Optimistic Fair Exchange of Ring Signatures

Lie Qu, Guilin Wang, and Yi Mu

Center for Computer and Information Security Research,
School of Computer Science and Software Engineering,
University of Wollongong, Wollongong,
NSW 2522, Australia
{lq594,guilin,ymu}@uow.edu.au

Abstract. An optimistic fair exchange (OFE) protocol is an effective tool helping two parties exchange their digital items in an equitable way with assistance of a trusted third party, called *arbitrator*, who is only required if needed. In previous studies, fair exchange is usually carried out between individual parties. When fair exchange is carried out between two members from distinct groups, anonymity of the signer in a group could be necessary for achieving better privacy. In this paper, we consider optimistic fair exchange of ring signatures (OFERS), i.e. two members from two different groups can exchange their ring signatures in a fair way with ambiguous signers. Each user in these groups has its own public-private key pair and is able to sign a message on behalf of its own group anonymously. We first define the security model of OFERS in the multi-user setting under adaptive chosen message, chosen-key and chosen public-key attacks. Then, based on verifiably encrypted ring signatures (VERS) we construct a concrete scheme by combining the technologies of ring signatures, public-key encryption and proof of knowledge. Finally, we show that our OFERS solution is provably secure in our security model, and preserving *signer-ambiguity* of ring signatures. To the best of our knowledge, this is the first (formal) work on this topic.

Keywords: optimistic fair exchange, ring signatures, privacy, verifiably encrypted signatures (VES).

1 Introduction

The concept of optimistic fair exchange (OFE) was first proposed by Asokan et al. [1]. By executing an OFE protocol, two parties in networks are able to fairly exchange their digital signatures with some help from an off-line trusted third party (TTP). An OFE protocol usually has at least the properties: *fairness, non-repudiation* and *optimism. Fairness* ensures that, if an honest party does not get a valid signature of the other party at the end of a fair exchange protocol, the other party cannot get that either. That is, either both two parties get each other's valid signature, or neither of them gets anything valuable. *Non-repudiation* guarantees that any party in a fair exchange protocol cannot repudiate or refute a valid signature after the protocol executed successfully. To reduce the load of the TTP, Asokan et al. proposed *optimistic* fair exchange [1].

M. Rajarajan et al. (Eds.): SecureComm 2011, LNICST 96, pp. 227–242, 2012.

In an OFE protocol, there is an off-line TTP, called *arbitrator*, who acts as a judge to settle the dispute between two parties and should only be involved when the protocol does not run correctly (e.g. some parties cheating or communication channel interrupted). The rare involvement of a TTP makes the fair exchange protocol more efficient and secure.

An conventional way to build optimistic fair exchange protocols is *verifiably encrypted signature* (VES), which was formally defined by Boneh et al. [2]. A VES is an ordinary signature encrypted using the public key of a TTP, together with a verifiable proof showing the validity of the encryption. Suppose Alice and Bob exchange their signatures on a message. Due to mutual distrust, neither of them wants to send his or her signature first. To solve this dilemma, Alice can send a VES generated under a TTP's public key to Bob first. Then, Bob is able to verify the validity of the VES together with a proof showing that Alice's signature encrypted in the VES can be recovered by the TTP, but cannot obtain the original signature from Alice unless Bob sends his own signature to Alice. After that, if Alice refuses to reveal Bob her signature, Bob can ask the TTP to decrypt Alice's VES and obtain her original signature.

In some cases, the anonymity of participants in fair exchange might be important in order to protect participants' privacy. For example, in the developed commercial society, the personal preferences of negotiators in business contract signing usually influence the terms of the final agreement. If a trading company A has the old contract signing records of an employee as a negotiator in another company B which is a potential trade cooperator of A, A can use these records to generalize the negotiator's trading habits, by which the company A might get advantages in the future contract negotiation with the company B. Hence it is desirable that the employees who have the right to independently sign a contract on behalf of their own company can sign contracts anonymously, which will prevent other companies from knowing the signer's trading habits. To this end, ring signatures invented by Rivest et al. [3] are the good primitive to provide the property *signer-ambiguity*, which was formally defined by Abe et al. [8]. Informally, in a ring signature scheme, the public keys of a group of users are collected spontaneously to form a public-key list. When a signer signs a message on behalf of such a ring, he uses the public-key list and adds his own private key as a glue value to issue a ring signature. A verifier cannot tell who the real signer is, because the ring signature is validated using all the public keys of the ring without revealing any information about who produced it.

In this paper, we study optimistic fair exchange of ring signatures (OFERS), in which users in each ring can fairly exchange their ring signatures with ambiguous signers for the other ring. To the best of our knowledge, this is the first work on the topic to present a formal security model of OFERS and a concrete solution with provable security. After introducing some preliminaries in Section 2, we first rigorously define the security model of OFERS in the multi-user setting under adaptive chosen message, chosen-key and chosen public-key attacks (Section 3). This is done by updating the formal models of OFE [5,6] in the scenario of ring signatures. Secondly, we present a concrete OFERS scheme (Section 4), which is constructed from verifiably encrypted ring signatures (VERS) based on Abe et al.'s scheme [8] under a TTP's public key, together with a proof of knowledge showing the validity of the original ring signature's encryption. Theoretically, any CCA2-secure [7] public-key encryption scheme can be used as such a proof of knowledge always exists (but may be not efficient). To provide practicality and high efficiency, Camenisch and Shoup's CCA2-secure encryption scheme [19] is particularly selected in the proposed scheme. Then, we formally show that the proposed OFERS solution is provably secure in our security model (Section 5). As the VES technique is employed, a notable feature of our scheme is that any holder (not necessarily the signer) of a valid ring signature can verifiably encrypt the ring signature to get a VERS without using any secret information from the signer. Due to this feature, our scheme not only preserves *signer-ambiguity* [8] of ring signatures, but also allows a signer to delegate a proxy (e.g. his/her secretary) to run OFERS after he/she produced a ring signature in advance. Finally, we discuss some extensions of our results and point out future work (Section 6).

2 Preliminaries

In this section, we introduce the technologies used in our OFERS scheme.

2.1 Ring Signature of All Discrete-Log Case

Abe et al. proposed an abstract scheme of a ring signature and several concrete examples in [8]. For the sake of simplicity, we choose the ring signature scheme of all discrete-log case in [8] as our signature scheme. And Abe et al. have proved that this ring signature scheme is unconditionally signer-ambiguous and existential unforgeability against adaptive chosen message and chosen public-key attacks. The details of the scheme are shown below:

Let p_i, q_i be large primes, $\langle g_i \rangle$ denote a prime subgroup of $\mathbb{Z}_{p_i}^*$ generated by g_i whose order is q_i. Let $y_i = g_i^{x_i} \bmod p_i$, where x_i is the secret key and (y_i, p_i, q_i, g_i) is the public key. $H_i : \{0,1\}^* \to \mathbb{Z}_{q_i}$ denotes a collision-resistant hash function. L is a list of (y_i, p_i, q_i, g_i), where $i = 0, ..., n-1$ and $n = |L|$. A signer with the secret key x_k generates a ring signature on a message m under L as follows:

1. Randomly select $\alpha \in \mathbb{Z}_{q_k}$ and compute $c_{k+1} = H_{k+1}(L, m, g_k^{\alpha} \bmod p_k)$.
2. For $i = k+1, ..., n-1, 0, ..., k-1$, randomly select $s_i \in \mathbb{Z}_{q_i}$ and compute $c_{i+1} = H_{i+1}(L, m, g_i^{s_i} y_i^{c_i} \bmod p_i)$, and then $s_k = \alpha - x_k c_k \bmod q_k$.
3. Send the verifier $(c_0, s_0, s_1, ..., s_{n-1})$ as the resulting ring signature on the message m under the public-key list L.

