9-2016

Provably secure robust optimistic fair exchange of distributed signatures

Yujue WANG
Singapore Management University, yjwang@smu.edu.sg

Qianhong WU
Beihang University

Duncan S. WONG
City University of Hong Kong

Bo QIN
Renmin University of China

Jian MAO
Beihang University

See next page for additional authors

DOI: https://doi.org/10.1016/j.future.2016.03.012

Published in Future Generation Computer Systems, September 2016, Volume 62, Pages 29-39.
http://doi.org/10.1016/j.future.2016.03.012

Provably secure robust optimistic fair exchange of distributed signatures

Yujue Wang [a,g], Qianhong Wu [b,e], Duncan S. Wong [c], Bo Qin [d,e,*], Jian Mao [b,h], Yong Ding [f]

[a] School of Information Systems, Singapore Management University, Singapore
[b] School of Electronic and Information Engineering, Beihang University, Beijing, China
[c] Department of Computer Science, City University of Hong Kong, Hong Kong, China
[d] Key Laboratory of Data Engineering and Knowledge Engineering, Ministry of Education, School of Information, Renmin University of China, Beijing, China
[e] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
[f] School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin, China
[g] State Key Laboratory of Integrated Services Networks, Xidian University, Xian, China
[h] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, China

HIGHLIGHTS

- We model optimistic fair exchange between two groups.
- We define security in optimistic fair exchange of distributed signatures (OFEDS).
- OFEDS supports most generic access structure.
- The first OFEDS scheme is presented with robustness and non-interactive properties.
- Our scheme is proven secure under standard assumption.

ABSTRACT

We introduce the concept of optimistic fair exchange of distributed signatures (OFEDS) which allows two groups of parties to fairly exchange digital signatures. Specifically, an authorized set of parties from each group can jointly take part in the protocol on behalf of the affiliated group to fulfill obligation, and a semi-trusted arbitrator will intervene in the protocol only when there are disputes between two sides. Our OFEDS extends the functionality of optimistic fair exchange of threshold signatures to a more generic case. We formalize the security model of OFEDS, in which besides the standard security requirements for existing optimistic fair exchange protocols, robustness is incorporated so that OFEDS can be successfully performed even when there exist some dishonest signers. We propose a non-interactive construction of OFEDS based on the well-established Computational Diffie–Hellman (CDH) assumption. Our proposal shows that there exists CDH-based OFEDS for any general monotone access structure. Theoretical and experimental analyses demonstrate our OFEDS construction has reasonable efficiency for real applications.

Keywords:
Optimistic fair exchange
Distributed signature
Secret sharing
Monotone span program
Verifiably encrypted signatures

1. Introduction

Optimistic fair exchange (OFE) [1] allows two parties to exchange electronic items (e.g., electronic contracts) in a fair manner, i.e., either both parties obtain their expected items or neither party can do. A standard OFE protocol involves three parties, namely, party *A*, party *B* and an arbitrator. Among them, the arbitrator is semi-trusted and is intended to help *A* and *B* for resolving disputes during the item exchange process. In practice, the arbitrator is usually keeping offline unless either party *A* or *B* is cheated. The OFE protocol is carried out in three regular moves and two standby moves. Specifically, the party *A* first generates a *partial signature* which may be a verifiable encryption [2] of her *full signature* and gives it to *B*, who validates the received item and sends back

* Correspondence to: School of Information, Renmin University of China, No. 59, Zhongguanchun Avenue, Haidian District, 100872, Beijing, China. Tel.: +86 10 6251 2492.
E-mail address: bo.qin@ruc.edu.cn (B. Qin).

Table 1
Comparison of OFE schemes in multi-user setting.

Scheme	Functionality
Dodis, Lee and Yum [36]	OFE between *two individuals* with *one arbitrator*
Huang et al. [37]	OFE between *two individuals* with *one arbitrator*
Huang et al. [38]	OFE between *two individuals* with *one arbitrator*
Küpçü and Lysyanskaya [39]	OFE between *two individuals* with *multiple arbitrators*
Huang, Wong and Susilo [4]	OFE between *two parties* on behalf of *respective groups*
Qu et al. [5]	OFE between *two parties* on behalf of *respective rings*
Wang et al. [6]	OFE between *two authorized sets* more than a *threshold*
This paper	OFE between *two authorized sets* of *generic form*

his full signature to A when the validation result is true. If A accepts B's full signature and further provides her full signature to B, then these two parties have successfully exchanged their digital items, i.e., full signatures. Otherwise, if A refuses to fulfill her responsibility after receiving B's valid item, then B can ask the arbitrator for resolution by running standby moves. Note that OFE is different from oblivious transfer (OT) protocol [3], where OT allows users to exchange *secret* information in a fair manner.

Since its introduction, OFE has been received considerable attentions, of which several protocols were proposed in multi-user settings, e.g., [4–6]. However, all the existing OFE protocols cannot effectively support the applications in a more complicated scenario as follows. For example, a group of manufactories trying to sign a contract with a group of selling companies so that they can unify prices to avoid over competition. The authorized subset (e.g., several qualified representatives) is capable to sign it on behalf of the affiliated group, while none of the unauthorized ones can do so. In fact, here, the authorized sets constitute a *monotone access structure* which is similar to that in secret sharing schemes [7–9] and distributed signature schemes [10–14], which capture the threshold ones as special cases. Threshold primitives [15–18] can only support regular access policies, i.e., a set is authorized if it comprises at least a quorum of parties. Therefore, even the OFE protocol of threshold signatures [6] cannot apply to this generic setting. Note that the most related protocols, i.e., OFE protocols of group/ring signatures [4,5], just allow a single member from different groups to fairly exchange their digital items on behalf of respective groups. These types of protocols also cannot be applied to such complicated scenarios. This motivates the work in this paper.

1.1. Our contribution

We observe a gap in existing OFE models and instantiated protocols (as shown in Table 1). Specifically, to the best of our knowledge, there are no studies in the public literatures which work on fairly exchanging distributed signatures with regard to general monotone access structures. The monotone access structure has the desirable expressiveness to define the authorized set, and hence, is flexible and versatile in practice. Therefore, motivated by the above scenario, we extend the existing OFE concept to a multi-user setting in which both groups A and B consist of multiple users, and the corresponding authorized subsets of A and B can perform the role of these groups for fairly exchanging digital items. Particularly, we are interested in providing a universal solution to such applications.

First, we introduce the notion of *optimistic fair exchange of distributed signatures* (OFEDS) and present a formal definition. We also formalize the security model of OFEDS which includes the security requirements of standard OFE protocols, such as ambiguity and security against signers, verifier and arbitrator, respectively. The difference lies in that these security requirements are defined on a group of parties, rather than a single party (see Section 3.2 for details). Furthermore, a security requirement called *robustness* is incorporated so that distributed signatures can be successfully exchanged in a fair way even if there are some malicious users (signers).

Second, we present a non-interactive CDH-based OFEDS construction with better expressiveness, where the generic access structure is modeled by *monotone span program* (MSP). It is shown that the proposed OFEDS protocol meets all the security requirements that defined in the security model. Since the facts that every monotone access structure can be realized by a *linear secret sharing scheme* (LSSS) and LSSS has been proved equivalent to MSP [7], our construction can apply to any monotone access structures. Therefore, our construction covers the existing threshold-oriented OFE protocols as a special case.

Third, we thoroughly analyze our OFEDS construction from both theoretical and experimental perspectives. Only the user-key-generation algorithm has linear computation costs with the cardinality of the signer set. The efficiencies of partial-/full-signature-reconstruction algorithms are related to the signer number in an authorized set. All the other algorithms have constant computations. Therefore, the performance analyses show that the proposed OFEDS protocol is practical to support real applications.

Compared to the preliminary version [19], the contribution of this full paper lies in that the majority of Section 1 is revised, and moreover, the formal security model of OFEDS as well as the security proofs of our construction in the random oracle model are presented. The performance evaluations and analyses of our OFEDS construction are also the new results of this paper.

1.2. Related work

OFE for exchanging items between two parties in a *fair* way was first proposed by Asokan, Schunter and Waidner [1], where the arbitrator is needed only in case of depute occurrence. *Fairness* is an important and desirable property in many real-world applications and has received plenty of research attentions. For example, fairness in e-payments [20] guarantees the involved parties could get either their bought goods or payments with the help of online/offline bank. Fairness is also a fundamental requirement for players when carrying out online contests [21] like e-auctions and e-games, in the sense that all of unfair competitions should be prevented from the system.

To prevent the verifier (i.e., party B) from abusing the received partial signature from party A, Huang et al. [22] introduced *ambiguous* OFE which enhances the security at the signer side. In [23], Huang, Wong and Susilo presented an interactive ambiguous OFE based on the designated confirmer signature such that when generating the partial signature, the signer should interact with the verifier. Furthermore, Wang et al. [24] managed to enhance the security at both sides of A and B, that is, the communication transcripts will leak nothing with regard to the involved parties. To this end, they introduced the concept of *perfect ambiguous* OFE. Recently, Huang et al. [25] investigated *privacy-preserving* OFE with more rigorous security in the sense that even the arbitrator cannot learn the resolved signatures, which further enhances the

security of perfect ambiguous OFE. For simplifying the certificate management as in the traditional public-key crypto-systems, Zhang et al. [26,27] presented identity based OFE protocols based on the identity based verifiably encrypted signatures.

Since the seminal work of [1], a lot of efforts have been devoted to optimistic fair exchange in multi-user settings. The existing multi-user OFE protocols fall into two categories. For the first type, each party is an independent user and different electronic item can be exchanged between different pair of related users. Asokan, Schunter and Waidner [28] proposed an optimistic protocol for multi-user fair exchange, where each user should sign on an expected global description matrix of the exchange and commit to all the electronic items which he will send to other parties, and thus each user can exchange electronic items with all others over unreliable networks. Note that in their protocol [28], a user can exchange different items with different users and reasonably much information should be broadcasted. Franklin and Tsudik [29] and Khill et al. [30] investigated a relatively simplified scenario where each user receives an item from the neighboring user and gives an item to next one. The same case was also studied by Bao et al. [31] but with an off-line trusted neutral party. Technically, their multi-user fair exchange protocol is designed based on verifiable encryption. Their protocol [31] was subsequently improved by González-Deleito and Markowitch [32] without assuming that the protocol initiator is also trusted by all users. Mukhamedov, Kremer and Ritter [33] revisited the problem in [29,34] using the strand space model. The exclusion scenarios in a multi-user setting have been discussed in [35].

