Ambiguous optimistic fair exchange: Definition and constructions

Qiong HUANG

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Duncan S. WONG

Willy SUSILO

Theoretical Computer Science 562 (2015) 177–193

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Ambiguous optimistic fair exchange: Definition and constructions ☆

Qiong Huang [a,*], Guomin Yang [c], Duncan S. Wong [b], Willy Susilo [c]

[a] *College of Informatics, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China*
[b] *Department of Computer Science, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong Special Administrative Region, China*
[c] *School of Computer Science and Software Engineering, University of Wollongong, Northfields Avenue, Wollongong, Australia*

ARTICLE INFO

Article history:
Received 2 June 2013
Received in revised form 15 July 2014
Accepted 23 September 2014
Available online 30 September 2014
Communicated by G. Ausiello

Keywords:
Optimistic fair exchange
Ambiguity
Signature
NIZK proof
Standard model

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* (AOFE), which is a variant of OFE and requires additionally that the verifier cannot convince anybody about the authorship of a partial signature generated by the signer. We present a formal security model for AOFE in the multi-user setting and chosen-key model, and propose a generic construction of AOFE that is provably secure under our model. Furthermore, we propose an efficient instantiation of the generic construction, security of which is based on Strong Diffie–Hellman assumption and Decision Linear assumption without random oracles.

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 is only called in 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 [2,16,3,11,33,14,29,32,38,4,34,15,21,35]. For all recently proposed schemes, an OFE protocol for signature

☆ A preliminary version of this paper appeared in ASIACRYPT 2008 [20]. This is a vastly extended version. This work is supported by the National Natural Science Foundation of China (Nos. 61103232, 61472146), the Guangdong Natural Science Foundation (No. S2013010011859) and the Research Fund for the Doctoral Program of Higher Education of China (No. 20114404120027). D.S. Wong is supported by a grant from the RGC of the HKSAR, China (Project No. CityU 121512). W. Susilo is supported by ARC Future Fellowship FT0991397.

* Corresponding author.
E-mail addresses: csqhuang-c@my.cityu.edu.hk (Q. Huang), gyang@uow.edu.au (G. Yang), duncan@cityu.edu.hk (D.S. Wong), wsusilo@uow.edu.au (W. Susilo).

http://dx.doi.org/10.1016/j.tcs.2014.09.043
0304-3975/

typically consists of three message flows. The initiator of OFE, Alice, first sends a *partial signature* σ_P to a responder, Bob, where σ_P is considered as Alice's partial commitment to her full signature which will be sent to Bob. But beforehand, Bob should send his full signature back to Alice first in the second message flow. After receiving Bob's full signature, Alice then sends her full signature to Bob in the third message flow. If Bob refuses to send his full signature to Alice in the second message flow, 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 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 by 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 Alice generates σ_P, 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 [15,21], the partial signature of Alice is a standard signature, which can only be generated by Alice. In many other OFE schemes, 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, regarding the validity and non-repudiation of a signature, as pointed out by Boyd and Foo [10], this raises the question of whether a non-interactive proof that a signature is encrypted is really having any difference from a signature itself, as the proof is already sufficient to convince any third party that the signer has committed to the message.

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, 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. For example, Alice wants to sign with Bob a contract of procuring Bob's company. After sending out her partial signature, Alice has no way to regret and cannot withdraw the procurement if Bob persists. However, Bob can pause the contract signing, and use Alice's partial signature to bargain for better offers with others. He then carries out a new OFE protocol with the one offering the best price to sign the contract. Bob can play the same trick iteratively until that no one can give an even better offer.

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 [21]), 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* (AOFE). It inherits all the formalized properties of OFE [15,21] 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.

1.1. Related works

There have been many OFE schemes proposed in the past [2,3,11,33,14,29,32,38,4,34,15,21]. In the following, we review some recent ones by starting from 2003 when Park, Chong and Siegel [33] proposed an OFE based on sequential two-party multi-signature. It was later broken and repaired by Dodis and Reyzin [14]. The scheme is *setup-driven* [39,40], which requires all users to register their keys with the arbitrator prior to conducting any transaction. In [32], 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 [15] 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. [15] 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 [21] 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 [15]) are proven in a more restricted model, called *certified-key* model, which requires the adversary to prove its knowledge of the corresponding private key before using a public key. In [21], 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 relying on the random oracle assumption.

In [16], 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.

Liskov and Micali [30] proposed an *online-untransferable signature* scheme, which can be considered as an enhanced version of designated confirmer signature. In such a scheme, there is also a party (confirmer) semi-trusted by both the signer and the recipient. A dishonest recipient, who is interacting with a signer, cannot convince a third party that the signature is generated by the signer. But both the signer and the conformer are able to convert a signature so that anyone can identity its owner. The online non-transferability of their scheme is similar to the *signer ambiguity* (see Definition 2) of AOFE. However, the *online attack* considered in [30] would not happen in AOFE, as the signature generation and verification are both non-interactive. Besides, the signing process of their scheme requires several rounds of interaction with the recipient, and the 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.

Works after [20]. There have been multiple works since the introduction of AOFE. To name a few, Chen et al. [13] introduced a new notion of *Verifiable Encryption of Chameleon Signatures*, and used it to construct a three-round abuse-free optimistic protocol for contract signing. A somewhat generic transform from *designated confirmer signature* (DCS) to AOFE was proposed in [22,23]. The underlying DCS scheme is required to enjoy a property named *samplability*, while is satisfied by only a few DCS schemes. The transform was later refined in [25], where the DCS is only required to satisfy standard properties, e.g. unforgeability and anonymity. A new notion called *group-oriented optimistic fair exchange* was proposed in [24], which considers the fair exchange between two groups of users, keeping the anonymity of the signer in its group. Another enhanced version of AOFE, named *Perfect* AOFE, was proposed in [36], in which a partial signature leaks no information about the actual signer or the intended verifier. This is useful for applications where the involved parties of an exchange wish to further protect their privacy on whether they are indeed involved in an exchange or not. This notion was further improved very recently. *Privacy-preserving* OFE was proposed in [26], in which the arbitrator could not learn the full signature even after the resolution process. Very recently, another variant of AOFE called *Attributed-based Optimistic Fair Exchange*, was introduced in [37], which integrates the advantage of both AOFE and (Ciphertext-Policy) Attributed-Based Encryption. Only the verifier who possesses appropriate credentials (issued by a credential center according to its attributes) can convert the signer's partial signature into a full one. It is worthy to notice that the schemes proposed in [22,23,25,26] are secure in the certified-key model, while the schemes proposed in this work and [37] are secure in the chosen-key model.

1.2. Our work

In this paper we make the following contributions. First, we propose the notion of *Ambiguous Optimistic Fair Exchange* (Ambiguous OFE, or AOFE in short) which allows a signer Alice to generate a partial signature in such a way that a verifier Bob cannot convince anyone 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 AOFE.

Second, for formalizing AOFE, 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 (the ambiguity considered in [15,21]), security against signers, security against verifiers and security against the arbitrator, AOFE has an additional requirement: *signer ambiguity*. It requires that the verifier can generate partial signatures which are (computationally) indistinguishable from the 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 form of) signer ambiguity, then it has (a weaker form of) security against verifiers.

Third, we revisit two generic methods in constructing OFE [15,21], and show that if we simply extend them to the construction of AOFE, the resulting schemes would be *insecure*. Specifically, they are insecure against the arbitrator under our proposed model.

Fourth, we propose a generic AOFE construction. It is based on the generic OFE construction of [15], but instead of using a CCA secure encryption scheme, an existentially unforgeable signature scheme under chosen message attacks and a simulation-sound NIZK proof system, we employ a selective-tag weakly CCA secure tag-based encryption [28], a weakly unforgeable signature [7], a strong one-time signature and a general NIZK proof system. The security of the generic construction is proven secure under our proposed multi-user setting and chosen-key model.

Last but not least, we propose a concrete and efficient instantiation of our generic construction of AOFE, the security of which is based on the intractability of Strong Diffie–Hellman problem and Decision Linear problem without random oracles.

Differences from [20]. In this paper we make the following changes, compared with [20]. First of all, the model of signer ambiguity is revised to correct some minor issue in [20]. Second, the proof of the theorem on the relation between signer ambiguity and security against verifiers (Theorem 1) is revised to improve the reduction factor. Third, a new section (Section 3) is added to discuss about the insecurity of simple extensions of two generic constructions of OFE to AOFE. Fourth, a generic construction of AOFE is proposed (in Section 4) with detailed security proofs, which covers the efficient construction proposed in [20].

1.3. Paper organization

In the next section, we define AOFE and propose a security model. We then show some relation among the formalized security requirements. A popular generic construction used in building OFE schemes is revisited in Section 4, and a new generic construction of AOFE is proposed and proved. In Section 5 we give an efficient instantiation of the construction, security of which does not rely on the random oracle heuristic. The paper is concluded in Section 6.

2. Ambiguous optimistic fair exchange

In AOFE, 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 to σ_P. This property is analogous to the non-transferability of designated verifier signature [27] and the ambiguity of concurrent signature [12]. Also, the AOFE verification algorithm should take the public keys of *both* signer and (designated) verifier as inputs, in contrast to that in the traditional definition of OFE [2,15,21].

Definition 1 *(Ambiguous Optimistic Fair Exchange).* A (non-interactive) *Ambiguous Optimistic Fair Exchange* (*AOFE*) scheme involves two users, i.e. a signer and a verifier, and an arbitrator, and consists of the following probabilistic polynomial-time (PPT) algorithms:

- PMGen: On input 1^k where k is a security parameter, it outputs the 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 1 or 0, 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 1 or 0, 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.

It is required that there is an efficient algorithm which given a pair (PK, SK), verifies if SK matches PK, i.e. (PK, SK) is an output of algorithm $\mathsf{Setup}^{\mathsf{User}}$. As in [15], PSig together with Res should be functionally equivalent to Sig.

