symmetry

MDPI

Article

A Generic Framework for Accountable Optimistic Fair Exchange Protocol †

Jia-Ch'ng Loh [1,*], Swee-Huay Heng [1,*] and Syh-Yuan Tan [2,*]

[1] Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
[2] School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK
* Correspondence: jasonlohjc@gmail.com (J.-C.L.); shheng@mmu.edu.my (S.-H.H.); syh-yuan.tan@newcastle.ac.uk (S.-Y.T.)

† This paper is an extended version of our paper published in Su C., Kikuchi H. (eds) Information Security Practice and Experience. ISPEC 2018, held in Tokyo, Japan, in September 2018. Lecture Notes in Computer Science, Springer, Cham, Switzerland, 2018; Volume 11125, pp. 229–309.

Received: 18 December 2018; Accepted: 21 January 2019; Published: 22 February 2019

Abstract: Optimistic Fair Exchange protocol was designed for two parties to exchange in a fair way where an arbitrator always remains offline and will be referred only if any dispute happens. There are various optimistic fair exchange protocols with different security properties in the literature. Most of the optimistic fair exchange protocols satisfy resolution ambiguity where a signature signed by the signer is computational indistinguishable from the one resolved by the arbitrator. Huang et al. proposed the first generic framework for accountable optimistic fair exchange protocol in the random oracle model where it possesses resolution ambiguity and is able to reveal the actual signer when needed. Ganjavi et al. later proposed the first generic framework in the standard model. In this paper, we propose a new generic framework for accountable optimistic fair exchange protocol in the standard model using ordinary signature, convertible undeniable signature, and ring signature scheme as the underlying building blocks. We also provide an instantiation using our proposed generic framework to obtain an efficient pairing-based accountable optimistic fair exchange protocol with short signature.

Keywords: accountability; convertible undeniable signature; optimistic fair exchange; ring signature

1. Introduction

A fair exchange protocol was first designed to overcome the issue of fairness during an exchange between two parties such as contract signing [1,2], digital exchange [3], certified mail [4–6], etc. It is widely accepted that at the end of the exchange protocol, both parties have either received their expected items or none of them have received anything. There are two types of fair exchange protocols, namely, protocols that involve the arbitrator and protocols that do not involve the arbitrator [7]. Protocol that involve an arbitrator can be further divided into three types, namely, inline arbitrator protocol [8,9], online arbitrator protocol [10,11], and offline arbitrator protocol [12–14]. In 1997, the offline arbitrator protocol, which is also known as optimistic fair exchange (OFE) protocol, was introduced by Asokan et al. [12] to overcome the disadvantage of the inline and online arbitrator protocols, where the arbitrator is required to always remain online. At the same time, both parties can never exchange a secret message in a fair manner without leaking some information to the arbitrator. In OFE protocol, the arbitrator always remains offline and is called for resolution if and only if any dispute happens (e.g., one of the parties is cheating or the communication channel is interrupted). Asokan et al.'s OFE protocol was later broken and formally redefined by Dodis and Reyzin [13].

For the rest of the paper, the notion of fair exchange protocol is generally referred to as fair exchange protocol for digital data. Figure 1 illustrates the OFE protocol. At first, the signer generates

a message and partial signature pair (m, σ_p^s) and sends to the verifier. The verifier then checks the validity of (m, σ_p^s) and returns (m, σ^v) to the signer. If everything goes well, the signer will reply (m, σ^s) to the verifier, and the protocol ends. However, if a dispute happens where the verifier sent (m, σ^v) to the signer, but the signer did not reply (m, σ^s) back to the verifier, the verifier can contact the arbitrator to resolve the issue. Figure 2 illustrates the resolution protocol. During the resolution, the verifier first sends (m, σ_p^s) and (m, σ^v) to the arbitrator. Once the arbitrator verifies the validity, the arbitrator resolves (m, σ_p^s) into (m, σ^s) and returns back to the verifier.

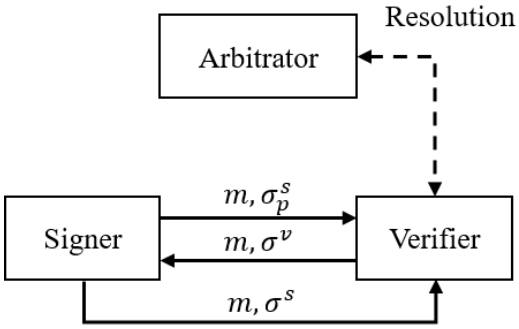

Figure 1. Optimistic Fair Exchange Protocol.

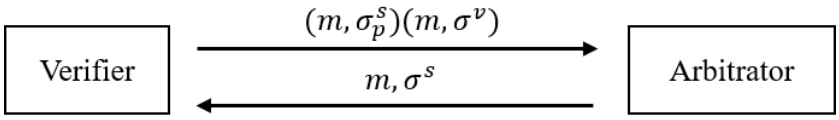

Figure 2. Resolution Protocol.

As the partial signature σ_p is publicly verifiable and non-repudiable, σ_p may not be completely useless to the verifier as σ_p evidently represents the signer's commitment. In this case, an unfair situation may occur for the signer if the verifier does not send out the full signature σ. Suppose that Bob received Alice's σ_p as an offer, Bob may show σ_p to Alice's competitor, and ask for a better offer. If there is a better offer, then Bob may stop running the protocol with Alice which indicates that he is not willing to negotiate, and instead Bob carries out a new run with a better dealer. Bob can repeat the same steps until the best dealer is found. This is undesirable since a fair negotiation is expected. Therefore, the notion of ambiguous optimistic fair exchange (AOFE) was proposed by Huang et al. [15] to solve the unfair situation above. In AOFE protocol, σ_p is non-transferable. More precisely, Bob has the ability to issue partial signatures that are computationally indistinguishable from those issued by Alice. Hence, the verifier will not be able to use the signer's σ_p as evidence of the signer being involved in the exchange protocol.

Although AOFE protocol has managed to solve the issue of the verifier in getting a better offer, in some cases, the parties involved in the exchange should remain anonymous before the exchange has been done. For example, in the event that Apple engages Intel to sign a contract that terminates their agreement. This information can be very valuable to third parties such as stockbrokers or other companies.

In order to overcome the above issue, the notion of perfect ambiguous optimistic fair exchange (PAOFE) was proposed by Wang et al. [16]. PAOFE was constructed by combining AOFE and public key encryption (PKE). It guarantees that σ_p leaks no information about the signer nor the verifier. Although the privacy of the involved parties is protected in a normal run of the PAOFE protocol, the arbitrator can actually gain knowledge while the dispute occurs, when the arbitrator is requested to resolve it. It is very undesirable that in some sensitive applications such as contract signing, where information leakage is not desired at all, the arbitrator is requested to resolve the dispute. Hence, the notion of privacy-preserving optimistic fair exchange (P^2OFE) was proposed by Huang et al. [17] where σ_p does not leak any information about the signer and the verifier even after

the resolution has been executed by the arbitrator. Later, a generic framework for P^2OFE protocols was proposed by Guo et al. [18] using tag-based public key encryption, ordinary signature, and one-time signature scheme.

Based on the above reviews from the notion of ordinary OFE to P^2OFE, it shows that σ can be classified into two types, namely, the actual signature generated by the signer and the resolved signature generated by the arbitrator. An OFE protocol should possess the resolution ambiguity where the actual signature and resolved signature are both computational indistinguishable. However, there is actually a threat that the arbitrator can perform resolution without having any valid proof checking, or the arbitrator might be corrupted by the verifier. Hence, the notion of accountable OFE was proposed by Huang et al. [19] to identify who is the one responsible for σ, and thus it forces the arbitrator and the signer to behave honestly in generating σ.

1.1. Motivation

In cryptography, a scheme is said to be provably secure if breaking the scheme is as hard as breaking the polynomial time hard problem. If the scheme is provably secure using only mathematical hard problems, it is said to be provably secure in the standard model. As the standard model is hard to be achieved, the random oracle model was later introduced by Bellare and Rogaway [20]. An idealistic hash function is used in the random oracle model where the hash function can return any uniformly random value for any input. However, the random oracle model is not preferable during the security proof of a scheme due to the nature of the random oracle being a black-box function.

The first generic framework for accountable OFE protocol in the random oracle model was introduced by Huang et al. [19] where the partial signature is an ordinary signature, and the full signature consists of a partial signature, a random salt, and an undeniable signature along with an *OR*-signature. It possesses resolution ambiguity due to the anonymity of undeniable signature scheme and the witness indistinguishability of *OR*-signature. In order to construct the *OR*-signature in their generic framework, one must use the private key of undeniable signature scheme to generate a signature based on proofs of knowledge (SPK) [21]. Due to the property of SPK, one can generate a proof to either claim or deny an undeniable signature during the stage of revealing the original signer in an accountable OFE protocol. A SPK can be constructed by applying the Fiat-Shamir heuristic [22] to a proof of knowledge where it is a zero-knowledge protocol that allows the signer to convince the verifier that he knows a secret without leaking it [23]. It is known that a SPK that transformed by applying Fiat-Shamir heuristic is secure in the random oracle model [22].

Ganjavi et al. [24] then proposed the first provably secure generic framework for accountable OFE protocol in the standard model. In their generic framework, the partial signature is also an ordinary signature, and the full signature consists of a partial signature and a traceable ring signature. The notion of traceable ring signature was proposed by Fujisaki and Suzuki [25]. It is a variant of ring signature having the additional property to restrict the anonymity of the signer. It possesses two additional security properties, namely, traceability and exculpability. Traceability ensures that the identity of the signer can be traced as long as the signer signs two different messages with respect to the same tag, whereas exculpability ensures that the signer cannot be accused of signing twice with respect to the same tag. However, to the best of our knowledge, there are very few traceable ring signature schemes [25–28] which can be adopted in the construction of accountable OFE protocol following Ganjavi et al.'s proposed generic framework. Hence, it is desirable if there exists another generic framework which is provably secure in the standard model.

1.2. Contribution

In this paper, we present a full version of our recent work [29], proposing another generic framework for accountable OFE protocol. As shown in Table 1, the partial signature in our newly proposed generic framework is also an ordinary signature, and the full signature is an intermediate solution between Huang et al. and Ganjavi et al.'s generic frameworks, where it consists of a partial

signature, a convertible undeniable signature, and a ring signature. There are two types of convertible undeniable signature scheme, namely, selectively convertible and universally convertible. Our generic framework requires the former which allows the signer to convert only a specific undeniable signature into a universally verifiable one. We show that the proposed generic framework is secure in the standard model under multi-user setting and chosen-key model as long as the underlying schemes satisfy certain security properties. We then exhibit an efficient pairing-based accountable OFE protocol with short signature as a concrete example following our proposed generic framework. Similar to Ganjavi et al.'s approach, we aim to construct an efficient accountable OFE protocol. We select the short signature scheme proposed by Boneh et al. [30] as the ordinary signature scheme with the combination of convertible undeniable signature scheme proposed by Li et al. [31] and ring signature scheme proposed by Shim [32], we manage to obtain an efficient pairing-based accountable OFE protocol with short signature. More specifically, the public and private key pair from Li et al.'s convertible undeniable signature scheme can be shared with Boneh et al.'s short signature scheme and Shim's ring signature scheme, though the derived protocol is only provably secure in the random oracle model.

Table 1. A comparison of the Generic Frameworks for Accountable optimistic fair exchange (OFE) Protocol.

Generic Framework	Partial Signature σ_p	Full Signature σ	Proof π	Standard Model	Random Oracle Model
Huang et al. [19]	OS	σ_p, US, r, *OR*-Signature	SPK	×	√
Ganjavi et al. [24]	OS	σ_p, TRS	TRS	√	√
Proposed	OS	σ_p, CUS, RS	token	√	√

r: Random salt; OS: Ordinary signature; US: Undeniable signature; RS: Ring signature; CUS: Convertible undeniable signature; TRS: Traceable ring signature; SPK: Signature based on proofs of knowledge.

