University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2013

Threshold-oriented optimistic fair exchange

Yang Wang

University of Wollongong, ywang@uow.edu.au

Man Ho Allen Au

University of Wollongong, aau@uow.edu.au

Joseph K. Liu

Institute for Infocomm Research

Tsz Hon Yuen

University of Hong Kong, thy738@uow.edu.au

Willy Susilo

University of Wollongong, wsusilo@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/eispapers>

Part of the [Engineering Commons](#), and the [Science and Technology Studies Commons](#)

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Threshold-oriented optimistic fair exchange

Abstract

Fair exchange protocol aims to allow two parties to exchange digital items in a fair manner. It is well-known that fairness can only be achieved with the help of a trusted third party, usually referred to as arbitrator. A fair exchange protocol is optimistic if the arbitrator is not involved in the normal execution of the fair exchange process. That is, its presence is necessary only when one of the exchanging parties is dishonest. Traditionally, the items being exchanged are digital signatures. In this paper, we consider the items to be threshold signatures. Specifically, the signatures are created by a subset of legitimate signers instead of a single signer. We define a security model for this new notion, and provide an concrete instantiation. Our instantiation can be proven secure in the random oracle model. Our definition covers the case when the item being exchanged is a secret key of an identity-based encryption where the master secret key is split amongst a set of authorities.

Keywords

oriented, fair, exchange, threshold, optimistic

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wang, Y., Au, M., Liu, J. K., Yuen, T. Hon. & Susilo, W. (2013). Threshold-oriented optimistic fair exchange. Lecture Notes in Computer Science, 7873 424-438.

Threshold-Oriented Optimistic Fair Exchange

Yang Wang¹, Man Ho Au¹, Joseph K. Liu², Tsz Hon Yuen³, and Willy Susilo^{1*}

¹ Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Australia

² Cryptography and Security Department, Institute for Infocomm Research, Singapore

³ Department of Computer Science, University of Hong Kong, Hong Kong
{yw990}@uowmail.edu.au, aau@uow.edu.au, ksliu@i2r.a-star.edu.sg,
johnyuenhk@gmail.com, wsusilo@uow.edu.au

Abstract. Fair exchange protocol aims to allow two parties to exchange digital items in a fair manner. It is well-known that fairness can only be achieved with the help of a trusted third party, usually referred to as arbitrator. A fair exchange protocol is optimistic if the arbitrator is not involved in the normal execution of the fair exchange process. That is, its presence is necessary only when one of the exchanging parties is dishonest. Traditionally, the items being exchanged are digital signatures. In this paper, we consider the items to be threshold signatures. Specifically, the signatures are created by a subset of legitimate signers instead of a single signer. We define a security model for this new notion, and provide an concrete instantiation. Our instantiation can be proven secure in the random oracle model. Our definition covers the case when the item being exchanged is a secret key of an identity-based encryption where the master secret key is split amongst a set of authorities.

1 Introduction

Optimistic fair exchange (OFE), first introduced by Asokan, Schunter and Waidner [1], is a kind of protocols aiming to guarantee fairness for two parties exchanging digital items. In OFE, a trusted third party named “arbitrator” is needed but only involved when there is a dispute between the participants. Traditionally the digital items of interest are digital signatures and the optimistic exchange of digital signatures constitutes an important part of any business transaction. Typically such a protocol comprises three message flows. First Alice the signer initiates the exchange by sending a partial signature to the receiver, say Bob. The partial signature serves as a commitment assuring Bob of Alice’s full signature at the end of the protocol. After verifying the validity of Alice’s partial signature, Bob sends its full signature to Alice in the second message flow. Later, Alice should send her full signature back to Bob and complete the exchange. In the case there is a network failure or Alice attempts to cheat by refusing to send

* This work is supported by ARC Future Fellowship FT0991397.

her own full signature, Bob can ask the arbitrator to make a resolution with Alice's partial signature and his own full signature. In this case the arbitrator will convert Alice's partial signature into a full one and send it back to Bob. Note that at the end of this exchange, either both Alice and Bob gain the other's full signature, or neither does. Thus the exchange is fair.

1.1 Related Work

As a useful tool in applications such as contract signing, electronic commerce and even peer-to-peer file sharing, OFE has been extensively researched since its introduction. There are several approaches in the construction of OFE, including schemes based on verifiably encrypted signatures [2, 6, 5, 16, 21, 19], and sequentially two-party multisignatures [8]. It was further showed that OFE can be constructed from OR signature [7], and conventional signatures and ring signatures [12]. Some desirable properties such as setup-free [22], stand-alone [22], abuse-free [9], signer ambiguity [11], resolution ambiguity [17] and accountability [13] are proposed in literatures as well.