For the second type OFE protocols in multi-user settings, the electronic items are exchanged between two groups of parties, i.e., a group of signers and a group of verifiers, which are summarized in Table 1. Dodis, Lee and Yum [36] observed that the single-party security of OFE protocols cannot guarantee multi-user security. In their formal security model [36], every party in each group also acts as an individual. Huang et al. [37] strengthened their security model [36] by introducing the chosen-key model into OFE in multi-user settings and considered the malicious arbitrator working in an adaptive model. On the other hand, some conditions (e.g., an extra property called strong resolution-ambiguity) were found by Huang et al. [38], under which the security of OFE in single-user setting is preserved in multi-user setting. In order to reduce the risk of relying on a single semi-trusted arbitrator, Küpçü and Lysyanskaya [39] considered the case where multiple arbitrators are involved. Huang, Wong and Susilo [4] proposed a group-oriented OFE which allows fair and anonymous exchange of digital items between two users on behalf of their respectively affiliated groups. In addition, they [4] also proposed a generic transformation from a group-oriented OFE into an ambiguous OFE [22]. Qu et al. [5] presented OFE of ring signatures free of group managers. Recently, a threshold-oriented OFE was introduced by Wang et al. [6] such that threshold signatures can be fairly exchanged, in which the group's duty is jointly taken by at least a threshold of its users.

2. Preliminaries

2.1. Bilinear pairing and computational assumptions

Bilinear Map [40]. Suppose $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ are cyclic groups of prime order q. The map $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is bilinear if it satisfies:

- Bilinearity: $\forall a, b \in_R \mathbb{G}$ and $\forall \mu, \nu \in_R \mathbb{Z}_q$, $\hat{e}(a^\mu, b^\nu) = \hat{e}(a, b)^{\mu\nu}$;
- Non-degeneracy: $\hat{e}(g, g) \neq 1$;
- Efficiency: the map $\hat{e}$ and the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ are efficiently computable.

Computational Diffie–Hellman (CDH) Problem [41]. Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order q. Given a triple (g, g^μ, g^ν) for randomly chosen values $\mu, \nu \in_R \mathbb{Z}_q^*$, compute $g^{\mu\nu}$.

Definition 1 (*CDH Assumption [41]*)**.** For any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ and any random values $\mu, \nu \in_R \mathbb{Z}_q^*$, the probability of solving CDH problem $\Pr[\mathcal{A}(g, g^\mu, g^\nu) = g^{\mu\nu}]$ is negligible.

2.2. Secret sharing

Secret sharing [42,43,7] is a method to securely distribute a secret among a group $\mathcal{S}$ of n parties, where each party gets one *share*, and the secret can be recovered only if an *authorized set* of parties pool their *shares* together. All the authorized sets constitute an *access structure* Π which satisfies *monotone increasing property* in which any set contains an authorized set is also authorized. In this paper, we are only interested in *prefect* secret sharing schemes, that is, any unauthorized sets cannot get any information about the shared secret. We use $\bar{\Pi} = 2^{\mathcal{S}} \setminus \Pi$ to denote the collection of all the unauthorized sets, where $2^{\mathcal{S}}$ is the power set of $\mathcal{S}$. Similarly, $\bar{\Pi}$ satisfies *monotone decreasing property*. Accordingly, we will use $\bar{\Pi}_X \subseteq \bar{\Pi}$ to denote the collection of all the *maximal unauthorized sets*.

Definition 2 (*Perfect Secret Sharing [7]*)**.** Let Π be a monotone access structure defined on a party set $\mathcal{S}$ and $\mathbb{F}$ be a finite set of secrets. Given a secret sharing scheme $\Phi = \langle \mathsf{Dis}(\cdot), \mathsf{Rec}(\cdot) \rangle$ realizing Π, where $\mathsf{Dis}(\cdot)$ and $\mathsf{Rec}(\cdot)$ are randomized distribution algorithm and reconstruction algorithm, respectively. We say that Φ is a *perfect* secret sharing scheme if the following two properties are satisfied.

- For any authorized set $A \in \Pi$ and any secret $k \in \mathbb{F}$, k can be definitely recovered from the corresponding shares given to A, that is, $\Pr[\mathsf{Rec}(A \xleftarrow{shares} \mathsf{Dis}(k)) = k] = 1$;
- For any unauthorized set $B \in \bar{\Pi}$ and any two distinct secrets $k_1, k_2 \in \mathbb{F}$, the distributions on the shares given to B of respective secrets, i.e, $B \xleftarrow{shares} \mathsf{Dis}(k_1)$ and $B \xleftarrow{shares} \mathsf{Dis}(k_2)$ are identical.

In this paper, any given monotone access structure will be realized by linear secret sharing schemes [7,10,44]. This indicates that both $\mathsf{Dis}(\cdot)$ and $\mathsf{Rec}(\cdot)$ are in fact performed as linear functions.

Definition 3 (*Monotone Span Program [45,7]*)**.** Let $\mathcal{S}$ be a party set, a and b be two positive integers. A monotone span program is defined as $\mathcal{M} = (\mathbb{F}, \vec{\tau}, M_{a\times b}, \rho)$, where $\mathbb{F}$ is a finite field, $\vec{\tau} \in \mathbb{F}^b$ is a target vector, $M_{a\times b}$ is an $a \times b$ matrix over $\mathbb{F}$ and $\rho : \{1, 2, \ldots, a\} \to \mathcal{S}$ labels each row of $M_{a\times b}$ by a party in $\mathcal{S}$. For any set $S \subseteq \mathcal{S}$, there exists a sub-matrix M_S which comprises all the rows of $M_{a\times b}$ that labeled by the parties in S. A set $S \subseteq \mathcal{S}$ is accepted by $\mathcal{M}$ if the target vector $\vec{\tau}$ can be spanned by the rows of M_S. An access structure Π which defined on $\mathcal{S}$ is accepted by $\mathcal{M}$ if and only if $\mathcal{M}$ accepts all the sets $S \in \Pi$.

MSP was first formalized by Karchmer and Wigderson [45] and has been implicitly discussed before by Brickell [46]. We will use MSP to model LSSS, i.e., there exists a matrix $M_{a\times b}$ such that $\mathsf{Dis}(\cdot)$ can be modeled by a mapping from the rows of $M_{a\times b}$ to the party set $\mathcal{S}$, and $\mathsf{Rec}(\cdot)$ can be modeled by the linear relationships among the rows in this matrix. Since MSP can realize any general monotone access structure, there may be several rows of $M_{a\times b}$ labeled to one party. For convenience and without loss of generality, we assume there are one-to-one correspondence between the rows of $M_{a\times b}$ and the parties in $\mathcal{S}$. Therefore, in the upcoming sections, we will use the vector $\vec{v}_i = \rho^{-1}(\mathfrak{s}_i)$ to denote the row of $M_{a\times b}$ related to the party $\mathfrak{s}_i \in \mathcal{S}$.

Table 2
Notations.

Notation	Meaning
$\mathcal{S}$	A set of signers $\{\mathfrak{s}_1, \ldots, \mathfrak{s}_n\}$
$\mathbb{S}$	A subset of signers $\mathbb{S} \subseteq \mathcal{S}$
S	An authorized set of signers
$\tilde{S}$	An unauthorized set of signers
Π	A monotone access structure
$\tilde{\Pi}$	The collection of unauthorized sets
λ	Security parameter
m	Message
PK_A, SK_A	Public/private keys of arbitrator
PK_U	Public key of the signer set
SK_i	Secret-key-share of signer $\mathfrak{s}_i$
$\hat{\sigma}_i, \sigma_i$	Partial-/Full-signature-fragment of signer $\mathfrak{s}_i$
$\hat{\sigma}, \sigma$	Partial/Full signature
q	A large prime
$\mathbb{G}, \mathbb{G}_T$	Cyclic groups associated with a bilinear map $\hat{e}$
H	A hash function

Table 3
Acronyms in OFEDS.

Acronym	Algorithm
AKGen	Arbitrator key generation algorithm
UKGen	User key generation algorithm
PSign, FSign	Distributed partial/full signature scheme
PSGen, FSGen	Partial/Full signature fragment generation algorithm
PSVrfy, FSVrfy	Partial/Full signature fragment verification algorithm
PSRCon, FSRCon	Partial/Full signature reconstruction algorithm
PVrfy, FVrfy	Partial/Full signature verification algorithm
Res	Resolution algorithm

3. Modeling optimistic fair exchange of distributed signatures

For ease of presentation in the rest parts, we assume that there are a group of signers but only a single verifier. Table 2 summarizes the frequently used notations when defining OFEDS framework and presenting OFEDS scheme, while Table 3 summarizes the acronyms of each algorithm in OFEDS.

3.1. Definition of OFEDS

Let Π be a monotone access structure realized by MSP on the signer set $\mathcal{S} = \{\mathfrak{s}_1, \ldots, \mathfrak{s}_n\}$ and $\lambda \in \mathbb{N}$ be a security parameter. A Π-optimistic fair exchange protocol of distributed signatures is defined as a series of non-interactive algorithms **OFEDS** $= \langle$AKGen, *UKGen*, *PSign* $= \langle$*PSGen*, *PSVrfy*, *PSRCon*, *PVrfy*$\rangle$, *FSign* $= \langle$*FSGen*, *FSVrfy*, *FSRCon*, *FVrfy*$\rangle$, *Res*$\rangle$, where all the algorithms are computable in time polynomial in λ.

AKGen: on input 1^λ, the (randomized) arbitrator key generation algorithm outputs a pair of public and private keys (PK_A, SK_A) for the arbitrator, denoted by, $(PK_A, SK_A) \leftarrow \mathsf{AKGen}(1^\lambda)$.

UKGen: on input 1^λ and a monotone access structure Π, the (randomized) user key generation algorithm outputs a public key PK_U for the signer set $\mathcal{S}$ and n secret-key-shares $(SK_1, \ldots, SK_n)$ for signers. This procedure is denoted by $(PK_U, SK_1, \ldots, SK_n) \leftarrow \mathsf{UKGen}(1^\lambda, \Pi)$.

