The Construction of Ambiguous Optimistic Fair Exchange from Designated Confirmer Signature without Random Oracles*

Qiong Huang[1], Duncan S. Wong[2], and Willy Susilo[3]

[1] South China Agricultural University, Guangzhou 510642, China.
[2] City University of Hong Kong, Hong Kong S.A.R., China.
[3] University of Wollongong, Wollongong, NSW 2522, Australia.
csqhuang@alumni.cityu.edu.hk, duncan@cityu.edu.hk, wsusilo@uow.edu.au

Abstract. Ambiguous Optimistic Fair Exchange (AOFE), introduced by Huang *et al.* in ASIACRYPT 2008, is an extension of OFE that enhances the fairness of the two communicating parties in the exchange of signatures. The first scheme was proven secure without random oracles while its partial signature contains dozens of group elements. Recently, interactive AOFE was introduced and the construction is more practical, where one partial signature only contains three group elements. It is based on the existence of Designated Confirmer Signature (DCS) with a special property where one is able to sample a confirmer signature efficiently from a signer's signature space. Nevertheless, we note that there are only a few DCS schemes that have this special property. Security of the interactive AOFE construction relies on the q-Computational and Decisional Hidden Strong Diffie-Hellman assumptions. In this paper, we propose a new construction of interactive AOFE from DCS, where the underlying DCS is standard and does not require any special property. We also propose a new DCS construction. By applying our transformation from DCS to interactive AOFE, we build a concrete interactive AOFE which is secure under more standard number-theoretic assumptions, namely Strong Diffie-Hellman and Decision Linear assumptions, without random oracles. A partial signature of the interactive AOFE contains six group elements, while a full signature contains two only.

Keywords: optimistic fair exchange, ambiguity, designated confirmer signature, standard model

* This work is supported by the National Natural Science Foundation of China (No. 61103232), the Research Fund for the Doctoral Program of Higher Education of China (No. 20114404120027), and the Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (No. LYM11033). D. S. Wong is supported by a grant from the RGC of the HKSAR, China (Project No. CityU 123511). W. Susilo is supported by ARC Future Fellowship FT0991397.

1 Introduction

How to exchange items between parties so that either both the parties get their counterpart's item or none of them does, is an important problem in e-commerce. Optimistic Fair Exchange (OFE), introduced by Asokan, Schunter and Waidner [1], is a kind of protocols for exchanging items between two parties say, Alice and Bob, in a fair manner. There is an arbitrator, which is semi-trusted by Alice and Bob and gets involved only when a party attempts to cheat the other or simply crashes. Asokan, Shoup and Waidner later proposed an OFE protocol for exchanging digital signatures [2]. In a typical run of OFE, Alice sends a *partial signature* to Bob, who in turn sends back his *full signature*, which triggers Alice to complete the protocol by releasing her full signature to Bob. If everything goes well, Alice and Bob should get each other's full signatures. However, if Alice refuses or fails to respond in the third move, Bob then resorts to the arbitrator for resolving Alice's partial signature into a full one. Since the introduction, OFE has attracted the attention of many researchers, i.e. [3, 13, 14, 23, 30].

In OFE, Alice's partial signature is generally self-authenticating and indicates her commitment to some message already. This may allow Bob to convince others that Alice has already committed herself to the message; while Alice obtains nothing. This could be unfair to Alice. Huang *et al.* [22] addressed this problem and proposed the notion of *ambiguous optimistic fair exchange* (AOFE), which is similar to the notion of *abuse-free optimistic contract signing* introduced by Garay *et al.* [15]. Different from the traditional OFE, AOFE enjoys the property of *signer ambiguity*. That is, Bob is able to produce partial signatures which are indistinguishable to those produced by Alice. Because of this property, given a valid partial signature from Alice, Bob cannot transfer its conviction to others any more.

1.1 Our Contributions

In this paper, we propose a new efficient and yet generic construction of AOFE from a primitive called designated confirmer signature (DCS) [10]. Compared with previous work on the construction of AOFE from DCS, e.g. [20, 21], our construction makes use of standard security properties of the underlying primitive, rather than any special property, e.g. *samplability* [20, 21]. Our AOFE protocol is interactive in the sense that the partial signature generation needs an interaction between the signer and the verifier. Below we give an intuition.

To partially sign a message, the signer produces a confirmer signature on it and then carries out a zero-knowledge proof with the verifier to show that the confirmer signature belongs to either the signer or the verifier. Thanks to the anonymity of the DCS scheme and the zero-knowledge property of the proof, a third party (except the arbitrator) cannot tell who is the real signer of the signature. We show that the resulting interactive AOFE protocol is secure in

the registered-key model[4] [4] without random oracles if the underlying DCS scheme is secure and the proof is sound and zero-knowledge.

To instantiate our construction of AOFE, we present a concrete and efficient DCS scheme, which is secure based on Strong Diffie-Hellman assumption [6] and Decision Linear assumption [7] without random oracles. It has short signatures and keys, and the confirmation/disavowal protocol is practically efficient. Compared with [20, 21], our scheme has much shorter signer public key and its security relies on relatively more standard assumptions. Compared with [33], the signature of our scheme is shorter, and the confirmation/disavowal protocol is more efficient. In Table 1 we give a comparison of our scheme with some existing DCS schemes in terms of sizes of confirmer public key **cpk**, confirmer secret key **csk**, signer public key **spk**, signer secret key **ssk**, confirmer signature σ and standard signature ζ, the underlying assumptions and the need of random oracles for security.

	[8]	[31]	[33]	[20, 21]	Ours
cpk	$5\mathbb{Z}_p$	$1\mathbb{G}$	$2\|N\| + 1\|n\| + 4\|\mathbb{Z}_{n^2}\|$	$1\mathbb{G}$	$4\mathbb{G}$
csk	$5\mathbb{Z}_q$	$1\mathbb{Z}_p$	$3(\kappa + \kappa_r)$	$1\mathbb{Z}_p$	$2\mathbb{Z}_p$
spk	$1\mathbb{Z}_N + k$	$1\mathbb{G}$	$\|N_0\| + \|N_1\| + 1SQ_{N_0} + 1SQ_{N_1}$	$163\mathbb{G}$	$2\mathbb{G}$
ssk	$1\mathbb{Z}_N$	$1\mathbb{Z}_p$	2κ	$1\mathbb{Z}_p$	$2\mathbb{Z}_p$
σ	$4\mathbb{Z}_p \approx 4K$	$2\mathbb{G} + 4\mathbb{Z}_p \approx 0.95K$	$9\mathbb{Z}_{n^2} + 1\mathbb{Z}_{N_0} + 1\mathbb{Z}_{N_1} \approx 22K$	$3\mathbb{G} \approx 0.47K$	$5\mathbb{G} + 1\mathbb{Z}_p \approx 0.96K$
ζ	$1\mathbb{Z}_N \approx 1K$	$2\mathbb{G} + 6\mathbb{Z}_p \approx 1.2K$	$3SQ_{n^2} + 1\mathbb{Z}^*_{N_0} + 1\mathbb{Z}^*_{N_1} \approx 10K$	$3\mathbb{G} \approx 0.47K$	$1\mathbb{G} + 1\mathbb{Z}_p \approx 0.32K$
Asmp	RSA+DDH	DDH+EUF-CMA	SRSA+DCRA+DDH	HSDH+DHSDH	SDH+DLIN
ROM	no	yes	no	no	no