In [3] and [15], OFE employing multiple arbitrators are discussed to reduce the trust placed on the single arbitrator. Unfortunately, the existing techniques are either expensive or rely on synchronized clocks, which is undesirable as achieving synchronization in a peer-to-peer setting in which the arbitrators do not even know each other is hard.

Most of the previous works on OFE are done in the individual setting, in which the two involving parties are individual users and they represents themselves. An interesting scenario in OFE is that either party consists of a group of users. In such a scenario, every single user in the group can execute transactions with another party on behalf of its group, but the actual signer's identity should be protected from outsiders. In [18], the authors employ a ring signature such that all the users' public keys are involved in the ring to hide the actual signer's identity and at the same time ensure that each signer can sign on behalf on the group. Later, optimistic fair exchange of group signatures is considered in [10]. The difference between optimistic fair exchange of ring signatures and that of group signatures is similar with the difference between ring signatures and group signatures. There are some subtle difference though. For instance, the user in ring signature setting can choose its key pair by himself while in the group signature setting, a group manager is responsible to the generation of a user's key pair. Furthermore, optimistic fair exchange of group signatures has the additional feature that the anonymity of a signer can be revoked by its group manager. One common characteristic that the above two constructions shares is that a single user in the group represents the whole group.

To the best of our knowledge, there is no previous work on OFE discussing about the scenario that only a least number of users together can represent the group. That is, for a group of n users, only at least t users of them together can sign on behalf of the group and make exchanges with other parties. We introduce the notion of *threshold-oriented optimistic fair exchange* (TOFE), which in essence is optimistic fair exchange of threshold signatures. In the exchange

outsiders have no idea which subset of the users actually participated in the generation of a threshold signature. This can be viewed as a natural way to reduce the trust placed on every single user of the group.

Besides, TOFE has other practical applications. For example, consider the case in which two parties intend to exchange a secret key of an identity-based encryption (IBE) [4]. In an identity-based setting, the key generation centre (KGC) is a high value target to adversaries as compromising the master key will break the whole system. Thus the master key is typically split amongst a set of authorities so that only when a threshold of authorities together can create a secret key for an identity [14]. Remember that the secret key of an identity can be viewed as a digital signature on the user's identity from the KGC [4]. Thus, fair exchange of secret key of an identity-based encryption also falls within the model of OFE. In case when the master key is split amongst a set of authorities and two KGCs, perhaps each for a certain geographic location, would like to exchange a secret key of a specific identity, threshold-oriented would be useful.

The table below summarizes the categories of exchanged digital items that has been discussed in the literatures.

Schemes	Digital Items Exchanged
traditional OFE	individual signatures
Qu et al. [18]	ring signatures
Huang et al. [10]	group signatures
Our Scheme	threshold signatures / secret keys of an IBE

Table 1. digital items that are exchanged in OFE

1.2 Contribution.

In this paper, we study optimistic fair exchange in a threshold-oriented setting. Specifically, we present a formal definition for TOFE. We propose a concrete construct and demonstrate that our construction is secure in the random oracle model.

Organization. The rest of the paper is organized as follows. In Section 2, we review notations and technical preliminaries. In Section 3, the syntax of TOFE and its security definitions are presented. We present our construction in Section 4. Finally, we conclude our paper in Section 5. In Appendix ??, we prove the security of our construction under well-known assumptions in the random oracle model.

2 Preliminary

If n is a positive integer, we use $[n]$ to denote the set $\{1, \dots, n\}$. If p is a prime, we use $\mathbb{Z}_p$ to denote the set $\{0, \dots, p-1\}$ and $\mathbb{Z}_p^*$ to denote the set $\{a \mid a \in \mathbb{Z}_p \wedge \gcd(a, p) = 1\}$.

2.1 Bilinear Pairing

Let $\mathbb{G}, \mathbb{G}_T$ be two cyclic groups such that $|\mathbb{G}| = |\mathbb{G}_T| = p$. We say that $\hat{e}$ is a bilinear map if $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ possesses the following properties.

- Elements of $\mathbb{G}, \mathbb{G}_T$ have unique binary representation.
- For all elements of $g, h \in \mathbb{G}, x, y \in \mathbb{Z}_p$, it holds that

$$\hat{e}(g^a, h^b) = \hat{e}(g, h)^{ab}$$

- There exists $g, h \in \mathbb{G}$ such that $\hat{e}(g, h)$ is not the identity element of $\mathbb{G}_T$.

2.2 Number-theoretic Assumptions

We review the following well-known computational assumptions.

