Ambiguous Optimistic Fair Exchange

Qiong Huang[1], Guomin Yang[1], Duncan S. Wong[1], and Willy Susilo[2]

[1] City University of Hong Kong, Hong Kong, China
{csqhuang@student,csyanggm@cs,duncan@}cityu.edu.hk
[2] University of Wollongong, Australia
wsusilo@uow.edu.au

Abstract. Optimistic fair exchange (OFE) is a protocol for solving the problem of exchanging items or services in a fair manner between two parties, a signer and a verifier, with the help of an arbitrator which is called in only when a dispute happens between the two parties. In almost all the previous work on OFE, after obtaining a partial signature from the signer, the verifier can present it to others and show that the signer has indeed committed itself to something corresponding to the partial signature *even* prior to the completion of the transaction. In some scenarios, this capability given to the verifier may be harmful to the signer. In this paper, we propose the notion of *ambiguous optimistic fair exchange* (A-OFE), which is an OFE but also requires that the verifier cannot convince anybody about the authorship of a partial signature generated by the signer. We present a formal security model for A-OFE in the multi-user setting and chosen-key model. We also propose an efficient construction with security proven without relying on the random oracle assumption.

1 Introduction

Optimistic Fair Exchange (OFE) allows two parties to fairly exchange information in such a way that at the end of a protocol run, either both parties have obtained the complete information from one another or none of them has obtained anything from the counter party. In an OFE, there is a third party, called Arbitrator, which only gets involved when a dispute occurred between the two parties. OFE is a useful tool in practice, for example, it can be used for performing contract signing, fair negotiation and similar applications on the Internet. Since its introduction [1], there have been many OFE schemes proposed [2, 3, 4, 12, 13, 14, 18, 21, 23, 24]. For all recently proposed schemes, an OFE protocol for signature typically consists of three message flows. The initiator of OFE, Alice, first sends a message σ_P, called *partial signature*, to the responder, Bob. The partial signature σ_P acts as Alice's partial commitment to her full signature which is to be sent to Bob. But Bob needs to send his full signature to Alice first in the second message flow. After receiving Bob's full signature, Alice sends her full signature to Bob in the third message flow. If in the second message flow that Bob refuses to send his full signature back to Alice, Alice's partial signature σ_P should have no use to Bob, so that Alice has no concern about giving away σ_P. However, after Bob has sent his full signature to

J. Pieprzyk (Ed.): ASIACRYPT 2008, LNCS 5350, pp. 74–89, 2008.

Alice while Alice refuses to send her full signature in the third message flow, then Bob can ask the Arbitrator to retrieve Alice's full signature from σ_P after sending both σ_P and Bob's full signature to the Arbitrator. To the best of our knowledge, among almost all the known OFE schemes, there is one common property about Alice's partial signature σ_P which has neither been captured in any of the security models for OFE nor been considered as a requirement for OFE. The property is that once σ_P is given out, at least one of the following statements is true.

1. Everyone can verify that σ_P must be generated by Alice because σ_P, similar to a standard digital signature, has the non-repudiation property with respect to Alice's public key;
2. Bob can show to anybody that Alice is the signer of σ_P.

For example, in the schemes proposed in [12, 18], the partial signature of Alice is a standard signature, which can only be generated by Alice. In many OFE schemes in the literature, Alice's signature is encrypted under the arbitrator's public key, and then a non-interactive proof is generated to show that the ciphertext indeed contains a signature of Alice. This is known as *verifiably encrypted signature*. However, this raises the question of whether a non-interactive proof that a signature is encrypted is really any different from a signature itself, since it alone is sufficient to prove to any third party that the signer has committed to the message [10].

This property may cause no concern in some applications, for example, in those where only the full signature is deemed to have some actual value to the receiving party. However, it may be undesirable in some other applications. Since σ_P is publicly verifiable and non-repudiative, in practice, σ_P may not be completely useless to Bob. Instead, σ_P has evidently shown Alice's commitment to the corresponding message. This may incur some unfair situation, to the advantage of Bob, if Bob does not send out his full signature. In contract signing applications, this could be undesirable because σ_P can already be considered as Alice's undeniable commitment to a contract in court while there is no evidence showing that Bob has committed to anything.

In another application, fair negotiation, the property above may also be undesirable. Suppose after obtaining σ_P from Alice on her offer, Bob may show it to Charlie, who is Alice's competitor, and ask Charlie for making a better offer. If Charlie's offer is better, then Bob may stop the OFE protocol run with Alice indicating that Bob is unwilling to conclude the negotiation with Alice, and instead carrying out a new OFE protocol run with Charlie. Bob can play the same game iteratively until that no one can give an even better offer. Then Bob can resolve the negotiation by sending his service (i.e. his full signature as the commitment to his service) to the highest bidder.

For making OFE be applicable to more applications and practical scenarios, in this paper, we propose to enhance the security requirements of OFE and construct a new OFE scheme which does not have the problems mentioned above. One may also think of this as an effort to make OFE more admissible as a viable fair exchange tool for real applications. We will build an OFE scheme which not only satisfies all the existing security requirements of OFE (with respect to the

strongest security model available [18]), but in addition to that, will also have σ_P be not self-authenticating and unable for Bob to demonstrate to others that Alice has committed herself to something. We call this enhanced notion of OFE as *Ambiguous Optimistic Fair Exchange* (A-OFE). It inherits all the formalized properties of OFE [12, 18] and has a new property introduced: *signer ambiguity*. It requires that a partial signature σ_P generated by Alice or Bob should look alike and be indistinguishable even to Alice and Bob.

