles. In the protocol, the signer and the arbitrator first generate their own key pairs. The full signature of a message m is a pair (s_1, s_2), where s_1 is the signer's conventional signature on the message m, and s_2 is a ring-signature on m and s_1. Either the signer or the arbitrator is able to generate s_2. This construction will satisfy strong resolution-ambiguity if the distribution of ring signatures generated by the signer is the same as that of ring signatures generated by the arbitrator (e.g., 2-User ring signature scheme without random oracles [5]).

A Concrete Protocol without Strong Resolution-Ambiguity
One example of optimistic fair exchange protocols *without* strong resolution-ambiguity is the single-user secure but multi-user *insecure* optimistic fair exchange protocol proposed in [10]. In this protocol, the full signature of a message m is $\sigma = (r, \delta)$, where δ is the signer's conventional signature on "$m\|y$", $y = f(r)$, and f is a trapdoor one-way permutation. The partial signature is defined as $\sigma' = (y, \delta)$. To convert (y, δ) to a full signature, the arbitrator uses his/her private key f^{-1} to compute $r = f^{-1}(y)$ and obtain the full signature (r, δ). Given a message m and its full signature (r, δ), it is hard to tell if (r, δ) is produced by Sig directly, or first generated by PSig and then by Res. Thus,

as shown in [10], the property "resolution-ambiguity" is satisfied. On the other hand, this protocol does not have strong resolution-ambiguity as f is a trapdoor one-way permutation. Suppose, otherwise, there is an algorithm $\mathsf{Convert}$ such that for a partial signature σ', the outputs of $\mathsf{Convert}(m, \sigma', \mathsf{SK}_{U_i}, f)$ have the same probability distribution as those of $\mathsf{Res}(m, \sigma', \mathsf{PK}_{U_i}, f^{-1})$. Note that for $\sigma' = (y, \delta)$, Res will output a pair (r, δ) such that $y = f(r)$. It follows that $\mathsf{Convert}(m, \sigma', \mathsf{SK}_{U_i}, f)$ must also output (r, δ) satisfying $y = f(r)$ if the protocol has strong resolution-ambiguity. This breaks the one-wayness of f, namely given y, there is an efficient algorithm $\mathsf{Convert}$ which can find r such that $f(r) = y$ without the trapdoor f^{-1}.

Notice that given a partial signature σ', the signer can generate a full signature σ such that σ is indistinguishable from the one converted by the arbitrator. To do that, the signer needs to maintain a list $\{(r, y) : y = f(r)\}$ when he/she produces the partial signature $\sigma' = (y, \delta)$. Later on, for a partial signature (y, δ), the signer can search the list and find the matching pair (r, y). In this case, the signer can generate a full signature (r, δ) which is indistinguishable from the one converted by the resolution algorithm Res. However, this approach does not satisfy the definition of $\mathsf{Convert}$ since it requires an additional input r. (Recall that the inputs of $\mathsf{Convert}$ are only SK_{U_i}, (m, σ') and APK.)

3.3 Security of Optimistic Fair Exchange Protocols with Strong Resolution-Ambiguity

Theorem 1 has shown that the security against the arbitrator in the single-user setting is preserved in the multi-user setting. This section considers the other two security notions, and we will prove that:

1. For optimistic fair exchange protocols with strong resolution-ambiguity, the security against the signer in the single-user setting remains in the multi-user setting (Theorem 2).
2. For optimistic fair exchange protocols with strong resolution-ambiguity, the security against the verifier in the single-user setting remains in the multi-user setting (Theorem 3).

Theorem 2. *An optimistic fair exchange protocol with strong resolution ambiguity is $(t, q_{CU}, q_R, \epsilon)$-secure against signers in the multi-user setting, if it is $(t + t_1 q_{CU} + t_2 q_R, q_R, \epsilon/q_{CU})$-secure against the singer in the single-user setting. Here, t_1 is the time unit depends on the validity of the proof of knowledge and t_2 is the time unit depends on the algorithm* $\mathsf{Convert}$ *in the protocol.*

Proof. We denote by $\mathcal{A}_S$ the adversary in the single-user setting and $\mathcal{A}_M$ in the multi-user setting. In the proof, we use the standard method by showing that for an optimistic fair exchange protocol with strong resolution-ambiguity, a successful $\mathcal{A}_M$ can be converted into a successful $\mathcal{A}_S$. We first give a high-level description of the proof.