Definition 1 (DL Assumption). Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p . The discrete logarithm assumption states that given a tuple $(g, Z) \in (\mathbb{G}, \mathbb{G})$, it is computationally infeasible to compute the value $z \in \mathbb{Z}_p$ such that $Z = g^z$.

Definition 2 (CDH Assumption). Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order p . The computational Diffie-Hellman assumption states that given a tuple $(g, g^a, g^b) \in (\mathbb{G}, \mathbb{G}, \mathbb{G})$, it is computationally infeasible to compute the value g^{ab} .

2.3 Secret Sharing

We review the principle of the well-known Shamir secret sharing scheme [20] here. Roughly speaking, a secret sharing scheme allows a user to divide a value into n pieces, called shares, so that any t share holders together can recover the secret. The major idea is that it takes t points to define a polynomial, say, $f(x)$ of degree $t-1$. One could generate f in such a way that $f(0)$ is secret to be shared. Each share is then a point $(i, f(i))$. Now with t points, one could recover the polynomial and thus the value $f(0)$. On the other hand, with only $t-1$ points, nothing about $f(0)$ would be revealed since there are exponentially many curves that pass through those $t-1$ points.

Preparation Let x be the secret to be shared. Randomly pick a polynomial f of degree $t-1$ such that $f(0) = x$. Each share is defined as $(i, f(i))$ for $i = 1$ to n .

Reconstruction One could make use of Lagrange interpolation to recover the value $f(0)$ when t points are given.

- Let $\mathcal{I}$ be a set such that $|\mathcal{I}| = t$ and that for all $i \in \mathcal{I}$, $f(i)$ is known.
- The Lagrange polynomial interpolation technique states that

$$f(x) := \sum_{i \in \mathcal{I}} f(i) \lambda_i(x),$$

where $\lambda_i(x)$, called the Lagrange basis polynomials, are defined as

$$\lambda_i(x) := \prod_{j \in \mathcal{I} \setminus \{i\}} \frac{x - j}{i - j}.$$

Since we are interested in $f(0)$ in the secret sharing scheme, we use λ_i to denote the value of $\lambda_i(0)$ and referred to it as the Lagrange coefficient.

- Thus, to recover the secret, one first computes the Lagrange coefficient λ_i as

$$\lambda_i := \prod_{j \in \mathcal{I} \setminus \{i\}} \frac{-j}{i - j}.$$

- Then, $f(0)$ can be recovered as

$$f(0) := \sum_{i \in \mathcal{I}} f(i) \lambda_i.$$

3 Definition of TOFE

3.1 Syntax

We adapt the definitions and security models of OFE from various literatures for our TOFE. For efficiency consideration, our definition of TOFE consists of non-interactive algorithms only. That is, we require that no communication between signers is necessary during the partial or full signature generation. The following enumerate the syntax of a construction of TOFE, which consists of seven algorithms. In addition, we adopt the common reference string model.

- *Common Reference String Generation* On input a security parameter 1^k , this algorithm outputs a common reference string param_{CRS} which includes the security parameter 1^k . We assume param_{CRS} is an implicit input to all algorithms described below.
- $(\text{pk}_A, \text{sk}_A) \leftarrow \text{AGen}()$ This algorithm outputs the arbitrator key pairs $(\text{pk}_A, \text{sk}_A)$.
- $(\text{pk}_U, \{\text{sk}_{U,i}\}_{i=1}^n) \leftarrow \text{UGen}(n, t)$ This algorithm takes as input the required number of signers n , the threshold t and output the public key of the user pk_U , together with n secret signing keys for the signers $\text{sk}_{U,i}$.
- $\text{PSign} = (\text{PSign}_{(s)}, \text{PSign}_{(v)}, \text{PSign}_{(g)})$ This is a suite of three algorithms which allows a subset of signers to create a partial signature.
 - $\hat{\sigma}_i \leftarrow \text{PSign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$ On input the public key of the arbitrator pk_A , a message M and a secret signing key of signer i , this algorithm outputs a partial signature share for signer i .

