Generic Construction of Privacy-Preserving Optimistic Fair Exchange Protocols

Qingwen Guo, Yuzhao Cui, Xiaomeng Zou, and Qiong Huang*
South China Agricultural University, Guangzhou 510642, China
{guoqingwen, yuzhaoc, zzxiaomengj}@stu.scau.edu.cn, qhuang@scau.edu.cn

Abstract

Privacy-preserving optimistic fair exchange ($\mathsf{P}^2\mathsf{OFE}$) is a kind of protocols which aim to solve the fairness problem in the exchange of digital signatures between two parties and in the meanwhile protect their privacy. In $\mathsf{P}^2\mathsf{OFE}$, no one else including the semi-trusted third party in charge of arbitration can confirm an exchange even after resolving a dispute. In this paper we present a black-box construction of $\mathsf{P}^2\mathsf{OFE}$ based on a tag-based public key encryption scheme and a standard digital signature scheme. Our construction follows the 'sign-then-encrypt' paradigm, and is secure in the standard model. Our construction is generic and admits more instantiations of $\mathsf{P}^2\mathsf{OFE}$.

Keywords: fair exchange, digital signature, privacy preserving, tag-based encryption

1 Introduction

Consider the scenario in which Alice and Bob want to sign a contract online, as shown in Figure 1. Since they are not face-to-face and do not trust each other, neither of them wants to move first to send their signature to the counterpart. It has been studied for decades in cryptography community that how Alice and Bob exchange their signatures so that either both of them has their counterpart's signature on the contract or none of them does. This is referred to as *fair exchange*. Asokan et al. introduced the notion of *optimistic fair exchange* (OFE) [1, 2] to solve the problem of fair exchange. In this notion, Alice and Bob semi-trust a third party, which is called the *arbitrator* and does nothing unless there is a dispute between Alice and Bob. To exchange signatures, Alice and Bob moves in three steps. First, Alice prepares an encapsulated version of her signature, called *partial* signature and denoted by σ_A, and sends it to Bob. To respond, Bob checks the validity of σ_A, computes and returns his own *full* signature ζ_B if it is valid. After confirming the validity of ζ_B, Alice gives her full signature ζ_A to Bob. If Bob accepts ζ_A, the exchange ends. Figure 2 shows a normal run of OFE. In case Bob does not receive Alice's full signature due to various reasons, Bob can ask the arbitrator to converting Alice's partial signature into a full version. This process is known as the *resolution*. Figure 3 shows a run of the resolution process.

Figure 1: Contract Signing

*Corresponding Author: 483 Wushan Road, Tianhe District, Guangzhou 510642, P.R. China, Email: qhuang@scau.edu.cn, Tel: +86-20-85285393

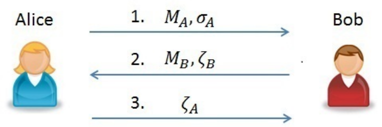


Figure 2: A Run of OFE

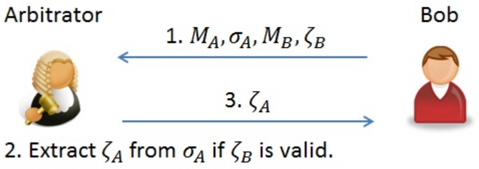

Arbitrator
Bob
1. $M_A, \sigma_A, M_B, \zeta_B$
3. ζ_A
2. Extract ζ_A from σ_A if ζ_B is valid.

Figure 3: Resolution Process of OFE

In Asiacrypt 2008 [14], Huang et al. pointed a fairness problem in OFE. That is, Alice's partial signature σ_A reveals her intention to do exchange with Bob. They proposed the notion of *ambiguous* OFE (AOFE), in which Bob could simulate Alice' partial signatures. Given a partial signature, no can would believe that the signature was generated by Alice, since it may also be generated by Bob. They then proposed a construction of AOFE [14, 16], which is secure in the standard model.

Since the introduction, there have been a bunch of works on AOFE, such as [7, 8, 9, 18, 17, 15, 10, 11]. For example, Wang *et al.* [20] pointed out that AOFE also leaks the information of the signer and the verifier's identities. Thus, they proposed the notion of *Perfect* AOFE (PAOFE). The new notion requires that a partial signature should not leak any information about the identities of the two exchanging parties. Furthermore, even the arbitrator can distinguish from a run of PAOFE what exchange has taken place.

Recently, based on [20], Huang et al. [11, 12] further studied to protect the privacy of involving parties, and proposed the notion of *privacy-preserving* OFE ($\mathsf{P^2OFE}$ in short), in which anyone other than Alice and Bob can confirm the exchange even after the resolution. This notion particularly applies to the scenario where two parties sign a sensitive contract and both of them do not want anyone to confirm their participant in the contract from the protocol transcript. The basic idea is to split the resolution process into two algorithms, Res^A and Res^V. Briefly speaking, to resolve a partial signature σ of Alice, the arbitrator runs Res^A to convert it to an intermediate value θ which is given to Bob. Bob runs Res^V to convert θ to a full signature ζ of Alice. Huang et al. [11, 12] proposed a concrete and efficient construction of $\mathsf{P^2OFE}$, which is secure based on the Strong Diffie-Hellman assumption [4] and Decision Linear assumption [5] in the standard model.

(**Our Contribution**.) To admit more constructions, in this paper we propose a generic construction of $\mathsf{P^2OFE}$ protocol. Our construction makes use of a tag-based public key encryption [19], a standard signature and a one-time signature, and follows (a slight variant of) the sign-then-encrypt paradigm. Roughly speaking, to produce a partial signature, Alice first generates a standard signature on the message, and splits it into two halves. Then Alice uses the tag-based public key encryption scheme to encrypt each half, with a freshly generated one-time verification key as the tag. Finally, it uses the one-time signature to authenticate the two ciphertexts. We show that this generic construction is secure under the security model given in [11, 12] without relying random oracle heuristic as well, assuming the underlying basic tools are secure.

(**Paper Organization**.) In the next section we give the definition of $\mathsf{P^2OFE}$ and its security model. In

Sec. 3 we introduce the basic knowledge needed for the generic construction. Then in Sec. 4 we propose the generic construction, and prove its security in Sec. 5. Finally, we conclude the paper in Sec. 6.

2 Privacy-Preserving OFE

2.1 Definition

Below we describe the definition of $\mathsf{P^2OFE}$, which is taken from [11].

Definition 2.1. *A Privacy-Preserving Optimistic Fair Exchange protocol (*$\mathsf{P^2OFE}$*) involves signers, verifiers and an arbitrator, and consists of the following probabilistic polynomial-time (PPT) algorithms/protocols:*

PMG. *Taking the security parameter* 1^k *as input, the algorithm outputs the system parameter* PM.

Akg. *Taking as input* PM*, the algorithm outputs a key pair for the arbitrator* $(\mathtt{Apk}, \mathtt{Ask})$.