$\mathcal{A}_S$ will act as the challenger of $\mathcal{A}_M$ in the proof and answer all queries from the latter. $\mathcal{A}_S$ will set the challenging public key PK^* of $\mathcal{A}_M$ as its own challenging public key, and set $\mathcal{A}_M$'s output as its own output. The most difficult part in the proof is how $\mathcal{A}_S$ can correctly answer resolution queries from $\mathcal{A}_M$. For resolution queries related to PK^*, $\mathcal{A}_S$ can use its own challenger to generate correct responses. However, this is not feasible for resolution queries about other public keys (since $\mathcal{A}_S$'s challenger only responds to queries about PK^*). Fortunately, such queries can be correctly answered by $\mathcal{A}_S$ if the optimistic fair exchange protocol has strong resolution-ambiguity. For a resolution query $(m, \sigma', \mathsf{PK}_{U_i})$, $\mathcal{A}_S$ can convert σ' to a full signature σ using the algorithm $\mathsf{Convert}$ and the private key SK_{U_i}. Due to Def. 4, this perfectly simulates the real game between $\mathcal{A}_M$ and the challenger in the multi-user setting. The private key SK_{U_i} can be extracted by $\mathcal{A}_S$ due to the validity of the proof of knowledge required in the creating-user phase.

The details of the proof appear in the full version of this paper. □

Theorem 3. *An optimistic fair exchange protocol with strong resolution ambiguity is $(t, q_{CU}, q_{PS}, q_R, \epsilon)$-secure against verifiers in the multi-user setting, if it is $(t + t_1 q_{CU} + t_2 q_R, q_{PS}, q_R, \epsilon)$-secure against the verifier in the single-user setting. Here, t_1 is the time unit depends on the validity of the proof of knowledge and t_2 is the time unit depends on the algorithm* $\mathsf{Convert}$ *in the protocol.*

Proof. The details of the proof appear in the full version of this paper.

Remark 2. Our analysis only shows that strong resolution-ambiguity is a sufficient condition for single-user secure optimistic fair exchange protocols remaining secure in the multi-user setting. It is not a necessary property for (multi-user secure) optimistic fair exchange protocols.

4 A New Optimistic Fair Exchange Protocol with Strong Resolution-Ambiguity

A new optimistic fair exchange protocol with strong resolution-ambiguity is proposed in this section. The protocol is based on Waters signature [19] from bilinear mappings. Definitions of bilinear mappings and computational Diffie-Hellman assumption can be found in [19].

4.1 The Proposed Protocol

Let $(\mathbb{G}, \mathbb{G}_T)$ be bilinear groups of prime order p and let g be a generator of $\mathbb{G}$. e denotes the bilinear mapping $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let n be the bit-string length of the message to be signed. For an element m in $\{0,1\}^n$, let $\mathcal{M} \subseteq \{1, 2, \cdots, n\}$ be the set of all i for which the i^{th} bit m_i is 1. The parameter Param is $(\mathbb{G}, \mathbb{G}_T, p, g, e, n)$.

- $\mathsf{Setup}^{\mathsf{TTP}}$. Given Param, the arbitrator chooses a random number $w \in \mathbb{Z}_p$ and calculates $W = g^w$. The arbitrator's public key APK is W, and the private key ASK is w.
- $\mathsf{Setup}^{\mathsf{User}}$. Given Param, this algorithm outputs a private signing key $\mathsf{SK}_{U_i} = (x_{U_i}, y_{U_i})$ and a public verification key $\mathsf{PK}_{U_i} = (X_{U_i}, Y_{U_i}, \boldsymbol{v}_{U_i})$, where

1. x_{U_i} and y_{U_i} are randomly chosen in $\mathbb{Z}_p$;
2. $X_{U_i} = e(g, g)^{x_{U_i}}$ and $Y_{U_i} = g^{y_{U_i}}$; and
3. $\boldsymbol{v}_{U_i}$ is a vector consisting of $n + 1$ elements $V_0, V_1, V_2, \cdots, V_n$. All these elements are randomly selected in $\mathbb{G}$.