- $valid/invalid \leftarrow \text{PSign}_{(v)}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma}_i, i)$ On input the public key of the arbitrator pk_A and that of the user pk_U , a message M , a partial signature share $\hat{\sigma}_i$ from signer i , this algorithm checks the validity of the partial signature share created by signer i .
- $\hat{\sigma} \leftarrow \text{PSign}_{(g)}(\text{pk}_A, \text{pk}_U, M, \{\hat{\sigma}_i\}_{i \in \mathcal{I}}, \mathcal{I})$ On input the public key of the arbitrator pk_A and that of the user pk_U , a message M , t partial signature shares $\{\hat{\sigma}_i\}$ for $i \in \mathcal{I}$ such that $\mathcal{I} \subset [n]$ and $|\mathcal{I}| = t$, this algorithm outputs a partial signature.
- $valid/invalid \leftarrow \text{PVer}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma})$ This algorithm checks the validity of a partial signature $\hat{\sigma}$ on message M based on the public key of the arbitrator pk_A , the public key of the user pk_U .
- $\text{Sign} = (\text{Sign}_{(s)}, \text{Sign}_{(v)}, \text{Sign}_{(g)})$ Similar to the partial signature generation process, the signing algorithm is also a set of three algorithms which allows a subset of signers to create a signature.
 - $\sigma_i \leftarrow \text{Sign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$ On input public key of the arbitrator pk_A , message M and secret signer key of signer i , this algorithm outputs a signature share for signer i .
 - $valid/invalid \leftarrow \text{Sign}_{(v)}(\text{pk}_A, \text{pk}_U, M, \sigma_i, i)$ This algorithm checks the validity of the signature share σ_i created by signer i based on the public key of the arbitrator pk_A , the public key of the user pk_U and message M .
 - $\sigma \leftarrow \text{Sign}_{(g)}(\text{pk}_A, \text{pk}_U, M, \{\sigma_i\}_{i \in \mathcal{I}}, \mathcal{I})$ On input the public key of the arbitrator pk_A and that of the user pk_U , a message M , t signature shares $\{\sigma_i\}$ for $i \in \mathcal{I}$ such that $\mathcal{I} \subset [n]$ and $|\mathcal{I}| = t$, this algorithm outputs a signature.
- $valid/invalid \leftarrow \text{Ver}(\text{pk}_A, \text{pk}_U, M, \sigma)$ This algorithm checks the validity of a signature σ on message M based on the public key of the arbitrator pk_A and that of the user pk_U .
- $\sigma \leftarrow \text{Res}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma}, \text{sk}_A)$ Given a valid partial signature $\hat{\sigma}$, a message M , public key of the user pk_U , key pair of the arbitrator $(\text{pk}_A, \text{sk}_A)$, this algorithm allows the arbitrator to output a signature on message M . Note that $\perp$ is returned if $invalid \leftarrow \text{PVer}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma})$.

Correctness A construction of TOFE is correct if the following conditions hold:

1. Any partial signature created by any t honest signers using PSign will be valid under PVer .
2. Any signature created by any t honest signers using Sign will be valid under Ver .
3. Any signature created by the arbitrator using Res based on a valid partial signature will be valid under Ver .

Furthermore, it is required that any signature created by the arbitrator using Res based on a valid partial signature will be indistinguishable to the signature created by any t honest signers using Sign .

3.2 A Typical Usage of the TOFE Algorithms

Note that in OFE with three message flows between the initiator Alice and the receiver Bob, the item to be sent by Bob is not restricted to any format. It could be a physical item such as money. For simplicity we assume the item to be sent by Bob is a digital signature. Nonetheless, it could be a ring signature, a group signature or a threshold signature. Below we show how Alice and Bob can conduct an exchange based on our definition of TOFE. Note that the party Alice in TOFE consists of a group of n signers, and only when t -out-of- n signers. An exchange is possible only when t signers agree to participate.

Our definition of TOFE does not require the set of t signers to communicate with each other. Below is a typical usage of our definition of TOFE algorithms.

1. *Partial Signature Shares Collection* Bob approaches each signer independently and the parties agree on the items to be exchanged. The signer, say signer i invokes $\text{PSign}_{(s)}$ and sends the share of the partial signature $\hat{\sigma}_i$ to Bob. Bob uses $\text{PSign}_{(v)}$ to verify the share.
2. *Partial Signature Generation* Upon collecting t partial signature shares, Bob invokes $\text{PSign}_{(g)}$ to generate a partial signature $\hat{\sigma}$. He invokes PVer to ensure its validity.
3. *Obligation Fulfillment* If the partial signature Bob obtained is valid, he fulfill his obligations. In this example, Bob sends his digital signature to all the signers involved.
4. *Signature Shares Collection* Each signer validates that Bob has fulfilled his obligations. In this example, each signer checks that the digital signature sent by Bob is valid. If yes, each signer, say signer i , invokes $\text{Sign}_{(s)}$ and sends the share of the signature σ_i to Bob, who checks its validity with $\text{Sign}_{(v)}$.
5. *Signature Generation* Upon collecting t signature shares, Bob invokes $\text{Sign}_{(g)}$ to generate a signature σ . He invokes Ver to ensure its validity. If yes, the exchange process is completed.
6. *Resolution* Suppose some signers refuse to send their signature shares, or that the signature created in signature generation is invalid, Bob can approach the arbitrator for assistance. Specifically, he approaches the arbitrator proves that he has fulfilled his obligation. After that, Bob submits the valid partial signature $\hat{\sigma}$ to the arbitrator. The arbitrator sends back the signature σ by invoking Res and this completes the exchange.
7. *Remarks* In this example, Bob can send his digital signature to the arbitrator as a proof of obligation fulfillment. Even if Bob is lying, the arbitrator can still give this digital signature to the signers should they also complain and thus the exchange could be completed regardless of what happen afterwards.

