Setup-driving Verifiably Committed Signatures within Standard Complexity Model*

Huafei Zhu

Department of Information Science and Electronics, Engineering, Zhejiang University (YuQuan Campus)
HangZhou, PR. China (Email: zhuhf@zju.edu.cn)

(Received July 21, 2005; revised and accepted Aug. 23, 2005)

Abstract

In this paper, a setup-driving verifiably committed signature based on the strong RSA assumption within the standard complexity model is presented. The idea behind our construction is that given any valid partial signature of a message m, if an arbitrator with its auxiliary input is able to generate variables called the resolution of message m such that the distribution of the resulting variables is indistinguishable from that generated by the primary signer alone from the viewpoints of all verifiers, then from which a committed signature can be derived.

Keywords: Fair exchange protocols, strong RSA assumption, verifiably committed signatures

1 Introduction

The research of fair exchange protocols has a rich history due to its fundamental importance, we refer the reader to [10] for general reference. A fair-exchange protocol typically consists of three participants: a client (a primary signer), a merchant (a co-signer) and a trusted third party (TTP). TTP can be on-line at the expense of the TTP becoming a potential bottleneck, or off-line, meaning that it only gets involved when something goes wrong. Off-line fair-exchange protocols can be classified into two categories: with or without initial-key-setup procedures. An off-line fair-exchange protocol is called *setup-free* if no initial-key-setup procedure run between a primary signer and its TTP is involved except for one requirement that the primary signer can obtain and verify TTP's certificate and vice versa. An off-line fair-exchange protocol is called *setup-driving* if an initial-key-setup protocol run between a primary signer and its TTP must be involved such that at the end of the key setup protocol, the primary signer and its TTP share prior auxiliary information. This shared auxiliary information enables TTP to convert any valid partial signature into the corresponding full signature if a confliction occurs between the primary signer and its co-signer (and thus the fairness of protocols can be achieved inherently).

In PODC 2003, Park, Chong, Siegel and Ray [15] provided a novel method of constructing fair exchange protocol by distributing the computation of RSA signature. This approach avoids the design of verifiable encryption scheme at the expense of having the arbitrator store a piece of prime signer's secret key (please refer to [1, 2, 3, 4] for more details). Based on Park et.al's study, Dodis and Reyzin [10] presented a unified model for non-interactive fair exchange protocols which results in a new primitive called verifiably committed signatures later. Verifiably committed signatures are the following thing: Alice can produce a partial signature to Bob; upon receiving what she needs from Bob, she can convert it to a full signature. If she refuses, the trusted third party Charlie can do it for her upon receipt of partial signature and proper verification that Bob fulfilled his obligation to Alice.

Park, Chong, Siegel and Ray's fair exchange protocol is actually a verifiably committed signature scheme since the mechanism of the non-interactive fair exchange is the same thing as a verifiably committed signature. Unfortunately this verifiably committed signature is totally breakable in the registration phase [10]. Dodis and Reyzin [10] then presented a remedy scheme by utilizing Boldyreva's non-interactive two-party multi-signature scheme [5].

Security in the random oracle model does not imply security in the real world. The existence of verifiably committed signature is obvious in the standard complexity model provided the underlying signature schemes are provably secure in the standard complexity model as two signatures with keys (pk_1, sk_1), (pk_2, sk_2), and let $PK = (pk_1, pk_2)$, $SK = (sk_1, sk_2)$ and $\sigma = (\sigma_1, \sigma_2)$ are sufficient to build a secure verifiably committed signature.

The challenge problem is to construct a verifiably committed signature consistent with a stand-alone signature scheme in the standard complexity model [19, 20]. In this paper, we are able to provide a setup-driving verifiably committed signature based on the strong RSA assumption. The idea behind the construction is that given any valid partial signature of message m, if an arbitrator with

*Partial results have been published in PKC'04 [18].

its auxiliary input is able to generate variables called the resolution of message m such that the distribution of the variables is indistinguishable from those generated by the primary signer alone from the viewpoints of any verifier, then from which a verifiably committed signature can be derived.

2 Syntax and Security Definitions

The following definition of verifiably committed signatures is formalized the SAME thing as non-interactive fair exchanges introduced by Park, Chong, Siegel and Ray [15]. Therefore, the committed schemes presented in this report should be viewed as the actual fair exchange protocols working in the real world.

Syntax: A verifiably committed signature involves a primary singer Alice, a co-signer (or a verifier) Bob and an arbitrator (or TTP) Charlie, and is given by the following efficient procedures:

Key generator KG: This is an interactive protocol between a primary signer and an arbitrator, by the end of which either one of the parties aborts, or the primary signer learns her secret signing key SK, the arbitrator learns his secret key ASK, and both parties agree on the primary signer's public key PK and partial verification key APK;

Fully signing algorithm Sig and its correspondent verification algorithm Ver: These are conventional signing and verification algorithms. $Sig(m, SK)$ run by the primary signer, outputs a full signature σ on m, while $Ver(m, \sigma, PK)$ run by any verifier, outputs 1 (accept) or 0 (reject);

Partially signing algorithm $PSig$ and the correspondent verification algorithm $PVer$: 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 APK. $PSig(m, SK, PK, APK)$, run by the primary signer, outputs a partial signature σ', while $PVer(m, \sigma', PK, APK)$, run by any verifier, outputs 1 (accept) or 0 (reject);

Resolution algorithm Res: This is a resolution algorithm run by the arbitrator in case the primary singer refuses to open her signature σ to the verifier, who in turn possesses a valid partial signature σ' on m and a proof that he fulfilled his obligation to the primary signer. In this case, $Res(m, \sigma', ASK, PK)$ should output a valid full signature of m.