For $i = 0, ..., n-1$, the verifier computes $e_i = g_i^{s_i} y_i^{c_i} \bmod p_i$, and then $c_{i+1} = H_{i+1}(L, m, e_i)$ if $i \neq n-1$. The verifier accepts the ring signature if $c_0 = H_0(L, m, e_{n-1})$, otherwise rejects.

2.2 Zero-knowledge Proof

In [9], Ateniese introduced an underlying proof of the equality of discrete logarithms, which is used for constructing verifiably encrypted signatures. In [11], Camenisch and Michels proposed a concrete scheme to prove the equality of discrete logarithms from different groups under the strong RSA assumption [12,13]. In this paper, we modify Camenisch and Michels' proof as our zero-knowledge proof so as to build a verifiably encrypted signature scheme based on Abe et al. [8]'s ring signature introduced above. Camenisch and Michels' proof is denoted by $PK\{(\alpha,\beta) : y_1 \overset{G_1}{=} g_1^{\alpha} \wedge y_2 \overset{G_2}{=} g_2^{\alpha} \wedge \tilde{y} \overset{\mathbb{Z}_n^*}{=} h_1^{\beta} h_2^{\alpha} \wedge (-2^l < \alpha < 2^l)\}$. The details of the proof are shown below:

n is the product of two sufficiently large safe primes and must be large enough to avoid factoring. h_1 and h_2 are two random elements with large order from $\mathbb{Z}_n$. Let G_1 and G_2 be two distinct groups of orders q_1 and q_2 such that $2^{l+1} < min(q_1, q_2)$, where l is an integer, and g_1 and g_2 are the generators of G_1 and G_2 respectively. Let $y_1 \overset{G_1}{=} g_1^x$ and $y_2 \overset{G_2}{=} g_2^x$, $\epsilon > 1$ is a security parameter which controls the tightness of the statistical zero-knowledgeness. If $-2^{(l-2)/\epsilon} < x < 2^{(l-2)/\epsilon}$, the prover can convince the verifier that $\log_{g_1}^{y_1} = \log_{g_2}^{y_2}$ in $\mathbb{Z}$ by the following steps:

1. The prover randomly chooses $r \in \mathbb{Z}_n$ and computes $\tilde{y} = h_1^r h_2^x \bmod n$, then randomly selects $r_1 \in \{-2^{l-2}, ..., 2^{l-2}\}$ and $r_2 \in \{-(n2^k)^{\epsilon}, ..., (n2^k)^{\epsilon}\}$, where k is the length of bits of the verifier's challenge, and computes the commitments: $t_1 = g_1^{r_1}$, $t_2 = g_2^{r_1}$, and $t_3 = h_1^{r_2} h_2^{r_1}$. After that, the prover sends (t_1, t_2, t_3) to the verifier.
2. The verifier returns a random challenge $c \in \{0,1\}^k$.
3. The prover computes the responses $s_1 = r_1 - cx$ and $s_2 = r_2 - cr$ in $\mathbb{Z}$, then sends (s_1, s_2) to the verifier.
4. The verifier accepts the proof if and only if $-2^{l-1} < s_1 < 2^{l-1}$, $t_1 = g_1^{s_1} y_1^c$, $t_2 = g_2^{s_1} y_2^c$ and $t_3 = h_1^{s_2} h_2^{s_1} \tilde{y}^c$ hold.

Note that the proof above is based on the strong RSA assumption. The prover should not know the factoring of n. Hence n, h_1, h_2 might be generated by the verifier or a trusted third party. Before executing the proof, the prover should check whether n is the product of two safe primes (see [14] for details) and whether h_1 and h_2 have large order (see [15] for details). To convert this interactive proof into a signature form on a message m, the prover can use a suitable hash function $h(\cdot)$, which is agreed by the verifier, to compute the hash value of all the public information instead of the verifier's challenge c (e.g. $c = h(m||\tilde{y}||y_1||y_2||g_1||g_2||t_1||t_2||t_3)$).

2.3 Encryption Scheme

In [9], Ateniese proposed a method to construct verifiably encrypted signatures by encrypting an ordinary signature using some specific public-key cryptosystems and giving a proof showing the validity of the signature's encryption. In such cryptosystems (e.g. Naccache-Stern [16], Okamoto-Uchiyama [17] and Paillier [18] public-key cryptosystems), computing a discrete logarithm using the secret key is an easy task, but without the secret key, it is still hard. However, all these public-key cryptosystems above do not satisfy the high level security which protects against adaptive chosen-ciphertext attacks (CCA2). In [19], Camenisch and Shoup proposed an adaptation of Paillier cryptosystem, which is proven secure against adaptive chosen ciphertext attacks under the decisional composite residuosity assumption [18]. To achieve the high level security, we use Camenisch and Shoup's scheme as our encryption scheme, which is briefly described as follows:

1. Randomly select two Sophie Germain primes p' and q', where $p' \neq q'$, and compute safe primes $p = 2p' + 1$, $q = 2q' + 1$ and $n = pq$. Then randomly select $x_1, x_2, x_3 \in_R [n^2/4]$ [1] and $g' \in \mathbb{Z}^*_{n^2}$, and compute $g = (g')^{2n}$, $y_1 = g_1^{x_1}$, $y_2 = g_2^{x_2}$, $y_3 = g_3^{x_3}$. Let $h = (1 + n \bmod n^2) \in \mathbb{Z}^*_{n^2}$, $\mathbf{abs}$: $\mathbb{Z}^*_{n^2} \to \mathbb{Z}^*_{n^2}$ map $(a \bmod n^2)$, where $0 < a < n^2$, to $(n^2 - a \bmod n^2)$ if $a > n^2/2$, and to $(a \bmod n^2)$ otherwise. Obviously for any $v \in \mathbb{Z}^*_{n^2}$, $v^2 = (\mathbf{abs}(v))^2$ holds. H is a collision-resistant hash function. A label L is some public information added to the ciphertext (e.g. user's identity or expiration time). The public key is (n, g, y_1, y_2, y_3), and the private key is (x_1, x_2, x_3).
2. To encrypt a message $m \in [n]$ with a label $L \in \{0,1\}^*$, randomly select $r \in_R [n/4]$ and compute $u = g^r$, $e = y_1^r h^m$ and $v = \mathbf{abs}((y_2 y_3^{H(u,e,L)})^r)$. The triple (u, e, v) is the resulting ciphertext.
3. To decrypt a ciphertext (u, e, v), first check whether $\mathbf{abs}(v) = v$ and $u^{2(x_2 + H(u,e,L)x_3)} = v^2$. If fail, output reject, otherwise compute $\hat{m} = (e/u^{x_1})^{2t}$, where $t = 2^{-1} \bmod n$. If $\hat{m}$ is of the form h^m for some $m \in [n]$, then output m, otherwise output reject.