Legends:

DLIN	: Decision Linear Assumption
DDH	: Decision Diffie-Hellman
SRSA	: Strong RSA Assumption
DCRA	: Decision Composite Residuosity Assumption
HSDH	: Hidden Strong Diffie-Hellman Assumption
DHSDH	: Decision Hidden Strong Diffie-Hellman Assumption
EUF-CMA	: Existential unforgeability (under chosen message attacks) of the underlying signature scheme

Security Parameters:
$\kappa = |n| = |N| = 1024$, $|N_0| = |N_1| = 2048$, $\kappa_r = 50$, $|\mathbb{G}| \approx 163$, $|\mathbb{Z}_p| \approx 163$ (for [8], we choose $|\mathbb{Z}_p| \approx 1024$).

Table 1. Comparison with some existing DCS schemes

1.2 Paper Organization

In the next section we review the related works on designated confirmer signature and optimistic fair exchange. The definition and security models of ambiguous optimistic fair exchange are given in Section 3. Our construction of interactive AOFE is then proposed in Section 4, followed by a section which gives the security analysis. In Section 6 we propose an efficient construction of designated

[4] In this model the adversary has to prove its knowledge of the secret key before using a public key.

confirmer signature, the security of which does not rely on the random oracle model. In Section 7 we compare our AOFE scheme with some existing schemes. The paper is concluded in Section 8.

2 Related Work

Designated Confirmer Signature. The notion of designated confirmer signature was proposed by Chaum [10] to alleviate the burden of the signer in undeniable signature [9]. In DCS, the signer designates a confirmer to confirm or disavow signatures for him, and the verifier cannot verify signatures alone. If a DCS scheme is *convertible*, the confirmer has the ability to extract the signer's standard signature from a valid confirmer signature. There have been a lot works on DCS since its introduction, e.g. [8,11,16,17,20,21,27,28,31–34]. Readers can refer to [20,21] for a brief review of the previous works.

The *de facto* security properties of a DCS scheme include *unforgeability* and *anonymity*. The former requires that no one but the signer is able to produce valid signatures; while the latter says that given a confirmer signature, no verifier is able to distinguish the identity of the signer. A popular approach in the design of DCS is known as the '*sign-then-encrypt*' paradigm. Intuitively, the confirmer holds a key pair $(\mathtt{Pk}_E, \mathtt{Sk}_E)$ for an encryption scheme E and the signer holds a key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$ for a signature scheme Σ. To sign a message M w.r.t. the confirmer, the signer computes a standard signature ζ on M using $\mathtt{Sk}_E$, and encrypts ζ under the confirmer's public key $\mathtt{Pk}_S$ to obtain the ciphertext C. Its confirmer signature is set to be C. To convert a confirmer signature C, the confirmer decrypts it to ζ using $\mathtt{Sk}_E$, and outputs ζ if it is valid under $\mathtt{Pk}_S$. In the confirmation (resp. disavowal) protocol, the confirmer proves to the verifier (interactively) that the confirmer signature C can (resp. cannot) be decrypted to a valid signature of the signer on message M. The unforgeability of the DCS scheme simply follows that of Σ. On the other hand, the (chosen ciphertext) security of E guarantees that given a ciphertext C, anyone who does not known $\mathtt{Sk}_E$, including the signer, is not able to tell C contains which signer's signature. Thus we have the anonymity.

Many DCS schemes follow this paradigm, e.g. [8,16,17]. The difficulty of implementing the paradigm is in the design of confirmation and disavowal protocols so that the scheme is efficient enough for practical use. It is known that the protocols can be constructed in general, using complex NP reduction. However, the efficiency is a big issue. As far as we know, there are only a few DCS schemes which have efficient confirmation and disavowal protocols, e.g. [16,20,21,31,33,34].

Wikström [33] revisited the aforementioned paradigm of constructing DCS, and proposed a similar generic construction, which makes use of a weak variant of CCA-secure cryptosystem, a signature scheme, and a weak form of zero-knowledge proofs. A concrete instantiation was also presented in [33], which is built from Cramer-Shoup version of Paillier encryption [29] and a twin-moduli signature. The confirmation and disavowal protocols, although do not involve any NP-reduction, are not efficient enough. The prover and the verifier have to

carry out a bunch of proofs of knowledge, and both of them need perform more than 150 exponentiation evaluations.

Huang *et al.* [20, 21] proposed a new variant of DCS, in which both the signer and the confirmer are able to not only confirm but also disavow signatures efficiently. They also presented a concrete construction, the security of which is based on new number-theoretic assumptions (e.g. Hidden Strong Diffie-Hellman assumption and Decision Hidden Strong Diffie-Hellman assumption) without random oracles. The new variant is useful in applications in which the signer prefers to retain the ability to disavow signatures, whereas there are still some cases in which the signer only wants to keep the ability of confirmation.

Ambiguous Optimistic Fair Exchange. Garay *et al.* [15] for the first time addressed the problem with the non-repudiation of a partial signature, and proposed an efficient abuse-free contract signing protocol, in which no one but the arbitrator can distinguish who produced which signature. The protocol makes use of a type of signatures called '*private contract signatures*', which is similar to but different from DCS. Their private contract signature scheme is built from designated-verifier signature [24], and is secure based on DDH assumption in the random oracle model [5] and the registered-key model [4], in which the adversary has to show its knowledge of the secret key before using a public key.

Huang *et al.* [22] proposed an efficient construction of AOFE based on the group signature scheme in [18]. Their scheme uses (the weakly secure) Boneh-Boyen signature [6] and Groth-Sahai non-interactive proof techniques [19]. The scheme is secure based on Strong Diffie-Hellman assumption [6] and Decision Linear assumption [7] in the chosen-key model [23,26] without random oracles, in which the adversary is allowed to use public keys arbitrarily. However, the scheme suffers from long signatures, which consist of more than 40 group elements.

Very recently, Huang *et al.* [20, 21] proposed a new approach to constructing *interactive* AOFE, in which the signer interacts with the verifier to produce the partial signature. Their construction applies to a specific class of DCS schemes, in which anyone is able to sample confirmer signatures from the signer's signature space efficiently, e.g. in polynomial time. However, not many DCS schemes enjoy this property, and thus limiting the application of Huang *et al.*'s construction. They also instantiated the construction using the DCS scheme proposed in the same paper. The resulting interactive protocol is secure without random oracles in the registered-key model.

3 Ambiguous Optimistic Fair Exchange

3.1 Definition

Essentially, AOFE is a variant of the traditional OFE, in which both of the exchanging parties can produce indistinguishable signatures on the same message. An AOFE scheme consists of the following probabilistic polynomial time algorithms/protocols:

PMGen. It takes 1^k as input where k is the security parameter and outputs the system parameter PM.

