University of Wollongong

Research Online

Faculty of Engineering and Information
Sciences - Papers: Part A

Faculty of Engineering and Information
Sciences

1-1-2014

Collusion-resistance in optimistic fair exchange

Yang Wang

University of Wollongong, ywang@uow.edu.au

Willy Susilo

University of Wollongong, wsusilo@uow.edu.au

Man Ho Au

University of Wollongong, aau@uow.edu.au

Guilin Wang

University of Wollongong, guilin@uow.edu.au

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

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

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

Collusion-resistance in optimistic fair exchange

Abstract

Optimistic fair exchange (OFE) is a type of cryptographic protocols aimed at solving the fair exchange problem over open networks with the help of a third party to settle disputes between exchanging parties. It is well known that a third party is necessary in the realization of a fair exchange protocol. However, a fully trusted third party may not be available over open networks. In this paper, the security of most of the proposed OFE protocols depends on the assumption that the third party is semitrusted in the sense that it may misbehave on its own but does not conspire with either of the main parties. The existing security models of OFE have not taken into account the case where the potentially dishonest third party may collude with a signer in the sense of sharing its secret key with the signer. In this paper, to reduce the trust level of the arbitrator and increase the security of OFE, we propose an enhanced security model that, for the first time, captures this scenario. We also show a separation between the existing model and our enhanced model with a concrete counter example. Finally, we revisit two popular approaches in the construction of OFE protocols, which are based on verifiably encrypted signature and conventional signature plus ring signature, respectively. Our result shows that the conventional signature plus ring signature approach remains valid in our enhanced model. However, for schemes based on verifiably encrypted signature, slight modifications are needed to guarantee the security.

Keywords

exchange, fair, collusion, optimistic, resistance

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wang, Y., Susilo, W., Au, M. Ho. & Wang, G. (2014). Collusion-resistance in optimistic fair exchange. *IEEE Transactions on Information Forensics and Security*, 9 (8), 1227-1239.

Collusion-Resistance in Optimistic Fair Exchange

Yang Wang, Willy Susilo*, *Senior Member, IEEE*, Man Ho Au, *Member, IEEE*, and Guilin Wang

Abstract—Optimistic fair exchange (OFE) is a type of cryptographic protocols aimed at solving the fair exchange problem over open networks with the help of a third party to settle disputes between exchanging parties. It is well known that a third party is necessary in the realization of a fair exchange protocol. However, a fully trusted third party may not be available over open networks. In the literature, the security of most of the proposed OFE protocols depends on the assumption that the third party is semi-trusted in the sense that it may misbehave on its own but does not conspire with either of the main parties. The existing security models of OFE have not taken into account the case where the potentially dishonest third party may collude with a signer in the sense of sharing its secret key with the signer. In this work, to reduce the trust level of the arbitrator and increase the security of OFE, we propose an enhanced security model that, for the first time, captures this scenario. We also show a separation between the existing model and our enhanced model with a concrete counter example. Finally, we revisit two popular approaches in the construction of OFE protocols, which are based on verifiably encrypted signature and conventional signature plus ring signature, respectively. Our result shows that conventional signature plus ring signature approach remains valid in our enhanced model. However, for schemes based on verifiably encrypted signature, slight modifications are needed to guarantee the security.

Index Terms—Optimistic Fair Exchange, collusion, enhanced model.

I. INTRODUCTION

THE widely use of open networks such as Internet has resulted in a growing prosperity of electronic commerce. Under normal circumstances, the networks where the exchanges take place are insecure and the participants may not trust each other. Disputes may occur during the exchange even if both participants act honestly. Thus, the fair exchange issue, namely how two mutually distrustful parties exchange digital items over open computer networks in a fair way, has been considered as a fundamental problem in both electronic transactions and cryptography.

In 1997, Asokan, Schunter and Waidner [1] introduced the notion of optimistic fair exchange (OFE) to solve this problem. An optimistic fair exchange protocol comprises three kinds of participants: a signer, a verifier and a third party named an “arbitrator”. Typically such a protocol is conducted in three message flows. First, Alice, the signer, initiates the protocol by delivering a partial signature to Bob the verifier. A valid

partial signature not only serves as evidence to Bob that Alice has committed to endorse a certain message, but also assures Bob that he will receive Alice’s full signature at the end of the protocol. The assurance depends on the fact that the arbitrator is capable of converting the partial signature into a full one. In the second step, Bob delivers his full signature to Alice. Later, if Alice is honest, she will send her full signature to Bob in the third step, and this completes the exchange process. Note that under the normal situation, participation of the arbitrator is not required and thus, the term “optimistic”.

It is well-known that fairness can not be achieved without the involvement of such a third party [2]. However, this does not mean that the arbitrator in OFE has to be fully trusted. In fact, a fully trusted third party is typically undesirable as it is hard to find such an entity over open networks. To reduce the trust level of the third party and the difficulty of selecting such an entity, Franklin and Reiter [3] considered possible misbehavior by the third party and suggest to use a semi-trusted third party in the sense that it may misbehave on its own but does not conspire with either of the main parties. Franklin and Reiter believe the third party could even be a random member of the network when such a semi-trusted assumption is ensured.

Note that the arbitrator in OFE should correctly make a resolution. If the arbitrator converts the signer’s partial signature into a full one without a verifier having submitted his own full signature, the signer will be disadvantaged. On the other hand, if the arbitrator refuses to convert the signer’s partial signature into a full one even if a verifier has submitted his own full signature, the verifier will be disadvantaged. Without this basic assumption that the arbitrator correctly make a resolution, the fairness of OFE will not be guaranteed.

However, since the third party could be a random member of the network, it seems realistic that it may potentially help one party like its business partner. Thus the collusion of the arbitrator with the signer or the verifier should be taken into account, as long as the arbitrator correctly makes the required resolution request whenever needed. In this paper we consider the scenario that the arbitrator may implicitly collude with the signer in the sense of sharing its secret arbitrator key with the signer. If the fairness for the verifier can still be achieved when this collusion is considered, the trust for the arbitrator will be reduced for the verifier side. This is meaningful in practice, as it is easier for the signer to choose an arbitrator unilaterally than both mutually distrusted parties to choose a common trusted third party as an arbitrator.

A. Related Work

Since its introduction, optimistic fair exchange (OFE) [4], [5], [6], [7], [8] has been extensively studied. Two representative primitives for constructing OFE are *verifiably encrypted*

Y. Wang is with Cyberspace Security College, PLA Information Engineering University, Zhengzhou, Henan 450002, China. W. Susilo and M. H. Au are with School of Computer Science and Software Engineering, University of Wollongong, NSW 2522, Australia. G. Wang is with Huawei International Pte Ltd. E-mail: (yw990@uowmail.edu.au, wsusilo@uow.edu.au, auu@uow.edu.au, wang.guilin@huawei.com). This work was done when Y. Wang and G. Wang were with the University of Wollongong, Australia.

* Corresponding Author. W. Susilo was supported by ARC Future Fellowship FT0991397.

signatures [9], [10], [5], [6], [7] and *sequential two-party multisignatures* [4]. Some desirable properties in OFE, such as abuse-free [11], verifiability of the third party [12] (also known as accountability in [9]), resolution ambiguity [13], stand-alone [14], setup-free [14] and signer ambiguity [15], have also been proposed in the literature. Resolution ambiguity means that the full signatures generated by the signer should be computationally indistinguishable from those generated by the arbitrator, and it has been considered as a fundamental requirement for OFE schemes [4], [16], [17], [18], [15], [19]. As the intervention of an arbitrator could be caused by a network failure rather than by the cheating of a signer, an OFE scheme with resolution ambiguity can avoid bad publicity for the signer. Another motivation for resolution ambiguity is that the holder of a full signature generated by the arbitrator should not be treated differently by others from one with a full signature generated by the signer. In the following, we review the formal security definitions for OFE schemes.

Early security definitions of OFE only considered the single-user case, *i.e.*, there is only one signer, one verifier along with one arbitrator. The models proposed in [1], [9] considered the cases where the signer or verifier could be a dishonest cheater but the arbitrator itself was assumed to always act honestly. Franklin and Reiter [3] for the first time considered the case where the arbitrator could be dishonest. Later, Dodis and Reyzin [4] explicitly proposed a model to cover the scenario that the dishonest arbitrator tries to forge a signature on behalf of the signer. In particular, the model still assumes that the arbitrator *will not collude* with the signer or with the receiver and is semi-trusted.

In practice, due to the number of users, it would be logical to allow a number of users to share the same arbitrator. In 2007, Dodis, Lee and Yum [16] studied OFE in the multi-user setting, where there are many signers and verifiers along with one arbitrator in a system. In the multi-user setting, dishonest users are allowed to collude to cheat a target user. Dodis *et al.* [16] pointed out that the security of an OFE in the single-user setting does not necessarily guarantee that in the multi-user setting. Independently, the multi-user security of OFE was also studied by Zhu, Susilo and Mu [17].

The *certified-key* model (also known as the *registered-key* model [20]) is widely used in studying the security of OFE protocols. In this model, whenever a query with respect to a public key is made, the adversary must show its knowledge of the corresponding private key. In 2008, Huang *et al.* [18] studied OFE in the multi-user setting and *chosen-key model*, where the adversary can make queries with respect to arbitrary public keys, without requiring to know the corresponding private keys. They separated the security of OFE between the certified-key model and the chosen-key model through a concrete example. A new paradigm of constructing OFE protocols secure in their model was also proposed by employing ring signatures.

B. Motivation

The correctness of an OFE system requires that the arbitrator is capable of converting the partial signature into a full one.

To what extent this statement is true is often overlooked. In fact, existing model only guarantees that no signer can create a partial signature that cannot be converted into a full one by the arbitrator based on the public parameter of the system as well as the arbitrator's public key. It is not clear whether or not the statement still hold if the attacker is given the arbitrator's secret key as well.

Thus, a natural and practical question is "if a malicious signer has access to the secret key of the arbitrator, would the verifier be disadvantaged?" This explicitly captures the situation that the potentially dishonest arbitrator to some extent colludes with the signer by offering the signer access to the secret arbitration key. We provide an affirmative answer to this question by demonstrating that a malicious signer is able to produce a valid partial signature that passes the verification of the verifier, but that partial signature would not be convertible to a full signature. In other words, a signer who has access to the arbitrator's secret key could be able to obtain the verifier's item without committing to anything else. Thus, there is a need to enhance the existing model to capture this attack, and to develop schemes that can be proven secure in this enhanced model.

C. Our Contributions

In this paper, we make the following contributions.

- 1) We propose an enhanced security model that captures the previously overlooked case in OFE schemes, in which the potentially dishonest arbitrator may collude with a signer by offering its secret arbitration key. Our enhanced security model assures collusion-resistance in OFE. We also show that our enhanced model is strictly stronger than existing multi-user setting and chosen-key model with a concrete example that shows the separation.
- 2) After defining the enhanced security model, we investigate the security of the existing schemes in our enhanced model. We revisit two well-known methodologies for constructing OFE schemes, namely verifiably encrypted signatures, and the combination of conventional signatures and ring signatures. Our result shows that these paradigms will remain secure in our enhanced model.

II. SYNTAX AND EXISTING SECURITY MODEL

Throughout the paper, the following notations are used. Let $k \in \mathbb{N}$ be a security parameter. For a finite set $\mathcal{S}$, $s \leftarrow \mathcal{S}$ denotes that an element s is chosen uniformly at random from $\mathcal{S}$. By $y \leftarrow A^O(x)$, we mean the algorithm A , on input x and having access to oracle O , outputs y . By $x := y$, we mean variable x is assigned with the value of y .