3.3 Security Model

Traditionally, any construction of optimistic fair exchange should be secure in three aspects, namely, security against signers, security against verifiers and security against the arbitrator respectively. As suggested by the respective names,

they intend to cover the scenarios when the named party is dishonest. We modify the traditional model in the threshold setting. Specifically, the verifier can collude with $t - 1$ malicious signers in our consideration of security against verifiers.

Security Against Signers This property guarantees that even when all the signers collude together, they cannot create a partial signature that passes the partial signature verification algorithm PVer yet it cannot be resolved into a full signature by the arbitrator. This property intends to protect honest verifiers. Specifically, we use the following three-phase game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ to define this property.

Initialization $\mathcal{A}$ specifies the number of signers n and the threshold t . $\mathcal{C}$ creates the common reference string param_{CRS} and invokes

$$(\text{pk}_A, \text{sk}_A) \leftarrow \text{AGen}(),$$

$$(\text{pk}_U, \{\text{sk}_{U,i}\}_{i=1}^n) \leftarrow \text{UGen}(n, t).$$

$\mathcal{C}$ gives $(\text{param}_{CRS}, \text{pk}_A, \text{pk}_U, \{\text{sk}_{U,i}\}_{i=1}^n)$ to $\mathcal{A}$.

Query $\mathcal{A}$ can adaptively issue the following query to $\mathcal{C}$.

– *Res Query*. $\mathcal{A}$ gives $(\hat{\sigma}, M)$ to $\mathcal{C}$, who invokes

$$\sigma \leftarrow \text{Res}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma}, \text{sk}_A)$$

and returns σ to $\mathcal{A}$.

End-Game $\mathcal{A}$ submits $(M^*, \hat{\sigma}^*)$ and wins the game if

$$\text{valid} \leftarrow \text{PVer}(\text{pk}_A, \text{pk}_U, M^*, \hat{\sigma}^*)$$

$$\text{invalid} \leftarrow \text{Ver}(\text{pk}_A, \text{pk}_U, M^*, \text{Res}(\text{pk}_A, \text{pk}_U, M^*, \hat{\sigma}^*, \text{sk}_A))$$

Security Against Verifiers This property guarantees that even when the verifier colludes with $t - 1$ signers, they cannot create a valid full signature. This property intends to protect honest signers. Our model is static in the sense that the subset of signers to be controlled by the attacker is fixed during the initialization phase. Specifically, we use the following three-phase game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ to define this property.

Initialization $\mathcal{A}$ specifies the number of signers n and the threshold t , together with an index set $\mathcal{I}' \subset [n]$ such that $|\mathcal{I}'| = t - 1$. $\mathcal{C}$ creates the common reference string param_{CRS} and invokes

$$(\text{pk}_A, \text{sk}_A) \leftarrow \text{AGen}(),$$

$$(\text{pk}_U, \{\text{sk}_{U,i}\}_{i=1}^n) \leftarrow \text{UGen}(n, t).$$

$\mathcal{C}$ gives $(\text{param}_{CRS}, \text{pk}_A, \text{pk}_U, \{\text{sk}_{U,i}\}_{i \in \mathcal{I}'})$ to $\mathcal{A}$.

Query $\mathcal{A}$ can adaptively issue the following query to $\mathcal{C}$.

- PSign_s Query. $\mathcal{A}$ gives (M, i) to $\mathcal{C}$, who invokes

$$\hat{\sigma}_i \leftarrow \text{PSign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$$
 and returns $\hat{\sigma}_i$ to $\mathcal{A}$.
- Sign_s Query. $\mathcal{A}$ gives (M, i) to $\mathcal{C}$, who invokes

$$\sigma_i \leftarrow \text{Sign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$$
 and returns σ_i to $\mathcal{A}$.
- Res Query. $\mathcal{A}$ gives $(\hat{\sigma}, M)$ to $\mathcal{C}$, who invokes

$$\sigma \leftarrow \text{Res}(\text{pk}_A, \text{pk}_U, M, \hat{\sigma}, \text{sk}_A)$$
 and returns σ to $\mathcal{A}$.