Setup$^{\mathbf{TTP}}$. It takes as input the system parameter PM and outputs a key pair for the arbitrator. We denote it by $(\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(1^k)$.

Setup$^{\mathbf{User}}$. It takes the system parameter PM (and optionally $\mathtt{Apk}$) as input and outputs a key pair for the user. We denote it by $(\mathtt{Pk}, \mathtt{Sk}) \leftarrow \mathsf{Setup}^{\mathsf{User}}(1^k, \mathtt{Apk})$.

PSig. This is the partial signature generation algorithm. It takes as input a message M, the signer's secret key $\mathtt{Sk}_i$, the signer's public key $\mathtt{Pk}_i$, the verifier's public key $\mathtt{Pk}_j$ and the arbitrator's public key $\mathtt{Apk}$, and outputs a partial signature σ. We denote it by $\sigma \leftarrow \mathsf{PSig}(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

PVer. This is for the verification of a partial signature. It can be either an algorithm or a protocol, depending on whether the verification requires the interaction between the signer and the verifier or not. The (common) input consists of $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$. If the verification is interactive, the signer has private input $\mathtt{Sk}_i$. We denote it by $b \leftarrow \mathsf{PVer}(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$, where b is the output of the verifier, which is 1 for acceptance and 0 for rejection.

Sig. This is the full signature generation algorithm. It takes as input $(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$ and outputs a full signature ζ. We denote it by $\zeta \leftarrow \mathsf{Sig}(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

Ver. This is for the verification of a full signature. It takes as input $(M, \zeta, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$ and outputs a bit b which is 1 if ζ is a valid full signature of $\mathtt{Pk}_i$ and 0 otherwise. We denote it by $b \leftarrow \mathsf{Ver}(M, \zeta, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$.

Res. This is for resolving a partial signature. It takes as input $(M, \mathtt{Ask}, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$ and outputs ζ if ζ is a valid full signature of $\mathtt{Pk}_i$, and $\perp$ otherwise.

The AOFE introduced in [22] is *non-interactive* in the sense that all the signature generation and verification algorithms are non-interactive. However, in this work we consider *interactive* AOFE (iAOFE in short), in which the partial signature verification is an interactive protocol between the signer and the verifier. For simplicity we treat PVer as a protocol universally for both interactive and non-interactive AOFE. If the scheme is non-interactive, then in the PVer protocol (which should be an algorithm) the signer does nothing and the verifier makes the decision alone.

3.2 Security Models

The security of AOFE was originally defined in the *chosen-key* model [22], in which the adversary is allowed to use any public key arbitrarily without showing its knowledge of the corresponding secret key. While in this work we consider AOFE in the *registered-key* model [4], which is weaker than the chosen-key model yet still practical.

Registered-Key Model. In this model the adversary has to prove its knowledge of the corresponding secret key before using a public key. Although this model puts limits to the adversary on using public keys, it is still a practical

model, and has been considered in many works, such as [13, 25]. Usually, an adversary in this model conducts a proof of knowledge of the secret key to the game challenger or simply submits the key pair or even the randomness used in key generation. In the rest of the paper we assume that the adversary has access to a key registration oracle O_{KR}, which takes as input a key pair $(\mathtt{Pk}, \mathtt{Sk})$, and returns $\mathtt{Pk}$ if the pair is a valid output of the key generation algorithm and $\perp$ otherwise.

Let $\mathcal{Q}(O)$ be the set of queries that the adversary submits to oracle O, where O could be any of the oracles below.

- O_{KR} is the key registration oracle.
- O_{PSig} takes as input $(M, \mathtt{Pk}_i)$ and returns a partial signature σ of the signer with public key $\mathtt{Pk}_A$, which is valid on M under $\mathtt{Pk}_A$, $\mathtt{Pk}_i$. The oracle then starts an execution of PVer with the adversary to show the validity of σ.
- O_{FakePSig} takes as input $(M, \mathtt{Pk}_i)$ and returns a partial signature σ generated using $\mathtt{Sk}_B$ and valid under $\mathtt{Pk}_i$, $\mathtt{Pk}_B$. The oracle then starts an execution of the PVer protocol with the adversary to show the validity of σ.
- O_{Res} takes as input $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$ and outputs ζ if it is a valid (standard) signature on M under $\mathtt{Pk}_i$, and $\perp$ otherwise.

If the public key submitted to any of O_{PSig}, O_{FakePSig} and O_{Res} was not ever submitted to O_{KR}, these oracles would simply return nothing to the adversary.

Signer Ambiguity. The signer ambiguity says that after obtaining the valid partial signature from the signer S, the verifier V cannot transfer the conviction to any third party. We require that V is able to produce signatures indistinguishable from those by S. Formally, we consider the game $\mathbb{G}_{\mathrm{sa}}$ depicted in Figure 1 (page 9), where Υ is $\mathcal{D}$'s state information. Note that after sending σ^* to $\mathcal{D}$ in the game, the challenger also starts an execution of the PVer protocol with $\mathcal{D}$ to show the validity of σ^* under $\mathtt{Pk}_A, \mathtt{Pk}_B$. The advantage of $\mathcal{D}$, denoted by $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k)$, is defined to be the gap between its success probability in the game and one half, i.e. $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k) = |\Pr[\mathcal{D}\ \mathrm{Succ}] - 1/2|$.

Definition 1 (Signer Ambiguity). *An AOFE scheme is* signer ambiguous *if there is no PPT distinguisher $\mathcal{D}$ such that $\mathtt{Adv}^{\mathrm{sa}}_{\mathcal{D}}(k)$ is non-negligible in k.*

Security Against Signers. It requires that (malicious) signer $\mathcal{A}$ cannot produce a partial signature, which looks good to V but cannot be resolved to a full signature by the honest arbitrator, ensuring the fairness for verifiers. V should always be able to obtain the full commitment of the signer if the signer has committed to a message. Formally, we consider the game $\mathbb{G}_{\mathrm{sas}}$ depicted in Figure 1. The advantage of $\mathcal{A}$ in the game, denoted by $\mathtt{Adv}^{\mathrm{sas}}_{\mathcal{A}}(k)$, is defined as its success probability.

Definition 2 (Security Against Signers). *An AOFE scheme is* secure against signers *if there is no PPT adversary $\mathcal{A}$ such that $\mathtt{Adv}^{\mathrm{sas}}_{\mathcal{A}}(k)$ is non-negligible in k.*

Security Against Verifiers. It requires that any efficient verifier $\mathcal{B}$ should not be able to convert a partial signature into a full one with non-negligible probability if it obtains no help from the signer or the arbitrator. This ensures the fairness for the arbitrator and the signer. Formally, we consider the game $\mathbb{G}_{\text{sav}}$ depicted in Figure 1. The advantage of $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ in the game, denoted by $\mathtt{Adv}^{\text{sav}}_{\mathcal{B}}(k)$, is defined as its success probability.