PSign: PSign can be viewed as a distributed signature scheme which enables an authorized set of signers to jointly generate a partial signature for any given message. In detail, it consists of the following four algorithms:

- PSGen: on input a message $m \in \{0,1\}^*$, a secret-key-share SK_i of signer $\mathfrak{s}_i$, the signer set's public key PK_U and the arbitrator's public key PK_A, the partial signature fragment generation algorithm computes a partial-signature-fragment $\hat{\sigma}_i$ for signer $\mathfrak{s}_i$. We denote this procedure by $\hat{\sigma}_i \leftarrow \mathsf{PSGen}(m, SK_i, PK_U, PK_A)$ for any $\mathfrak{s}_i \in \mathcal{S}$.
- PSVrfy: on input a message $m \in \{0,1\}^*$, a partial-signature-fragment $\hat{\sigma}_i$ of signer $\mathfrak{s}_i \in \mathcal{S}$, the signer set's public key PK_U and the arbitrator's public key PK_A, the partial signature fragment verification algorithm validates $\hat{\sigma}_i$ and outputs "1" if it is `valid` for m, and "0" otherwise. We denote this procedure by $\{1,0\} \leftarrow \mathsf{PSVrfy}(m, \hat{\sigma}_i, PK_U, PK_A)$.
- PSRCon: on input a message $m \in \{0,1\}^*$, some partial-signature-fragments $\{\hat{\sigma}_i \mid \mathfrak{s}_i \in \mathbb{S}\}$ where $\mathbb{S} \subseteq \mathcal{S}$, the signer set's public key PK_U and the arbitrator's public key PK_A, the partial signature reconstruction algorithm outputs a partial signature $\hat{\sigma}$ or $\perp$ according to Π. We denote this procedure by $\{\hat{\sigma}, \perp\} \leftarrow \mathsf{PSRCon}(m, \{\hat{\sigma}_i \mid \mathfrak{s}_i \in \mathbb{S} \wedge \mathbb{S} \subseteq \mathcal{S}\}, PK_U, PK_A, \Pi)$.
- PVrfy: on input a message $m \in \{0,1\}^*$, a partial signature $\hat{\sigma}$, the signer set's public key PK_U and the arbitrator's public key PK_A, the partial signature verification algorithm validates $\hat{\sigma}$ and outputs "1" if it is `valid` for m, and "0" otherwise. The procedure is denoted by $\{1,0\} \leftarrow \mathsf{PVrfy}(m, \hat{\sigma}, PK_U, PK_A)$.

FSign: Similar to PSign, FSign allows an authorized set of signers to jointly generate a full signature for any given message:

- FSGen: on input a message $m \in \{0,1\}^*$, a secret-key-share SK_i, the signer set's public key PK_U and the arbitrator's public key PK_A, the full signature fragment generation algorithm computes a full-signature-fragment σ_i for signer $\mathfrak{s}_i$. We denote this procedure by $\sigma_i \leftarrow \mathsf{FSGen}(m, SK_i, PK_U, PK_A)$ for any $\mathfrak{s}_i \in \mathcal{S}$.
- FSVrfy: on input a message $m \in \{0,1\}^*$, a full-signature-fragment σ_i of signer $\mathfrak{s}_i \in \mathcal{S}$, the signer set's public key PK_U and the arbitrator's public key PK_A, the full signature fragment verification algorithm validates σ_i and outputs "1" if it is `valid` for m, and "0" otherwise. We denote this procedure by $\{1,0\} \leftarrow \mathsf{FSVrfy}(m, \sigma_i, PK_U, PK_A)$.
- FSRCon: on input a message $m \in \{0,1\}^*$, some full-signature-fragments $\{\sigma_i \mid \mathfrak{s}_i \in \mathbb{S}\}$ where $\mathbb{S} \subseteq \mathcal{S}$, the signer set's public key PK_U and the arbitrator's public key PK_A, the full signature reconstruction algorithm outputs a full signature σ or $\perp$ according to Π. We denote this procedure by $\{\sigma, \perp\} \leftarrow \mathsf{FSRCon}(m, \{\sigma_i \mid \mathfrak{s}_i \in \mathbb{S} \wedge \mathbb{S} \subseteq \mathcal{S}\}, PK_U, PK_A, \Pi)$.
- FVrfy: on input $m \in \{0,1\}^*$, a full signature σ, the signer set's public key PK_U and the arbitrator's public key PK_A, the full signature verification algorithm validates σ and outputs "1" if it is `valid` for m, and "0" otherwise. The procedure is denoted by $\{1,0\} \leftarrow \mathsf{FVrfy}(m, \sigma, PK_U, PK_A)$.

Res: on input a message $m \in \{0,1\}^*$, a partial signature $\hat{\sigma}$, the signer set's public key PK_U and the arbitrator's public–private key pair (PK_A, SK_A), the resolution algorithm outputs a full signature σ for message m if $\hat{\sigma}$ is valid for m, and $\perp$ otherwise. We denote this procedure by $\{\sigma, \perp\} \leftarrow \mathsf{Res}(m, \hat{\sigma}, PK_U, SK_A, PK_A)$.

A typical procedure of optimistic fair exchange of distributed signatures is shown in Fig. 1.

3.2. Security requirements

We proceed to define the security model of **OFEDS**, i.e., Π-optimistic fair exchange protocol of distributed signatures.

Correctness for a Π-OFEDS means that a valid partial (resp. full) signature for any message can be definitely reconstructed from an authorized set of valid partial-(resp. full)-signature-fragments, and a valid partial signature can always be resolved into a valid full signature by the arbitrator.

Definition 4 (*Correctness*). **Π-OFEDS** is said to be *correct* if for any $(PK_A, SK_A) \leftarrow \mathsf{AKGen}(1^\lambda)$, any $(PK_U, SK_1, \ldots, SK_n) \leftarrow \mathsf{UKGen}(1^\lambda, \Pi)$, any $S \in \Pi$ and any message $m \in_R \{0,1\}^*$, all the following statements hold

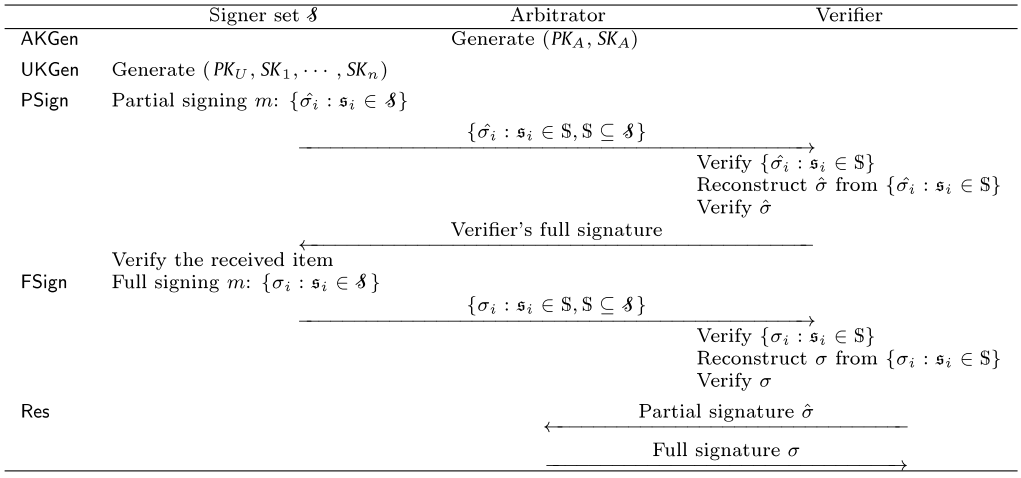

Fig. 1. A procedure of OFEDS.

- $\mathsf{PVrfy}(m, \hat{\sigma}, PK_U, PK_A) = 1$ for any $\hat{\sigma} \leftarrow \mathsf{PSRCon}(m, \{\mathsf{PSGen}(m, SK_i, PK_U, PK_A) \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$;
- $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$ for any $\sigma \leftarrow \mathsf{FSRCon}(m, \{\mathsf{FSGen}(m, SK_i, PK_U, PK_A) \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$;
- $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$ for any $\sigma \leftarrow \mathsf{Res}(m, \hat{\sigma}, PK_U, SK_A, PK_A)$.

Resolution ambiguity for a Π-OFEDS means that a full signature reconstructed from an authorized set of full-signature-fragments cannot be distinguished from the one resolved by the arbitrator.

Definition 5 (*Resolution Ambiguity*)**.** Π-**OFEDS** is said to be *ambiguous* if for any $(PK_A, SK_A) \leftarrow \mathsf{AKGen}(1^\lambda)$, any $(PK_U, SK_1, \ldots, SK_n) \leftarrow \mathsf{UKGen}(1^\lambda, \Pi)$, any $S \in \Pi$ and any message $m \in_R \{0,1\}^*$, there exists no polynomial-time algorithm which can distinguish $\sigma \leftarrow \mathsf{Res}(m, \hat{\sigma}, PK_U, SK_A, PK_A)$ from $\sigma \leftarrow \mathsf{FSRCon}(m, \{\mathsf{FSGen}(m, SK_i, PK_U, PK_A) \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$, where $\hat{\sigma} \leftarrow \mathsf{PSRCon}(m, \{\mathsf{PSGen}(m, SK_i, PK_U, PK_A) \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$.

Robustness for a Π-OFEDS means that the electronic items can be still successfully fairly exchanged even if there are an unauthorized set of signers deviate from the protocol.

Definition 6 (*Robustness*)**.** Suppose there is a malicious adversary $\mathcal{A}$ who controls an unauthorized set $\tilde{S} \in \bar{\Pi}$ of signers. Π-**OFEDS** is said to be $\bar{\Pi}$*-robust* if for any $(PK_A, SK_A) \leftarrow \mathsf{AKGen}(1^\lambda)$, any $(PK_U, SK_1, \ldots, SK_n) \leftarrow \mathsf{UKGen}(1^\lambda, \Pi)$ and any message $m \in_R \{0,1\}^*$, there exists $S \subseteq \mathcal{S} \setminus \tilde{S}$ such that

- $\mathsf{PVrfy}(m, \hat{\sigma}, PK_U, PK_A) = 1$ for any $\hat{\sigma} \leftarrow \mathsf{PSRCon}(m, \{\hat{\sigma_i}' \mid \mathfrak{s}_i \in \tilde{S}\} \cup \{\hat{\sigma_i} \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$, where $\hat{\sigma_i} \leftarrow \mathsf{PSGen}(m, SK_i, PK_U, PK_A)$;
- $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$ for any $\sigma \leftarrow \mathsf{FSRCon}(m, \{\sigma_i' \mid \mathfrak{s}_i \in \tilde{S}\} \cup \{\sigma_i \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$, where $\sigma_i \leftarrow \mathsf{FSGen}(m, SK_i, PK_U, PK_A)$;
- $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$ for any $\sigma \leftarrow \mathsf{Res}(m, \hat{\sigma}, PK_U, SK_A, PK_A)$, where $\hat{\sigma} \leftarrow \mathsf{PSRCon}(m, \{\hat{\sigma_i}' \mid \mathfrak{s}_i \in \tilde{S}\} \cup \{\hat{\sigma_i} \mid \mathfrak{s}_i \in S\}, PK_U, PK_A, \Pi)$ and $\hat{\sigma_i} \leftarrow \mathsf{PSGen}(m, SK_i, PK_U, PK_A)$.