Correctness: 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 $M \in \{0,1\}^*$, let $\mathbf{PK} = \{PK_i, PK_j\}, \sigma_P \leftarrow \mathsf{PSig}(M, SK_i, PK_i, PK_j, APK)$, $\sigma_F \leftarrow \mathsf{Sig}(M, SK_i, PK_i, PK_j, APK)$, we have:

$$\mathsf{PVer}(M, \sigma_P, \mathbf{PK}, APK) = 1,$$

$$\mathsf{Ver}(M, \sigma_F, PK_i, PK_j, APK) = 1,$$

$$\text{and} \quad \mathsf{Ver}\big(M, \mathsf{Res}(M, \sigma_P, ASK, \mathbf{PK}), PK_i, PK_j, APK\big) = 1.$$

2.1. Security properties

Chosen-key model. We consider the security of AOFE in the multi-user setting [15] and the chosen-key model, which was introduced by Lysyanskaya et al. [31] in the context of sequential aggregate signature. In the chosen-key model, the adversary can choose any public key, except that the challenge/target public key(s) should be honestly generated. We also require that there exists an efficient algorithm for checking the validity of the challenge key pair(s) output by the adversary, i.e. if (PK, SK) is a possible output of $\mathsf{Setup}^{\mathsf{User}}$.

Resolution ambiguity. During an exchange, the verifier may be unable to receive the signer's full signature due to the poor internet connection. In this case the verifier could ask the arbitrator to resolve the signer's partial signature that it has already received. If the full signature output by algorithm Res has a different structure from that output by algorithm Sig, it may bring bad effect to the credit of the signer, as others may think the signer was cheating in the exchange. To protect the interest of the signer, we need *resolution ambiguity*[1] in this case.

Resolution ambiguity requires that the 'resolved signatures', e.g. $\mathsf{Res}(M, \mathsf{PSig}(M, SK_i, PK_i, PK_j, APK), ASK, \{PK_i, PK_j\})$, output by the arbitrator should be (computationally) *indistinguishable* from the 'actual signatures' generated by the signer, e.g. $\mathsf{Sig}(M, SK_i, PK_i, PK_j, APK)$. We stress that there are some cases in which a signer who is unable or refuses to return its full commitment, should be accounted, then the resolution ambiguity is not required any more, i.e. the OFE scheme should be accountable.

Signer ambiguity. Informally, given a partial signature σ_P from a signer A, a verifier B should not be able to convince others that σ_P was generated by A. To capture this, we borrow the idea of defining the *ambiguity* in concurrent signatures [12], and require that B should be able to simulate partial signatures that look *indistinguishable* from those generated by A. We need the existence of a simulation algorithm FPSig, that takes as input $(M, SK_B, PK_A, PK_B, APK)$ and outputs a partial signature σ_P that is valid under PK_A, PK_B. This is also the reason why a verifier should be equipped with a public/secret key pair, and its 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.

$$\begin{aligned}
&\mathsf{PM} \leftarrow \mathsf{PMGen}\big(1^k\big)\\
&(APK, ASK) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM})\\
&\big(M, \big\{(PK_i, SK_i)\big\}_{i\in\{A,B\}}, st\big) \leftarrow D^{O_{\mathsf{Res}}}(APK)\\
&b \leftarrow \{0,1\}\\
&\sigma_P \leftarrow \begin{cases} \mathsf{PSig}(M, SK_A, PK_A, PK_B, APK) & \text{if } b=0\\ \mathsf{FPSig}(M, SK_B, PK_A, PK_B, APK) & \text{if } b=1\end{cases}\\
&b' \leftarrow D^{O_{\mathsf{Res}}}(st, \sigma_P)\\
&\text{Succ. of } D := \big[b' = b \wedge \big(M, \sigma_P, \{PK_A, PK_B\}\big) \notin \mathit{Query}(D, O_{\mathsf{Res}})\big]
\end{aligned}$$

where st is the state information of D; oracle O_{Res} takes as input a valid 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 that D issued to oracle O_{Res}. The advantage of D, denoted by $\mathtt{Adv}_D^{\mathsf{SA}}(k)$, is defined as the gap between its success probability in the experiment above and 1/2, i.e. $\mathtt{Adv}_D^{\mathsf{SA}}(k) = |\Pr[b' = b] - 1/2|$.

Definition 2 *(Signer ambiguity).* An AOFE scheme is *signer ambiguous* if for any PPT algorithm D, $\mathtt{Adv}_D^{\mathsf{SA}}(k)$ is negligible in k.

Remark 1. A similar notion introduced in [16,30] requires that the signer's partial signature can be simulated in an indistinguishable way. However, their '*indistinguishability*' [16,30] is defined in the CPA fashion, i.e. the adversary has no oracle access for resolving partial signatures; while our definition of signer ambiguity is of CCA type, i.e. the adversary has access to O_{Res}.

We also remark that this level of signer ambiguity may be the best that we can get. In Definition 2, D is required to output well-formed key pairs for both A and B. If, for example, PK_B is maliciously chosen by D so that it is the hash of PK_A, then no one probably knows the corresponding secret key SK_B, and there is no way to ensure that the verifier can simulate the signer's partial signatures. As far as we know, the definition of anonymity of ring signatures also imposes a similar requirement. In the definition of *anonymity w.r.t. adversarially-chosen keys* and *anonymity against full key exposure* [6], the two target key pairs, (PK_{i_0}, SK_{i_0}) and (PK_{i_1}, SK_{i_1}), are required to be well-formed/honestly generated. The difference is that the two target key pairs in [6] are prepared by the challenger, while in Definition 2 they are chosen by the distinguisher. It is readily seen that our definition of signer ambiguity is stronger (or at least not weaker).

Security against signers. It requires 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 arbitrator. This ensures the fairness for verifiers, that is, if the signer has committed to a message w.r.t an (honest) verifier, the verifier should always be able to get the signer's full commitment. Formally, we consider the following experiment:

$$\begin{aligned}
\mathsf{PM} &\leftarrow \mathsf{PMGen}\big(1^k\big)\\
(APK, ASK) &\leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM})\\
(PK_B, SK_B) &\leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, APK)
\end{aligned}$$

[1] Resolution ambiguity is the same as the *ambiguity* defined in [15,21]. Here we rename it in order to avoid any confusion, as we will define another kind of ambiguity, e.g. *signer ambiguity*.

$$
\begin{aligned}
(M, \sigma_P, PK_A) &\leftarrow A^{O^B_{\mathsf{PSig}}, O_{\mathsf{Res}}}(APK, PK_B) \\
\sigma_F &\leftarrow \mathsf{Res}\big(M, \sigma_P, ASK, \{PK_A, PK_B\}\big) \\
\text{Succ. of } A &:= \big[\mathsf{PVer}\big(M, \sigma_P, \{PK_A, PK_B\}, APK\big) = 1 \wedge \mathsf{Ver}(M, \sigma_F, PK_A, PK_B, APK) \\
&\qquad = 0 \wedge (M, PK_A) \notin Query\big(A, O^B_{\mathsf{PSig}}\big)\big]
\end{aligned}
$$

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 valid 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}. 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. The advantage of A in the experiment, denoted by $\mathtt{Adv}^{\mathsf{SAS}}_A(k)$, is defined to be A's success probability.

Definition 3 *(Security against signers).* An AOFE scheme is *secure against signers* if there is no PPT adversary A such that $\mathtt{Adv}^{\mathsf{SAS}}_A(k)$ 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 [8]. Formally, we consider the following experiment:

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

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_P 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, with the limitation described in the experiment. The advantage of B in the experiment, denoted by $\mathtt{Adv}^{\mathsf{SAV}}_B(k)$, is defined to be B's success probability in the experiment above.

Definition 4 *(Security against verifiers).* An AOFE scheme is *secure against verifiers* if there is *no* PPT adversary B such that $\mathtt{Adv}^{\mathsf{SAV}}_B(k)$ is *non-negligible* in k.

Security against the arbitrator. Intuitively, security against the arbitrator requires that no PPT adversary C including the arbitrator, should be able to generate with non-negligible probability a valid full signature without explicitly asking the signer to do so. It 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:

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

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, denoted by $\mathtt{Adv}^{\mathsf{SAA}}_C(k)$, is defined to be C's success probability.

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

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

Definition 6 *(Secure AOFE).* An AOFE scheme is *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. Weak variants of security models

Intuitively, if an AOFE scheme is not secure against verifiers, the scheme cannot be signer ambiguous, because a malicious verifier can convert 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 below. In the definition of signer ambiguity (Definition 2), the two public/secret key pairs are selected by D. In a slightly weaker variant, the two key pairs are selected by the challenger, and then given to D. This is comparable to the ambiguity definition of concurrent signature [12], 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 (PK_A, PK_B, SK_B) and oracle access to O_{PSig}. We call this variant *the weak signer ambiguity*.

In the definition of security against verifiers (Definition 4), the verifier's public key PK_B is adaptively selected by the adversary B. In a weaker variant, the challenger selects (PK_B, SK_B) and gives the pair to B. The rest of the model remains unchanged. We call this variant *weak security against verifiers*. Below we show that if an AOFE scheme is weakly signer ambiguous and secure against the arbitrator, then it is weakly secure against verifiers.

Theorem 1. *In AOFE, weak signer ambiguity and security against the arbitrator (Definition 5) together imply weak security against verifiers.*

Proof. Suppose that an AOFE 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 after making at most q queries of the form $(\cdot, PK_B)$ to oracle O_{PSig}. By the security against the arbitrator, with overwhelming probability B has 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 the $(q+1-i)$-th, $\cdots$, q-th queries of the form $(\cdot, PK_B)$ submitted to O_{PSig} are answered with partial signatures generated using SK_B. Let B's success probability in $\mathsf{Ex}^{(i)}$ be ϵ_i. Note that $\epsilon_0 = \epsilon$, and in $\mathsf{Ex}^{(q)}$ all queries $(\cdot, PK_B)$ to O_{PSig} are answered with partial signatures generated using SK_B. Since B also knows SK_B (via corruption), it can use SK_B to generate partial signatures using SK_B on any message. Therefore, making queries $(\cdot, PK_B)$ to O_{PSig} does not help B on winning the experiment if the 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.

The non-negligible gap, $\epsilon_0 - \epsilon_q$, between B's advantage in $\mathsf{Ex}^{(0)}$ and that in $\mathsf{Ex}^{(q)}$ translates into a non-negligible gap between B's advantage in a pair of neighboring hybrid experiments. We construct a PPT algorithm D that uses B as a subroutine to break the weak signer ambiguity.