1.3. Organisation of the Paper

The organisation of the paper is as follows. In Section 2, we recall the formal definitions and security models of accountable OFE protocol in the multi-user setting and chosen-key model. In Section 3, we provide a brief review on the notion of bilinear pairings. We also recall the definitions and security models of ordinary signature, convertible undeniable signature, and ring signature scheme which are served as the underlying building blocks for the proposed generic framework. In Section 4, we propose a new generic framework for accountable OFE protocol and provide its security analysis in the standard model. In Section 5, we provide an instantiation of an efficient pairing-based accountable OFE protocol with short signature. Finally, we conclude this paper in Section 6.

2. Definitions and Security Models of Accountable Optimistic Fair Exchange Protocol

In this section, we recall the formal definitions and security models of accountable OFE protocol in the muilti-user setting and chosen-key model which formalised by Huang et al. [19]. The security model of OFE protocol is setup-driven if the initial key registration needs to be done between the signer and the arbitrator, and the model is setup-free if that is not required. Since most of the existing exchange protocols consider more than one signer in the system, an OFE protocol should be applicable to multi-user setting, but items are exchanged between one signer and one verifier. More precisely, a multi-user setting OFE protocol consists of many signers and many verifiers along with only one arbitrator [33]. As previous works only considered the certified-key model, Huang et al. [14] then proposed a secure OFE protocol in the multi-user setting and chosen-key model. In contrast to the certified-key model, the adversary in chosen-key model is able to make queries with respect to the public key even without showing the knowledge of the private key.

2.1. Accountable OFE Protocol

An accountable OFE protocol consists of the following algorithms:

- $\boldsymbol{PMGen}$: On input a security parameter 1^k, it outputs a public parameter PM.
- $\boldsymbol{Setup^A}$: On input PM, it generates an arbitrator's public and private key pair (APK, ASK).
- $\boldsymbol{Setup^U}$: On input PM, it generates a user's public and private key pair (UPK_i, USK_i).
- $\boldsymbol{PSign}$: On input a message m and (USK_i, APK), it generates a partial signature σ_p.
- $\boldsymbol{PVer}$: On input $(m, \sigma_p, UPK_i, APK)$, it validates (m, σ_p) and outputs "1" if σ_p is valid on UPK_i or "0" otherwise.
- $\boldsymbol{Sign}$: On input $(m, \sigma_p, USK_i, APK)$, it generates a full signature σ.
- $\boldsymbol{Ver}$: On input (m, σ, UPK_i, APK), it validates (m, σ) under (UPK_i, APK) and outputs "1" if σ is valid or "0" otherwise.
- $\boldsymbol{Res}$: On input $(m, \sigma_p, ASK, UPK_i)$, it resolves σ_p by first checking its validity. If σ_p is valid on UPK_i, it generates a full signature σ or outputs " $\perp$ " otherwise.
- $\boldsymbol{Prove^A}$: On input $(m, \sigma, UPK_i, APK, ASK)$, it generates an arbitrator proof π^A that can claim or deny whether σ was generated by using APK.
- $\boldsymbol{Prove^U}$: On input $(m, \sigma, UPK_i, APK, USK_i)$, it generates a user proof π^U that can claim or deny whether σ was generated by using UPK_i.
- $\boldsymbol{Open}$: On input $(m, \sigma, UPK_i, APK, \pi)$, it first validates (m, σ) under (UPK_i, APK). It then outputs "UPK_i" if π can prove σ is generated by the algorithm $\boldsymbol{Sign}$ or "APK" if σ is generated by the algorithm $\boldsymbol{Res}$. Otherwise, it outputs " $\perp$ " which indicates π is invalid and it cannot be opened.

Correctness: The following algorithms will always output "1" if σ is generated correctly. If σ is a valid on (UPK_i, APK) and π is generated correctly, the algorithm $Open$ will always output either "UPK_i" or "APK".

$$
\begin{aligned}
&-\boldsymbol{PVer}(m, \boldsymbol{PSign}(m, USK_i, APK), UPK_i, APK) = \text{“1”}\\
&-\boldsymbol{Ver}(m, \boldsymbol{Sign}(m, \boldsymbol{PSign}(m, USK_i, APK), USK_i, APK), UPK_i, APK) = \text{“1”}\\
&-\boldsymbol{Ver}(m, \boldsymbol{Res}(m, PSign(m, USK_i, APK), UPK_i, ASK), UPK_i, APK) = \text{“1”}\\
&-\boldsymbol{Open}(m, \sigma, UPK_i, APK, \boldsymbol{Prove^A}(m, \sigma, UPK_i, APK, ASK)) = \text{“}UPK_i\text{”}or\text{“}APK\text{”}\\
&-\boldsymbol{Open}(m, \sigma, UPK_i, APK, \boldsymbol{Prove^U}(m, \sigma, UPK_i, APK, USK_i)) = \text{“}UPK_i\text{”}or\text{“}APK\text{”}
\end{aligned}
$$

2.2. Accessible Oracles

The following oracles are all the accessible oracles that define for an adversary $\mathcal{A}$ in the accountable OFE protocol.

- Partial Sign Oracle $\mathcal{O}_{PSign}$: On input (m, UPK_i), it runs $\boldsymbol{PSign}(m, USK_i, APK) \rightarrow \sigma_p$ and returns σ_p as a partial signature.
- Full Sign Oracle $\mathcal{O}_{Sign}$: On input (m, σ_p, UPK_i), it runs $\boldsymbol{Sign}(m, \sigma_p, USK_i, APK) \rightarrow \sigma$ and returns σ as a full signature.
- Resolution Oracle $\mathcal{O}_{Res}$: On input (m, σ_p, UPK_i), it runs $\boldsymbol{Res}(m, \sigma_p, ASK, UPK_i) \rightarrow \sigma$ and returns σ as a resolved signature.
- Arbitrator Prove Oracle $\mathcal{O}_{Prove^A}$: On input (m, σ) under (UPK_i, APK), it runs $\boldsymbol{Prove^A}(m, \sigma, UPK_i, APK, ASK) \rightarrow \pi^A$ and returns π^A as an arbitrator proof.
- User Prove Oracle $\mathcal{O}_{Prove^U}$: On input (m, σ) under (UPK_i, APK), it runs $\boldsymbol{Prove^U}(m, \sigma, UPK_i, APK, USK_i) \rightarrow \pi^U$ and returns π^U as a user proof.

2.3. Security Properties

An accountable OFE protocol possesses resolution ambiguity, accountability, security against signers, security against verifiers, and security against arbitrator. Its security models in the multi-user setting and chosen-key model are defined as the game between a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

2.3.1. Resolution Ambiguity

A full signature σ generated by the signer or resolved by the arbitrator should be computationally indistinguishable. Note that this security model is referred from the transparent third party property as defined by [19].

- **Phase 1**: $\mathcal{C}$ runs $\boldsymbol{PMGen}(1^k) \rightarrow PM$ and $\boldsymbol{Setup}^{\boldsymbol{A}}(PM) \rightarrow (APK, ASK)$. $\mathcal{C}$ then passes APK to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ can make queries to all oracles defined in Section 2.2. At the end, $\mathcal{A}$ outputs a challenge message and partial signature pair $(\hat{m}, \hat{\sigma}_p)$ with the restriction that $\boldsymbol{PVer}(\hat{m}, \hat{\sigma}_p, UPK_i, APK) = 1$.
- **Phase 3**: $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$ and generates a challenge signature $\hat{\sigma}$. If $b = 0$, $\hat{\sigma} = \boldsymbol{Sign}(\hat{m}, \hat{\sigma}_p, USK_i, APK)$. Otherwise, $\hat{\sigma} = \boldsymbol{Res}(\hat{m}, \hat{\sigma}_p, ASK, UPK_i)$.
- **Phase 4**: Once $\mathcal{A}$ receives $\hat{\sigma}$, $\mathcal{A}$ can still continue to make queries to all oracles with the restriction that $(\hat{m}, \hat{\sigma})$ has never been queries to $\mathcal{O}_{Prove^A}$ or $\mathcal{O}_{Prove^U}$. At the end, $\mathcal{A}$ outputs the guess b'. $\mathcal{A}$ wins the game if $b' = b$.

Definition 1. *An OFE protocol is $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-resolution ambiguous if no PPT $\mathcal{A}$ can have success probability more than $\varepsilon + \frac{1}{2}$ in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, q_{Res} queries to $\mathcal{O}_{Res}$, q_{Prove^A} queries to $\mathcal{O}_{Prove^A}$, and q_{Prove^U} queries to $\mathcal{O}_{Prove^U}$ in time t.*

2.3.2. Accountability

An OFE protocol possesses accountability if it satisfies three types of accountability as follows [19]:

- **Type I**: It is impossible for a dishonest signer to produce a full signature σ that can be proven as an output of the algorithm $\boldsymbol{Res}$.
  - **Phase 1**: $\mathcal{C}$ runs $\boldsymbol{PMGen}(1^k) \rightarrow PM$ and $\boldsymbol{Setup}^{\boldsymbol{A}}(PM) \rightarrow (APK, ASK)$. $\mathcal{C}$ then passes APK to $\mathcal{A}$.
  - **Phase 2**: $\mathcal{A}$ can make queries to all oracles defined in Section 2.2. At the end, $\mathcal{A}$ chooses a challenge user's public key $U\hat{P}K$ and passes it to $\mathcal{C}$.
  - **Phase 3**: $\mathcal{A}$ continues to make queries to $\mathcal{O}_{Res}$ and $\mathcal{O}_{Prove^A}$ only as $\mathcal{C}$ does not know $U\hat{S}K$.
  - **Phase 4**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ that is valid on $(U\hat{P}K, APK)$ and a proof $\hat{\pi}$ with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Res}$. $\mathcal{A}$ wins the game if $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, U\hat{P}K, APK, \hat{\pi}) =$ "APK".

  Definition 2. *An OFE protocol is $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-type I accountable if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, q_{Res} queries to $\mathcal{O}_{Res}$, q_{Prove^A} queries to $\mathcal{O}_{Prove^A}$, and q_{Prove^U} queries to $\mathcal{O}_{Prove^U}$ in time t.*

- **Type II**: It is impossible for a dishonest arbitrator to resolve a full signature σ that can be proven as an output of the algorithm $\boldsymbol{Sign}$.
  - **Phase 1**: $\mathcal{A}$ chooses a challenge arbitrator's public key $A\hat{P}K$ and passes it to $\mathcal{C}$.
  - **Phase 2**: $\mathcal{A}$ can make queries to all oracles defined in Section 2.2 except $\mathcal{O}_{Res}$ and $\mathcal{O}_{Prove^A}$ due to $\mathcal{C}$ does not have the knowledge of $A\hat{S}K$.
  - **Phase 3**: $\mathcal{A}$ outputs a valid $(\hat{m}, \hat{\sigma})$ on $(UPK_i, A\hat{P}K)$ and a proof $\hat{\pi}$ with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Sign}$. $\mathcal{A}$ wins the game if and only if $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, UPK_i, A\hat{P}K, \hat{\pi}) =$ "UPK_i".

  Definition 3. *An OFE protocol is $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-type II accountable if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, and q_{Prove^U} queries to $\mathcal{O}_{Prove^U}$ in time t.*

- **Type III**: It is impossible for the signer and the arbitrator to both claim or deny a valid full signature σ.

