# Multi-party Stand-alone and Setup-free Verifiably Committed Signatures

Huafei Zhu[1], Willy Susilo[2], Yi Mu[2]

[1] Cryptography Lab, Institute for Infocomm Research, A-star, Singapore
Email: huafei@i2r.a-star.edu.sg
[2] School of Computer Science and Software Engineering, University of Wollongong, Australia, Email: {wsusilo, ymu}@uow.edu.au

**Abstract.** In this paper, we first demonstrate a gap between the security of verifiably committed signatures in the two-party setting and the security of verifiably committed signatures in the multi-party setting. We then extend the state-of-the-art security model of verifiably committed signatures in the two-party setting to that of multi-party setting. Since there exists trivial setup-driven solutions to multi-party verifiably committed signatures (e.g., two-signature based solutions, we propose solutions to the multi-party stand-alone verifiably committed signatures in the setup-free model, and show that our implementation is provably secure under the joint assumption that the underlying Zhu's signature scheme is secure against adaptive chosen-message attack, Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Paillier's encryption is semantically secure and one-way as well as the existence of collision-free one-way hash functions.

**Keywords**: multi-party, setup-free, stand-alone, verifiably committed signatures.

## 1 Introduction

Optimistic fair-exchange protocols was first introduced by Asokan et al, in [1] and formally studied in [2], [3] and [14] in the context of verifiably encrypted signatures. Very recently, Dodis and Reyzin[11] have formalized a unified model for fair-exchange protocols as a new cryptographic primitive called verifiably committed signatures in the two-party setting. Zhu and Bao[20] have shown that the existence of verifiably encrypted signatures implies the existence of the verifiably committed signatures while the existence of verifiably committed signatures does not imply the existence of verifiably encrypted signatures. As a result, the notion of verifiably committed signatures is a general extension of the notion of verifiably encrypted signatures.

A verifiably committed signature can be setup-driven or setup-free[19]. A verifiably committed signature is called setup-driven if an initial key setup protocol between a primary signer and its trusted third party (TTP) must be involved such that at the end of the key setup protocol, the primary signer and its TTP share a prior auxiliary string. This shared auxiliary information enables TTP

to convert any valid partial signature into the corresponding full signature if a conflict occurs between the primary signer and its verifier. A verifiably committed signature is called setup-free if an individual participant needs not to contact his/her arbitrator(s) even for the registration purpose. Namely, no initial key setup procedure between a primary signer and his/her TTP is involved except for one requirement that the primary signer can obtain and verify TTP's certificate and vice versa.

A verifiably committed signature can be stand-alone[19] or not[18]. A verifiably committed signature is called stand-alone if on input a valid partial signature scheme, the distribution of outputs of a resolution algorithm is identical with the distribution of signatures generated by a full signing algorithm. A verifiably committed signature is called non-stand-alone if it is not stand-alone.

The state-of-the-art verifiably committed signatures are only considered in the two-party setting (a primary signer and a verifier, together with an off-line arbitrator). We are interested in studying stand-alone and setup-free verifiably committed signatures in the multi-party setting throughout the paper by demonstrating that the security of two-party setup-free verifiably committed signatures does not guarantee the security of multi-party setup-free verifiably committed signatures.

We stress that the existence of multi-party verifiably committed signatures in the setup-driven model is obvious assuming that the underlying signatures are secure in the sense of [13]. That is, suppose a primary signer's public and secret key pair $(pk_1, sk_1)$ is the public key and secret key pair for the first signature scheme, and at the same time the prime signer and its TTP share another public/secret key $(pk_2, sk_2)$ of the second signature scheme. By $pk$= $(pk_1, pk_2)$ we denote the public key of the entire signature scheme, and by $sk$=$(sk_1, sk_2)$, we denote the corresponding secret keys. Now given a message $m$, the primary signer produces its partial signature $\sigma_1$ on the message $m$. A full signature of the message $m$ is defined as $\sigma$ =$(\sigma_1, \sigma_2)$, where $\sigma_2$ is the signature of $m$ corresponding the public/secret key pair ($pk_2$, $sk_2$). It is easy to verify that this two-signature based solution is a multi-party verifiably committed signature scheme since the security of public key signatures in the two-party setting is preserved in the multi-party setting[6]. This leaves an interesting research problem: *how to implement multi-party stand-alone and setup-free verifiably committed signatures in the standard complexity model*?

The contribution of this paper is of three-fold. In the first fold, we demonstrate that there is a gap between the security of two-party verifiably committed signatures and multi-party verifiably committed signatures. In the second fold, we extend the state-of-the-art security definition of verifiably committed signatures in the two-party setting to that of the multi-party case. In the third fold, we propose an efficient implementation of multi-party stand-alone and setup-free verifiably committed signatures. We are able to show that our implementation is provably secure under the joint assumption that the underlying Zhu's signature scheme is secure against adaptive chosen-message attack, Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Pail-

lier's encryption is semantically secure and one-way as well as the existence of collision-free one-way hash functions. To the best of our knowledge, this is the first implementation of stand-alone and setup-free verifiably committed signature scheme which is provably secure in the multi-party setting.

The rest of this paper is organized as follows: in Section 2, a gap between the security of two-party verifiably committed signatures and multi-party verifiably committed signatures is demonstrated. In Section 3, syntax and security definitions of stand-alone and setup-free verifiably committed signatures in the multi-party setting are introduced and formalized. In Section 4, building blocks on which our implementation is based are briefly sketched. An efficient implementation of multi-party stand-alone and setup-free verifiably committed signatures is proposed in Section 5, and we conclude our work in Section 6.