Definition 3 (Security Against Verifiers). *An AOFE scheme is* secure against verifiers *if there is no probabilistic polynomial-time adversary $\mathcal{B}$ such that $\mathtt{Adv}^{\text{sav}}_{\mathcal{B}}(k)$ is non-negligible in k.*

Security Against the Arbitrator. This is for ensuring the unforgeability of the signer's signatures. It says that no efficient adversary $\mathcal{C}$, even the arbitrator, is able to generate with non-negligible probability a valid full signature without explicitly asking the signer for generating one. Formally, we consider the game $\mathbb{G}_{\text{saa}}$ depicted in Figure 1. The advantage of $\mathcal{C}$ in this game, denoted by $\mathtt{Adv}^{\text{saa}}_{\mathcal{C}}(k)$, is defined as its success probability.

Definition 4 (Security Against the Arbitrator). *An AOFE scheme is* secure against the arbitrator *if there is no PPT adversary $\mathcal{C}$ such that $\mathtt{Adv}^{\text{saa}}_{\mathcal{C}}(k)$ is non-negligible in k.*

Remark 1. Our definitions of signer ambiguity and security against verifiers are slightly weaker than those considered in [20–22]. In our definition of signer ambiguity, the two challenge public keys $(\mathtt{Pk}_A, \mathtt{Pk}_B)$ (along with their corresponding secret keys) are given to the adversary, rather than the adversary chooses one of them as [20, 21] does. Similarly, in our definition of security against verifiers, the two public keys are also chosen by the challenger and given to the adversary. Nevertheless, it is this slight weakening in the security which enables us to construct AOFE protocols from DCS schemes with standard security properties instead of any special property like samplability [20, 21], in a general way, as we shall see in Section 4.

Similar to [22], it is straightforward to establish the relation between security against verifiers and signer ambiguity. We have the following lemma and therefore, we need not consider security against verifiers in proving the security of an AOFE scheme.

Lemma 1. *If an AOFE scheme is both signer ambiguous (Definition 1) and secure against the arbitrator (Definition 4), it is secure against verifiers (Definition 3) as well.*

Definition 5 (Secure AOFE). *An AOFE scheme is said to be* secure in the multi-user setting and registered-key model *(or simply,* secure*), if it satisfies signer ambiguity (Definition 1), security against signers (Definition 2), and security against the arbitrator (Definition 4).*

Game $\mathbb{G}_{\mathrm{sa}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \Upsilon) \leftarrow \mathcal{D}^{O_{\mathsf{KR}}, O_{\mathsf{Res}}}(\mathtt{Apk}, (\mathtt{Pk}_A, \mathtt{Sk}_A), (\mathtt{Pk}_B, \mathtt{Sk}_B)), \quad b \leftarrow \{0,1\},$

$$\sigma^* \leftarrow \begin{cases} \mathsf{PSig}(M^*, \mathtt{Sk}_A, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) & \text{if } b = 0 \\ \mathsf{FakePSig}(M^*, \mathtt{Sk}_B, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) & \text{otherwise} \end{cases},$$

$b' \leftarrow \mathcal{D}^{O_{\mathsf{KR}}, O_{\mathsf{Res}}}(\Upsilon, \sigma^*),$

Succ. of $\mathcal{D} := [b' = b \wedge (M^*, \sigma, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}) \notin \mathcal{Q}(O_{\mathsf{Res}})]$.

Game $\mathbb{G}_{\mathrm{saa}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (M^*, \mathtt{Pk}_B, \zeta^*) \leftarrow \mathcal{C}^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}}(\mathtt{Ask}, \mathtt{Apk}, \mathtt{Pk}_A),$

Succ. of $\mathcal{C} := [\mathsf{Ver}(M^*, \zeta, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 1 \wedge (M^*, \mathtt{Pk}_B) \notin \mathcal{Q}(\mathcal{O}_{\mathsf{PSig}})]$.

Game $\mathbb{G}_{\mathrm{sas}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \mathtt{Pk}_A, \sigma^*) \leftarrow \mathcal{A}^{O_{\mathsf{KR}}, O_{\mathsf{FakePSig}}, O_{\mathsf{Res}}}(\mathtt{Apk}, \mathtt{Pk}_B), \quad \zeta^* \leftarrow \mathsf{Res}(M^*, \sigma^*, \mathtt{Ask}, \mathtt{Pk}_A, \mathtt{Pk}_B),$

Succ. of $\mathcal{A} := [\mathsf{PVer}(M^*, \sigma^*, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}, \mathtt{Apk}) = 1$

$\wedge\, \mathsf{Ver}(M^*, \zeta^*, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 0 \wedge (M^*, \mathtt{Pk}_A) \notin \mathcal{Q}(O_{\mathsf{FakePSig}})]$.

Game $\mathbb{G}_{\mathrm{sav}}$:

$\mathsf{PM} \leftarrow \mathsf{PMGen}(1^k), \quad (\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathsf{Setup}^{\mathsf{TTP}}(\mathsf{PM}),$

$(\mathtt{Pk}_A, \mathtt{Sk}_A) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}), \quad (\mathtt{Pk}_B, \mathtt{Sk}_B) \leftarrow \mathsf{Setup}^{\mathsf{User}}(\mathsf{PM}, \mathtt{Apk}),$

$(M^*, \Upsilon) \leftarrow \mathcal{B}_1^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}, O_{\mathsf{Res}}}(\mathtt{Apk}, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Sk}_B),$

$\sigma^* \leftarrow \mathsf{PSig}(M^*, \mathtt{Sk}_A, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}), \quad \zeta^* \leftarrow \mathcal{B}_2^{O_{\mathsf{KR}}, O_{\mathsf{PSig}}, O_{\mathsf{Res}}}(\Upsilon, \sigma^*),$

Succ. of $\mathcal{B} := [\mathsf{Ver}(M^*, \zeta^*, \mathtt{Pk}_A, \mathtt{Pk}_B, \mathtt{Apk}) = 1 \wedge (M^*, \cdot, \{\mathtt{Pk}_A, \mathtt{Pk}_B\}) \notin \mathcal{Q}(O_{\mathsf{Res}})]$.

Fig. 1. Security models of AOFE

4 Our Construction of iAOFE

In this part we present a construction of interactive AOFE based on a designated confirmer signature (DCS) scheme. Before describing our construction, let us give a brief introduction of DCS first.

In DCS, there are a signer S, a verifier V and a confirmer C. S and C run algorithms SKg and CKg to produce their public/secret key pairs, respectively. The signer can run Sig algorithm to produce its standard signatures, which can be verified by V by calling the Ver algorithm. S can also run DCSig to produce confirmer signatures by designating C as the confirmer, which could not be verified by the verifier alone. To prove the validity/invalidity of a confirmer signature, C runs $\mathsf{Confirm}/\mathsf{Disavow}$ protocol with V. There are two confirmation protocols, $\mathsf{ConfirmS}$ and $\mathsf{ConfirmC}$, run by S and C to prove the validity of a confirmer signature, respectively. Besides, C is able to convert (valid) confirmer signatures to standard ones.