- **Phase 1**: $\mathcal{C}$ runs $\boldsymbol{PMGen}(1^k) \rightarrow PM$. $\mathcal{A}$ is then given PM to run both $\boldsymbol{Setup^U}(PM) \rightarrow (U\hat{P}K, U\hat{S}K)$ and $\boldsymbol{Setup^A}(PM) \rightarrow (A\hat{P}K, A\hat{S}K)$.
- **Phase 2**: $\mathcal{A}$ outputs a valid $(\hat{m}, \hat{\sigma})$ on $(U\hat{P}K, A\hat{P}K)$ and two proofs $(\hat{\pi}^U, \hat{\pi}^A)$. $\mathcal{A}$ wins the game if and only if either one of the following statements holds:
  1. $\hat{\sigma}$ is both claimed by the signer and the arbitrator. Such that
     $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, U\hat{P}K, A\hat{P}K, \hat{\pi}^U) \rightarrow U\hat{P}K \wedge$
     $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, U\hat{P}K, A\hat{P}K, \hat{\pi}^A) \rightarrow A\hat{P}K$
  2. $\hat{\sigma}$ is both denied by the signer and the arbitrator. Such that
     $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, U\hat{P}K, A\hat{P}K, \hat{\pi}^U) \rightarrow A\hat{P}K \wedge$
     $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, U\hat{P}K, A\hat{P}K, \hat{\pi}^A) \rightarrow U\hat{P}K$

Definition 4. *An OFE is (t, ε)-type III accountable if no PPT $\mathcal{A}$ can have success probability more than ε in its game in time t.*

2.3.3. Security against Signers

It is impossible for a dishonest signer to produce a valid partial signature σ_p which cannot be resolved by the arbitrator using $\boldsymbol{Res}$ [24].

- **Phase 1**: $\mathcal{C}$ runs $\boldsymbol{Setup^A}(PM) \rightarrow (APK, ASK)$ and passes APK to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ can make queries to $\mathcal{O}_{Res}$.
- **Phase 3**: $\mathcal{A}$ outputs a challenge message and partial signature pair $(\hat{m}, \hat{\sigma}_p)$ on UPK_i. $\mathcal{A}$ wins the game if $\boldsymbol{PVer}(\hat{m}, \hat{\sigma}_p, U\hat{P}K, APK) = 1 \wedge \boldsymbol{Ver}(\hat{m}, \boldsymbol{Res}(\hat{m}, \hat{\sigma}_p, ASK, U\hat{P}K), U\hat{P}K, APK) = 0$.

Definition 5. *An OFE protocol is $(t, q_{Res}, \varepsilon)$-secure against signers if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_{Res} queries to $\mathcal{O}_{Res}$ in time t.*

2.3.4. Security against Verifiers

It is impossible for a dishonest verifier to produce a valid full signature σ without the assistance from the signer or the arbitrator. The security model is referred from [24]. Note that we allow $\mathcal{A}$ to access $\mathcal{O}_{Sign}$ as we want to simulate the scenario that a dishonest verifier can forge a full signature on either the signer or the arbitrator.

- **Phase 1**: $\mathcal{C}$ first runs $\boldsymbol{PMGen}(1^k) \rightarrow PM$ and both $\boldsymbol{Setup^U}(PM) \rightarrow (UPK_i, USK_i)$ and $\boldsymbol{Setup^A}(PM) \rightarrow (APK, ASK)$. $\mathcal{A}$ is then given (UPK_i, APK).
- **Phase 2**: $\mathcal{A}$ can make queries to $\mathcal{O}_{PSign}$, $\mathcal{O}_{Sign}$, and $\mathcal{O}_{Res}$.
- **Phase 3**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ on (UPK_i, APK) with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Sign}$ or $\mathcal{O}_{Res}$. $\mathcal{A}$ wins the game if $\boldsymbol{Ver}(\hat{m}, \hat{\sigma}, UPK_i, APK) = 1$.

Definition 6. *An OFE protocol is $(t, q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-secure against verifiers if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, and q_{Res} queries to $\mathcal{O}_{Res}$ in time t.*

2.3.5. Security against Arbitrator

It is impossible for a dishonest arbitrator to produce a valid σ without having the corresponding σ_p from the signer [24].

- **Phase 1**: $\mathcal{C}$ runs $\boldsymbol{PMGen}(1^k) \rightarrow PM$ and passes to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ runs $\boldsymbol{Setup^A}(PM) \rightarrow (APK, ASK)$ and sends APK to $\mathcal{C}$.
- **Phase 3**: $\mathcal{A}$ can make queries to $\mathcal{O}_{PSign}$.
- **Phase 4**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ on (UPK_i, APK) with the restriction that $(\hat{m}, UPK_i)$ has not been a query to $\mathcal{O}_{PSign}$. $\mathcal{A}$ wins the game if $\boldsymbol{Ver}(\hat{m}, \hat{\sigma}, UPK_i, APK) = 1$.

Definition 7. *An OFE protocol is $(t, q_{PSign}, \varepsilon)$-secure against arbitrator if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$ in time t.*

2.3.6. Security in the Multi-User Setting and Chosen-Key Model

A secure OFE protocol in the multi-user setting and chosen-key model should satisfy the following properties, namely, security against signers, security against verifiers, and security against arbitrator as defined in Sections 2.3.3–2.3.5 respectively [33,34].

Definition 8. *An accountable OFE protocol is secure in the multi-user setting and chosen-key model if it is accountable, secure against signers, secure against verifiers, and secure against arbitrator.*

3. Preliminaries

In this section, we first provide a brief review on the notion of bilinear pairings. We then review some variants of digital signature scheme such as ordinary signature, convertible undeniable signature, and ring signature scheme. We also review their respective security models.

3.1. Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be cyclic groups of prime order p and two generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. The map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map which satisfies the following properties [35]:

- Bilinearity: for all $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, and $(a, b) \in \mathbb{Z}_p$, we have $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.
- Non-degeneracy: if (g_1, g_2) is a generator of $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\hat{e}(g_1, g_2)$ is a generator of $\mathbb{G}_T$, which also implies $\hat{e}(g_1, g_2) \neq 1$.
- Computability: there exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.

3.2. Ordinary Signature Scheme

Ordinary signature is a publicly verifiable digital signature. The message and signature pair can be verified as long as the signer's public key is known. It was formalised by Goldwasser et al. [36] with the following three algorithms:

- ***KeyGen***: On input security parameter 1^k, it outputs a public and private key pair (pk, sk).
- ***Sign***: On input a message and private key (m, sk), it outputs an ordinary signature σ^{os}.
- ***Verify***: On input (m, σ^{os}, pk), it outputs "1" if σ^{os} is valid and outputs "0" otherwise.

Correctness. Every ordinary signature generated in a correct way is always accepted to be a valid signature, such that $\boldsymbol{Verify}(m, Sign(m, sk), pk) \to$ "1".

Unforgeability

Unforgeability ensures that there is no computational way to forge a valid ordinary signature on the public key pk. Its security model is defined as the following game between a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenge $\mathcal{C}$ [36].

- **Setup**: $\mathcal{C}$ runs $\boldsymbol{KeyGen}(1^k) \to (pk, sk)$, then $\mathcal{A}$ is given pk.
- **Queries**: $\mathcal{A}$ can query the Sign Oracle $\mathcal{O}_S$: On input a message m, it outputs a signature σ^{os} that is valid on pk.
- **Output**: At the end, $\mathcal{A}$ is required to output a challenge message and signature pair $(\hat{m}, \hat{\sigma}^{os})$ that is valid on pk, with the restriction that $\hat{m}$ has not been a query to $\mathcal{O}_S$ before.

Definition 9. *An ordinary signature scheme is (t, q_S, ε)-existential unforgeable against chosen message attack (EUF-CMA) if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_S queries to $\mathcal{O}_S$ in time t.*

3.3. Convertible Undeniable Signature

An undeniable signature is a special featured digital signature proposed by Chaum and van Antwerpen [37] which is only verifiable with the help of the signer. Unlike ordinary digital signature, undeniable signature has a distinctive feature, i.e., without the help of the signer, the verifier will not be able to verify the validity of the undeniable signature. The notion of convertible undeniable signature was proposed by Boyar et al. [38] which is an extension of undeniable signature that allows the signer to transform an undeniable signature into a universally verifiable ordinary digital signature. There are two types of convertible undeniable signature, namely, selectively convertible and universally convertible. The selectively convertible undeniable signature allows the signer to convert only a specific undeniable signature into a universally verifiable one by releasing a token, and the universally convertible undeniable signature allows the signer to release a universal token that can publicly verify every undeniable signature. A convertible undeniable signature scheme has the following algorithms:

- ***KeyGen***: On input a security parameter, 1^k, outputs a signer public and private key pair (pk, sk).
- ***Sign***: On input a message and a signer private key, (m, sk), outputs an undeniable signature σ^{us}.
- ***Confirmation/Disavowal Protocol***: An interactive protocol that runs between the signer and the verifier on common input (pk, m, σ^{us}). The signer uses sk to check the validity of σ^{us}, the output is a non-transferable proof $(\text{“}Accept\text{”}/\text{“}Deny\text{”})$ that shows σ^{us} is valid/invalid on (m, pk).
- ***SConvert***: On input (sk, m, σ^{us}), it computes a selective token π^S which can be used to publicly verify (m, σ^{us}) on pk.
- ***SVerify***: On input $(pk, m, \sigma^{us}, \pi^S)$, it outputs " $\perp$ " if π^S is an invalid token on pk. It outputs "1" if (m, σ^{us}, pk) is a valid signature and outputs "0" otherwise.

Completeness and ***Soundness***. ***Completeness*** can be defined as a valid (invalid) signature that can always be proven valid (invalid) and ***Soundness*** can be defined as a valid (invalid) signature that cannot be proven as invalid (valid). The following two cases describe their definitions:

1. If σ^{us} is valid on pk, then
   - $\textbf{\textit{Confirmation/Disavowal Protocol}}(m, \sigma^{us}, pk, sk) \rightarrow \text{“}Accept\text{”}$
   - $\textbf{\textit{SVerify}}(m, \sigma^{us}, pk, \textbf{\textit{SConvert}}(m, \sigma^{us}, sk)) \rightarrow \text{“1”}$
   - $\textbf{\textit{UVerify}}(m, \sigma^{us}, pk, \textbf{\textit{UConvert}}(sk)) \rightarrow \text{“1”}$
2. Or else, if σ^{us} is invalid on pk, then
   - $\textbf{\textit{Confirmation/Disavowal Protocol}}(m, \sigma^{us}, pk, sk) \rightarrow \text{“}Deny\text{”}$
   - $\textbf{\textit{SVerify}}(m, \sigma^{us}, pk, \textbf{\textit{SConvert}}(m, \sigma^{us}, sk)) \rightarrow \text{“0”}$
   - $\textbf{\textit{UVerify}}(m, \sigma^{us}, pk, \textbf{\textit{UConvert}}(sk)) \rightarrow \text{“0”}$

3.3.1. Unforgeability

The same as in Section 3.2, it implies the inability to forge an undeniable signature. Its security model is defined as the following game between a PPT $\mathcal{A}$ and $\mathcal{C}$ in the undeniable signature setting.

- **Setup**: $\mathcal{C}$ runs $\textbf{\textit{KeyGen}}(1^k) \rightarrow (pk, sk)$, then $\mathcal{A}$ is given pk.
- **Queries**: $\mathcal{A}$ is allowed to make queries to the following oracles:
  - Sign Oracle $\mathcal{O}_S$: On input a message m, it outputs an undeniable signature σ^{us} that is valid on pk.
  - Confirmation/Disavowal Oracle $\mathcal{O}_{CD}$: On input any message and signature pair (m, σ^{us}), it runs the protocol with $\mathcal{A}$ and outputs a non-transferable proof to show the validity of σ^{us}.
  - (For convertible schemes only) SConvert Oracle $\mathcal{O}_{SC}$: On input a message and signature pair (m, σ^{us}), it outputs a selective token π^S.