End-Game $\mathcal{A}$ submits $(M^*, \hat{\sigma}^*)$ and wins the game if

$$\text{valid} \leftarrow \text{Ver}(\text{pk}_A, \text{pk}_U, M^*, \hat{\sigma}^*)$$

and that $(M^*, \cdot)$ did not appear in any $\text{Sign}_{(s)}$ query. Furthermore, if there exists a $\text{PSign}_{(s)}$ query with input $(M^*, \cdot)$, $(\cdot, M^*)$ should not appear as input in any Res query.

Security Against the Arbitrator This property guarantees that the arbitrator cannot create a signature on behalf of the user unless it is given a valid partial signature. In TOFE, we allow the arbitrator to collude with $t - 1$ signers. As in the case of security against verifiers, our model is static in the sense that the subset of signers to be controlled by the attacker is fixed during the initialization phase. Specifically, we use the following three-phase game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ to define this property.

Initialization $\mathcal{A}$ specifies the number of signers n and the threshold t , together with an index set $\mathcal{I}' \subset [n]$ such that $|\mathcal{I}'| = t - 1$. $\mathcal{C}$ creates the common reference string param_{CRS} and invokes

$$(\text{pk}_A, \text{sk}_A) \leftarrow \text{AGen}(),$$

$$(\text{pk}_U, \{\text{sk}_{U,i}\}_{i=1}^n) \leftarrow \text{UGen}(n, t).$$

$\mathcal{C}$ gives $(\text{param}_{CRS}, \text{pk}_A, \text{pk}_U, \{\text{sk}_{U,i}\}_{i \in \mathcal{I}'}, \text{sk}_A)$ to $\mathcal{A}$.

Query $\mathcal{A}$ can adaptively issue the following query to $\mathcal{C}$.

- PSign_s Query. $\mathcal{A}$ gives (M, i) to $\mathcal{C}$, who invokes

$$\hat{\sigma}_i \leftarrow \text{PSign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$$
 and returns $\hat{\sigma}_i$ to $\mathcal{A}$.
- Sign_s Query. $\mathcal{A}$ gives (M, i) to $\mathcal{C}$, who invokes

$$\sigma_i \leftarrow \text{Sign}_{(s)}(\text{pk}_A, M, \text{sk}_{U,i})$$
 and returns σ_i to $\mathcal{A}$.

End-Game $\mathcal{A}$ submits $(M^*, \hat{\sigma}^*)$ and wins the game if

$$\text{valid} \leftarrow \text{Ver}(\text{pk}_A, \text{pk}_U, M^*, \hat{\sigma}^*)$$

and that $(M^*, \cdot)$ did not appear in any $\text{Sign}_{(s)}$ query nor $\text{PSign}_{(s)}$ query.

4 Construction

Our TOFE is motivated by the ordinary OFE by [5]. Indeed, when $t = n = 1$, our construction degenerates to their scheme.

Common Reference String Our construction works in the common reference string model. For a security parameter 1^k , let $\mathbb{G}, \mathbb{G}_T$ be cyclic groups of prime order p with g as a generator of $\mathbb{G}$, where p is a k -bit prime. Further, let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map. The common reference string is defined to be

$$\text{param}_{CRS} := (1^k, \mathbb{G}, \mathbb{G}_T, p, g, \hat{e}).$$

AGen On input param_{CRS} , the arbitrator picks at random $y \in_R \mathbb{Z}_p$ and computes $Y = g^y$. The public key and secret key of the arbitrator is defined as

$$(\text{pk}_A, \text{sk}_A) := (Y, y).$$

UGen On input param_{CRS} , the required number of signers n and the threshold t , the user picks at random a polynomial of degree $t-1$ in $\mathbb{Z}_p$, say f . Assume the signers are indexed by i , for $i = 1$ to n , with $n \geq t \geq 1$. The user further picks at random a hash function $H : \{0, 1\}^* \rightarrow \mathbb{G}$. Note that H is to be modelled as a random oracle.

For $i = 1$ to n , the secret signing key of signer i is defined as $f(i)$.

The user computes the public key as

$$\text{pk}_U := (H, X, X_1, \dots, X_n) := (H, g^{f(0)}, g^{f(1)}, \dots, g^{f(n)}).$$

The value $f(0)$, which is the actual master secret, should be deleted. This ensures only a set of t signers together could create a threshold signature.

Remarks: If the signers are supposed to be honest, only X is necessary. The other components allows dishonest signer to be identified.