Given (APK, PK_A, PK_B, SK_B), algorithm D selects i uniformly from $\{1, \cdots, q\}$, invokes B on input (APK, PK_A, PK_B, SK_B), and then simulates the oracles for 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 the obtained signature to B. If $\ell > q+1-i$, D simply uses SK_B to produce a partial signature on M. At the end of the simulation, when B outputs (M^*, σ_F^*). If B succeeds in the experiment, D outputs $b' = 0$; otherwise, D outputs a random bit b'.

If D's challenger uses SK_A and follows PSig algorithm to produce σ_P^*, i.e. $b = 0$, the view of B is identical to that in $\mathsf{Ex}^{(i-1)}$ and the probability that D outputs $b' = 0$ is

$$\bar{\epsilon}_i \stackrel{\text{def}}{=} \epsilon_{i-1} + \frac{1}{2}(1 - \epsilon_{i-1}) = \frac{1}{2} + \frac{1}{2}\epsilon_{i-1}.$$

If the challenger uses SK_B and follows the simulation algorithm FPSig to produce σ_P^*, i.e. $b = 1$, the view of B is identical to that in $\mathsf{Ex}^{(i)}$, and D outputs $b' = 0$ with probability

$$\underline{\epsilon}_i \stackrel{\text{def}}{=} \epsilon_i + \frac{1}{2}(1 - \epsilon_i) = \frac{1}{2} + \frac{1}{2}\epsilon_i.$$

Denote by $\Pr[b'=0|b=0]$ (resp. $\Pr[b'=0|b=1]$) the probability that D outputs $b'=0$ when σ_P^* is output by algorithm Sig (resp. FPSig). Since the index i is uniformly distributed in $\{1,\cdots,q\}$, we have that

$$\begin{aligned}\Pr\big[b'=b\big] &= \Pr\big[b'=0\wedge b=0\big]+\Pr\big[b'=1\wedge b=1\big]\\ &=\frac{1}{2}+\frac{1}{2}\big(\Pr\big[b'=0|b=0\big]-\Pr\big[b'=0|b=1\big]\big)\\ &=\frac{1}{2}+\frac{1}{2}\left(\frac{1}{q}\sum_{j=1}^{q}\bar{\epsilon}_j-\frac{1}{q}\sum_{j=1}^{q}\underline{\epsilon}_j\right)\\ &=\frac{1}{2}+\frac{1}{2q}\left(\sum_{j=1}^{q}\left(\frac{1}{2}+\frac{1}{2}\epsilon_{i-1}\right)-\sum_{j=1}^{q}\left(\frac{1}{2}+\frac{1}{2}\epsilon_i\right)\right)\\ &=\frac{1}{2}+\frac{1}{4q}(\epsilon_0-\epsilon_q)\end{aligned}$$

which is non-negligibly larger than $\frac{1}{2}$ since q is polynomial in the security parameter. This contradicts with the weak signer ambiguity assumption. □

Corollary 1. *In AOFE, signer ambiguity (Definition 2) and security against the arbitrator (Definition 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 Section 2.1 is at least as strong as the weak signer ambiguity. Hence this corollary follows the theorem above directly.

3. Previous constructions revisited

In this section we first analyze the extension of two generic constructions to AOFE, and show that the resulting schemes are actually insecure under our proposed security model.

3.1. The first try

In [21] Huang et al. provided a simple and straightforward method in constructing efficient optimistic fair exchange schemes. In their proposal, each user has two key pairs, one for public key signature, and the other for ring signature. The partial signature of a user consists of only a (standard) signature on the message, while the full signature includes additionally a ring signature under the ring consisting of the signer and the arbitrator.

The main difference between OFE and AOFE is that in addition to all the security properties of OFE, AOFE also enjoys signer ambiguity. A natural way to extend their method in the construction of AOFE is to use two ring signatures in the scheme, i.e. the partial signature consists of a ring signature on M under the ring of the signer and the verifier, and the full signature includes additionally a ring signature on M under the ring of the signer and the arbitrator.

(Insecurity): It seems that we achieve signer ambiguity via this method, due to the anonymity of the underlying ring signature scheme. However, this construction is insecure against the arbitrator. That is, the arbitrator can frame any signer on any message without asking the signer to sign any message. To do this, the arbitrator C colludes with a verifier U_j, and selects a target user U_i and a target message M. U_j generates a partial signature σ_P on M with regard to the group $\{U_i,U_j\}$, and claims that σ_P was generated by U_i. Then C resolves it to σ_F by producing a ring signature under the ring $\{C,U_i\}$. The two rings intersect at U_i, and thus σ_F is binding to U_i.

3.2. The second try

In [15] Dodis et al. revisited a generic construction of optimistic fair exchange. Roughly, the partial signature of a user includes an encryption c of his signature σ on the message M generated under the arbitrator's public key, and an NIZK proof π showing that c contains the user's signature on M. The user's full signature on M is σ. They showed that this generic construction is a secure OFE scheme in the multi-user setting (under the certified-key model).

One may trivially extends this scheme to AOFE. Namely, the NIZK proof shows that c contains either the signer's signature on M or the verifier's signature on M, using the signature and the randomness for the encryption as the witness. This may seem to be a secure AOFE scheme. Namely, the NIZK proof shows the membership of the following language:

$$L=\big\{\big((\mathtt{pk}_{\mathsf{TA}},PK_i,PK_j,c,M),(\sigma,r)\big):c=\mathcal{E}.\mathsf{Enc}(\mathtt{pk}_{\mathsf{TA}},\sigma;r)\wedge\big(\mathcal{S}.\mathsf{Ver}(PK_i,\sigma,M)=1\vee\mathcal{S}.\mathsf{Ver}(PK_j,\sigma,M)=1\big)\big\}$$

where $\mathrm{pk}_{\mathsf{TA}}$ is the public key of the arbitrator, $\mathcal{E}$ is the encryption scheme and $\mathcal{S}$ is the signature scheme. However, the following attack demonstrates that it is actually *insecure* under the security model proposed in Section 2.1.

(The attack): We consider the security against the arbitrator (see Definition 5). Let C be an adversary. Given the challenge public key PK_A, C randomly selects a message M, and generates two public keys, say, PK_B, PK_D. Then it asks the oracle O_{PSig} to sign M w.r.t. PK_D, and obtains a ciphertext c, and an NIZK proof π_{AD} showing that c contains a signature generated by either user A or user D. C then uses the arbitrator's secret key to recover user A's signature σ, and re-encrypts σ under the arbitrator's public key using a fresh randomness r'. Let the new ciphertext be c'. Finally, the adversary produces a new NIZK proof π_{AB} showing that c' is an encryption of a signature on M generated by either A or B, using σ, r' as the witness. By the validity of (c, π_{AD}), we can easily get that (c', π_{AB}) is also a valid output. Note that in the whole attack C didn't submit (M, PK_B) to the oracle O_{PSig}. Thus, the security against the arbitrator breaks. However, it's not hard to prove that this generic construction is secure against the arbitrator under a weaker model which differs from the one described in Section 2.1 only in that it's required $(M, \cdot) \notin Query(C, O_{\mathsf{PSig}})$ instead of $(M, PK_B) \notin Query(C, O_{\mathsf{PSig}})$. This weak security against the arbitrator is guaranteed by the unforgeability (EUF-CMA) of the underlying signature scheme.

From the attacks above, we learn that constructing a secure AOFE scheme is not a trivial task. The introduction of signer ambiguity to OFE makes the security against the arbitrator more subtle. Besides, the security against the arbitrator of AOFE seems to be stronger than that of OFE defined in [15,21]. However, they in fact are not comparable. In the model considered in this paper, the public key of the verifier is involved in the generation of a partial signature, thus every query the adversary submits to oracles includes an additional public key; while this is not the case in the model of OFE. In the security against the arbitrator (Definition 5), the adversary is allowed to obtain the signer's signature on M (but with respect to any public key rather than PK_B). This is similar with the strong unforgeability of digital signatures [19].

4. Our generic construction

Now we propose a generic construction of AOFE in the standard model, which is similar with the one widely used in building OFE schemes. Namely, the signer's signature is encrypted under the arbitrator's public key, and then a non-interactive proof is given to show that the ciphertext contains the signer's signature on the message. As pointed by Boyd et al. [10], the non-interactive proof is not much different from the signer's signature, as it's also sufficient to prove to others that the signer is bound to the message. Since AOFE requires that a verifier cannot prove to others that the signer is bound to a message, in the generic construction the signer has the verifier involved in the proof. That is, the signer provides a non-interactive proof showing that the ciphertext contains either the signer's signature or the verifier's signature on the message.

4.1. The proposal

Let $\mathcal{S} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a public key signature scheme that is *weakly existentially unforgeable under chosen message attacks* [7,19], and $\mathsf{OTS} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a *strong one-time* signature scheme. Let $\mathcal{E} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be a tag-based public key encryption scheme that is *selective-tag weakly* CCA *secure*, and $\Pi = (\mathsf{Kg}, \mathsf{Prv}, \mathsf{Ver}, (\mathsf{Sim}_1, \mathsf{Sim}_2))$ be an NIZK proof system for the following language:

$$L = \big\{\big((\mathrm{pk}_{\mathsf{TA}}, PK_i, PK_j, c, \mathsf{tag}, M), (\sigma, r)\big) :$$
$$c = \mathcal{E}.\mathsf{Enc}(\mathrm{pk}_{\mathsf{TA}}, \mathsf{tag}, \sigma; r) \wedge \big(\mathcal{S}.\mathsf{Ver}(PK_i, \sigma, M) = 1 \vee \mathcal{S}.\mathsf{Ver}(PK_j, \sigma, M) = 1\big)\big\}.$$

Algorithm $\Pi.\mathsf{Kg}$ takes as input 1^k and outputs a common reference string $\mathtt{crs}$. Prv and Ver are the prover strategy and verification algorithm respectively. $(\mathsf{Sim}_1, \mathsf{Sim}_2)$ is the simulator, where Sim_1 takes as input 1^k and outputs a simulated common reference string $\mathtt{crs}$ that's indistinguishable from a real one, along with the corresponding trapdoor τ_S, and Sim_2 takes as input $(\mathtt{crs}, \tau_S, x)$ and outputs a non-interactive proof whose distribution is indistinguishable from that of proofs output by the prover. Note that the membership of the language above is efficiently checkable, so we have that $L \in \mathcal{NP}$. Fig. 1 (on page 186) describes our generic construction of AOFE, named GAOFE.