- **Output**: At the end, $\mathcal{A}$ is required to output a challenge message and undeniable signature pair $(\hat{m}, \hat{\sigma}^{us})$, with the restriction that $\hat{m}$ has not been a query to $\mathcal{O}_S$. If the scheme is convertible, pk must not have been queried to $\mathcal{O}_{UC}$. $\mathcal{A}$ wins the game if $(\hat{m}, \hat{\sigma}^{us})$ is valid on pk.

Definition 10. *An undeniable signature, convertible undeniable signature, or designated confirmer signature scheme is $(t, q_S, q_{CD}, q_{SC}, q_{UC}, \varepsilon)$-EUF-CMA if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_S queries to $\mathcal{O}_S$, q_{CD} queries to $\mathcal{O}_{CD}$, q_{SC} queries to $\mathcal{O}_{SC}$, and q_{UC} queries to $\mathcal{O}_{UC}$ in time t.*

3.3.2. Anonymity

The notion of anonymity was proposed by Galbraith and Mao [39]. This security property requires that given a valid message and signature pair and two possible signers' public keys (pk_0, pk_1), there is no computational way to decide who the real signer is. Huang et al. [40] later further studied the anonymity in order to cover the convertible schemes. Its security model is defined as the following game between a PPT $\mathcal{A}$ and $\mathcal{C}$.

- **Setup:** $\mathcal{C}$ first runs $\boldsymbol{KeyGen}(1^k) \rightarrow (sk_0, pk_0)$ and $\boldsymbol{KeyGen}(1^k) \rightarrow (sk_1, pk_1)$ and sends (pk_0, pk_1) to $\mathcal{A}$.
- **Queries I:** Same as in Section 3.3.1.
- **Output I**: At some point, $\mathcal{A}$ outputs a challenge message $\hat{m}$ to request a challenge signature $\hat{\sigma}^{us}$. If the scheme is deterministic, $\hat{m}$ is restricted where it has not been submitted to $\mathcal{O}_S$ during **Queries I**. $\mathcal{C}$ responds by randomly choosing $b \in \{0, 1\}$ and generates a challenge signature $\hat{\sigma}^{us} = \boldsymbol{Sign}_{sk_b}(\hat{m})$ that is valid on either pk_0 or pk_1.
- **Queries II**: Once $\mathcal{A}$ obtains $\hat{\sigma}^{us}$, $\mathcal{A}$ can continue making queries to the accessible oracles as in **Queries I**. If the scheme is deterministic, $\hat{m}$ is restricted to be submitted to $\mathcal{O}_S$. An additional restriction is added where any $(\hat{m}, \cdot)$ in the equivalence class of $(\hat{m}, \hat{\sigma}^{us})$ is not allowed to submit to $\mathcal{O}_{CD}$ (and $\mathcal{O}_{SC}$ if the scheme is convertible).
- **Output II:** $\mathcal{A}$ outputs a guess b' and wins the game if $b' = b$.

Definition 11. *An undeniable signature, convertible undeniable signature, or designated confirmer signature scheme is $(t, q_S, q_{CD}, q_{SC}, q_{UC}, \varepsilon)$-anonymous if no PPT $\mathcal{A}$ can have success probability more than $\varepsilon + \frac{1}{2}$ in its game with at most q_S queries to $\mathcal{O}_S$, q_{CD} queries to $\mathcal{O}_{CD}$, q_{SC} queries to $\mathcal{O}_{SC}$, and q_{UC} queries to $\mathcal{O}_{UC}$ in time t.*

3.4. Ring Signature

Ring signature was introduced by Rivest et al. [41]. It is a group-oriented signature with the anonymity property where the signer can sign on behalf of a group of members, and the ring signature is publicly verifiable without revealing the actual signer. A ring signature scheme consists of the following three algorithms:

- $\boldsymbol{KeyGen}$: On input 1^k, it outputs a public and private key pair (pk, sk).
- $\boldsymbol{Sign}$: On input a message, a private key, and a list of public keys (m, sk, PK_L) where $PK_L = (pk_1, ..., pk_n)$ with n members, it outputs a ring signature σ^{rs}.
- $\boldsymbol{Verify}$: On input (m, σ^{rs}, PK_L), it outputs "1" if σ^{rs} is valid and output "0" otherwise.

Correctness. Every ring signature that generated in a correct way can always be accepted with the equation $\boldsymbol{Verify}(m, \boldsymbol{Sign}(m, sk, PK_L), PK_L) =$ "1".

3.4.1. Unforgeability

This security property ensures that there is no computational way to forge a ring signature with only the knowledge of a list of public keys $PK_L = (pk_i, ..., pk_n)$ of n members. Its security model is defined as the following game between a PPT $\mathcal{A}$ and $\mathcal{C}$ [42].

- **Setup**: $\mathcal{C}$ runs $KeyGen(1^k)$ for n times to generate n public and private key pair $((pk_i, sk_i), ..., (pk_n, sk_n))$, where n is the number of members. $\mathcal{A}$ is given $PK_L = (pk_i, ..., pk_n)$.
- **Queries**: $\mathcal{A}$ can query the Sign Oracle $\mathcal{O}_{Sign}$: On input (m, PK_L^*, e), where $PK_L^* \in PK_L$ is a sub list of members within PK_L and e is a selected member. It then runs $\boldsymbol{Sign}(m, sk_e, PK_L^*)$ to produce a ring signature σ^{rs} to $\mathcal{A}$.
- **Output**: At the end, $\mathcal{A}$ is required to output a challenge message and ring signature pair $(\hat{m}, \hat{\sigma}^{rs})$ on a challenge sub list of members $\hat{PK}_L$ with the restriction that $\hat{m}$ has not been a query to $\mathcal{O}_{Sign}$ before. $\mathcal{A}$ wins the game if $\boldsymbol{Verify}(\hat{m}, \hat{\sigma}^{rs}, \hat{PK}_L) =$ "1"

Definition 12. *A ring signature scheme is (t, q_S, ε)-existential unforgeable against chosen subring attack (EUF-CSA) if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_S queries to $\mathcal{O}_S$ in time t.*

3.4.2. Anonymity

The definition of anonymity for ring signature scheme can be phrased in either a computational or an unconditional sense [43]. This security property requires that given a valid (m, σ^{rs}) and two possible signers' public keys (pk_0, pk_1), there is no computational way to decide who the real signer is.

- **Setup**: Same as in Section 3.4.1.
- **Queries**: Same as in Section 3.4.1.
- **Output**: At the end, $\mathcal{A}$ is required to output a challenge message and a sub list of members $(\hat{m}, \hat{PK}_L)$ and two distinct indices $(e_0, e_1) \in \{1, ..., n\}$ such that $(pk_{e_0}, pk_{e_1}) \in \hat{PK}_L$. $\mathcal{C}$ then chooses $b \in \{0, 1\}$ randomly and computes a challenge ring signature $\hat{\sigma}^{rs} = \boldsymbol{Sign}(\hat{m}, sk_{e_b}, \hat{PK}_L)$. $\mathcal{A}$ is given $\hat{\sigma}^{rs}$ and is required to output a guess b'. $\mathcal{A}$ wins the game if $b' = b$.

Definition 13. *A ring signature scheme is (t, q_S, ε)-anonymous with respect to adversarially chosen keys if no PPT $\mathcal{A}$ can have success probability more than ε in its game with at most q_S queries to $\mathcal{O}_S$ in time t.*

4. Generic Transformation

4.1. Generic Framework

We propose a generic framework for accountable OFE protocol using ordinary signature, convertible undeniable signature, and ring signature scheme as the underlying building blocks. The partial signature is an ordinary signature, $\sigma_p = \sigma^{os}$, and the full signature consists of a partial signature, a convertible undeniable signature, and a ring signature, $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$. Let OS $= (\boldsymbol{KeyGen}, \boldsymbol{Sign}, \boldsymbol{Verify})$ be an ordinary signature scheme, CUS $= (\boldsymbol{KeyGen}, \boldsymbol{Sign}, \boldsymbol{Confirmation/Disavowal\ Protocol}, \boldsymbol{SConvert}, \boldsymbol{SVerify})$ be a convertible undeniable signature scheme, and RS $= (\boldsymbol{KeyGen}, \boldsymbol{Sign}, \boldsymbol{Verify})$ be a ring signature scheme. We need a hash function $H : \{0, 1\}^* \to \mathcal{M}$, where $\mathcal{M}$ is the message space. An accountable OFE protocol consists of the following algorithms:

- $\boldsymbol{PMGen}$: On input the security parameter 1^k, it generates the public parameters PM needed for the ordinary signature, convertible undeniable signature, and ring signature scheme.
- $\boldsymbol{Setup^A}$: On input PM, it runs CUS.$\boldsymbol{KeyGen}(1^k) \to (apk^{us}, ask^{us})$ and RS.$\boldsymbol{KeyGen}(1^k) \to (apk^{rs}, ask^{rs})$ to compute an arbitrator public and private key pair $(APK, ASK) = ((apk^{us}, apk^{rs}), (ask^{us}, ask^{rs}))$.
- $\boldsymbol{Setup^U}$: On input PM, it runs OS.$\boldsymbol{KeyGen}(1^k) \to (pk_i^{os}, sk_i^{os})$, CUS.$\boldsymbol{KeyGen}(1^k) \to (pk_i^{us}, sk_i^{us})$, and RS.$\boldsymbol{KeyGen}(1^k) \to (pk_i^{rs}, sk_i^{rs})$ to compute a user public and private key pair $(UPK_i, USK_i) = ((pk_i^{os}, pk_i^{us}, pk_i^{rs}), (sk_i^{os}, sk_i^{us}, sk_i^{rs}))$.
- $\boldsymbol{PSign}$: On input a message and a signer private key (m, USK_i), it runs OS.$Sign(m, sk_i^{os}) \to \sigma^{os}$ and outputs a partial signature $\sigma_p = \sigma^{os}$.