A. Definition

Definition 1: A non-interactive OFE scheme involves signers, verifiers and the arbitrator, and consists of the following (probabilistic) polynomial-time algorithms:

- **Setup^{Arb}:** On input 1^k , the algorithm outputs the arbitrator's secret/public key pair (ASK, APK).

- $\text{Setup}^{\text{User}}$: On input 1^k and APK, it generates a user's secret/public key pair (SK, PK) . In this paper (SK_i, PK_i) is used to denote the user U_i 's key pair.
- Sig and Ver : These are the (full) signing and verification algorithms of OFE respectively. $\text{Sig}(m, SK_i, \text{APK})$, outputs a full signature σ under PK_i on message m , where m is chosen by user U_i from the message space $\mathcal{M}$, while $\text{Ver}(m, \sigma, PK_i, \text{APK})$ outputs $\top$ or $\perp$, indicating σ is U_i 's valid full signature on message m or not.
- PSig and PVer : These are partial signing and verification algorithms respectively. $\text{PSig}(m, SK_i, \text{APK})$ generates a partial signature σ_P on message m under the public key PK_i , while $\text{PVer}(m, \sigma_P, PK_i, \text{APK})$ outputs $\top$ or $\perp$, indicating σ_P is valid with respect to message m and the public key PK_i or not.
- Res : The arbitrator runs this resolution algorithm when making a resolution. $\text{Res}(m, \sigma_P, \text{ASK}, PK_i)$ outputs either a full signature σ on message m under the public key PK_i , or $\perp$ indicating the failure of resolving a partial signature.

Correctness property means the output is what we want if everybody follows the algorithms. More specifically, it states that

- $\text{Ver}(m, \text{Sig}(m, SK_i, \text{APK}), PK_i, \text{APK}) = \top$,
 $\text{PVer}(m, \text{PSig}(m, SK_i, \text{APK}), PK_i, \text{APK}) = \top$, and
 $\text{Ver}(m, \text{Res}(m, \text{PSig}(m, SK_i, \text{APK}), \text{ASK}, PK_i), PK_i, \text{APK}) = \top$.

Resolution ambiguity property states that any "resolved signature" $\text{Res}(m, \text{PSig}(m, SK_i, \text{APK}), \text{ASK}, PK_i)$ outputted by the arbitrator based on the signer's partial signature is computationally indistinguishable from the "actual signature" $\text{Sig}(m, SK_i, \text{APK})$ generated by the signer.

B. Security in Multi-User setting and Chosen-key Model

The security of an OFE scheme comprises three aspects: security against signers, security against verifiers, and security against the arbitrator. The security in the multi-user setting and chosen-key model [18] is captured by the following three experiments, in which the adversary can make queries to the resolution oracles with respect to adversarially chosen public keys and may not even know the corresponding secret keys.

SECURITY AGAINST SIGNERS. Intuitively this aspect of security requires that every valid partial signature generated by the signer can be resolved to a full signature by the arbitrator. Formally, we require that no PPT adversary $\mathcal{A}$ can succeed in the following experiment with non-negligible probability.

$$\begin{aligned} \text{Setup}^{\text{Arb}}(1^k) &\rightarrow (\text{ASK}, \text{APK}) \\ (m, \sigma_P, PK^*) &\leftarrow \mathcal{A}^{O_{\text{Res}}}(\text{APK}) \\ \sigma &\leftarrow \text{Res}(m, \sigma_P, \text{ASK}, PK^*) \\ \text{success of } \mathcal{A} &:= \left[\bigwedge \begin{array}{l} \text{PVer}(m, \sigma_P, PK^*, \text{APK}) = \top \\ \text{Ver}(m, \sigma, PK^*, \text{APK}) = \perp \end{array} \right] \end{aligned}$$

where oracle O_{Res} takes as input a tuple (m, σ_P, PK_i) such that $\text{PVer}(m, \sigma_P, PK_i, \text{APK}) = \top$ (i.e., a valid partial signature σ_P on message m under the public key PK_i), and outputs a full signature σ .

SECURITY AGAINST VERIFIERS. This aspect of security requires that the verifier himself is not able to generate the signer's full signature even if given a partial one. Formally, we require that no PPT adversary $\mathcal{A}$ can succeed in the following experiment with non-negligible probability.

$$\begin{aligned} \text{Setup}^{\text{Arb}}(1^k) &\rightarrow (\text{ASK}, \text{APK}) \\ \text{Setup}^{\text{User}}(1^k) &\rightarrow (SK, PK) \\ (m, \sigma) &\leftarrow \mathcal{A}^{O_{\text{PSig}}, O_{\text{Res}}}(\text{PK}, \text{APK}) \\ \text{success of } \mathcal{A} &:= \left[\bigwedge \begin{array}{l} \text{Ver}(m, \sigma, \text{PK}, \text{APK}) = \top \\ (m, \cdot) \notin \text{Query}(\mathcal{A}, O_{\text{Res}}) \end{array} \right] \end{aligned}$$

where oracle O_{Res} is the same as in the experiment of security against signers, $\text{Query}(\mathcal{A}, O_{\text{Res}})$ is the queries $\mathcal{A}$ made to oracle O_{Res} , and oracle O_{PSig} takes as input a message m and outputs a partial signature σ_P under the challenge public key PK . Note that the signing oracle which models the full signatures outputted by the algorithm Sig is not needed, as it can be functionally replaced by O_{PSig} and O_{Res} .

SECURITY AGAINST THE ARBITRATOR. This aspect of security requires that the arbitrator is not able to generate the signer's full signature unless holding a corresponding partial one. Formally we require that no PPT adversary $\mathcal{A}$ can succeed in the following experiment with non-negligible probability.

$$\begin{aligned} \text{Setup}^{\text{User}}(1^k) &\rightarrow (SK, PK) \\ (\text{ASK}^*, \text{APK}) &\leftarrow \mathcal{A}(\text{PK}) \\ (m, \sigma) &\leftarrow \mathcal{A}^{O_{\text{PSig}}}(\text{ASK}^*, \text{APK}, PK) \\ \text{success of } \mathcal{A} &:= \left[\bigwedge \begin{array}{l} \text{Ver}(m, \sigma, \text{PK}, \text{APK}) = \top \\ m \notin \text{Query}(\mathcal{A}, O_{\text{PSig}}) \end{array} \right] \end{aligned}$$

where ASK^* is $\mathcal{A}$'s state information, oracle O_{PSig} is the same as in the experiment of security against verifiers, and $\text{Query}(\mathcal{A}, O_{\text{PSig}})$ is the queries $\mathcal{A}$ made to oracle O_{PSig} .

III. THE ENHANCED MODEL

It is easy to see that the existing models do not capture the possible collusion amongst the arbitrator and signer. That is, when a dishonest signer has access to the arbitrator's secret key, security of the schemes proven in this model will be unclear. To capture this case and make the security model more practical, we propose the following enhanced model about security against signers.

SECURITY AGAINST SIGNERS. We require that no PPT adversary $\mathcal{A}$ can succeed in the following experiment with non-negligible probability.

$$\begin{aligned} \text{Setup}^{\text{Arb}}(1^k) &\rightarrow (\text{ASK}, \text{APK}) \\ (m, \sigma_P, PK^*) &\leftarrow \mathcal{A}(\text{ASK}, \text{APK}) \\ \sigma &\leftarrow \text{Res}(m, \sigma_P, \text{ASK}, PK^*) \\ \text{success of } \mathcal{A} &:= \left[\bigwedge \begin{array}{l} \text{PVer}(m, \sigma_P, PK^*, \text{APK}) = \top \\ \text{Ver}(m, \sigma, PK^*, \text{APK}) = \perp \end{array} \right] \end{aligned}$$

In other words, no signer, even with the knowledge of the arbitrator's secret key, should be able to produce a partial signature that looks good to a verifier but cannot be resolved to a full signature. The case that the arbitrator and a dishonest signer may implicitly collude is captured by allowing the adversary $\mathcal{A}$ having the arbitrator's secret key as an input.

A. Implications and Limitations of our Enhanced Model

Our model captures the case when the arbitrator's key is leaked to the attacker, or that someone having access to the arbitration key is colluding with the attacker. One limitation of our model is that it only captures the behavior of a dishonest arbitrator whose secret key is generated in complete accordance with the setup algorithm. This includes deleting all randomness used during key generation but not explicitly contained within the arbitration key. This is analogous to the situation for certificateless cryptosystem in which the attacker could be the key generation centre (KGC) itself. The nature of the KGC is modelled in two ways: one assumes the attacker has access to the KGC's master key which is generated honestly while the other assumes that the KGC's master key is created by the attacker. Readers are referred to [21] for the discussion of the various models of certificateless cryptosystem.

We choose to model an arbitrator whose secret key is correctly generated for two reasons. Firstly, we make the observation that existing models already fall short in capturing attacks from such an arbitrator. Secondly, we find that existing schemes following a certain design approach are immune against this kind of attack. We leave the formalization of a model that captures the behavior of an arbitrator who adversarially chooses its public key and the construction of schemes secure in this sense as an open problem.

In an orthogonal direction, we remark that a similar extension could not be applied to the security against verifiers. In other words, the arbitrator must be completely honest to the signer. Otherwise, the verifier could just convert the partial signature from the signer with the help of the arbitrator without the need to fulfill the obligation.

IV. SEPARATION OF THE PROPOSED MODEL AND THE EXISTING MODEL

In this section, we shall demonstrate a separation of our enhanced model and the existing model [18], which was reviewed in Section II-B. In order to do this, we will present a concrete OFE scheme, called A*-OFE¹, that is secure in the multi-user setting and chosen key model reviewed in Section II-B. Then, we will show a concrete attack against this scheme in the enhanced model. This shows that our model is strictly stronger than the existing model, as it is trivial to show that the security in the enhanced model implies that in the multi-user setting and chosen key model.

A. High Level Description

Prior to proposing the concrete construction that will serve as a counterexample, we will first provide the high level description. In this example, the full signature is a Schnorr signature [22] (c, s) on a message m under the signer's public key X such that $c = H(X || g || g^s X^c || m)$, where g is a generator of the group G in the Schnorr signature setting and H is a hash function. Suppose the order of g is of ℓ -bit and in

the following we use γ to denote the value $\ell - 1$. The partial signature comprises $c, S = g^s$, an encryption of the exponent s and a proof that the encryption is done correctly. That is, to generate a partial signature, the signer does the following.

- 1) The signer releases c and $S = g^s$.
- 2) Let b_j for $j = 0$ to γ be the binary representation of s . That is, $s = \sum_{j=0}^{\gamma} 2^j b_j$, where $b_j \in \{0, 1\}$.
- 3) Since no efficient encryption of exponents in the Schnorr signature setting is known, the signer encrypts the individual bits b_j of s .
 - The signer encodes bit 0 as the identity element in group G while bit 1 is encoded as the generator g . Thus to encrypt the j -th binary bit b_j , the signer equivalently encrypts the group element g^{b_j} .
 - For $j = 0$ to γ , the signer independently and randomly chooses r_j and computes

$$A_j = g^{r_j}, B_j = h^{r_j}, C_j = g^{b_j} Y^{r_j}.$$

Note that (A_j, B_j, C_j) is in fact the encryption of the bit b_j under the arbitrator's public key (h, Y) .