Note that in Fig. 1 (page 186) we omit the description of the parameter generation algorithm just for simplicity. This algorithm simply calls the corresponding parameter generation algorithms of the encryption scheme and the signature scheme to generate their parameters, which will be used in the other algorithms of GAOFE.

Remark 3. Note that in the construction, the public keys of both the signer and the verifier, i.e. PK_i and PK_j, are included in the message to be signed of the one-time signature δ. This is important, as otherwise the scheme would be vulnerable to an attack which compromises the security against signers. Specifically, the signer Alice runs as the verifier an execution of the protocol with Bob. After obtaining Bob's partial signature $\sigma_{P,B}$, Alice aborts this execution, and then restarts a new execution as the signer with Bob. She sends $\sigma_{P,B}$ as her partial signature to Bob. As $\sigma_{P,B}$ is a valid signature with regard to the group consisting of Alice and Bob, and the partial signature doesn't specify who the signer is and who the receiver is, Bob would view it as a valid one, and then returns his full signature. At this time, Alice aborts this execution again. Bob then resorts to the arbitrator for resolving $\sigma_{P,B}$ to a full one. However, since it was originally generated by Bob himself, the arbitrator can only resolve it to Bob's full signature. So in this case, Bob cannot get Alice's full signature, and thus the security against

- $\mathsf{Setup}^{\mathsf{TTP}}$: Given PM, the arbitrator runs $\mathcal{E}.\mathsf{Kg}(1^k)$ to generate a key pair, $(\mathtt{pk}_{\mathsf{TA}}, \mathtt{sk}_{\mathsf{TA}})$, and invokes $\Pi.\mathsf{Kg}(1^k)$ to produce a common reference string. It publishes $APK = (\mathtt{pk}_{\mathsf{TA}}, \mathtt{crs})$ and stores $ASK = \mathtt{sk}_{\mathsf{TA}}$ secretly.
- $\mathsf{Setup}^{\mathsf{User}}$: Each user U_i runs $\mathcal{S}.\mathsf{Kg}(1^k)$ to generate a key pair for the signature scheme, $(\mathtt{pk}_i, \mathtt{sk}_i)$, and publishes $PK_i = \mathtt{pk}_i$ and stores $SK_i = \mathtt{sk}_i$.
- PSig: To partially sign a message M with verifier U_j, the user U_i does the following.
  - Generate a new pair of one-time key for OTS, i.e. $(otvk, otsk) \leftarrow \mathsf{OTS.Kg}(1^k)$.
  - Compute a signature σ on $otvk$, i.e. $\sigma \leftarrow \mathcal{S}.\mathsf{Sig}(SK_i, otvk)$.
  - Encrypt the signature under the arbitrator's public key using randomness r with respect to tag $otvk$, i.e. $c \leftarrow \mathcal{E}.\mathsf{Enc}(\mathtt{pk}_{\mathsf{TA}}, otvk, \sigma; r)$.
  - Produce an NIZK proof π using witness (σ, r), i.e.

$$\pi \leftarrow \Pi.\mathsf{Prv}\big(\mathtt{crs}, (\mathtt{pk}_{\mathsf{TA}}, PK_i, PK_j, c, otvk, otvk), (\sigma, r)\big).$$

  - Sign the ciphertext, the proof and the message using $otsk$, i.e.

$$\delta \leftarrow \mathsf{OTS.Sig}(otsk, c\|\pi\|M\|PK_i\|PK_j).$$

  The partial signature σ_P is composed of $(c, \pi, \delta, otvk)$.
- PVer: On receiving U_i's partial signature $\sigma_P = (c, \pi, \delta, otvk)$ on message M, user U_j checks if $\mathsf{OTS.Ver}(otvk, \delta, c\|\pi\|M\|PK_i\|PK_j) = 1$ and $\Pi.\mathsf{Ver}(\mathtt{crs}, (\mathtt{pk}_{\mathsf{TA}}, PK_i, PK_j, c, otvk, otvk), \pi) = 1$. If either fails, it rejects; otherwise, it accepts.
- Sig: To sign a message M with verifier U_j, user U_i first computes a partial signature $\sigma_P = (c, \pi, \delta, otvk)$ as above, and then sets its full signature to be $\sigma_F = (\sigma, \sigma_P)$.
- Ver: On receiving U_i's full signature $\sigma_F = (\sigma, (c, \pi, \delta, otvk))$ on message M, the verifier first checks if $\mathsf{PVer}(M, (c, \pi, \delta, otvk), \{PK_i, PK_j\}, APK) = 1$ and $\mathcal{S}.\mathsf{Ver}(\mathtt{pk}_i, \sigma, otvk) = 1$. If either fails, it outputs 0 (reject); otherwise, it outputs 1 (accept).
- Res: On receiving from U_j a partial signature $\sigma_P = (c, \pi, \delta, otvk)$ claimed to be generated by U_i, the arbitrator checks the validity of σ_P by calling algorithm PVer. If it's invalid, it returns $\perp$ to U_j. Otherwise, it recovers σ from c by computing $\sigma \leftarrow \mathcal{E}.\mathsf{Dec}(\mathtt{sk}_{\mathsf{TA}}, otvk, c)$. If $\mathcal{S}.\mathsf{Ver}(PK_i, \sigma, otvk) = 1$, the arbitrator returns σ to U_j; otherwise, it returns $\perp$.

Fig. 1. Our generic construction of AOFE, GAOFE.

signers is broken. Therefore, it seems necessary to specify in the message to be signed the identities of the signer and the verifier. Besides the validity check of the partial signature, the verifier should also check if the public key of the receiver specified in the signature, i.e. PK_j, is his. If not, it should reject. However, the inclusion of public keys doesn't contradict the signer ambiguity. The embedding of PK_i and PK_j to the message to be signed doesn't require any private information, and anyone can do it. The verifier is still able to use his secret key to produce indistinguishable partial signatures. We also note that the attack above does not happen in ordinary OFE at all. Thus, this attack shows a big difference between AOFE and OFE.

4.2. Security analysis

Theorem 2. GAOFE *is a secure ambiguous optimistic fair exchange scheme.*

The theorem follows the following lemmas directly:

Lemma 1. GAOFE *is resolution ambiguous.*

Proof. Guaranteed by the security against signers (as shown in Lemma 3), if a partial signature σ_P is valid, then with overwhelming probability that the arbitrator can extract the signer's signature σ on the message. Conditioned on that the resolution succeeds, the signature output by the arbitrator is the same as that output by the signer. Therefore, the distribution of the output by the arbitrator is indistinguishable from that of the signer's signatures. So the construction above is resolution ambiguous. □

Lemma 2. GAOFE *is signer ambiguous.*

Before presenting the proof, we describe how the simulation algorithm FPSig works. To simulate a partial signature on M, the verifier U_j generates a fresh one-time key pair $(otvk, otsk)$ and uses its own secret key SK_j to generate its signature σ on $otvk$. The rest remains the same as algorithm PSig.

Proof. Let D be a PPT adversary against the security against verifiers. We modify the experiment so that the challenger runs $\Pi.\mathsf{Sim}_1$ algorithm to generate the common reference string $\mathtt{crs}$ along with a simulation trapdoor τ_S. By the common reference string indistinguishability of Π, D's advantage in new experiment is negligibly close to that in the original experiment.

Second, we modify the experiment so that to answer each query the adversary submits to O_{PSig} and to prepare the challenge partial signature, the challenger calls $\Pi.\mathsf{Sim}_2$ on input τ_S to produce the proof π, instead of calling the prover strategy with the witness (σ, r). Guaranteed by the zero-knowledge property of Π, this modification brings only a negligible difference to D's advantage.

Third, the experiment is modified again so that for each valid query submitted by the adversary to oracle O_{Res}, i.e. $(M, \sigma_P, \{PK_i, PK_j\})$ where $\sigma_P = (c, \pi, \delta, otvk)$, if $otvk$ was ever used by O_{PSig} in answering B's partial signing query, i.e. (M', PK') is the query and $\sigma' = (c', \pi', \delta', otvk)$ is the answer, but $(c, \pi, M, \{PK_i, PK_j\}, \delta) \neq (c', \pi', M', \{PK_A, PK'\}, \delta')$, the experiment is aborted. As will be discussed in the proof of Lemma 5, this case happens with negligible probability, guaranteed by the strong one-time unforgeability of OTS. Hence, D's advantage does not change noticeably.

Assume that D can win this experiment with non-negligible advantage ϵ_D, we then use it to build another PPT algorithm D' to break the selective-tag CCA security of $\mathcal{E}$, as below.

D' first calls $\mathsf{OTS.Kg}(1^k)$ to generate a one-time key pair $(otvk^*, otsk^*)$, and submits $otvk^*$ to its challenger as the challenge tag, which then returns a public key pk of $\mathcal{E}$. D' runs $\Pi.\mathsf{Sim}_1(1^k)$ to generate $(\mathtt{crs}, \tau_S)$, and calls $\mathcal{S}.\mathsf{Kg}(1^k)$ to generate a key pair for the honest user U_A, say, (PK_A, SK_A). It invokes D on input $(APK, PK_A) = ((pk, \mathtt{crs}), PK_A)$, and then begins to simulate oracle O_{Res} for D.

Given a query $(M, \sigma_P, \{PK_i, PK_j\})$ where $\sigma_P = (c, \pi, \delta, otvk)$, D' first checks if the query passes the PVer algorithm. If not, it returns $\perp$ to D; otherwise, the soundness of Π implies that c contains a valid signature σ on $otvk$ with respect to either PK_i or PK_j. D' forwards the ciphertext c and the tag $otvk$ to its own decryption oracle, and obtains σ. If either $\mathcal{S}.\mathsf{Ver}(PK_i, \sigma, otvk) = 1$ or $\mathcal{S}.\mathsf{Ver}(PK_j, \sigma, otvk) = 1$ holds, D' returns σ to D.