UKg. *Taking as input* PM *and the arbitrator's public key* $\mathtt{Apk}$*, the algorithm outputs a user key pair* $(\mathtt{Pk}, \mathtt{Sk})$.

PSig. *Taking as input a message M, the signer's secret key* $\mathtt{Sk}_i$ *and public key* $\mathtt{Pk}_i$*, the verifier's public key* $\mathtt{Pk}_j$ *and the arbitrator's public key* $\mathtt{Apk}$*, the algorithm outputs a partial signature* σ.

PVer. *The signer with secret key* $\mathtt{Sk}_i$ *and the verifier with secret key* $\mathtt{Sk}_j$ *run the protocol on common input* $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$ *to verify a partial signature* σ*. The verifier's output is a decision bit, which is 1 for acceptance and 0 for rejection.*

Sig. *Taking as input* $(M, \mathtt{Sk}_i, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$*, the algorithm outputs a full signature* ζ.

Ver. *Taking as input* $(M, \zeta, \mathtt{Pk}_i, \mathtt{Pk}_j, \mathtt{Apk})$*, the algorithm outputs a decision bit b, which is 1 for acceptance and 0 for rejection.*

Res. *The resolution protocol is splitted into two algorithms,* Res^A *and* Res^V*. The arbitrator runs the former to convert a partial signature* σ *to an intermediate value* θ*. The verifier runs the latter to convert* θ *to a full signature* ζ*.*

2.2 Security Model

Briefly speaking, a secure $\mathsf{P^2OFE}$ protocol should satisfy *resolution ambiguity*, *signer ambiguity*, *perfect ambiguity*, *security against signers* and *security against the arbitrator* [11, 12]. Definitions of these properties given below are from [11, 12]. All the properties are defined in the certified-key model [3, 10], in which every public key should be certified before being used. It is modeled as an oracle O_{KR}. The adversary queries O_{KR} by submitting a key pair $(\mathtt{Pk}, \mathtt{Sk})$, and is returned 1 if $(\mathtt{Pk}, \mathtt{Sk})$ is a valid key pair, and 0 otherwise. Below we omit O_{KR} in the experiments and omit the generation of system parameters PM, for the sake of simplicity.

Resolution Ambiguity. It requires that the full signatures output by the Sig algorithm should be indistinguishable from those output by the the resolution protocol Res.

Signer Ambiguity. It requires that the verifier could produce partial signatures which look similar with those produced by the signer. Formally, let FPSig be a PPT algorithm that is run by the verifier to simulate the signer's partial signatures. For any PPT signer $\mathscr{A}$, it succeeds with at most negligible advantage in the following experiment $\mathbf{Exp}_{\mathsf{sa}}$:

$$
\begin{aligned}
(\mathtt{Apk},\mathtt{Ask}) &\leftarrow \mathsf{Akg}(\mathsf{PM})\\
(\mathtt{Pk}_\gamma,\mathtt{Sk}_\gamma) &\leftarrow \mathsf{Ukg}(\mathsf{PM},\mathtt{Apk}),\ \forall\gamma\in\{S,V\}\\
(M^*,\Upsilon) &\leftarrow \mathscr{A}^{O_{\mathsf{Res}^A}}(\{(\mathtt{Pk}_\gamma,\mathtt{Sk}_\gamma)\}_{\gamma\in\{S,V\}},\mathtt{Apk})\\
b &\leftarrow \{0,1\}\\
\sigma^* &\leftarrow \begin{cases} \mathsf{PSig}(M^*,\mathtt{Sk}_S,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk}) & \text{if } b=0\\ \mathsf{FPSig}(M^*,\mathtt{Sk}_V,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk}) & \text{if } b=1\end{cases}\\
b' &\leftarrow \mathscr{A}^{O_{\mathsf{Res}^A}}(\Upsilon,\sigma^*)\\
\text{Succ. of } \mathscr{A} &:= [b'=b\wedge(M^*,\sigma^*,\mathtt{Pk}_S,\mathtt{Pk}_V)\notin\mathscr{Q}(\mathscr{A},O_{\mathsf{Res}^A})\\
&\qquad \wedge(M^*,\sigma^*,\mathtt{Pk}_V,\mathtt{Pk}_S)\notin\mathscr{Q}(\mathscr{A},O_{\mathsf{Res}^A})],
\end{aligned}
$$

where O_{Res^A} takes as input $(M,\sigma,\mathtt{Pk}_i,\mathtt{Pk}_j)$ and outputs an intermediate value θ, and $\mathscr{Q}(\mathscr{A},O_{\mathsf{Res}^A})$ contains all the queries issued by $\mathscr{A}$ to O_{Res^A}. The advantage of $\mathscr{A}$ in the experiment above is defined as $\mathtt{Adv}_{\mathsf{sa}}^{\mathscr{A}}(1^k) := |\Pr[Succ_{\mathsf{sa}}]-1/2|$, where $Succ_{\mathsf{sa}}$ denotes the event that $\mathscr{A}$ succeeds in $\mathbf{Exp}_{\mathsf{sa}}$.

Definition 2.2 (Signer Ambiguity). *A* $\mathsf{P^2OFE}$ *protocol is* signer ambiguous *if there is no PPT $\mathscr{A}$, such that $\mathtt{Adv}_{\mathsf{sa}}^{\mathscr{A}}(1^k)$ is non-negligible.*

Perfect Ambiguity. It requires that except the signer and the verifier, no one could distinguish whether the given signature was generated honestly by signer S w.r.t. the verifier V, or randomly selected from the signature space. Let Sim be a PPT algorithm run by anyone to simulate signatures w.r.t. S and V. Formally, for any PPT adversary $\mathscr{A}$, it succeeds in the following experiment $\mathbf{Exp}_{\mathsf{pa}}$ with only negligible advantage:

$$
\begin{aligned}
(\mathtt{Apk},\mathtt{Ask}) &\leftarrow \mathsf{Akg}(\mathsf{PM})\\
(\mathtt{Pk}_\gamma,\mathtt{Sk}_\gamma) &\leftarrow \mathsf{Ukg}(\mathsf{PM},\mathtt{Apk}),\ \forall\gamma\in\{S,V\}\\
(M^*,\Upsilon) &\leftarrow \mathscr{A}^{O_{\mathsf{PSig}^V},O_{\mathsf{FPSig}},O_{\mathsf{Res}^V}}(\mathtt{Pk}_S,\mathtt{Sk}_S,\mathtt{Pk}_V,\mathtt{Apk},\mathtt{Ask})\\
b &\leftarrow \{0,1\}\\
\sigma^* &\leftarrow \begin{cases} \mathsf{PSig}(M^*,\mathtt{Sk}_S,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk}), & \text{if } b=0\\ \mathsf{Sim}(\mathtt{Apk},\mathtt{Pk}_S,\mathtt{Pk}_V) & \text{, if } b=1\end{cases}\\
b' &\leftarrow \mathscr{A}^{O_{\mathsf{PSig}^V},O_{\mathsf{FPSig}},O_{\mathsf{Res}^V}}(\Upsilon,\sigma^*)\\
\theta^* &\leftarrow \mathsf{Res}^A(M^*,\mathtt{Ask},\sigma^*,\mathtt{Pk}_S,\mathtt{Pk}_V)\\
\text{Succ. of } \mathscr{A} &:= [b'=b\wedge(M^*,\theta^*,\mathtt{Pk}_S)\notin\mathscr{Q}(\mathscr{A},O_{\mathsf{Res}^V})],
\end{aligned}
$$