A secure Π-OFEDS should ensure that even when all the signers are dishonest, they still cannot fool the verifier and the arbitrator. In other words, a partial signature reconstructed from their partial-signature-fragments should always be resolved into a valid full signature.

Definition 7 (*Security Against Signers*)**.** A Π-**OFEDS** is *secure against signers*, if there is no adversary $\mathcal{A}$ who controls all the signers can win in the following game in polynomial time:

After received the access structure Π from $\mathcal{A}$, the challenger runs algorithms $\mathsf{AKGen}(1^\lambda)$ and $\mathsf{UKGen}(1^\lambda, \Pi)$ to get the keys $(PK_A, SK_A, PK_U, SK_1, \ldots, SK_n)$ and sends $(PK_A, PK_U, SK_1, \ldots, SK_n)$ back to $\mathcal{A}$.

The adversary $\mathcal{A}$ adaptively sends q_R ($q_R \in \mathbb{N}$) resolution queries $(m_i, \hat{\sigma}^{(i)})$ to the challenger, where $m_i \in_R \{0,1\}^*$. For each query $(m_i, \hat{\sigma}^{(i)})$, if $\mathsf{PVrfy}(m_i, \hat{\sigma}^{(i)}, PK_U, PK_A) = 1$, then the challenger answers $\sigma^{(i)} \leftarrow \mathsf{Res}(m_i, \hat{\sigma}^{(i)}, PK_U, SK_A, PK_A)$.

Eventually, the adversary $\mathcal{A}$ wins in the game if she outputs a pair $(m, \hat{\sigma})$ that satisfies

- $m \notin \{m_i \mid i \in [1, q_R]\}$;
- $\mathsf{PVrfy}(m, \hat{\sigma}, PK_U, PK_A) = 1$;
- $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 0$ for the resolved full signature $\sigma \leftarrow \mathsf{Res}(m, \hat{\sigma}, PK_U, SK_A, PK_A)$.

A secure Π-OFEDS should also ensure that even when the verifier colludes with an unauthorized set of signers, they are still unable to generate a valid full-signature-forgery.

Definition 8 (*Security Against Verifier*)**.** A Π-**OFEDS** is *secure against verifier*, if there is no adversary $\mathcal{A}$ who controls the verifier and an unauthorized set $\tilde{S} \in \bar{\Pi}$ of signers can win in the following game in polynomial time:

After received an access structure Π and a corrupted set $\tilde{S}$ from $\mathcal{A}$, the challenger runs algorithms $\mathsf{AKGen}(1^\lambda)$ and $\mathsf{UKGen}(1^\lambda, \Pi)$ to get the keys $(PK_A, SK_A, PK_U, SK_1, \ldots, SK_n)$ and sends $(PK_A, PK_U, \{SK_i \mid \mathfrak{s}_i \in \tilde{S}\})$ back to $\mathcal{A}$.

The adversary $\mathcal{A}$ can adaptively make the following queries to the challenger, of which PSGen queries and Res queries cannot be requested for the same messages. Since $\mathcal{A}$ has obtained all the secret-key-shares of the signers in the corrupted set $\tilde{S}$, it is reasonable to simply let all the following PSGen and FSGen queries be made for uncorrupted signers.

- PSGen queries: the adversary $\mathcal{A}$ can adaptively submit q_P ($q_P \in \mathbb{N}$) such queries $(m_i, \mathfrak{s}_j)$ to the challenger, where $m_i \in_R \{0,1\}^*$ and $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. For each query, the challenger responds with $\hat{\sigma_j}^{(i)} \leftarrow \mathsf{PSGen}(m_i, SK_j, PK_U, PK_A)$;
- FSGen queries: the adversary $\mathcal{A}$ can adaptively submit q_F ($q_F \in \mathbb{N}$) such queries $(m_i, \mathfrak{s}_j)$ to the challenger, where $m_i \in_R \{0,1\}^*$ and $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. For each query, the challenger responds with $\sigma_j^{(i)} \leftarrow \mathsf{FSGen}(m_i, SK_j, PK_U, PK_A)$;

- Res queries: the adversary $\mathcal{A}$ can adaptively submit q_R ($q_R \in \mathbb{N}$) resolution queries $(m_i, \hat{\sigma}^{(i)})$ to the challenger where $m_i \in_R \{0,1\}^*$. For each query, if $\mathsf{PVrfy}(m_i, \hat{\sigma}^{(i)}, PK_U, PK_A) = 1$, then the challenger answers $\sigma^{(i)} \leftarrow \mathsf{Res}(m_i, \hat{\sigma}^{(i)}, PK_U, SK_A, PK_A)$.

Eventually, the adversary $\mathcal{A}$ wins in the game if she outputs a pair (m, σ) such that $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$, where $m \in \{0,1\}^*$ has neither been requested before in FSGen queries nor been asked in Res queries.

A secure Π-OFEDS should also ensure that even when the arbitrator colludes with an unauthorized set of signers, they still cannot produce a valid full-signature-forgery.

Definition 9 (*Security Against Arbitrator*)**.** A Π-**OFEDS** is *secure against arbitrator*, if there is no adversary $\mathcal{A}$ who controls the arbitrator and an unauthorized set $\tilde{S} \in \bar{\Pi}$ of signers can win in the following game in polynomial time:

After received an access structure Π and a corrupted set $\tilde{S}$ from $\mathcal{A}$, the challenger runs algorithms $\mathsf{AKGen}(1^\lambda)$ and $\mathsf{UKGen}(1^\lambda, \Pi)$ to get the keys $(PK_A, SK_A, PK_U, SK_1, \ldots, SK_n)$ and sends $(PK_A, SK_A, PK_U, \{SK_i \mid \mathfrak{s}_i \in \tilde{S}\})$ back to $\mathcal{A}$.

The adversary $\mathcal{A}$ can adaptively make the following queries to the challenger. Similar to the above security game against verifier, we can also simply let all the following queries be made with regard to uncorrupted signers.

- PSGen queries: the adversary $\mathcal{A}$ can adaptively submit q_P ($q_P \in \mathbb{N}$) queries $(m_i, \mathfrak{s}_j)$ to the challenger, where $m_i \in_R \{0,1\}^*$ and $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. For each query, the challenger responds with $\hat{\sigma}_j^{(i)} \leftarrow \mathsf{PSGen}(m_i, SK_j, PK_U, PK_A)$;
- FSGen queries: the adversary $\mathcal{A}$ can adaptively submit q_F ($q_F \in \mathbb{N}$) queries $(m_i, \mathfrak{s}_j)$ to the challenger, where $m_i \in_R \{0,1\}^*$ and $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. For each query, the challenger responds with $\sigma_j^{(i)} \leftarrow \mathsf{FSGen}(m_i, SK_j, PK_U, PK_A)$.

Eventually, the adversary $\mathcal{A}$ wins in the game if she outputs a pair (m, σ) such that $\mathsf{FVrfy}(m, \sigma, PK_U, PK_A) = 1$, where $m \in \{0,1\}^*$ has neither been requested before in PSGen queries nor been asked in FSGen queries.

4. Construction

4.1. A CDH-based OFEDS protocol

We proceed to give our construction which is inspired by Wang et al.'s protocol [6]. Let q be a prime which is determined by the security parameter $\lambda (\lambda \in \mathbb{N})$. Suppose $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient bilinear map, where $\mathbb{G} = \langle g \rangle$ and $\mathbb{G}_T$ are cyclic groups of order q. Let $H : \{0,1\}^* \to \mathbb{G}$ be a hash function and Π be a monotone access structure realized by MSP on the signer set $\mathcal{S} = \{\mathfrak{s}_1, \ldots, \mathfrak{s}_n\}$. Our construction comprises the following algorithms.

AKGen: This algorithm produces the arbitrator's public–private key pair as $(PK_A, SK_A) = (g^y, y)$, where $y \in_R \mathbb{Z}_q^*$.

UKGen: This algorithm chooses a random vector $\vec{w} \in_R (\mathbb{Z}_q)^b$ and computes $SK_i = \vec{w} \cdot \vec{v}_i \bmod q$ for each signer $\mathfrak{s}_i \in \mathcal{S}$. Here, b is the column number of MSP matrix. The public key of the signer set $\mathcal{S}$ is set to be

$$PK_U = (PK_0, PK_1, \ldots, PK_n) = \left(g^{SK_0}, g^{SK_1}, \ldots, g^{SK_n}\right),$$

where $SK_0 = \vec{w} \cdot \vec{\tau} \bmod q$.