At some time, D submits $(M^*, (PK_A, SK_A), (PK_B, SK_B))$. D' first checks if both the key pairs are valid. If not, D aborts and outputs a random bit. Otherwise, it does the following:

1. Run $\mathcal{S}.\mathsf{Sig}$ twice to generate signatures σ_0, σ_1 on $otvk^*$ using SK_A, SK_B respectively.
2. Submit σ_0, σ_1 to its own challenger, which returns a ciphertext c^* of σ_b with respect to tag $otvk^*$ for some random bit $b \in \{0, 1\}$.
3. Call the simulator $\Pi.\mathsf{Sim}_2$ on input the trapdoor τ_S to generate a proof π^* for $(\mathtt{pk}, PK_A, PK_B, c^*, otvk^*)$.
4. Use $otsk^*$ to compute a one-time signature δ^* on $c^* \| \pi^* \| M^* \| PK_A \| PK_B$.

The challenge signature prepared by D' is $\sigma_P^* = (c^*, \pi^*, \delta^*, otvk^*)$. If $b = 0$, σ_P^* is a valid partial signature output by algorithm PSig; otherwise, σ^* is a simulated partial signature output by algorithm FPSig.

D' returns σ_P^* to the adversary, and then continues to simulate the oracle O_{Res}. Let $(M, \sigma_P, \{PK_i, PK_j\})$ be any of its valid queries, where $\sigma_P = (c, \pi, \delta, otvk)$. We distinguish the following two cases:

1. $otvk \neq otvk^*$. In this case, D' simulates O_{Res} in the same way as above.
2. $otvk = otvk^*$. We say, this case will not happen in the experiment. First of all, we can exclude the subcase $(c^* \| \pi^* \| M^* \| PK_A \| PK_B, \delta^*) = (c \| \pi \| M \| PK_i \| PK_j, \delta)$, because the adversary is prohibited from asking O_{Res} for resolving $(M^*, \sigma_P^*, \{PK_A, PK_B\})$. On the other hand, if $(c^* \| \pi^* \| M^* \| PK_A \| PK_B, \delta^*) \neq (c \| \pi \| M \| PK_i \| PK_j, \delta)$, according to the experiment's specification, the experiment is aborted.

Finally, A outputs a bit d. D then outputs a bit $b' = d$ and halts.

It's readily seen that the oracle O_{Res} is simulated indistinguishably by D'. If D succeeds in the experiment, D' also succeeds in outputting the bit b'. So D's advantage is

$$\mathtt{Adv}_{D'}^{\mathrm{T-PKE}}(k) \geq \mathtt{Adv}_{D}^{\mathrm{SA}}(k) = \epsilon_D$$

which is non-negligible by hypothesis. Therefore, the selective-tag CCA security of $\mathcal{E}$ is broken. □

Lemma 3. GAOFE *is secure against signers.*

Proof. If a partial signature $(c, \pi, \delta, otvk)$ of U_A is valid on message M with respect to verifier U_B, by the soundness of Π and the perfect completeness of $\mathcal{E}$, with overwhelming probability the arbitrator can recover from the ciphertext c a valid signature σ on M generated by either U_A or U_B. On the other hand, due to the security against the arbitrator (as shown in Lemma 5), we know that only with negligible probability can the adversary A forge a signature on behalf of an honest user U_B. Therefore, the valid signature must be generated by the adversary itself. Hence, the adversary can only break the security against signers with negligible probability. □

Lemma 4. GAOFE *is secure against verifiers.*

Proof. Let B be an efficient adversary against the security against verifiers. Let $\mathcal{P}$ be the set of partial signatures returned by the oracle O_{PSig}, and let (M^*, PK_B, σ_F^*) be B's final output, where $\sigma_F^* = (\sigma_P^*, \sigma^*)$. First of all, we modify the experiment so that if $\sigma_P^* \notin \mathcal{P}$, we abort it. Guaranteed by the security against the arbitrator (shown in Lemma 5), the modification leads to only a negligible difference in B's success probability.

Second, we modify the experiment so that the challenger runs $\Pi.\mathsf{Sim}_1$ algorithm to generate the common reference string $\mathtt{crs}$ along with a simulation trapdoor τ_S, and to answer each query the adversary submits to O_{PSig} the challenger calls $\Pi.\mathsf{Sim}_2$ with τ_S to produce the proof π, instead of calling the prover strategy with the witness (σ, r). By the zero knowledge property of Π, we know that the modification brings a negligible difference to B's success probability.

Third, the experiment is modified again so that for each valid query submitted by the adversary to oracle O_{Res}, i.e. $(M, \sigma_P, \{PK_i, PK_j\})$ where $\sigma_P = (c, \pi, \delta, otvk)$, if $otvk$ was ever used by O_{PSig} in answering B's partial signing query, i.e. (M', PK') is the query and $\sigma' = (c', \pi', \delta', otvk)$ is the answer, but $(c, \pi, M, \{PK_i, PK_j\}, \delta) \neq (c', \pi', M', \{PK_A, PK'\}, \delta')$, the experiment is aborted. If the case happens, obviously, it indicates that the adversary produces a forgery for the signature scheme OTS. Guaranteed by the strong one-time unforgeability of OTS, we have that the modification affects B's success probability negligibly as well.

Next we show that B's success probability in this experiment, say, ϵ_B, is negligible in k. Assume that ϵ_B is non-negligible, we then use B to build another PPT algorithm D to break the selective-tag CCA security of $\mathcal{E}$.

D first invokes the algorithm $\mathcal{S}.\mathsf{Kg}(1^k)$ to generate a pair of one-time key, $(otvk^*, otvk^*)$, submits $otvk^*$ to its challenger as the target tag, and receives a challenge public key pk of $\mathcal{E}$ from the challenger. It runs $\Pi.\mathsf{Sim}_1(1^k)$ to generate $(\mathtt{crs}, \tau_S)$, and calls $\mathcal{S}.\mathsf{Kg}$ to generate (PK_A, SK_A). Suppose that B will submit at most q distinct query to oracle O_{PSig}, which is polynomial in the security parameter. D picks i at random from the set $\{1, \cdots, q\}$, and sets $\mathtt{pk}_{\mathsf{TA}} = pk$, $APK = (\mathtt{crs}, \mathtt{pk}_{\mathsf{TA}})$. It invokes B on input (APK, PK_A), and then begins to simulate oracles for B as follows.

- O_{Res}: On input a resolution query, $(M, \sigma_P, PK_i, PK_j)$, D first checks the validity of the query, and returns $\perp$ if it doesn't pass the PVer algorithm. It asks its decryption oracle to decrypt c with respect to tag $otvk$, and receives σ from it. D returns σ if $\mathcal{S}.\mathsf{Ver}(PK_i, \sigma, otvk) = 1$ or $\mathcal{S}.\mathsf{Ver}(PK_j, \sigma, otvk) = 1$.
- O_{PSig}: Let (M, PK') be the j-th distinct query B submits to the oracle. If $j \neq i$, this query is dealt with by D like a real user. If $j = i$, D computes $\sigma^* \leftarrow \mathcal{S}.\mathsf{Sig}(SK_A, otvk^*)$. It sets $\sigma_0 = \sigma^*$ and randomly selects σ_1 from the range of $\mathcal{S}.\mathsf{Sig}(\cdot, \cdot)$, and forwards σ_0, σ_1 to its challenger, which randomly chooses one of them say, σ_b for some bit $b \in \{0, 1\}$, to encrypt. Let the ciphertext be c^*. D then calls algorithm $\Pi.\mathsf{Sim}_2$ with the simulation trapdoor τ_S to produce a proof π^* on $(\mathtt{pk}_{\mathsf{TA}}, PK_A, PK', c^*, otvk^*, otvk^*)$. It then completes the generation of the partial signature by computing $\delta^* = \mathsf{OTS.Sig}(otsk^*, c^* \| \pi^* \| M \| PK_A \| PK')$, and returns $\sigma_P^* = (c^*, \pi^*, \delta^*, otvk^*)$ back to the adversary B.

Finally B outputs (M^*, PK_B, σ_F^*), where $\sigma_F^* = (\sigma_P^*, \sigma^*)$. If the output doesn't satisfy the winning condition, D aborts and outputs a random bit. By the specification of the experiment, we know that $\sigma_P^* \in \mathcal{P}$. If (M^*, PK_B) is not the i-th distinct query to oracle O_{PSig}, alternatively, D's guess of i is incorrect, D aborts and outputs a random bit. If $\mathsf{Ver}(M^*, \sigma_F^*, PK_A, PK_B, APK) = 1$, D outputs 0; otherwise, it outputs 1.

We now consider the simulation of oracles O_{Res} and O_{PSig}. Let $(M, \sigma_P = (c, \pi, \delta, otvk), \{PK_i, PK_j\})$ be a valid query to O_{Res}. If $c \neq c^*$, D can handle this query as above. We then focus on the other case, $c = c^*$.

- If $otvk \neq otvk^*$, D can simply forward $(otvk, c^*)$ to its decryption oracle as the tag $otvk$ is different from the challenge one, $otvk^*$, and obtain the signature σ.
- If $otvk = otvk^*$, D is not allowed to ask its oracle to decrypt c^* with respect to $otvk^*$. However, as discussed in the proof of Lemma 2, this case will not happen.

Regarding the oracle O_{PSig}, it's perfectly simulated when $j \neq i$. For $j = i$, if c^* is an encryption of σ_0, the view of B is identical to that in a real attack, and B will succeed in the experiment with probability ϵ_B, which is non-negligible. If c^* is an encryption of σ_1, the view of the adversary is indistinguishable from that in a real attack, due to the zero knowledge property of Π. Since c^* is independent of PK_A, and thus provides no help to B in generating σ_F^*. In this case, if B successfully produces a valid full signature σ_F^*, it should be that B forges a full signature on behalf of the honest user U_A, thus breaking the security against arbitrator. As shown in Lemma 5, B's success probability in this case is negligible.

Let b' be the bit that D outputs. We want to show that $|\Pr[b' = 1 \wedge b = 1] - \Pr[b' = 0 \wedge b = 1]|$ is non-negligible in k. No matter $b = 0$ or $b = 1$, the probability that D aborts due to an incorrect guess of i is the same, and when it aborts, it outputs 0 with probability exactly one-half. Therefore, we only need to focus on the case in which D guesses i correctly, which happens with probability $1/q$. Denote this event by Corr.