- ***PVer***: On input (m, σ_p, UPK_i), it can validate σ_p by running OS.$\boldsymbol{Ver}(m, \sigma^{os}, pk_i^{os})$. It outputs "1" if σ_p is valid and outputs "0" otherwise.
- ***Sign***: On input $(m, \sigma_p, USK_i, APK, UPK_i)$. Let $m' = H(m, \sigma_p, UPK_i)$. It runs CUS.$\boldsymbol{Sign}(m', sk_i^{us}) \rightarrow \sigma^{us}$ and RS.$\boldsymbol{Sign}(H(\sigma^{us}), sk_i^{rs}, PK_L) \rightarrow \sigma^{rs}$, where $PK_L = (pk_i^{rs}, apk^{rs})$ and outputs a full signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$.
- ***Ver***: On input (m, σ, UPK_i, APK), it can verify $\sigma = (\sigma_p = \sigma^{os}, \sigma^{us}, \sigma^{rs})$ by running OS.$\boldsymbol{Verify}(m, \sigma^{os}, pk_i^{os})$ and RS.$\boldsymbol{Verify}(H(\sigma^{us}), \sigma^{rs}, PK_L)$, where $PK_L = (pk_i^{rs}, apk^{rs})$. Therefore, if σ_p and σ^{rs} are valid, this algorithm outputs "1" and "0" otherwise.
- ***Res***: On input $(m, \sigma_p, ASK, APK, UPK_i)$, it first checks the validity of σ_p by running OS.$\boldsymbol{Verify}(m, \sigma^{os}, pk_i^{os})$. It outputs " $\perp$ " if σ_p is invalid. Otherwise, it continues to compute $m' = H(m, \sigma_p, UPK_i)$. It then runs CUS.$\boldsymbol{Sign}(m', ask^{us}) \rightarrow \sigma^{us}$ and RS.$\boldsymbol{Sign}(H(\sigma^{us}), ask_i^{rs}, PK_L) \rightarrow \sigma^{rs}$, where $PK_L = (pk_i^{rs}, apk^{rs})$ and outputs a full signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$
- ***Prove***$^{\boldsymbol{A}}$: On input $(m, \sigma, ASK, APK, UPK_i)$, it first runs $\boldsymbol{Ver}(m, \sigma, UPK_i, APK)$ to check its validity and continue if and only if it is valid. Then it computes $m' = H(m, \sigma_p, UPK_i)$ and runs CUS.$\boldsymbol{SConvert}(m', \sigma^{us}, ask^{us}) \rightarrow \pi^A$ and outputs a proof $\pi = \pi^A$. Otherwise, it outputs " $\perp$ ".
- ***Prove***$^{\boldsymbol{U}}$: On input $(m, \sigma, USK_i, APK, UPK_i)$, it first runs $\boldsymbol{Ver}(m, \sigma, UPK_i, APK)$ to check its validity and continue if and only if it is valid. Then it computes $m' = H(m, \sigma_p, UPK_i)$ and runs CUS.$\boldsymbol{SConvert}(m', \sigma^{us}, sk_i^{us}) \rightarrow \pi^U$ and outputs a proof $\pi = \pi^U$. Otherwise, it outputs " $\perp$ ".
- ***Open***: On input $(m, \sigma, UPK_i, APK, \pi)$, it first runs $\boldsymbol{Ver}(m, \sigma, UPK_i, APK)$ to check its validity and continue if and only if it is valid. Otherwise, it outputs " $\perp$ ". It computes $m' = H(m, \sigma_p, UPK_i)$ and parses π in the following cases:
  - If $\pi = \pi^A$, it runs CUS.$\boldsymbol{Verify}(m', \sigma^{us}, \pi^A, apk^{us}) \rightarrow b \in \{0, 1\}$. If $b = 1$, it outputs "APK" which indicates σ^{us} is originally generated by the arbitrator using ask^{us}. Otherwise, it outputs "UPK_i". If the output is " $\perp$ ", it means π is invalid.
  - Else if $\pi = \pi^U$, it runs CUS.$\boldsymbol{Verify}(m', \sigma^{us}, \pi^U, pk_i^{us}) \rightarrow b \in \{0, 1\}$. If $b = 1$, it outputs "UPK_i" which indicates σ^{us} is originally generated by the signer using sk_i^{us}. Otherwise, it outputs "APK". If the output is " $\perp$ ", it means π is invalid.

Correctness. The correctness of our generic framework follows the correctness of the underlying ordinary signature, convertible undeniable signature, and ring signature scheme.

4.2. Security Analysis

In this subsection, we provide the security analysis on our proposed framework. The proof approach for resolution ambiguity and accountability are inspired by Huang et al. [19], and the proof approach for security against signers, security against verifiers, and security against arbitrator are inspired by Ganjavi et al. [24].

4.2.1. Resolution Ambiguity

Lemma 1. *Our proposed generic framework is resolution ambiguity if the underlying convertible undeniable signature scheme and ring signature scheme satisfy anonymity.*

Proof. As the full signature contains a partial signature, a convertible undeniable signature, and a ring signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$, the resolution ambiguity follows the anonymity of the underlying convertible undeniable signature scheme and ring signature scheme. □

4.2.2. Type I Accountability

Lemma 2. *Our proposed generic framework is* $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$*-type I accountable if the underlying convertible undeniable signature scheme is* $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$*-EUF-CMA and complete and sound.*

Proof. Let $\mathcal{A}$ be the PPT adversary which $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-breaks the type I accountability, we build a PPT algorithm $\mathcal{D}$ which runs $\mathcal{A}$ as a subroutine and $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-breaks the EUF-CMA and the completeness and soundness of the underlying convertible undeniable signature scheme with the success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, q_{Res} queries to $\mathcal{O}_{Res}$, q_{Prove^A} queries to $\mathcal{O}_{Prove^A}$, and q_{Prove^U} queries to $\mathcal{O}_{Prove^U}$ in time t.

- **Phase 1**: On input (pk_0^{us}, pk_0^{rs}) to $\mathcal{D}$, $\mathcal{D}$ sets $APK = (pk_0^{us}, pk_0^{rs})$ and passes to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ can make queries to its accessible oracles defined in Section 2.2. At the end, $\mathcal{A}$ runs $\boldsymbol{Setup^U}(PM) \rightarrow (UPK_i, USK_i)$ to generate users' private and public key pairs. $\mathcal{A}$ then passes a challenge public key $\hat{UPK} = (pk^{os}, pk_1^{us}, pk_1^{rs})$ to $\mathcal{D}$.
- **Phase 3**: $\mathcal{A}$ can make queries to the following oracles:
  - Resolution Oracle $\mathcal{O}_{Res}$: On input $(m, \sigma_p, \hat{UPK})$, $\mathcal{D}$ requests σ^{us} from convertible undeniable signature scheme's $\mathcal{O}_S$ on input (m', pk_0^{us}), where $m' = H(m, \sigma_p, \hat{UPK})$. $\mathcal{D}$ then requests σ^{rs} from ring signature scheme's $\mathcal{O}_S$ on input $(H(\sigma^{us}), PK_L, e)$, where $PK_L = (pk_0^{rs}, pk_1^{rs})$ and $e = 1$ is the selected public key position in PK_L. Note that $(\sigma^{us}, \sigma^{rs})$ is generated with (sk_0^{us}, sk_0^{rs}) respectively. Finally, $\mathcal{D}$ returns a signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$ to $\mathcal{A}$.
  - Arbitrator Prove Oracle $\mathcal{O}_{Prove^A}$: On input $(m, \sigma = (\sigma_p, \sigma^{us}, \sigma^{rs}))$, $\mathcal{D}$ requests a selective token π^S on pk_0^{us} from convertible undeniable signature scheme's $\mathcal{O}_{SC}$ on input (m', σ^{us}), where $m' = H(m, \sigma_p, \hat{UPK})$. $\mathcal{D}$ then returns an arbitrator proof $\pi^A = \pi^S$.
- **Phase 4**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ that is valid on $(\hat{UPK}, APK)$ and a proof $\hat{\pi}$ with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Res}$. Note that $\hat{\pi}$ be can either π^A by pk_0^{us} or π^U by pk_1^{us}.

Assume that $\mathcal{A}$ wins the game because $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, \hat{UPK}, APK, \pi) =$ "APK" and $\hat{\sigma}$ is not generated from ***Res***, there exist two possible cases:

- **Case 1**: $\hat{\sigma}^{us}$ is generated by using sk_0^{us}, so CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_0^{us}, \hat{\pi}^A) =$ "1" and CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_1^{us}, \hat{\pi}^U) =$ "0" hold.
- **Case 2**: $\hat{\sigma}^{us}$ is generated by using sk_1^{us}, but $\hat{\pi}$ is not sound. Hence, CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_0^{us}, \hat{\pi}^A) =$ "1" and CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_1^{us}, \hat{\pi}^U) =$ "0".

Hence, if $\mathcal{D}$ takes $(\hat{m}, \hat{\sigma}^{us}, \hat{\pi})$ as the output, $\mathcal{D}$ breaks the EUF-CMA of convertible undeniable signature scheme in **Case 1** and breaks the completeness and soundness of the underlying convertible undeniable signature scheme in **Case 2**. This shows that there exists a PPT $\mathcal{D}$ which can either $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-break the EUF-CMA or the completeness and soundness of the underlying convertible undeniable signature scheme if there exists $\mathcal{A}$ which can $(t, q_{PSign}, q_{Sign}, q_{Res}, q_{Prove^A}, q_{Prove^U}, \varepsilon)$-break the type I accountability. This contradicts the EUF-CMA and the completeness and soundness of the underlying convertible undeniable signature scheme, hence our OFE protocol is type I accountable. □

4.2.3. Type II Accountability

Lemma 3. *Our generic framework is $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-type II accountable if the underlying convertible undeniable signature scheme is $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-EUF-CMA and complete and sound.*

Proof. Let $\mathcal{A}$ be the PPT adversary which $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-breaks the type II accountability, we build a PPT algorithm $\mathcal{D}$ which runs $\mathcal{A}$ as a subroutine and $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-breaks the EUF-CMA and the completeness and soundness of the underlying convertible undeniable signature scheme with the success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$, q_{Sign} queries to $\mathcal{O}_{Sign}$, and q_{Prove^U} queries to $\mathcal{O}_{Prove^U}$ in time t.

- **Phase 1**: On input PM to $\mathcal{D}$, $\mathcal{D}$ then passes PM to $\mathcal{A}$ which then runs $\boldsymbol{Setup^A}(PM) \rightarrow (APK = (pk_0^{us}, pk_0^{rs}), ASK = (sk_0^{us}, sk_0^{rs}))$ and passes APK to $\mathcal{D}$.
- **Phase 2**: $\mathcal{A}$ can make queries to the following oracles for a selected $UPK_i = (pk_i^{os}, pk_i^{us}, pk_i^{rs})$:
  - Partial Sign Oracle $\mathcal{O}_{PSign}$: On input (m, UPK_i), $\mathcal{D}$ requests a signature σ^{os} from ordinary signature scheme's $\mathcal{O}_S$ on input (m, pk_i^{os}). $\mathcal{D}$ then returns a partial signature $\sigma_p = \sigma^{os}$.
  - Full Sign Oracle $\mathcal{O}_{Sign}$: On input (m, σ_p, UPK_i), $\mathcal{D}$ requests σ^{us} from convertible undeniable signature scheme's $\mathcal{O}_S$ on input (m', pk_i^{us}), where $m' = H(m, \sigma_p, UPK_i)$. $\mathcal{D}$ then requests σ^{rs} from ring signature scheme's $\mathcal{O}_S$ on input $(H(\sigma^{us}), PK_L, e)$, where $PK_L = (pk_0^{rs}, pk_i^{rs})$ and $e = 2$ is the selected public key position in PK_L. Note that $(\sigma^{us}, \sigma^{rs})$ is generated with (sk_i^{us}, sk_i^{rs}) respectively.
  - User Prove Oracle $\mathcal{O}_{Prove^U}$: On input $(m, \sigma = (\sigma_p, \sigma^{us}, \sigma^{rs}))$, $\mathcal{D}$ requests a selective token π^S on pk_i^{us} from convertible undeniable signature scheme's $\mathcal{O}_{SC}$ on input (m', σ^{us}), where $m' = H(m, \sigma_p, UPK_i)$. $\mathcal{D}$ then returns a user proof $\pi^U = \pi^S$.
- **Phase 3**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ that is valid on (UPK_i, APK) and a proof $\hat{\pi}$, with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Sign}$. Note that $\hat{\pi}$ can either be the π^A by pk_0^{us} or the π^U by pk_1^{us}.

Assume that $\mathcal{A}$ wins the game because $\boldsymbol{Open}(\hat{m}, \hat{\sigma}, UPK_i, APK, \pi) =$ "UPK_i" and $\hat{\sigma}$ is not generated from $\boldsymbol{Sign}$, there exist two possible cases:

- **Case 1**: $\hat{\sigma}^{us}$ is generated by using sk_i^{us}, so CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_0^{us}, \hat{\pi}^A) =$ "0" and CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_i^{us}, \hat{\pi}^U) =$ "1" hold.
- **Case 2**: $\hat{\sigma}^{us}$ is generated by using sk_0^{us}, but $\hat{\pi}$ is not sound. Therefore, CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_0^{us}, \hat{\pi}^A) =$ "0" and CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, pk_i^{us}, \hat{\pi}^U) =$ "1".