where O_{PSig^V} takes as input $(M,\mathtt{Pk}')$, and outputs a partial signature σ; O_{FPSig} takes as input $(M,\mathtt{Pk}')$, and outputs a simulated partial signature; O_{Res^V} takes as input $(M,\theta,\mathtt{Pk}')$, and outputs the full signature ζ; and $\mathscr{Q}(\mathscr{A},O_{\mathsf{Res}^V})$ contains all the queries issued by $\mathscr{A}$ to O_{Res^V}. The advantage of $\mathscr{A}$ in the experiment above is defined as $\mathtt{Adv}_{\mathsf{pa}}^{\mathscr{A}}(1^k) := |\Pr[Succ_{\mathsf{pa}}]-1/2|$, where $Succ_{\mathsf{pa}}$ denotes the event that $\mathscr{A}$ succeeds in the experiment $\mathbf{Exp}_{\mathsf{pa}}$.

Definition 2.3 (Perfect Ambiguity). *A* $\mathsf{P^2OFE}$ *protocol is* perfect ambiguous *if there is no PPT adversary $\mathscr{A}$ such that $\mathtt{Adv}_{\mathsf{pa}}^{\mathscr{A}}(1^k)$ is non-negligible.*

Security against Signers. it requires that the signer could not produce a partial signature which can pass the partial verification, but cannot be resolved to a full one. Formally, we consider the following experiment $\mathbf{Exp}_{\mathsf{sas}}$:

$$\begin{aligned}
(\mathtt{Apk},\mathtt{Ask}) &\leftarrow \mathsf{Akg}(\mathsf{PM})\\
(\mathtt{Pk}_V,\mathtt{Sk}_V) &\leftarrow \mathsf{Ukg}(\mathsf{PM},\mathtt{Apk})\\
(M^*,\mathtt{Pk}_S,\sigma^*) &\leftarrow \mathscr{A}^{O_{\mathsf{FPSig}},O_{\mathsf{Res}}}(\mathtt{Pk}_V,\mathtt{Apk})\\
\theta^* &\leftarrow \mathsf{Res}^A(M^*,\mathtt{Ask},\sigma^*,\mathtt{Pk}_S,\mathtt{Pk}_V)\\
\zeta^* &\leftarrow \mathsf{Res}^V(M^*,\mathtt{Sk}_V,\theta^*,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk})\\
\text{Succ. of } \mathscr{A} &:= [\mathsf{PVer}(M^*,\sigma^*,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk},\mathtt{Sk}_V)=1\\
&\quad\wedge \mathsf{Ver}(M^*,\zeta^*,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk})=0\\
&\quad\wedge (M^*,\mathtt{Pk}_S)\notin \mathscr{Q}(\mathscr{A},O_{\mathsf{FPSig}})],
\end{aligned}$$

where $O_{\mathsf{Res}} = \langle O_{\mathsf{Res}^A}, O_{\mathsf{Res}^V}\rangle$ takes as $(M,\sigma,\mathtt{Pk}')$, and outputs the corresponding full signature ζ or $\perp$; and $\mathscr{Q}(\mathscr{A},O_{\mathsf{FPSig}})$ contains all the queries issued by $\mathscr{A}$ to O_{FPSig}. The advantage of $\mathscr{A}$ in the experiment above is defined as $\mathtt{Adv}_{\mathsf{sas}}^{\mathscr{A}}(1^k) := \Pr[Succ_{\mathsf{sas}}]$, where $Succ_{\mathsf{sas}}$ denotes the event that $\mathscr{A}$ succeeds in the experiment $\mathbf{Exp}_{\mathsf{sas}}$.

Definition 2.4 (Security against Signers)**.** *A* $\mathsf{P}^2\mathsf{OFE}$ *protocol is* secure against signers *if there is no PPT adversary* $\mathscr{A}$ *such that* $\mathtt{Adv}_{\mathsf{pa}}^{\mathscr{A}}(1^k)$ *is non-negligible.*

Security against the Arbitrator. It requires that no one could produce valid signatures on behalf of the signer. Formally, we consider the following experiment $\mathbf{Exp}_{\mathsf{saa}}$:

$$\begin{aligned}
(\mathtt{Apk},\mathtt{Ask}) &\leftarrow \mathsf{Akg}(\mathsf{PM})\\
(\mathtt{Pk}_S,\mathtt{Sk}_S) &\leftarrow \mathsf{Ukg}(\mathsf{PM},\mathtt{Apk})\\
(M^*,\mathtt{Pk}_V,\zeta^*) &\leftarrow \mathscr{A}^{O_{\mathsf{PSig}}}(\mathtt{Pk}_S,\mathtt{Apk},\mathtt{Ask})\\
\text{Succ. of } \mathscr{A} &:= [\mathsf{Ver}(M^*,\zeta^*,\mathtt{Pk}_S,\mathtt{Pk}_V,\mathtt{Apk})=1\\
&\quad\wedge (M^*,\mathtt{Pk}_1)\notin \mathscr{Q}(\mathscr{A},O_{\mathsf{PSig}})],
\end{aligned}$$

where O_{PSig} takes as input a message M and a public key $\mathtt{Pk}'$ and outputs a partial signature σ; and $\mathscr{Q}(\mathscr{A},O_{\mathsf{PSig}})$ contains all the queries issued by $\mathscr{A}$ to O_{PSig}. The advantage of $\mathscr{A}$ in the experiment above is defined as $\mathtt{Adv}_{\mathsf{saa}}^{\mathscr{A}}(1^k) := \Pr[Succ_{\mathsf{saa}}]$, where $Succ_{\mathsf{saa}}$ denotes the event that $\mathscr{A}$ succeeds in $\mathbf{Exp}_{\mathsf{saa}}$.

Definition 2.5 (Security against the Arbitrator)**.** *A* $\mathsf{P}^2\mathsf{OFE}$ *protocol is* secure against the arbitrator *if there is no PPT adversary* $\mathscr{A}$ *such that* $\mathtt{Adv}_{\mathsf{saa}}^{\mathscr{A}}(1^k)$ *is non-negligible.*

3 Preliminaries

3.1 Tag-based Public Key Encryption

A tag-based public key encryption [19] consists of the following probabilistic polynomial-time algorithms.