Recall the ring signature scheme presented in Section 2.1. Suppose the signer generates a ring signature $(c_0, s_0, s_1, ..., s_{n-1})$. In the verification of this signature, the verifier needs to compute $e_i = g_i^{s_i} y_i^{c_i}$, where $i = 0, 1, ..., n-1$. In order to convert the ring signature into a verifiably encrypted ring signature (VERS), the signer sends the verifier $w_i = g_i^{s_i}$ instead of s_i and encrypts s_i using a TTP's public key. The verifier can do the verification by computing $e_i = w_i y_i^{c_i}$ instead, but s_i is 'hidden' in w_i since in this ring signature scheme computing a discrete logarithm is hard, which means the verifier has not got the full ring signature yet. Beside that, the signer needs to give a zero-knowledge proof for convincing the verifier that the encrypted s_i is just the s_i hidden in w_i. Note that encrypting only one value in $(s_0, s_1, ..., s_{n-1})$ can also ensure the initial ring signature hidden partially, which means the verifier still cannot draw the full ring signature from the partially encrypted ring signature even though he gets the most parts of the initial ring signature. Encrypting one value makes the cost of generating a VERS does not depend on the size of the public-key list, which improves the efficiency of the generation of a VERS.

[1] For a positive integer a, $[a]$ denotes the set $\{0, 1, ..., a-1\}$.

To produce a verifiably encrypted ring signature, suppose the signer randomly chooses s_u, where $0 \leqslant u \leqslant n-1$, from $(s_0, s_1, ..., s_{n-1})$ as the hidden value, and encrypts s_u using Camenisch and Shoup's encryption scheme above. Let $(\mathsf{n}, \mathsf{g}, \mathsf{y}_1, \mathsf{y}_2, \mathsf{y}_3, \mathsf{h})$ be the public key of a TTP. H is a collision-resistant hash function, and L is the public label. The signer computes s_u's ciphertext $\mathsf{u} = \mathsf{g}^{\mathsf{t}}, \mathsf{e} = \mathsf{y}_1{}^{\mathsf{t}}\mathsf{h}^{s_u}, \mathsf{v} = \mathbf{abs}((\mathsf{y}_2\mathsf{y}_3^{\mathsf{H}(\mathsf{u},\mathsf{e},\mathsf{L})})^{\mathsf{t}})$, where $\mathsf{t} \in_R [\mathsf{n}/4]$. After that, by modifying the zero-knowledge proof introduced in Section 2.2, the signer gives a non-interactive proof: $PK\{(s_u, \mathsf{t}, r) : w = g_u^{s_u} \wedge \mathsf{u}^2 = \mathsf{g}^{2\mathsf{t}} \wedge \mathsf{e}^2 = \mathsf{y}_1{}^{2\mathsf{t}}\mathsf{h}^{2s_u} \wedge \mathsf{v}^2 = (\mathsf{y}_2\mathsf{y}_3^{\mathsf{H}(\mathsf{u},\mathsf{e},\mathsf{L})})^{2\mathsf{t}} \wedge \hat{w} = h_1^r h_2^{s_u} \wedge -2^l < s_u < 2^l\}$ to convince the verifier that the TTP can extract s_u using its secret key and recover the original ring signature completely. Note that anyone beside the signer has the capability to convert a valid ring signature into a VERS without knowing any secret information from the signer. The property *signer-ambiguity* [8] is well preserved since the hidden value can be arbitrarily chosen in $(s_0, ..., s_{n-1})$ and no secret of the signer is needed for producing a VERS based on a given ring signature. In our verifiably encrypted ring signature scheme, for the sake of simplicity, we specify s_{n-1} as the hidden value encrypted using a TTP's public key no matter who the signer is. The details are shown in Section 4.

3 Security Definitions

In [5], Dodis et al. presented a formal security model of optimistic fair exchange under *adaptive chosen message attacks* in a *multi-user setting*, in which the optimistic fair exchange protocol can be executed between different signers and different verifiers. That is, multiple pairs of users can run the two-party fair exchange protocol without compromising security. In adaptive chosen message attacks [20], an adversary can access the signing oracle by asking for signatures on arbitrary messages. In ring signatures, there are multiple users belonging to each public-key list. So the multi-user setting is necessary for fair exchange of ring signatures. Furthermore, Huang et al. [6] extended Dodis et al.'s model by considering *chosen-key model*, i.e. an adversary may win a computational game if it is allowed to employ some public keys without knowing the corresponding private keys. By providing this extra flexibility, the chosen-key model is stronger than the certified-key model (shown in [6]). In addition, we also consider *chosen public-key attacks* in the setting of ring signatures, which is proposed by Abe et al. [8]. In chosen public-key attacks, any adversary who wants to forge a ring signature is only allowed to use arbitrary subsets of the initially considered public-key list to access the signing oracle, but cannot append new public keys to the initial public-key list. Therefore, in our security definitions specified below, all the four factors above are addressed in the setting of OFERS as a whole.

Definition 1. (Syntax) `Optimistic fair exchange of ring signatures` (`OFERS`) *consists of seven probabilistic polynomial-time algorithms.*

- $\mathbf{Setup^{TTP}}$: *On input a security parameter* ***Param****, the arbitrator executes the algorithm to generate a public-private key pair (APK, ASK) and some auxiliary information if necessary.*
- $\mathbf{Setup^{User}}$: *On input* ***Param*** *and (optionally) the arbitrator's public key with the auxiliary information, the algorithm outputs public-private key pairs* $(PK_i,\ SK_i)$ *for every user in the ring. The public keys form a public-key list L.*
- $\mathbf{RSig}(\boldsymbol{m},\ \boldsymbol{L},\ SK_s)$: *A signer* U_s *in the ring executes the algorithm by inputting a message m, a public-keys list L including* PK_s *and its corresponding private key* SK_s*, then outputs a ring signature* σ*.*
- $\mathbf{RVer}(\boldsymbol{m},\ \boldsymbol{L},\ \sigma)$: *On input a message m, a ring signature* σ *on m under a public-key list L, a verifier executes the algorithm to output either* 1 *or* 0*, which means accept or reject respectively.*
- $\mathbf{PRSig}(\boldsymbol{m},\ \boldsymbol{L},\ \boldsymbol{\sigma},\ \boldsymbol{APK})$: *On input a message m, a signer's public-key list L, a ring signature* σ *on m under L, and the arbitrator's public key APK, the algorithm outputs a verifiably partial ring signature* θ*.*
- $\mathbf{PRVer}(\boldsymbol{m},\ \boldsymbol{L},\ \boldsymbol{\theta},\ \boldsymbol{APK})$: *On input a message m, a signer's public-key list L, a verifiably partial ring signature* θ *on m under L, and the arbitrator's public key APK, the verifier executes the algorithm to output either* 1 *or* 0*, which means accept or reject respectively.*
- $\mathbf{Res}(\boldsymbol{m},\ \boldsymbol{L},\ \boldsymbol{\theta},\ \boldsymbol{ASK})$: *The resolution algorithm is executed by the arbitrator if the verifier does not receive the full ring signature* σ *from the signer ring, but has got the corresponding verifiably partial ring signature* θ*. On input a message m, a signer's public-key list L and a verifiably partial ring signature* θ *on m under L, if* θ *is valid and the verifier has fulfilled its obligation to the signer, the arbitrator extracts the full ring signature* σ *from* θ *using its private key ASK and reveals it to the verifier, otherwise rejects.*

Since there are three roles (*signer, verifier, arbitrator*) in OFERS, we should consider how each role may violate different aspects of security, i.e. different security properties. Here we require the arbitrator should not be able to cheat some participant by colluding with the other participant in the protocol since such a collusive adversarial arbitrator can break the fair exchange trivially. Moreover, the property *signer-ambiguity* should also be addressed as it is the heritage of ring signatures.