PSign: PSign consists of the following four algorithms:

- PSGen: Given a message $m \in \{0,1\}^*$, each signer $\mathfrak{s}_i \in \mathcal{S}$ randomly picks an element $r_i \in_R \mathbb{Z}_q^*$ and computes a partial-signature-fragment $\hat{\sigma}_i = (\hat{\alpha}_i, \hat{\beta}_i)$, where
$$\hat{\alpha}_i = H(m)^{SK_i} \cdot PK_A^{r_i} \quad \text{and} \quad \hat{\beta}_i = g^{r_i}.$$
- PSVrfy: Given a message $m \in \{0,1\}^*$ and a purported partial-signature-fragment $\hat{\sigma}_i = (\hat{\alpha}_i, \hat{\beta}_i)$, this algorithm outputs "1" if the following equality holds
$$\hat{e}(\hat{\alpha}_i, g) \stackrel{?}{=} \hat{e}(H(m), PK_i) \cdot \hat{e}(PK_A, \hat{\beta}_i).$$
Otherwise, it outputs "0".
- PSRCon: Given a message $m \in \{0,1\}^*$ and a group of valid partial-signature-fragments $\{\hat{\sigma}_i \mid \mathfrak{s}_i \in S \wedge S \subseteq \mathcal{S}\}$ on m. If $S \in \Pi$, then there exist a group of values $\{c_i \in \mathbb{Z}_q \mid \mathfrak{s}_i \in S\}$ which can be found by solving the system of linear equations such that
$$\vec{\tau} = \sum_{\mathfrak{s}_i \in S} c_i \vec{v}_i \bmod q.$$
Thereby, the partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$ can be reconstructed by computing
$$\hat{\alpha} = \prod_{\mathfrak{s}_i \in S} \hat{\alpha}_i^{c_i} \quad \text{and} \quad \hat{\beta} = \prod_{\mathfrak{s}_i \in S} \hat{\beta}_i^{c_i}.$$
Otherwise, it outputs $\perp$.
- PVrfy: Given a message $m \in \{0,1\}^*$ and a purported partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$, this algorithm outputs "1" if the following equality holds
$$\hat{e}(\hat{\alpha}, g) \stackrel{?}{=} \hat{e}\,(H(m), PK_0) \cdot \hat{e}(PK_A, \hat{\beta}). \tag{1}$$
Otherwise, it outputs "0".

FSign: FSign consists of the following four algorithms:

- FSGen: Given a message $m \in \{0,1\}^*$, each signer $\mathfrak{s}_i \in \mathcal{S}$ computes a full-signature-fragment as $\sigma_i = H(m)^{SK_i}$.
- FSVrfy: Given a message $m \in \{0,1\}^*$ and a purported full-signature-fragment σ_i, this algorithm outputs "1" if $\hat{e}(\sigma_i, g) \stackrel{?}{=} \hat{e}\,(H(m), PK_i)$ holds; otherwise it outputs "0".
- FSRCon: Given a message $m \in \{0,1\}^*$ and a group of valid full-signature-fragments $\{\sigma_i \mid \mathfrak{s}_i \in S \wedge S \subseteq \mathcal{S}\}$ on m. If $S \in \Pi$, then there exist a group of values $\{c_i \in \mathbb{Z}_q \mid \mathfrak{s}_i \in S\}$ as discussed in algorithm PSRCon. Therefore, the full signature σ can be reconstructed as follows
$$\sigma = \prod_{\mathfrak{s}_i \in S} \sigma_i^{c_i}.$$
Otherwise, it outputs $\perp$.
- FVrfy: Given a message $m \in \{0,1\}^*$ and a purported full signature σ, this algorithm outputs "1" if $\hat{e}(\sigma, g) \stackrel{?}{=} \hat{e}(H(m), PK_0)$ holds; otherwise it outputs "0".

Res: Given a message $m \in \{0,1\}^*$ and a partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$, this algorithm outputs a full signature $\sigma = \hat{\alpha}/\hat{\beta}^{SK_A}$ if $\hat{\sigma}$ is valid under Equality (1); otherwise it outputs $\perp$.

Theorem 1. *The above proposed Π-OFEDS protocol is correct.*

Proof. For any authorized set $S \in \Pi$, there exists a collection of values $\{c_i \in \mathbb{Z}_q \mid \mathfrak{s}_i \in S\}$ such that

$$\sum_{\mathfrak{s}_i \in S} c_i \vec{v}_i = \vec{\tau} \bmod q,$$

because S is accepted by MSP. In fact, these values can be found by solving the system of linear equations. Therefore,

$$SK_0 = \vec{w} \cdot \vec{\tau} = \vec{w} \cdot \left(\sum_{\mathfrak{s}_i \in S} c_i \vec{v}_i\right) = \sum_{\mathfrak{s}_i \in S} c_i\,(\vec{w} \cdot \vec{v}_i) = \sum_{\mathfrak{s}_i \in S} c_i SK_i \bmod q.$$

Then, the reconstruction of the partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$ from partial-signature-fragments $\{\hat{\sigma}_i = (\hat{\alpha}_i, \hat{\beta}_i) \mid \mathfrak{s}_i \in S\}$ is due to

$$\hat{\alpha} = \prod_{\mathfrak{s}_i \in S} \hat{\alpha}_i^{c_i} = H(m)^{\sum_{\mathfrak{s}_i \in S} c_i SK_i} PK_A^{\sum_{\mathfrak{s}_i \in S} c_i r_i} = H(m)^{SK_0} PK_A^r,$$

and

$$\hat{\beta} = \prod_{\mathfrak{s}_i \in S} \hat{\beta}_i^{c_i} = g^{\sum_{\mathfrak{s}_i \in S} c_i r_i} = g^r,$$

where $r = \sum_{\mathfrak{s}_i \in S} c_i r_i \bmod q$ is also random because all the r_i's are randomly chosen. Furthermore, the reconstructed partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$ can be validated due to the following facts

$$\begin{aligned}\hat{e}(\hat{\alpha}, g) &= \hat{e}\left(H(m)^{SK_0} PK_A^r, g\right) \\ &= \hat{e}\left(H(m)^{SK_0}, g\right) \cdot \hat{e}\left(PK_A^r, g\right) \\ &= \hat{e}\left(H(m), g^{SK_0}\right) \cdot \hat{e}\left(PK_A, g^r\right) \\ &= \hat{e}\left(H(m), PK_0\right) \cdot \hat{e}\left(PK_A, \hat{\beta}\right).\end{aligned}$$

Similarly, the full signature σ can be reconstructed from full-signature-fragments $\{\sigma_i \mid \mathfrak{s}_i \in S\}$ as follows

$$\sigma = \prod_{\mathfrak{s}_i \in S} \sigma_i^{c_i} = \prod_{\mathfrak{s}_i \in S} \left(H(m)^{SK_i}\right)^{c_i} = H(m)^{\sum_{\mathfrak{s}_i \in S} c_i SK_i} = H(m)^{SK_0},$$

and its validity is easy to see.

Given a valid partial signature $\hat{\sigma} = (\hat{\alpha}, \hat{\beta})$, the corresponding full signature can be resolved as follows

$$\sigma = \hat{\alpha} / \hat{\beta}^{SK_A} = \frac{H(m)^{SK_0} PK_A^r}{(g^r)^y} = \frac{H(m)^{SK_0} g^{yr}}{g^{yr}} = H(m)^{SK_0},$$

which can also be easily validated. □

4.2. Security analysis

The resolution ambiguity of our protocol is straightforward and the proof is omitted.

Theorem 2. *The proposed Π-OFEDS is resolution ambiguous.*

Theorem 3. *The proposed Π-OFEDS is robust if the union set of any two unauthorized sets does not cover the signer set.*

Proof. Suppose an unauthorized set $\tilde{S} \in \bar{\Pi}$ is controlled by the adversary $\mathcal{A}$. If the union set of any two unauthorized sets cannot cover the signer set, then $\mathcal{S} \setminus \tilde{S} \in \Pi$. In our construction (specifically, by PSVrfy and FSVrfy), all the invalid partial- and full-signature-fragments can be detected before executing the reconstruction algorithms, i.e., PSRCon and FSRCon. Therefore, the partial and full signature can be further successfully reconstructed from the other ones with regard to $\mathcal{S} \setminus \tilde{S}$. □

Theorem 4. *The proposed Π-OFEDS protocol is secure against signers in the random oracle model, assuming the CDH assumption holds.*

Proof. It is sufficient to show that for any given valid message and partial signature pair $(m, \hat{\sigma})$ under the public keys (PK_A, PK_U) of the arbitrator and the signer set, a valid full signature σ for m can always be successfully resolved. In fact, if the Equality (1) holds, then $\sigma = \hat{\alpha} / (\hat{\beta}^{SK_A})$ will satisfy $\hat{e}(\sigma, g) = \hat{e}\left(H(m)^{SK_0}, g\right) = \hat{e}\left(H(m), PK_0\right)$. □

Theorem 5. *The proposed Π-OFEDS protocol is secure against verifier in the random oracle model, assuming the CDH assumption holds.*

Proof. Suppose there is an adversary $\mathcal{A}$ who controls the verifier and an unauthorized signer set $\tilde{S} \in \bar{\Pi}$, and breaks the proposed protocol. We will construct an algorithm $\mathcal{E}$ to solve the CDH problem on $\mathbb{G}$ by interacting with $\mathcal{A}$. Specifically, the algorithm $\mathcal{E}$ is given a random CDH problem instance (g^μ, g^ν) and manages to output $g^{\mu\nu}$. In the following discussions, we assume that for any message m, a hash query should be requested by $\mathcal{A}$ before it is involved in any other queries. Also, for a forgery $(m, \hat{\sigma})$ or (m, σ) outputted by $\mathcal{A}$, a hash for m should have been asked before.

We distinguish two types of adversaries,

- Type-1 adversary $\mathcal{A}_1$, who will output a forgery (m, σ) where m has been requested for partial-signature-fragments;
- Type-2 adversary $\mathcal{A}_2$, who will output a forgery (m, σ) where m has not been requested for partial-signature-fragments.

Type-1 adversary $\mathcal{A}_1$. We construct $\mathcal{E}_1$ to solve the given CDH instance by interacting with $\mathcal{A}_1$.