Correctness of verifiably committed signatures states that:

- $Ver(m, Sig(m, SK), PK) = 1$;
- $PVer(m, PSig(m, SK, PK, APK), PK, APK) = 1$;
- $Ver(m,\ Res(PSig(m,\ SK,\ PK,\ APK),\ ASK,\ APK,\ PK),\ PK) = 1$.

2.1 Security of Verifiably Committed Signatures

Recall that a verifiably committed signature is formalized the same thing as a non-interactive fair exchange. The security of verifiably committed signature scheme should consist of ensuring three aspects: security against a primary signer Alice, security against a verifier Bob, and security against a co-singer/abitrator Charlie. And refer the reader [10, 15, 19, 20] for further reference.

Security against malicious primary signer Alice: Intuitively, a primary signer Alice should not provide a partial signature which is valid both from the viewpoints of a co-signer and an arbitrator but which will not be opened into the primary signer's full signature by the honest arbitrator[1]. More formally, Let P be an oracle simulating the partial signing procedure $PSig$, and R be an oracle simulating the resolution procedure Res. Let k be system security parameter. We require that any probabilistic polynomial time Adv succeeds with at most negligible probability in the following experiment.

Experiment 1 (security against malicious primary signer Alice):

- Key generation: $(SK^*, PK, ASK, APK) \leftarrow KG^*(1^k)$, where KG^* denotes the run of key generator KG with the dishonest primary signer by the adversary, and SK^* denotes the adversary's states.
- Res oracle query: In this phase, for each adaptively chosen message m_j, the adversary computes its partial signature $\sigma_j{}'$ for m_j. Finally the adversary forward $\sigma_j{}'$ to the oracle R to obtain the full signature σ_j of message m_j, where $1 \leq j \leq p(k)$, and $p(\cdot)$ is a polynomial. At the end of R oracle query, the adversary produces a message and its full signature pair (m, σ), i.e., $(m, \sigma') \leftarrow Adv^R(SK^*, PK, APK)$, $\sigma \leftarrow Adv(m, \sigma', SK^*, APK, PK)$, where $m \neq m_j$, $1 \leq j \leq p(k)$.
- Success of Adv := $[PVer(m, \sigma', APK, PK) = 1 \wedge Ver(m, \sigma, PK) = 0]$.

Definition 1 *A verifiably verifiably committed signature is secure against malicious primary signer Alice, 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* $KG(\cdot)$*,* $PSig(\cdot)$ *and* $R(\cdot)$*.*

Security against malicious co-signer Bob: We consider the following scenario: suppose a primary signer Alice and a co-signer Bob are trying to exchange signature in a fair way. Alice wants to commit to the transaction by providing her partial signature. Of course, it should be computationally infeasible for Bob to compute the correspondent

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

full signature from any partial signature. More formally, we require that any probabilistic polynomial time adversary Adv succeeds with at most negligible probability in the following experiment:

Experiment 2 (security against malicious co-signer Bob):

- Key generation: $(SK, PK, ASK, APK) \leftarrow KG(1^k)$, where KG is run by the honest primary signer and honest arbitrator/TTP Charlie. Adversary Adv are admitted to make queries to the two orales P and R.
- P and R oracle query: For each adaptively chosen message m_j, the adversary obtains the partial signature $\sigma_j{}'$ of message m_j by querying the partial signing oracle P. Then the adversary forward $\sigma_j{}'$ to the resolution oracle R to obtain the full signature σ_j of message m_j, where $1 \leq j \leq p(k)$, and $p(\cdot)$ is a polynomial. At the end of oracle both P and R queries, the adversary produces a message-full signature pair $(m, \sigma) \leftarrow Adv^{P,R}(PK, APK)$.
- Success of adversary $Adv := [Ver(m, \sigma, PK) = 1 \wedge m \notin Query(Adv, R)]$, where $Query(Adv, R)$ is the set of valid queries the adversary Adv asked to the resolution oracle R, i.e., (m, σ') such that $PVer(m, \sigma') = 1$.

Definition 2 *A verifiably verifiably committed signature is secure against any malicious co-signer Bob, if any probabilistic polynomial time adversary Adv associated with partial signing oracle P and the resolution oracle R, succeeds with at most negligible probability, where the probability takes over coin tosses in $KG(\cdot)$, $P(\cdot)$ and $R(\cdot)$.*

Security against malicious arbitrator Charlie: Even though the arbitrator is semi-trusted, the primary signer does not want this arbitrator to produce a valid signature which the primary signer did not intend on producing. To achieve this goal, we require that any probabilistic polynomial time adversary Adv associated with partial signing oracle P, succeeds with at most negligible probability in the following experiment:

Experiment 3 (security against malicious arbitrator Charlie):

- Key generation: $(SK, PK, ASK^*, APK) \leftarrow KG^*(1^k)$, where $KG^*(1^k)$ is run by the dishonest cosigner or arbitrator. Adversary Adv are admitted to make queries to the partial signing oracle P.
- P oracle query: For each adaptively chosen message m_j, the adversary obtains the partial signature $\sigma_j{}'$ for m_j from the oracle P, where $1 \leq j \leq p(k)$, and $p(\cdot)$ is a polynomial. At the end of the partial partial signing oracle query, the adversary produces a message-full signature pair (m, σ), i.e., $(m, \sigma) \leftarrow Adv^P(ASK^*, PK, APK)$.
- Success of adversary $Adv := [Ver(m, \sigma, PK) = 1 \wedge m \notin Query(Adv, P)]$, where $Query(Adv, P)$ is the set of valid queries Adv asked to the partial oracle P, i.e., (m, σ') such that $PVer(m, \sigma') = 1$.