(Related Work): There have been many OFE schemes proposed in the past [2, 3, 4, 12, 13, 18, 21, 23, 24]. In the following, we review some recent ones by starting from 2003 when Park, Chong and Siegel [24] proposed an OFE based on sequential two-party multi-signature. It was later broken and repaired by Dodis and Reyzin [13]. The scheme is *setup-driven* [25, 26], which requires all users to register their keys with the arbitrator prior to any transaction. In [23], Micali proposed another scheme based on a CCA2 secure public key encryption with the property of *recoverable randomness* (i.e., both plaintext and randomness used for generating the ciphertext can be retrieved during decryption). Later, Bao et al. [4] showed that the scheme is not fair, where a dishonest party, Bob, can obtain the full commitment of another party, Alice, without letting Alice get his obligation. They also proposed a fix to defend against the attack.

In PKC 2007, Dodis, Lee and Yum [12] considered OFE in a multi-user setting. Prior to their work, almost all previous results considered the single-user setting only which consists of a single signer and a single verifier (along with an arbitrator). The more practical multi-user setting considers a system to have multiple signers and verifiers (along with the arbitrator), so that a dishonest party can collude with other parties in an attempt of cheating. Dodis et al. [12] showed that security of OFE in the single-user setting does not necessarily imply the security in the multi-user setting. They also proposed a formal definition of OFE in the multi-user setting, and proposed a generic construction, which is *setup-free* (i.e. no key registration is required between users and the arbitrator) and can be built in the random oracle model [5] if there exist one-way functions, or in the standard model if there exist trapdoor one-way permutations.

In CT-RSA 2008, Huang, Yang, Wong and Susilo [18] considered OFE in the multi-user setting and *chosen-key* model, in which the adversary is allowed to choose public keys arbitrarily without showing its knowledge of the corresponding private keys. Prior to their work, the security of all previous OFE schemes (including the one in [12]) are proven in a more restricted model, called *certified-key* (or *registered-key*) model, which requires the adversary to prove its knowledge of the corresponding private key before using a public key. In [18], Huang et al. gave a formal security model for OFE in the multi-user setting and chosen-key model, and proposed an efficient OFE scheme based on ring signature. In their scheme, a partial signature is a conventional signature and a full signature is a two-member ring signature in additional to the conventional signature. The security of their scheme was proven without random oracles.

Liskov and Micali [22] proposed an *online-untransferable signature* scheme, which in essence is an enhanced version of designated confirmer signature, with the extra property that a dishonest recipient, who is interacting with a signer, cannot convince a third party that the signature is generated by the signer. Their scheme is fairly complex and the signing process requires several rounds of interaction with the recipient. Besides, their scheme works in the certified-key model, and is not setup-free, i.e. there is a setup stage between each signer and the confirmer, and the confirmer needs to store a public/secret key pair for each signer, thus a large storage is required for the confirmer.

In [14], Garay, Jakobsson and MacKenzie introduced a similar notion for optimistic contract signing, named *abuse-freeness*. It requires that no party can ever prove to a third party that he is capable of choosing whether to validate or invalidate a contract. They also proposed a construction of abuse-free optimistic contract signing protocol. The security of their scheme is based on DDH assumption under the random oracle model. Besides they did not consider the multi-user setting for their contract signing protocol.

(Our Contributions): In this paper we make the following contributions.

1. We propose the notion of *Ambiguous Optimistic Fair Exchange* (Ambiguous OFE or A-OFE in short) which allows a signer Alice to generate a partial signature in such a way that a verifier Bob cannot convince anybody about the authorship of this partial signature, and thus cannot prove to anybody that Alice committed herself to anything prematurely. Realizing the notion needs to make the partial signature ambiguous with respect to Alice and Bob. We will see that this requires us to include both Alice and Bob's public keys into the signing and verification algorithms of A-OFE.
2. For formalizing A-OFE, we propose a strong security model in the multi-user setting and chosen-key model. Besides the existing security requirements for OFE, that is, resolution ambiguity[1], security against signers, security against verifiers and security against the arbitrator, A-OFE has an additional requirement: *signer ambiguity*. It requires that the verifier can generate partial signatures whose distribution is (computationally) indistinguishable from that of partial signatures generated by the signer. We also evaluate the relations among the security requirements and show that if a scheme has security against the arbitrator and (a weaker variant of) signer ambiguity, then it already has (a weaker variant of) security against verifiers.
3. We propose the first efficient A-OFE scheme and prove its security in the multi-user setting and chosen-key model without random oracle. It is based on Groth and Sahai's idea of constructing a fully anonymous group signature scheme [15, 16] and the security relies on the decision linear assumption and strong Diffie-Hellman assumption.

(Paper Organization): In the next section, we define A-OFE and propose a security model for it. We also show some relation among the formalized security

[1] Resolution ambiguity is just another name for the ambiguity considered in [12, 18].

requirements of A-OFE. In Sec. 3, we introduce some preliminaries which are used in our construction, which is described in Sec. 4. In Sec. 5, we prove the security of our scheme in the standard model, and compare our scheme with other two related work.

2 Ambiguous Optimistic Fair Exchange

In an A-OFE scheme, we require that after receiving a partial signature σ_P from Alice (the signer), Bob (the verifier) cannot convince others but himself that Alice has committed herself to σ_P. This property is closely related to the non-transferability of designated verifier signature [19] and the ambiguity of concurrent signature [11]. Similarly, we require that the verification algorithm in A-OFE should also take as the public keys of both signer and (designated) verifier as inputs, in contrast to that in the traditional definition of OFE [1, 2, 12, 18].