Setup: When receiving a corrupted set $\tilde{S}$ and an access structure Π represented by MSP from $\mathcal{A}_1$, the algorithm $\mathcal{E}_1$ responds with the corresponding parameters as follows. The algorithm $\mathcal{E}_1$ picks $y \in_R \mathbb{Z}_q^*$, and sets $PK_A = g^{y\nu}$ and $PK_0 = g^\mu$. Since $\bar{\Pi}$ is monotone decreasing, there exists a maximal unauthorized set $\hat{S} \in \bar{\Pi}_X$ such that $\tilde{S} \subseteq \hat{S}$. To calculate the secret-key-shares for the signers in $\hat{S}$, the algorithm $\mathcal{E}_1$ first picks $\vec{w}' \in_R (\mathbb{Z}_q)^b$, and then computes

$$SK_i = \vec{w}' \cdot \vec{v}_i \bmod q \quad \text{and} \quad PK_i = g^{SK_i}$$

for each signer $\mathfrak{s}_i \in \hat{S}$. Furthermore, for each signer $\mathfrak{s}_j \in \mathcal{S} \setminus \hat{S}$, we know $\hat{S} \cup \{\mathfrak{s}_j\} \in \Pi$, that is, $\hat{S} \cup \{\mathfrak{s}_j\}$ is accepted by MSP. Accordingly, $\vec{v}_j$ can be linearly combined by $\vec{\tau}$ and $\{\vec{v}_i \mid \mathfrak{s}_i \in \hat{S}\}$ as follows

$$\vec{v}_j = c_0 \vec{\tau} + \sum_{\mathfrak{s}_i \in \hat{S}} c_i \vec{v}_i \bmod q,$$

where all the coefficients are over $\mathbb{Z}_q$ and can be found by solving the system of linear equations. Therefore, for each signer $\mathfrak{s}_j \in \mathcal{S} \setminus \hat{S}$, the algorithm $\mathcal{E}_1$ can calculate

$$PK_j = PK_0^{c_0} \cdot \prod_{\mathfrak{s}_i \in \hat{S}} PK_i^{c_i}.$$

In this way, the algorithm $\mathcal{E}_1$ obtains $PK_U = (PK_0, \ldots, PK_n)$. At last, the algorithm $\mathcal{E}_1$ gives $\mathcal{A}_1$ the tuple $(PK_A, PK_U, \{SK_i \mid \mathfrak{s}_i \in \tilde{S}\})$. Due to the perfectness of MSP, SK_0 will not be leaked to $\mathcal{A}_1$.

Queries: The adversary $\mathcal{A}_1$ can adaptively submit the following queries to $\mathcal{E}_1$. To respond, the algorithm $\mathcal{E}_1$ should maintain a list of messages on which $\mathcal{A}_1$ has been requested in each of the following queries.

- Hash queries: Without loss of generality, we assume that $\mathcal{A}_1$ asks for exactly q_H such queries in total and $\mathcal{E}_1$ can choose a value $\hbar \in_R [1, q_H]$. To respond a hash query on message m_i, if $m_i \neq m_\hbar$, the algorithm $\mathcal{E}_1$ sends $H(m_i) = g^{\theta_i}$ to $\mathcal{A}_1$, where $\theta_i \in_R \mathbb{Z}_q^*$; otherwise, the algorithm $\mathcal{E}_1$ sends back $H(m_i) = g^\nu$.
- PSGen queries: The adversary $\mathcal{A}_1$ asks for a partial-signature-fragment on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}_1$ responds with $(PK_j^{\theta_\ell} PK_A^r, g^r)$, where $r \in_R \mathbb{Z}_q^*$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}_1$ picks $r \in_R \mathbb{Z}_q^*$ and returns a partial-signature-fragment as follows
  - if $\mathfrak{s}_j \in \hat{S} \setminus \tilde{S}$, then $\mathcal{E}_1$ responds with $\left((g^\nu)^{SK_j} PK_A^r, g^r\right)$;

– if $\mathfrak{s}_j \in \mathcal{S} \setminus \hat{S}$, then $\mathcal{E}_1$ returns
$$\left((g^{\nu})^{\sum_{\mathfrak{s}_i \in \hat{S}} c_i SK_i + yr}, (g^{\mu})^{-c_0 y^{-1}} g^{r}\right),$$
where c_i's are the same as we have mentioned above and $-c_0 y^{-1}$ is computed over $\mathbb{Z}_q$;

- FSGen queries: The adversary $\mathcal{A}_1$ asks for a full-signature-fragment query on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}_1$ responds with $PK_j^{\theta_\ell}$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}_1$ aborts.
- Res queries: The adversary $\mathcal{A}_1$ asks for a resolution query on the pair $(m, \hat{\sigma} = (\hat{\alpha}, \hat{\beta}))$. If $\hat{\sigma}$ is invalid for m according to Equality (1), then $\mathcal{E}_1$ responds with $\perp$. Otherwise, if m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}_1$ responds with $PK_0^{\theta_\ell}$; if $m = m_\hbar$, the algorithm $\mathcal{E}_1$ aborts.

`Outputs`: Finally, the adversary $\mathcal{A}_1$ outputs a pair (m, σ) of message and full signature to $\mathcal{E}_1$. Since m must have been requested for a hash value, we have $\Pr[m = m_\hbar] = 1/q_H$. This implies that $\mathcal{E}_1$ can succeed in outputting σ as the solution for the given CDH instance with probability $1/q_H$, which is due to

$$\begin{aligned}\hat{e}(\sigma, g) &= \hat{e}\big(H(m), PK_0\big) = \hat{e}\big(H(m_\hbar), PK_0\big)\\ &= \hat{e}(g^{\nu}, g^{\mu}) = \hat{e}(g, g)^{\mu\nu}.\end{aligned}$$

Type-2 adversary $\mathcal{A}_2$. We construct $\mathcal{E}_2$ to solve the given CDH instance by interacting with $\mathcal{A}_2$.

`Setup`: When receiving a corrupted set $\tilde{S}$ and an access structure Π which is represented by MSP from $\mathcal{A}_2$, the algorithm $\mathcal{E}_2$ responds with the corresponding parameters as follows. The algorithm $\mathcal{E}_2$ picks $y \in_R \mathbb{Z}_q^*$ as SK_A, and sets $PK_A = g^y$ and $PK_0 = g^{\mu}$. The algorithm $\mathcal{E}_2$ then proceeds in the same way as `Setup` with regard to $\mathcal{A}_1$, and at last gives $\mathcal{A}_2$ the tuple $(PK_A, PK_U, \{SK_i \mid \mathfrak{s}_i \in \tilde{S}\})$. Due to the perfectness of LSSS defined by MSP, the secret key SK_0 is uniformly random to $\mathcal{A}_2$.

`Queries`: The adversary $\mathcal{A}_2$ can adaptively submit the following queries to $\mathcal{E}_2$. To respond, the algorithm $\mathcal{E}_2$ maintains a list of messages on which $\mathcal{A}_2$ has been requested in each of the following queries.

- Hash queries: Assume $\mathcal{A}_2$ asks for exactly q_H such queries and $\mathcal{E}_2$ chooses $\hbar \in_R [1, q_H]$. To respond a hash query on message m_i, if $m_i \neq m_\hbar$, the algorithm $\mathcal{E}_2$ sends $H(m_i) = g^{\theta_i}$ to $\mathcal{A}_2$, where $\theta_i \in_R \mathbb{Z}_q^*$; otherwise, the algorithm $\mathcal{E}_2$ sends back $H(m_i) = g^{\nu}$.
- PSGen queries: The adversary $\mathcal{A}_2$ asks for a partial-signature-fragment on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}_2$ responds with $(PK_j^{\theta_\ell} PK_A^{r}, g^{r})$, where $r \in_R \mathbb{Z}_q^*$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}_2$ aborts.
- FSGen queries: The adversary $\mathcal{A}_2$ asks for a full-signature-fragment query on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}_2$ responds with $PK_j^{\theta_\ell}$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}_2$ aborts.
- Res queries: The adversary $\mathcal{A}_2$ asks for a full-signature resolution query on a pair $(m, \hat{\sigma} = (\hat{\alpha}, \hat{\beta}))$. If $\hat{\sigma}$ is a valid partial signature of m according to Equality (1), then $\mathcal{E}_1$ responds with $\hat{\alpha}/(\hat{\beta})^{y}$; otherwise, the algorithm $\mathcal{E}_2$ responds with $\perp$.

`Outputs`: When responding for a Res query, the algorithm $\mathcal{E}_2$ can succeed in outputting $\sigma = \hat{\alpha}/(\hat{\beta})^{y}$ as the solution for the given CDH instance with probability $1/q_H$. Since each message m in Res queries has been requested for a hash value before, we have $\Pr[m = m_\hbar] = 1/q_H$. This indicates $\Pr[\sigma = g^{\mu\nu}] = 1/q_H$ due to the following facts

$$\begin{aligned}\hat{e}(\sigma, g) &= \hat{e}\left(\hat{\alpha}/(\hat{\beta})^{y}, g\right)\\ &= \frac{\hat{e}(\hat{\alpha}, g)}{\hat{e}\left((\hat{\beta})^{y}, g\right)} = \frac{\hat{e}\left(H(m), PK_0\right)\hat{e}(PK_A, \hat{\beta})}{\hat{e}\left((\hat{\beta})^{y}, g\right)}\\ &= \hat{e}\left(H(m_\hbar), PK_0\right) = \hat{e}(g^{\nu}, g^{\mu}) = \hat{e}(g, g)^{\mu\nu}.\end{aligned}$$

This completes the proof. □

Theorem 6. *The proposed Π-OFEDS protocol is secure against arbitrator in the random oracle model, assuming the CDH assumption holds.*

Proof. Suppose there is an adversary $\mathcal{A}$ who controls the arbitrator and an unauthorized set $\tilde{S} \in \bar{\Pi}$ of signers, and breaks the proposed protocol. We will construct an algorithm $\mathcal{E}$ to solve the CDH problem on $\mathbb{G}$ by interacting with $\mathcal{A}$. That is, the algorithm $\mathcal{E}$ is given a random CDH problem instance (g^{μ}, g^{ν}) and is required to output $g^{\mu\nu}$. The following discussions also assume that a hash query for any message m should be requested by $\mathcal{A}$ before any other queries. For a forgery (m, σ) outputted by $\mathcal{A}$, a hash for m must have been requested before.

`Setup`: After received a corrupted set $\tilde{S}$ and an access structure Π represented by MSP from $\mathcal{A}$, the algorithm $\mathcal{E}$ proceeds as follows to respond with parameters. The algorithm $\mathcal{E}$ picks $y \in_R \mathbb{Z}_q^*$, and sets $SK_A = y$, $PK_A = g^{y}$ and $PK_0 = g^{\mu}$. All the other parameters such as SK_i ($\mathfrak{s}_i \in \hat{S}$), PK_0 and PK_i ($\mathfrak{s}_i \in \mathcal{S}$) are generated in the same way as `Setup` in the proof of Theorem 5. At last, the algorithm $\mathcal{E}$ gives $\mathcal{A}$ the tuple $(PK_A, SK_A, PK_U, \{SK_i \mid \mathfrak{s}_i \in \tilde{S}\})$. Due to the perfectness of LSSS defined by MSP, the secret key SK_0 is uniformly random to $\mathcal{A}$.

`Queries`: The adversary $\mathcal{A}$ can adaptively submit the following queries to $\mathcal{E}$. To respond, the algorithm $\mathcal{E}$ maintains a list of messages on which $\mathcal{A}$ has been requested in each of the following queries.