- The set of tuples (A_j, B_j, C_j) for $j = 0$ to γ constitute the encryption of the exponent s .
- 4) The signer then makes a proof of knowledge that the ciphertexts have been generated correctly. Naturally the proof of knowledge consists of two parts.
 - The first part ensures that the discrete logarithm of the value encrypted in (A_j, B_j, C_j) , when added together after applying the proper weights, is the exponent s . More specifically, the first part itself is a zero knowledge proof of knowledge of a set of values $\{s_j, r_j\}_{j=0}^{\gamma}$ such that $A_j = g^{r_j}, B_j = h^{r_j}, C_j = g^{s_j} Y^{r_j}$ and $S = g^{\sum_{j=0}^{\gamma} 2^j s_j}$.
 - The second part guarantees that the discrete logarithm of the value encrypted in (A_j, B_j, C_j) can only be 0 or 1. It can be viewed as a zero knowledge proof of knowledge of values R_j such that either $C_j = Y^{R_j}$ or $C_j = g Y^{R_j}$ holds.

To convert a partial signature into a full one, the arbitrator decrypts the ciphertexts and gains a sequence of plaintexts. If all the plaintexts are either the identity element in group G or the generator g , the arbitrator decodes the identity element and g as bit 0 and 1, respectively, and outputs the value whose binary representation is exactly the sequence of these bits. Otherwise the arbitrator returns $\perp$ to indicate failure in making a resolution.

Since the signer has no access to the arbitrator's secret key, the proof of knowledge of a set of values $\{s_j, r_j\}_{j=0}^{\gamma}$ such that $C_j = g^{s_j} Y^{r_j}, S = g^{\sum_{j=0}^{\gamma} 2^j s_j}$ and the proof of knowledge of values R_j such that $C_j = Y^{R_j}$ or $C_j = g Y^{R_j}$ together would imply that $s_j \in \{0, 1\}$ and $r_j = R_j$. Thus the above proof of the set of relationships achieve what the signer would like to convince the verifier. That is, the sum of s_j after applying the appropriate weights would be the component s of the Schnorr signature and that s_j can only be 0 or 1. Thus the arbitrator can decrypt the ciphertexts and output the exponent s such that $S = g^s$.

¹The name A*-OFE represents that the arbitrator may implicitly collude with the signer in our enhanced model.

B. Construction of A*-OFE

Formally we assume that we have a public group G with a generator g of prime order q of ℓ bits. Denote by γ the value $\ell - 1$. We also assume a cryptographic hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$.

The construction of A*-OFE is as follows.

- **Setup^{Arb}**: The arbitrator chooses random elements $h \in G$ and $y \in \mathbb{Z}_q$, and computes $Y = g^y$. The public key is set as $\text{APK} = (h, Y)$, and the arbitrator keeps $\text{ASK} = y$ as private.
- **Setup^{User}**: Each user U_i chooses a secret value $x_i \in \mathbb{Z}_q$, and calculates $X_i = g^{x_i}$. The user sets $(\text{SK}_i, \text{PK}_i) = (x_i, X_i)$.
- **PSig**: To partially sign a message m , a user U_i does as follows:

- 1) chooses a random $t \in \mathbb{Z}_q$ and then computes $c \in \mathbb{Z}_q$ and $s \in \mathbb{Z}_q$ such that

$$c = H(\text{PK}_i || g || g^t || m) \quad \text{and} \quad s = t - cx_i \quad (\text{in } \mathbb{Z}_q).$$

- 2) let $s = \sum_{j=0}^{\gamma} 2^j b_j$, where for each $0 \leq j \leq \gamma$, $b_j \in \{0, 1\}$. Note that this binary representation always exists and is unique in $\mathbb{Z}_q^2$. Let further $S = g^s$ and $E_j = g^{2^j}$ for $0 \leq j \leq \gamma$. Thus

$$S = \prod_{j=0}^{\gamma} (E_j)^{b_j}.$$

- 3) U_i chooses uniformly at random $0 \leq r_j \leq \mathbb{Z}_q$ for each $0 \leq j \leq \gamma$, and computes

$$A_j = g^{r_j}, \quad B_j = h^{r_j}, \quad C_j = g^{b_j} Y^{r_j}.$$

- 4) For $0 \leq j \leq \gamma$, if $b_j = 0$, U_i chooses uniformly at random elements $t_{j0}, c_{j1}, s_{j1} \in \mathbb{Z}_q$ and sets

$$T_{j0} = Y^{t_{j0}}, \quad T_{j1} = \left(\frac{C_j}{g}\right)^{c_{j1}} Y^{s_{j1}}.$$

Otherwise, U_i chooses uniformly at random $c_{j0}, s_{j0}, t_{j1} \in \mathbb{Z}_q$, and sets

$$T_{j0} = (C_j)^{c_{j0}} Y^{s_{j0}}, \quad T_{j1} = \left(\frac{C_j}{g}\right)^{t_{j1}}.$$

Besides, for $0 \leq j \leq \gamma$, U_i chooses uniformly at random $t_j, \tilde{t}_j \in \mathbb{Z}_q$, and sets

$$\tilde{S} = \prod_{j=0}^{\gamma} (E_j)^{\tilde{t}_j}, \quad \tilde{A}_j = g^{t_j}, \quad \tilde{B}_j = h^{t_j}, \quad \tilde{C}_j = g^{\tilde{t}_j} Y^{t_j}.$$

- 5) For $0 \leq j \leq \gamma$, denote $\tilde{T}_j = \tilde{A}_j || \tilde{B}_j || \tilde{C}_j$ and $\tilde{T}_j' = T_{j0} || T_{j1}$. Let $A = \tilde{T}_0 || \dots || \tilde{T}_\gamma$, and $B = \tilde{T}_0' || \dots || \tilde{T}_\gamma'$. The user U_i computes

$$\tilde{c} = H(\text{PK}_i || c || S || m || \tilde{S} || A || B).$$

²While the value s is unique in $\mathbb{Z}_q$, the set $\{b_j\}_{j=0}^{\ell-1}$ such that $g^s = g^{\sum_{j=0}^{\ell-1} 2^j b_j}$ may not be unique. For example, suppose $s = 1 \in \mathbb{Z}_q$ and that $q < 2^\ell - 1$, two sets of $\{b_j\}_{j=0}^{\ell-1}$ exist. They corresponds to the binary representation of 1 and $q + 1$. However, later we will discuss this will not affect the security of the scheme.

For $0 \leq j \leq \gamma$, if $b_j = 0$, U_i computes

$$c_{j0} = \tilde{c} - c_{j1} \quad (\text{in } \mathbb{Z}_q), \quad s_{j0} = t_{j0} - c_{j0} r_j \quad (\text{in } \mathbb{Z}_q).$$

Otherwise, U_i computes

$$c_{j1} = \tilde{c} - c_{j0} \quad (\text{in } \mathbb{Z}_q), \quad s_{j1} = t_{j1} - c_{j1} r_j \quad (\text{in } \mathbb{Z}_q).$$

Furthermore, for $0 \leq j \leq \gamma$, U_i computes

$$\tilde{s}_j = \tilde{t}_j - \tilde{c} b_j \quad (\text{in } \mathbb{Z}_q), \quad s_j = t_j - \tilde{c} r_j \quad (\text{in } \mathbb{Z}_q).$$

- 6) For $0 \leq j \leq \gamma$, denote $T_j = A_j || B_j || C_j$. The partial signature is set as $\sigma_P := (c, S, T_0, \dots, T_\gamma, c_{00}, c_{01}, s_{00}, s_{01}, \dots, c_{\gamma 0}, c_{\gamma 1}, s_{\gamma 0}, s_{\gamma 1}, \tilde{c}, \tilde{s}_0, s_0, \dots, \tilde{s}_\gamma, s_\gamma)$.
- **PVer**: Given a partial signature σ_P from user U_i , a verifier does as follows.

- 1) The verifier checks whether $c, \tilde{c} \in \mathbb{Z}_q$, and for $0 \leq j \leq \gamma$,

$$c_{j0}, c_{j1}, s_{j0}, s_{j1}, \tilde{s}_j, s_j \in \mathbb{Z}_q, \quad c_{j0} + c_{j1} = \tilde{c} \quad (\text{in } \mathbb{Z}_q).$$

- 2) The verifier checks whether

$$c = H(\text{PK}_i || g || S(X_i)^c || m).$$

- 3) The verifier computes $\tilde{S} = S^{\tilde{c}} \cdot \prod_{j=0}^{\gamma} (E_j)^{\tilde{s}_j}$ and for $0 \leq j \leq \gamma$,

$$T_{j0} = (C_j)^{c_{j0}} Y^{s_{j0}}, \quad T_{j1} = \left(\frac{C_j}{g}\right)^{c_{j1}} Y^{s_{j1}}.$$

- 4) Besides, for $0 \leq j \leq \gamma$, the verifier decomposes T_j as $A_j || B_j || C_j$ and computes

$$\tilde{A}_j = g^{s_j} (A_j)^{\tilde{c}}, \quad \tilde{B}_j = h^{s_j} (B_j)^{\tilde{c}}, \quad \tilde{C}_j = g^{\tilde{s}_j} Y^{s_j} (C_j)^{\tilde{c}}.$$

- 5) Denote $\tilde{T}_j = \tilde{A}_j || \tilde{B}_j || \tilde{C}_j$ and $\tilde{T}_j' = T_{j0} || T_{j1}$. Let $A = \tilde{T}_0 || \dots || \tilde{T}_\gamma$, and $B = \tilde{T}_0' || \dots || \tilde{T}_\gamma'$. The verifier checks whether

$$\tilde{c} = H(\text{PK}_i || c || S || m || \tilde{S} || A || B).$$

- 6) If all the above equations hold, then the verifier returns $\top$, otherwise $\perp$ is returned.

- **Sig**: To fully sign a message m , a user U_i chooses a random $t \in \mathbb{Z}_q$ and then computes $c \in \mathbb{Z}_q$ and $s \in \mathbb{Z}_q$ such that

$$c = H(\text{PK}_i || g || g^t || m) \quad \text{and} \quad s = t - cx_i \quad (\text{in } \mathbb{Z}_q).$$

The full signature is set as $\sigma := (c, s)$.

- **Ver**: Given a full signature $\sigma := (c, s)$ from user U_i , a verifier verifies whether $c \in \mathbb{Z}_q, s \in \mathbb{Z}_q$ and

$$c = H(\text{PK}_i || g || g^s (X_i)^c || m).$$

If so, $\top$ is returned and otherwise, $\perp$ is returned.

- **Res**: For the user U_i 's partial signature σ_P , the arbitrator
 - 1) first checks whether $\text{PVer}(m, \sigma_P, \text{PK}_i) = \top$. If so, continues; otherwise, returns $\perp$.
 - 2) for $0 \leq j \leq \gamma$, compute $D_j = C_j / (A_j)^y$. If $D_j = g$, sets $b_j = 1$, and if $D_j = 1$, sets $b_j = 0$. Otherwise, it responds to $\mathcal{A}$ with $\perp$.
 - 3) If $\perp$ is not returned, returns $s = \sum_{j=0}^{\gamma} 2^j b_j \mod q^3$.

³Note that even if the set of binary values $\{b_j\}$ is not unique, the value s computed by the arbitrator modulo q will be unique as long as $S = g^{\sum_{j=0}^{\ell-1} 2^j b_j}$ and that $b_j \in \{0, 1\}$.

C. Security Analysis.

It is not hard to verify that, in the above construction, any signature created by Sig will be valid under Ver, and that any signature created by the arbitrator using Res based on a partial signature generated by PSig will be valid under Ver. To make sure the correctness property of the above construction holds, we show that any partial signature created by PSig will be valid under PVer.