If $b = 0$, as we discussed above, the probability that B outputs a valid σ_F^* is ϵ_B. Thus, D outputs 0 with probability at least ϵ_B. On the other side, i.e. $b = 1$, as discussed above, the probability that D outputs 0 is ε, which is the maximum probability that an efficient can break the security against the arbitrator, and is negligible. So we get the following:

$$\begin{aligned}
\mathrm{Adv}_D^{\mathrm{T-PKE}}(k) &= \left| \Pr\left[b' = b\right] - \frac{1}{2} \right| \\
&= \left| \Pr\left[b' = b \wedge \mathsf{Corr}\right] + \Pr\left[b' = b \wedge \neg\mathsf{Corr}\right] - \frac{1}{2} \right|
\end{aligned}$$

$$= \left|\Pr[\mathsf{Corr}]\Pr\left[b' = b|\mathsf{Corr}\right] + \Pr[\neg\mathsf{Corr}]\Pr\left[b' = b|\neg\mathsf{Corr}\right] - \frac{1}{2}\right|$$
$$= \left|\frac{1}{q}\left(\frac{1}{2}\Pr\left[b' = 0|b = 0 \wedge \mathsf{Corr}\right] + \frac{1}{2}\left(1 - \Pr\left[b' = 0|b = 1 \wedge \mathsf{Corr}\right]\right)\right) + \left(1 - \frac{1}{q}\right)\frac{1}{2} - \frac{1}{2}\right|$$
$$= \frac{1}{2q}\left|\Pr\left[b' = 0|b = 0 \wedge \mathsf{Corr}\right] - \Pr\left[b' = 0|b = 1 \wedge \mathsf{Corr}\right]\right|$$
$$\geq \frac{1}{2q}|\epsilon_{\mathcal{B}} - \varepsilon|$$

which is also non-negligible. Therefore, the selective-tag CCA security of $\mathcal{E}$ is broken. □

Lemma 5. GAOFE *is secure against the arbitrator.*

Proof. Let C be a PPT adversary C which breaks the security against the arbitrator, and (M^*, σ_F^*, PK_B) be its final output, where $\sigma_F^* = (\sigma_P^*, \sigma^*)$ and $\sigma_P^* = (c^*, \pi^*, \delta^*, otvk^*)$. We distinguish the following two cases.

1. $otvk^*$ never appeared in oracle O_{PSig}'s answers. In this case, we can use C's capability to build a PPT adversary F_0 to break the weak unforgeability of $\mathcal{S}$.
   Suppose that C issues at most q distinct queries to oracle O_{PSig}. F_0 runs $\mathcal{S}.\mathsf{Kg}(1^k)q$ times to generate q one-time key pairs, say, $(otvk_1, otvk_1), \cdots, (otvk_q, otsk_q)$. It submits $\{otvk_1, \cdots, otvk_q\}$ to its challenger, which then returns a public key pk and the corresponding signatures, $\{\sigma_1, \cdots, \sigma_q\}$. F_0 then invokes C on input 1^k, which returns a public key of the arbitrator, $APK = (\mathtt{pk}_{\mathsf{TA}}, \mathtt{crs})$. F_0 then returns $PK_A = pk$ to C, and begins to simulate the partial signing oracle O_{PSig} for C. On input the i-th distinct query (M, PK_{j_i}), F_0 chooses a random string r from the randomness space of $\mathcal{E}$, uses r to encrypt σ_i under $\mathtt{pk}_{\mathsf{TA}}$ with respect to the tag $otvk_i$, and uses (σ_i, r) as the witness to compute an NIZK proof π. F_0 runs OTS.Sig with secret key $otsk_i$ to generate a one-time signature δ on $c\|\pi\|M\|PK_A\|PK_{j_i}$. It returns $(c, \pi, \delta, otvk_i)$ to C. It's readily seen that the simulation of O_{PSig} is perfect.
   Finally, C outputs (M^*, PK_B, σ_F^*) where $\sigma_F^* = ((c^*, \pi^*, \delta^*, otvk^*), \sigma^*)$, and wins the experiment with non-negligible probability ϵ_C. F_0 outputs $(otvk^*, \sigma^*)$. By the validity of C's output, we know that $\mathcal{S}.\mathsf{Ver}(pk, \sigma^*, otvk^*) = 1$. Since $otvk^*$ is fresh, i.e., $otvk^* \notin \{otvk_1, \cdots, otvk_q\}$, F_0 didn't ask its signing oracle to return a signature on $otvk^*$. Thus, $(otvk^*, \sigma^*)$ is a valid forgery for $\mathcal{S}$. Therefore, F_0 breaks the security of $\mathcal{S}$ with probability at least the same as C.
2. $otvk^*$ appeared in one of O_{PSig}'s answers to C's partial signing queries. Again, we assume that C issues at most q queries to O_{PSig}, which is polynomial in k. We use C to build an algorithm F_1 to break the security of OTS.
   Given a public key $otvk^*$ of OTS and a one-time signing oracle, F_1 invokes C on input 1^k and obtains $APK = (\mathtt{pk}_{\mathsf{TA}}, \mathtt{crs})$ from it. It then calls $\mathcal{S}.\mathsf{Kg}(1^k)$ to generate a key pair for user A, say, (PK_A, SK_A), and randomly selects i from $\{1, \cdots, q\}$. F_1 gives PK_A to C, and then begins to simulate oracle O_{PSig} for it.
   If $j \neq i$, F_1 simulates the oracle like an honest user U_A does. If $j = i$, it uses SK_A to generate a signature σ on $otvk^*$, selects a random string r, computes $c \leftarrow \mathcal{E}.\mathsf{Enc}(\mathtt{pk}_{\mathsf{TA}}, otvk^*, \sigma; r)$ and uses (σ, r) as the witness to produce an NIZK proof π. F_1 then submits $c\|\pi\|M\|PK_A\|PK_i$ to its signing oracle, and obtains a signature δ^*. It then returns $(c, \pi, \delta, otvk^*)$ back to C.
   Finally C outputs (M^*, σ_F^*, PK_B) where $\sigma_F^* = (\sigma_P^*, \sigma^*)$ and $\sigma_P^* = (c^*, \pi^*, \delta^*, otvk^*)$. F_1 outputs $(c^*\|\pi^*\|M^*\|PK_A\|PK_B, \delta^*)$. Assume that C wins the experiment. So we have that $\mathsf{OTS.Ver}(otvk^*, \delta^*, c^*\|\pi^*\|M^*\|PK_A\|PK_B) = 1$ and C didn't issue a query on input (M^*, PK_B). Since the pair (M^*, PK_B) is fresh, we have that
   $$c^*\|\pi^*\|M^*\|PK_A\|PK_B \neq c\|\pi\|M\|PK_A\|PK_i.$$
   So δ^* is a valid signature on a new message. Therefore, the security of OTS is broken.

Since we do not know which of the two cases above will happen, we simply toss a coin b, and run the algorithm F_b. Still, we have non-negligible probability to break the security of either $\mathcal{S}$ or OTS, if C wins its experiment with non-negligible probability. □

Remark 4. In our construction, the signer uses its secret key to generate a 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 [19], this combination leads to a strongly unforgeable signature scheme. It's not hard to see that our proposed AOFE scheme actually achieves a stronger version of security against the arbitrator. 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 produce another full signature on M, say, σ_F', such that $\mathsf{Ver}(M, \sigma_F', PK_A, PK_B, APK) = 1$. The claim can be shown using the proof given above without much modification.

5. A concrete scheme without random oracles

In this section, we propose an AOFE scheme, which is based on Groth and Sahai's idea of constructing a fully anonymous group signature scheme [17,18]. Before describing the scheme, we first introduce the assumptions and building tools used in our construction.

5.1. Assumptions

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

(**Decision Linear Assumption (DLIN)** [9]): 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 $1/2$:

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

(**Strong Diffie–Hellman Assumption (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:

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

5.2. Building tools

(Tag-based encryption scheme): We use a tag-based public key encryption scheme $\mathcal{E}$ with security based on the DLIN assumption reviewed above. Below is a brief review of a suitable scheme due to Kiltz [28].

Key generation: $(pk, sk) = ((F, H, K, L), (\kappa, \lambda)) \leftarrow \mathcal{E}.K(p, \mathbb{G}_1, \mathbb{G}_T, \hat{e}, g)$, where $F = g^\kappa$ and $H = g^\lambda$;

Encryption: Let $M \in \mathbb{G}_1$ be a message, $\mathsf{tag} \in \mathbb{Z}_p$ a tag, and $r, s \in \mathbb{Z}_p$ the randomness. The ciphertext y is computed as $y = (y_1, y_2, y_3, y_4, y_5) = (F^r, H^s, M \cdot g^{r+s}, (g^{\mathsf{tag}}K)^r, (g^{\mathsf{tag}}L)^s)$;

Decryption: The validity of a ciphertext can be checked without knowing the secret key. Anyone can check if $\hat{e}(F, y_4) = \hat{e}(y_1, g^{\mathsf{tag}}K)$ and $\hat{e}(H, y_5) = \hat{e}(y_2, g^{\mathsf{tag}}L)$. If any one fails, $\perp$ is returned. Otherwise, M is recovered by computing $M = y_3 \cdot y_1^{-1/\kappa} \cdot y_2^{-1/\lambda}$.

Note that the plaintext can also be recovered if the discrete logarithms of K and L with respect to g are known. Assume that $K = g^{\kappa'}$ and $L = g^{\lambda'}$. After checking the validity of $y = (y_1, y_2, y_3, y_4, y_5)$, M can be computed as $M = y_3 \cdot y_4^{-1/(\mathsf{tag}+\kappa')} \cdot y_5^{-1/(\mathsf{tag}+\lambda')}$. Kiltz proved [28] that under the DLIN assumption, the tag-based encryption scheme is selective-tag weakly chosen-ciphertext secure, that is, an adversary $\mathcal{A}$ cannot tell which message was encrypted under tag^* (selected by $\mathcal{A}$ though) before seeing the public key, even when it has access to a decryption oracle that decrypts ciphertexts under any tag other than tag^*. Readers may refer to [28] for the definition of selective-tag weakly CCA security.