Definition 3 *A verifiably verifiably committed signature is secure against malicious arbitrator Charlie, 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 $KG(\cdot)$, $P(\cdot)$.*

Definition 4 *A verifiably verifiably committed signature is secure if it is secure against malicious primary signer Alice, malicious co-signer Bob and malicious arbitrator Charlie.*

3 Verifiably Committed Signatures from Strong RSA Assumption - Theoretical Considerations

3.1 The Description of Verifiably Committed Signature Scheme

We utilize Zhu's signature as primary building block to construct verifiably committed signature scheme [16]. We remark that the use of Zhu's signature is not essential. The Cramer-Shoup's signature including trapdoor hash signature [9], Camenisch and Lysyanskaya [7] and Fischlin's signature scheme [11] are all suitable for our purpose. However, among the signatures mentioned above, Zhu's signature is the most efficient (please refer to appendix 1 and appendix 2 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 $y^e = Xg^th^{H(m)} \bmod n$ 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 $X = y^eg^{-t}h^{-H(m)} \bmod n$. If the equation is valid, then the signature is valid. Otherwise, it is rejected.

Strong RSA assumption: Strong RSA assumption was introduced by Baric and Pfitzmann [6] and Fujisaki and Okamoto [12]: The strong RSA assumption is that it is hard, on input an RSA modulus n and an element $z \in Z_n^*$, to compute values $e > 1$ and y such that $y^e = z \bmod n$. More formally, we assume that for all polynomial time circuit families A_k, there exists a negligible function $\nu(k)$ such that:

$$Pr[n \leftarrow G(1^k), z \leftarrow Z_n^*, (e, y) \leftarrow A_k(n, z) : e > 1 \wedge y^e = z \mod n] = \nu(k)$$

The following lemma, due to Guillou-Quisquater [14], is useful to prove the security of the verifiably committed signature scheme.

Lemma 1 (Guillou-Quisquater Lemma) *Suppose* $w^e = z^b$ *and* $d = \gcd(e, b)$. *Then there exists an efficient algorithm computing the* (e/d)*-th root of* z.

Zhu's signature scheme is immune to adaptive chosen-message attack (in the sense of Goldwasser, Micali and Rivest [13]) under joint assumptions of the strong RSA problem as well as the existence of collision free hash function.

Based on Zhu's signature scheme, we are ready to describe the new verifiably committed signature below.

Key generation algorithm: We choose two safe primes $p = 2p' + 1$, $q = 2q' + 1$ and compute $N = pq$. Denote the quadratic residue of Z_N^* by QR_N. Let x, h_1, h_2 be elements chosen uniformly at random from the cyclic group QR_N. Let $PriG$ be a prime generator. On input 1^k, it generates $2s+1$ primes, each with bit length $(l+1)$. The prime pair $\{e_{i,1}, e_{i,2}\}$ is indexed by some $i \in I$ $(1 \le i \le s)$. The public key (X, g_1, g_2) is computed from x, h_1, h_2 and $(e_{1,2}, e_{2,2}, \cdots e_{s,2})$ as follows:

$$\begin{aligned} X &\leftarrow x^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N \\ g_1 &\leftarrow {h_1}^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N \\ g_2 &\leftarrow {h_2}^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N \end{aligned}$$

Denote a subset of index set in which each index i has been used to sign some message by I_{used}. We then build a public accessible prime list table $PriT$ as follows. On input $i \in I_{used}$, $PriT$ outputs $\{e_{i,1}, e_{i,2}\}$.

The primary signer's public key PK is $(N, X, g_1, g_2, H, PriT, I_{used})$.

The private key SK is $\{x, h_1, h_2, p, q, (e_{i,1}, e_{i,2}), 1 \le i \le s)\}$, where H is a publicly known collision-free hash function.

The APK of the arbitrator is $(N, X, g_1, g_2, H, PriT, I_{used})$.

The secret key of the co-signer ASK is $\{x, h_1, h_2, (e_{1,2}, e_{2,2}, \cdots, e_{s,2})\}$.

Partial signing algorithm $PSig$ and correspondent verification algorithm $PVer$: To sing a message m, we choose $i \in I \setminus I_{used}$ and a random string $t_{i,1} \in \{0,1\}^l$. The equation:

$$y_{i,1}^{e_{i,1}} = X g_1^{t_{i,1}} g_2^{H(m)} \bmod N.$$

is solved for $y_{i,1}$

We then update the index I_{used} by accumulating

$$I_{used} \leftarrow I_{used} \cup \{i\}$$

The partial signature of message m is $\sigma' = (i, e_{i,1}, t_{i,1}, y_{i,1})$.

On upon receiving a putative partial signature $\sigma' = (i, e_{i,1}, t_{i,1}, y_{i,1})$, the verification algorithm checks whether $i \in I_{used}$ or not, if $i \notin I_{used}$, then it outputs 0, otherwise, it runs $PriT$, on input i to obtain a prime pair $(e_{i,1}, e_{i,2})$, and it outputs 1, i.e., $PVer(m, \sigma') = 1$ if $\sigma'(m)$ satisfies the equation:

$$X = y_{i,1}^{e_{i,1}} g_1^{-t_{i,1}} g_2^{-H(m)} \bmod N$$

Full signing algorithm Sig and correspondent verification algorithm Ver: To fully sign the message m, for the given i, we obtain the prime pair $\{e_{i,1}, e_{i,2}\}$ by running $PriT$ on input $i \in I_{used}$. Then we choose a random string $t_{i,2} \in \{0,1\}^l$ uniformly at random and compute $y_{i,2}$ from the equation:

$$y_{i,2}^{e_{i,2}} = X g_1^{t_{i,2}} g_2^{H(t_{i,1}||m)} \bmod N$$

The corresponding full signature σ of the message m is defined below:

$$\sigma := (i, e_{i,1}, e_{i,2}, t_{i,1}, t_{i,2}, y_{i,1}, y_{i,2})$$

To verify the correctness of full signature scheme σ, the verification algorithm checks whether $i \in I_{used}$ or not, if $i \notin I_{used}$, then it outputs 0, otherwise, it runs $PriT$, on input i to obtain a prime pair $(e_{i,1}, e_{i,2})$. Finally it tests whether the following equations are valid:

$$\begin{aligned} X &= y_{i,1}^{e_{i,1}} g_1^{-t_{i,1}} g_2^{-H(m)} \bmod N \text{ and} \\ X &= y_{i,2}^{e_{i,2}} g_1^{-t_{i,2}} g_2^{-H(t_{i,1}||m)} \bmod N \end{aligned}$$