Security Against Signers: For the fairness to verifiers, it is required that except negligible probability, any probabilistic polynomial-time (PPT) adversarial signer $\mathcal{A}$ should be not able to generate a verifiably partial ring signature, which can be accepted by verifiers, but cannot be recovered to a valid full ring signature by an honest arbitrator. The property is formally defined by the following game:

$$\begin{aligned}
\mathbf{Setup^{TTP}}(\mathbf{Param}) &\longrightarrow (ASK, APK)\\
(m, L^*, \theta) &\longleftarrow \mathcal{A}^{O_{Res}}(APK)\\
\sigma &\longleftarrow \mathbf{Res(}m, L^*, \theta, ASK\mathbf{)}\\
\text{Success of } \mathcal{A} &= [\mathbf{PRVer}(m, L^*, \theta, APK){=}1 \wedge \mathbf{RVer}(m, L^*, \sigma){=}0]
\end{aligned}$$

where O_{Res} denotes a resolution oracle, which takes as input a verifiably partial ring signature on a message m under a public-key list L, and outputs a full ring signature σ on m under L. In this game, the adversary $\mathcal{A}$ is allowed to *arbitrarily* (i.e. not necessarily following the key generation algorithm) generate public keys to form a list L^*. For each public key in L^*, $\mathcal{A}$ may not know the corresponding private key. The chosen-key model is therefore accommodated here.

Definition 2 (Security Against Signers). Optimistic fair exchange of ring signatures is said to be `secure against signers` *if there is no PPT adversarial signer $\mathcal{A}$ who wins the game above with non-negligible probability.*

Security Against Verifiers: The property of *security against verifiers* requires that, without help from the signer or the arbitrator, any PPT adversarial verifier $\mathcal{B}$ should not be able to extract a full ring signature from the corresponding verifiably partial ring signature with non-negligible probability. The property is formally defined by the following game:

$$\begin{array}{rcl} \mathbf{Setup^{TTP}}(\mathbf{Param}) & \longrightarrow & (ASK, APK) \\ \mathbf{Setup^{User}}(\mathbf{Param}) & \longrightarrow & (SK_i, PK_i) \\ (m, L', \sigma) & \longleftarrow & \mathcal{B}^{O_{PRSig}, O_{Res}}(APK, L) \\ \text{Success of } \mathcal{B} & = & [\mathbf{RVer}(m, L', \sigma){=}1 \wedge (m, L', \cdot) \notin \textit{Query}(\mathcal{B}, O_{Res})] \end{array}$$

where L' is an arbitrary subset of the initial public-key list L consisting of all the PK_i, the oracle O_{Res} has been defined in the previous game, and the partial ring signature signing oracle O_{PRSig}, given as input a message m and a public key list L'', outputs a verifiably partial ring signature on m under L'' using the arbitrator's public key APK. The *Query($\mathcal{B}, O_{Res}$)* is the set of valid queries which $\mathcal{B}$ asks to O_{Res}. In this game, $\mathcal{B}$ can ask the arbitrator for resolving any verifiably partial ring signature with respect to any sublist of L. Note that here chosen-public key attacks are considered, as the adversary $\mathcal{B}$ is only required to output a valid ring signature under L' which is a subset of L but not necessarily L. Moreover, L' does not contain any public key generated by $\mathcal{B}$. Otherwise, $\mathcal{B}$ can win the game above trivially.

Definition 3 (Security Against Verifiers). Optimistic fair exchange of ring signatures is said to be `secure against verifiers` *if there is no PPT adversarial verifier $\mathcal{B}$ who wins the game above with non-negligible probability.*

Security Against the Arbitrator: For the fairness to signers, the property of *security against the arbitrator* requires that except negligible probability, any PPT adversarial arbitrator $\mathcal{C}$ should not be able to produce a full ring signature without demanding the signer to generate a verifiably partial ring signatures. The property is formally defined by the following game:

$$\begin{array}{rcl} \mathbf{Setup^{User}}(\mathbf{Param}) & \longrightarrow & (PK_i, SK_i) \\ (ASK^*, APK) & \longleftarrow & \mathcal{C}(L) \\ (m, L', \sigma) & \longleftarrow & \mathcal{C}^{O_{PRSig}}(ASK^*, APK, L) \\ \text{Success of } \mathcal{C} & = & [\mathbf{RVer}(m, L', \sigma){=}1 \wedge (m, L') \notin \textit{Query}(\mathcal{C}, O_{PRSig})] \end{array}$$

where the oracles O_{Res}, O_{PRSig}, the public-key lists L' and L have been described in the previous games, and ASK^* is the state information of $\mathcal{C}$, which may not correspond to the arbitrator's public key APK. *Query($\mathcal{C}, O_{PRSig}$)* is the set of valid queries which $\mathcal{C}$ asks to O_{PRSig}. We remark that this game considers both chosen-key and chosen public-key attacks in the multi-user setting, as the adversary $\mathcal{C}$ (a malicious arbitrator) does not need to know the corresponding private key of the public key APK and can choose any sublist L' of the initial public-key list to forge a ring signature.

Definition 4 (Security Against the Arbitrator). Optimistic fair exchange of ring signatures is said to be `secure against the arbitrator` *if there is no PPT adversarial arbitrator $\mathcal{C}$ who wins the game above with non-negligible probability.* In [8], Abe et al. specified the security definition of *signer-ambiguity.* In our OFERS scheme, the signer should be still ambiguous in its own ring. By updating Abe et al.'s definition in the setting of OFERS, we formally define signer-ambiguity as follows:

*Definition 5 (Signer Ambiguity). Let $L = \{PK_i\}$ be an initial public-key list, where each PK_i is generated by running **Setup**$^{\textbf{User}} \rightarrow (PK_i, SK_i)$, and APK be the arbitrator's public key generated by running **Setup**$^{\textbf{TTP}} \rightarrow (APK, ASK)$. An OFERS protocol is called* `perfectly signer-ambiguous`, *if for any message m, any public-key list L, any public key APK of the arbitrator, any valid full ring signature $\sigma \leftarrow$ **RSign**(m, L, SK_s), and an associated verifiably partial ring signature $\theta \leftarrow$ **PRSig**(m, L, σ, APK), where SK_s is the signer's private key, given $(m, L, \theta, \sigma, APK)$, any unbound adversary $\mathcal{D}$ outputs index i such that $SK_s = SK_i$ with probability exactly $\frac{1}{|L|}$, where $|L|$ denotes the size of L.*

Remark 1. Comparing with Abe et al.'s perfect signer-ambiguity [8] for ring signatures, we also provide the verifiably partial ring signature θ of a full ring signature σ to the adversary $\mathcal{D}$, which allows $\mathcal{D}$ acquiring more information to break signer-ambiguity. In fact, this is necessary because the signer-ambiguity in ring signatures does not always guarantee the same property for OFERS (refer to the counterexample discussion in Section 5). As the unbound adversary $\mathcal{D}$ can derive all private keys from L, the above definition essentially means that for fixed (m, L, APK), the distributions of θ and σ generated by using any private key SK_i are identical. In addition, Definition 5 specifies *perfect signer-ambiguity*, and it can be easily extended to define *statistical* and *computational signer-ambiguity*, two weaker versions of ambiguity.