The Construction. Below we present our construction of interactive AOFE. Compared with previous work on the construction of AOFE from DCS, e.g. [20, 21], our construction makes use of the standard security properties of the underlying DCS, rather than any special property, e.g. *samplability* [20, 21]. Intuitively, in our construction of interactive AOFE, U_i's partial signature σ on a message M is simply its confirmer signature. Since the DCS scheme is anonymous, no one but the confirmer is able to tell σ was produced by U_i or U_j. Let Σ be a DCS scheme. Our AOFE scheme works as below, where U_i is the signer and U_j is the verifier.

PMGen. It generates all the necessary system parameters for Σ.

Setup$^{\mathbf{TTP}}$. The arbitrator computes $(\mathtt{Cpk}, \mathtt{Csk}) \leftarrow \Sigma.\mathsf{CKg}(1^k)$, and sets its key pair as $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$.

Setup$^{\mathbf{User}}$. Each user computes $(\mathtt{Spk}, \mathtt{Ssk}) \leftarrow \Sigma.\mathsf{SKg}(1^k)$, and sets its key pair as $(\mathtt{Pk}, \mathtt{Sk}) := (\mathtt{Spk}, \mathtt{Ssk})$.

PSsig. To partially sign a message M for U_j, U_i computes $\sigma \leftarrow \Sigma.\mathsf{DCSig}(\mathtt{Sk}_i, \hat{M})$, where $\hat{M} = M \| \mathtt{Pk}_j$, and sends σ to U_j.

PVer. Given a partial signature σ, U_i and U_j carry out an execution of a zero-knowledge proof Π which is the OR combination of two independent copies of $\Sigma.\mathsf{ConfirmS}$, to show that σ is a valid confirmer signature on $\hat{M}$ of either U_i or U_j. U_i plays the role of the prover in the proof. U_j outputs 1 if it accepts at the end of the proof, and 0 otherwise.

Sig. To fully sign a message M, U_i computes $\zeta \leftarrow \Sigma.\mathsf{Sig}(\mathtt{Sk}_i, \hat{M}, \mathtt{Apk})$, and sends it to U_j.

Ver. Given a full signature ζ, U_j outputs $\Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$.

Res. Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, the arbitrator computes and returns $\zeta \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}, \sigma, \mathtt{Pk}_i)$ to U_j if $1 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$ and $\perp$ otherwise.

Remark 2. As we can see from the construction above, the resulting interactive AOFE protocol is solely based on the underlying DCS scheme. The proof run between the signer and the verifier is an OR composition of two copies of the confirmation protocol of the DCS scheme. There are standard technique of composing (the Σ-protocol [12] version of) the confirmation protocol.

The correctness of the construction above is obvious, and we skip the details here. In the next section we analyze the security of the construction under the models given in Fig. 1.

5 Security Analysis

Since our AOFE protocol is built from a DCS scheme, before proving the security of our AOFE scheme (under the models given in Sec. 3.2), let us briefly describe the security models of DCS.

A secure DCS scheme satisfies two security properties. One is unforgeability, which requires that no one but the signer be able to generate valid (standard) signatures. Even the confirmer could not forge either. The other property is anonymity, which requires no one but the confirmer be able to tell a given confirmer signature was generated by which signer. If the signer does not store the signatures it ever produced, it cannot distinguish either. Due to the page limit we defer the detailed security definitions of DCS into the full version.

Now we begin to analyze the security of our construction of interactive AOFE. We have the following theorem.

Theorem 1. *The interactive AOFE scheme above is secure (Definition 5) provided that* Σ *is secure and the proof* Π *is sound and zero-knowledge.*

It follows the following lemmas immediately.

Lemma 2. *The interactive AOFE scheme is signer-ambiguous if* Σ *is anonymous and the proof* Π *is zero-knowledge.*

Proof. To simulate U_i's partial signature on a message M, U_j computes $\sigma' \leftarrow \Sigma.\mathsf{DCSig}(\mathtt{Sk}_j, \hat{M})$ where $\hat{M} = M \| \mathtt{Pk}_j$, and outputs σ' as the simulated partial signature. Guaranteed by the anonymity of Σ, we know that σ' looks indistinguishable from U_i's partial signature on M. Below we prove that the simulated signature is indistinguishable from the output of a real signer.

Let $\mathcal{D}$ be a distinguisher which can tell U_j's simulated signatures apart from U_i's real signatures with probability $1/2 + \epsilon$, where ϵ is non-negligible. We use it to build another algorithm $\mathcal{D}'$ for breaking the anonymity of Σ.

Given the system parameters, two key pairs $(\mathtt{Spk}_0, \mathtt{Ssk}_0), (\mathtt{Spk}_1, \mathtt{Ssk}_1)$ and a confirmer public key $\mathtt{Cpk}$, $\mathcal{D}'$ sets $\mathtt{Apk} := \mathtt{Cpk}$, $(\mathtt{Pk}_A, \mathtt{Sk}_A) := (\mathtt{Spk}_0, \mathtt{Ssk}_0)$ and $(\mathtt{Pk}_B, \mathtt{Sk}_B) := (\mathtt{Spk}_1, \mathtt{Ssk}_1)$, and invokes $\mathcal{D}$ on input $(\mathtt{Apk}, (\mathtt{Pk}_A, \mathtt{Sk}_A), (\mathtt{Pk}_B, \mathtt{Sk}_B))$. The oracles are simulated by $\mathcal{D}'$ as follows:

O_{KR}**.** Given a key pair $(\mathtt{Pk}, \mathtt{Sk})$, if it is not well-formed, $\mathcal{D}'$ returns $\perp$; otherwise, it stores the pair and returns $\mathtt{Pk}$.

O_{Res}. Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, $\mathcal{D}'$ sets $\hat{M} := M \| \mathtt{Pk}_j$ and forwards $(\hat{M}, \sigma, \mathtt{Pk}_i)$ to its extraction oracle, which returns ζ. It returns $\perp$ to the distinguisher if $\zeta = \perp$ or $0 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}, \zeta, \mathtt{Pk}_i, \mathtt{Apk})$, and ζ otherwise.

When $\mathcal{D}$ submits a challenge message M^*, $\mathcal{D}'$ forwards $\hat{M}^* := M^* \| \mathtt{Pk}_B$ to its own challenger, which tosses a coin b and returns a confirmer signature σ^* on $\hat{M}^*$ valid under $\mathtt{Spk}_b$. It then sends σ^* to $\mathcal{D}$, and runs the simulator of protocol Π to prove that σ^* is a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$. The disgintuisher continues to issuing queries, which are handled by $\mathcal{D}'$ as above. Finally, $\mathcal{D}'$ outputs the bit b' that $\mathcal{D}$ outputs.

Assume that $\mathcal{D}$ wins its game, and thus it did not send a query on input $(M^*, \sigma^*, \{\mathtt{Pk}_A, \mathtt{Pk}_B\})$ to the resolution oracle. Hence, $\mathcal{D}'$ did not make an extraction query on $(\hat{M}^*, \sigma^*, \mathtt{Spk}_0)$ nor $(\hat{M}^*, \sigma^*, \mathtt{Spk}_1)$, and wins its own game as well.