- Sig. Given a message m, the signer U_i uses the private key x_{U_i} to generate a Waters signature $\sigma = (\sigma_1, \sigma_2)$, where $\sigma_1 = g^{x_{U_i}} \cdot (V_0 \prod_{i \in \mathcal{M}} V_i)^r$, $\sigma_2 = g^r$ and r is a random number in $\mathbb{Z}_p$.
- Ver. Given a message-signature pair (m, σ) and U_i's public key $\mathsf{PK}_{U_i} = (X_{U_i}, Y_{U_i}, \boldsymbol{v}_{U_i})$, this algorithm outputs `valid` if $e(\sigma_1, g) = X_{U_i} \cdot e(V_0 \prod_{i \in \mathcal{M}} V_i,\ \sigma_2)$. Otherwise, this algorithm outputs `invalid`.
- PSig. Given a message m and the arbitrator's public key W, the signer U_i first runs Sig to obtain a full signature (σ_1, σ_2). After that, U_i calculates $\sigma_1' = \sigma_1 \cdot W^{y_{U_i}}$ and $\sigma_2' = \sigma_2$. The partial signature σ' is (σ_1', σ_2').
- PVer. Given a pair (m, σ'), U_i's public key PK_{U_i} and arbitrator's public key APK (which is W), one parses σ' as (σ_1', σ_2'). This algorithm outputs `valid` if $e(\sigma_1', g) = X_{U_i} \cdot e(Y_{U_i}, W) \cdot e(V_0 \prod_{i \in \mathcal{M}} V_i, \sigma_2')$. Otherwise, it outputs `invalid`.
- Res. Given a valid partial signature σ' of the message m under a public key $\mathsf{PK}_{U_i} = (X_{U_i}, Y_{U_i}, \boldsymbol{v}_{U_i})$, the arbitrator first parses σ' as (σ_1', σ_2'). After that, the arbitrator uses the private key w to calculate $\sigma_1 = \sigma_1' \cdot (Y_{U_i})^{-w}$ and $\sigma_2 = \sigma_2'$. The arbitrator then chooses a random number $r' \in \mathbb{Z}_p$ and calculates $\sigma_1^R = \sigma_1 \cdot (V_0 \prod_{i \in \mathcal{M}} V_i)^{r'}$ and $\sigma_2^R = \sigma_2 \cdot g^{r'}$. The output of the algorithm Res is (σ_1^R, σ_2^R).

Analysis of Our Protocol. It is evident that our protocol is setup-free and stand-alone. We show that it also satisfies strong resolution-ambiguity.

One can find an algorithm $\mathsf{Convert}$, which is the same as Sig, such that given any partial signature σ', the outputs of $\mathsf{Convert}$ are indistinguishable from those produced by Res, both of which are uniformly distributed in the valid signature space of Waters signature. Thus, the proposed protocol also satisfies strong resolution-ambiguity.

The following theorem shows that the protocol is secure in the multi-user setting.

Theorem 4. *The proposed protocol is multi-user secure under computational Diffie-Hellman assumption.*

Proof. The details of the proof appear in the full version of this paper. □

4.2 Comparison to Previous Protocols

Table. 1 compares the known optimistic fair exchange protocols which have the same properties as the newly proposed one (namely, non-interactive, setup-free, stand-alone and multi-user secure without random oracles). The comparison is made from the following aspects: (1) underlying complexity assumption, (2) partial signature size and full signature size, and (3) the computational cost of signing and verifying partial signatures and full signatures. We consider the cost of signing and verifying partial signatures since the signer must generate

Table 1. Multi-user Secure Stand-Alone and Setup-Free Optimistic Fair Exchange Protocols without Random Oracles

	Our Protocol	[15]	[20]	[18]
Complexity Assumption	CDH	CDH	CT-CDH	SDH
Full Signature	Waters [19]	Waters [19]	Waters [19]	BB [7]
Signature Size$^{\mathsf{PSig}}$	$2\lvert\mathbb{G}\rvert$	$3\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert$	$2\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert$
Signing Cost$^{\mathsf{PSig}}$	$\mathsf{C}_{\mathsf{W}}+\ 1\mathsf{Exp}_{\mathbb{G}}$	$\mathsf{C}_{\mathsf{W}}+\ 2Exp_{\mathbb{G}}$	C_{W}	$\mathsf{C}_{\mathsf{BB}}+2Exp_{\mathbb{G}}$
Verification Cost$^{\mathsf{PVer}}$	$2BM+1\mathsf{BM}$	$3BM$	$2BM+1\mathsf{BM}$	$2BM+4\mathsf{BM}$

Notations.
CDH: Computational Diffie-Hellman assumption.
CT-CDH: Chosen-target computational Diffie-Hellman assumption [6].
SDH: Strong Diffie-Hellman assumption.
$|\mathbb{G}|$: bit length of an element in $\mathbb{G}$, $|\mathbb{Z}_p|$: bit length of an element in $\mathbb{Z}_p$.
C_{W}: Computational cost of generating one Waters signature [19].
C_{BB}: Computational cost of generating one BB signature [7].
$Exp_{\mathbb{G}}$: Exponentiation in $\mathbb{G}$, $\mathsf{Exp}_{\mathbb{G}}$: Pre-computable exponentiation in $\mathbb{G}$.
BM: Bilinear mapping, BM: Pre-computable bilinear mapping.