If both equations are valid, then the verification function outputs $Ver(m, \sigma) = 1$, otherwise, it outputs 0;

Resolution algorithm Res: Given a partial signature $\sigma' = (i, e_{i,1}, t_{i,1}, y_{i,1})$ of message m, the arbitrator runs the prime list table $PriT$ on input $i \in I_{used}$ to obtain the pair of primes $(e_{i,1}, e_{i,2})$, and checks whether $e_{i,1}$ is a component of partial signature σ' (such a prime $e_{i,1}$ is called a valid prime). If it is valid then the arbitrator checks the valid of the following equation:

$$y_{i,1}^{e_{i,1}} = X g_1^{t_{i,1}} g_2^{H(m)} \bmod N$$

If it is valid, the arbitrator then computes:

$$\begin{aligned} X_i &\leftarrow x^{e_{1,2}\cdots e_{i-1,2}e_{i+1,2}\cdots e_{s,2}} \\ g_{i,1} &\leftarrow {h_1}^{e_{1,2}\cdots e_{i-1,2}e_{i+1,2}\cdots e_{s,2}} \text{ and} \\ g_{i,2} &\leftarrow {h_2}^{e_{1,2}\cdots e_{i-1,2}e_{i+1,2}\cdots e_{s,2}}. \end{aligned}$$

Finally, the co-singer chooses a random string $t'_{i,2} \in \{0,1\}^l$ and computes $y_{i,2}$ from the following equation:

$$y_{i,2} = X_i {g_{i,1}}^{t'_{i,2}} {g_{i,2}}^{H(t_{i,1}||m)} \bmod N.$$

The output of the resolution algorithm is $(i, e_{i,1}, e_{i,2}, t_{i,1}, t'_{i,2}, y_{i,1}, y_{i,2})$. Obviously,

$$X = y_{i,2}^{e_{i,2}} g_1^{-t'_{i,2}} g_2^{-H(t_{i,1}||m)} \bmod N.$$

We remark that the choice of random string $t'_{i,2} \in \{0,1\}^l$ in the resolution phase does not dependent on the random string $t_{i,2}$ in the full signature algorithm. If we insist on the same string used in the resolution algorithm Res, then the random pair $(t_{i,1}, t_{i,2})$ can be listed as public known random string set which is also indexed by the set I.

We also remark that the number of signature is bounded by s, where $s(\cdot)$ is a polynomial of security parameter k. This is an interesting property as a primary signer can specify the number of signatures for each certificate during its validity duration.

3.2 The Proof of Security

In this subsection, we are able to prove that the main result stated below:

Theorem 1 *The verifiably committed signature is secure under the strong RSA assumption and the assumption that H is collision resistant in the standard complexity model.*

Proof: Security against the primary signer Alice is trivial since the arbitrator holds ASK in the protocol. **Q.E.D.**

Security against the verifier Bob: Assume that protocol is not secure against the verifier attack. That is, there is an adversary playing the role of verifier in the actually protocol, who is able to forge a full signature σ of a message m ($m \neq m_i$, $1 \leq i \leq f$) with non-negligible probability after it has queried partial signing oracle and resolution oracle of messages $m_1, \cdots, m_f$, each is chosen adaptively by the adversary. Let $(i, e_{i,1}, e_{i,2}, t_{i,1}, t'_{i,2}, y_{i,1}, y_{i,2})$ be the full signature provided by the partial signing oracle and the resolution oracle corresponding to a set of messages m_i ($1 \leq i \leq f$). We consider three types of forgeries as that in [9]:

1) for some $1 \leq j \leq f$, $e_k = e_{j,2}$ and $t'_{k,2} = t'_{j,2}$, where $k \notin \{1, \cdots, f\}$;

2) for some $1 \leq j \leq f$, $e_k = e_{j,2}$ and $t'_{k,2} \neq t'_{j,2}$, where $k \notin \{1, \cdots, f\}$;

3) for all $1 \leq j \leq f$, $e_k \neq e_{j,2}$, where $k \notin \{1, \cdots, f\}$.

We should show that any forgery scheme of the three types will lead to a contradiction to the assumptions of the theorem. This renders any forgery impossible.

By the security definition, the adversary can query the types of oracles: partial signing oracle and resolution oracle. Therefore we should describe the two oracles in the following simulation according to the forgery types defined above.

Type 1 forgery: On input (z, e), where $z \in Z_N^*$, e is a $(l+1)$-bit prime, we choose $(2f-1)$ primes $(e_{i,1}, e_{i,2})$ for $1 \leq i \neq j \leq f$, each with length $(l+1)$-bit. The j-th prime pair is defined by $(e_{j,1}, e)$. We compute PK and APK by choosing $z_1, z_2 \in Z_N^*$ uniformly at random and computing

$$\begin{aligned}
g_1 &\leftarrow {z_1}^{2e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}} z^B \\
B &= 2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}\cdots e_{f,1}e_{f,2} \\
g_2 &\leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}} \\
X &\leftarrow {z_2}^{2\beta e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}} z^C \\
C &= 2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots \\
&\quad e_{f,1}e_{f,2}(-\alpha)
\end{aligned}$$

where $\alpha \in \{0,1\}^{l+1}$ and $\beta \in Z_N$ are chosen uniformly at random.