Given a partial signature $\sigma_P = (c, S, T_0, \dots, T_\gamma, c_{00}, c_{01}, s_{00}, s_{01}, \dots, c_{\gamma 0}, c_{\gamma 1}, s_{\gamma 0}, s_{\gamma 1}, \tilde{c}, \tilde{s}_0, s_0, \dots, \tilde{s}_\gamma, s_\gamma)$ generated by PSig, we verify the equations in PVer step by step for clarity as follows.

- 1) Note that, for $0 \leq j \leq \gamma$, if $b_j = 0$, $c_{j0} = \tilde{c} - c_{j1}$, $s_{j0} = t_{j0} - c_{j0}r_j$; otherwise, $c_{j1} = \tilde{c} - c_{j0}$, $s_{j1} = t_{j1} - c_{j1}r_j$. Thus the equation $c_{j0} + c_{j1} = \tilde{c}$ in step 1 of PVer holds.
- 2) Since $S = g^s$, $s = t - cx_i$, we have $c = H(\text{PK}_i || g || S(X_i^c || m)) = H(\text{PK}_i || g || g^{s+cx_i} || m) = H(\text{PK}_i || g || g^t || m)$. The equation in step 2 of PVer holds.
- 3) Note that $S = \prod_{j=0}^{\gamma} (E_j)^{b_j}$, $\tilde{s}_j = \tilde{t}_j - \tilde{c}b_j$, $\tilde{S} = \prod_{j=0}^{\gamma} (E_j)^{\tilde{t}_j}$, $\tilde{s}_j = \tilde{t}_j - \tilde{c}b_j$. Thus $\tilde{S} = \prod_{j=0}^{\gamma} (E_j)^{\tilde{s}_j + \tilde{c}b_j} = \prod_{j=0}^{\gamma} (E_j)^{\tilde{c}b_j} \cdot \prod_{j=0}^{\gamma} (E_j)^{\tilde{s}_j} = S^{\tilde{c}} \cdot \prod_{j=0}^{\gamma} (E_j)^{\tilde{s}_j}$. Besides, $C_j = g^{b_j Y^{r_j}}$ and when $b_j = 0$, $T_{j0} = Y^{s_{j0} + c_{j0}r_j} = (C_j)^{c_{j0}} Y^{s_{j0}}$ when $b_j = 0$. $T_{j1} = (\frac{C_j}{g})^{c_{j1}} Y^{s_{j1}}$ holds unconditionally when $b_j = 0$. Similarly, the equations $T_{j0} = (C_j)^{c_{j0}} Y^{s_{j0}}$, $T_{j1} = (\frac{C_j}{g})^{c_{j1}} Y^{s_{j1}}$ also hold when $b_j = 1$. That is, the equations in step 3 of PVer hold.
- 4) Note that $A_j = g^{r_j}$, $B_j = h^{r_j}$, $C_j = g^{b_j Y^{r_j}}$, $\tilde{A}_j = g^{t_j}$, $\tilde{B}_j = h^{t_j}$, $\tilde{C}_j = g^{t_j Y^{r_j}}$ and $\tilde{s}_j = \tilde{t}_j - \tilde{c}b_j$, $s_j = t_j - \tilde{c}r_j$. We have $\tilde{A}_j = g^{s_j + \tilde{c}r_j} = g^{s_j} (A_j)^{\tilde{c}}$, $\tilde{B}_j = h^{s_j + \tilde{c}r_j} = h^{s_j} (B_j)^{\tilde{c}}$, $\tilde{C}_j = g^{s_j + \tilde{c}b_j Y^{r_j} + \tilde{c}r_j} = g^{s_j} Y^{s_j} (C_j)^{\tilde{c}}$. The equations in step 4 of PVer hold.
- 5) The equation in step 5 of PVer holds unconditionally due to the fact that the same input to a hash function always leads to the same output.

Thus any partial signature created by PSig will be valid under PVer. The correctness property of the above construction holds.

Since the signature generated by Sig and that generated by the arbitrator using Res based on a valid partial signature are both Schnorr signatures, the resolution ambiguity property also holds.

Next we show the specific construction is secure in the multi-user setting and chosen-key model reviewed in Section II-B.

Theorem 1: The A*-OFE scheme is secure against signers in the multi-user setting and chosen-key model under the discrete logarithm assumption.

Proof. Suppose an adversary $\mathcal{A}$ breaks the security against signers. We show how to construct an algorithm $\mathcal{R}$ that solves the discrete logarithm problem. This will contradict with the discrete logarithm assumption.

Note that algorithm $\mathcal{R}$ is given random elements $u \in G$. Its goal is to output an integer $\alpha \in \mathbb{Z}_q$ such that $u = g^\alpha$. Algorithm $\mathcal{R}$ simulates the challenger and interacts with adversary $\mathcal{A}$ as follows.

Setup. Algorithm $\mathcal{R}$ chooses a random integer $y \in \mathbb{Z}_q$ and sets $h = u$, $Y = h^y$. Note that the distributions of h, Y are statistically close to the uniform distribution on G . $\mathcal{R}$ forwards the values h, Y to adversary $\mathcal{A}$, who returns its public key $\text{PK}^* = X_A$.

Hash Queries. At any time adversary $\mathcal{A}$ can query the random oracle H . To response to these queries, $\mathcal{R}$ maintains a list of tuples $\langle \text{string}^{(i)}, c_i \rangle$ as explained below. We refer to this list as H -list. The list is initially empty. When $\mathcal{A}$ queries the oracle H at a point $\text{string} \in \{0, 1\}^*$, algorithm $\mathcal{R}$ responds as follows:

- 1) If the query string already appears on the H -list in some tuple $\langle \text{string}, c \rangle$, then algorithm $\mathcal{R}$ responds with $H(\text{string}) = c \in \mathbb{Z}_q$.
- 2) Otherwise, $\mathcal{R}$ generates a random $c \in \mathbb{Z}_q$, adds the tuple $\langle \text{string}, c \rangle$ to the H -list and responds to $\mathcal{A}$ as $H(\text{string}) = c \in \mathbb{Z}_q$.

Note that c is uniform in $\mathbb{Z}_q$ and is independent of $\mathcal{A}$'s current view as required.

Res Queries. Given a resolution query $(m, \sigma_P, \text{PK}_i)$ where $\text{PK}_i = X_i$ is the signer's public key and $\sigma_P := (c, S, T_0, \dots, T_\gamma, \dots)$, algorithm $\mathcal{R}$ responds to this query as follows:

- 1) checks whether $\text{PVer}(\sigma_P, m, \text{PK}_i) = \top$. If so, continues; otherwise, $\mathcal{R}$ responds to $\mathcal{A}$ with a special symbol $\perp$.
- 2) for $0 \leq j \leq \gamma$, decompose T_j as $A_j || B_j || C_j$, and compute $D_j = C_j / (B_j)^y$. If $D_j = g$, sets $b_j = 1$, and if $D_j = 1$, sets $b_j = 0$. Otherwise, $\mathcal{R}$ responds to $\mathcal{A}$ with $\perp$.
- 3) If $\perp$ is not returned, $\mathcal{R}$ forwards $s = \sum_{j=0}^{\gamma} 2^j b_j$ to $\mathcal{A}$ as the response.

Output. It is easy to see that the above hash queries and resolution queries are perfectly simulated. Finally, $\mathcal{A}$ halts. It either admits failure, in which case so does $\mathcal{R}$, or it returns a partial signature σ_P^* on message m^* , where $\sigma_P^* := (c^*, S^*, T_0^*, \dots, T_\gamma^*, c_{00}^*, c_{01}^*, s_{00}^*, s_{01}^*, \dots, c_{\gamma 0}^*, c_{\gamma 1}^*, s_{\gamma 0}^*, s_{\gamma 1}^*, \tilde{c}^*, \tilde{s}_0^*, s_0^*, \dots, \tilde{s}_\gamma^*, s_\gamma^*)$ and $T_j^* = A_j^* || B_j^* || C_j^*$ for $0 \leq j \leq \gamma$, such that $\text{PVer}(\sigma_P^*, m^*, \text{PK}^*) = \top$, but it can not be resolved to a valid full signature by the resolution algorithm Res.

By the General Forking Lemma [23] (a standard rewinding technique in random oracle model, which states that if an adversary on inputs drawn from some distribution, produces an output that has some property, then with non-negligible probability, if the adversary is re-run on new inputs but with the same random tape, its second output will also have the property), with non-negligible probability algorithm $\mathcal{R}$ is able to gain another partial signature σ_P' (by running $\mathcal{A}$ again) on the same message m^* , where $\sigma_P' := (c^*, S^*, T_0^*, \dots, T_\gamma^*, c_{00}^*, c_{01}^*, s_{00}^*, s_{01}^*, \dots, c_{\gamma 0}^*, c_{\gamma 1}^*, s_{\gamma 0}^*, s_{\gamma 1}^*, \tilde{c}', \tilde{s}_0', s_0', \dots, \tilde{s}_\gamma', s_\gamma')$ and $\tilde{c}^* \neq \tilde{c}'$, such that $\text{PVer}(\sigma_P', m^*, \text{PK}^*) = \top$, but it can not be resolved to a valid full signature by the resolution algorithm Res.

Without loss of generality, we may assume that $\tilde{c}' > \tilde{c}^*$. For $0 \leq j \leq \gamma$, we have $(C_j^*)^{\tilde{c}^*} g^{\tilde{s}_j^*} Y^{s_j^*} = (C_j^*)^{\tilde{c}' } g^{\tilde{s}_j'} Y^{s_j'}$. It follows that $(C_j^*)^{\tilde{c}' - \tilde{c}^*} = g^{\tilde{s}_j^* - \tilde{s}_j'} Y^{s_j^* - s_j'}$. Let $\tau_j = (\tilde{s}_j^* -$

$\tilde{s}_j'(\tilde{c}' - \tilde{c}^*)^{-1}, v_i = (s_j^* - s_j')(\tilde{c}' - \tilde{c}^*)^{-1}$. Hence $C_j^* = g^{\tau_j} Y^{v_j}$.

For $0 \leq j \leq \gamma$, because $(A_j^*)^{\tilde{c}^*} g^{s_j^*} = (A_j^*)^{\tilde{c}'} g^{s_j'}$, $(B_j^*)^{\tilde{c}^*} h^{s_j^*} = (B_j^*)^{\tilde{c}'} h^{s_j'}$, it follows $A_j^* = g^{v_i}, B_j^* = h^{v_i}$.