4 The Scheme

In our OFERS scheme, we use verifiably encrypted ring signatures (VERS) to construct verifiably partial ring signatures. In this section, we first present how to produce a VERS, and then give an optimistic fair exchange protocol of ring signatures. The generation and verification of ring signatures are similar to Abe et al.'s ring signature in all discrete-log case (see Section 2.1) except some limitation of selecting α and s_i. For the sake of simplicity, in our VERS scheme, we always encrypt the last s_i, i.e. s_{n-1}, as the hidden value. Obviously this does not affect the scheme's security since any s_i in $(s_0, ..., s_{n-1})$ can be the hidden value no matter who the signer is. Then we use Camenisch and Shoup's CCA2-secure encryption scheme and give a proof:

$$PK\{(s_{n-1}, \mathsf{t}, r) : w = g_{n-1}^{s_{n-1}} \wedge \mathsf{u}^2 = \mathsf{g}^{2\mathsf{t}} \wedge \mathsf{e}^2 = \mathsf{y_1}^{2\mathsf{t}}\mathsf{h}^{2s_{n-1}} \wedge \mathsf{v}^2 = (\mathsf{y_2}\mathsf{y_3}^{\mathsf{H(u,e,L)}})^{2\mathsf{t}} \wedge$$
$$\hat{w} = h_1^r h_2^{s_{n-1}} \wedge -2^l < s_{n-1} < 2^l\}$$

for convincing the verifier the validity of the encryption (see Section 2.3).

4.1 Verifiably Encrypted Ring Signature

The generation of a VERS consists of two steps. One is producing a conventional ring signature consisting of three algorithms denoted by **RS** = (**RKG, Sig, Ver**), the other is encrypting the ring signature consisting of three algorithms denoted by **EN** = (**Gen, Enc, Dec**) with a zero-knowledge showing the validity of the ring signature's encryption. Suppose there are two rings called R_I and R_J. U_i and U_j denote the users in these two rings respectively. A signer U_k in the ring R_I sends a VERS on a message m to a verifier in the ring R_J. L_I and L_J denote the public-key list of the ring R_I and R_J, and $n_I = |L_I|$ and $n_J = |L_J|$ denote the size of L_I and L_J respectively.

Setup$^{\mathbf{TTP}}$: On input the security parameter **Param**, the arbitrator executes the key generation algorithm to output the public key $(\mathsf{n}, \mathsf{g}, \mathsf{y}_1, \mathsf{y}_2, \mathsf{y}_3, \mathsf{h})$ and the private key $(\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3)$ under Camenisch and Shoup's encryption scheme [19]. q_A denotes the order of g, and l is an integer such that $2^{l+1} < q_A$. Meanwhile, the arbitrator generates h_1, h_2 and n, which are used in the zero-knowledge proof introduced in Section 2.2 (the modulus n must be large enough to avoid factoring but does not need to depend on **Param**) and publishes $(\mathsf{n}, \mathsf{g}, \mathsf{y}_1, \mathsf{y}_2, \mathsf{y}_3, \mathsf{h}, h_1, h_2, n, l)$.

Setup$^{\mathbf{User}}$: The setup of users is similar to the ring signature scheme in Section 2.1. For the user U_i, let $y_i = g_i^{x_i} \bmod p_i$, where the order of g_i is $q_i > 2^{l+1}$. x_i is the secret key and (y_i, p_i, q_i, g_i) is the public key. $H_i : \{0,1\}^* \to \mathbb{Z}_{q_i}$ is a collision-resistant hash function.

RSign: The signer U_k in the ring R_I signs a message m by executing the algorithm below:

1. Randomly select $\alpha \in \mathbb{Z}_{q_k}$, and compute $c_{k+1} = H_{k+1}(L_I, m, g_k^\alpha \bmod p_k)$.
2. For $i = k+1, \ldots, n_I - 1, 0, 1, \ldots, k-1$, randomly select $s_i \in (-2^{(l-2)/\epsilon}, 2^{(l-2)/\epsilon})$, and compute $c_{i+1} = H_{i+1}(L_I, m, g_i^{s_i} y_i^{c_i} \bmod p_i)$.
3. Compute $s_k = \alpha - x_k c_k \bmod q_k$, where $s_k \in (-2^{(l-2)/\epsilon}, 2^{(l-2)/\epsilon})$. If $s_k \notin (-2^{(l-2)/\epsilon}, 2^{(l-2)/\epsilon})$, properly reselect α and run the Step 1 to 3 again until s_k lies in the right interval. The resulting ring signature is $\sigma_I = (c_0, s_0, \ldots, s_{n_I-1})$.

RVer: For $i = 0, \ldots, n_I - 1$, the verifier computes $e_i = g_i^{s_i} y_i^{c_i} \bmod p_i$, then compute $c_{i+1} = H_{i+1}(L_I, m, e_i)$ if $i \neq n_I - 1$. If $c_0 = H_0(L_I, m, e_{n_I - 1})$, the verifier accepts σ_I as a valid ring signature, reject otherwise.

PRSig: The algorithm is used for converting a full ring signature σ_I to a verifiably encrypted ring signature θ_I. Let $\hat{h} : \{0,1\}^* \rightarrow \{0,1\}^\eta$ be a collision-resistant hash function and the public label $\mathsf{L} = m||L_I$.

1. Compute $w = g_{n_I-1}^{s_{n_I-1}}$ and encrypt s_{n_I-1} by computing

$$\mathsf{u} = \mathsf{g}^{\mathsf{t}}, \qquad \mathsf{e} = \mathsf{y}_1^{\mathsf{t}} \mathsf{h}^{s_{n_I-1}}, \qquad \mathsf{v} = \mathsf{abs}(\mathsf{y}_2 \mathsf{y}_3^{\mathsf{H(u,e,L)}})^{\mathsf{t}}$$

   under Camenisch and Shoup's encryption scheme.
2. Randomly select $r \in \mathbb{Z}_n$, $r_1 \in (-2^{l-2},\ 2^{l-2})$, $r_2 \in (-(n2^\eta)^\epsilon, (n2^\eta)^\epsilon)$ and $r_3 \in (-(\mathsf{n}2^\eta)^\epsilon,\ (\mathsf{n}2^\eta)^\epsilon)$, compute $\hat{w} = h_1^r h_2^{s_{n_I-1}} \bmod n$ and $t_1 = g_{n_I-1}^{r_1}$, $t_2 = h_1^{r_2} h_2^{r_1}$, $\mathsf{u}' = \mathsf{g}^{r_3}$, $\mathsf{e}' = \mathsf{y_1}^{r_3} \mathsf{h}^{r_1}$ and $\mathsf{v}' = (\mathsf{y}_2 \mathsf{y}_3^{\mathsf{H(u,e,L)}})^{r_3}$ in their own groups.
3. Compute $\hat{c} = \hat{h}(L_I, m, w, \hat{w}, \mathsf{u}, \mathsf{e}, \mathsf{v}, g_{n_I-1}, \mathsf{g}, h_1, h_2, t_1, t_2, \mathsf{u}'^2, \mathsf{e}'^2, \mathsf{v}'^2)$ and $v_1 = r_1 - \hat{c} s_{n_I-1}$, $v_2 = r_2 - \hat{c} r$, $v_3 = r_3 - \hat{c}\mathsf{t}$ in $\mathbb{Z}$. The resulting VERS is $\theta_I = (c_0, s_0, ..., s_{n_I-2}, w, \mathsf{u}, \mathsf{e}, \mathsf{v}, \hat{w}, \hat{c}, t_1, t_2, \mathsf{u}', \mathsf{e}', \mathsf{v}',\ v_1, v_2, v_3)$.