Definition 1 (Ambiguous Optimistic Fair Exchange). *An* ambiguous optimistic fair exchange *(*A-OFE *in short) scheme involves two users (a signer and a verifier) and an arbitrator, and consists of the following (probabilistic) polynomial-time algorithms:*

- PMGen*: On input* 1^k *where* k *is a security parameter, it outputs a system parameter* PM*.*
- $\mathsf{Setup}^{\mathsf{TTP}}$*: On input* PM*, the algorithm generates a public arbitration key* APK *and a secret arbitration key* ASK*.*
- $\mathsf{Setup}^{\mathsf{User}}$*: On input* PM *and (optionally)* APK*, the algorithm outputs a public/secret key pair* (PK, SK)*. For user* U_i*, we use* (PK_i, SK_i) *to denote its key pair.*
- Sig *and* Ver*:* $\mathsf{Sig}(M, SK_i, PK_i, PK_j, APK)$ *outputs a (full) signature* σ_F *on* M *of user* U_i *with the designated verifier* U_j*, where message* M *is chosen by user* U_i *from the message space* $\mathcal{M}$ *defined under* PK_i*, while* $\mathsf{Ver}(M, \sigma_F, PK_i, PK_j, APK)$ *outputs* accept *or* reject*, indicating* σ_F *is* U_i*'s valid full signature on* M *with designated verifier* U_j *or not.*
- PSig *and* PVer*: They are partial signing and verification algorithms respectively.* $\mathsf{PSig}(M, SK_i, PK_i, PK_j, APK)$ *outputs a partial signature* σ_P*, while* $\mathsf{PVer}(M, \sigma_P, \mathbf{PK}, APK)$ *outputs* accept *or* reject*, where* $\mathbf{PK} = \{PK_i, PK_j\}$*.*
- Res*: This is the resolution algorithm.* $\mathsf{Res}(M, \sigma_P, ASK, \mathbf{PK})$*, where* $\mathbf{PK} = \{PK_i, PK_j\}$*, outputs a full signature* σ_F*, or* $\perp$ *indicating the failure of resolving a partial signature.*

Note that we implicitly require that there is an efficient algorithm which given a a pair of (SK, PK), verifies if SK matches PK, i.e. (SK, PK) is an output of algorithm $\mathsf{Setup}^{\mathsf{User}}$. As in [12], PSig together with Res should be functionally equivalent to Sig.

For the correctness, we require that for any $k \in \mathbb{N}$, $\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k)$, $(APK, ASK) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM})$, $(PK_i, SK_i) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)$,

$(PK_j, SK_j) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)$, and $M \in \mathcal{M}(PK_i)$, let $\mathbf{PK} = \{PK_i, PK_j\}$, we have the following

$\mathsf{PVer}(M, \mathsf{PSig}(M, SK_i, PK_i, PK_j, APK), \mathbf{PK}, APK) = \mathsf{accept}$,

$\mathsf{Ver}(M, \mathsf{Sig}(M, SK_i, PK_i, PK_j, APK), PK_i, PK_j, APK) = \mathsf{accept}$, and

$\mathsf{Ver}(M, \mathsf{Res}(M, \mathsf{PSig}(M, SK_i, PK_i, PK_j, APK), ASK, \mathbf{PK}), PK_i, PK_j, APK) = \mathsf{accept}$.

2.1 Security Properties

(Resolution Ambiguity): The *resolution ambiguity* property requires that any 'resolved signature' $\mathsf{Res}(M, \mathsf{PSig}(M, SK_i, PK_i, PK_j, APK), ASK, \{PK_i, PK_j\})$ is *computationally indistinguishable* from an 'actual signature' generated by the signer, $\mathsf{Sig}(M, SK_i, PK_i, PK_j, APK)$. It is identical to 'ambiguity' defined in [12, 18]. Here we just use another name, in order to avoid any confusion, as we will define another kind of ambiguity next.

(Signer Ambiguity): Informally, *signer ambiguity* means that given a partial signature σ_P from a signer A, a verifier B should not be able to convince others that σ_P was indeed generated by A. To capture this property, we use the idea of defining *ambiguity* in concurrent signature [11]. We require that B can generate partial signatures that look *indistinguishable* from those generated by A. This is also the reason why a verifier should also have a public/secret key pair, and the verifier's public key should be included in the inputs of PSig and Sig. Formally, we define an experiment in which D is a probabilistic polynomial-time distinguisher.

$$
\begin{aligned}
\mathsf{PM} &\leftarrow \mathsf{PMGen}(1^k) \\
(APK, ASK) &\leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}) \\
(M, (PK_0, SK_0), (PK_1, SK_1), \delta) &\leftarrow D^{O_{\mathsf{Res}}}(APK) \\
b &\leftarrow \{0, 1\} \\
\sigma_P &\leftarrow \mathsf{PSig}(M, SK_b, PK_b, PK_{1-b}, APK) \\
b' &\leftarrow D^{O_{\mathsf{Res}}}(\delta, \sigma_P) \\
\text{success of } D &:= [b' = b \wedge (M, \sigma_P, \{PK_0, PK_1\}) \notin \mathit{Query}(D, O_{\mathsf{Res}})]
\end{aligned}
$$

where δ is D's state information, oracle O_{Res} takes as input a valid[2] partial signature σ_P of user U_i on message M with respect to verifier U_j, i.e. $(M, \sigma_P, \{PK_i, PK_j\})$, and outputs a full signature σ_F on M under PK_i, PK_j, and $\mathit{Query}(D, O_{\mathsf{Res}})$ is the set of valid queries D issued to the resolution oracle O_{Res}. In this oracle query, D can arbitrarily choose a public key PK without knowing the corresponding private key. However, we do require that there exists a PPT algorithm to check the validity of the two key pairs output by D, i.e. if SK_b matches PK_b for $b = 0, 1$, or if (PK_b, SK_b) is a possible output of $\mathsf{Setup}^{\mathsf{User}}$. The advantage of D, $\mathtt{Adv}_D^{\mathrm{SA}}(k)$, is defined to be the gap between its success probability in the experiment above and $1/2$, i.e. $\mathtt{Adv}_D^{\mathrm{SA}}(k) = |\Pr[b' = b] - 1/2|$.

[2] By 'valid', we mean that σ_P is a valid partial signature on M under public keys PK_i, PK_j, alternatively, the input $(M, \sigma_P, PK_i, PK_j)$ of O_{Res} satisfies the condition that $\mathsf{PVer}(M, \sigma_P, \{PK_i, PK_j\}, APK) = \mathsf{accept}$.