Hence, if $\mathcal{D}$ takes $(\hat{m}, \hat{\sigma}^{us}, \hat{\pi})$ as the output, $\mathcal{D}$ breaks the EUF-CMA of convertible undeniable signature scheme in **Case 1** and breaks the completeness and soundness of convertible undeniable signature scheme in **Case 2**. This shows that there exists a PPT $\mathcal{D}$ that can either $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-break the EUF-CMA or the completeness and soundness of the underlying convertible undeniable signature scheme if there exists $\mathcal{A}$ which can $(t, q_{PSign}, q_{Sign}, q_{Prove^U}, \varepsilon)$-break the type II accountability. This contradicts the EUF-CMA and the completeness and soundness of the underlying convertible undeniable signature scheme, hence our OFE protocol is type II accountable. □

4.2.4. Type III Accountability

Lemma 4. *Our proposed generic framework is (t, ε)-type III accountable if the underlying convertible undeniable signature scheme is (t, ε)-complete and sound.*

Proof. Let $\mathcal{A}$ be the PPT adversary which (t, ε)-breaks the type III accountability, we build a PPT algorithm $\mathcal{D}$ which runs $\mathcal{A}$ as a subroutine and (t, ε)-breaks the completeness and soundness of the underlying convertible undeniable signature scheme with the success probability more than ε in its game in time t.

- **Phase 1**: On input PM to $\mathcal{D}$, $\mathcal{D}$ then passes PM to $\mathcal{A}$ which then runs $\boldsymbol{Setup^A}(PM) \rightarrow (APK, ASK)$ and $\boldsymbol{Setup^U}(PM) \rightarrow (UPK_i, USK_i)$.
- **Phase 2**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$ that is valid on (UPK_i, APK) and two proofs $(\hat{\pi}^U, \hat{\pi}^A)$.

At the end of the game, $\mathcal{D}$ takes $(\hat{m}, \hat{\sigma}^{us}, \hat{\pi}^U, \hat{\pi}^A)$ as the output. $\mathcal{D}$ breaks the completeness and soundness of the underlying convertible undeniable signature scheme if either one of the following statements holds:

1. A valid $(\hat{m}, \hat{\sigma}^{us})$ on UPK_i but CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, APK, \hat{\pi}^{A}) =$ "1" $\wedge$ CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, UPK_i, \hat{\pi}^{U}) =$ "0"
2. A valid $(\hat{m}, \hat{\sigma}^{us})$ on APK but CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, APK, \hat{\pi}^{A}) =$ "0" $\wedge$ CUS.$\boldsymbol{SVerify}(\hat{m}, \hat{\sigma}^{us}, UPK_i, \hat{\pi}^{U}) =$ "1"

This shows that there exists a PPT $\mathcal{D}$ which can (t, ε)-break the completeness and soundness of the underlying convertible undeniable signature if there exists $\mathcal{A}$ which can (t, ε)-break the type III accountability with the success probability more than ε in its game in time t. This contradicts the completeness and soundness of the convertible undeniable signature scheme, hence our OFE protocol is type III accountable. □

4.2.5. Security against Signers

Lemma 5. *Our proposed generic framework is unconditionally secure against signers.*

Proof. The security against signers in our generic framework follows unconditionally as a full signature contains a partial signature, a convertible undeniable signature, and a ring signature, $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$, the arbitrator can always convert σ_p to σ, by generating σ^{us} on $m' = H(m, \sigma_p, UPK_i)$ and σ^{rs}. □

4.2.6. Security against Verifiers

Lemma 6. *Our proposed generic framework is $(t, q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-secure against verifiers if the underlying convertible undeniable signature scheme is $(t + t_1 q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-EUF-CMA and ring signature scheme is $(t + t_1 q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-EUF-CSA.*

Proof. Let $\mathcal{A}$ be the PPT adversary which $(t, q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-break the security against verifiers, we build a PPT algorithm $\mathcal{D}$ which runs $\mathcal{A}$ as a subroutine and $(t + t_1 q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-breaks the EUF-CMA of the underlying convertible undeniable signature scheme and the EUF-CSA of the underlying ring signature scheme with the success probability more than ε in its game with at most q_{Sign} queries to $\mathcal{O}_{Sign}$ and q_{Res} queries to $\mathcal{O}_{Res}$ in time $t + t_1 q_{PSign}$, where $t_1 q_{PSign}$ is the time cost to generate a partial signature.

- **Phase 1**: On input two challenge public key pairs $((pk_0^{us}, pk_0^{rs}), (pk_1^{us}, pk_1^{rs}))$ to $\mathcal{D}$, $\mathcal{D}$ first runs OS.$\boldsymbol{KeyGen} \rightarrow (pk^{os}, sk^{os})$. $\mathcal{D}$ then chooses $b \in \{0, 1\}$ and sets $APK = (pk_b^{us}, pk_b^{rs})$ and $UPK = (pk^{os}, pk_{1-b}^{us}, pk_{1-b}^{rs})$. $\mathcal{A}$ is given (APK, UPK).
- **Phase 2**: $\mathcal{A}$ can make queries to the following oracles:
  - Partial Sign Oracle $\mathcal{O}_{PSign}$: On input (m, UPK), $\mathcal{D}$ returns OS.$\boldsymbol{Sign}(m, sk^{os}) \rightarrow \sigma_p$ to $\mathcal{A}$.
  - Full Sign Oracle $\mathcal{O}_{Sign}$: On input (m, σ_p, UPK), $\mathcal{D}$ requests σ^{us} from convertible undeniable signature scheme's $\mathcal{O}_S$ on input (m', pk_{1-b}^{us}), where $m' = H(m, \sigma_p, UPK)$. $\mathcal{D}$ then requests σ^{rs} from ring signature scheme's $\mathcal{O}_S$ on input $(H(\sigma^{us}), PK_L, e)$, where $PK_L = (pk_{1-b}^{rs}, pk_b^{rs})$ and $e = 1$ is the selected public key position in PK_L. Note that $(\sigma^{us}, \sigma^{rs})$ is generated with $(sk_{1-b}^{us}, sk_{1-b}^{rs})$ respectively. Finally, $\mathcal{D}$ returns a signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$ to $\mathcal{A}$.
  - Resolution Oracle $\mathcal{O}_{Res}$: This oracle is similar to $\mathcal{O}_{Sign}$ above, but $(\sigma^{us}, \sigma^{rs})$ is generated with (sk_b^{us}, sk_b^{rs}) respectively, where σ^{us} is from convertible undeniable signature scheme's $\mathcal{O}_S$ on input (m', pk_b^{us}) and σ^{rs} is from ring signature scheme's $\mathcal{O}_S$ on input $(H(\sigma^{us}), PK_L, e)$ where $PK_L = (pk_{1-b}^{rs}, pk_b^{rs})$ and $e = 2$ is the selected public key position in PK_L.
- **Phase 3**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$, where $\hat{\sigma} = (\hat{\sigma}_p, \hat{\sigma}^{us}, \hat{\sigma}^{rs})$ with the restriction that $\hat{\sigma}$ is not generated from $\mathcal{O}_{Sign}$ or $\mathcal{O}_{Res}$.

At the end of the game, $\mathcal{D}$ takes $(\hat{m}', \hat{\sigma}^{us}, \hat{\sigma}^{rs})$ as the output where $\hat{m}' = H(\hat{m}, \hat{\sigma}_p, UPK)$. $\mathcal{D}$ breaks the EUF-CMA of the underlying convertible undeniable signature scheme if $(\hat{m}', \hat{\sigma}^{us})$ is valid on either

pk_0^{us} or pk_1^{us}. $\mathcal{D}$ also breaks the EUF-CSA of the underlying ring signature scheme if $\hat{\sigma}^{rs}$ is valid on either pk_0^{rs} or pk_1^{rs}. Therefore, this shows that there exists a PPT $\mathcal{D}$ which can $(t + t_1 q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-break the EUF-CMA of the underlying convertible undeniable signature scheme and EUF-CSA of the underlying ring signature scheme if there exists $\mathcal{A}$ which can $(t, q_{PSign}, q_{Sign}, q_{Res}, \varepsilon)$-break the security against verifiers with the success probability more than ε in its game with at most q_{Sign} queries to $\mathcal{O}_{Sign}$ and q_{Res} queries to $\mathcal{O}_{Res}$ in time $t + t_1 q_{PSign}$. This contradicts the EUF-CMA of the underlying convertible undeniable signature scheme and the EUF-CSA of the underlying ring signature scheme, hence our OFE protocol is secure against verifiers. □

4.2.7. Security against Arbitrator

Lemma 7. *Our proposed generic framework is $(t, q_{PSign}, \varepsilon)$-secure against the arbitrator if the underlying ordinary signature scheme is $(t, q_{PSign}, \varepsilon)$-EUF-CMA.*

Proof. Let $\mathcal{A}$ be the PPT adversary which $(t, q_{PSign}, \varepsilon)$-breaks the security against the arbitrator, we build a PPT algorithm $\mathcal{D}$ which runs $\mathcal{A}$ as a subroutine and $(t, q_{PSign}, \varepsilon)$-breaks the EUF-CMA of the underlying ordinary signature scheme with the success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$ in time t.

- **Phase 1**: On input a challenge public key pk^{os} to $\mathcal{D}$. $\mathcal{D}$ first generates public parameters PM and passes to $\mathcal{A}$.
- **Phase 2**: $\mathcal{A}$ then runs $\boldsymbol{Setup}^{\mathcal{A}}(PM) \rightarrow (APK, ASK)$ and sends APK to $\mathcal{D}$.
- **Phase 3**: $\mathcal{A}$ can make queries to Partial Sign Oracle $\mathcal{O}_{PSign}$: On input (m, UPK), where $\mathcal{D}$ requests a signature σ^{os} from ordinary signature scheme's $\mathcal{O}_S$ on input (m, pk^{os}). $\mathcal{D}$ returns a partial signature $\sigma_p = \sigma^{os}$.
- **Phase 4**: $\mathcal{A}$ outputs a challenge message and signature pair $(\hat{m}, \hat{\sigma})$, where $\hat{\sigma} = (\hat{\sigma}_p, \hat{\sigma}^{us}, \hat{\sigma}^{rs})$ with the restriction that $(\hat{m}, UPK)$ has not been queried to $\mathcal{O}_{PSign}$.

At the end of the game, $\mathcal{D}$ takes $(\hat{m}, \hat{\sigma}_p)$ as the output. $\mathcal{D}$ breaks the EUF-CMA of the underlying ordinary signature scheme if OS.$\boldsymbol{Verify}(\hat{m}, \hat{\sigma}_p, pk^{os}) = 1$. This shows that there exists a PPT $\mathcal{D}$ which can $(t, q_{PSign}, \varepsilon)$-break the EUF-CMA of the underlying ordinary signature scheme if there exists $\mathcal{A}$ which can $(t, q_{PSign}, \varepsilon)$-break the security against arbitrator with the success probability more than ε in its game with at most q_{PSign} queries to $\mathcal{O}_{PSign}$ in time t. This contradicts the EUF-CMA of the underlying ordinary signature scheme, hence our OFE protocol is secure against arbitrator. □

Theorem 1. *Our proposed generic framework is secure in the multi-user setting and chosen-key model.*

Proof. The proof follows directly from Lemmas 1–7. □

5. An Instantiation of Accountable Optimisitc Fair Exchange Protocol

The derived protocol is built from Boneh et al.'s short signature scheme [30], Li et al.'s convertible undeniable signature scheme [31], and Shim's ring signature scheme [32], following our proposed generic framework. We first review the respective underlying schemes.

5.1. Boneh et al.'s Short Signature Scheme

The same public parameters $PM = (q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, H)$ is used following the definition as in Section 3.1, and $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ is a hash function. The scheme consists of the following algorithms [30]:

- ***KeyGen***: It randomly picks $x \in \mathbb{Z}_q$ and computes $X = g_2^x$. It then returns a public and private key pair $(pk, sk) = (X, x)$.
- ***Sign***: On input a message and a private key (m, sk), it returns an ordinary signature $\sigma^{os} = H(m)^x$.
- ***Verify***: On input (m, σ^{os}, pk), it checks whether $\hat{e}(H(m), X) \stackrel{?}{=} \hat{e}(\sigma^{os}, g_2)$. It outputs "1" if σ is valid and "0" otherwise.