## 2 A gap between two-party and multi-party verifiably committed signatures

A stand-alone and setup-free verifiably committed signature in the two-party setting based on Cramer and Shoup's signature scheme has been presented in [19]. We are about to demonstrate that although this scheme is provably secure in the two-party setting, it is not secure in the multi-party setting. To show this gap, we first sketch their scheme below:

- primary signer's key generation algorithm $\mathsf{KG}_A$: on input $k_A$, a primary signer Alice runs $\mathsf{KG}_A$ to generate two large safe primes $p_A$ and $q_A$ such that $p_A - 1 = 2p'_A$ and $q_A - 1 = 2q'_A$, where $p'_A, q'_A$ are two $l'$-bit primes. $\mathsf{KG}_A$ also chooses two random elements $x_A$ and $h_B$ from $QR_{n_A}$, where $n_A = p_A q_A$ and $QR_{n_A}$ is the quadratic residue of $Z^*_{n_A}$. Finally, $\mathsf{KG}_A$ outputs a description of a group $G$ of order $s$, and two random elements $g_1$ and $g_2$ of $G$ with order $s$. We stress that in the Cramer and Shoup's signature scheme the choice of group $G$ is independent with $n_A$ (see [8] for more details).
  The public key of Alice is $(n_A, h_A, x_A, g_1, g_2, H)$, along with an appropriate description of $G$ including $s$, where $H$ is a collision-free cryptographic hash function with output length $l$-bit (say, $l$=160). The private key is $(p_A, q_A)$. The primary signer Alice now proves to her CA that all values are correctly generated and then obtains her certificate $\mathrm{Cert}_A$ from her CA;
- arbitrator's key generation algorithm $\mathsf{KG}$: on input $k'$, an arbitrator runs $\mathsf{KG}$ to generate a $k'$-bit RSA modulus $N = p_c q_c$, where $p_c$, $q_c$ are two large safe primes.
  The public key of the arbitrator is $\mathsf{APK}=((1+N), N)$. The private key is $\mathsf{ASK}=(p_c, q_c)$. The arbitrator should prove to his CA that the public and secret key pair is correctly generated and then obtains his certificate $\mathrm{Cert}_B$ from his CA;
- full signing algorithm $\mathsf{Sig}$: To sign a message $m$, Alice runs $\mathsf{Sig}$ to choose at random a $(l+1)$-bit prime number $e$, a string $t \in Z_s$. The equation

$y^e = x_A h_A{}^{H(g_1^t g_2^{H(m)})} \bmod n_A$ is solved for $y$. The corresponding signature $\sigma$ of the message $m$ is $(e, t, y)$.

- full verification algorithm Vf: given a putative triple $(e, t, y)$, a verifier Bob runs Vf to check whether $e$ is an odd $(l+1)$-bit number. If so, Bob further checks the validation of the equation $x_A = y^e h_A{}^{-H(g_1^t g_2^{H(m)})} \bmod n_A$. If the equation is valid, then Bob accepts, otherwise, he rejects.
- partial signing algorithm PSig: on input a message $m$, Alice runs PSig to choose a $(l+1)$-bit prime $e$ and a string $t \in Z_s$. The equation $y^e = x_A h_A{}^{H(g_1^t g_2^{H(m)})} \bmod n_A$ is solved for $y$.
  Alice then computes $u = g_1^t$ and $c = (1+N)^t r^N \bmod N^2$ together with a proof pr that she knows that $u$ contains the same number as the encryption and $t \in I$ using Boudot's protocol [4]. The partial signature $\sigma'$ of message $m$ is defined by $(e, y, u, c, \mathsf{pr})$.
- partial verification algorithm PVf: given a putative signature $\sigma' = (e, y, u, c, \mathsf{pr})$, Bob runs PVf to check whether $e$ is an odd $(l+1)$-bit number. Second PVf checks the validity of the equation $x_A = y^e h_A{}^{-H(u g_2^{H(m)})} \bmod n_A$. If the equation is valid, then PVf further checks the validity of proof pr that $u$ contains the same number as the encryption, and then uses Boudot's protocol to verify that the encrypted value $t \in I$. If it is valid then the verifier accepts, otherwise, it rejects.
- resolution algorithm Res: given $\sigma' = (e, y, u, c, \mathsf{pr})$ and a proof that Bob fulfilled his obligation to the primary signer. The arbitrator first checks validity of the request message. If so, the arbitrator then runs Res to output a valid full signature of $(e, y, t)$ using his decryption key, otherwise, Res rejects the request.

Suppose now an adversary Eve generates two large safe primes $p_E$ and $q_E$ such that $p_E = 2p'_E + 1$ and $q_E = 2q'_E + 1$, where $p'_E, q'_E$ are two $l'$-bit primes. Eve also chooses two random elements $x_E, h_E \in QR_{n_E}$, where $n_E = p_E q_E$ and $QR_{n_E}$ is the quadratic residue of $Z^*_{n_E}$. Eve's now reuses Alice's partial public key $(G, s, g_1, g_2)$. We stress that the reuse of Alice's partial public key is not a problem since the public data $(G, s, g_1, g_2, H)$ can be chosen independently with the private key $(n_A, p_A, q_A)$. We now can show how the malicious verifier Bob and Eve attack Alice below:

- Alice gives her partial signature $(e, y, u, c, \mathsf{pr}_A)$ to Bob, where $\mathsf{pr}_A$ is Alice's proof that $u$ contains the same number as that of $c$ and the encrypted value $t \in I$;
- Bob gives his partial signature $(e', y', u, c, \mathsf{pr}_B)$ to the malicious Eve, where $\mathsf{pr}_B \leftarrow \mathsf{pr}_A$. We stress that although the malicious Bob does not know the exactly hiding value $t \in I$, he can provide a valid proof $\mathsf{pr}_B$ by copying Alice's $\mathsf{pr}_A$.
- Eve asks TTP to open Bob's signature by forwarding partial signature $(e', y', u, c, \mathsf{pr}_B)$ and a proof that Eve fulfilled her obligation to Bob;
- TTP opens $t$ such that $u = g^t$ and $c = E(t, r)$ if and only if $(e', y', u, c, \mathsf{pr}_B)$ and a proof that Eve fulfilled her obligation to Bob are valid; Finally, TTP sends $t$ back to Eve;

– Eve gives $t$ to Bob, who now has the full signature of Alice.

The counterexample shows that the security of verifiably committed signatures in the single setting does not imply the security of verifiably committed signatures in the multi-party setting. We stress that in the above counterexample, the common reference string (the description of $G$ is shared between Alice and Eve) is reused. This is possible since the description of $G$ is independent with APK.

## 3 Multi-party stand-alone and setup-free verifiably committed signatures: syntax and security definitions

### 3.1 Syntax

We now extend (stand-alone and setup-free) two-party verifiably committed signatures [11], [19] and [20] to the multi-party verifiably committed signatures setting.