PSign The partial signature generation process consists of three sub-algorithms.

- *Generation of a Partial Signature Share* On input param_{CRS} , pk_A , a message M and the signing key of signer i $f(i)$, signer i randomly picks $r_i \in_R \mathbb{Z}_p$ and outputs the partial signature share as

$$\hat{\sigma}_i := (\alpha_i, \beta_i) := (H(M)^{f(i)} Y^{r_i}, g^{r_i}).$$

- *Verification of a Partial Signature Share* The partial signature share $\hat{\sigma}_i$ can be verified by evaluating the following relation:

$$\hat{e}(\alpha_i, g) \stackrel{?}{=} \hat{e}(H(M), X_i) \hat{e}(Y, \beta_i).$$

Remarks: We would like to remark again that this is not necessary if the signers are assumed to be trusted.

- *Generation of a Partial Signature* When t partial signature shares, say, $\hat{\sigma}_i$ for $i \in \mathcal{I} \subset [n]$ such that $|\mathcal{I}| = t$ on the same message, say M , have been collected, anyone can output the partial signature on message M as:

$$\hat{\sigma} := (\alpha, \beta) := \left(\prod_{i \in \mathcal{I}} \alpha_i^{\lambda_i}, \prod_{i \in \mathcal{I}} \beta_i^{\lambda_i} \right).$$

where λ_i is defined as

$$\lambda_i := \prod_{j \in \mathcal{I} \setminus \{i\}} \frac{-j}{i-j}.$$

As discussed, $f(0) = \sum_{i \in \mathcal{I}} f(i) \lambda_i$.

PVer On input param_{CRS} , pk_A , pk_U , a message M and a partial signature $\hat{\sigma}$, the algorithm outputs **valid** if and only if the following equality holds:

$$\hat{e}(\alpha, g) = \hat{e}(H(M), X) \hat{e}(Y, \beta).$$

Sign The full signature generation process consists of three sub-algorithms as well.

- *Generation of a Signature Share* On input param_{CRS} , pk_A , a message M and the signing key of signer i $f(i)$, signer i outputs the signature share as

$$\sigma_i := H(M)^{f(i)}.$$

- *Verification of a Signature Share* The signature share σ_i can be verified by evaluating the following relation:

$$\hat{e}(\sigma_i, g) \stackrel{?}{=} \hat{e}(H(M), X_i).$$

Remarks: We would like to remark again that this is not necessary if the signers are assumed to be trusted.

- *Generation of a Signature* When t signature shares, say, σ_i for $i \in \mathcal{I} \subset [n]$ such that $|\mathcal{I}| = t$ on the same message, say M , have been collected, anyone can output the signature on message M as:

$$\sigma := \prod_{i \in \mathcal{I}} \sigma_i^{\lambda_i}$$

where λ_i is defined as

$$\lambda_i := \prod_{j \in \mathcal{I} \setminus \{i\}} \frac{-j}{i-j}.$$

Ver On input param_{CRS} , pk_A , pk_U , a message M and a signature σ , the algorithm outputs **valid** if and only if the following equality holds:

$$\hat{e}(\sigma, g) = \hat{e}(H(M), X).$$

Res On input param_{CRS} , pk_A , pk_U , a message M , a partial signature $\hat{\sigma}$ and the secret key of the arbitrator y , the full signature can be computed as follows.

- Check that $\hat{\sigma}$ is a valid partial signature by evaluating the relation

$$\hat{e}(\alpha, g) \stackrel{?}{=} \hat{e}(H(M), X) \hat{e}(Y, \beta).$$

- Output σ as

$$\sigma := \alpha / \beta^y.$$

Regarding the security of our construction of TOFE, we have the following theorem, whose proof can be found in the full paper.

Theorem 1. *Our construction of name is secure against signers, verifiers and the arbitrator under the CDH assumption in the random oracle model.*

5 Conclusion

We present the first threshold-oriented fair exchange protocol which allows a subset of signers to exchange a digital item with a counter party. Indeed, in our specific construction, the item being exchanged is a threshold signature. We define formal security model for TOFE, present an efficient construction and show that it is secure in the random oracle model under well-known assumptions. We leave construction of TOFE in the standard model as an open problem.