PRVer: The verifier first computes $\hat{c}' = \hat{h}(L_I, m, w, \hat{w}, \mathsf{u}, \mathsf{e}, \mathsf{v}, g_{n_I-1}, \mathsf{g}, h_1, h_2,$ $g_{n_I-1}^{v_1} w^{\hat{c}}, h_1^{v_2} h_2^{v_1} \hat{w}^{\hat{c}}, \mathsf{g}^{2v_3} \mathsf{u}^{2\hat{c}}, \mathsf{y_1}^{2v_3} \mathsf{h}^{2v_1} \mathsf{e}^{2\hat{c}}, (\mathsf{y}_2 \mathsf{y}_3^{\mathsf{H(u,e,L)}})^{2v_3} \mathsf{v}^{2\hat{c}})$, and checks whether $\hat{c}' = \hat{c}$ and $-2^{l-1} < v_1 < 2^{l-1}$. If any condition does not hold, outputs the VERS θ_I is invalid, otherwise computes $e_i = g_i^{s_i} y_i^{c_i}$ for $i = 0, \ldots, n_I - 2$ and $e_{n_I-1} = w y_{n_I-1}^{c_{n_I-1}}$, and then computes $c_{i+1} = H_{i+1}(L_I, m, e_i)$ if $i \neq n_I - 1$. If c_0 $= H_0(L_I, m, e_{n_I-1})$, the verifier accepts θ_I, reject otherwise.

Res: After the verifier shows a proof that he has fillfulled his obligation to the signer, the arbitrator decrypts the ciphertext $(\mathsf{u}, \mathsf{e}, \mathsf{v})$ using its secret key $(\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3)$ to extract s_{n_I-1}, and reveals the full ring signature σ_I to the verifier.

4.2 Optimistic Fair Exchange of Ring Signatures

By applying the verifiably encrypted ring signature scheme above, an optimistic fair exchange protocol of ring signatures can easily be set up. Suppose two users U_i and U_j in the rings R_I and R_J respectively exchange their ring signatures on a message m. The optimistic fair exchange protocol proceeds as follows:

1. U_i computes his ring signature σ_I=**RSign**(m, L_I, SK_i), and converts this ring signature into a VERS θ_I=**PRSig**(m, L_I, σ_I, APK) using the arbitrator's public key APK, then sends θ_I to R_J.
2. U_j checks whether **PRVer**(m, L_I, θ_I, APK)= 1. If no, U_j quits, otherwise U_j computes his ring signature σ_J and sends it to R_I.
3. U_i checks whether **RVer**(m, L_J, σ_J)=1, if no, U_i stops the protocol, otherwise U_i sends σ_I to R_J.

4. U_j checks whether $\mathbf{RVer}(m, L_I, \sigma_I)$=1, if yes, U_j accepts this ring signature. If σ_I is invalid or U_j receives nothing from R_I, U_j sends the arbitrator θ_I and σ_J to apply for resolution. The arbitrator first checks whether σ_J is valid, if yes, the arbitrator runs the algorithm $\mathbf{Res}(m, L_I, \theta_I, ASK)$ to recover σ_I, then sends σ_I to R_J and σ_J to R_I. If σ_J is invalid, the arbitrator will send a signal to both R_I and R_J to inform U_i and U_j that the protocol has been terminated.

Note that after Step 1, U_j can decide to carry on the protocol at any time he wants, which might give U_j some advantages. To solve this problem, before the protocol runs, U_i and U_j can set up a time point at which the protocol must be completed.

5 Security Proof

In this session, we prove that our OFE protocol for ring signatures is secure in the multi-user setting under adaptive chosen message, chosen-key and chosen public-key attacks. Let **RS** = (**RKG, RSig, RVer**) denote Abe et al.'s ring signature scheme, **EN**=(**Gen, Enc, Dec**) denote Camenisch-Shoup public-key encryption scheme, and π be a non-interactive zero-knowledge proof showing the proper encryption of a full ring signature. We have the following theorem:

Theorem 1: The proposed optimistic fair exchange of ring signatures is secure, i.e. satisfies Definitions 2-5, if the underlying **RS** *is secure with signer-ambiguity and existential unforgeability against adaptive chosen message and chosen public-key attacks,* **EN** *is secure against adaptive chosen ciphertext attacks (CCA2), and π is a simulation-sound non-interactive zero-knowledge proof.*

Proof. Security against signers: In our OFERS protocol, a valid verifiably encrypted ring signature $\theta = (c_0, s_0, s_1, \cdots, s_{n-2}, w, \mathsf{u}, \mathsf{e}, \mathsf{v}, \hat{w}, \hat{c}, t_1, t_2, \mathsf{u}', \mathsf{e}', \mathsf{v}', v_1, v_2, v_3)$ consists of three parts. The first part $(c_0, s_0, s_1, \cdots, s_{n-2}, w)$ is a 'ring signature', where s_{n-1} is hidden in $w = g_{n-1}^{s_{n-1}}$. The second part $(\mathsf{u}, \mathsf{e}, \mathsf{v})$ is the ciphertext of encrypting s_{n-1} under the arbitrator's public key, where $\mathsf{u} = \mathsf{g}^{\mathsf{t}}$, $\mathsf{e} = \mathsf{y}_1^{\mathsf{t}} \mathsf{h}^{s_{n-1}}$ and $\mathsf{v} = \mathsf{abs}(\mathsf{y}_2 \mathsf{y}_3^{\mathsf{H}(\mathsf{u},\mathsf{e},\mathsf{L})})^{\mathsf{t}}$ for some t. The third part $(\hat{w}, \hat{c}, t_1, t_2, \mathsf{u}', \mathsf{e}', \mathsf{v}', v_1, v_2, v_3)$ provides a non-interactive zero-knowledge proof:

$$\pi = PK\{(s_{n-1}, \mathsf{t}, r) : w = g_{n-1}^{s_{n-1}} \wedge \mathsf{u}^2 = \mathsf{g}^{2\mathsf{t}} \wedge \mathsf{e}^2 = \mathsf{y}_1{}^{2\mathsf{t}} \mathsf{h}^{2s_{n-1}} \wedge \mathsf{v}^2 = (\mathsf{y}_2 \mathsf{y}_3^{\mathsf{H}(\mathsf{u},\mathsf{e},\mathsf{L})})^{2\mathsf{t}}$$
$$\wedge\, \hat{w} = h_1^r h_2^{s_{n-1}} \wedge -2^l < s_{n-1} < 2^l\},$$

which shows that the encrypted s_{n-1} is the same value hidden in w. Suppose an adversary $\mathcal{A}$ breaks the security against signers in our OFERS protocol by forging a VERS $\theta = (c_0, s_0, s_1, \cdots, s_{n-2}, w, \mathsf{u}, \mathsf{e}, \mathsf{v}, \hat{w}, \hat{c}, t_1, t_2, \mathsf{u}', \mathsf{e}', \mathsf{v}', v_1, v_2, v_3)$ w.r.t a public-key list L^* generated by himself, where $w = g_{n-1}^{s_{n-1}}$ but $\mathsf{e} = \mathsf{y}_1^{\mathsf{t}} \mathsf{h}^{s'_{n-1}}$ for

$s'_{n-1} \neq s_{n-1}$. For each public key in L^*, $\mathcal{A}$ may not know the corresponding private key. According to Definition 2, $\mathcal{A}$ wins the game of *security against signers* if and only if the corresponding full ring signature of θ is $\sigma = (c_0, s_0, s_1, ..., s_{n-2}, s_{n-1})$ and $(\mathsf{u}, \mathsf{e}, \mathsf{v})$ is decrypted to get s'_{n-1}, where $s'_{n-1} \neq s_{n-1}$. However, this is infeasible due to the *soundness* of the zero-knowledge proof π. Hence our OFERS protocol is secure against signers if π is a non-interactive zero-knowledge proof (NIZK).