The view of $\mathcal{D}$ in this simulated game is the same as that in a real attack, except that the proof of the validity of σ^*. However, since the protocol Π is zero-knowledge, the simulated proof causes only a negligible difference to the view of $\mathcal{D}$, denoted by δ. Therefore, if $\mathcal{D}$ breaks the signer ambiguity with non-negligible advantage ϵ, $\mathcal{D}'$ breaks the anonymity of Σ with advantage at least $\epsilon - \delta$, which is non-negligible as well. □

Lemma 3. *The interactive AOFE scheme is secure against signers if* Σ *is sound and unforgeable and* Π *is sound.*

Proof. Let $\mathcal{A}$ be a malicious signer which can break the security against signers with non-negligible probability. We make use of it to construct another algorithm $\mathcal{A}'$ to break the unforgeability of Σ.

Given $(\mathtt{Cpk}, \mathtt{Csk}, \mathtt{Spk}^*)$, algorithm $\mathcal{A}'$ sets $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$ and $\mathtt{Pk}_B := \mathtt{Spk}^*$, and invokes the adversary $\mathcal{A}$ on input $(\mathtt{Apk}, \mathtt{Pk}_B)$. It then begins to simulate the oracles for $\mathcal{A}$ as below:

O_{KR}. Same as in the proof of Proposition 2.

O_{FakePSig}. Given $(M, \mathtt{Pk}_i)$, if $\mathtt{Pk}_i \neq \mathtt{Pk}_B$, $\mathcal{A}'$ computes the simulated signature σ using its knowledge of $\mathtt{Sk}_i$, since we are working in the registered-key model. Otherwise, it forwards $M \| \mathtt{Pk}_B$ to its signing oracle, and obtains a confirmer signature σ. In either case, $\mathcal{A}'$ returns σ to $\mathcal{A}$.

O_{Res}. Given $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$, $\mathcal{A}'$ perfectly computes the answer using its knowledge of $\mathtt{Ask}$.

Finally, $\mathcal{A}$ outputs $(M^*, \mathtt{Pk}_A, \sigma^*)$, and starts an execution of Π with $\mathcal{A}'$ to show that σ^* is a valid confirmer signature on $\hat{M}^* := M^* \| \mathtt{Pk}_B$ under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$. $\mathcal{A}'$ then computes

$$\zeta_A^* \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}^*, \sigma^*, \mathtt{Pk}_A) \quad \text{and} \quad \zeta_B^* \leftarrow \Sigma.\mathsf{Ext}(\mathtt{Ask}, \hat{M}^*, \sigma^*, \mathtt{Pk}_B).$$

Suppose $\mathcal{A}$ wins the game. By the soundness of Π, we have that with overwhelming probability σ^* is indeed a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_B$,

but ζ^* is not a valid standard signature under $\mathtt{Pk}_A$. Therefore, it holds that ζ_B^* is a valid standard signature under $\mathtt{Pk}_B$. $\mathcal{A}'$ then outputs $(\hat{M}^*, \zeta_B^*)$, and wins the game. If $\mathcal{A}$ succeeds in breaking the security against signers with non-negligible probability, so does $\mathcal{A}'$ in breaking the unforgeability of Σ. □

Lemma 4. *The interactive AOFE scheme is secure against the arbitrator if* Σ *is unforgeable and* Π *is zero-knowledge.*

Proof. Let $\mathcal{C}$ be a malicious arbitrator. Below we show how to use it to build an algorithm $\mathcal{C}'$ to break the unforgeability of Σ.

Given $(\mathtt{Spk}^*, \mathtt{Cpk}, \mathtt{Csk})$, $\mathcal{C}'$ sets $\mathtt{Pk}_A = \mathtt{Spk}^*$ and $(\mathtt{Apk}, \mathtt{Ask}) := (\mathtt{Cpk}, \mathtt{Csk})$, and invokes $\mathcal{C}$ on input $(\mathtt{Ask}, \mathtt{Apk}, \mathtt{Pk}_A)$. The oracle queries are answered by $\mathcal{C}'$ as below:

O_{KR}**.** Same as in the proof of Proposition 2.

O_{PSig}**.** Given $(M, \mathtt{Pk}_j)$, $\mathcal{C}'$ forwards $\hat{M} := M \| \mathtt{Pk}_j$ to its signing oracle and obtains a confirmer signature σ. It sends σ to $\mathcal{C}$ and then runs the simulator to prove to $\mathcal{C}$ that σ is a valid confirmer signature under either $\mathtt{Pk}_A$ or $\mathtt{Pk}_j$.

Finally, $\mathcal{C}$ outputs $(M^*, \mathtt{Pk}_B, \zeta^*)$. Suppose that it wins the game. We have that $1 \leftarrow \Sigma.\mathsf{Ver}(\hat{M}^*, \zeta^*, \mathtt{Spk}, \mathtt{Cpk})$, where $\hat{M}^* := M^* \| \mathtt{Pk}_B$. By the hypothesis, $\mathcal{C}$ did not issue a partial signing query on input $(M^*, \mathtt{Pk}_B)$, and hence $\mathcal{C}'$ did not send $\hat{M}^*$ to its signing oracle for a confirmer signature. If $\mathcal{C}$ succeeds in breaking the security against the arbitrator, so does $\mathcal{C}'$ in breaking the unforgeability of Σ with at least the same advantage. □

6 A New Construction of Designated Confirmer Signature

The unforgeability of DCS requires that no one but the signer can produce valid signatures, while the anonymity requires no one but the designated confirmer can tell the validity of a given confirmer signatures. Hence it is very natural to construct a DCS from a standard signature scheme Σ and a public key encryption scheme E, which is also known as the '*sign-then-encrypt*' paradigm. Many constructions of DCS follow this paradigm, such as [16, 17, 33] and etc. The difficulty of implementing the paradigm is in the design of confirmation and disavowal protocol. It is known that the protocols can be constructed generally, using complex NP reduction. However, the efficiency is a big issue. The resulting protocols may not be useful in practice.

Intuitively, in the paradigm, the confirmer holds a key pair $(\mathtt{Pk}_E, \mathtt{Sk}_E)$ for E and the signer holds a key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$ for Σ. To sign a message M with respect to the confirmer, the signer first computes a standard signature ζ on M using $\mathtt{Sk}_E$, and then encrypts ζ under the confirmer's public key $\mathtt{Pk}_S$ to obtain the ciphertext c. Its confirmer signature is set to be c. Given c, the confirmer uses $\mathtt{Sk}_E$ to decrypt it to obtain ζ, and outputs it if it is valid under $\mathtt{Pk}_S$. In the confirmation (resp. disavowal) protocol, the confirmer proves to the verifier

(interactively) that a confirmer signature c can (resp. cannot) be decrypted to the signer's valid signature on M. The unforgeability of the DCS scheme simply follows that of Σ. On the other hand, the chosen ciphertext security of E guarantees that given a ciphertext c, anyone who does not known $\mathtt{Sk}_E$, including the signer, is not able to tell the signature hidden in c belongs to which signer. Thus we have the anonymity.