**Definition 1.** *A multi-party stand-alone and setup-free verifiably committed signature scheme consists of the following algorithms:*

- *arbitrator key generation algorithm KG: on input a security parameter $k$, it returns a public key and secret key pair $(pk, sk)$;*
- *individual key generation algorithms IKG: on input a security parameter $k_i$, it returns a public key and secret key pair $(pk_i, sk_i)$.*
- *full signing and verification algorithms(Sig, Vf): these are conventional signing and verification algorithms. on input a message $m_j$, $pk_i$ and $sk_i$, Sig outputs a full signature $\sigma_{i,j}$ on $m_j$; on input a putative signature $(m_j, \sigma_{i,j}, pk_i)$, Vf will output 1 (accept) or 0 (reject);*
- *partial signing and verification algorithms(PSig, PVf): these are partial signing and verification algorithms, which are similar to ordinary signing and verification algorithms, except they can depend on the public arbitration key $pk$. That is, on input a message ($m_j$, $sk_i$, $pk_i$, $pk$), PSig outputs a partial signature $\sigma'_{i,j}$; on input a putative partial signature ($m_j$, $\sigma'_{i,j}$, $pk_i$, $pk$), PVf outputs 1 (accept) or 0 (reject);*
- *resolution algorithm Res: this is a resolution algorithm run by the arbitrator in case the primary signer $pk_i$ refuses to open her signature $\sigma_{i,j}$ to the verifier, who in turn possesses a valid partial signature $\sigma'_{i,j}$ on $m_j$ and a proof that he fulfilled his obligation to the primary signer[1]. In this case, Res($m_j$, $\sigma'_{i,j}$, $pk_i$, $sk$, $pk$) should output a valid full signature $\sigma_{i,j}$ of $m_j$.*

**Correctness** the correctness property of a multi-party verifiably committed signatures states that:

- $\mathsf{Vf}(m_j, \mathsf{Sig}(m_j, sk_i, pk_i)) = 1$ $(\forall\, j, \forall\, i)$;
- $\mathsf{PVf}(m_j, \mathsf{PSig}(m_j, sk_i, pk_i, pk), pk_i, pk) = 1$ $(\forall\, j, \forall\, i)$;
- $\mathsf{Vf}(m_j, \mathsf{Res}(\mathsf{PSig}(m_j, sk_i, pk_i, pk), sk, pk, pk_i), pk_i) = 1$ $(\forall\, j, \forall\, i)$.

[1] The definition does not deal with any specific question of how a verifier proves to the arbitrator that he/she fulfilled his/her obligation to the primary signer.

### 3.2 The definitions of security

We extend the security definition of Dodis and Reyzin[11] in the two party setting to the multi-party setting. The security definition of multi-party stand-alone and setup-free verifiably committed signatures consists of the following three aspects: security against any primary signer, security against any verifier and security against any arbitrator/TTP.

**Security against malicious primary signer**: intuitively, an individual primary signer should not provide a partial signature which is valid both from the viewpoints of a verifier and an arbitrator but which will not be opened into the primary signer's full signature by the honest arbitrator. More precisely, By $k_i$, we denote the system security parameter of individual user $i$; By $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,\cdots)}$, we denote an oracle of the partial signing procedure $\mathsf{PSig}(pk_i, sk_i, \cdots)$ and by $\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\cdots)}$ an oracle of the resolution procedure $\mathsf{Res}(pk_i, pk, sk, \cdots)$. We require that any probabilistic polynomial time $\mathsf{Adv}$ succeeds with at most negligible probability in the following game.

- arbitrator key generation algorithm $\mathsf{KG}$: on input a security parameter $k$, it outputs $(sk, pk)$;
- individual key generation algorithm $\mathsf{IKG}$: on input a security parameter $k_i$, it outputs $(sk_i^*, pk_i)$, where $\mathsf{IKG}^*(k_i)$ denotes the run of key generator $\mathsf{IKG}$ with the corrupted primary signer $pk_i$ by the adversary, and $sk_i^*$ denotes the adversary's states.
  The honest primary signer $j$ $(j \neq i)$ runs $\mathsf{IKG}$ on input $k_j$ and obtains a public and secret key pair $(pk_j, sk_j)$. The adversary obtains $(pk_j, sk_j)$ and $pk_i$ but not $sk_i^*$ $(1 \leq i, j \leq t(k')$ and $j \neq i)$.
- resolution oracle query $\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\cdots)}$: for each adaptively chosen message $m_j$, the adversary computes its partial signature $\sigma'_{i,j}$ for $m_j$ and forwards $\sigma'_{i,j}$ to the oracle $\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\sigma'_{i,j})}$ to obtain full signature $\sigma_{i,j}$ of message $m_j$, where $1 \leq j \leq t(k_i)$, and $t(\cdot)$ is a polynomial.
- at the end of $\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\cdots)}$ oracle query, the adversary produces a message and its full signature pair $(m_*, \sigma_{i,*})$, i.e.,

$$(m_*, \sigma'_{i,*}) \leftarrow Adv^{\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\cdots)}}(sk_i^*, pk_i, pk); m_* \neq m_j, 1 \leq j \leq t(k');$$

$$\sigma_{i,*} \leftarrow Adv(m_*, \sigma'_{i,*}, sk^*, pk, pk_i)$$

- success of $succ$=$[\mathsf{PVf}(m_*, \sigma'_{i,*}, pk, pk_i) = 1 \wedge \mathsf{Vf}(m_*, \sigma_{i,*}, pk_i) = 0]$.

**Definition 2.** *A multi-party verifiably committed signature is secure against malicious primary signer $pk_i$, if any probabilistic polynomial time adversary Adv associated with resolution oracle, succeeds with at most negligible probability, where the probability takes over coin tosses in $\mathsf{IKG}^*(k_i, \cdot)$, $\mathsf{PSig}(pk_i, \cdot)$ and $\mathcal{O}^{\mathsf{Res}(pk_i,pk,sk,\cdots)}$.*

**Security against malicious verifier**: suppose a primary signer $pk_i$ and a verifier $v$ are trying to exchange signature in a fair way. The primary signer $pk_i$ wants to commit to the transaction by providing his/her partial signature. Of course, it should be computationally infeasible for the verifier $v$ to compute the corresponding full signature from any partial signature[2]. More formally, we require that any probabilistic polynomial time adversary $Adv$ succeeds with at most negligible probability in the following game:

- arbitrator key generation algorithm $\mathsf{KG}$: on input a security parameter $k$, it outputs $(sk, pk)$;
- individual key generation $\mathsf{IKG}$: on input a security parameter $k_j$, it outputs $(sk_j, pk_j)$, where $\mathsf{IKG}(k_j)$ denotes the run of key generator $\mathsf{IKG}$ with the corrupted primary signer $pk_j$ by the adversary, and $sk_j$ denotes the adversary's states. The honest primary signer $i$ $(i \neq j)$ runs $\mathsf{IKG}(k_i)$, obtains a public and secret key pair $(pk_i, sk_i)$. The adversary obtains $(pk_j, sk_j)$ and $pk_i$ but not $sk_i$ $(1 \leq i, j \leq t(k')$ and $j \neq i)$.
- $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdot)}$ and $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)}$ oracle queries: for each adaptively chosen message $m_j$, the adversary obtains a partial signature $\sigma'_{i,j}$ of message $m_j$ by querying the partial signing oracle $\mathcal{O}^{\mathsf{PSig}(i,m_j)}$. The adversary forwards $\sigma'_{i,j}$ to the resolution oracle $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\sigma'_{i,j})}$ to obtain the full signature $\sigma_{i,j}$ of message $m_j$, where $1 \leq j \leq t(k_i)$, and $t(\cdot)$ is a polynomial.
- at the end of oracle queries to $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$ and $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)}$, the adversary outputs a message-partial signature pair $(m_*, \sigma'_{i,*})$. On input $(m_*, \sigma'_{i,*})$, the adversary further outputs a message-full signature pair $(m_*, \sigma_{i,*}) \leftarrow Adv^{\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\sigma'_{i,*})},\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\sigma'_{i,*})}}$.
- success of adversary $succ$: $= [\mathsf{Vf}(m_*, \sigma_{i,*}, pk_i) = 1 \wedge m_* \notin Query\ (Adv, \mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)})]$, where $Query(\ Adv, \mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)})$ is the set of valid queries the adversary $Adv$ asked to the resolution oracle $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)}$, i.e., $(m_*, \sigma'_{i,*})$ such that $\mathsf{Vf}(m_*, \sigma'_{i,*}) = 1$.

**Definition 3.** *A multi-party verifiably committed signature is secure against a malicious verifier, if any probabilistic polynomial time adversary Adv which is associated with a partial signing oracle* $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$ *and a resolution oracle* $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)}$*, succeeds with at most negligible probability, where the probability takes over coin tosses in* $(pk_i, sk_i) \leftarrow \mathsf{IKG}(k_i)$ *and* $(pk, sk) \leftarrow \mathsf{KG}(k)$*,* $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$ *and* $\mathcal{O}^{\mathsf{Res}(pk_i,sk,pk,\cdots)}$*.*

**Security against semi-trusted arbitrator**: even though the arbitrator is semi-trusted, a primary signer does not want this arbitrator to produce a valid signature which the primary signer do not intend on producing. To achieve this goal, we require that any probabilistic polynomial time adversary $Adv$ associated

[2] The security preventing a malicious third party from forging valid partial signatures is stated as security against any malicious arbitrator below as a malicious arbitrator is the most powerful adversary in the security model.

with partial signing oracle $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$, succeeds with at most negligible probability in the following game:

- key generation algorithm $\mathsf{KG}^*$: on input security parameter $k$, $\mathsf{KG}^*(k)$ outputs $(sk^*, pk)$, where $\mathsf{KG}^*(k)$ is run by the dishonest arbitrator.
- individual key algorithm $\mathsf{IKG}$: on input a security parameter $k_j$, it outputs $(sk_j, pk_j)$, where $\mathsf{IKG}(k_j)$ denotes the run of key generator $\mathsf{IKG}$ with the corrupted primary signer $pk_j$ by the adversary, and $sk_j$ denotes the adversary's states. The honest primary signer $i$ $(i \neq j)$ runs $\mathsf{IKG}(k_i)$, obtains a public and secret key pair $(pk_i, sk_i)$. The adversary obtains $(pk_j, sk_j)$ and $pk_i$ but not $sk_i$ $(1 \leq i, j \leq t(k')$ and $j \neq i)$.
- $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$ oracle query: for each adaptively chosen message $m_j$, the adversary obtains the partial signature $\sigma'_{i,j}$ for $m_j$ from the oracle $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,m_j)}$, where $1 \leq j \leq t(k')$.
- at the end of the partial partial signing oracle query, the adversary produces a message-full signature pair $(m_*, \sigma_{i,*})$, i.e.,

$$(m_*, \sigma_{i,*}) \leftarrow Adv^{\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,m_*)}}(sk^*, pk, pk_i).$$

- success of adversary is defined as follows:

$$succ = [\mathsf{Vf}(m, \sigma, pk_i) = 1 \wedge m_* \notin Query(Adv, \mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)})]$$

  where $Query(\ Adv, \mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$ is the set of valid queries $Adv$ asked to the partial oracle such that $\mathsf{PVf}(m_j, \sigma'_{i,j}) = 1$.

**Definition 4.** *A multi-party verifiably committed signature is secure against malicious arbitrator, if any probabilistic polynomial time adversary Adv associated with partial signing oracle P, succeeds with at most negligible probability, where the probability takes over coin tosses in* $(pk_i, sk_i) \leftarrow \mathsf{IKG}(k_i)$ *and* $(pk, sk^*) \leftarrow \mathsf{KG}^*(k)$, $\mathcal{O}^{\mathsf{PSig}(pk_i,sk_i,pk,\cdots)}$.

**Definition 5.** *A multi-party verifiably committed signature is secure if it is secure against any malicious primary signer, malicious verifier and malicious arbitrator.*

## 4 Building blocks

Before we propose our implementation, we would like to sketch the following building blocks on which our protocol is based.

### 4.1 Paillier's cryptographic system

Paillier investigated a novel computational problem, called Composite Residuosity Class Problem, and its applications to public key cryptography in [15]. Our construction of multi-party verifiably committed signatures will heavily rely on this probabilistic encryption scheme sketched below.

- the public key is a $\kappa$-bit RSA modulus $N = PQ$, where $P$, $Q$ are two large safe primes, where $|P|=|Q| = 2\kappa$. the private key is $(P, Q)$;
- the plain-text space is $Z_N$ and the cipher-text space is $Z^*_{N^2}$;
- to encrypt $\alpha \in Z_N$, one chooses $R_a \in Z^*_N$ uniformly at random and computes the cipher-text as $E_{PK}(a, R_a) = (1 + N)^a R_a^N \bmod N^2$.
- given $c = (1 + N)^a R_a^N \bmod N^2$, and trapdoor information $(P, Q)$, one can first computes $c_1$ $(= c \bmod N)$, and then compute $R_a$ from the equation $R_a = c_1^{N^{-1} \bmod \phi(N)} \bmod N$; Finally, one can compute $a$ from the equation $cR_a^{-N} \bmod N^2 = 1 + aN$.
- the encryption function is homomorphic, i.e., $E_{PK}(a_1, R_1) \times E_{PK}(a_2, R_2) \bmod N^2 = E_{PK}(a_1 + a_2 \bmod N,\ R_1 \times R_2 \bmod N)$.