Since the simulator knows each $e_{i,1}$ ($1 \leq i \leq f$), therefore it is easy to compute the partial signing oracle of message m_i ($1 \leq i \leq f$). And it is also easy to compute the resolution of i-th message $i \neq j$ queried to resolution oracle query Res. What we need to show is how to simulate the j-th resolution oracle query. This can be done as follows:

$$\begin{aligned}
{y_{j,2}}^{e_{j,2}} &= X{g_1}^{t'_{j,2}}{g_2}^{H(t_{j,1}||m_j)} \\
&= {z_2}^{2\beta\prod_{1,\cdots f}(e_{i,1}e_{i,2})}{z_1}^{2t'_{j,2}\prod_{1,\cdots f}(e_{i,1}e_{i,2})} \times z^B \\
B &= 2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots \\
&\quad e_{f,1}e_{f,2}(-\alpha + t'_{j,2} + H(t_{j,1}||m_j))
\end{aligned}$$

Now we set $-\alpha + t'_{j,2} + H(t_{j,1}||m_j) = 0$, i.e., $t'_{j,2} = \alpha - H(t_{j,1}||m_j)$. To show that the simulation is not trivial, we should show that $t'_{j,2}$ is uniformly distributed over $\{0,1\}^l$ with non-negligible amount. Since $\alpha \in \{0,1\}^{l+1}$ is chosen uniformly at random, the probability that $t'_{j,2}$ belongs to the correct interval and it does so with the correct uniform distribution can be computed as follows:

$$\frac{(2^{l+1} - 1 - H(t_{j,1}||m_j) - 2^l + 1) + H(t_{j,1}||m_j)}{(2^{l+1} - 1 - H(t_{j,1}||m_j)) - (-H(t_{j,1}||m_j)) + 1} = 1/2$$

Suppose the adversary is able to forge a faking signature $(k, e_{k,1}, e_{k,2}, t'_{k,1}, t'_{k,2}, y_{k,1}, y_{k,2})$ of message m_k, where $e_{k,2} = e_{j,2}$ and $t'_{k,2} = t'_{j,2}$, $k \notin \{1, \cdots, f\}$. We can not assume that $e_{k,2} = e_{j,2}$, $t'_{k,2} = t'_{j,2}$ and $y_{k,2} = y_{j,2}$ as H is a collision free hash function. Now we have two equations:

$$\begin{aligned}
{y_{j,2}}^{e_{j,2}} &= X{g_1}^{t'_{j,2}}{g_2}^{H(t_{j,1}||m_j)} \quad \text{and} \\
{y_{j,2}}^{e_{j,2}} &= X{g_1}^{t'_{j,2}}{g_2}^{H(t_{j,1}||m_j)}
\end{aligned}$$

It follows that

$$\begin{aligned}
(\frac{y_{j,2}}{y_{k,2}})^{e_{j,2}} &= {g_2}^{H(t_{j,1}||m_j) - H(t_{k,1}||m_k)} = z^B \\
B &= 2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots \\
&\quad e_{f,1}e_{f,2}(H(t_{j,1}||m_j) - H(t_{k,1}||m_k))
\end{aligned}$$

where $e_{j,2} = e$. Consequently, one is able to extract the e-th root of z with non-negligible probability. It contradicts the standard RSA assumption.

Type 2 forgery: On input z and e, where $z \in Z_N^*$, e is a $(l+1)$-bit prime, we choose $(2f-1)$ primes $(e_{i,1}, e_{i,2})$ for $1 \leq i \neq j \leq f$. The j-th prime pair is defined by

$(e_{j,1}, e)$. We compute PK and APK by choosing $z_1, z_2 \in Z_N^*$ uniformly at random and computing

$$\begin{aligned} g_1 &\leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}} \\ g_2 &\leftarrow {z_1}^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j,2}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}} \\ X &\leftarrow g_1^{-\alpha}{z_2}^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j,2}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}} \end{aligned}$$

where $z_1, z_2 \in Z_N$ and $\alpha \in \{0,1\}^l$ are chosen uniformly at random. Since QR_N is a cyclic group, we can assume that g_1, g_2 are generators of QR_N with overwhelming probability.

where $z_1, z_2 \in Z_N$ and $\alpha \in \{0,1\}^l$ are chosen uniformly at random. Since QR_N is a cyclic group, we can assume that g_1, g_2 are generators of QR_N with overwhelming probability.

Since $e_{i,1}$ for $1 \le i \le f$ are known therefore, the partial singing oracle is perfect from the point views of the adversary. To simulate the i-th message m_i $(i \ne j)$ to the resolution oracle, we select a random string $t'_{i,2} \in \{0,1\}^l$ and computes:

$$\begin{aligned} y_{i,2}^{e_{i,2}} &= X{g_1}^{t'_{i,2}}{g_2}^{H(t_{i,1}||m_i)} \\ &= (({z_1}^{H(t_{i,1}||m_i)}z_2)^B z^{2e_{i,1}(t'_{i,2}-\alpha)\prod_{s\ne i,j} e_{s,1}e_{s,2}})^{e_{i,2}} \\ B &= 2e_{1,1}e_{1,2}\cdots e_{i-1,1}e_{i-1,2}e_{i,1}e_{i+1,1}e_{i+1,2}\cdots e_{f,1}e_{f,2} \end{aligned}$$

The output of resolution oracle is $(i, e_{i,2}, y_{i,2}, t'_{i,2})$.