Kg. On input a security parameter 1^k, it outputs a public/private key pair $(\mathtt{Pk},\mathtt{Sk}) \leftarrow \mathsf{Kg}(1^k)$.

Enc. On input a message M, a tag $t \in \{0,1\}^*$, and a public key $\mathtt{Pk}$, it outputs a ciphertext $C \leftarrow \mathsf{Enc}(M,t,\mathtt{Pk})$.

Dec. On input a ciphertext C, a tag $t \in \{0,1\}^*$ and a secret key $\mathtt{Sk}$, it outputs a message $M \leftarrow \mathsf{Dec}(C,t,\mathtt{Sk})$. Notice that M could be a failure symbol if C is not well-formed.

The security of tag-based public key encryption is defined via the following game.

Initialize Phase. The adversary $\mathscr{A}$ prepares a tag t^*, and submits it to the challenger C.

Setup Phase. The challenge C prepares a public key $\mathtt{Pk}$ and gives it to $\mathscr{A}$.

Query Phase 1. The adversary issues polynomially many decryption queries. It submits a tag t and a ciphertext C to the challenger, which then returns the corresponding decryption result.

Challenge Phase. $\mathscr{A}$ prepares two equal-length messages M_0, M_1 and submits them to C, which then choose a random bit b and computes $C^* \leftarrow \mathsf{Enc}(M_b, t^*, \mathtt{Pk})$ and returns C^* to $\mathscr{A}$.

Query Phase 2. The adversary continues to issuing decryption queries as in Query Phase 1, which the only restriction that its decryption query (t, C) should be different from (t^*, C^*).

Guess Phase. Finally, the adversary outputs a bit b' and wins the game if $b' = b$.

Denote the winning probability of $\mathscr{A}$ in the game above by $\Pr[b' = b]$.

Definition 3.1 (Selective-tag CCA Security)**.** *A tag-based public key encryption scheme is said to be* selective-tag CCA-secure *if for any PPT adversary* $\mathscr{A}$*,*

$$|\Pr[b' = b] - 1/2| \leq \operatorname{negl}(1^k),$$

where $\operatorname{negl}(1^k)$ *is a negligible function.*

3.2 Signature

A digital signature scheme consists of the following probabilistic polynomial-time algorithms [13, 6].

Kg. On input a security parameter 1^k, it outputs a public/private key pair $(\mathtt{Pk}, \mathtt{Sk}) \leftarrow \mathsf{Kg}(1^k)$.

Sign. On input a message M and a secret key $\mathtt{Sk}$, it outputs a signature $\sigma \leftarrow \mathsf{Sig}(M, \mathtt{Sk})$.

Ver. On input a message M, a signature σ and a public key $\mathtt{Pk}$, it outputs a bit $b \leftarrow \mathsf{Ver}(M, \sigma, \mathtt{Pk})$, which is 1 if σ is a valid signature on M under $\mathtt{Pk}$, and 0 otherwise.

For security of signature schemes, we consider the following game.

Setup Phase. The challenger C prepares a public key $\mathtt{Pk}$ and gives it to the adversary $\mathscr{A}$.

Query Phase. The adversary issues polynomially many signing queries. It submits a message M to C, which then computes a signature σ on M under $\mathtt{Pk}$, and returns σ to $\mathscr{A}$.

Forge Phase. Finally, the adversary outputs a message M^* and a forged signature σ^*. It wins the game if $\mathsf{Ver}(M^*, \sigma^*, \mathtt{Pk}) = 1$ and it did not submit M^* to C for a signature in the Query Phase.

Definition 3.2 (EUF-CMA Security)**.** *A signature scheme is said to be* existentially unforgeable under chosen-message attacks *(EUF-CMA secure) if for any PPT adversary* $\mathscr{A}$*, its winning probability in the game above is negligible.*

One-time signature is a signature scheme in which the secret key $\mathtt{Sk}$ can only be used to sign one message at most [13, 6]. We consider the following game.

Setup Phase. The challenger C prepares a public key $\mathtt{Pk}$ and gives it to the adversary $\mathscr{A}$.

Query Phase. The adversary submits a message M to C, which then computes a signature σ on M under $\mathtt{Pk}$, and returns σ to $\mathscr{A}$.

Forge Phase. Finally, the adversary outputs a message M^* and a forged signature σ^*. It wins the game if $\mathsf{Ver}(M^*, \sigma^*, \mathtt{Pk}) = 1$ and $(M^*, \sigma^*) \neq (M, \sigma^*)$.

Definition 3.3 (Strong One-Time Signature)**.** *A signature scheme is said to be a* strong one-time signature *if for any PPT adversary $\mathscr{A}$, its winning probability in the game above is negligible.*

4 A Generic Construction of $\mathsf{P^2OFE}$

To capture more constructions of $\mathsf{P^2OFE}$, a generic construction is necessary. Below we are going to propose a generic construction of $\mathsf{P^2OFE}$.

Let $\mathscr{E} = (\mathsf{Kg}, \mathsf{Enc}, \mathsf{Dec})$ be a tag-based public key encryption schemes with randomness space $\mathscr{R}$ and tag space $\mathscr{T}$, and $\mathscr{S} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a standard signature scheme. Let $\hat{\mathscr{S}} = (\mathsf{Kg}, \mathsf{Sig}, \mathsf{Ver})$ be a strong one-time signature scheme. The construction works as below.

Akg. The arbitrators generates its key pair by invoking $(\mathtt{Apk}, \mathtt{Ask}) \leftarrow \mathscr{E}.\mathsf{Kg}(1^k)$.

Ukg. Each user runs $(\mathtt{Pk}^E, \mathtt{Sk}^E) \leftarrow \mathscr{E}.\mathsf{Kg}(1^k)$ and $(\mathtt{Pk}^S, \mathtt{Sk}^S) \leftarrow \mathscr{S}.\mathsf{Kg}(1^k)$, and sets its key pair to be $(\mathtt{Pk}, \mathtt{Sk}) := ((\mathtt{Pk}^E, \mathtt{Pk}^S), (\mathtt{Sk}^E, \mathtt{Sk}^S))$.

PSig. Given a message M to partially sign with respect to the verifier V with public key $\mathtt{Pk}_V = (\mathtt{Pk}^E_V, \mathtt{Pk}^S_V)$, the signer S (with public key $\mathtt{Pk}_S = (\mathtt{Pk}^E_S, \mathtt{Pk}^S_S)$ and secret key $\mathtt{Sk}_S = (\mathtt{Sk}^E_S, \mathtt{Sk}^S_S)$) does as follows.