### 4.2 Fujisaki-Okamoto commitment scheme

Let $\tau$ be a security parameter. The public key is a $\tau$-bit RSA modulus $n=pq$, where $p$, $q$ are two large safe primes. We assume that neither a committer nor a receiver knows factorization $n$. Let $g_1$ be a generator of $QR_n$ and $g_2$ be an element of large order of the group generated by $g_1$ such that both discrete logarithm of $g_1$ in base $g_2$ and the discrete logarithm of $g_2$ in base $g_1$ are unknown by the committer or the receiver. We denote $C(a, r_a) = g_1^a g_2^{r_a} \bmod n$ a commitment to $a$ in bases $(g_1, g_2)$, where $r_a$ is randomly selected over $\{0, 2^s n\}$, where $s$ is a security parameter. This commitment scheme first appeared in [12] and reconsidered by Damgård and Fujisaki [10] is statistically hiding and computationally binding, i.e.,

- a committer is unable to commit itself to two values $a_1, a_2$ such that $a_1 \neq a_2$ in $Z$ by the same commitment unless the committed can factor $n$ or solves the discrete logarithm of $g_1$ in base $g_2$ or the the discrete logarithm of $g_2$ in base $g_1$;
- $C(a, r_a)$ statistically reveals no information to the receiver, i.e., there is a simulator which outputs simulated commitments to $a$ which are statistically indistinguishable from true ones.
- this commitment is homomorphic, i.e., $C(a+b, r_a+r_b) = C(a, r_a) \times C(b, r_b)$.

### 4.3 Boudot's protocol

With the help of Fujisaki-Okamoto commitment scheme, an efficient protocol allows Alice to prove to Bob that a committed number $x \in [a, b]$ belongs to the desired interval $[a, b]$ $(0 < a \in Z$ and $a < b \in Z)$, has been proposed by Boudot [4]. The idea behind Boudot's protocol is that to achieve a proof of membership without tolerance, the size of $x$ is first enlarged, and then Alice proves to Bob that the value $2^T x$ lies in interval $< 2^T a - 2^T, 2^T b + 2^T >$ with tolerance (a proof with tolerance is easier than a proof without tolerance, we refer the reader to [4] for further reference), and thus $x \in [a, b]$. Boudot's protocol is zero-knowledge proof of knowledge and it is sound assuming that the underlying Fujisaki-Okamoto commitment scheme is statistically hiding and computationally binding.

### 4.4 Proof equality of a committed number and an encryption in different moduli

An efficient implementation for proving the equality of a committed number and an encryption has been proposed by Damgård and Jurik[9]:

- let $\lambda$ be maximum bit length of $x$. Let $C$ be a commitment $C(x, r_x) = g_1^x g_2^{r_x} \bmod n$ computed from Fujisaki-Okamoto commitment scheme and $E$ be a cipher-text $E(x, R_x) = (1+N)^x {R_x}^N \bmod N^2$ computed from Paillier's encryption scheme, a prover should provide a proof that $C$ and $E$ hide the same value $x$.
- the prover chooses at random $\omega \in \{0,1\}^{\lambda+2l}$, where $l$ is a security parameter. The prover sends $C' = g_1^{\omega} g_2^{r_\omega}$ and $E' = E(\omega, R_\omega)$ to the verifier. Here we assume that the security parameter $\kappa$ of Paillier's system is larger than $(\lambda + 2l)$
- the verifier chooses a $l$-bit challenge $f$;
- the prover opens the encryptions $C'C^f \bmod n$ and $E'E^f \bmod N^2$, to reveal in both cases the number $z = \omega + xf$ defined over the integer domain. The verifier checks the opening were correct.

The protocol can be made non-interactive in the standard way using a hash function $\mathcal{RO}$ and the Fiat-Shamir technique. It is also statistically zero-knowledge in the random oracle mode.

### 4.5 Proof equality of a committed number and a discrete logarithm in different moduli

Let $l$, $t$ and $s$ be three security parameters. Assume that a prover Alice holds a secret value $x \in \{0, T\}$. We denote by $E_1 = g_1^x g_2^r \bmod n_1$, be a commitment computed from Fujisaki-Okamoto commitment scheme and $E_2 = g^x \bmod n_2$ be a discrete logarithm of $QR_{n_2}$ modulo $n_2$, where $n_2 = p_2 q_2$, $p_2 = 2p_2' + 1$, $q_2 = 2q_2' + 1$ and $QR_{n_2} = < g >$. A prover Alice wants to prove to a verifier Bob that she knows $x$ and $r \in \{-2^s n_1 + 1, 2^s n_1 - 1\}$ such that $E_1 = g_1^x g_2^r \bmod n_1$ and $E_2 = g^x \bmod n_2$.

- Alice picks random strings $\omega \in \{1, \cdots, 2^{l+t}T - 1\}$ and $\rho \in \{1, \cdots, 2^{l+t+s}n - 1\}$. Alice then computes $\pi_1 = g_1^{\omega} g_2^{\rho} \bmod n_1$ and $\pi_2 = g^{\omega} \bmod n_2$; Finally, Alice sends $(\pi_1, \pi_2)$ to Bob;
- Bob sends $f \in \{0,1\}^{2t}$ to Alice;
- Alice computes $\tau_1 = \omega + fx$ and $\tau_2 = \rho + fr$ (over the integer domain $Z$);
- Bob checks whether $g_1^{\tau_1} g_2^{\tau_2} = \pi_1 E_1^f \bmod n_1$ and $g^{\tau_1} = \pi_2 E_2^f \bmod n_2$.

This protocol originally appeared in [5] and independently in [7] is a zero-knowledge proof of equality of a committed number and a discrete logarithm in different moduli. Again, the protocol can be made non-interactive in the standard way using a hash function $\mathcal{RO}$ and the Fiat-Shamir technique. It is also statistically zero-knowledge in the random oracle mode.

### 4.6 Zhu's signature scheme

Our multi-party verifiably committed signatures is built on the top of Zhu's signature (see [16], [17] and [18] for more details).