Below we present a concrete and efficient instantiation of the above paradigm, which is based on Boneh-Boyen signature [6] and a variant of the linear encryption [7]. The confirmation and disavowal protocols in the construction are simple and efficient, and do not use any complex reduction. In the scheme we assume that the message space is $\mathbb{Z}_p$ for simplicity. The space can be extended to $\{0,1\}^*$ by applying a collision-resistant hash function to the message before signing.

6.1 The Construction

Let $\mathbb{G}, \mathbb{G}_T$ be two cyclic multiplicative groups of prime order p, and g a random generator of $\mathbb{G}$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an admissible bilinear pairing and $\mathtt{H} : \mathbb{G}^3 \to \mathbb{Z}_p$ a collision-resistant hash function. Our first DCS scheme, denoted by Σ, works as follows:

Ckg. The confirmer chooses at random $F, G, K, L \in \mathbb{G}$ so that $F^{\xi_1} = G^{\xi_2} = g$ for some known $\xi_1, \xi_2 \in \mathbb{Z}_p$. It then sets $\mathtt{Apk} = (F, G, K, L)$ and $\mathtt{Ask} = (\xi_1, \xi_2)$.

SKg. The signer chooses at random $x, y \in \mathbb{Z}_p$ and computes $X = g^x$, $Y = g^y$. It sets $\mathtt{Spk} = (X, Y)$ and $\mathtt{Ssk} = (x, y)$.

Sig. To sign a message M, the signer selects at random $r \in \mathbb{Z}_p$ and computes $S = g^{1/(x+M+yr)}$. In case that $x + M + yr = 0 \bmod p$, it chooses another r and repeats the computation. Its signature on M is $\zeta = (S, r)$.

Ver. Given (M, ζ) where $\zeta = (S, r)$, the verifier checks if $\hat{e}(S, Xg^M Y^r) = \hat{e}(g, g)$. It accepts if the equation holds, and rejects otherwise.

DCSig. Given a message M, the signer randomly selects $r, s, t \in \mathbb{Z}_p$ and computes

$$S = g^{1/(x+M+yr)}, \quad \sigma_1 = F^s, \quad \sigma_2 = G^t, \quad \sigma_3 = S \cdot g^{s+t},$$
$$\sigma_4 = (g^\alpha K)^s \quad \text{and} \quad \sigma_5 = (g^\alpha L)^t,$$

where $\alpha = \mathtt{H}(\sigma_1, \sigma_2, \sigma_3)$. Again, if $x + M + yr = 0 \bmod p$, the signer chooses another r and repeats the process. Its confirmer signature on M is $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$.

Ext. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer computes

$$S = \sigma_3 / (\sigma_1^{\xi_1} \sigma_2^{\xi_2}) \quad \text{and} \quad \alpha = \mathtt{H}(\sigma_1, \sigma_2, \sigma_3).$$

If either of the following equations does not hold, it returns $\perp$; otherwise, it returns $\zeta = (S, r)$:

$$\hat{e}(\sigma_4, F) = \hat{e}(\sigma_1, g^\alpha K) \tag{1}$$

$$\hat{e}(\sigma_5, G) = \hat{e}(\sigma_2, g^{\alpha} L) \tag{2}$$
$$\hat{e}(g, g) = \hat{e}(S, X g^{M} Y^{r}) \tag{3}$$

ConfirmS. To prove the validity of a confirmer signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$ on a message M that it ever generated, the signer makes use of the randomness (s, t) used in the signature generation to carry out the following proof of knowledge with the verifier

$$\text{PoK}\left\{(s,t) : F^{s} = \sigma_1 \wedge G^{t} = \sigma_2 \wedge \hat{e}(\sigma_3 g^{-s-t}, X g^{M} Y^{r}) = \hat{e}(g,g)\right\} \tag{4}$$

if both equations (1) and (2) hold, and does nothing otherwise.

ConfirmC. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer and the verifier carry out the following (zero-knowledge) proof of knowledge

$$\text{PoK}\left\{(\xi_1, \xi_2) : F^{\xi_1} = g \wedge G^{\xi_2} = g \wedge \hat{e}(\sigma_3 \sigma_1^{-\xi_1} \sigma_2^{-\xi_2}, X g^{M} Y^{r}) = \hat{e}(g,g)\right\} \tag{5}$$

if both equations (1) and (2) hold, and do nothing otherwise.

Disavow. Given (M, σ) where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, r)$, the confirmer and the verifier carry out an execution of the following (zero-knowledge) proof of knowledge

$$\text{PoK}\left\{(\xi_1, \xi_2) : F^{\xi_1} = g \wedge G^{\xi_2} = g \wedge \hat{e}(\sigma_3 \sigma_1^{-\xi_1} \sigma_2^{-\xi_2}, X g^{M} Y^{r}) \neq \hat{e}(g,g)\right\} \tag{6}$$

if both equations (1) and (2) hold, and do nothing otherwise.

The correctness and extraction ambiguity of the DCS scheme above can be verified trivially. The following theorem shows that the DCS scheme above is secure under the models given in Sec. 5.

Theorem 2. *The DCS scheme Σ is secure if Strong Diffie-Hellman assumption and Decision Linear assumption hold, and the hash function* `H` *is collision-resistant.*

Due to the page limit we defer the detailed proof of the theorem and the definitions of the assumptions into the full version.

6.2 Non-interactive AOFE

Huang *et al.*'s non-interactive AOFE is obtained by applying Fiat-Shamir heuristic to their interactive AOFE protocol, specifically, to the confirmation proof of the signature's validity. Via the same technique, we can obtain a non-interactive AOFE protocol as well.

7 Comparison

In Table 2 we compare the interactive AOFE protocol instantiated with the DCS scheme proposed in Section 6, with previous AOFE protocols. The second column shows if the protocol require interaction between the signer and the verifier in order to verify a partial signature. The third and fourth columns show the size of a partial signature and that of a full signature, respectively. The fifth column indicates whether the protocol works under the registered-key model or chosen-key model. The sixth column lists the basic number-theoretic assumptions used for guaranteeing the security. The last column shows whether the security of the protocols rely on the random oracle model or not.