(Non-interactive proofs): Recently, Groth and Sahai [18] proposed a general methodology for constructing simple and efficient non-interactive witness indistinguishable (NIWI) proofs and non-interactive zero-knowledge (NIZK) proofs that work for bilinear groups, without requiring complex NP-reductions. They proposed efficient non-interactive (NI) proofs for a set of equations in a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_T, \hat{e}, g)$ over variables in $\mathbb{G}_1$ and $\mathbb{Z}_p$, such as pairing products, i.e. $\hat{e}(x_1, y_1) \cdot \hat{e}(x_2, y_2) = T$, or multi-exponentiations, i.e. $x_1^{\delta_1} x_2^{\delta_2} = 1$, having solutions $x_i \in \mathbb{G}_1$, $\delta_j \in \mathbb{Z}_p$, so that all equations are simultaneously satisfied. Their NI proofs can be based on subgroup decision assumption, (symmetric) external Diffie–Hellman ((S)XDH) assumption, and decision linear (DLIN) assumption. In our construction, we will use their DLIN-based technique.

(Commitment scheme): Groth–Sahai's DLIN-based proofs consist of two commitment schemes, one for committing to variables of $\mathbb{G}_1$, and the other one to variables in $\mathbb{Z}_p$. The common reference string for either of the two schemes, can be generated in either of two indistinguishable ways.

For the first commitment scheme, a real common reference string is set up to $U = F^R$, $V = H^S$, and $W = g^{R+S}$, a commitment to a variable $x \in \mathbb{G}_1$ can be respectively expressed as $c = (c_1, c_2, c_3) = (F^r U^t, H^s V^t, g^{r+s} W^t x)$ for randomness $r, s \in \mathbb{Z}_p$. The key for extracting x from c is $xk = (\kappa, \lambda) = (\log_g F, \log_g H)$, so the scheme is perfectly binding. A simulated common reference string consists of F, H, $U = F^R, V = H^S$ and $W = g^T$ where $T \neq R + S$, and thus the scheme is also perfectly hiding.

For the commitment to a variable $\delta \in \mathbb{Z}_p$, a real common reference string consists of F, H, $U' = F^{R'}$, $V' = H^{S'}$ and $W' = g^{T'}$ where $T' \neq R' + S'$, and a commitment to δ is expressed as $c' = (c'_1, c'_2, c'_3) = (F^{r'}(U')^\delta, H^{s'}(V')^\delta, g^{r'+s'}(W')^\delta)$

for randomness $r', s' \in \mathbb{Z}_p$. The commitment scheme is perfectly binding. A simulated common reference string consists of $F, H, U = F^{R'}, V = H^{S'}, W = g^{R'+S'}$. The commitment scheme then becomes perfectly hiding. The simulation trapdoor is $tk = (R', S')$, and we can use it to reveal a commitment to 0 to any other value $\delta \in \mathbb{Z}_p$. Due to the DLIN assumption, we have that any PPT adversary cannot tell a real common reference string apart from a simulated one.

(Groth–Sahai Proofs): Groth–Sahai NI proof system for bilinear groups consists of four PPT algorithms, (K_{NI}, P, V, X), where the key generator K_{NI} takes as input a system parameter $(p, \mathbb{G}_1, \mathbb{G}_T, \hat{e}, g)$ and outputs a common reference string $\mathtt{crs} = (F, H, U, V, W, U', V', W')$ and an extraction key xk; the algorithm P takes as input $\mathtt{crs}$, a problem instance and a witness $(\cdots, x_i, \cdots, \delta_j, \cdots)$ and outputs a proof π; the verification algorithm V takes as input $\mathtt{crs}$, a problem instance, and a proof π, and outputs 1 indicating that π is valid or 0 indicating that π is invalid; and the extraction algorithm X takes as input $\mathtt{crs}$, a problem instance, and a proof π, and outputs $(\cdots, x_i, \cdots)$.

Groth–Sahai NI proofs have perfect completeness, and perfect soundness on a real common reference string. Besides, they have perfect partial knowledge: the extraction algorithm will extract $(\cdots, x_i, \cdots)$ from the proof, such that there is a solution for the equations using these x_i's. Groth–Sahai proofs also have perfect witness-indistinguishability on a simulated common reference string: if there are many possible witnesses for the equations being satisfiable, the proof π does not reveal anything about which witness was used by the prover in generating π.

In our construction (Section 5.4), we use two NI proof systems: NIWI and NIZK. The NIWI proof system is used for showing that a BB-signature $\overline{\sigma}$ is valid with respect to either PK_i or PK_j, i.e.

$$\mathrm{NIWI}\big\{\alpha : \hat{e}\big(\alpha, g^{\mathrm{H}(otvk)}PK_i\big) = \hat{e}(g, g) \vee \hat{e}\big(\alpha, g^{\mathrm{H}(otvk)}PK_j\big) = \hat{e}(g, g)\big\}.$$

The NIZK proof system is used for showing that the commitment $c = (c_1, c_2, c_3) = (F^{r_c}U^t, H^{s_c}V^t, g^{r_c+s_c}W^t\overline{\sigma})$ in the NIWI proof π_1 and the ciphertext $y = (y_1, y_2, y_3, y_4, y_5) = (F^{r_y}, H^{s_y}, g^{r_y+s_y}\overline{\sigma}, (g^{\mathsf{tag}}K)^{r_y}, (g^{\mathsf{tag}}L)^{s_y})$, where $\mathsf{tag} = \mathrm{H}(otvk)$, contain the same $\overline{\sigma}$. This is equivalent to showing the knowledge of a solution to the equation $(c_1^{-1}y_1)F^rU^t = 1 \ \wedge \ (c_2^{-1}y_2)H^sV^t = 1 \ \wedge \ (c_3^{-1}y_3)g^{r+s}W^t = 1$. If c and y contain different messages, then there will be no r, s, t satisfying all the equations above. Groth and Sahai [18] showed a way to turn the set of equations above to a *tractable* set, which has zero-knowledge proofs. The set of equations above is equivalent to the following one, which is tractable:

$$\phi = 1 \wedge \big(c_1^{-1}y_1\big)^{\phi}F^rU^t = 1 \wedge \big(c_2^{-1}y_2\big)^{\phi}H^sV^t = 1 \wedge \big(c_3^{-1}y_3\big)^{\phi}g^{r+s}W^t = 1.$$

This NIZK proof system was also used by Groth in constructing a fully anonymous group signature scheme [17]. Readers may refer to [17,18] for more details.

5.3. High level description of our construction

As mentioned in the introduction, many OFE schemes in the literature follows a generic framework: Alice encrypts her signature under the arbitrator's public key, and provides a proof showing that the ciphertext contains her signature. To extend this framework to AOFE, we may let Alice encrypt her signature under the arbitrator's public key and provide a proof showing that the ciphertext contains either her signature or Bob's signature.

Our concrete construction below follows the framework, which is based on the idea of Groth in constructing a fully anonymous group signature [17]. In detail, 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 [28], with the one-time verification key as the tag. The non-interactive proof is based on a technique due to Groth and Sahai [18]. It is efficient and does not 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.

5.4. The concrete scheme and security analysis

Our proposed concrete scheme, AOFE, is shown in Fig. 2.

Theorem 3. *The proposed AOFE is secure in the multi-user setting and chosen-key model without random oracles, provided that DLIN assumption and SDH assumption hold.*

AOFE follows the framework of the generic construction proposed in Section 4. 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

- PMGen takes 1^k and outputs $\mathsf{PM} = (1^k, p, \mathbb{G}_1, \mathbb{G}_T, \hat{e}, g)$ so that $\mathbb{G}_1$ and $\mathbb{G}_T$ are cyclic groups of prime order p; g is a random generator of $\mathbb{G}_1$; $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ 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 [28], and xk is the extraction key of the NIWI proof system [17,18], 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 $\mathrm{H}(otvk)$, i.e. $\overline{\sigma} \leftarrow g^{1/(x_i+\mathrm{H}(otvk))}$.
  3. Compute a tag-based encryption [28] y of $\overline{\sigma}$, i.e. $y = (y_1, y_2, y_3, y_4, y_5) \leftarrow \mathcal{E}.E_{pk}(\overline{\sigma}, \mathrm{H}(otvk))$, where $pk = (F, H, K, L)$.
  4. Compute an NIWI proof π_1 showing that $\overline{\sigma}$ is a valid signature under either PK_i or PK_j, i.e. $\pi_1 \leftarrow P_{WI}(\mathtt{crs}, (\hat{e}(g, g), PK_i, PK_j, \mathrm{H}(otvk)), (\overline{\sigma}))$, which shows that the following holds:
$$\hat{e}\big(\overline{\sigma}, PK_i \cdot g^{\mathrm{H}(otvk)}\big) = \hat{e}(g, g) \vee \hat{e}\big(\overline{\sigma}, PK_j \cdot g^{\mathrm{H}(otvk)}\big) = \hat{e}(g, g).$$
  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. $\sigma_{ot} \leftarrow \mathcal{S}.S_{otsk}(M\|\pi_1\|y\|\pi_2\|PK_i\|PK_j)$.

  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 $M\|\pi_1\|y\|\pi_2\|PK_i\|PK_j$ under $otvk$.
  2. If π_1 is a valid NIWI proof, i.e. $V_{WI}(\mathtt{crs}, (\hat{e}(g, g), PK_i, PK_j, \mathrm{H}(otvk)), \pi_1) \stackrel{?}{=} 1$.
  3. If π_2 is a valid NIZK proof, i.e. $V_{ZK}(\mathtt{crs}, (y, \pi_1), \pi_2) \stackrel{?}{=} 1$.
- 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 $\mathsf{PVer}(M, \sigma_P, \{PK_i, PK_j\}, APK) \stackrel{?}{=} 1$, and if $\hat{e}(\overline{\sigma}, PK_i \cdot g^{\mathrm{H}(otvk)}) \stackrel{?}{=} \hat{e}(g, g)$. 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.

Fig. 2. Our proposed concrete scheme of AOFE, AOFE.

arbitrator are guaranteed by the weak unforgeability of BB-signature scheme. Proof of the security of the scheme almost follows that of our generic construction of AOFE.