1. Run $\mathscr{S}.\mathsf{Sig}(\mathtt{Sk}^S_S, M)$ to get the signature ζ.
2. Randomly split ζ into two halves ζ' and ζ'' so that $\zeta = \zeta' \odot \zeta''$ for some operation $\odot$ defined over the group that ζ falls in.
3. Run $\hat{\mathscr{S}}.\mathsf{Kg}(1^k)$ to get a one-time key pair $(\hat{vk}, \hat{sk})$.
4. Randomly picks $r', r'' \leftarrow \mathscr{R}$ and computes

$$\begin{aligned} C' &\leftarrow \mathscr{E}.\mathsf{Enc}(\mathtt{Apk}, \hat{vk}, \zeta'; r'), \\ C'' &\leftarrow \mathscr{E}.\mathsf{Enc}(\mathtt{Pk}^E_V, \hat{vk}, \zeta''; r''), \\ \delta &\leftarrow \hat{\mathscr{S}}.\mathsf{Sig}(\hat{sk}, C' \| C''). \end{aligned}$$

Its partial signature on M is set to be $\sigma := (C', C'', \hat{vk}, \delta)$. The signer sends σ to V.

PVer. Given a partial signature $\sigma = (C', C'', \hat{vk}, \delta)$, both S and V check the validity of the two ciphertexts if ciphertexts of the encryption schemes are public verifiable and that of the one-time signature, i.e. whether the following equation holds:

$$\hat{\mathscr{S}}.\mathsf{Ver}(\hat{vk}, C' \| C'', \delta) = 1. \tag{1}$$

V outputs 0 if either fails. Otherwise, S and V proceed to carry out the following witness-indistinguishable proof, in which S acts as the prover:

$$\begin{aligned} \pi := \mathrm{PoK}\big\{(\zeta', \zeta'', r', r'') | C' &\leftarrow \mathscr{E}.\mathsf{Enc}(\mathtt{Apk}, \hat{vk}, \zeta'; r') \\ \wedge C'' &\leftarrow \mathscr{E}.\mathsf{Enc}(\mathtt{Pk}^E_V, \hat{vk}, \zeta'', r'') \\ &\wedge (\mathscr{S}.\mathsf{Ver}(\mathtt{Pk}^S_S, M, \zeta' \odot \zeta'') = 1 \\ &\vee \mathscr{S}.\mathsf{Ver}(\mathtt{Pk}^S_V, M, \zeta' \odot \zeta'') = 1)\big\}. \end{aligned} \tag{2}$$

V outputs 1 if it accepts the proof, and 0 otherwise.

Sig. Given a message M, the signer S runs $\zeta \leftarrow \mathscr{S}.\mathsf{Sig}(\mathtt{Sk}_S^S, M)$, and returns ζ as its full signature on M.

Ver. Given a full signature ζ on message M from the signer S, the verifier V computes $b \leftarrow \mathscr{S}.\mathsf{Ver}(\mathtt{Pk}_S^S, M, \zeta)$, and outputs the bit b.

Res^A. Given a partial signature $\sigma = (C', C'', \hat{vk}, \delta)$ from the verifier V, the arbitrator performs the validity check on σ as V does in **PVer** algorithm, and returns $\perp$ if any check fails. Otherwise, it runs $\mathscr{E}.\mathsf{Dec}(\mathtt{Ask}, \hat{vk}, C')$ to get ζ'. If $\zeta' = \perp$, the arbitrator returns $\perp$; otherwise, it returns $\theta := (\zeta', C'', \hat{vk})$ to V.

Res^V. Given an intermediate signature $\theta = (\zeta', C'', \hat{vk})$, the verifier V computes $\zeta'' \leftarrow \mathscr{E}.\mathsf{Dec}(\mathtt{Sk}_V^E, \hat{vk}, C'')$ and $\zeta \leftarrow \zeta' \odot \zeta''$. It outputs ζ if $\mathscr{S}.\mathsf{Ver}(\mathtt{Pk}_S^S, M, \zeta)$ returns 1, and $\perp$ otherwise.

Correctness can be verified naturally and we omit it here.

Remark. By slightly modifying our generic construction of $\mathsf{P^2OFE}$ as below, we get a construction of perfect ambiguous optimistic fair exchange protocol.

1. Change the order of resolution. Namely, the verifier does the first level resolution by running Res^V on the partial signature, and submits the output to the arbitrator, who then runs Res^A to produce the full signature.

2. Replace the encryption scheme $\mathscr{E}$ (used by the verifier) with another one which is CCA-secure and has the property of *key privacy*.

It is not hard to show that after the modifications above, the resulting construction is a perfect ambiguous optimistic fair exchange protocol. We omit the proof here.

5 Security Analysis

Below we show the security of our generic construction.

Theorem 5.1. *Our generic construction of* $\mathsf{P^2OFE}$ *is secure if* $\mathscr{E}$ *is selective-tag CCA-secure,* $\mathscr{S}$ *is existentially unforgeable under chosen-message attacks, and the proof* π *is a witness indistinguishable proof.*

Lemma 5.2. *The generic construction is resolution ambiguous.*

The proof is obvious and we omit it here.

Lemma 5.3. *The generic construction is signer ambiguous if* $\mathscr{E}$ *is selective-tag CCA-secure,* $\hat{\mathscr{S}}$ *is one-time strongly unforgeable.*

The FPSig algorithm works as follows. The simulator (i.e. the verifier V) uses its own secret signing key $\mathtt{Sk}_V^S$ to generate the signature ζ and then follows the rest of PSig algorithm to produce the (simulated) partial signature σ.

Proof. Let $\mathscr{A}$ be an adversary that breaks the signer ambiguity with advantage ε. We construct another efficient algorithm $\mathscr{D}$ that breaks the chosen-ciphertext security of $\mathscr{E}$.

Algorithm $\mathscr{D}$ generates a fresh one-time key pair $(\hat{vk}^*, \hat{sk}^*)$, submits $\hat{vk}^*$ to its challenger, and is returned the challenge public key pk^*. It sets $\mathtt{Apk} := pk^*$, runs Ukg twice to generate the target signer's key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$ and the target verifier's key pair $(\mathtt{Pk}_V, \mathtt{Sk}_V)$. It then invokes the adversary on input $((\mathtt{Pk}_S, \mathtt{Sk}_S), (\mathtt{Pk}_V, \mathtt{Sk}_V), \mathtt{Apk})$.

After receiving a resolution query $(M, \sigma, \mathtt{Pk}_i, \mathtt{Pk}_j)$ from the adversary, where $\sigma = (C', C'', \hat{vk}, \delta)$, if $\hat{vk}$ contained in σ is equal to $\hat{vk}^*$, $\mathscr{D}$ simply returns $\perp$. Guaranteed by the strong one-time unforgeability of $\hat{\mathscr{S}}$, the probability that $\hat{vk} = \hat{vk}^*$ is negligible. It brings only a negligible effect to the adversary's advantage in the game. $\mathscr{D}$ then checks the validity of σ as in **PVer** algorithm and returns $\perp$ if σ is invalid; otherwise, it sends C' to its own decryption oracle, and is returned the 'plaintext' ζ'. If $\zeta' = \perp$, $\mathscr{D}$ returns $\perp$; otherwise, it returns $\theta = (\zeta', C'', \hat{vk})$ to the adversary.