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: Proc. 4th ACM Conference on Computer and Communications Security. pp. 8–17 (1997)
2. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. Advances in Cryptology-Eurocrypt 1998, Lecture Notes in Computer Science 1403 pp. 591 – 606 (1998)
3. Avoine, G., Vaudenay, S.: Optimistic fair exchange based on publicly verifiable secret sharing. In: In Australasian Conference on Information Security and Privacy CURRICULUM VIT ACISP04. pp. 74–85. Springer-Verlag (2004)
4. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer (2001)
5. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. Advances in cryptology - Eurocrypt 2003, Lecture Notes in Computer Science 2656, 416–432 (2003)
6. Camenisch, J., Damgård, I.: Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In: Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology. pp. 331–345. ASIACRYPT '00, Springer-Verlag, London, UK (2000), <http://dl.acm.org/citation.cfm?id=647096.716992>

7. Dodis, Y., Lee, P.J., Yum, D.H.: Optimistic fair exchange in a multi-user setting. In: Proceedings of the 10th international conference on Practice and theory in public-key cryptography. pp. 118–133. PKC'07, Springer-Verlag, Berlin, Heidelberg (2007), <http://dl.acm.org/citation.cfm?id=1760564.1760577>
8. Dodis, Y., Reyzin, L.: Breaking and repairing optimistic fair exchange from podc 2003. In: Proceedings of the 3rd ACM workshop on Digital rights management. pp. 47–54. DRM '03, ACM, New York, NY, USA (2003), <http://doi.acm.org/10.1145/947380.947387>
9. Garay, J.A., Jakobsson, M., MacKenzie, P.D.: Abuse-free optimistic contract signing. In: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology. pp. 449–466. CRYPTO '99, Springer-Verlag, London, UK, UK (1999), <http://dl.acm.org/citation.cfm?id=646764.703965>
10. Huang, Q., Wong, D.S., Susilo, W.: Group-oriented fair exchange of signatures. Inf. Sci. 181(16), 3267–3283 (Aug 2011), <http://dx.doi.org/10.1016/j.ins.2011.03.024>
11. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Ambiguous optimistic fair exchange. In: Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7–11, 2008. Proceedings. pp. 74–89 (2008)
12. Huang, Q., Yang, G., Wong, D.S., Susilo, W.: Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles. In: Proceedings of the 2008 The Cryptographers' Track at the RSA conference on Topics in cryptology. pp. 106–120. CT-RSA'08, Springer-Verlag, Berlin, Heidelberg (2008), <http://dl.acm.org/citation.cfm?id=1791688.1791698>
13. Huang, X., Mu, Y., Susilo, W., Wu, W., Zhou, J., Deng, R.H.: Preserving transparency and accountability in optimistic fair exchange of digital signatures. IEEE Transactions on Information Forensics and Security 6(2), 498–512 (2011)
14. Kate, A., Goldberg, I.: Distributed private-key generators for identity-based cryptography. In: Proceedings of the 7th international conference on Security and cryptography for networks. pp. 436–453. SCN'10, Springer-Verlag, Berlin, Heidelberg (2010), <http://dl.acm.org/citation.cfm?id=1885535.1885574>
15. Kp, A., Lysyanskaya, A.: Optimistic fair exchange with multiple arbiters. In: Proceedings of the 15th European conference on Research in computer security. pp. 488–507. ESORICS'10, Springer-Verlag, Berlin, Heidelberg (2010), <http://dl.acm.org/citation.cfm?id=1888881.1888919>
16. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: EUROCRYPT, 2006. pp. 465–485. Springer (2006)
17. Markowitch, O., Kremer, S.: An optimistic non-repudiation protocol with transparent trusted third party. In: Information Security Conference 2001, Lecture Notes in Computer Science. pp. 363–378. Springer-Verlag (2001)
18. Qu, L., Wang, G., Mu, Y.: Optimistic fair exchange of ring signatures. In: Rajarajan, M., Piper, F., Wang, H., Kesidis, G. (eds.) Security and Privacy in Communication Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 96, pp. 227–242. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-31909-9_13
19. Rckert, M., Schrder, D.: Security of verifiably encrypted signatures and a construction without random oracles. In: Proceedings of the 3rd International Conference Palo Alto on Pairing-Based Cryptography. pp. 17–34. Pairing '09, Springer-Verlag, Berlin, Heidelberg (2009)

20. Shamir, A.: How to share a secret. *Commun. ACM* 22(11), 612–613 (1979)
21. Zhang, J., Mao, J.: A novel verifiably encrypted signature scheme without random oracle. In: *Proceedings of the 3rd international conference on Information security practice and experience*. pp. 65–78. ISPEC'07, Springer-Verlag, Berlin, Heidelberg (2007), <http://dl.acm.org/citation.cfm?id=1759508.1759518>
22. Zhu, H., Bao, F.: Stand-alone and setup-free verifiably committed signatures. In: *CT-RSA. Lecture Notes in Computer Science*, vol. 3860, pp. 159–173 (2006)