One may notice that an NIWI proof π_1 and an NIZK proof π_2 are used in the generation of partial signatures in AOFE, while only an NIZK proof is required in the generic construction GAOFE. This is not to say the instantiation deviates from the generic construction. In fact, proofs π_1 and π_2 functionally serve as the NIZK proof in GAOFE. To see it, notice that the simulator of the NIZK proof (in GAOFE) is mainly used in the proof of signer ambiguity. A simulated partial signature is generated by first using the intended verifier U_j's secret key to generate a (conventional) signature and then running the NIZK simulator to produce a simulated proof to show that the ciphertext contains either the signer U_i's signature or U_j's signature. While in the instantiation, the NIWI proof π_1 is simulated by using U_j's signature to show that there is a signature of either U_i or U_j, and the NIZK proof π_2 is simulated by calling the corresponding simulator to show that the (tag-based) ciphertext contains a signature same as the witness used in π_1.

6. Conclusion

In this paper, we proposed the notion of ambiguous optimistic fair exchange (AOFE), and gave a formal security model for it. We discussed the relationship among some variants of the model, and showed that signer ambiguity and security against the arbitrator together imply security against verifiers (in a weaker sense). We revisited two generic constructions of OFE, and showed that they cannot be simply extended to AOFE. We then proposed a generic construction of AOFE, and proved its security under the proposed multi-user setting and chosen-key model. We also proposed a concrete and efficient construction of AOFE in bilinear groups, security of which is based on Decision Linear assumption and Strong Diffie–Hellman assumption without random oracles.

References

[1] N. Asokan, M. Schunter, M. Waidner, Optimistic protocols for fair exchange, in: ACM CCS'97, ACM, 1997, pp. 7–17.
[2] N. Asokan, V. Shoup, M. Waidner, Optimistic fair exchange of digital signatures (extended abstract), in: Advances in Cryptology – EUROCRYPT 98, in: Lecture Notes in Computer Science, vol. 1403, Springer, 1998, pp. 591–606.

[3] N. Asokan, V. Shoup, M. Waidner, Optimistic fair exchange of digital signatures, IEEE J. Sel. Areas Comm. 18 (4) (2000) 593–610.
[4] F. Bao, G. Wang, J. Zhou, H. Zhu, Analysis and improvement of Micali's fair contract signing protocol, in: ACISP04, in: Lecture Notes in Computer Science, vol. 3108, Springer, 2004, pp. 176–187.
[5] M. Bellare, P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in: ACM CCS'93, ACM, 1993, pp. 62–73.
[6] A. Bender, J. Katz, R. Morselli, Ring signatures: stronger definitions, and constructions without random oracles, in: TCC06, in: Lecture Notes in Computer Science, vol. 3876, Springer, 2006, pp. 60–79.
[7] D. Boneh, X. Boyen, Short signatures without random oracles, in: EUROCRYPT04, in: Lecture Notes in Computer Science, vol. 3027, Springer, 2004, pp. 56–73.
[8] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in: Advances in Cryptology – EUROCRYPT 2003, in: Lecture Notes in Computer Science, vol. 2656, Springer, 2003, pp. 416–432.
[9] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: Advances in Cryptology – CRYPTO 2004, in: Lecture Notes in Computer Science, vol. 3152, Springer, 2004, pp. 41–55.
[10] C. Boyd, E. Foo, Off-line fair payment protocols using convertible signatures, in: Advances in Cryptology – ASIACRYPT 98, in: Lecture Notes in Computer Science, vol. 1514, Springer, 1998, pp. 271–285.
[11] J. Camenisch, I. Damgård, Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes, in: ASIACRYPT00, in: Lecture Notes in Computer Science, vol. 1976, Springer, 2000, pp. 331–345.
[12] L. Chen, C. Kudla, K.G. Paterson, Concurrent signatures, in: EUROCRYPT04, in: Lecture Notes in Computer Science, vol. 3027, Springer, 2004, pp. 287–305.
[13] X. Chen, F. Zhang, H. Tian, Q. Wu, Y. Mu, K. Kim, Three-round abuse-free optimistic contract signing with everlasting secrecy, in: Proceedings of FC 2010, in: Lecture Notes in Computer Science, vol. 6052, Springer, 2010, pp. 304–311.
[14] Y. Dodis, L. Reyzin, Breaking and repairing optimistic fair exchange from PODC 2003, in: ACM Workshop on Digital Rights Management, DRM 2003, ACM, 2003, pp. 47–54.
[15] Y. Dodis, P.J. Lee, D.H. Yum, Optimistic fair exchange in a multi-user setting, in: PKC07, in: Lecture Notes in Computer Science, vol. 4450, Springer, 2007, pp. 118–133.
[16] J.A. Garay, M. Jakobsson, P. MacKenzie, Abuse-free optimistic contract signing, in: CRYPTO99, in: Lecture Notes in Computer Science, vol. 1666, Springer, 1999, pp. 449–466.
[17] J. Groth, Fully anonymous group signatures without random oracles, in: K. Kurosawa (Ed.), ASIACRYPT07, in: Lecture Notes in Computer Science, vol. 4833, Springer, 2007, pp. 164–180.
[18] J. Groth, A. Sahai, Efficient non-interactive proof systems for bilinear groups, in: EUROCRYPT08, in: Lecture Notes in Computer Science, vol. 4965, Springer, 2008, pp. 415–432, also at Cryptology ePrint Archive, Report 2007/155, http://eprint.iacr.org/.
[19] Q. Huang, D.S. Wong, J. Li, Y. Zhao, Generic transformation from weakly to strongly unforgeable signatures, J. Comput. Sci. Tech. 23 (2) (2008) 240–252.
[20] Q. Huang, G. Yang, D.S. Wong, W. Susilo, Ambiguous optimistic fair exchange, in: ASIACRYPT08, in: Lecture Notes in Computer Science, vol. 5350, Springer, 2008, pp. 74–89.
[21] Q. Huang, G. Yang, D.S. Wong, W. Susilo, Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles, in: CT-RSA08, in: Lecture Notes in Computer Science, vol. 4964, Springer, 2008, pp. 106–120.
[22] Q. Huang, D.S. Wong, W. Susilo, A new construction of designated confirmer signature and its application to optimistic fair exchange – (extended abstract), in: Proceedings of Pairing-Based Cryptography, Pairing 2010, in: Lecture Notes in Computer Science, vol. 6487, Springer, 2010, pp. 41–61.
[23] Q. Huang, D.S. Wong, W. Susilo, Efficient designated confirmer signature and DCS-based ambiguous optimistic fair exchange, IEEE Trans. Inf. Forensics Secur. 6 (4) (2011) 1233–1247.
[24] Q. Huang, D.S. Wong, W. Susilo, Group-oriented fair exchange of signatures, Inform. Sci. 181 (16) (2011) 3267–3283.
[25] Q. Huang, D.S. Wong, W. Susilo, The construction of ambiguous optimistic fair exchange from designated confirmer signature without random oracles, in: Proceedings of Public Key Cryptography, PKC 2012, in: Lecture Notes in Computer Science, vol. 7293, Springer, 2012, pp. 120–137.
[26] Q. Huang, D.S. Wong, W. Susilo, P2ofe: privacy-preserving optimistic fair exchange of digital signatures, in: Proceedings of RSA Conference, Cryptographers' Track, CT-RSA 2014, in: Lecture Notes in Computer Science, vol. 8366, Springer, 2014, pp. 367–384.
[27] M. Jakobsson, K. Sako, R. Impagliazzo, Designated verifier proofs and their applications, in: EUROCRYPT96, in: Lecture Notes in Computer Science, vol. 1070, Springer, 1996, pp. 143–154.
[28] E. Kiltz, Chosen-ciphertext security from tag-based encryption, in: TCC06, in: Lecture Notes in Computer Science, vol. 3876, Springer, 2006, pp. 581–600.
[29] S. Kremer, Formal analysis of optimistic fair exchange protocols, Ph.D. thesis, Université Libre de Bruxelles, 2003.
[30] M. Liskov, S. Micali, Online-untransferable signatures, in: PKC08, in: Lecture Notes in Computer Science, vol. 4939, Springer, 2008, pp. 248–267.
[31] A. Lysyanskaya, S. Micali, L. Reyzin, H. Shacham, Sequential aggregate signatures from trapdoor permutations, in: C. Cachin, J. Camenisch (Eds.), EUROCRYPT04, in: Lecture Notes in Computer Science, vol. 3027, Springer, May 2004, pp. 74–90.
[32] S. Micali, Simple and fast optimistic protocols for fair electronic exchange, in: ACM Symposium on Principles of Distributed Computing, PODC 2003, ACM, 2003, pp. 12–19.
[33] J.M. Park, E.K. Chong, H.J. Siegel, Constructing fair-exchange protocols for e-commerce via distributed computation of RSA signatures, in: PODC 2003, ACM, 2003, pp. 172–181.
[34] G. Wang, An abuse-free fair contract signing protocol based on the RSA signature, in: Proceedings of 14th International Conference on World Wide Web, WWW 2005, ACM, 2005, pp. 412–421.
[35] G. Wang, An abuse-free fair contract signing protocol based on the RSA signature, IEEE Trans. Inf. Forensics Secur. 5 (1) (March 2010) 158–168.
[36] Y. Wang, M.H. Au, W. Susilo, Perfect ambiguous optimistic fair exchange, in: Proceedings of ICICS 2012, in: Lecture Notes in Computer Science, vol. 7618, Springer, 2012, pp. 142–153.
[37] Y. Wang, M.H. Au, W. Susilo, Attribute-based optimistic fair exchange: how to restrict brokers with policies, Theoret. Comput. Sci. 527 (2014) 83–96.
[38] H. Zhu, Constructing optimistic fair exchange protocols from committed signatures, Cryptology ePrint Archive, Report 2005/012, http://eprint.iacr.org/, 2003.
[39] H. Zhu, F. Bao, Stand-alone and setup-free verifiably committed signatures, in: CT-RSA06, in: Lecture Notes in Computer Science, vol. 3860, Springer, 2006, pp. 159–173.
[40] H. Zhu, W. Susilo, Y. Mu, Multi-party stand-alone and setup-free verifiably committed signatures, in: PKC07, in: Lecture Notes in Computer Science, vol. 4450, Springer, 2007, pp. 134–149.