At some time the adversary $\mathscr{A}$ submits a message M^* of its choice, $\mathscr{D}$ uses $\mathtt{Sk}_S$ and $\mathtt{Sk}_V$ to generate signatures ζ_0^* and ζ_1^*, respectively, and splits them into random shares $\zeta_0', \zeta_1', \zeta''$ so that $\zeta_0^* = \zeta_0' \odot \zeta''$ and $\zeta_1^* = \zeta_1' \odot \zeta''$. It then sends messages $m_0^* = \zeta_0'$ and $m_1^* = \zeta_1'$ to its own challenger, and is returned a ciphertext C^* which is an encryption of m_b^* with tag $\hat{vk}^*$ for an unknown bit b. It sets $(C')^* = C^*$ and computes $(C'')^* \leftarrow \mathscr{E}''.\mathsf{Enc}(\mathtt{Apk}, \hat{vk}^*, \zeta'')$ and $\delta^* \leftarrow \hat{\mathscr{S}}.\mathsf{Sig}(\hat{sk}^*, (C')^* \| (C'')^*)$. $\mathscr{D}$ sets $\sigma^* := ((C')^*, (C'')^*, \hat{vk}^*, \delta^*)$, and returns it to $\mathscr{A}$.

$\mathscr{D}$ then continues to answer the resolution queries submitted by $\mathscr{A}$ as before, except that it discards the queries $(M^*, \sigma^*, \mathtt{Pk}_S, \mathtt{Pk}_V)$ and $(M^*, \sigma^*, \mathtt{Pk}_V, \mathtt{Pk}_S)$. Finally, the adversary outputs a bit b' as its guess of whether σ^* is generated by the signer or simulated by the verifier. $\mathscr{D}$ outputs b' as its guess of b.

It is not hard to see that σ^* is correctly distributed and that if $b = 0$, σ^* is a real partial signature generated by the target signer, and otherwise it is a simulated partial signature generated by the verifier. If $\mathscr{A}$'s guess is correct, so is $\mathscr{D}$'s query. Therefore, $\mathscr{D}$'s advantage in breaking the selective-tag IND-CCA security of $\mathscr{E}'$ is negligibly close to ε. If ε is non-negligible, so is $\mathscr{D}$'s advantage. □

Lemma 5.4. *The generic construction is perfectly ambiguous if $\mathscr{E}$ is selective-tag IND-CCA secure and the proof π is witness indistinguishable.*

Before presenting the proof, we describe the public Sim algorithm. Given $(\mathtt{Apk}, \mathtt{Pk}_S, \mathtt{Pk}_V)$, the simulation algorithm randomly selects ζ', ζ'' from the corresponding domain, and follows the PSig algorithm to generate a simulated signature $\sigma = (C', C'', \hat{vk}, \delta)$.

Proof. Let $\mathscr{A}$ be an adversary that breaks the perfect ambiguity with advantage ε. We use it to construct another algorithm $\mathscr{D}$ to break the selective-tag IND-CCA security of the underlying encryption scheme $\mathscr{E}$.

Algorithm $\mathscr{D}$ generates a fresh one-time key pair $(\hat{vk}^*, \hat{sk}^*)$, submits $\hat{vk}^*$ to its challenger, and is returned the challenge public key pk^*. It runs **Akg** to generate the arbitrator's key pair $(\mathtt{Apk}, \mathtt{Ask})$, runs $\mathscr{S}.\mathsf{Kg}(1^k)$ to generate a signature key pair $(\mathtt{Pk}_V^S, \mathtt{Sk}_V^S)$ for V, and runs **Ukg** to generate the target signer's key pair $(\mathtt{Pk}_S, \mathtt{Sk}_S)$. It sets $\mathtt{Pk}_V^E := pk^*$ and invokes the adversary on input $(\mathtt{Pk}_S, \mathtt{Sk}_S, \mathtt{Pk}_V, \mathtt{Apk}, \mathtt{Ask})$. $\mathscr{D}$ then starts to simulate oracles for $\mathscr{A}$ as below.

- O_{PSig^V}: All queries to oracle O_{PSig^V} can be answered using the knowledge of $\mathtt{Sk}_V^S$ and following the PSig algorithm.

- O_{FPSig}: All queries to oracle O_{PSig^V} can be answered by $\mathscr{D}$ using the knowledge of $\mathtt{Sk}_V^S$ and following the FPSig algorithm (presented before the proof of Lemma 5.3). $\mathscr{D}$ then runs the proof π to show that the returned signature is valid with respect to $\mathtt{Pk}_S$ and $\mathtt{Pk}_V$. Guaranteed by the witness indistinguishability of π, the proof looks almost the same as that in **PVer** and brings only negligible difference to the adversary's advantage.

- O_{Res^V}: Given a query $(M, \theta, \mathtt{Pk}')$ where $\theta = (\zeta', C'', \hat{vk})$, $\mathscr{D}$ forwards $(C'', \hat{vk})$ to its own decryption oracle and obtains the decryption result ζ''. If $\zeta'' = \perp$, $\mathscr{D}$ returns $\perp$ to the adversary as well. Otherwise, it computes $\zeta = \zeta' \odot \zeta''$ and returns ζ to $\mathscr{A}$ if $\mathscr{S}.\mathsf{Ver}(\mathtt{Pk}', M, \zeta) = 1$, and $\perp$ otherwise.

At some time the adversary submits a message M^*. $\mathscr{D}$ runs $\zeta^* \leftarrow \mathscr{S}.\mathsf{Sig}(\mathtt{Sk}_S^S, M^*)$. It then selects random ζ', ζ_1'' and computes ζ_0'' so that $\zeta^* = \zeta' \odot \zeta_0''$. It generates a fresh one-time key pair $(\hat{vk}^*, \hat{sk}^*) \leftarrow \hat{\mathscr{S}}.\mathsf{Kg}(1^k)$, sends two messages $m_0 = \zeta_0''$ and $m_1 = \zeta_1''$ to its challenger and is returned the challenge ciphertext C^* (with respect to the tag $\hat{vk}^*$). $\mathscr{D}$ sets $(C'')^* := C^*$, computes

$$(C')^* \leftarrow \mathscr{E}.\mathsf{Enc}(\mathtt{Pk}_S^E, \hat{vk}^*, \zeta'),\ \ \delta^* \leftarrow \hat{\mathscr{S}}.\mathsf{Sig}(\hat{sk}^*, (C')^* \| (C'')^*),$$

and returns the challenge partial signature $\sigma^* := ((C')^*, (C'')^*, \hat{vk}^*, \delta^*)$ to $\mathscr{A}$. If the bit b chosen by $\mathscr{D}$'s challenger is $b = 0$, the partial signature σ^* is a real signature output by the PSig algorithm. On the other hand, if $b = 1$, σ^* is a possible output of Sim algorithm.