Moreover, for $0 \leq j \leq \gamma$, because $(C_j^*)^{c_{j0}^*} Y^{s_{j0}^*} = (C_j^*)^{c_{j0}'} Y^{s_{j0}'}$, $(\frac{C_j^*}{g})^{c_{j1}^*} Y^{s_{j1}^*} = (\frac{C_j^*}{g})^{c_{j1}'} Y^{s_{j1}'}$, $c_{j0}^* + c_{j1}^* = \tilde{c}^*$ and $c_{j0}' + c_{j1}' = \tilde{c}'$, we have the following three cases:

case1: If $c_{j0}^* \neq c_{j0}'$ and $c_{j1}^* \neq c_{j1}'$, then we have $(C_j^*)^{c_{j0}^* - c_{j0}'} = Y^{s_{j0}^* - s_{j0}'}$ and $(\frac{C_j^*}{g})^{c_{j1}^* - c_{j1}'} = Y^{s_{j1}^* - s_{j1}'}$. Thus

we have $C_j^* = Y^{\frac{s_{j0}^* - s_{j0}'}{c_{j0}^* - c_{j0}'}} = gY^{\frac{s_{j1}^* - s_{j1}'}{c_{j1}^* - c_{j1}'}}$. Denote $\beta = y(\frac{s_{j1}^* - s_{j1}'}{c_{j1}^* - c_{j1}'} - \frac{s_{j0}^* - s_{j0}'}{c_{j0}^* - c_{j0}'})$. It follows $1 = gh^\beta$. Algorithm $\mathcal{R}$ can simply output $\alpha = -\beta^{-1}$ and breaks the discrete logarithm assumption when this case happens. Thus, we just safely omit this case.

case2: If $c_{j0}^* = c_{j0}'$ and $c_{j1}^* \neq c_{j1}'$. It follows that $(\frac{C_j^*}{g})^{c_{j1}^* - c_{j1}'} = Y^{s_{j1}^* - s_{j1}'}$. Denote $v_j' = (s_{j1}^* - s_{j1}')(\frac{c_{j1}^*}{c_{j1}'} - c_{j1}^*)^{-1}$. Then we have $C_j^* = gY^{v_j'}$. Since we already have $C_j^* = g^{\tau_j} Y^{v_j}$, it follows $1 \equiv g^{\tau_j - 1} Y^{v_j - v_j'}$. Thus it must be the case $\tau_j = 1$ and $v_i = v_i'$, as otherwise algorithm $\mathcal{R}$ can simply output $\alpha = (1 - \tau_j)(y v_j - y v_j')^{-1}$ and breaks the discrete logarithm problem. Therefore $C_j^* = gY^{v_j}$ when this case happens.

case3: If $c_{j0}^* \neq c_{j0}'$ and $c_{j1}^* = c_{j1}'$, then likewise under the discrete logarithm assumption we can gain that $\tau_j = 0$ and $C_j^* = Y^{v_j}$ when this case happens.

Furthermore, since $(S^*)^{\tilde{c}^*} \prod_{i=0}^{\gamma} (E_i)^{\tilde{s}_i^*} = (S^*)^{\tilde{c}'} \prod_{i=0}^{\gamma} (E_i)^{\tilde{s}_i'}$, we have $(S^*)^{\tilde{c}' - \tilde{c}^*} = \prod_{i=0}^{\gamma} (E_i)^{\tilde{s}_i^* - \tilde{s}_i'}$. It follows that $(S^*) = \prod_{i=0}^{\gamma} (E_i)^{\tau_i}$.

From the above analysis, it is readily to see that the arbitrator always returns a value $s^* = \sum_{j=0}^{\gamma} 2^j b_j \bmod q$ where $b_j = 0$ if $\tau_j = 0$, and $b_j = 1$ if $\tau_i = 1$. It is easy to verify $S^* = \prod_{j=0}^{\gamma} (E_j)^{\tau_j} = \prod_{j=0}^{\gamma} g^{2^j b_j} = g^{s^*}$, which means that under the discrete logarithm assumption, the scheme is secure against signers. $\square$

Theorem 2: The A*-OFE scheme is secure against the arbitrator in the multi-user setting and chosen-key model under the discrete logarithm assumption.

Proof. Suppose an adversary $\mathcal{A}$ breaks the security against the arbitrator. We show how to construct an algorithm $\mathcal{R}$ that solves the discrete logarithm problem. This will contradict with the discrete logarithm assumption. Algorithm $\mathcal{R}$ is given a random element $X \in G$ as input, and its goal is to output $x \in \mathbb{Z}_q$ such that $g^x = X$.

Setup: $\mathcal{R}$ sets $PK^* := X$ and sends it to the arbitrator.

Hash Queries. Hash queries are simulated in the same way as described in Theorem 1.

PSig Queries. At any time $\mathcal{A}$ can request a partial signature under challenge key $PK^* = X$. We respond to the i th partial signing query on message $m^{(i)}$ as follows:

- 1) We choose $c^{(i)}, s^{(i)} \in \mathbb{Z}_q$ at random, set $c^{(i)} := H(PK^* || g || g^{s^{(i)}} X^{c^{(i)}} || m^{(i)})$, and add $\langle PK^* || g || S^{(i)} X^{c^{(i)}} || m^{(i)}, c^{(i)} \rangle$ where $S^{(i)} = g^{s^{(i)}}$ to the H -list.
- 2) Let $s^{(i)} = \sum_{j=0}^{\gamma} 2^j b_j^{(i)}$, where for each $0 \leq j \leq \gamma$, $b_j^{(i)} \in \{0, 1\}$. We choose uniformly at random $0 \leq$

$r_j^{(i)} \leq \mathbb{Z}_q$ for each $0 \leq j \leq \gamma$, and compute

$$A_j^{(i)} = g^{r_j^{(i)}}, B_j^{(i)} = h^{r_j^{(i)}}, C_j^{(i)} = g^{b_j^{(i)}} (A_j^{(i)})^{y_1} (B_j^{(i)})^{y_2}.$$

- 3) We randomly choose $\tilde{c}^{(i)} \in \mathbb{Z}_q$. For $0 \leq j \leq \delta$, we choose uniformly at random $c_{j0}^{(i)}, s_{j0}^{(i)}, s_{j1}^{(i)} \in \mathbb{Z}_q$, and compute $c_{j1}^{(i)} = \tilde{c}^{(i)} - c_{j0}^{(i)}$ and

$$T_{j0}^{(i)} = (C_j)^{c_{j0}^{(i)}} Y^{s_{j0}^{(i)}}, T_{j1}^{(i)} = (\frac{C_j}{g})^{c_{j1}^{(i)}} Y^{s_{j1}^{(i)}}.$$

Besides, for $0 \leq j \leq \delta$, choose uniformly at random $0 \leq \tilde{s}_j^{(i)}, s_j^{(i)} \leq \mathbb{Z}_q$. Compute $\tilde{S}^{(i)} = S^{\tilde{c}^{(i)}} \cdot \prod_{j=0}^{\gamma} (E_j)^{\tilde{s}_j^{(i)}}$, and for $0 \leq j \leq \delta$, $\tilde{A}_j^{(i)} = g^{s_j^{(i)}} (A_j^{(i)})^{\tilde{c}^{(i)}}$, $\tilde{B}_j^{(i)} = g^{s_j^{(i)}} (B_j^{(i)})^{\tilde{c}^{(i)}}$, $\tilde{C}_j^{(i)} = g^{s_j^{(i)}} Y^{s_j^{(i)}} (C_j^{(i)})^{\tilde{c}^{(i)}}$.

- 4) For each $0 \leq j \leq \gamma$, denote $\tilde{T}_j^{(i)} = \tilde{A}_j^{(i)} || \tilde{B}_j^{(i)} || \tilde{C}_j^{(i)}$, $\tilde{T}_j'^{(i)} = T_{j0}^{(i)} || T_{j1}^{(i)}$ and $T_j^{(i)} = A_j^{(i)} || B_j^{(i)} || C_j^{(i)}$. Let $A^{(i)} = \tilde{T}_0^{(i)} || \dots || \tilde{T}_\gamma^{(i)}$, and $B^{(i)} = \tilde{T}_0'^{(i)} || \dots || \tilde{T}_\gamma'^{(i)}$. We set

$$\tilde{c}^{(i)} := H(PK^* || c^{(i)} || S^{(i)} || m^{(i)} || \tilde{S}^{(i)} || A^{(i)} || B^{(i)})$$

and add

$$\langle PK^* || c^{(i)} || S^{(i)} || m^{(i)} || \tilde{S}^{(i)} || A^{(i)} || B^{(i)}, \tilde{c}^{(i)} \rangle$$

to the H -list.

- 5) The partial signature is returned to $\mathcal{A}$ as $\sigma_P^{(i)} := (c^{(i)}, S^{(i)}, T_0^{(i)}, \dots, T_\gamma^{(i)}, c_{00}^{(i)}, c_{01}^{(i)}, s_{j0}^{(i)}, s_{j1}^{(i)}, \dots, c_{\gamma 0}^{(i)}, c_{\gamma 1}^{(i)}, s_{\gamma 0}^{(i)}, \tilde{c}^{(i)}, \tilde{s}_0^{(i)}, s_0^{(i)}, \dots, \tilde{s}_\gamma^{(i)}, s_\gamma^{(i)})$.

Output. It is obvious that the above hash queries and partial signing queries are perfectly simulated. Finally, $\mathcal{A}$ halts. It either admits failure, in which case so does $\mathcal{R}$, or it returns a full signature σ^* on message m^* , where $\sigma^* := (c^*, s^*)$ such that $\text{Ver}(\sigma^*, m^*, PK^*) = \top$.

Suppose adversary $\mathcal{A}$ makes Q_h hash queries and Q_p partial signing queries, and then produces a new full signature forgery. For $1 \leq i \leq Q_p$, let $m^{(i)}$ be the i th message submitted to the partial signing oracle and $\sigma_P^{(i)} := (c^{(i)}, S^{(i)} = g^{s^{(i)}}, \dots)$ be the reply for this query. Let $\sigma^* := (c^*, s^*)$ be the forgery on message m^* . If there does not exist an index $1 \leq i \leq Q_p$ such that $m^{(i)} = m^*, c^{(i)} = c^*, S^{(i)} = g^{s^*}$, then due to the randomness of the outputs of hash oracle, with overwhelming probability $\mathcal{A}$ has submitted the string $X || g || g^{s^*} X^{c^*} || m^*$ to the hash query. Therefore by the General Forking Lemma [23] (a rewinding technology in the random oracle model), with non-negligible probability adversary $\mathcal{A}$ is able to produce another full signature forgery σ' on the same message m^* , where $\sigma_P := (c', s')$ such that $c^* \neq c'$. $\mathcal{R}$ thus have $g^{s^*} X^{c^*} = g^{s'} X^{c'}$. This enable $\mathcal{R}$ to output $x = (s^* - s')(c' - c^*)^{-1}$ and break the discrete logarithm assumption. $\square$

Theorem 3: The A*-OFE scheme is secure against verifiers in the multi-user setting and chosen-key model under the decisional Diffie-Hellman assumption.

Proof. Suppose an adversary $\mathcal{A}$ makes Q_h hash queries, Q_p partial signing queries and Q_r resolution queries, and then wins by producing a new full signature forgery $\sigma^* := (c^*, s^*)$ on message m^* . For $1 \leq i \leq Q_p$, let $m^{(i)}$ be the i th message

submitted to the partial signing oracle and $\sigma_P^{(i)} := (c^{(i)}, S^{(i)} = g^{s^{(i)}}, \dots)$ be the reply for this query. To show the security, we first defines a sequence of games, and then show via a series of claims that if $\mathcal{A}$ is successful against Game x , then it will also be successful in Game $x + 1$.

Game 0. This is the real experiment between the challenger and an adversary $\mathcal{A}$ as defined in security against verifiers in Section II-B, in the random oracle model. This means that the challenger firstly correctly generates the arbitrator's and target signer's key pairs by running algorithms $\text{Setup}^{\text{Arb}}$ and $\text{Setup}^{\text{User}}$, respectively. The challenger forwards the arbitrator's public key $\text{APK} := (h, Y)$ and target signer's public key $\text{PK}^* := X$ to the adversary $\mathcal{A}$, and then provides accesses to the hash oracle, partial signing oracle and resolution oracle as well.

Game 1. This is the same as Game 0, with the exception that we make it a rule that $\mathcal{A}$ fails if there does not exist an index $1 \leq i \leq Q_p$ such that $m^{(i)} = m^*, c^{(i)} = c^*, S^{(i)} = g^{s^*}$.

Game 2. This is the same as Game 1, with the exception that we simulate the challenger as follows. We choose at random $y_1, y_2 \in \mathbb{Z}_q$ and set $Y = g^{y_1} h^{y_2}$.

Hash Queries. The simulation of hash queries is exactly the same as in Theorem 1.