Definition 2 (Signer Ambiguity). *An OFE scheme is said to be* signer ambiguous *if for any probabilistic polynomial-time algorithm D, $\mathtt{Adv}_D^{\mathrm{SA}}(k)$ is* negligible *in k.*

Remark 1. We note that a similar notion was introduced in [14, 22]. It's required that the signer's partial signature can be simulated in an indistinguishable way. However, the '*indistinguishability*' in [14, 22] is defined in CPA fashion, giving the adversary no oracle that resolves a partial signature to a full one, while our definition of signer ambiguity is done in the CCA fashion, allowing the adversary to ask for resolving any partial signature except the challenge one to a full signature, which is comparable to the CCA security of public key encryption schemes.

(Security Against Signers): We require that no PPT adversary A should be able to produce a partial signature with non-negligible probability, which looks good to a verifier but cannot be resolved to a full signature by the honest arbitrator. This ensures the fairness for verifiers, that is, if the signer has committed to a message with respect to an (honest) verifier, the verifier should always be able to obtain the full commitment of the signer. Formally, we consider the following experiment:

$$
\begin{aligned}
\mathsf{PM} &\leftarrow \mathsf{PMGen}(1^k)\\
(APK, ASK) &\leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM})\\
(PK_B, SK_B) &\leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)\\
(M, \sigma_P, PK_A) &\leftarrow A^{O^B_{\mathsf{PSig}}, O_{\mathsf{Res}}}(APK, PK_B)\\
\sigma_F &\leftarrow \mathsf{Res}(M, \sigma_P, ASK, \{PK_A, PK_B\})\\
\text{success of } A := [&\mathsf{PVer}(M, \sigma_P, \{PK_A, PK_B\}, APK) = \mathsf{accept}\\
&\wedge \mathsf{Ver}(M, \sigma_F, PK_A, PK_B, APK) = \mathsf{reject}\\
&\wedge (M, PK_A) \notin Query(A, O^B_{\mathsf{PSig}})]
\end{aligned}
$$

where oracle O_{Res} is described in the previous experiment, O^B_{PSig} takes as input (M, PK_i) and outputs a partial signature on M under PK_i, PK_B generated using SK_B, and $Query(A, O^B_{\mathsf{PSig}})$ is the set of queries made by A to oracle O^B_{PSig}. In this experiment, the adversary can arbitrarily choose a public key PK_i, and it may not know the corresponding private key of PK_i. Note that the adversary is not allowed to corrupt PK_B, otherwise it can easily succeed in the experiment by simply using SK_B to produce a partial signature under public keys PK_A, PK_B and outputting it. The advantage of A in the experiment $\mathtt{Adv}_A^{\mathrm{SAS}}(k)$ is defined to be A's success probability.

Definition 3 (Security Against Signers). *An OFE scheme is said to be* secure against signers *if there is* no *PPT adversary A such that $\mathtt{Adv}_A^{\mathrm{SAS}}(k)$ is* non-negligible *in k.*

(Security Against Verifiers): This security notion requires that any PPT verifier B should not be able to transform a partial signature into a full signature with non-negligible probability if no help has been obtained from the

signer or the arbitrator. This requirement has some similarity to the notion of *opacity* for verifiably encrypted signature [9]. Formally, we consider the following experiment:

$$\begin{aligned}
\mathsf{PM} &\leftarrow \mathsf{PMGen}(1^k)\\
(APK, ASK) &\leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM})\\
(PK_A, SK_A) &\leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)\\
(M, PK_B, \sigma_F) &\leftarrow B^{O_{\mathsf{PSig}}, O_{\mathsf{Res}}}(PK_A, APK)\\
\text{success of } B := &[\mathsf{Ver}(M, \sigma_F, PK_A, PK_B, APK) = \mathsf{accept} \wedge\\
&\quad (M, \cdot, \{PK_A, PK_B\}) \notin Query(B, O_{\mathsf{Res}})]
\end{aligned}$$

where oracle O_{Res} is described in the experiment of signer ambiguity, $Query(B, O_{\mathsf{Res}})$ is the set of valid queries B issued to the resolution oracle O_{Res}, and oracle O_{PSig} takes as input a message M and a public key PK_j and returns a valid partial signature σ_F on M under PK_A, PK_j generated using SK_A. In the experiment, B can ask the arbitrator for resolving any partial signature with respect to any pair of public keys (adaptively chosen by B, probably without the knowledge of the corresponding private keys), with the limitation described in the experiment. The advantage of B in the experiment $\mathtt{Adv}_B^{\mathrm{SAV}}(k)$ is defined to be B's success probability in the experiment above.

Definition 4 (Security Against Verifiers). *An OFE scheme is said to be* secure against verifiers *if there is* no *PPT adversary B such that $\mathtt{Adv}_B^{\mathrm{SAV}}(k)$ is* non-negligible *in k.*

(Security Against the Arbitrator): Intuitively, an OFE is secure against the arbitrator if no PPT adversary C including the arbitrator, should be able to generate with non-negligible probability a full signature without explicitly asking the signer for generating one. This ensures the fairness for signers, that is, no one can frame the actual signer on a message with a forgery. Formally, we consider the following experiment:

$$\begin{aligned}
\mathsf{PM} &\leftarrow \mathsf{PMGen}(1^k)\\
(APK, ASK^*) &\leftarrow C(\mathsf{PM})\\
(PK_A, SK_A) &\leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)\\
(M, PK_B, \sigma_F) &\leftarrow C^{O_{\mathsf{PSig}}}(ASK^*, APK, PK_A)\\
\text{success of } C := &[\mathsf{Ver}(M, \sigma_F, PK_A, PK_B, APK) = \mathsf{accept} \wedge\\
&\quad (M, PK_B) \notin Query(C, O_{\mathsf{PSig}})]
\end{aligned}$$

where the oracle O_{PSig} is described in the previous experiment, ASK^* is C's state information, which might not be the corresponding private key of APK, and $Query(C, O_{\mathsf{PSig}})$ is the set of queries C issued to the oracle O_{PSig}. The advantage of C in this experiment $\mathtt{Adv}_C^{\mathrm{SAA}}(k)$ is defined to be C's success probability.