To sign the j-th message m_j, the signing oracle sets $t'_{j,2} \leftarrow \alpha$ and computes:

$${y_{j,2}}^{e_{j,2}} = (({z_1}^{H(t_{j,1}||m_i)}z_2)^{2e_{j,1}\prod_{s\ne j} e_{s,1}e_{s,2}})^{e_{j,2}}$$

where $e_{j,2} = e$.

Let $Res(m_k) = (k, e_{k,2}, y_{k,2}, t'_{k,2})$ be a legal signature generated by the adversary of message $m_k \ne m_i$ for all $1 \le i \le f$. By the assumption, we know that

$$\begin{aligned} {y_{k,2}}^{e_{k,2}} &= X{g_1}^{t'_{k,2}}{g_2}^{H(t'_{k,1}||m_k)} \text{ and} \\ {y_{j,2}}^{e_{j,2}} &= X{g_1}^{t'_{j,2}}{g_2}^{H(t'_{j,1}||m_j)}. \end{aligned}$$

Consequently, we have the following equation:

$$(\frac{y_{k,2}}{y_{j,2}})^{e_{j,2}} = {g_1}^{t'_{k,2}-t'_{j,2}}{g_2}^{H(t'_{k,1}||m_k)-H(t'_{j,1}||m_j)}$$

Equivalently,

$$\begin{aligned} & z^{2(\alpha-t'_{k,2})e_{j,1}\prod_{i\ne j} e_{i,1}e_{i,2}} \\ = \; & ({z_1}^{2e_{j,1}(H(t'_{j,1}||m_j)-H(t'_{k,1}||m_k))\prod_{i\ne j} e_{i,1}e_{i,2}})^{e_{j,2}} \end{aligned}$$

Since $t'_{j,2} = \alpha$ and $t_{k,2} \ne t'_{j,2}$, it follows that $\alpha - t'_{k,2} \ne 0$. We then apply Guillou-Quisquater lemma to extract the e-th root of z. This contradicts the standard RSA assumption.

Type 3 forgery: On input z, where $z \in Z_N^*$, we choose $2f$ primes $(e_{i,1}, e_{i,2})$ for $1 \le i \le f$ and compute the PK and ASK as follows:

$$\begin{aligned} g_1 &\leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}} \text{ and} \\ g_2 &\leftarrow g_1^a, X \leftarrow g_1^b \end{aligned}$$

where $a, b \in \{1, n^2\}$.

Since the simulator knows all prime pairs, it follows it can simulate both partial signing and resolution queries. Let $Res(m_k) = (k, e_{k,2}, y_{k,2}, t'_{k,2})$ be a legal signature generated by the adversary of message $m_k \ne m_i$ for all $1 \le i \le f$. It yields the equation

$${y_{k,2}}^{e_{k,2}} = Xg_1^{t'_{k,2}}{g_2}^{H(t_{k,1}||m_k)} = z^E$$

where $E = 2(b + t'_{k,2} + aH(t_{k,1}||m_k))e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}$

Since we are able to compute the $\frac{e}{E}$-th root of z provided e is a not a divisor of E according to the lemma of Guillou and Qusiquater [14], it is sufficient to show that e is not a divisor of E with non-negligible probability. Due to the the fact that $\gcd(e, e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}) = 1$, it is sufficient to show that e is not a divisor of $b+t+aH(t_{k,1}||m_k))$ with non-negligible probability. Since $b \in (1, n^2)$, it follows that one can write $b = b'p'q' + b''$. Therefore, the probability that $b + t + aH(m) \equiv 0 \bmod e$ is about $1/e$.

Security against the arbitrator Charlie: Even though the arbitrator is semi-trusted, the primary signer does not want this arbitrator to produce valid signature which the primary signer did not intend on producing. In other words, if the arbitrator is able to forge a partial signature of a message m, then we make use of Charlie as a subroutine to break the strong RSA assumption. Since Charlie holds the correspondent ASK, therefore we can assume that Charlie succeeds in forging a valid partial signature with non-negligible probability. The simulation is the same as the proof of Zhu's signature, therefore omitted.

4 Verifiably Committed Signature from Independent Signatures - Real World Applications

Although the existence of verifiably committed signature is obvious since two arbitrary signature schemes with keys (pk_1, sk_1), (pk_2, sk_2), and let $PK = (pk_1, pk_2)$, $SK = (sk_1, sk_2)$ and $\sigma = (\sigma_1, \sigma_2)$ is sufficient to build a secure verifiably committed signature even in the standard complexity model. If one is able to construct efficient yet secure verifiably committed signatures from two arbitrary schemes with considerable efficiency compared with the theoretically interested schemes, then the effect is also non-trivial from the point views of practical applications. Motivated by the real world considerations, we try to provide two types of verifiably committed signatures from a pair of independent ordinary signatures. We emphasize that the schemes presented in this section is interesting only from the point views of practical applications.

4.1 Verifiably Committed Signature from a Pair of Independent Zhu's Signatures

The first verifiably committed signature is from a pair of independent Zhu's signatures described as follows:

Key generation algorithm: $(N_1, N_2, X_1, X_2, g_1, h_1, g_2, h_2, H) \leftarrow Setup(1^k)$, where $N_i = p_i q_i$ and $p_i = 2p'_i + 1$, $q_i = 2q'_i + 1$, $i = 1, 2$. Let G_i be the quadratic residue of $Z^*_{N_i}$. Let g_1, h_1 be two generators of group G_1 and g_2, h_2 be two generators of G_2. Let $X_1 \in QR_{N_1}$ and $X_2 \in QR_{N_2}$ are two random chosen elements. Let H be a collision free hash function with output length l, eg, l= 160. The public key $PK = (N_1, N_2, X_1, X_2, g_1, h_1, g_2, h_2, H)$, $APK = (N_2, X_2, g_2, h_2, H)$. The secret key $SK = (p_1, q_1, p_2, q_2)$ and $ASK = (p_2, q_2)$.