PSig Queries. The simulation of partial signing queries is exactly the same as in Theorem 2.

Res Queries. $\mathcal{A}$ can request a resolution query $(m, \sigma_P, \text{PK}_i)$ where $\text{PK}_i = X_i$ is a signer's public key and $\mathcal{A}$ may not know its corresponding secret key, and $\sigma_P := (c, S, T_0, \dots, T_\gamma, c_{00}, c_{01}, s_{00}, s_{01}, \dots, c_{\gamma 0}, c_{\gamma 1}, s_{\gamma 0}, s_{\gamma 1}, \tilde{c}, \tilde{s}_0, s_0, \dots, \tilde{s}_\gamma, s_\gamma)$. We respond to this query as follows:

- check whether $\text{PVer}(m, \sigma_P, \text{PK}_i) = \top$. If so, continue; otherwise, return $\perp$.
 - 1) for $0 \leq j \leq \delta$, decompose T_j as $A_j || B_j || C_j$, and compute $D_j = C_j / (A_j)^{y_1} (B_j)^{y_2}$. If $D_j = g$, set $b_j = 1$, and if $D_j = 1$, set $b_j = 0$. Otherwise, respond to $\mathcal{A}$ with $\perp$.
 - 2) If $\perp$ is not returned, forward $s = \sum_{j=0}^{\gamma} 2^j b_j \bmod q$ to $\mathcal{A}$ as the response.
- When $\text{PVer}(\sigma_P, m, \text{PK}_i) = \perp$, respond to $\mathcal{A}$ with $\perp$.

Game 3. This is the same as Game 2, with the exception that at the beginning of the game we guess an index $1 \leq i^* \leq Q_p$, and $\mathcal{A}$ fails if $m^{(i^*)} \neq m^*$, or $c^{(i^*)} \neq c^*$, or $S^{(i^*)} \neq g^{s^*}$.

Game 4. This is the same as Game 3, with the exception that we make some changes to the i^* th partial signing oracle in the above game. Namely, for each $0 \leq j \leq \gamma$, instead of choosing uniformly at random $r_j^{(i^*)} \in \mathbb{Z}_q$, and computing

$$A_j^{(i^*)} = g^{r_j^{(i^*)}}, B_j^{(i^*)} = h^{r_j^{(i^*)}}, C_j^{(i^*)} = g^{b_j^{(i^*)}} (A_j^{(i^*)})^{y_1} (B_j^{(i^*)})^{y_2},$$

we choose independently and uniformly at random $\alpha_j, \beta_j, \delta_j \in \mathbb{Z}_q$, and compute

$$A_j^{(i^*)} = g^{\alpha_j}, B_j^{(i^*)} = g^{\beta_j}, C_j^{(i^*)} = g^{b_j^{(i^*)}} g^{\delta_j}.$$

Game 5. This is the same as Game 4, with the exception that for $0 \leq j \leq \gamma$, we randomly and independently choose $\mu_j \in \mathbb{Z}_q$ and replacing $C_j^{(i^*)} = g^{b_j^{(i^*)}} g^{\delta_j}$ with $C_j^{(i^*)} = g^{\mu_j}$.

Next we link the probability that $\mathcal{A}$ wins in these games via a series of claims. Let S_x be the event that $\mathcal{A}$ wins in Game x .

Claim 1

$$\Pr[S_1] = \Pr[S_0] - \epsilon_{\text{SAA}}, \quad (1)$$

where ϵ_{SAA} is the advantage of an adversary $\mathcal{A}'$ in the security against the arbitrator discussed above.

Suppose $\mathcal{A}$ generates a new full signature forgery $\sigma^* := (c^*, s^*)$ on message m^* , but there does not exist an index $1 \leq i \leq Q_p$ such that $m^{(i)} = m^*, c^{(i)} = c^*, S^{(i)} = g^{s^*}$. Then we build an adversary $\mathcal{B}$ in the security against the arbitrator. $\mathcal{B}$ correctly generates its key pair (ASK, APK) by running algorithm $\text{Setup}^{\text{Arb}}$ and then receives its own target signer's public key X . $\mathcal{B}$ forwards X and APK to $\mathcal{A}$. Whenever $\mathcal{A}$ requests a hash query or a partial signing query, $\mathcal{B}$ submits this query to its own hash oracle or partial signing oracle, respectively, and forwards the reply to $\mathcal{A}$. Whenever $\mathcal{A}$ requests a resolution query, $\mathcal{B}$ makes a resolution using its secret key ASK . Note that the simulation is perfect. Finally $\mathcal{B}$ outputs $\mathcal{A}$'s forgery $\sigma^* := (c^*, s^*)$ on message m^* and wins in the security against the arbitrator that was discussed in Theorem 2.

Claim 2

$$|\Pr[S_2] - \Pr[S_1]| \leq \text{negl}(k). \quad (2)$$

Define F to be the event that in a resolution query in Game 2, $\text{PVer}(\sigma_P, m, \text{PK}_i) = \top$ but there exists some $1 \leq j \leq \gamma$ such that $A_j = g^{r_j}, B_j = g^{r'_j}$ but $r \neq r' \bmod q$. If F does not occur, with $A_j = g^{r_j}, B_j = h^{r_j}$, then $Y^{r_j} = (A_j)^{y_1} B_j^{y_2}$; therefore the decryption oracle computes $D_j = C_j / Y^{r_j}$, just as it should be. That is to say, the Game 1 and Game 2 proceed statistically indistinguishably, unless event F occurs. Thus we have $|\Pr[S_0] - \Pr[S_1]| \leq \Pr[F]$, and only need to show that $\Pr[F] \leq \text{negl}(k)$. To prove this inequality, it suffices to notice that in a valid partial signature σ_P , a non-interactive proof of knowledge $\text{PK}\{(r_j) : A_j = g^{r_j} \wedge B_j = h^{r_j}\}$ was essentially conducted, and the knowledge error is negligible in k even for an adversary with infinite power.

Claim 3

$$\Pr[S_3] = \frac{\Pr[S_2]}{Q_p}. \quad (3)$$

The only difference between these Game 2 and Game 3 is that we guess a random index i^* . This value is used nowhere in the game, and independent to the adversary. Once the adversary makes a forgery, we only declare him successful if the index of his forgery matches our guess, which will occur with $1/Q_p$ probability.

Claim 4

$$|\Pr[S_4] - \Pr[S_3]| \leq l \cdot \epsilon_{\text{ddh}}, \quad (4)$$

where ϵ_{ddh} is the DDH-advantage of some efficient algorithm $\mathcal{R}$ (and hence negligible under the DDH assumption).

To show this, we divide the transition from Game 3 to Game 4 into a sequence of sub-games from Game $0'$ to Game $(\gamma + 1)'$. Game $0'$ is exactly the same as Game 3. Game $(j + 1)'$ is the same as Game j' , with the exception that in the i^* th partial signing query instead of choosing uniformly at random $r_j^{(i^*)} \in \mathbb{Z}_q$, and computing

$$A_j^{(i^*)} = g^{r_j^{(i^*)}}, B_j^{(i^*)} = h^{r_j^{(i^*)}}, C_j^{(i^*)} = g^{b_j^{(i^*)}} (A_j^{(i^*)})^{y_1} (B_j^{(i^*)})^{y_2}.$$

We choose uniformly at random $\alpha_j, \beta_j, \delta_j \in \mathbb{Z}_q$, and compute

$$A_j^{(i^*)} = g^{\alpha_j}, B_j^{(i^*)} = g^{\beta_j}, C_j^{(i^*)} = g^{b_j^{(i^*)}} g^{\delta_j}.$$

Note that Game $(\gamma + 1)'$ is exactly the same as Game 4. Let F_j be the event that $\mathcal{A}$ wins in Game j' .

Next we show that any difference between $\Pr[F_j]$ and $\Pr[F_{j+1}]$ can be parlayed into a corresponding DDH-advantage of an algorithm $\mathcal{R}$. Algorithm $\mathcal{R}$ runs as follows. It takes as input (g_2, u_1, u_2) , and interacts with $\mathcal{A}$. $\mathcal{R}$ sets $h = g_2$. It simulates the oracles as in Game j' except that in the i^* th partial signing query it computes $A_j^{(i^*)} = u_1, B_j^{(i^*)} = u_2, C_j^{(i^*)} = g^{b_j^{(i^*)}} (u_1)^{y_1} (u_2)^{y_2}$. If $\mathcal{A}$ outputs a full signature $\sigma^* := (c^*, s^*)$ on message m^* such that $m^* = m^{(i^*)}, c^* = c^{(j^*)}, g^{s^*} = S^{(j^*)}$, $\mathcal{R}$ output 1, else output a random bit $b \in \{0, 1\}$.

When the input to $\mathcal{R}$ is of the form (g_2, g^r, g_2^r) where r is randomly chosen, then computation proceeds just as in Game j' . When the input to $\mathcal{R}$ is of the form $(g_2, g^r, g_2^{r'})$, where $r, r' \in \mathbb{Z}_q$ are independently uniformly chosen, we only need to argue that $\varepsilon = (u_1)^{y_1} (u_2)^{y_2}$ is random to adversary $\mathcal{A}$'s view. If so, the simulation of $\mathcal{R}$ proceeds just the same as in Game $(j + 1)'$.

To see this, consider the point $\mathbf{Q} = (y_1, y_2) \in \mathbb{Z}_q^2$. Let $g_2 = g^w$. At the beginning of the attack, this is a random point on the line

$$\log_g Y = y_1 + w y_2, \quad (5)$$

determined by the arbitrator's public key. Note that the i^* th partial signing query should not be submitted to the resolution oracle, otherwise $\sigma^* := (c^*, s^*)$ will not be a new forgery on message $m^* = m^{(i^*)}$. Thus, due to the same reason as in Claim 2, with overwhelming probability the resolution oracle only decrypts with respect to (A_i, B_i, C_i) of the form $(g^{r_i}, h^{r_i}, g^{b_i} (A_i)^{y_1} (B_i)^{y_2})$, and the adversary $\mathcal{A}$ obtains only linear dependent relations $r_i \log_g Y = r_i y_1 + r_i w y_2$ (since $(A_i)^{y_1} (B_i)^{y_2} = g^{r_i y_1} h^{r_i y_2} = Y^{r_i}$). That means, no further information about $\mathbf{Q}$ is leaked.

Consider now the output $(A_j^{(i^*)}, B_j^{(i^*)}, C_j^{(i^*)})$ of $\mathcal{R}$'s partial signing reply for $\mathcal{A}$'s i^* th partial signing query. We have

$$\log_g \varepsilon = r y_1 + r' y_2. \quad (6)$$

Clearly, (5) and (6) are linearly independent when $r \neq r'$. ε is a perfect one time pad, and thus random to the adversary $\mathcal{A}$'s view.

Therefore, for each $0 \leq j \leq \gamma$, we have $|\Pr[F_j] - \Pr[F_{j+1}]| \leq \epsilon_{\text{ddh}}$, which is followed by $|\Pr[S_4] - \Pr[S_3]| =$

$|\Pr[F_{\gamma+1}] - \Pr[F_0]| \leq (\gamma + 1) \cdot \epsilon_{\text{ddh}}$. Note that $l = \gamma + 1$, the claim holds.

Claim 5

$$\Pr[S_5] = \Pr[S_4]. \quad (7)$$

Note that in game 5, g^{μ_j} is a randomly chosen element from G , the change from Game 5 to Game 4 was purely conceptual. Thus we have $\Pr[S_5] = \Pr[S_4]$.