- Hash queries: Assume that $\mathcal{A}$ asks for exactly q_H such queries and $\mathcal{E}$ chooses $\hbar \in_R [1, q_H]$. To respond a hash query on message m_i, if $m_i \neq m_\hbar$, the algorithm $\mathcal{E}$ sends $H(m_i) = g^{\theta_i}$ to $\mathcal{A}$, where $\theta_i \in_R \mathbb{Z}_q^*$; otherwise, the algorithm $\mathcal{E}$ sends back $H(m_i) = g^{\nu}$.
- PSGen queries: The adversary $\mathcal{A}$ asks for a partial-signature-fragment on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}$ responds with $(PK_j^{\theta_\ell} PK_A^{r}, g^{r})$, where $r \in_R \mathbb{Z}_q^*$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}$ aborts.
- FSGen queries: The adversary $\mathcal{A}$ asks for a full-signature-fragment query on message m of the signer $\mathfrak{s}_j \in \mathcal{S} \setminus \tilde{S}$. If m has been queried as m_ℓ ($\ell \neq \hbar$), then $\mathcal{E}$ responds with $PK_j^{\theta_\ell}$; otherwise, i.e., $m = m_\hbar$, the algorithm $\mathcal{E}$ aborts.

`Outputs`: Finally, the adversary $\mathcal{A}$ outputs a pair (m, σ) of message and full signature to $\mathcal{E}$. As m has been asked for a hash value, we have $\Pr[m = m_\hbar] = 1/q_H$. Note that when $m = m_\hbar$, we get $\hat{e}(\sigma, g) = \hat{e}(H(m), PK_0) = \hat{e}(H(m_\hbar), PK_0) = \hat{e}(g^{\nu}, g^{\mu}) = \hat{e}(g, g)^{\mu\nu}$, which implies that $\mathcal{E}$ can succeed in outputting σ as the solution for the given CDH instance with probability $1/q_H$. □

4.3. Performance analysis

We analyze the efficiency of our OFEDS protocol. The results are summarized in Tables 4 and 5, where n is the cardinality of the signer set $\mathcal{S}$. Table 4 illustrates the performance of our instantiated OFEDS protocol in terms of element sizes of arbitrator's public key PK_A and private key SK_A, signer set's public key PK_U, signer's secret-key-share SK_i, partial-signature $\hat{\sigma}$ (-fragment $\hat{\sigma}_i$) and full-signature

Table 4
Performance of OFEDS in terms of the element size.

	PK_A	SK_A	PK_U	SK_i	$\hat{\sigma}_i, \hat{\sigma}$	σ_i, σ
Element size	$1s_G$	$1s_q$	$(n+1)s_G$	$1s_q$	$2s_G$	$1s_G$

Table 5
Computation costs of OFEDS.

Algorithm		Computation costs
AKGen		1E
UKGen		$(n+1)(b\mathtt{M}+(b-1)\mathtt{A}+\mathtt{E})$
PSign	PSGen	$1\mathtt{h}+3\mathtt{E}+1\mathtt{M}$
	PSVrfy	$1\mathtt{h}+3\mathtt{P}+1\mathtt{M}$
	PSRCon	$1\mathtt{L}+2\|S\|\mathtt{E}+2(\|S\|-1)\mathtt{M}$
	PVrfy	$1\mathtt{h}+3\mathtt{P}+1\mathtt{M}$
FSign	FSGen	$1\mathtt{h}+1\mathtt{E}$
	FSVrfy	$1\mathtt{h}+2\mathtt{P}$
	FSRCon	$1\mathtt{L}+\|S\|\mathtt{E}+(\|S\|-1)\mathtt{M}$
	FVrfy	$1\mathtt{h}+2\mathtt{P}$
Res		$1\mathtt{h}+3\mathtt{P}+1\mathtt{E}+2\mathtt{M}$

σ (-fragment σ_i). The element sizes of bilinear group $\mathbb{G}$ and finite field $\mathbb{Z}_q$ are denoted by s_G and s_q, respectively. It can be seen that only the signer set's public key PK_U whose size is linear with the number of signers, while all the other parameters contain constant elements. Therefore, the proposed OFEDS protocol is efficient in storage and communication.

Table 5 summarizes the computation costs of the proposed OFEDS scheme. In the table, we use the notations $\mathtt{h}$, $\mathtt{L}$, $\mathtt{A}$, $\mathtt{M}$, $\mathtt{E}$, $\mathtt{P}$ to denote a hash evaluation of H, the costs for solving the system of equations, one addition, one multiplication, one exponentiation and one bilinear pairing, respectively. We also use b and $|S|$ to represent the column of MSP matrix and the cardinality of an authorized set S, respectively. It is easy to see that only the user-key-generation-algorithm UKGen whose computation costs are associated with the signer number of set $\mathcal{S}$. Note that UKGen can be executed in advance, i.e., before exchanging exact items. Therefore, the proposed OFEDS protocol enjoys desirable online performance and is practical in real applications.

We proceed to evaluate experimental performance of our OFEDS protocol. The experiments are conducted in C programming language by employing Pairing Based Cryptography library (PBC,[1] http://crypto.stanford.edu/pbc/), and are carried out on a laptop with Intel(R) Core(TM) i5-5200U CPU @ 2.20 GHz and 4GB RAM. The elliptic curve is of type $y^2 = x^3 + x$ with $|q| = 160$ bits and $s_G = 512$ bits. It is assumed that there are in total 30 signers, i.e., $n = 30$. It seems that 30 is a reasonable bound of signer number for most application cases. In the experiment, the arbitrator and each signer first publish the corresponding public keys, and keep private key or private key shares secretly. Then, the signers will work jointly according to a pre-defined access structure to exchange a randomly chosen message with a verifier. That is, they commit to their item by sending partial-signature-fragments to the verifier, who can validate the received information and further reconstruct a partial signature if they are associated to an authorized set. In a similar way, the signers generate and give the full-signature-fragments to the verifier so that a full signature can be reconstructed from an authorized set. The verifier can also get a valid one by interacting with the arbitrator according to the reconstructed partial signature, especially when the reconstructed full signature is invalided. The detailed performance of each algorithm is elaborated as follows.

[1] PBC library provides sufficient APIs that can be used to implement pairing-based crypto-systems as well as conventional crypto-systems.

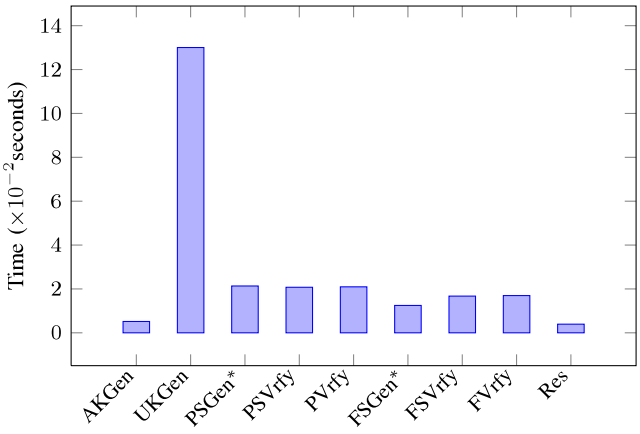

Fig. 2. Performance evaluation of OFEDS with $n = 30$.

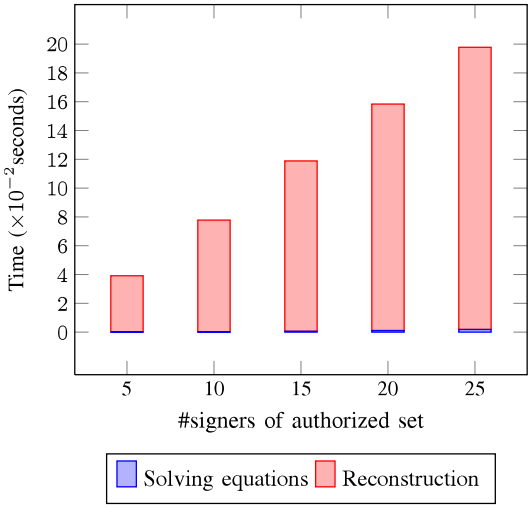

Fig. 3. Partial signature reconstruction ($n = 30$).

The computation costs of the algorithms except PSRCon and FSRCon are shown in Fig. 2. The algorithm AKGen requires one exponentiation on $\mathbb{G}$ regardless the number of signers, which takes roughly 5.2 ms. Although UKGen needs 130ms, the average generation time of one pair of public/private keys is much more efficient, i.e., just 4.19 ms. There is only one hash evaluation and one exponentiation of FSGen. It takes about half time of running PSGen to generate one signature fragment. The algorithms PSVrfy and PVrfy take roughly equal time. The similar case happens to the algorithms FSVrfy and FVrfy. When Res resolves a full-signature, it takes no more than 4ms in our experiment. Both algorithms PSRCon and FSRCon are determined by the cardinality of the authorized signer set. In fact, each authorized set should hold at least b rows of MSP matrix when reconstructing partial-/full-signature. We consider several cases with $b = 5, 10, 15, 20, 25$, the experimental results are shown in Figs. 3 and 4, respectively. Note that during partial-/full-signature reconstruction, a system of linear equations over $\mathbb{Z}_q$ should be solved before carrying out real reconstruction procedure. It can be seen that each case in Fig. 4 is about twice as efficient as its counterpart in Fig. 3. This is coincident with the theoretical analyses.

5. Conclusion

In this paper, we introduced the notion of optimistic fair exchange of distributed signatures. This primitive allows two groups to fairly exchange digital items in a way such that their

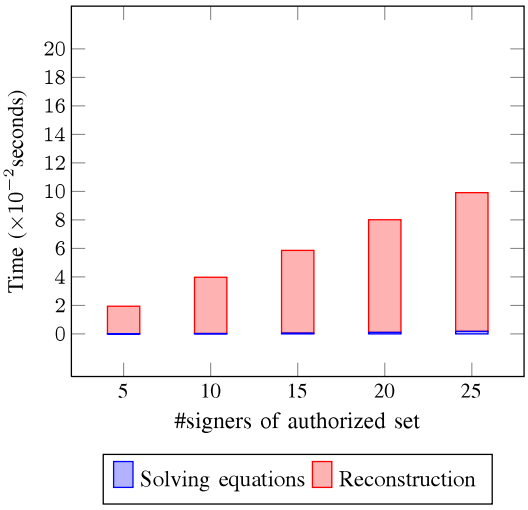

Fig. 4. Full signature reconstruction ($n = 30$).

duties are jointly carried out by authorized sets of respective group members. We formalized the security model for OFEDS which captures the standard security requirements of existing OFE as well as robustness to ensure the sender's duty can be successfully performed even when there are some dishonest signers. Our OFEDS extends OFE of threshold signatures to a more general case and thus supports more complicated applications. We also proposed a CDH-based OFEDS construction and proved its security under standard assumptions in the random model. Our theoretical and experimental performance analyses further show that the proposed OFEDS construction has reasonable efficiency to support real-world applications.