Partial signing algorithm $PSig$ and correspondent verification algorithm $PVer$: To partially sign a message m, an $(l + 1)$-bit prime e and an l-bit string t are chosen uniformly at random. The equation

$$y_1^e = X_1 g_1^t h_1^{H(m)} \bmod N_1$$

is solved for y_1. The partial signature of message m is $\sigma' = (e, y_1, t)$. The partial verification algorithm outputs 1, i.e., $PVer(m, \sigma') = 1$ if $\sigma'(m)$ satisfies the equation:

$$y_1^e = X_1 g_1^t h_1^{H(m)} \bmod N_1.$$

Full signing algorithm Sig and correspondent verification algorithm Ver: To fully sign a message m, the primary signer solves the equation

$$y_2^e = X_2 g_2^t h_2^{H(m)} \bmod N_2$$

is solved for y_2. The corresponding full signature of the message m is $\sigma = (e, t, y_1, y_2)$. To verify the correctness of signature scheme, it tests whether the equations

$$\begin{aligned} y_1^e &= X_1 g_1^t h_1^{H(m)} \bmod N_1 \text{ and} \\ y_2^e &= X_2 g_2^t h_2^{H(m)} \bmod N_2. \end{aligned}$$

If both equations are valid, then the verification function outputs $Ver(m, \sigma) = 1$, otherwise, it outputs 0;

Resolution algorithm Res: Given a partial signature $\sigma' = (e, y_1, t)$ of message m, the arbitrator computes y_2 from the equation $y_2^e = X_2 g_2^t h_2^{H(m)} \bmod N_2$. The output of $Res(m, \sigma') = \sigma(m) := (e, t, y_1, y_2)$.

We remark that the modulus N_1 and N_2 are chosen independently except for the same bit-length ($|N_1| = |N_2| = 2k$, k is the system security parameter). The safe primes (p_1, q_1) are chosen by the primary signer while (p_2, q_2) are jointly chosen by the primary signer and the arbitrator, e.g., the arbitrator chooses (p_2, q_2) uniformly at random for a prime number set, signs-then-encrypts the prime numbers, and sends the cipher-text to the primary signer.

We remark that at the registration stage in a fair exchange system, a primary signer Alice has to prove to the certificate authority (CA) that N_1 and N_2 are products of safe primes without revealing (p_1, q_1) and (p_2, q_2). This can be done using zero-knowledge protocol of Camenisch and Michels [8]. After verifying the construction of N_1, N_2, the CA issues a certificate $Cert_{N_1,N_2}$.

We also remark that if e is an $(l + 1)$-bit prime chosen uniformly at random for partially sign a message m, which is co-prime with $\phi(N_1)$ then we can assume that gcd$(e, \phi(N_2)$=1 also duo to the fact that $N_i = p_i q_i$ and $p_i = 2p'_i + 1$, $q_i = 2q'_i + 1$, p_i, q_i, and p'_i and q'_i are primes with the same bit length for $i = 1, 2$.

There are two independent signatures used in our verifiably committed signature scheme nevertheless the protocol is efficient and is nontrivial as we can reuse the random string t and prime number e.

Similarly, it is not hard for one to prove the security against primary signer Alice, verifier Bob and arbitrator Charlie respectively under the strong RSA assumption and the assumption that H is collision resistant.

4.2 Verifiably Committed Signature from Cramer-Shoup Scheme Defined over QR_N

In this subsection, we define a Cramer-Shoup like (CS-like) trapdoor hash scheme in a quadratic residue as the trapdoor information allows the arbitrator to control the full signature as follows:

- Key generation algorithm: $(N_1, N_2, X_1, X_2, h, g_1, g_2, H) \leftarrow Setup(1^k)$, where $N_i = p_i q_i$ and $p_i = 2p'_i + 1$, $q_i = 2q'_i + 1$, $i = 1, 2$. Let p_1, p_2, q_1, q_2 be four large primes such that $p_i - 1 = 2p'_i$ and $q_i - 1 = 2q'_i$, where p'_i, q'_i are two l'-bit strings, $i = 1, 2$. Let $N_i = p_i q_i$ and QR_{N_i} be the quadratic residue of $Z^*_{N_i}$. Let X_1, h be two generators of QR_{N_1}. Let X_2, g_1, g_2 be three generators of QR_{N_2}. The public key is $(N_1, N_2, X_1, X_2, h, g_1, g_2, H)$. The private key is (p_1, q_1, p_2, q_2).

- Signature algorithm: To sign a message m, an $(l+1)$-bit prime e and a string $t \in \{0, 1\}^l$ is chosen uniformly at random. The equation:

$$y^e = X_1 h^{H(X_2 g_1^t g_2^{H(m)} \bmod N_2)} \bmod N_1$$

  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 first checks that e is an odd $(l+1)$-

bit number. Second it checks the validity of the equation:

$$X_1 = y^e h^{-H(X_2 g_1^t g_2^{H(m)} \bmod N_2)} \bmod N_1.$$

If the equation is valid, then the verifier accepts, otherwise, it rejects.

The proof of security is very similar to that of Cramer-Shoup's signature scheme [9], therefore omitted.