Claim 6

$$\Pr[S_5] \leq \epsilon_{\text{dl}}, \quad (8)$$

where ϵ_{dl} is the discrete logarithm advantage, (and hence negligible under the discrete logarithm assumption).

To show this, we build a algorithm $\mathcal{R}$ that employs the adversary $\mathcal{A}$ in Game 5 to solve the discrete logarithm problem. Algorithm $\mathcal{R}$ runs as follows. It takes as input S . Its goal is to output $s \in \mathbb{Z}_q$ such that $g^s = S$. It interacts with $\mathcal{A}$, playing the role of the simulator in Game 5. $\mathcal{R}$ simulates the environments in the same way as in Game 5 with the exception that it sets $S^{(i^*)} = S$. The simulation is perfect. Finally when $\mathcal{A}$ wins in Game 5 by outputting (c^*, s^*) on message m^* , $\mathcal{R}$ outputs $s = s^*$ and solves the discrete logarithm problem.

Claim 7

$$\text{Adv}_{\mathcal{A}}(1^k) \leq \epsilon_{\text{SAA}} + Q_p(l \cdot \epsilon_{\text{ddh}} + \epsilon_{\text{dl}}) + \text{negl}(k), \quad (9)$$

where $\text{Adv}_{\mathcal{A}}(1^k)$ is advantage of $\mathcal{A}$ in the real experiment defined in the multi-user setting and chosen key model.

As a sequence of equations (1), (2), (3), (4), (7) and (8) gained above, we have equation (9). The whole proof is done. $\square$

D. A Concrete Attack against $\mathbf{A}^*$ -OFE in the Enhanced Model

We will show the above $\mathbf{A}^*$ -OFE is insecure in our enhanced model. When the arbitrator reveals its secret key $y \in \mathbb{Z}_q$ to a signer whose secret/public key pair is $(x, X = g^x)$, the malicious signer $\mathcal{A}$ can create a partial signature that passes the inspection of partial signature verification algorithm, but it can not be resolved to a valid full signature.

Taking further investigation into the above concrete construction, one may observe that there is a loophole about the proof made in the counterexample: the knowledge r_j used in the first part of the proof such that $C_j = g^{s_j} Y^{r_j}$ could be different with the knowledge R_j used in the second part of proof such that $C_j = Y^{R_j}$ or $C_j = g Y^{R_j}$. If that is true, the value s_j would not be 0 or 1. That is, the plaintext decrypted by the arbitrator in this case would be g^{s_j} rather than the identity element or the generator g of group G as it is supposed to be, and thus the arbitrator will fail to make a resolution.

Note that finding r_j, R_j such that $C_j = g^{s_j} Y^{r_j}$ and that $C_j = Y^{R_j}$ (or $C_j = g Y^{R_j}$) while $r_j \neq R_j$ is hard if the discrete logarithm of Y to the base g is unknown. However, in our enhanced model, this discrete logarithm value is known, and therefore finding such values r_j, R_j is feasible. This is the rationale of how our counterexample can be attacked when an

adversary is allowed to have access to the arbitrator's secret key.

The concrete attack by $\mathcal{A}$ is as follows.

- 1) Choose a random $t \in \mathbb{Z}_q$ and compute $c \in \mathbb{Z}_q$ and $s \in \mathbb{Z}_q$ such that

$$c = H(X||g||g^t||m) \text{ and } s = t - cx \text{ (in } \mathbb{Z}_q\text{)}.$$

Note that s and y are independent and both random, thus with probability $1/2$ we have $s \geq y$.

- 2) Let $s = \sum_{j=0}^{\gamma} 2^j b_j$ and $s - y = \sum_{j=0}^{\gamma} 2^j d_j$, where for each $0 \leq j \leq \gamma$, $b_j, d_j \in \{0, 1\}$. Note that the binary representation always exist and can be uniquely determined. Denote $S = g^s$, $\gamma = \ell - 1$ and $E_j = g^{2^j}$ for $0 \leq j \leq \gamma$. Let further $b'_j = d_j$ for $1 \leq j \leq \gamma$, and $b'_0 = d_0 + y$. Thus

$$S = \prod_{j=0}^{\gamma} (E_j)^{b'_j}.$$

- 3) $\mathcal{A}$ chooses uniformly at random $0 \leq r_j \leq \mathbb{Z}_q$ for each $0 \leq j \leq \gamma$, and computes

$$A_j = g^{r_j}, B_j = h^{r_j}, C_j = g^{b'_j} Y^{r_j}.$$

Note that a trick here that will be used by the malicious signers is that

$$C_0 = g^{b'_0} Y^{r_0} = g^{d_0} Y^{r_0+1}.$$

- 4) If $d_0 = 0$, $\mathcal{A}$ chooses uniformly at random elements $t_{00}, c_{01}, s_{01} \in \mathbb{Z}_q$ and sets

$$T_{00} = Y^{t_{00}}, T_{01} = \left(\frac{C_0}{g}\right)^{c_{01}} Y^{s_{01}}.$$

Otherwise, $\mathcal{A}$ chooses uniformly at random $c_{00}, s_{00}, t_{01} \in \mathbb{Z}_q$, and sets

$$T_{00} = (C_j)^{c_{00}} Y^{s_{00}}, T_{01} = \left(\frac{C_0}{g}\right)^{t_{01}}.$$

- 5) For $1 \leq j \leq \gamma$, if $b'_j = 0$, $\mathcal{A}$ chooses uniformly at random elements $t_{j0}, c_{j1}, s_{j1} \in \mathbb{Z}_q$ and sets

$$T_{j0} = Y^{t_{j0}}, T_{j1} = \left(\frac{C_j}{g}\right)^{c_{j1}} Y^{s_{j1}}.$$

Otherwise, $\mathcal{A}$ chooses uniformly at random $c_{j0}, s_{j0}, t_{j1} \in \mathbb{Z}_q$, and sets

$$T_{j0} = (C_j)^{c_{j0}} Y^{s_{j0}}, T_{j1} = \left(\frac{C_j}{g}\right)^{t_{j1}}.$$

Besides, for $0 \leq j \leq \gamma$, $\mathcal{A}$ chooses uniformly at random $t_j, \tilde{t}_j \in \mathbb{Z}_q$, and sets

$$\tilde{S} = \prod_{j=0}^{\gamma} (E_j)^{\tilde{t}_j}, \tilde{A}_j = g^{t_j}, \tilde{B}_j = h^{t_j}, \tilde{C}_j = g^{\tilde{t}_j} Y^{t_j}.$$

- 6) For $0 \leq j \leq \gamma$, denote $\tilde{T}_j = \tilde{A}_j || \tilde{B}_j || \tilde{C}_j$ and $\tilde{T}'_j = T_{j0} || T_{j1}$. Let $A = \tilde{T}_0 || \dots || \tilde{T}_\gamma$, and $B = \tilde{T}'_0 || \dots || \tilde{T}'_\gamma$. $\mathcal{A}$ computes

$$\tilde{c} = H(X || c || S || m || \tilde{S} || A || B).$$

If $d_j = 0$, the $\mathcal{A}$ computes

$$c_{00} = \tilde{c} - c_{01} \text{ (in } \mathbb{Z}_q\text{)}, s_{00} = t_{00} - c_{00}(r_0 + 1) \text{ (in } \mathbb{Z}_q\text{)}.$$

Otherwise, $\mathcal{A}$ computes

$$c_{01} = \tilde{c} - c_{00} \text{ (in } \mathbb{Z}_q\text{)}, s_{01} = t_{01} - c_{01}(r_0 + 1) \text{ (in } \mathbb{Z}_q\text{)}.$$

For $1 \leq j \leq \gamma$, if $b'_j = 0$, $\mathcal{A}$ computes

$$c_{j0} = \tilde{c} - c_{j1} \text{ (in } \mathbb{Z}_q\text{)}, s_{j0} = t_{j0} - c_{j0}r_j \text{ (in } \mathbb{Z}_q\text{)}.$$

Otherwise, $\mathcal{A}$ computes

$$c_{j1} = \tilde{c} - c_{j0} \text{ (in } \mathbb{Z}_q\text{)}, s_{j1} = t_{j1} - c_{j1}r_j \text{ (in } \mathbb{Z}_q\text{)}.$$

Furthermore, for $0 \leq j \leq \gamma$, $\mathcal{A}$ computes

$$\tilde{s}_j = \tilde{t}_j - \tilde{c}b'_j \text{ (in } \mathbb{Z}_q\text{)}, s_j = t_j - \tilde{c}r_j \text{ (in } \mathbb{Z}_q\text{)}.$$

- 7) For $0 \leq j \leq \gamma$, denote $T_j = A_j || B_j || C_j$. The partial signature is set as $\sigma_P := (c, S, T_0, \dots, T_\gamma, c_{00}, c_{01}, s_{00}, s_{01}, \dots, c_{\gamma 0}, c_{\gamma 1}, s_{\gamma 0}, s_{\gamma 1}, \tilde{c}, \tilde{s}_0, s_0, \dots, \tilde{s}_\gamma, s_\gamma)$.

It is easy to check that $\text{PVer}(\sigma_P, m, \text{PK}_i) = \top$. Now, we show that the resolution about this partial signature will return the symbol $\perp$, indicating failure. Indeed, the arbitrator will compute $D_j = C_j / (A_j)^y$ for $0 \leq j \leq \gamma$ in a resolution process. Since $D_0 = C_0 / (A_0)^y = g^{b'_0} = g^{d_0} Y$, rather than g or 1 as it should be. Thus the arbitrator will return $\perp$ by following the resolution algorithm. In this case, without any doubt, verifiers would be disadvantaged with probability $1/2$. The malicious signer can also repeat step 1 several times, for example n times, until a value s is found such that $s \geq y$. In this case, the signer can cheat with probability as least $1 - 1/2^n$.

V. PREVIOUS PARADIGMS REVISITED

In this section, we revisit the commonly used approaches in the construction of OFE in the enhanced security model.

A. Verifiably Encrypted Signature Paradigm

When Dodis *et al.* [16] proposed the security model of OFE in the multi-user setting and the certified-key model, they also showed that the popular paradigm for constructing OFE from verifiably encrypted signatures still leads to secure schemes in their model. Afterwards, Huang *et al.* [18] showed that verifiably encrypted signature based generic construction of OFE is also secure in the multi-user setting and chosen-key model. We will review the construction here and show that it is also secure in our enhanced model with a slight modification, which places an additional requirement to the underlying proof system of the verifiably encrypted signature scheme.

Given a public-key encryption scheme $\mathcal{E} = (\text{KGen}, \text{Enc}, \text{Dec})$ and a signature scheme $\mathcal{S} = (\text{KGen}, \text{Sig}, \text{Ver})$, we assume that an encryption key pair (ek, dk) and a signature key pair (sk, vk) have been generated, respectively. Let Π be a non-interactive zero-knowledge (NIZK) proof system for the NP-language $L = \{(c, m, ek, vk) | \exists \sigma [c = \mathcal{E}.\text{Enc}_{ek}(\sigma) \wedge \mathcal{S}.\text{Ver}_{vk}(m, \sigma) = 1]\}$. To partially sign a message, the signer with secret key $SK_i := sk$

produces a signature σ using $\mathcal{S}$ on a message m and encrypts the signature under the arbitrator's public key $\text{APK} := ek$ to gain a ciphertext c . Then, the signer runs Π to generate a NIZK proof π showing that c is indeed an encryption of a signature σ on message m . The partial signature is set as $\sigma_P := (c, \pi)$. The signer's full signature is set as σ . In a resolution, the arbitrator simply uses its secret key $\text{ASK} := dk$ to decrypts the ciphertext c and returns the signature σ .