Security against verifiers: Suppose an adversarial verifier $\mathcal{B}$ breaks the security against verifiers in the proposed OFERS protocol. We now construct a distinguisher $\bar{\mathcal{B}}$, who can successfully distinguish the encryption of two messages with the same length of its choice from a challenger in the CCA2 game for Camenisch-Shoup encryption scheme with non-negligible probability. Note that $\bar{\mathcal{B}}$ is allowed to access the decryption oracle O_{Dec} of the encryption scheme. According to Definition 3, $\mathcal{B}$ wins the game of *security against verifiers* if $\mathcal{B}$ produces a valid ring signature σ on a message m under a public-key list L' without asking the resolution oracle O_{Res} any query (m, L', θ). As (m, L', σ) is a successful forgery of $\mathcal{B}$, the situation that $\mathcal{B}$ did not ask any corresponding VERS θ of σ via the partial ring signature signing oracle O_{PRSig} is negligible due to *security against the arbitrator* proved below. Hence we require that $\mathcal{B}$ gets θ from O_{PRSig} here. Now we show how to construct $\bar{\mathcal{B}}$ in detail.

For the given target Camenisch-Shoup encryption scheme **EN**=(**Gen, Enc, Dec**) with the public key APK, the distinguisher $\bar{\mathcal{B}}$ repeatedly executes Abe et al.'s key generation algorithm, **RKG** $\rightarrow \{PK_i, SK_i\}$, to form a public-key list L. Then $\bar{\mathcal{B}}$ sends (APK, L) to $\mathcal{B}$ as the input of the OFERS protocol. Let k be the total number of the queries that $\mathcal{B}$ issues to O_{PRSig}. After arbitrarily selecting j from $\{1, 2, ..., k\}$, $\bar{\mathcal{B}}$ simulates O_{PRSig}'s response to each query (m_i, L_i) issued by $\mathcal{B}$, where $i = 1, 2, ..., k$, $L_i \subseteq L$ and $n_i = |L_i|$, as follows:

1. If $i \neq j$, $\bar{\mathcal{B}}$ signs the message m_i w.r.t L_i using the private key SK_0 to generate a ring signature $\sigma_i = \mathbf{RSig}(m_i, L_i, SK_0) = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-1}})$ and returns a VERS $\theta_i = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-2}}, w_i, \varepsilon_i, \pi_i)$, where $w_i = g_{i_{n_i-1}}^{s_{i_{n_i-1}}}$ and $\varepsilon_i = \mathbf{Enc}_{APK}(s_{i_{n_i-1}})$ under Camenisch-Shoup encryption scheme, and π_i is a NIZK proof showing that ε_i encrypts the same value hidden in w_i, i.e. $s_{i_{n_i-1}}$.
2. If $i = j$, $\bar{\mathcal{B}}$ computes $\sigma_i = \mathbf{RSig}(m_i, L_i, SK_0) = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-1}})$ and chooses a proper $\hat{s}_{i_{n_i-1}}$ in the same interval of $s_{i_{n_i-1}}$ but $s_{i_{n_i-1}} \neq \hat{s}_{i_{n_i-1}}$. Then $\bar{\mathcal{B}}$ sets $\dot{s}_1 = s_{i_{n_i-1}}$ and $\dot{s}_0 = \hat{s}_{i_{n_i-1}}$ and sends $\dot{s}_1$ and $\dot{s}_0$ to its CCA2 challenger. The challenger returns a ciphertext ε_b, which equals either $\mathbf{Enc}_{APK}(s_{i_{n_i-1}})$ or $\mathbf{Enc}_{APK}(\hat{s}_{i_{n_i-1}})$. After that, $\bar{\mathcal{B}}$ returns $\theta_i = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-2}}, w_i, \varepsilon_i, \pi_i)$, where $w_i = g_{i_{n_i-1}}^{s_{i_{n_i-1}}}$, $\varepsilon_i = \varepsilon_b$, and π_i is a *simulated* NIZK proof showing that ε_i encrypts the same value hidden in w_i.

The distinguisher $\bar{\mathcal{B}}$ simulates the resolution oracle O_{Res}'s response to $\mathcal{B}$'s queries (m_i, L_i, θ_i) using the decryption oracle O_{Dec} as follows:

1. If π_i is valid and $L_i \neq L_j$, $\bar{\mathcal{B}}$ asks O_{Dec} to extract the plaintext $s_{i_{n_i-1}}$ from ε_i and returns the ring signature $\sigma_i = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-1}})$ on m_i under L_i.
2. If π_i is valid and $L_i = L_j$, $\bar{\mathcal{B}}$ checks whether $m_i = m_j$. If yes, $\bar{\mathcal{B}}$ aborts the simulation and sends a random bit to its CCA2 challenger. Otherwise, $\bar{\mathcal{B}}$ asks O_{Dec} to extract the plaintext $s_{i_{n_i-1}}$ from ε_i and returns the ring signature $\sigma_i = (c_{i_0}, s_{i_0}, ..., s_{i_{n_i-1}})$ on m_i under L_i.
3. If π_i is invalid, $\bar{\mathcal{B}}$ returns $\mathcal{B}$ a random value.

If $\varepsilon_b = \mathbf{Enc}_{APK}(\hat{s}_{i_{n_i-1}})$ (i.e. $b = 0$), θ_i looks valid but, in fact, $\sigma_i = (c_{i_0}, s_{i_0}, ..., \hat{s}_{i_{n_i-1}})$ is not a valid ring signature because of $\hat{s}_{i_{n_i-1}} \neq s_{i_{n_i-1}}$. The probability of $\mathcal{B}$ forging a valid ring signature on m_j is therefore negligible. If $\varepsilon_b = \mathbf{Enc}_{APK}(s_{i_{n_i-1}})$ (i.e. $b = 1$), ε_j is an valid encryption of $s_{i_{n_i-1}}$ which is a part of a valid ring signature on m_j. The attack environment required by $\mathcal{B}$ is perfectly simulated. Suppose (m, L', σ) is the forgery of $\mathcal{B}$, if $m = m_j$ and $L' = L_j$, $\bar{\mathcal{B}}$ outputs 1 and wins the CCA2 game by indicating that $\dot{s}_1 = s_{i_{n_i-1}}$ is the plaintext of ε_b, otherwise $\bar{\mathcal{B}}$ sends a random bit to the CCA2 challenger. Consequently, if $\mathcal{B}$ wins the game of *security against verifiers* with a non-negligible probability, $\bar{\mathcal{B}}$'s advantage against its CCA2 challenger is also non-negligible. Hence our OFERS protocol is secure against verifiers if the underlying encryption scheme **EN** is CCA2-secure.