$\mathscr{A}$ continues to issuing queries except that it cannot issue a resolution query $(M^*, \theta^*, \mathtt{Pk}_S)$ to O_{Res^V}, where $\theta^* = (\zeta', (C'')^*, \hat{vk}^*)$, and $\mathscr{D}$ answers the queries as before.

Finally, $\mathscr{A}$ outputs a bit b' as its guess of whether σ^* was output by PSig or Sim. $\mathscr{D}$ outputs b' as its guess of b. If $\mathscr{A}$'s guess is correct, so is $\mathscr{D}$'s query. Therefore, $\mathscr{D}$'s advantage in breaking the selective-tag IND-CCA security of $\mathscr{E}$ is negligibly close to ε, due to the (negligible) difference caused by the zero-knowledge simulation of proof π. If ε is non-negligible, so is $\mathscr{D}$'s advantage. □

Lemma 5.5. *The generic construction is secure against signers if $\mathscr{S}$ is EUF-CMA secure and the proof π is sound and witness indistinguishable.*

Proof. Let $\mathscr{A}$ be an efficient adversary that breaks the security against signers with advantage ε. We use it to build another efficient algorithm $\mathscr{F}$ that breaks the EUF-CMA security of $\mathscr{S}$.

Given a public key pk^*, $\mathscr{F}$ runs **Akg** to generate the arbitrator's key pair $(\mathtt{Apk}, \mathtt{Ask})$, and runs $\mathscr{E}.\mathsf{Kg}(1^k)$ to generate an encryption key pair $(\mathtt{Pk}_V^E, \mathtt{Sk}_V^E)$. It sets $\mathtt{Pk}_V^S := pk^*$, and invokes the adversary on input $(\mathtt{Pk}_V, \mathtt{Apk})$. Next $\mathscr{F}$ begins to simulate oracles for the adversary as below.

- O_{FPSig}. Given $(M, \mathtt{Pk}')$, $\mathscr{F}$ sends M to its signing oracle and is returned a signature ζ. It then follows the rest of PSig algorithm to generate the partial signature $\sigma = (C', C'', \hat{vk}, \delta)$, and returns it to the adversary. Guaranteed by the witness indistinguishability of π, the proof looks almost the same as that in **PVer** and brings only negligible difference to the adversary's advantage.

- O_{Res}. Given $(M, \sigma, \mathtt{Pk}')$ where $\mathtt{Pk}' = ((\mathtt{Pk}^E)', (\mathtt{Pk}^S)')$ and $\sigma = (C', C'', \hat{vk}, \delta)$, $\mathscr{F}$ uses $\mathtt{Ask}$ to run Res^A algorithm to do the first level resolution. If the output is $\theta = \perp$, $\mathscr{F}$ returns $\perp$ as well. Otherwise, $\mathscr{F}$ parses θ as $(\zeta', C'', \hat{vk})$, and uses $\mathtt{Sk}_V^E$ to decrypt C'' with tag $\hat{vk}$. If the decryption outputs $\zeta'' = \perp$, $\mathscr{F}$ returns $\perp$ as well; otherwise, it computes $\zeta \leftarrow \zeta' \odot \zeta''$. If $\mathscr{S}.\mathsf{Ver}((\mathtt{Pk}^S)', M, \zeta) = 1$, $\mathscr{F}$ returns ζ; otherwise, it returns $\perp$.

At some time the adversary outputs $(M^*, \mathtt{Pk}_S, \sigma^*)$. $\mathscr{F}$ extracts a full signature ζ^* from σ^* in the same way as simulating the O_{Res} oracle. Suppose that $\mathscr{A}$ wins the game. According to the winning conditions, we know that $\mathscr{A}$ did not issue a query $(M^*, \mathtt{Pk}_S)$ to the FPSig oracle, and that ζ^* is not a valid signature with respect to $\mathtt{Pk}_S^S$ under $\mathscr{S}$ but σ^* could pass the checks in PVer. It means that the adversary could convince $\mathscr{F}$ to accept that σ^* is a partial signature valid with respect to $\mathtt{Pk}_S$ and $\mathtt{Pk}_V$. Guaranteed by the soundness of the proof π, we have that a signature valid (under $\mathscr{S}$) with respect to either $\mathtt{Pk}_S$ or $\mathtt{Pk}_V$ is contained in σ^*. However, since ζ^* is not a valid signature with respect to $\mathtt{Pk}_S$, it must be a valid signature with respect to $\mathtt{Pk}_V$. $\mathscr{F}$ outputs (M^*, ζ^*) as its forgery against $\mathscr{S}$ for $\mathscr{S}.\mathsf{Ver}(\mathtt{Pk}_V^S, M^*, \zeta^*) = 1$.

It is easy to see that the simulation of the two oracles is almost perfect. Thus, if $\mathscr{A}$ wins the game of security against signers with advantage at least (negligibly close to) ε, our algorithm $\mathscr{F}$ also wins the unforgeability game of the signature scheme $\mathscr{S}$ with advantage negligibly close ε, where the negligible difference is due to the soundness of the proof π provided by $\mathscr{A}$ to convince $\mathscr{F}$ the validity of σ^* and the witness indistinguishability of π in the simulation of O_{FPSig}. If ε is non-negligible, so is $\mathscr{F}$'s advantage. □

Lemma 5.6. *The generic construction is secure against the arbitrator if $\mathscr{S}$ is EUF-CMA secure.*

Proof. Let $\mathscr{A}$ be an efficient adversary that breaks the security against the arbitrator with advantage ε. We use it to build another algorithm $\mathscr{F}$ that breaks the EUF-CMA security of $\mathscr{S}$.

Given a public key pk^*, $\mathscr{F}$ runs **Akg** to generate a key pair $(\mathtt{Apk}, \mathtt{Ask})$ for the arbitrator and runs $\mathscr{E}.\mathsf{Kg}(1^k)$ to generate a key pair $(\mathtt{Pk}_S^E, \mathtt{Sk}_S^E)$. It sets $\mathtt{Pk}_S^S := pk^*$, and invokes the adversary on input $(\mathtt{Pk}_S, \mathtt{Apk}, \mathtt{Ask})$.

On input a partial signature query $(M, \mathtt{Pk}')$, $\mathscr{F}$ sends M to its signing oracle, and obtains the corresponding signature ζ. It then splits ζ to ζ', ζ'', generates $(\hat{vk}, \hat{sk})$ as prescribed, and encrypts ζ', ζ'' with tag $\hat{vk}$ under $\mathtt{Apk}$ and $\mathtt{Pk}'$, respectively. Let the ciphertexts be C' and C''. $\mathscr{F}$ authenticates $C'\|C''$ using $\hat{sk}$ and obtains the one-time signature δ. It returns $\sigma := (C', C'', \hat{vk}, \delta)$ to $\mathscr{A}$. Obviously, σ is properly distributed.