Definition 5 (Security Against the Arbitrator). *An OFE scheme is said to be* secure against the arbitrator *if there is* no *PPT adversary C such that $\mathtt{Adv}_C^{\mathrm{SAA}}(k)$ is* non-negligible *in k.*

Remark 2. In A-OFE, both signer U_A and verifier U_B are equipped with public/secret key pairs (of the same structure), and U_A and U_B can generate indistinguishable partial signatures on the same message. If the security against the arbitrator holds for U_A (as described in the experiment above), it should also hold for U_B. That is, even when colluding with U_A (and other signers), the arbitrator should not be able to frame U_B for a full signature on a message, if it has not obtained a partial signature on the message generated by U_B.

Definition 6 (Secure Ambiguous Optimistic Fair Exchange). *An A-OFE scheme is said to be* secure *in the multi-user setting and chosen-key model if it is resolution ambiguous, signer ambiguous, secure against signers, secure against verifiers and secure against the arbitrator.*

2.2 Weaker Variants of the Model

In this section, we evaluate the relation between the signer ambiguity and security against verifiers. Intuitively, if an A-OFE scheme is not secure against verifiers, the scheme cannot be signer ambiguous because a malicious verifier can convert with non-negligible probability a signer's partial signature to a full one which allows the verifier to win the signer ambiguity game. For technical reasons, we first describe some weakened models before giving the proof for a theorem regarding the relation.

In our definition of signer ambiguity (Def. 2), the two public/secret key pairs are selected by the adversary D. In a weaker form, the key pairs can be selected by the challenger, and D is allowed to corrupt these two keys. This is comparable to the ambiguity definition for concurrent signature [11], or the strongest definition of anonymity of ring signature considered in [6], namely *anonymity against full key exposure*. We can also define an even weaker version of signer ambiguity, in which D is given two public keys, PK_A, PK_B, the oracle access of O_{PSig} which returns U_A's partial signatures, and is allowed to corrupt PK_B. We call this form of signer ambiguity as *weak signer ambiguity*.

In the definition of security against verifiers (Def. 4), the verifier's public key PK_B is adaptively selected by the adversary B. In a weaker model, PK_B can be generated by the challenger and the corresponding user secret key can be corrupted by B. The rest of the model remains unchanged. We call this as *weak security against verifiers*. Below we show that if an OFE scheme is weakly signer ambiguous and secure against the arbitrator, then it is also weakly secure against verifiers.

Theorem 1. *In A-OFE, weak signer ambiguity and security against the arbitrator (Def. 5) together imply weak security against verifiers.*

Proof. Suppose that an A-OFE scheme is not weakly secure against verifiers. Let B be the PPT adversary that has non-negligible advantage ϵ in the experiment of weak security against verifiers and B make at most q queries of the form $(\cdot, PK_B)$

to oracle O_{PSig}. Due to the security against the arbitrator, B must have queried O_{PSig} in the form $(\cdot, PK_B)$. Hence the value of q is at least one. Denote the experiment of weak security against verifiers by $\mathsf{Ex}^{(0)}$. Note that in $\mathsf{Ex}^{(0)}$ all queries to O_{PSig} are answered with partial signatures generated using SK_A. We now define a series of experiments, $\mathsf{Ex}^{(1)}, \cdots, \mathsf{Ex}^{(q)}$, so that $\mathsf{Ex}^{(i)}$ $(i \geq 1)$ is the same as $\mathsf{Ex}^{(i-1)}$ except that starting from the $(q+1-i)$-th query to O_{PSig} up to the q-th query of the form $(\cdot, PK_B)$, they are answered with partial signatures generated using SK_B. Let B's success probability in experiment $\mathsf{Ex}^{(i)}$ be ϵ_i. Note that $\epsilon_0 = \epsilon$, and in experiment $\mathsf{Ex}^{(q)}$ all queries of the form $(\cdot, PK_B)$ to O_{PSig} are answered with partial signatures generated using SK_B. Since B also knows SK_B (through corruption), it can use SK_B to generate partial signatures using SK_B on any message. Therefore, making queries of the form $(\cdot, PK_B)$ to O_{PSig} does not help B on winning the experiment if answers are generated using SK_B. It is equivalent to the case that B does not issue any query $(\cdot, PK_B)$ to O_{PSig}. Hence guaranteed by the security against the arbitrator, we have that B's advantage in $\mathsf{Ex}^{(q)}$ is negligible as B has to output a full signature without getting any corresponding partial signature.

Since the gap, $|\epsilon_0 - \epsilon_q|$, between B's advantage in $\mathsf{Ex}^{(0)}$ and that in $\mathsf{Ex}^{(q)}$ is non-negligible, there must exist an $1 \leq i \leq q$ such that $|\epsilon_{i-1} - \epsilon_i|$ is at least $|\epsilon_0 - \epsilon_q|/q$, which is non-negligible as well. Let i^* be such an i. We show how to make use of the difference of B's advantage in $\mathsf{Ex}^{(i^*-1)}$ and $\mathsf{Ex}^{(i^*)}$ to build a PPT algorithm D to break the weak signer ambiguity.

Given APK and PK_A, PK_B, D first asks its challenger for SK_B, and then invokes B on (APK, PK_A, PK_B). D randomly selects an i^* from $\{1, \cdots, q\}$, and simulates the oracles for B as follows. If B asks for SK_B, D simply gives it to B. The oracle O_{Res} is simulated by D using its own resolution oracle. If B makes a query (M, PK_j) to O_{PSig} where $PK_j \neq PK_B$, D forwards this query to its own partial signing oracle, and returns the obtained answer back to B. Now consider the ℓ-th query of the form (M, PK_B) made by B to O_{PSig}. If $\ell < q + 1 - i^*$, D forwards it to its own oracle, and returns the obtained answer. If $\ell = q + 1 - i^*$, D requests its challenger for the challenge partial signature σ_P^* on M and returns it to B. If $\ell > q + 1 - i^*$, D simply uses SK_B to produce a partial signature on M. At the end of the simluation, when B outputs (M^*, σ_F^*), if B succeeds in the experiment, D outputs 0; otherwise, D outputs 1.