- Key generation algorithm: Let $p, q$ be two large safe primes (i.e., $p-1 = 2p'$ and $q-1 = 2q'$, where $p', q'$ are two primes with length $(l'+1)$). Let $n = pq$ and $QR_n$ be the quadratic residue of $Z_n^*$. Let $X, g, h \in QR_n$ be three generators chosen uniformly at random. The public key is $(n, g, h, X, H)$, where $H$ is a collision free hash function with output length $l$. The private key is $(p, q)$.
- Signature algorithm: To sign a message $m$, a $(l+1)$-bit prime $e$ and a string $t \in \{0,1\}^l$ are chosen at random. The equation $y^e = Xg^t h^{H(m)} \mathrm{mod} n$ is solved for $y$. The corresponding signature of the message $m$ is $(e, t, y)$.
- Verification algorithm: Given a putative triple $(e, t, y)$, the verifier checks that $e$ is an $(l+1)$-bit odd number. Then it checks the validity of $X = y^e g^{-t} h^{-H(m)} \mathrm{mod} n$. If the equation is valid, then the signature is valid. Otherwise, it is rejected.

Zhu's signature scheme is provably secure against adaptive chosen-message attack under joint assumptions that the strong RSA problem is hard and the discrete logarithm defined over $QR_n$ is hard as well as the underlying hash function $H$ is collision free.

## 5 Stand-alone, setup-free verifiably committed signatures in multi-party setting

### 5.1 Implementation

With the help of these building blocks listed above, we can now describe our implementation of multi-party stand-alone, setup-free verifiably committed signatures below.

- arbitrary key generation algorithms ($\mathsf{KG}_E$, $\mathsf{KG}_C$): on input a security parameter $\kappa$, an arbitrary runs $\mathsf{KG}_E$ (it is a key generator of Paillier's encryption algorithm) to generate $\kappa$-bit RSA modulus $N = PQ$, where $P$, $Q$ are two large safe primes. The plain-text space is $Z_N$ and the cipher-text space is $Z_{N^2}^*$.
  On input $\tau$, the arbitrator runs $\mathsf{KG}_C$ (it is a key generator of Okamot-Fujisaki's commitment scheme) to generate $\tau$-bit RSA modulus $N_c{=}P_cQ_c$, where $P_c$ and $Q_c$ are two large prime numbers. $\mathsf{KG}_C$ also outputs two random elements $g, h \in QR_{N_c}$.
  The public key $pk = (pk_E, pk_C)$, where $pk_E = (1+N, N)$ and $pk_C = (N_c, g, h)$. The secret keys $sk = (sk_E, sk_C)$, where $sk_E = (P, Q)$ and $sk_C = (P_c, Q_c)$.
- individual key generation algorithm $\mathsf{IKG}$: on input a security parameter $(k_i, l_i, l_i')$, the $i^{th}$ user runs $\mathsf{IKG}$ (it is a key generation algorithm of Zhu's signature scheme) to generate two large primes $p_i$ and $q_i$ such that $p_i - 1 = 2p_i'$ and $q_i - 1 = 2q_i'$, where $p_i', q_i'$ are two $(l_i'+1)$-bit strings.

Let $n_i = p_i q_i$ and $QR_{n_i}$ be the quadratic residue of $Z^*_{n_i}$. Let $g_i, h_i$ be two generators of $QR_{n_i}$ chosen uniformly at random. The public key is the $i^{th}$ user is $(n_i, g_i, h_i, x_i, H_i)$, where $x_i \in QR_{n_i}$ and $H_i$ is a collision free hash function with output length $l_i$. The private key is $(p_i, q_i)$.

- full signature algorithm Sig: to sign a message $m_j$, a $(l_i + 1)$-bit prime $e_{i,j}$ and a $l_i$ bit string $t_{i,j}$ are chosen at random. The equation ${y_{i,j}}^{e_{i,j}} = x_i g_i^{t_{i,j}} h_i^{H_i(m_j)} \bmod n_i$ is solved for $y_j$. The corresponding signature $\sigma_{i,j}$ of the message $m_j$ is $(e_{i,j}, t_{i,j}, y_{i,j})$.
- verification algorithm Vf: given a putative triple $(e_{i,j}, t_{i,j}, y_{i,j})$, Vf first checks that $e_{i,j}$ is an odd $(l_i + 1)$-bit number. Second it checks the validation that $x_i = {y_{i,j}}^{e_{i,j}} g_i^{-t_{i,j}} {h_i}^{-H_i(m_j)} \bmod n_i$. If the equation is valid, then Vf accepts, otherwise, it rejects.
- partial signing algorithm PSig: on input a message $m_j$, $(l_i+1)$-bit prime $e_{i,j}$ and a $l_i$ string $t_{i,j}$ are chosen at random. The equation ${y_{i,j}}^{e_{i,j}} = x_i g_i^{t_{i,j}} {h_i}^{H_i(m_j)} \bmod n_i$ is solved for $y_j$. Then the $i^{th}$ user (say Alice) further performs the following computations:
  - $u_{i,j} \leftarrow g_i^{t_{i,j}}$;
  - $E_{i,j} \leftarrow E(pk_E, t_{i,j})$, where $E(pk_E, t_{i,j}) = (1+N)^{t_{i,j}} R_{i,j}^N \bmod N^2$;
  - $C_{i,j} \leftarrow C(pk_c, t_{i,j})$, where $C(pk_C, t_{i,j}) = g^{t_{i,j}} h^{r_{i,j}} \bmod N_c$;
  - a proof $\mathsf{pr}_{i,j}$ that she knows that $u_{i,j}$ contains the same number as that hidden by $E(pk_E, t_{i,j})$ as well as $t_{i,j}$ is a $l_i$-bit string. More precisely, the proof $\mathsf{pr}_{i,j}$ consists of the following three statements:
    - the prover runs the protocol specified in Section 4.4 and proves to the verifier Bob the equality of the committed number by $C_{i,j}$ and the encrypted number by $E_{i,j}$;
    - the prover runs the protocol specified in Section 4.5 and proves to the verifier Bob the equality of the committed number by $C_{i,j}$ and the discrete logarithm by $u_{i,j}$ on base $g_i$;
    - the prover runs the protocol specified in Section 4.3 and proves to the verifier Bob that the committed number by $C_{i,j}$ lies in the interval $\{0, 2^{l_i} - 1\}$ .