At last, the adversary outputs $(M^*, \mathtt{Pk}_V, \zeta^*)$. If $\mathscr{S}.\mathsf{Ver}(\mathtt{Pk}_S^S, M^*, \zeta^*) = 1$ and $\mathscr{A}$ did not issue a partial signature query on input $(M^*, \mathtt{Pk}_V)$, $\mathscr{F}$ outputs (M^*, ζ^*) as its forgery of $\mathscr{S}$. It is not hard to see that if $\mathscr{A}$ wins in the game of security against the arbitrator, so does $\mathscr{F}$ in its unforgettability game. And $\mathscr{F}$'s advantage in breaking the EUF-CMA security of $\mathscr{S}$ is at least the advantage of $\mathscr{A}$ in breaking the security against the arbitrator. □

6 Conclusion

In this paper we proposed a generic construction of $\mathsf{P^2OFE}$, which makes use of a tag-based encryption, a standard signature, a strong one-time signature and a witness-indistinguishable proof system. We proved that the construction is secure if the underlying primitives are secure without using the random oracle heuristic. Our work enriches the construction of $\mathsf{P^2OFE}$ protocols.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61472146, 61402184), Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2014A030306021), Guangdong Program for Special Support of Top-notch Young Professionals (No. 2015TQ01X796), and Pearl River Nova Program of Guangzhou (No. 201610010037).

References

[1] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *Proc. of the 4th ACM Conference on Computer and Communication Security (CCS'97), Zurich, Switzerland*, pages 7–17. ACM, April 1997.

[2] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures (extended abstract). In *Proc. of the 1998 International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'98), Espoo, Finland*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer, Berlin, Heidelberg, May-June 1998.

[3] B. Barak, R. Canetti, J. B. Nielsen, and R. Pass. Universally composable protocols with relaxed set-up assumptions. In *Proc. of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04), Rome, Italy*, pages 186–195. IEEE Computer Society, December 2004.

[4] D. Boneh and X. Boyen. Short signatures without random oracles. In *Proc. of the 2004 International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), Interlaken, Switzerland*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, Berlin, Heidelberg, May 2004.

[5] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. of the 24th Annual International Cryptology Conference (CRYPTO'04), Santa Barbara, California, USA*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, Berlin, Heidelberg, August 2004.

[6] Q. Huang, D. S. Wong, J. Li, and Y. Zhao. Generic transformation from weakly to strongly unforgeable signatures. *Journal of Computer Science and Technology*, 23(2):240–252, March 2008.

[7] Q. Huang, D. S. Wong, and W. Susilo. A new construction of designated confirmer signature and its application to optimistic fair exchange. In *Proc. of the 4th International Conference on Pairing-Based Cryptography (Pairing'10), Yamanaka Hot Spring, Japan*, volume 6487 of *Lecture Notes in Computer Science*, pages 41–61. Springer, Berlin, Heidelberg, December 2010.

[8] Q. Huang, D. S. Wong, and W. Susilo. Efficient designated confirmer signature and DCS-based ambiguous optimistic fair exchange. *IEEE Transactions on Information Forensics and Security*, 6(4):1233–1247, 2011.

[9] Q. Huang, D. S. Wong, and W. Susilo. Group-oriented fair exchange of signatures. *Information Sciences*, 181(16):3267–3283, 2011.

[10] Q. Huang, D. S. Wong, and W. Susilo. The construction of ambiguous optimistic fair exchange from designated confirmer signature without random oracles. In *Proc. of the 15th International Conference on Practice and Theory in Public Key Cryptography (PKC '12), Darmstadt, Germany*, volume 7293 of *Lecture Notes in Computer Science*, pages 120–137. Springer, Berlin, Heidelberg, May 2012.

[11] Q. Huang, D. S. Wong, and W. Susilo. P^{2}OFE: Privacy-Preserving Optimistic Fair Exchange of Digital Signatures. In *Proc. of the 2014 Cryptographer's Track at the RSA Conference (CT-RSA'14), San Francisco, California, USA*, volume 8366 of *Lecture Notes in Computer Science*, pages 367–384. Springer, Berlin, Heidelberg, February 2014.

[12] Q. Huang, D. S. Wong, and W. Susilo. How to protect privacy in optimistic fair exchange of digital signatures. *Information Sciences*, 325:300–315, 2015.

[13] Q. Huang, D. S. Wong, and Y. Zhao. Generic transformation to strongly unforgeable signatures. In *Proc. of the 5th International Conference on Applied Cryptography and Network Security (ACNS'07), Zhuhai, China*, volume 4521 of *Lecture Notes in Computer Science*, pages 1–17. Springer, Berlin, Heidelberg, June 2007.

[14] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. Ambiguous optimistic fair exchange. In *Proc. of the 14th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'08), Melbourne, Australia*, volume 5350 of *Lecture Notes in Computer Science*, pages 74–89. Springer, Berlin, Heidelberg, December 2008.

[15] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. A new efficient optimistic fair exchange protocol without random oracles. *International Journal of Information Security*, 11(1):53–63, 2012.

[16] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. Ambiguous optimistic fair exchange: Definition and constructions. *Theoretical Computer Science*, 562:177–193, 2015.

[17] X. Huang, Y. Mu, W. Susilo, W. Wu, and Y. Xiang. Optimistic fair exchange with strong resolution-ambiguity. *IEEE Journal on Selected Areas in Communications*, 29(7):1491–1502, 2011.

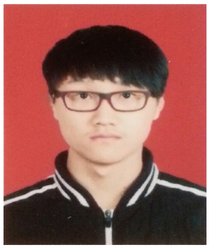
[18] X. Huang, Y. Mu, W. Susilo, W. Wu, J. Zhou, and R. H. Deng. Preserving transparency and accountability in optimistic fair exchange of digital signatures. *IEEE Transactions on Information Forensics and Security*, 6(2):498–512, 2011.

[19] E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *Proc. of the 3rd Theory of Cryptography Conference (TCC '06), New York, New York, USA*, volume 3876 of *Lecture Notes in Computer Science*, pages 581–600. Springer, Berlin, Heidelberg, March 2006.

[20] Y. Wang, M. H. Au, and W. Susilo. Perfect ambiguous optimistic fair exchange. In *Proc. of the 14th International Conference on Information and Communications Security (ICICS'12), Hong Kong*, volume 7618 of *Lecture Notes in Computer Science*, pages 142–153. Springer, Berlin, Heidelberg, October 2012.

Author Biography

Qingwen Guo is a master student at South China Agricultural University. His research interests is applied cryptography, especially the design of digital signatures and security protocols.



Yuzhao Cui is a master student at South China Agricultural University. His research interests is applied cryptography, especially the design of public key encryption schemes and security protocols.

Xiaomeng Zou is a master student at South China Agricultural University. Her research interests is applied cryptography, especially the design of block-chain based security protocols.

Qiong Huang got his PhD degree from City University of Hong Kong in 2010. Now he is a professor at South China Agricultural University. His research interests include cryptography and information security, in particular, cryptographic protocols design and analysis.