5.2. Li et al.'s Convertible Undeniable Signature Scheme

The same public parameters PM as in Boneh et al.'s short signature scheme is used in this scheme with the following algorithms [31]:

- ***KeyGen***: It randomly picks $x, y \in \mathbb{Z}_q^*$ to compute $X = g_2^x$ and $Y = g_2^y$. It outputs a public and private key pair $(pk, sk) = ((X, Y), (x, y))$.
- ***Sign***: On input a message and private key (m, sk), it computes an undeniable signature $\sigma^{us} = H(m)^{xy}$.
- ***Confirmation/Disavowal Protocol***: Given a message and signature pair (m, σ^{us}), it can confirm or deny σ^{us} with the following designated verifier non-interactive zero knowledge proof of knowledge $(DVPK)$:

$$DVPK(y : \hat{e}(\sigma^{us}, g_2) = \hat{e}(H(m), X)^y \wedge Y = g_2^y) \text{ or } DVPK(y : \hat{e}(\sigma^{us}, g_2) \neq \hat{e}(H(m), X)^y \wedge Y = g_2^y)$$

- ***SConvert***: On input (m, σ^{us}, sk), it computes a converter $\pi^S = H(m)^y \in \mathbb{G}_1$.
- ***SVerify***: On input $(m, \sigma^{us}, pk, \pi^S)$, it first verifies π^S by checking whether $\hat{e}(\pi^S, g_2) \stackrel{?}{=} \hat{e}(H(m), Y)$ or not. If π^S is valid, then it proceeds to validate σ^{us} by checking whether $\hat{e}(\sigma^{us}, g_2) \stackrel{?}{=} \hat{e}(\pi^S, X)$ holds or not.

5.3. Shim's Ring Signature Scheme

The same public parameters PM as in Boneh et al.'s short signature are used in this scheme with the following algorithms [32]:

- ***KeyGen***: For a user i, it randomly picks $x_i \in \mathbb{Z}_q^*$ to compute $X_i = g_2^{x_i}$. It outputs a public and private key pair $(pk_i, sk_i) = (X_i, x_i)$.
- ***Sign***: Let $PK_L = \{pk_1, ..., pk_n\}$ be a list of users' public keys with n members. On input a signer's public and private key pair (pk_s, sk_s) and a message $m \in \{0,1\}^*$, it first randomly chooses $Z_i \in \mathbb{G}_1$ and computes $z_i = h(Z_i, m, PK_L)$ for $i = 1, ..., n$ and $i \neq s$. It then chooses a random salt $r \in \mathbb{Z}_q$ and computes (Z_s, z_s, V), where

$$Z_s = g_2^r \prod_{i \neq s}^{n} pk_i Z_i \qquad z_s = h(Z_s, m, PK_L) \qquad V = g_1^{r + z_s x_s}$$

  Finally, it outputs a ring signature $\sigma^{rs} = (Z_i, ...Z_n, V)$.
- ***Verify***: On input (m, σ^{rs}, PK_L), where $PK_L = \{pk_1, ..., pk_n\}$ is a list of users' public keys with n members. It first computes $z_i = h(Z_i, m, PK_L)$ for $i = 1, ..., n$. It then checks whether $\hat{e}(V, g_2) \stackrel{?}{=} \hat{e}(\prod_{i=1}^{n} pk_i^{z_i} Z_i, g_2)$ holds or not. If it holds, it outputs "1" and "0" otherwise.

5.4. The Derived Accountable Optimistic Fair Exchange Protocol

In order to reduce the key pair needed, we use the same approach as in the concrete protocol proposed by [19], where we utilise the public and private key pair from Li et al.'s convertible undeniable signature scheme to construct the ordinary signature and the ring signature. The signature in our protocol is short and it is more efficient in generating signature and proof as compared to Huang et al. and Ganjavi et al.'s concrete protocols. However, it takes longer to verify a signature due to the pairing-based setting, and this results in the derived protocol to be secure in the random oracle model only.

Let OS = (***KeyGen***, ***Sign***, ***Verify***) be Boneh et al.'s signature scheme, CUS = (***KeyGen***, ***Sign***, ***Confirmation/Disavowal Protocol***, ***SConvert***, ***SVerify***) be Li et al.'s convertible undeniable signature scheme, and RS = (***KeyGen***, ***Sign***, ***Verify***) be Shim's ring signature scheme. The derived accountable OFE protocol consists of the following algorithms:

- ***PMGen***: On input 1^k, it generates $(q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}_1$,$\mathbb{G}_2$,$\mathbb{G}_T$ are cyclic groups of prime order q, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are two generators, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map. Let $H_1, H_2 : \{0,1\}^* \to \mathbb{G}_1$, $H_3 : \{0,1\}^* \to \mathcal{M}$, and $h_1, h_2 : \{0,1\}^* \to \mathbb{Z}_p^*$, where $\mathcal{M}$ is the message space. Finally, it outputs $PM = (q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, H_1, H_2, H_3, h_1, h_2)$.
- ***Setup*^{A}**: On input PM, it runs CUS.***KeyGen***$(1^k) \to (pk, sk)$, where $pk = (g_2^{x_a}, g_2^{y_a}) = (X_a, Y_a)$ and $sk = (x_a, y_a) \in \mathbb{Z}_q^*$. Note that (x_a, X_a) will be used for ring signature later. Lastly, it returns an arbitrator public and private key pair $(APK, ASK) = ((X_a, Y_a), (x_a, y_a))$.
- ***Setup*^{U}**: On input PM, it runs CUS.***KeyGen***$(1^k) \to (pk, sk)$, where $pk = (g_2^{x_i}, g_2^{y_i}) = (X_i, Y_i)$ and $sk = (x_i, y_i) \in \mathbb{Z}_q^*$. Note that (x_i, X_i) will be used for ordinary signature and ring signature later. Lastly, it returns a user public and private key pair $(UPK_i, USK_i) = ((X_i, Y_i), (x_i, y_i))$.
- ***PSign***: On input (m, USK_i), it runs OS.***Sign*** to compute an ordinary signature, $\sigma^{os} = H_1(m)^{x_i}$. It outputs a partial signature $\sigma_p = \sigma^{os}$.
- ***PVer***: On input (m, σ_p, UPK_i), it runs OS.***Verify*** to check the validity by comparing $\hat{e}(H_1(m), X_i) \stackrel{?}{=} \hat{e}(\sigma_p, g_2)$. It returns "1" if the equation holds and "0" otherwise.
- ***Sign***: On input $(m, \sigma_p, USK_i, UPK_i, APK)$, it runs ***PSig***$(m, \sigma_p, UPK_i)$ and continues if and only if σ_p is valid. Let $m' = H_3(m, \sigma_p, UPK_i)$, it runs CUS.*Sign* to generate a convertible undeniable signature, $\sigma^{us} = H_2(m')^{x_i y_i}$. It then runs RS.***Sign*** to generate a ring signature, σ^{rs}. Let $PK_L = \{X_a, X_i\}$, it randomly chooses $Z_a \in \mathbb{G}_1$ and computes $z_a = h(Z_a, H_3(\sigma^{us}), PK_L)$. It then chooses a random salt $r \in \mathbb{Z}_q$ and computes (Z_i, z_i, V):

$$Z_i = g_2^r X_a Z_a \qquad z_i = h(Z_i, H_3(\sigma^{us}), PK_L) \qquad V = g_1^{r + z_i x_i}$$

  Finally, it outputs a full signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$ where $\sigma^{rs} = (Z_a, Z_i, V)$.
- ***Ver***: On input (m, σ, UPK_i, APK), it first runs ***PVer***(m, σ_p, UPK_i) and continues if and only if σ_p is valid. It then runs RS.***Verify*** to verify σ^{rs}. Let $PK_L = \{X_a, X_i\}$, it then computes $z_a = h(Z_a, H_3(\sigma^{us}), PK_L)$ and $z_i = h(Z_i, H_3(\sigma^{us}), PK_L)$. It then checks whether $\hat{e}(V, g_2) \stackrel{?}{=} \hat{e}(X_a^{z_a} Z_a \cdot X_i^{z_i} Z_i, g_2)$ holds or not. If it holds, it outputs "1" and "0" otherwise.
- ***Res***: On input $(m, \sigma_p, ASK, UPK_i, APK)$, it runs ***PVer***$(m, \sigma_p, UPK_i)$ and continues if and only if σ_p is valid. Let $m' = H_3(m, \sigma_p, UPK_i)$, it runs CUS.***Sign*** to compute a convertible undeniable signature, $\sigma^{us} = H_2(m')^{x_a y_a}$. It then runs RS.***Sign*** to generate a ring signature, σ^{rs}. Let $PK_L = \{X_a, X_i\}$, it first randomly chooses $Z_i \in \mathbb{G}_1$ and computes $z_i = h(Z_i, H_3(\sigma^{us}), PK_L)$. It then chooses a random salt $r \in \mathbb{Z}_q$ and computes (Z_a, z_a, V):

$$Z_a = g_2^r X_i Z_i \qquad z_a = h(Z_a, H_3(\sigma^{us}), PK_L) \qquad V = g_1^{r + z_a x_a}$$

  Finally, it outputs a full signature $\sigma = (\sigma_p, \sigma^{us}, \sigma^{rs})$ where $\sigma^{rs} = (Z_a, Z_i, V)$.
- ***Prove*^{A}**: On input $(m, \sigma, ASK, UPK_i, APK)$, it first runs ***Ver***(m, σ, UPK_i, APK) and continues if and only if σ is valid. Let $m' = H_3(m, \sigma_p, UPK_i)$, it runs CUS.***SConvert*** to compute a proof $\pi^A = H_2(m')^{y_a}$. Otherwise, it outputs " $\perp$ "
- ***Prove*^{U}**: On input $(m, \sigma, USK_i, UPK_i, APK)$, it first runs ***Ver***(m, σ, UPK_i, APK) and continues if and only if σ is valid. Let $m' = H_3(m, \sigma_p, UPK_i)$, it runs CUS.***SConvert*** to compute a proof $\pi^U = H_2(m')^{y_i}$. Otherwise, it outputs " $\perp$ "
- ***Open***: On input $(m, \sigma, \pi, UPK_i, APK)$, it runs ***Ver***$(m, \sigma, UPK_i, APK)$ and continues if and only if σ is valid. Let $m' = H_3(m, \sigma_p, UPK_i)$, it runs CUS.***SVerify*** to verify σ^{us}, where

  - If $\pi = \pi^A$, it first checks the validity of π^A by running $\hat{e}(\pi^A, g_2) \stackrel{?}{=} \hat{e}(H_2(m'), Y_a)$ and outputs " $\perp$ " if π^A is invalid. Otherwise, it proceeds to validate σ^{us} by running $\hat{e}(\sigma^{us}, g_2) \stackrel{?}{=} \hat{e}(\pi^A, X_a)$. If the equation holds, it means σ^{us} was signed by the arbitrator and outputs "APK", otherwise it outputs "UPK_i".

– If $\pi = \pi^U$, it first checks the validity of π^U by running $\hat{e}(\pi^U, g_2) \stackrel{?}{=} \hat{e}(H_2(m'), Y_i)$ and outputs " $\perp$ " if π^U is invalid. Otherwise, it proceeds to validate σ^{us} by running $\hat{e}(\sigma^{us}, g_2) \stackrel{?}{=} \hat{e}(\pi^U, X_i)$. If the equation holds, it means σ^{us} was signed by the signer and outputs "UPK_i", otherwise it outputs "APK".

Security Analysis

In this section, we show the derived protocol is secure in the multi-user setting and chosen-key model which follows from Theorem 1.