  The partial signature is denoted by $\sigma'_{i,j} = (e_{i,j}, y_{i,j}, u_{i,j}, c_{i,j}, \mathsf{pr}_{i,j})$.
- The corresponding partial signature verification algorithm PVf: given a putative signature $\sigma'_{i,j} = ((e_{i,j}, y_{i,j}, u_{i,j}, c_{i,j}, \mathsf{pr}_{i,j}))$, the verifier Bob performs the following checks:
  - checking $e_{i,j}$ is an odd $(l_i + 1)$-bit number.
  - checking the validity of the equation $x_i = {y_{i,j}}^{e_{i,j}} g_i^{-t_{i,j}} {h_i}^{-H_i(m_j)} \bmod n_i$.
  - checking the validity of proof $\mathsf{pr}_{i,j}$;
  - if all checks are valid then the verifier accepts, otherwise, it rejects.
- resolution algorithm Res: given $\sigma'_{i,j} = ((e_{i,j}, y_{i,j}, u_{i,j}, c_{i,j}\ \mathsf{pr}_{i,j}))$, and a proof that Bob fulfilled his obligation to the primary signer $pk_i$. If the verification is passed, then the arbitrator outputs a valid full signature $(e_{i,j}, y_{i,j}, t_{i,j})$ using his decryption key $sk_E$, otherwise, it rejects.

This ends the description of our protocol. We stress that the technique presented in this section can be easily extended to the case where the underlying signature

scheme is Cramer-Shoup's hash signature such that individual group $G_i$ is chosen independently.

### 5.2 The proof of security

The proof of security follows that presented in [19]. We also stress that the technique presented in this section can be applied to the case where the underlying signature scheme is Cramer-Shoup's hash signature with the restriction that individual group $G_i$ is chosen independently and is never reused.

**Lemma 1.** *The verifiably committed signature is secure against malicious primary signer in the multi-party setting.*

*Proof.* Suppose the $i^th$ user Alice is able to provide a valid partial signature $\sigma'_{i,j} = (e_{i,j}, y_{i,j}, u_{i,j}, E_{i,j}, C_{i,j}, \mathsf{pr}_{i,j})$ corresponding to a message $m_j$, where the valid proof $\mathsf{pr}_{i,j}$ means that she knows that $u_{i,j}$ contains the same number as the encryption $E_{i,j}$ and the encrypted value $t_{i,j} \in I$, $I = \{0, 2^{l_i} - 1\}$. Since $\sigma'_{i,j}$ is valid from the viewpoints of its verifier and TTP, by rewinding Alice, both verifier and cosigner can extract $t_{i,j} \in I$ such that

$$u_{i,j} = g_1^{t_{i,j}}, E_{i,j} = E(pk_E, t_{i,j}), y_{i,j}^{e_{i,j}} = x_i g_i^{t_{i,j}} h_i^{H_i(m_j)}, t_{i,j} \in I.$$

It follows that the designated TTP can always transform any valid partial signature scheme into the correspondenting valid signature $\sigma_{i,j}$=$(e_{i,j}, y_{i,j}, t_{i,j})$.

**Lemma 2.** *Our construction is secure against malicious verifier under the joint assumptions that Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Paillier's encryption scheme is semantically secure and one-way.*

*Proof.* We convert any attacker $\mathcal{B}$ that attacks our verifiably committed signature scheme into an inverter $\mathcal{B}'$ of the underlying encryption scheme. That is, given a random cipher-text $E_{i,j}$, $\mathcal{B}'$ will obtain the corresponding plain-text $m_j$ with non-negligible probability with the help of the attacker $\mathcal{B}$. This can be done as follows:

- $\mathcal{B}'$ runs $\mathsf{IKG}$ to generate the $i^{th}$ primary signer's public/secret key $(pk_i, sk_i)$ as that in the real verifiably committed signature scheme and obtains the public and secret key pair $(pk_i, sk_i)$.
- $\mathcal{B}'$ then runs $\mathsf{KG}$ to generate the arbitrator's public/secret key $(pk, sk)$ as that in the real verifiably committed signature scheme and obtains $pk$ but not $sk$ from the arbitrator.

Given the target cipher-text $E_{i,j}$, we first describe a simulator of the partial signature oracle $\mathcal{O}^{\mathsf{PSig}(pk_i, sk_i, pk, \cdots)}$ as follows:

Let $q_{\mathsf{PSig}}$ be the total number of queries made by $\mathcal{B}$, and let $\iota$ be a random number chosen from $\{1, q_{\mathsf{PSig}}\}$ by $\mathcal{B}'$.

- If $i \in \{1, q_{\mathsf{PSig}}\}$ and $i \neq \iota$, then $\mathcal{B}'$ runs the partial signing oracle as the real partial signature scheme;
- If $i \in \{1, q_{\mathsf{PSig}}\}$ and $i = \iota$, for the given target cipher-text $E_{i,j}$, $\mathcal{B}'$ chooses a random string $f_{i,j}$, $z_{i,j}$ and $u_{i,j}$ in the correct interval specified in the real protocol and then $\mathcal{B}'$ computes $E'_{i,j}$ from the equation $E(pk_E, z) = E'_{i,j}$ $E_{i,j}^{f_{i,j}}$. At the same time, it computes $u'_{i,j}$ from the equation $g_i^{z_{i,j}} = u'_{i,j}$ $u_{i,j}^{f_{i,j}}$.
- Given $u_{i,j}$, $\mathcal{B}'$ computes $(e_{i,j}, y_{i,j})$ from the equation $y_{i,j}^{e_{i,j}} = x_i u_{i,j} h_i^{H_i(m_j)}$, this is possible since $\mathcal{B}'$ knows the secret key $sk_i$ (notice that $\mathcal{B}'$ assigns $f_{i,j}$ to be the hash value of the random oracle $\mathcal{RO}$ if the specified protocol in Section 4.5 is non-interactive).
  Similarly, for the given $u_{i,j}$, there exists a simulator that can simulate views for the following proofs:
  - a proof of equality of the committed number $C_{i,j}$ and the discrete logarithm $\log_{g_i}(u_{i,j})$, where $C_{i,j}$ is a forgery commitment;
  - a proof of equality of the committed number by $C_{i,j}$ and the encrypted number by $E_{i,j}$;
  - a proof that the committed number by $C_{i,j}$ lies in the correct interval.

  Such a simulator can be defined by the concatenation of individual simulators for the above zero-knowledge proof systems since Boudot's protocol, Damgård and Jurik's protocol, as well as Boudot, and Camenisch and Michels' protocols are zero-knowledge proof systems (see Section 4.3, Section 4.4 and Section 4.5 for more details). As a result, the existence of such a simulator following the definition of the zero-knowledge proof system immediately.