The correctness property of the construction from this paradigm holds due to the correctness of the signature scheme $\mathcal{S}$ and the encryption scheme $\mathcal{E}$, and the completeness of the NIZK proof system Π , which guarantees that a proof made by an honest prover will always be accepted if the statement being proven is true.

Note that the partial signature contains a ciphertext which is the encryption of the signer's full signature. The resolution ambiguity property holds as a result of the fact that the resolution process is just a decryption operation in which the arbitrator decrypts the ciphertext and gains the signer's full signature.

Typically the security against signers in the multi-user setting and chosen-key model is due to the soundness of the NIZK proof system Π . The soundness property of ordinary proof systems states that the prover should not be able to convince the verifier of a false statement with non-negligible probability. Normally the adversary in the soundness model is only given the public parameters as input. Here, to make the OFE schemes constructed from verifiably encrypted signatures also secure in our enhanced model, we emphasize that the soundness property of Π must hold even when the adversary is also explicitly given the secret decryption key dk as input, like the model of verifiable encryption discussed in [24].

For security against verifiers in the multi-user setting and chosen key model, the simulation-sound property [25] is required. Intuitively, the simulation-sound property states that a polynomially bounded party is incapable of simulating a new proof of a false statement even after having seen a number of simulated proofs of its choice. This property guarantees that an adversary in the security against verifiers experiment is not able to submit a resolution query containing an accepting proof of a false statement even after seeing a simulated proof π generated by the simulator. Note that the soundness property of Π does not need to hold in a stronger sense that the adversary is also given the secret decryption key dk as input, as a malicious verifier in an OFE scheme never should have the arbitrator's secret key ASK . Otherwise the verifier will simply use it to decrypt a partial signature. Hence, we have the following theorem:

Theorem 4: If $\mathcal{E}$ is a CCA2-secure encryption scheme, $\mathcal{S}$ is UF-CMA-secure signature scheme, and Π is a simulation-sound NIZK proof system in which soundness is preserved even when an adversary is given the secret decryption key dk as input, then the above OFE scheme constructed from verifiably encrypted signatures is secure in our enhanced model.

Proof. We only need to show the security against signers in our enhanced model. To break it, an adversary has to generate a partial signature $\sigma_P = (c, \pi)$ on message m , where π is

an accepting proof but $(c, m, ek, vk) \notin L$. However, this is infeasible even when the adversary has the secret decryption key $\text{ASK} = dk$ as input, because the soundness property of Π now holds even when a dishonest prover is explicitly given the secret decryption key dk as input.

The proof about security against verifiers and security against the arbitrator is the same as that in [16], and therefore we omit it. $\square$

B. Ring signature Paradigm

Huang *et al.* [18] showed that OFE schemes can be constructed based on conventional signatures and ring signatures, and the constructed schemes are secure in the multi-user setting and chosen-key model. We briefly review some intuitions of the construction here.

Given a conventional signature scheme $\text{SIG} = (\text{KGen}, \text{Sig}, \text{Ver})$ and a ring signature scheme $\text{RS} = (\text{KGen}, \text{Sig}, \text{Ver})$, each signer has a key pair for SIG and a key pair for RS . The arbitrator's key pair is a key pair for RS . To partially sign a message, the signer produces a conventional signature σ_S using SIG . Note that the partial signature in the OFE is set as $\sigma_P := \sigma_S$. The full signature σ consists of the partial signature σ_S and a ring signature σ_{RS} generated using RS with respect to message m and the partial signature σ_P while the ring members are the signer and the arbitrator. That is, $\sigma = (\sigma_S, \sigma_{\text{RS}})$. To resolve a partial signature, the arbitrator simply uses its secret key for RS and generates a ring signature σ_{RS} .

The correctness property of OFE protocols constructed following this paradigm holds because of the fact that correctly generated conventional signatures and ring signatures are always valid. The resolution ambiguity property follows the basic anonymity property of a ring signature scheme, which intuitively states that two ring signatures generated by two different members from the same ring respectively are indistinguishable.

Theorem 5: If SIG is a conventional signature scheme that is UF-CMA-secure, and RS is a secure ring signature scheme that is with basic anonymity and existential unforgeability under an adaptive attack, against a static adversary, then the above OFE scheme constructed from conventional signatures and ring signatures is secure in our enhanced model.

Proof. We only need to show the security against signers in our enhanced model. To break it, an adversary has to generate a partial signature σ_P on message m , such that the arbitrator is not able to generate a ring signature with respect to this message m and the partial signature σ_P . However, this is infeasible even if the adversary has the arbitrator's secret key, as the arbitrator can always do so under the ring consisting of the adversary and itself, which holds unconditionally.

The proof about security against verifiers and security against the arbitrator is the same as that in [18], and therefore, we omit it. $\square$

VI. CONCLUSION

A typical meaningful work of modern cryptography is to define proper security models able to capture practical attacks.

Then protocols can be designed and evaluated based on these models. In this paper, we addressed a previously being ignored problem in OFE where a potentially dishonest arbitrator could collude with a signer by sharing its secret key with the signer. We proposed a new security model that for the first time captures this issue, which can be viewed as an enhancement of the previous models. A concrete OFE example was offered to show that our enhanced model guarantees stronger security than the previous models do. We also revisited the two well-known general constructions of OFE protocols, namely the construction based on verifiably encrypted signatures and that based on ring signatures, and showed that protocols secure in our enhanced model can still be constructed following these two paradigms.

REFERENCES

- [1] N. Asokan, M. Schunter, and M. Waidner, "Optimistic protocols for fair exchange," in *Proc. 4th ACM CCS*, 1997, pp. 8–17.
- [2] H. Pagnia and F. C. Grtner, "On the impossibility of fair exchange without a trusted third party," Darmstadt University of Technology, Department of Computer Science, Tech. Rep. TUD-BS-1999-02.
- [3] M. K. Franklin and M. K. Reiter, "Fair exchange with a semi-trusted third party," in *Proc of the 4th ACM CCS*, 1997, pp. 1–5.
- [4] Y. Dodis and L. Reyzin, "Breaking and repairing optimistic fair exchange from podc 2003," in *Proceedings of the 3rd ACM workshop on Digital rights management*, 2003, pp. 47–54.
- [5] S. Lu, R. Ostrovsky, A. Sahai, H. Shacham, and B. Waters, "Sequential aggregate signatures and multisignatures without random oracles," in *EUROCRYPT*, 2006. Springer, 2006, pp. 465–485.
- [6] J. Zhang and J. Mao, "A novel verifiably encrypted signature scheme without random oracle," in *Proc of the 3rd Intl conf on Inf security practice and experience (ISPEC07)*, 2007, pp. 65–78.
- [7] M. Rückert and D. Schröder, "Security of verifiably encrypted signatures and a construction without random oracles," in *Proceedings of Pairing 09*, 2009, pp. 17–34.
- [8] Q. Huang, D. S. Wong, and W. Susilo, "Group-oriented fair exchange of signatures," *Inf. Sci.*, vol. 181, pp. 3267–3283, August 2011.
- [9] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," *Eurocrypt 1998, LNCS 1403*, pp. 591 – 606, 1998.
- [10] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Eurocrypt 2003, LNCS*, vol. 2656, pp. 416–432, 2003.
- [11] J. A. Garay, M. Jakobsson, and P. D. MacKenzie, "Abuse-free optimistic contract signing," in *CRYPTO '99*, 1999, pp. 449–466.
- [12] N. Asokan, V. Shoup, and M. Waidner, "Asynchronous protocols for optimistic fair exchange," in *IEEE Symp on S&P*, 1998, pp. 86–99.
- [13] O. Markowitch and S. Kremer, "An optimistic non-repudiation protocol with transparent trusted third party," in *ISC 2001, LNCS*, 2001, pp. 363–378.
- [14] H. Zhu and F. Bao, "Stand-alone and setup-free verifiably committed signatures," in *CT-RSA '06, LNCS*, vol. 3860, 2006, pp. 159–173.
- [15] Q. Huang, G. Yang, D. S. Wong, and W. Susilo, "Ambiguous optimistic fair exchange," in *ASIACRYPT 2008, LNCS*, 2008, pp. 74–89.
- [16] Y. Dodis, P. J. Lee, and D. H. Yum, "Optimistic fair exchange in a multi-user setting," in *Proc of PKC 07*, 2007, pp. 118–133.
- [17] H. Zhu, W. Susilo, and Y. Mu, "Multi-party stand-alone and setup-free verifiably committed signatures," in *Proc of PKC 07*, 2007, pp. 134–149.
- [18] Q. Huang, G. Yang, D. S. Wong, and W. Susilo, "Efficient optimistic fair exchange secure in the multi-user setting and chosen-key model without random oracles," in *Proc of the CT-RSA 2008*, 2008, pp. 106–120.
- [19] X. Huang, Y. Mu, W. Susilo, W. Wu, J. Zhou, and R. H. Deng, "Preserving transparency and accountability in optimistic fair exchange of digital signatures," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 498–512, 2011.
- [20] B. Barak, R. Canetti, J. B. Nielsen, and R. Pass, "Universally composable protocols with relaxed set-up assumptions," in *FOCS'04*, 2004, pp. 186–195.
- [21] A. W. Dent, "A survey of certificateless encryption schemes and security models," *Int. J. Inf. Sec.*, vol. 7, no. 5, pp. 349–377, 2008.
- [22] C.-P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.
- [23] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc of the 13th ACM conf on Computer and comm security*, ser. CCS '06, 2006, pp. 390–399.
- [24] J. Camenisch and V. Shoup, "Practical verifiable encryption and decryption of discrete logarithms," in *CRYPTO*, 2003, pp. 126–144.
- [25] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," in *Proc. of the 40th Annual Symposium on Foundations of Computer Science*, ser. FOCS '99, 1999, pp. 543–553.

Yang Wang received his Ph.D. degree at the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, Australia. He is now a lecturer in Cyberspace Security College, PLA Information Engineering University, Zhengzhou, Henan 450002, China. His research interests include applied cryptography, fair exchange and security analysis of protocols.

has published numerous publications in the area of digital signature schemes and encryption schemes.

Willy Susilo (SM01) received the Ph.D. degree in computer science from the University of Wollongong, Wollongong, Australia. He is a Professor with the School of Computer Science and Software Engineering and the Director of Centre for Computer and Information Security Research, University of Wollongong. He currently holds the prestigious ARC Future Fellow awarded by the Australian Research Council. His main research interests include cryptography and information security. His main contribution is in the area of digital signature schemes.

and conference papers, including 2 papers in the ACM CCS conference that were named as 'Runners-up for PET Award 2009: Outstanding Research in Privacy Enhancing Technologies.

Man Ho Au (M12) obtained his bachelors (2003) and masters (2005) degrees in the Department of Information Engineering, Chinese University of Hong Kong. He received his PhD from the University of Wollongong (UOW) in July 2009. Currently, he is a lecturer at the School of Computer Science and Software Engineering, UOW. He works in the area of information security. In particular, his research interests include applying public-key cryptographic techniques to systems with security and privacy concerns. He has published over 60 referred journal

design and analysis of digital signatures and security protocols.

Guilin Wang received his PhD degree (2001) in computer science from the Institute of Software, Chinese Academy of Sciences, China. He is currently a senior researcher with Huawei International Pte Ltd, Singapore. Before joining Huawei in Sept of 2013, he was a senior lecturer in the University of Wollongong, Australia; a lecturer in the University of Birmingham, UK, and a research scientist in the Institute for Infocomm Research (I2R), Singapore. His research interest is mainly in the area of applied cryptography and its applications, in particular, the