a partial signature in each exchange, which will be verified by the verifier and could also be checked again by the arbitrator. Therefore, the efficiency of signing and verifying partial signatures is at least as important as that of full signatures. In Table. 1, the most efficient one is the protocol constructed from the verifiably encrypted signature scheme in [18], whose security assumption is strong Diffie-Hellman assumption (SDH). The other three protocols are all based on Waters signature, but the security of the protocol in [20] can only be reduced to a stronger assumption: chosen-target computational Diffie-Hellman assumption (CT-CDH). Our protocol and the one proposed in [15] are designed in a similar manner. When compared with [15], our protocol has a shorter partial signature size and is more efficient in signing and verifying partial signatures. This is achieved at the cost of larger key size (one more pair (y_{U_i}, Y_{U_i}) in $\mathbb{Z}_p \times \mathbb{G}$).

5 Conclusion

This paper shows several new results about optimistic fair exchange in the multi-user setting. We formally defined the *Strong Resolution-Ambiguity* in optimistic fair exchange and demonstrated several concrete optimistic fair exchange protocols with that property. In the certified-key model, we prove that for optimistic fair exchange protocols with strong resolution-ambiguity, the security in the single-user setting can guarantee the security in the multi-user setting. In addition to theoretical investigations, a new construction of optimistic fair exchange with strong resolution-ambiguity was proposed. The new protocol is setup-free, stand-alone, and provably secure in the multi-user setting without random oracles.

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: Proceedings of the 4th ACM conference on Computer and Communications Security, pp. 7–17. ACM Press, New York (1997)
2. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures (Extended abstract). In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 591–606. Springer, Heidelberg (1998)
3. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communication 18(4), 593–610 (2000)
4. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993)
5. Bender, A., Katz, J., Morselli, R.: Ring signatures: Stronger definitions, and constructions without random oracles. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 60–79. Springer, Heidelberg (2006)
6. Boldyreva, A.: Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-Group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)
7. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004)
8. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
9. Camenisch, J., Damgård, I.: Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 331–345. Springer, Heidelberg (2000)
10. Dodis, Y., Lee, P.J., Yum, D.H.: Optimistic fair exchange in a multi-user setting. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 118–133. Springer, Heidelberg (2007)
11. Dodis, Y., Reyzin, L.: Breaking and repairing optimistic fair exchange from PODC 2003. In: Proceedings of the 3rd ACM Workshop on Digital Rights Management, pp. 47–54. ACM, New York (2003)
12. Garay, J.A., Jakobsson, M., MacKenzie, P.: Abuse-free optimistic contract signing. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 449–466. Springer, Heidelberg (1999)
13. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Ambiguous optimistic fair exchange. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 74–89. Springer, Heidelberg (2008)
14. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 106–120. Springer, Heidelberg (2008)
15. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 465–485. Springer, Heidelberg (2006)
16. Markowitch, O., Kremer, S.: An optimistic non-repudiation protocol with transparent trusted third party. In: Davida, G.I., Frankel, Y. (eds.) ISC 2001. LNCS, vol. 2200, pp. 363–378. Springer, Heidelberg (2001)

17. Park, J.M., Chong, E.K.P., Siegel, H.J.: Constructing fair-exchange protocols for E-commerce via distributed computation of RSA signatures. In: Proceedings of the twenty-second annual symposium on Principles of distributed computing, pp. 172–181. ACM, New York (2003)
18. Rückert, M., Schröder, D.: Security of verifiably encrypted signatures and a construction without random oracles. In: Shacham, H. (ed.) Pairing 2009. LNCS, vol. 5671, pp. 17–34. Springer, Heidelberg (2009)
19. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
20. Zhang, J., Mao, J.: A novel verifiably encrypted signature scheme without random oracle. In: Dawson, E., Wong, D.S. (eds.) ISPEC 2007. LNCS, vol. 4464, pp. 65–78. Springer, Heidelberg (2007)
21. Zhou, J., Gollmann, D.: A fair non-repudiation protocol. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy, Washington DC, pp. 55–61. IEEE, Los Alamitos (1996)
22. Zhu, H., Bao, F.: More on stand-alone and setup-free verifiably committed signatures. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 148–158. Springer, Heidelberg (2006)
23. Zhu, H., Bao, F.: Stand-alone and setup-free verifiably committed signatures. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 159–173. Springer, Heidelberg (2006)
24. Zhu, H., Susilo, W., Mu, Y.: Multi-party stand-alone and setup-free verifiably committed signatures. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 134–149. Springer, Heidelberg (2007)