- **Resolution Ambiguity**: This property requires that the underlying convertible undeniable signature and ring signature scheme satisfy anonymous. The derived protocol is resolution ambiguous which follows Lemma 1, such that the underlying Li et al.'s convertible undeniable signature scheme [31] is proven invisible based on One-more Decisional Co-Tripartite-Diffie-Hellman (1m-DCTDH) in the random oracle model, where it is also well known that the invisibility and anonymity are equivalent as proven by Galbraith and Mao [39]. Besides, the underlying Shim's ring signature scheme is unconditionally anonymous as shown by the author [32].
- **Accountability**: This property requires that the underlying convertible undeniable signature scheme satisfies EUF-CMA and completeness and soundness. The derived protocol is accountable which follows Lemmas 2–4, such that the underlying Li et al.'s convertible undeniable signature scheme [31] achieves EUF-CMA based on Computational co-Diffie-Hellman (Co-CDH) in the random oracle model. The completeness and soundness of Li et al.'s scheme is unconditionally satisfied as shown by the author.
- **Security against Signers**: This property is unconditionally satisfied which follows Lemma 5 as the generic framework follows the same construction as in Huang et al. [19] and Ganjavi et al. [24], such that the arbitrator can always convert a partial signature into a full signature by generating a convertible undeniable signature and ring signature.
- **Security against Verifiers**: This property requires that the underlying convertible undeniable signature and ring signature scheme satisfy EUF-CMA and EUF-CSA respectively. The derived protocol is secure against verifiers which follows Lemma 6, such that the underlying Li et al.'s convertible undeniable signature scheme [31] and Shim's ring signature scheme [32] are both proven EUF-CMA and EUF-CSA respectively based on Co-CDH in the random oracle model.
- **Security against Arbitrator**: This property requires that the underlying ordinary signature scheme satisfies EUF-CMA. The derived protocol is secure against arbitrator which follows Lemma 7, such that the underlying Boneh et al.'s ordinary signature [30] acheives EUF-CMA based on Co-CDH in the random oracle model.

6. Conclusions

We proposed a generic framework to construct accountable OFE protocol in the multi-user setting and chosen-key model which is proven secure in the standard model by using the ordinary signature, convertible undeniable signature, and ring signature scheme as the underlying building blocks. We also provided a concrete instantiation of accountable OFE protocol based our proposed generic framework in the random oracle model.

Author Contributions: All the authors have contributed equally to this research work.

Funding: This work was supported by the Malaysia government's Fundamental Research Grant Schemes (FRGS/1/2015/ICT04/MMU/03/5 and FRGS/1/2018/ICT04/MMU/01/1).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bao, F.; Wang, G.; Zhou, J.; Zhu, H. Analysis and Improvement of Micali's Fair Contract Signing Protocol. In *Information Security and Privacy*; Wang, H., Pieprzyk, J., Varadharajan, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 176–187.
2. Ben-Or, M.; Goldreich, O.; Micali, S.; Rivest, R.L. A fair protocol for signing contracts. *IEEE Trans. Inf. Theory* **1990**, *36*, 40–46. [CrossRef]
3. Park, J.M.; Chong, E.K.P.; Siegel, H.J. Constructing Fair-exchange Protocols for E-commerce via Distributed Computation of RSA Signatures. In Proceedings of the PODC '03 Twenty-Second Annual Symposium on Principles of Distributed Computing, Boston, MA, USA, 13–16 July 2003; ACM: New York, NY, USA, 2003; pp. 172–181. [CrossRef]
4. Abadi, M.; Glew, N.; Horne, B.; Pinkas, B. Certified email with a light on-line trusted third party: Design and implementation. *Int. World Wide Web Conf.* **2002**, *2*, 387–395.
5. Ateniese, G.; Nita-Rotaru, C. Stateless-Recipient Certified E-Mail System Based on Verifiable Encryption. In *Topics in Cryptology—CT-RSA 2002*; Preneel, B., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 182–199.
6. Imamoto, K.; Sakurai, K. A Certified E-mail System with Receiver's Selective Usage of Delivery Authority. In *Progress in Cryptology—INDOCRYPT 2002*; Menezes, A., Sarkar, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 326–338.
7. AlOtaibi, A.; Aldabbas, H. A review of fair exchange protocols. *Int. J. Comput. Netw. Commun.* **2012**, *4*, 307. [CrossRef]
8. Bahreman, A.; Tygar, J. Certified electronic mail. In Proceedings of the 1994 Network and Distributed System Security Symposium (NDSS 1994), New York, NY, USA, February 1994; pp. 3–19.
9. Coffey, T.; Saidha, P.; Burrows, P. Analysing the Security of a Non-repudiation Communication Protocol with Mandatory Proof of Receipt. In Proceedings of the ISICT '03 1st International Symposium on Information and Communication Technologies, Dublin, Ireland, 24–26 September 2003; Trinity College Dublin: Dublin, Ireland, 2003; pp. 351–356.
10. Cox, B.; Tygar, J.D.; Sirbu, M. NetBill Security and Transaction Protocol. In Proceedings of the USENIX Workshop on Electronic Commerce, New York, NY, USA, 11–12 July 1995; Volume 1.
11. Deng, R.H.; Gong, L.; Lazar, A.A.; Wang, W. Practical protocols for certified electronic mail. *J. Netw. Syst. Manag.* **1996**, *4*, 279–297. [CrossRef]
12. Asokan, N.; Schunter, M.; Waidner, M. Optimistic Protocols for Fair Exchange. In Proceedings of the CCS '97 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, 1–4 April 1997; ACM: New York, NY, USA, 1997; pp. 7–17. [CrossRef]
13. Dodis, Y.; Reyzin, L. Breaking and Repairing Optimistic Fair Exchange from PODC 2003. In Proceedings of the DRM '03 3rd ACM Workshop on Digital Rights Management, Washington, DC, USA, 27 October 2003; ACM: New York, NY, USA, 2003; pp. 47–54. [CrossRef]
14. 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 *Topics in Cryptology—CT-RSA 2008*; Malkin, T., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 106–120.
15. Huang, Q.; Yang, G.; Wong, D.S.; Susilo, W. Ambiguous Optimistic Fair Exchange. In *Advances in Cryptology–ASIACRYPT 2008*; Pieprzyk, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 74–89.
16. Wang, Y.; Au, M.H.; Susilo, W. Perfect Ambiguous Optimistic Fair Exchange. In *Information and Communications Security*; Chim, T.W., Yuen, T.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 142–153.
17. Huang, Q.; Wong, D.S.; Susilo, W. P2OFE: Privacy-Preserving Optimistic Fair Exchange of Digital Signatures. In *Topics in Cryptology—CT-RSA 2014*; Benaloh, J., Ed.; Springer: Cham, Switzerland, 2014; pp. 367–384.
18. Guo, Q.; Cui, Y.; Zou, X.; Huang, Q. Generic Construction of Privacy-Preserving Optimistic Fair Exchange Protocols. *J. Internet Serv. Inf. Secur.* **2017**, *7*, 44–56.
19. 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 Trans. Inf. Forensics Secur.* **2011**, *6*, 498–512. [CrossRef]

20. Bellare, M.; Rogaway, P. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the CCS '93 1st ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; ACM: New York, NY, USA, 1993; pp. 62–73. [CrossRef]
21. Bellare, M.; Goldreich, O. On Defining Proofs of Knowledge. In *Advances in Cryptology—CRYPTO' 92*; Brickell, E.F., Ed.; Springer: Berlin/Heidelberg, Germany, 1993; pp. 390–420.
22. Fiat, A.; Shamir, A. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology—CRYPTO' 86*; Odlyzko, A.M., Ed.; Springer: Berlin/Heidelberg, Germany, 1987; pp. 186–194.
23. Cramer, R.; Damgård, I.; Schoenmakers, B. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology—CRYPTO '94*; Desmedt, Y.G., Ed.; Springer: Berlin/Heidelberg, Germany, 1994; pp. 174–187.
24. Ganjavi, R.; Asaar, M.R.; Salmasizadeh, M. A traceable optimistic fair exchange protocol. In Proceedings of the 2014 11th International ISC Conference on Information Security and Cryptology, Tehran, Iran, 3–4 September 2014; pp. 161–166. [CrossRef]
25. Fujisaki, E.; Suzuki, K. Traceable Ring Signature. In *Public Key Cryptography–PKC 2007*; Okamoto, T., Wang, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 181–200.
26. Fujisaki, E. Sub-linear Size Traceable Ring Signatures without Random Oracles. In *Topics in Cryptology—CT-RSA 2011*; Kiayias, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 393–415.
27. Gu, K.; Wu, N. Constant Size Traceable Ring Signature Scheme without Random Oracles. Cryptology ePrint Archive, Report 2018/288, 2018. Available online: https://eprint.iacr.org/2018/288 (accessed on 6 June 2018).
28. Hu, C.; Li, D. Forward-Secure Traceable Ring Signature. In Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007), Qingdao, China, 30 July–1 August 2007; Volume 3, pp. 200–204. [CrossRef]
29. Loh, J.C.; Heng, S.H.; Tan, S.Y. A Generic Framework for Accountable Optimistic Fair Exchange Protocol. In *Lecture Notes in Computer Science, Proceeding of the 14th International Conference on Information Security Practice and Experience, Tokyo, Japan, 25–27 September 2018*; Su, C., Kikuchi, H., Eds.; Springer: New York, NY, USA, 2018; Volume 11125, pp. 299–309.
30. Boneh, D.; Lynn, B.; Shacham, H. Short Signatures from the Weil Pairing. In *Advances in Cryptology—ASIACRYPT 2001*; Boyd, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 514–532.
31. Li, F.; Gao, W.; Wang, Y.; Wang, X. Short Convertible Undeniable Signature From Pairing. *J. Softw.* **2013**, *8*, 2983–2990. [CrossRef]
32. Shim, K.A. An efficient ring signature scheme from pairings. *Inf. Sci.* **2015**, *300*, 63–69. [CrossRef]
33. Dodis, Y.; Lee, P.J.; Yum, D.H. Optimistic Fair Exchange in a Multi-user Setting. In *Public Key Cryptography—PKC 2007*; Okamoto, T., Wang, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 118–133.
34. Zhu, H.; Susilo, W.; Mu, Y. Multi-party Stand-Alone and Setup-Free Verifiably Committed Signatures. In *Public Key Cryptography—PKC 2007*; Okamoto, T., Wang, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 134–149.
35. Boneh, D.; Franklin, M. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology—CRYPTO 2001*; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 213–229.
36. Goldwasser, S.; Micali, S.; Rivest, R.L. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM J. Comput.* **1988**, *17*, 281–308. [CrossRef]
37. Chaum, D.; van Antwerpen, H. Undeniable Signatures. In *Advances in Cryptology—CRYPTO' 89 Proceedings*; Brassard, G., Ed.; Springer: New York, NY, USA, 1990; pp. 212–216.
38. Boyar, J.; Chaum, D.; Damgård, I.; Pedersen, T. Convertible Undeniable Signatures. In *Advances in Cryptology-CRYPT0' 90*; Menezes, A.J., Vanstone, S.A., Eds.; Springer: Berlin/Heidelberg, Germany, 1991; pp. 189–205.
39. Galbraith, S.D.; Mao, W. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In *Topics in Cryptology—CT-RSA 2003*; Joye, M., Ed.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 80–97.
40. Huang, X.; Mu, Y.; Susilo, W.; Wu, W. Provably Secure Pairing-Based Convertible Undeniable Signature with Short Signature Length. In *Pairing-Based Cryptography—Pairing 2007*; Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 367–391.

41. Rivest, R.L.; Shamir, A.; Tauman, Y. How to Leak a Secret. In *Advances in Cryptology—ASIACRYPT 2001*; Boyd, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 552–565.
42. Bender, A.; Katz, J.; Morselli, R. Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles. In *Theory of Cryptography*; Halevi, S., Rabin, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 60–79.
43. Bender, A.; Katz, J.; Morselli, R. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptol.* **2009**, *22*, 114–138. [CrossRef]