It's easy to see that D guesses the correct i^* with probability at least $1/q$. Now suppose that D's guess of i^* is correct. If σ_P^* was generated by D's challenger using SK_A, i.e. $b = 0$, the view of B is identical to that in $\mathsf{Ex}^{(i^*-1)}$. On the other side, if σ_P^* was generated using SK_B, i.e. $b = 1$, the view of B is identical to that in $\mathsf{Ex}^{(i^*)}$. Let b' be the bit output by D. Since D outputs 0 only if B succeeds in the experiment, we have $\Pr[b' = 0 | b = 0] = \epsilon_{i^*-1}$ and $\Pr[b' = 0 | b = 1] = \epsilon_{i^*}$. Therefore, the advantage of D in attacking the weak signer ambiguity over random guess is

$$\begin{aligned}\left|\Pr[b'=b]-\frac{1}{2}\right| &= \left|\Pr[b'=0\wedge b=0]+\Pr[b'=1\wedge b=1]-\frac{1}{2}\right| \\ &= \left|\Pr[b'=0\wedge b=0]+\big(\Pr[b=1]-\Pr[b'=0\wedge b=1]\big)-\frac{1}{2}\right| \\ &= \frac{1}{2}\left|\Pr[b'=0|b=0]-\Pr[b'=0|b=1]\right| \\ &\geq \frac{1}{2q}\left|\epsilon_{i^*-1}-\epsilon_{i^*}\right| \geq \frac{1}{2q^2}\left|\epsilon_0-\epsilon_q\right|\end{aligned}$$

which is also non-negligible. This contradicts the weak signer ambiguity assumption. □

Corollary 1. *In A-OFE, signer ambiguity (Def. 2) and security against the arbitrator (Def. 5) together imply weak security against verifiers.*

Letting an adversary select the two challenge public keys gives the adversary more power in attacking signer ambiguity. Therefore, signer ambiguity defined in Sec. 2.1 is at least as strong as the weak signer ambiguity. Hence this corollary follows directly the theorem above.

3 Preliminaries

(Admissible Pairings): Let $\mathbb{G}_1$ and $\mathbb{G}_T$ be two cyclic groups of large prime order p. $\hat{\mathsf{e}}$ is an *admissible pairing* if $\hat{\mathsf{e}}:\mathbb{G}_1\times\mathbb{G}_1\rightarrow\mathbb{G}_T$ is a map with the following properties: (1) *Bilinear*: $\forall R,S\in\mathbb{G}_1$ and $\forall a,b\in\mathbb{Z}$, $\hat{\mathsf{e}}(R^a,S^b)=\hat{\mathsf{e}}(R,S)^{ab}$; (2) *Non-degenerate*: $\exists R,S\in\mathbb{G}_1$ such that $\hat{\mathsf{e}}(R,S)\neq 1$; and (3) *Computable*: there exists an efficient algorithm for computing $\hat{\mathsf{e}}(R,S)$ for any $R,S\in\mathbb{G}_1$.

(**Decision Linear Assumption (DLN)**[8]:) Let $\mathbb{G}_1$ be a cyclic group of large prime order p. The Decision Linear Assumption for $\mathbb{G}_1$ holds if for any PPT adversary $\mathcal{A}$, the following probability is negligibly close to 1/2.

$$\Pr[F,H,W\leftarrow\mathbb{G}_1; r,s\leftarrow\mathbb{Z}_p; Z_0\leftarrow W^{r+s}; Z_1\leftarrow\mathbb{G}_1; d\leftarrow\{0,1\}:\mathcal{A}(F,H,W,F^r,H^s,Z_d)=d]$$

(**q-Strong Diffie-Hellman Assumption (q-SDH)**[7]): The q-SDH problem in $\mathbb{G}_1$ is defined as follows: given a $(q+1)$-tuple $(g,g^x,g^{x^2},\cdots,g^{x^q})$, output a pair $(g^{1/(x+c)},c)$ where $c\in\mathbb{Z}_p^*$. The q-SDH assumption holds if for any PPT adversary $\mathcal{A}$, the following probability is negligible.

$$\Pr\left[x\leftarrow\mathbb{Z}_p^*:\mathcal{A}(g,g^x,\cdots,g^{x^q})=(g^{\frac{1}{x+c}},c)\right]$$

4 Ambiguous OFE without Random Oracles

In this section, we propose an A-OFE scheme, which is based on Groth and Sahai's idea of constructing a fully anonymous group signature scheme [15, 16]. Before describing the scheme, we first describe our construction in a high level.

4.1 High Level Description of Our Construction

As mentioned in the introduction part, many OFE schemes in the literature follows a generic framework: Alice encrypts her signature under the arbitrator's public key, and then provides a proof showing that the ciphertext indeed contains her signature on the message. To extend this framework to ambiguous optimistic fair exchange, we let Alice encrypt her signature under the arbitrator's public key and provide a proof showing that the ciphertext contains either her signature on the message or Bob's signature on it. Therefore, given Alice's partial signature, Bob cannot convince others that Alice was committed herself to something, as he can also generate this signature.