Security against the arbitrator: Suppose an adversarial arbitrator $\mathcal{C}$ breaks the security against the arbitrator in the proposed OFERS protocol. We construct a forger $\bar{\mathcal{C}}$ for Abe et al.'s ring signature scheme **RS** = (**RKG, RSig, RVer**) with access to a signing oracle O_{RSig}.

For the initial public-key list L given to the forger $\bar{\mathcal{C}}$, the adversarial arbitrator $\mathcal{C}$ takes L as input and then outputs (ASK^*, APK), where APK is set as the arbitrator's public key for Camenisch-Shoup encryption scheme, and ASK^* is the state information which may not correspond to APK. (ASK^*, APK, L) is the input of the OFERS protocol. After that, $\mathcal{C}$ begins to ask queries to the partial ring signature signing oracle O_{PRSig}, for which the responses can be perfectly simulated by $\bar{\mathcal{C}}$ using O_{RSig}: For any message m_i and any sublist $L'' \subseteq L$, $\bar{\mathcal{C}}$ asks its signing oracle O_{RSig} to get a ring signature σ_i, then encrypts σ_i under APK to get a VERS θ_i and generates the NIZK proof π_i. Finally, $\mathcal{C}$ outputs the forgery (m', σ') such that $\mathbf{RVer}(m', L', \sigma') = 1$ and $(m', L') \notin Query(\mathcal{C}, O_{PRSig})$, which means $\bar{\mathcal{C}}$ never asks O_{RSig} to response a valid ring signature on m' w.r.t L'. In our OFERS protocol, σ' is just the conventional ring signature on m' w.r.t L', so $\bar{\mathcal{C}}$ has succeeded for obtaining σ' as the forgery of the message m' without asking the signing oracle O_{RSig}. It is contradictory to the existential unforgeability of Abe et al.'s ring signature scheme against adaptive chosen message and chosen public-key attacks. Hence our OFERS protocol must be secure against the arbitrator.

Signer ambiguity: Suppose that our OFERS protocol does not meet signer ambiguity, which means that there is an unbound adversary $\mathcal{D}$ can tell which private key SK_s was used to produce a given tuple $(m, L, \theta, \sigma, APK)$ with the probability not equal to $1/|L|$. Then, from $\mathcal{D}$ we now construct an adversary $\bar{\mathcal{D}}$

that breaks signer ambiguity of Abe et al.'s ring signature scheme, which thus leads to a contradiction. For a given initial public-key list L in Abe et al.'s scheme we run the key generation algorithm of Chamenisch-Shoup encryption scheme to get the arbitrator's key pair (ASK, APK). For a target (m, L, σ, APK), $\bar{\mathcal{D}}$ runs **PRSig** algorithm to get θ, i.e. $\theta \leftarrow \mathbf{PRSig}(m, L, \sigma, APK)$. By forwarding $(m, L, \theta, \sigma, APK)$ to $\mathcal{D}$, $\bar{\mathcal{D}}$ just outputs the index returned by $\mathcal{D}$ as its guess which private key was used to issue (m, L, σ, APK). It is easy to see that $\bar{\mathcal{D}}$ breaks the signer-ambiguity of Abe et al.'s ring signature scheme with the exact same probability as $\mathcal{D}$ breaks the signer-ambiguity of our OFERS protocol. □

Remark 2. In the proofs above, we do not give the specific details about the underlying (Abe et al.'s) ring signature scheme and (Camenisch and Shoup's) encryption scheme, as our construction (specified in Section 4) can be extended to a generic scheme, i.e. based on any secure ring signature scheme and encryption scheme, the associated proofs can be obtained by simply adapting the proofs above. In addition, from our proofs we can see that a secure ring signature scheme with signer-ambiguity does not necessarily guarantee an OFERS protocol preserving the same property. The counterexample is very simple: just modify our OFERS protocol such that the VERS θ includes a public key PK_i which indicates that the private key SK_i was used to issue the corresponding ring signature σ. For this scheme, it is not difficult to see that the proofs for the first three properties still hold, but not for signer-ambiguity since, with the reminder PK_i, the adversary can tell with the probability 1 that SK_i was used to issue a tuple $(m, L, \theta, \sigma, APK)$. Further discussions on these two issues will be given in the full version of the paper.

6 Conclusion

In this paper, for achieving better privacy in optimistic fair exchange, we present the first solution of optimistic fair exchange of ring signatures (OFERS) by first formally defining its security model in the multi-user setting under adaptive chosen message, chosen-key, and chosen public-key attacks. We have also proposed a concrete scheme of verifiably encrypted ring signature (VERS) and used it to build an optimistic fair exchange protocol. The proposed scheme is proved to be *secure against signers*, *verifiers* and *the arbitrator* and satisfy the property *signer-ambiguity* under our security definitions. As future work, it is interesting to design efficient OFERS protocols for different types of signatures, such as Abe et al.'s RSA-based ring signatures or mixed-type ring signatures [8], and achieve other more security properties in OFERS, e.g. abuse-freeness.

References

1. Asokan, N., Schunter, M., Waidner, M.: Optimistic protocols for fair exchange. In: CCS 1997 Proceedings of the 4th ACM Conference on Computer and Communications Security, pp. 7–17 (1997)
2. Boneh, D., Gentry, C., Lynn, B.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)

3. Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001)
4. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communication 18, 593–610 (2000)
5. Dodis, Y., Lee, P.J., Yum, D.H.: Optimistic Fair Exchange in a Multi-user Setting. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 118–133. Springer, Heidelberg (2007)
6. 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: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 106–120. Springer, Heidelberg (2008)
7. Bleichenbacher, D.: Chosen Ciphertext Attacks against Protocols Based on the RSA Encryption Standard PKCS #1. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 1–12. Springer, Heidelberg (1998)
8. Abe, M., Ohkubo, M., Suzuki, K.: 1-out-of-n Signatures from a Variety of Keys. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 415–432. Springer, Heidelberg (2002)
9. Ateniese, G.: Verifiable Encryption of Digital Signatures and Appliciation. ACM Transactions on Information and System Security 7, 1–20 (2004)
10. Bellare, M., Goldreich, O.: On Defining Proofs of Knowledge. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 390–420. Springer, Heidelberg (1993)
11. Camenisch, J., Michels, M.: Separability and Efficiency for Generic Group Signature Schemes (Extended Abstract). In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 413–430. Springer, Heidelberg (1999)
12. Barić, N., Pfitzmann, B.: Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 480–494. Springer, Heidelberg (1997)
13. Fujisaki, E., Okamoto, T.: Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 16–30. Springer, Heidelberg (1997)
14. Camenisch, J., Michels, M.: Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 107–122. Springer, Heidelberg (1999)
15. Gennaro, R., Krawczyk, H., Rabin, T.: RSA-Based Undeniable Signatures. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 132–149. Springer, Heidelberg (1997)
16. Naccache, D., Stern, J.: A new public key cryptosystem based on higher resudues. In: 5th ACM Conference on Computer and Communications Security, pp. 59–66 (1998)
17. Okamoto, T., Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 308–318. Springer, Heidelberg (1998)
18. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
19. Camenisch, J., Shoup, V.: Practical Verifiable Encryption and Decryption of Discrete Logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg (2003)
20. Goldwasser, S., Micali, S., Rivest, R.: A digital signature scheme secrue against adaptive chosen-message attacks. SIAM Journal on Computing 17, 281–308 (1988)