$\mathcal{B}'$ simulates $\mathcal{O}^{\mathsf{Res}(pk_i, sk, pk, \cdots)}$ oracle queries as follows:

- If $(m_j, \sigma'_{i,j})$ that is in the partial signature query list and if $j \neq \iota$, then $\mathcal{O}^{\mathsf{Res}(pk_i, sk, pk, \cdots)}$ outputs $t_i$;
- If $(m_j, \sigma'_{i,j})$ that is in the partial signature query list and if $j = \iota$, then $\mathcal{O}^{\mathsf{Res}(pk_i, sk, pk, \cdots)}$ outputs $\perp$;
- If $(m_j, \sigma'_{i,j})$ that is not in the partial signature query list, then $\mathcal{O}^{\mathsf{Res}(pk_i, sk, pk, \cdots)}$ outputs $\perp$.

Notice that the probability that the simulator outputs $\perp$ is $1 - 1/q_{\mathsf{PSig}}$ for the queries whose partial signatures are listed in the $\mathcal{O}^{\mathsf{PSig}(pk_i, sk_i, pk, \cdots)}$ oracle query. Thus when the adversary outputs a valid full signature $(m^*, \sigma^*)$ whose partial signature is in the list of $\mathcal{O}^{\mathsf{PSig}(pk_i, sk_i, pk, \cdots)}$ oracle query, the probability that $\mathcal{B}'$ can invert the target cipher-text $E_{i,j}$ with probability at least $\epsilon/q_{\mathsf{PSig}}$, where $\epsilon$ stands for the probability that $\mathcal{B}$ can break our verifiably committed signature scheme.

**Lemma 3.** *Our construction is secure against malicious arbitrator under the joint assumptions that the underlying Zhu's signature scheme is secure against*

*adaptive chosen-message attack, Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Paillier's encryption scheme is semantically secure.*

*Proof.* Suppose an arbitrator is able to forgery partial signature $\sigma'_{i,j}$ with non-negligible probability, then by rewinding the arbitrator, we can extract $t_{i,j}$ from the valid proof $\mathsf{pr}_{i,j}$. It follows that the arbitrator is able to output a valid forgery signature from Zhu's signature scheme with non-negligible probability. Since the underlying Zhu's signature scheme signature has proved to be secure against adaptive chosen-message attack under joint assumptions of the strong RSA problem as well as the existence of collision free hash function. It follows that our construction is secure against semi-trusted arbitrator under joint assumptions that the hardness of the strong-RSA problem and the existence of collision free hash functions.

In summary, we have proved the main result below:

**Theorem 1.** *The stand-alone, setup-free verifiably committed signature scheme constructed above is provably secure under the joint assumption that the underlying Zhu's signature scheme is secure against adaptive chosen-message attack, Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Paillier's encryption is semantically secure and one-way.*

## 6 Conclusion

In this paper, we have demonstrated a gap between the security of a two-party verifiably committed signatures and the security of multi-party verifiably committed signatures. We also have extended Dodis and Leyzin's security model for the two-party verifiably committed signatures to the multi-party setting. Finally, we have implemented an efficient stand-alone and setup-free verifiably committed signatures in the multi-party setting and shown that our implementation is provably secure under the joint assumptions that the underlying Zhu's signature scheme is secure against adaptive chosen-message attack, Fujisaki-Okamoto's commitment scheme is statistically hiding and computationally binding and Paillier's encryption is semantically secure and one-way.

## References


1. N.Asokan, M.Schunter and M.Waidner: Optimistic Protocols for Fair Exchange. ACM Conference on Computer and Communications Security 1997: 7 - 17.
2. N.Asokan, V.Shoup and M.Waidner: Optimistic Fair Exchange of Digital Signatures (Extended Abstract). EUROCRYPT 1998: 591 - 606.
3. D.Boneh, C.Gentry, B.Lynn and H.Shacham: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. EUROCRYPT 2003: 416 -432
4. F.Boudot: Efficient Proofs that a Committed Number Lies in an Interval. Proc. of EUROCRYPT 2000: 431 - 444, Springer Verlag.

5. F.Boudot and J.Traore: Efficient Publicly Verifiable Secret Sharing Schemes with Fast or Delayed Recovery. ICICS'99, 87- 102.
6. Ran Canetti: Universally Composable Signature, Certification, and Authentication. CSFW 2004, 219.
7. J.Camenisch and M.Michels: Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes. EUROCRYPT 1999: 107-122.
8. R.Cramer and V.Shoup. Signature scheme based on the Strong RAS assumption. 6th ACM Conference on Computer and Communication Security, Singapore, ACM Press, November 1999.
9. I.Damgård and M.Jurik: Client/Server Tradeoffs for Online Elections. Proc. of Public Key Cryptography 2002: 125 - 140. Springer Verlag.
10. I.Damgård and E.Fujisaki: A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order. Proc. of ASIACRYPT 2002: 125 - 142, Springer Verlag.
11. Y.Dodis and L.Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003, ACM Workshop on Digital Rights Management (DRM), October 2003.
12. E.Fujisaki and T.Okamoto. Statistically zero knowledge protocols to prove modular polynomial relations. Crypto'97. 16 - 30, 1997.
13. S.Goldwasser, S.Micali and R.L.Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2): 281 - 308 (1988).
14. S.Lu, R.Ostrovsky, A.Sahai, H.Shacham and B.Waters: Sequential Aggregate Signatures and Multisignatures Without Random Oracles. EUROCRYPT 2006: 465-485
15. P.Paillier: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Proc. of EUROCRYPT 1999: 223 - 238, Springer Verlag.
16. H.Zhu. New Digital Signature Scheme Attaining Immunity to Adaptive Chosen-message attack. Chinese Journal of Electronics, Vol.10, No.4, Page 484-486, Oct, 2001.
17. H. Zhu. A formal proof of Zhu's signature scheme, http://eprint.iacr.org/, 2003/155.
18. H.Zhu: Constructing Committed Signatures from Strong-RSA Assumption in the Standard Complexity Model. Public Key Cryptography 2004: 101-114.
19. H.Zhu and F.Bao: Stand-Alone and Setup-Free Verifiably Committed Signatures. CT-RSA 2006: 159-173.
20. H.Zhu and F.Bao: More on Stand-Alone and Setup-Free Verifiably Committed Signatures. ACISP 2006: 148 -158.