Our concrete construction below follows the aforementioned framework, which is based on the idea of Groth in constructing a fully anonymous group signature scheme [15]. In more details, Alice's signature consists of a weakly secure BB-signature [7] and a strong one-time signature. Since only the BB-signature is related to Alice's identity, we encrypt it under the arbitrator's public key using Kiltz' tag-based encryption scheme [20], with the one-time verification key as the tag. The non-interactive proof is based on a newly developed technique by Groth and Sahai [16], which is efficient and doesn't require any complex NP-reduction. The proof consists of two parts. The first part includes a commitment to Alice's BB-signature along with a non-interactive witness indistinguishable (NIWI) proof showing that either Alice's BB-signature or Bob's BB-signature on the one-time verification key is in the commitment. The second part is non-interactive zero-knowledge (NIZK) proof (of knowledge) showing that the commitment and the ciphertext contains the same thing. These two parts together imply that the ciphertext contains a BB-signature on the message generated by either Alice or Bob. Both the ciphertext and the proof are authenticated using the one-time signing key. Guaranteed by the strong unforgeability of the one-time signature, no efficient adversary can modify the ciphertext or the proof.

The NIWI proof system consists of four (PPT) algorithms, K_{NI}, P_{WI}, V_{WI} and X_{xk}, where K_{NI} is the key generation algorithm which outputs a common reference string $\mathtt{crs}$ and an extraction key xk; P_{WI} takes as input $\mathtt{crs}$, the statement to be proved x, and a corresponding witness w, and outputs a proof π; V_{WI} is the corresponding verification algorithm; and X_{xk} takes as input $\mathtt{crs}$ and a valid proof π, outputs a witness w'. The NIZK proof shares the same common reference string with the NIWI proof. P_{ZK} and V_{ZK} are the proving and verification algorithms of the NIZK proof system respectively. Due to the page limit, we refer readers to [16] for detained information about the non-interactive proofs and to [15] for an introduction to the building tools needed for our construction.

4.2 The Scheme

Now we propose our A-OFE scheme. It works as follows:

- PMGen takes 1^k and outputs $\mathsf{PM} = (1^k, p, \mathbb{G}_1, \mathbb{G}_T, \hat{\mathsf{e}}, g)$ so that $\mathbb{G}_1$ and $\mathbb{G}_T$ are cyclic groups of prime order p; g is a random generator of $\mathbb{G}_1$; $\hat{\mathsf{e}} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow$

$\mathbb{G}_T$ is an admissible bilinear pairing; and group operations on $\mathbb{G}_1$ and $\mathbb{G}_T$ can be efficiently performed.

- $\mathsf{Setup}^{\mathsf{TTP}}$: The arbitrator runs the key generation algorithm of the non-interactive proof system to generate a common reference string $\mathtt{crs}$ and an extraction key xk, i.e. $(\mathtt{crs}, xk) \leftarrow K_{NI}(1^k)$, where $\mathtt{crs} = (F, H, U, V, W, U', V', W')$. It also randomly selects $K, L \leftarrow \mathbb{G}_1$, and sets $(APK, ASK) = ((\mathtt{crs}, K, L), xk)$, where F, H, K, L together form the public key of the tag-based encryption scheme [20], and xk is the extraction key of the NIWI proof system [15, 16], which is also the decryption key of the tag-based encryption scheme.
- $\mathsf{Setup}^{\mathsf{User}}$: Each user U_i randomly selects $x_i \leftarrow \mathbb{Z}_p$, and sets $(PK_i, SK_i) = (g^{x_i}, x_i)$.
- PSig: To partially sign a message m with verifier U_j, user U_i does the following:
  1. call the key generation algorithm of $\mathcal{S}$ to generate a one-time key pair $(otvk, otsk)$;
  2. use SK_i to compute a BB-signature $\overline{\sigma}$ on $\mathtt{H}(otvk)$, i.e. $\overline{\sigma} \leftarrow g^{\frac{1}{x_i + \mathtt{H}(otvk)}}$;
  3. compute an NIWI proof π_1 showing that $\overline{\sigma}$ is a valid signature under either PK_i or PK_j, i.e. $\pi_1 \leftarrow P_{WI}(\mathtt{crs}, (\hat{\mathsf{e}}(g,g), PK_i, PK_j, \mathtt{H}(otvk)), (\overline{\sigma}))$, which shows that the following holds:
  $$\hat{\mathsf{e}}(\overline{\sigma}, PK_i \cdot g^{\mathtt{H}(otvk)}) = \hat{\mathsf{e}}(g,g) \vee \hat{\mathsf{e}}(\overline{\sigma}, PK_j \cdot g^{\mathtt{H}(otvk)}) = \hat{\mathsf{e}}(g,g)$$
  4. compute a tag-based encryption ([20]) y of $\overline{\sigma}$, i.e. $y = (y_1, y_2, y_3, y_4, y_5) \leftarrow \mathcal{E}.E_{pk}(\overline{\sigma}, \mathsf{tag})$, where $pk = (F, H, K, L)$ and $\mathsf{tag} = \mathtt{H}(otvk)$;
  5. compute an NIZK proof π_2 showing that y and the commitment C to $\overline{\sigma}$ in π_1 contain the same $\overline{\sigma}$, i.e. $\pi_2 \leftarrow P_{ZK}(\mathtt{crs}, (y, \pi_1), (r, s, t))$;
  6. use $otsk$ to sign the whole transcript and the message M, i.e. $\sigma_{ot} \leftarrow \mathcal{S}.S_{otsk}(M, \pi_1, y, \pi_2)$.