Both of the interactive AOFE protocol proposed in [20,21] and ours are built from a DCS scheme, and are secure in the standard model. The protocol in [20,21] requires a special property of DCS, named *samplability*, while our protocol only makes use of standard security properties of the underlying DCS scheme. In the comparison, we consider that the signer's partial signature merely consists of its confirmer signature on the message, while leave the proof of the validity of it to the verification part. Compared with [20, 21], our protocol has longer partial signature, but smaller standard signature. In addition, the security of our protocol relies on relatively more standard assumptions, while the protocol in [20, 21] relies on newly proposed assumptions.

	interact?	**Pk**	**Apk**	**PSig**	**Sig**	**PK Model**	**Asmp**	**ROM**
[15]	no	$1\mathbb{G}$	$1\mathbb{G}$	$2\mathbb{G} + 8\mathbb{Z}_p$	$2\mathbb{G} + 12\mathbb{Z}_p$	registered	DDH	yes
[22]	no	$1\mathbb{G}$	$10\mathbb{G}$	$45\mathbb{G} + 1\mathbb{Z}_p$	$46\mathbb{G} + 1\mathbb{Z}_p$	chosen	SDH+DLIN	no
[20, 21]	no	$2\mathbb{G}$	$1\mathbb{G}$	$3\mathbb{G} + 4\mathbb{Z}_p$	$3\mathbb{G}$	registered	HSDH+DHSDH	no
[20, 21]	yes	$163\mathbb{G}$	$1\mathbb{G}$	$3\mathbb{G}$	$3\mathbb{G}$	registered	HSDH+DHSDH	yes
Ours	yes	$2\mathbb{G}$	$4\mathbb{G}$	$5\mathbb{G} + 1\mathbb{Z}_p$	$1\mathbb{G} + 1\mathbb{Z}_p$	registered	SDH+DLIN	no

Table 2. Comparison with existing AOFE protocols

8 Conclusion

In this paper we showed how to build an interactive ambiguous optimistic fair exchange protocol using a designated confirmer signature scheme with slight modifications. The resulting protocol is almost as efficient as the underlying DCS scheme. It makes use of standard security properties of the underlying DCS, and is secure without random oracles. We also proposed a concrete and efficient construction of designated confirmer signature, which is secure based on Strong Diffie-Hellman assumption and decision linear assumption without random oracles, and to the best of our knowledge has the shortest standard signature.

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: CCS. pp. 7–17. ACM (1997)
2. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures (extended abstract). In: EUROCRYPT98. LNCS, vol. 1403, pp. 591–606. Springer (1998)
3. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communication 18(4), 593–610 (2000)
4. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: FOCS04. pp. 186–195. IEEE Computer Society (2004)
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS. pp. 62–73. ACM (1993)
6. Boneh, D., Boyen, X.: Short signatures without random oracles. In: EUROCRYPT04. LNCS, vol. 3027, pp. 56–73. Springer (2004)
7. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO04. LNCS, vol. 3152, pp. 41–55. Springer (2004)
8. Camenisch, J., Michels, M.: Confirmer signature schemes secure against adaptive adversaries (extended abstract). In: EUROCRYPT00. LNCS, vol. 1807, pp. 243–258. Springer (2000)
9. Chaum, D.: Zero-knowledge undeniable signatures. In: EUROCRYPT90. LNCS, vol. 473, pp. 458–464. Springer (1990)
10. Chaum, D.: Designated confirmer signatures. In: EUROCRYPT94. LNCS, vol. 950, pp. 86–91. Springer (1995)
11. Chen, L.: Efficient fair exchange with verifiable confirmation of signatures. In: ASIACRYPT98. LNCS, vol. 1514, pp. 286–299. Springer (1998)
12. Damgård, I.: On Σ-protocols. Course on Cryptologic Protocol Theory, Aarhus University (2009), `http://www.daimi.au.dk/~ivan/Sigma.pdf`
13. Dodis, Y., Lee, P.J., Yum, D.H.: Optimistic fair exchange in a multi-user setting. In: Okamoto, T., Wang, X. (eds.) PKC07. LNCS, vol. 4450, pp. 118–133. Springer (2007), also at Cryptology ePrint Archive, Report 2007/182
14. Dodis, Y., Reyzin, L.: Breaking and repairing optimistic fair exchange from PODC 2003. In: DRM03. pp. 47–54. ACM (2003)
15. Garay, J.A., Jakobsson, M., MacKenzie, P.: Abuse-free optimistic contract signing. In: CRYPTO99. LNCS, vol. 1666, pp. 449–466. Springer (1999)
16. Gentry, C., Molnar, D., Ramzan, Z.: Efficient designated confirmer signatures without random oracles or general zero-knowledge proofs (extended abstract). In: ASIACRYPT05. LNCS, vol. 3788, pp. 662–681. Springer (2005)
17. Goldwasser, S., Waisbard, E.: Transformation of digital signature schemes into designated confirmer signature schemes. In: TCC04. LNCS, vol. 2951, pp. 77–100. Springer (2004)
18. Groth, J.: Fully anonymous group signatures without random oracles. In: Kurosawa, K. (ed.) ASIACRYPT07. LNCS, vol. 4833, pp. 164–180. Springer (2007), also at Cryptology ePrint Archive, Report 2007/186
19. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N. (ed.) EUROCRYPT08. LNCS, vol. 4965, pp. 415–432. Springer (2008)
20. Huang, Q., Wong, D.S., Susilo, W.: A new construction of designated confirmer signature and its application to optimistic fair exchange - (extended abstract). In: Pairing10. LNCS, vol. 6487, pp. 41–61. Springer (2010)

21. Huang, Q., Wong, D.S., Susilo, W.: Efficient designated confirmer signature and DCS-based ambiguous optimistic fair exchange. IEEE Transactions on Information Forensics and Security 6(4), 1233–1247 (2011)
22. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Ambiguous optimistic fair exchange. In: ASIACRYPT08. LNCS, vol. 5350, pp. 74–89. Springer (2008)
23. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles. In: CT-RSA08. LNCS, vol. 4964, pp. 106–120. Springer (2008)
24. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: EUROCRYPT96. LNCS, vol. 1070, pp. 143 – 154. Springer (1996)
25. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: EUROCRYPT06. LNCS, vol. 4004, pp. 465–485. Springer (2006)
26. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT04. LNCS, vol. 3027, pp. 74–90. Springer (May 2004)
27. Michels, M., Stadler, M.: Generic constructions for secure and efficient confirmer signature schemes. In: EUROCRYPT98. LNCS, vol. 1403, pp. 406–421. Springer (1998)
28. Okamoto, T.: Designated confirmer signatures and public key encryption are equivalent. In: CRYPTO94. LNCS, vol. 839, pp. 61–74. Springer (1994)
29. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: EUROCRYPT99. LNCS, vol. 1592, pp. 223–238. Springer (1999)
30. Park, J.M., Chong, E.K., Siegel, H.J.: Constructing fair-exchange protocols for e-commerce via distributed computation of RSA signatures. In: PODC03. pp. 172–181. ACM (2003)
31. Wang, G., Baek, J., Wong, D.S., Bao, F.: On the generic and efficient constructions of secure designated confirmer signatures. In: PKC07. LNCS, vol. 4450, pp. 43–60. Springer (2007)
32. Wang, G., Xia, F.: A pairing based designated confirmer signature scheme with unified verification. Technical report, School of Computer Science, University of Birmingham (Dec 2009)
33. Wikström, D.: Designated confirmer signatures revisited. In: TCC07. LNCS, vol. 4392, pp. 342–361. Springer (2007)
34. Zhang, F., Chen, X., Wei, B.: Efficient designated confirmer signature from bilinear pairings. In: ASIACCS08. pp. 363–368. ACM (2008)