Acknowledgments

This paper is partially supported by the National Key Basic Research Program (973 program) through project 2012CB315905, by the Natural Science Foundation of China through projects 61370190, 61532021, 61272501, 61402029, 61472429 and 61202465, by the Beijing Natural Science Foundation through project 4132056, by the Guangxi natural science foundation through project 2013GXNSFBB053005, the Innovation Fund of China Aerospace Science and Technology Corporation, Satellite Application Research Institute through project 2014-CXJJ-TX-10.

References

[1] N. Asokan, M. Schunter, M. Waidner, Optimistic Protocols for Fair Exchange, ACM, New York, NY, USA, 1997, pp. 7–17.
[2] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps, in: E. Biham (Ed.), EUROCRYPT 2003, Springer, 2003, pp. 416–432.
[3] S. Kim, S. Kim, G. Lee, Secure verifiable non-interactive oblivious transfer protocol using RSA and Bit commitment on distributed environment, Future Gener. Comput. Syst. 25 (2009) 352–357.
[4] Q. Huang, D.S. Wong, W. Susilo, Group-oriented fair exchange of signatures, Inform. Sci. 181 (2011) 3267–3283.
[5] L. Qu, G. Wang, Y. Mu, Optimistic fair exchange of ring signatures, in: M. Rajarajan, F. Piper, H. Wang, G. Kesidis (Eds.), Security and Privacy in Communication Networks, Springer, 2012, pp. 227–242.
[6] Y. Wang, M. Au, J. Liu, T. Yuen, W. Susilo, Threshold-oriented optimistic fair exchange, in: J. Lopez, X. Huang, R. Sandhu (Eds.), Network and System Security, Springer, 2013, pp. 424–438.
[7] A. Beimel, Secret-sharing schemes: A survey, in: Y. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, C. Xing (Eds.), Coding and Cryptology, Springer, 2011, pp. 11–46.
[8] Y. Wang, Q. Wu, D.S. Wong, B. Qin, Y. Mu, J. Liu, Further ideal multipartite access structures from integer polymatroids, Sci. China Inf. Sci. 58 (2015) 1–13.
[9] J. Zhang, F. Zhang, Information-theoretical secure verifiable secret sharing with vector space access structures over bilinear groups and its applications, Future Gener. Comput. Syst. 52 (2015) 109–115.
[10] J. Herranz, C. Padró, G. Sáez, Distributed rsa signature schemes for general access structures, in: C. Boyd, W. Mao (Eds.), Information Security, Springer, 2003, pp. 122–136.
[11] J. Herranz, G. Sáez, Verifiable secret sharing for general access structures, with application to fully distributed proxy signatures, in: R.N. Wright (Ed.), Financial Cryptography, Springer, 2003, pp. 286–302.
[12] I. Damgård, R. Thorbek, Linear integer secret sharing and distributed exponentiation, in: M. Yung, Y. Dodis, A. Kiayias, T. Malkin (Eds.), PKC 2006, Springer, 2006, pp. 75–90.
[13] Y. Wang, D.S. Wong, Q. Wu, S.S.M. Chow, B. Qin, J. Liu, Practical distributed signatures in the standard model, in: J. Benaloh (Ed.), CT-RSA 2014, Springer, 2014, pp. 307–326.
[14] Y. Wang, D.S. Wong, Q. Wu, S.S.M. Chow, B. Qin, J. Liu, Y. Ding, Practical (fully) distributed signatures provably secure in the standard model, Theoret. Comput. Sci. 595 (2015) 143–158.
[15] S. Chang, D.S. Wong, Y. Mu, Z. Zhang, Certificateless threshold ring signature, Inform. Sci. 179 (2009) 3685–3696.
[16] H. Yuan, F. Zhang, X. Huang, Y. Mu, W. Susilo, L. Zhang, Certificateless threshold signature scheme from bilinear maps, Inform. Sci. 180 (2010) 4714–4728.
[17] B. Qin, Q. Wu, L. Zhang, O. Farràs, J. Domingo-Ferrer, Provably secure threshold public-key encryption with adaptive security and short ciphertexts, Inform. Sci. 210 (2012) 67–80.
[18] H. Xiong, F. Li, Z. Qin, Certificateless threshold signature secure in the standard model, Inform. Sci. 237 (2013) 73–81.
[19] Y. Wang, Q. Wu, D.S. Wong, B. Qin, J. Liu, J. Mao, Optimistic fair exchange of distributed signatures, in: CSC 2014, IET, 2014, pp. 85–90.
[20] X. Chen, J. Li, J. Ma, W. Lou, D.S. Wong, New and efficient conditional e-payment systems with transferability, Future Gener. Comput. Syst. 37 (2014) 252–258.
[21] W. Wang, P. Xu, L.T. Yang, H. Li, A design for cloud-assisted fair-play management system of online contests with provable security, Future Gener. Comput. Syst. 52 (2015) 137–146.
[22] Q. Huang, G. Yang, D.S. Wong, W. Susilo, Ambiguous optimistic fair exchange, in: J. Pieprzyk (Ed.), ASIACRYPT 2008, Springer, 2008, pp. 74–89.
[23] Q. Huang, D.S. Wong, W. Susilo, The construction of ambiguous optimistic fair exchange from designated confirmer signature without random oracles, Inform. Sci. 228 (2013) 222–238.
[24] Y. Wang, M.H. Au, W. Susilo, Perfect ambiguous optimistic fair exchange, in: T.W. Chim, T.H. Yuen (Eds.), Information and Communications Security, Springer, 2012, pp. 142–153.
[25] Q. Huang, D.S. Wong, W. Susilo, P^2OFE: Privacy-preserving optimistic fair exchange of digital signatures, in: J. Benaloh (Ed.), CT-RSA 2014, Springer, 2014, pp. 367–384.
[26] L. Zhang, Q. Wu, B. Qin, Identity-based verifiably encrypted signatures without random oracles, in: J. Pieprzyk, F. Zhang (Eds.), Provable Security, Springer, 2009, pp. 76–89.
[27] L. Zhang, Q. Wu, B. Qin, Identity-based optimistic fair exchange in the standard model, Secur. Commun. Netw. 6 (2013) 1010–1020.
[28] N. Asokan, M. Schunter, M. Waidner, Optimistic Protocols for Multi-Party Fair Exchange, Technical Report IBM Research Report RZ 2892, IBM Zürich Research Laboratory, Zürich, 1996.
[29] M. Franklin, G. Tsudik, Secure group barter: Multi-party fair exchange with semi-trusted neutral parties, in: R. Hirchfeld (Ed.), Financial Cryptography, Springer, 1998, pp. 90–102.
[30] I. Khill, J. Kim, I. Han, J. Ryou, Multi-party fair exchange protocol using ring architecture model, Comput. Secur. 20 (2001) 422–439.
[31] F. Bao, R. Deng, K.Q. Nguyen, V. Varadharajan, Multi-party fair exchange with an off-line trusted neutral party, in: Tenth International Workshop on Database and Expert Systems Applications, Proceedings, IEEE, 1999, pp. 858–862.
[32] N. González-Deleito, O. Markowitch, An optimistic multi-party fair exchange protocol with reduced trust requirements, in: K. Kim (Ed.), ICISC 2001, Springer, 2002, pp. 258–267.
[33] A. Mukhamedov, S. Kremer, E. Ritter, Analysis of a multi-party fair exchange protocol and formal proof of correctness in the strand space model, in: A. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security, Springer, 2005, pp. 255–269.
[34] N. González-Deleito, O. Markowitch, Exclusion-freeness in multi-party exchange protocols, in: A.H. Chan, V. Gligor (Eds.), Information Security, Springer, 2002, pp. 200–209.
[35] N. González-Deleito, O. Markowitch, Exclusions and related trust relationships in multi-party fair exchange protocols, Electron. Commer. Res. Appl. 6 (2007) 343–357.
[36] Y. Dodis, P. Lee, D. Yum, Optimistic fair exchange in a multi-user setting, in: T. Okamoto, X. Wang (Eds.), PKC 2007, Springer, 2007, pp. 118–133.

[37] Q. Huang, G. Yang, D.S. Wong, W. Susilo, Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles, in: T. Malkin (Ed.), CT-RSA 2008, Springer, 2008, pp. 106–120.
[38] X. Huang, Y. Mu, W. Susilo, W. Wu, Y. Xiang, Further observations on optimistic fair exchange protocols in the multi-user setting, in: P. Nguyen, D. Pointcheval (Eds.), PKC 2010, Springer, 2010, pp. 124–141.
[39] A. Küpçü, A. Lysyanskaya, Optimistic fair exchange with multiple arbiters, in: D. Gritzalis, B. Preneel, M. Theoharidou (Eds.), ESORICS 2010, Springer, 2010, pp. 488–507.
[40] D. Boneh, X. Boyen, Short signatures without random oracles and the sdh assumption in bilinear groups, J. Cryptol. 21 (2008) 149–177.
[41] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, J. Cryptol. 17 (2004) 297–319.
[42] G.R. Blakley, Safeguarding cryptographic keys, in: International Workshop on Managing Requirements Knowledge, IEEE Computer Society, pp. 313–317.
[43] A. Shamir, How to share a secret, Commun. ACM 22 (1979) 612–613.
[44] G.J. Simmons, W. Jackson, K. Martin, The geometry of shared secret schemes, Bull. ICA 1 (1991) 230–236.
[45] M. Karchmer, A. Wigderson, On span programs, in: Proceedings of the Eighth Annual Structure in Complexity Theory Conference, 1993, pp. 102–111.
[46] E.F. Brickell, Some ideal secret sharing schemes, in: J.J. Quisquater, J. Vandewalle (Eds.), EUROCRYPT'89, Springer, 1990, pp. 468–475.