  The partial signature σ_P of U_i on message M then consists of $(otvk, \sigma_{ot}, \pi_1, y, \pi_2)$.
- PVer: After obtaining U_i's partial signature $\sigma_P = (otvk, \sigma_{ot}, \pi_1, y, \pi_2)$, the verifier U_j checks the following. If any one fails, U_j rejects; otherwise, it accepts.
  1. if σ_{ot} is a valid one-time signature on (M, π_1, y, π_2) under $otvk$;
  2. if π_1 is a valid NIWI proof, i.e. $V_{WI}(\mathtt{crs}, (\hat{\mathsf{e}}(g,g), PK_i, PK_j, \mathtt{H}(otvk)), \pi_1) \stackrel{?}{=} \mathsf{accept}$;
  3. if π_2 is a valid NIZK proof, i.e. $V_{ZK}(\mathtt{crs}, (y, \pi_1), \pi_2) \stackrel{?}{=} \mathsf{accept}$;
- Sig: To sign a message M with verifier U_j, user U_i generates a partial signature σ_P as in PSig, and set the full signature σ_F as $\sigma_F = (\sigma_P, \overline{\sigma})$.
- Ver: After receiving σ_F on M from U_i, user U_j checks if $\mathsf{PVer}(M, \sigma_P, \{PK_i, PK_j\}, APK) \stackrel{?}{=} \mathsf{accept}$, and if $\hat{\mathsf{e}}(\overline{\sigma}, PK_i \cdot g^{\mathtt{H}(otvk)}) \stackrel{?}{=} \hat{\mathsf{e}}(g,g)$. If any of the checks fails, U_j rejects; otherwise, it accepts.
- Res: After receiving U_i's partial signature σ_P on message M from user U_j, the arbitrator firstly checks the validity of σ_P. If invalid, it returns $\perp$ to U_j. Otherwise, it extracts $\overline{\sigma}$ from π_1 by calling $\overline{\sigma} \leftarrow X_{xk}(\mathtt{crs}, \pi_1)$. The arbitrator returns $\overline{\sigma}$ to U_j.

5 Security Analysis

Theorem 2. *The proposed A-OFE scheme is secure in the multi-user setting and chosen-key model (without random oracle) provided that DLN assumption and q-SDH assumption hold.*

Intuitively, the resolution ambiguity is guaranteed by the extractability and soundness of the NIWI proof of knowledge system. The signer ambiguity and security against verifiers are due to the CCA security of the encryption scheme. Security against signers and security against the arbitrator are guaranteed by the (weak) unforgeability of BB-signature scheme. Due to the page limit, we leave the detailed proof in the full version of this paper.

Remark 3. In our construction, the signer uses its secret key to generate a BB-signature on a fresh one-time verification key, while the message is signed using the corresponding one-time signing key. As shown by Huang et al. in [17], this combination leads to a strongly unforgeable signature scheme. It's not hard to see that our proposed A-OFE scheme actually achieves a stronger version of security against the verifier. That is, even if the adversary sees the signer U_A's full signature σ_F on a message M with verifier U_B, it cannot generate another σ'_F on M such that $\mathsf{Ver}(M, \sigma'_F, PK_A, PK_B, APK) = \mathsf{accept}$. The claim can be shown using the proof given in this paper without much modification.

(*Comparison*): We note that schemes proposed in [14, 22] have similar properties as our ambiguous OFE, i.e. (online, offline) *non-transferability*. Here we make a brief comparison with these two schemes. First of all, our A-OFE scheme is better than them in terms of the level of non-transferability. In [14, 22], the non-transferability is defined only in the CPA fashion. The adversary is not given an oracle for converting a partial signature to a full one. While in our definition of A-OFE, we define the ambiguity in the CCA fashion, allowing the adversary to ask for resolving a partial signature to a full one. Second, in terms of efficiency, our scheme outperforms the scheme proposed in [22], and is slightly slower than [14]. The generation of a partial signature of their scheme requires linear (in security parameter k) number of encryptions, and the size of a partial signature is also linear in k. While in our scheme both the computation cost and size of a partial signature are constant. The partial signature of ousr scheme includes about 41 group elements plus a one-time verification key and a one-time signature. Third, both our scheme and the scheme in [14] only require one move in generating a partial signature, while the scheme in [22] requires four moves. Fourth, in [22], there is a setup phase between each signer and the confirmer, in which the confirmer generates an encryption key pair for each signer. Therefore, the confirmer has to store a key pair for each signer, leading to a large storage. While our scheme and [14] don't need such a phase. Fifth, in terms of security, our scheme and [22] are provably secure without random oracles. But the scheme in [14] is only provably secure in the random oracle model.

Acknowledgements

We are grateful to the anonymous reviewers of Asiacrypt 2008 for their invaluable comments. The first three authors were supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (RGC Ref. No. CityU 122107).

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: CCS, pp. 7–17. ACM, 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. Bao, F., Wang, G., Zhou, J., Zhu, H.: Analysis and improvement of Micali's fair contract signing protocol. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 176–187. Springer, Heidelberg (2004)
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM CCS, pp. 62–73. ACM, New York (1993)
6. 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), `http://eprint.iacr.org/`
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., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
9. 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)
10. Boyd, C., Foo, E.: Off-line fair payment protocols using convertible signatures. In: Ohta, K., Pei, D. (eds.) ASIACRYPT 1998. LNCS, vol. 1514, pp. 271–285. Springer, Heidelberg (1998)
11. Chen, L., Kudla, C., Paterson, K.G.: Concurrent signatures. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 287–305. Springer, Heidelberg (2004)
12. 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)
13. Dodis, Y., Reyzin, L.: Breaking and repairing optimistic fair exchange from PODC 2003. In: DRM 2003, pp. 47–54. ACM, New York (2003)
14. 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)
15. Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 164–180. Springer, Heidelberg (2007)

16. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
17. Huang, Q., Wong, D.S., Li, J., Zhao, Y.: Generic transformation from weakly to strongly unforgeable signatures. Journal of Computer Science and Technology 23(2), 240–252 (2008)
18. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Efficient 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)
19. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
20. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
21. Kremer, S.: Formal Analysis of Optimistic Fair Exchange Protocols. PhD thesis, Université Libre de Bruxelles (2003)
22. Liskov, M., Micali, S.: Online-untransferable signatures. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 248–267. Springer, Heidelberg (2008)
23. Micali, S.: Simple and fast optimistic protocols for fair electronic exchange. In: PODC 2003, pp. 12–19. ACM, New York (2003)
24. Park, J.M., Chong, E.K., Siegel, H.J.: Constructing fair-exchange protocols for e-commerce via distributed computation of RSA signatures. In: PODC 2003, pp. 172–181. ACM, New York (2003)
25. 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)
26. 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)