Based on the above CS-like signature scheme, we are now able to describe alternative verifiably committed signature scheme below:

Key generation algorithm: $(N_1, N_2, X_1, X_2, h, g_1, g_2, H) \leftarrow Setup(1^k)$, where $N_i = p_i q_i$ and $p_i = 2p_i' + 1$, $q_i = 2q_i' + 1$, $i = 1, 2$. Let p_1, p_2, q_1, q_2 be four large primes such that $p_i - 1 = 2p_i'$ and $q_i - 1 = 2q_i'$, where p_i', q_i' are two l'-bit strings, $i = 1, 2$. Let $N_i = p_i q_i$ and QR_{N_i} be the quadratic residue of $Z_{N_i}^*$. Let X_1, h be two generators of QR_{N_1}. Let X_2, g_1, g_2 be three generators of QR_{N_2}. The primary signer's public key PK is $(N_1, N_2, X_1, X_2, h, g_1, g_2, H)$, the private key SK is (p_1, q_1, p_2, q_2). The APK of the arbitrator is (N_2, X_2, g_1, g_2, H), and the secret key ASK is (p_2, q_2).

Partial signing algorithm $PSig$ and correspondent verification algorithm $PVer$: To partially sign a message m, a $(l+1)$-bit prime e and a l-bit string t are chosen at random. The equation:

$$y_1^e = X_1 h^{H(X_2 g_1^t g_2^{H(m)} \bmod N_2)} \bmod N_1$$

is solved for y_1. The partial signature of message m is $\sigma' = (e, t, y_1)$. The partial verification algorithm outputs 1, i.e., $PVer(m, \sigma') = 1$ if $\sigma'(m)$ satisfies the equation:

$$y_1^e = X_1 h^{H(X_2 g_1^t g_2^{H(m)} \bmod N_2)} \bmod N_1$$

Full signing algorithm Sig and correspondent verification algorithm Ver: The equation

$$y_2^e = X_2 g_1^t g_2^{H(m)} \bmod N_2$$

is solved for y_2. The corresponding full signature of the message m is $\sigma = (e, t, y_1, y_2)$. To verify the correctness of full signature scheme, it tests whether the equations

$$\begin{aligned} y_1^e &= X_1 h^{H(X_2 g_1^t g_2^{H(m)} \bmod N_2)} \bmod N_1 \text{ and} \\ y_2^e &= X_2 g_1^t g_2^{H(m)} \bmod N_2. \end{aligned}$$

If both equations are valid, then the verification function outputs $Ver(m, \sigma) = 1$, otherwise, it outputs 0;

Resolution algorithm Res: Given a partial signature $\sigma' = (e, t, y_1)$ of message m, the arbitrator computes y_2 from the equation

$$y_2^e = X_2 g_1^t g_2^{H(m)} \bmod N_2.$$

The output of $Res(m, \sigma') = \sigma(m) := (e, t, y_1, y_2)$.

The remarks on the the first verifiably committed signature scheme is also suitable for this scheme. And the proof of security of the second scheme is very similar with that of the first one, therefore omitted.

5 Conclusion

In this report, we provide a setup-driving verifiably committed signature from the strong RSA assumption based on Zhu's signature scheme. As the verifiably committed signature formalized the same thing as the fair exchange protocol, our scheme is actually a fair exchange protocol with provably secure.

Acknowledgements

The research (project number 60273058) is supported by National Natural Science Foundation of China.

References

[1] N. Asokan, M. Schunter, and M. Waidner, "Optimistic protocols for fair exchange," in *ACM Conference on Computer and Communications Security*, pp. 7–17, 1997.

[2] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures (extended abstract)," in *EUROCRYPT*, pp. 591–606, 1998.

[3] G. Ateniese, "Efficient verifiable encryption (and fair exchange) of digital signatures," in *6th ACM Conference on Computer and Communications Security (ACM CCS'99)*, pp. 138–146.

[4] F. Bao, R. Deng, and W. Mao, "Efficient and practical fair exchange protocols," in *Proceedings of 1998 IEEE Symposium on Security and Privacy*, Oakland, pp. 77–85, 1998.

[5] A. Boldyreva, "Threshold signatures, multisigntaures and blind signatures based on the Gap Diffie Helman group signature scheme," in *PKC 2003*, LNCS 2567, pp. 31–46, 2003.

[6] N. Braic and B. Pfitzmann, "Collision free accumulators and fail-stop signature scheme without trees," in *Eurocrypt'97*, pp. 480–494, 1997.

[7] J. Camenisch, and A. Lysyanskaya, "A signature scheme with efficient protocols," *SCN 2002*, pp.268–289, 2002.

[8] J. Camenisch, "Markus michels: proving in zero-knowledge that a number is the product of two safe primes," in *EUROCRYPT*, pp. 107–122, 1999.

[9] R. Cramer and V. Shoup, "Signature scheme based on the Strong RAS assumption," in *6th ACM Conference on Computer and Communication Security*, Singapore, ACM Press, November 1999.

[10] Y. Dodis, and L. Reyzin, "Breaking and repairing optimistic fair exchange from PODC 2003", in *ACM Workshop on Digital Rights Management (DRM)*, October 2003.

[11] M. Fischlin, "The Cramer-Shoup strong-RSA signature scheme revisited," in *Public Key Cryptography*, pp. 116–129, 2003.

[12] E. Fujisaki, and T. Okamoto, "Statistical zero-knowledge protocols to prove modular polynomial relations," in *Crypto'97*, LNCS 1294, pp. 16–30, Springer-verlag, 1997.

[13] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.

[14] L. Guillou, and J.Q uisquater, "A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory," in *Eurocrypto'88*, pp. 123–128, 1988.

[15] J. M. Park, E. Chong, H. Siegel, and I. Ray, "Constructing fair-exchange protocols for E-commerce via distributed computation of RSA signatures," in *PODC 2003*, pp. 172–181.

[16] H. Zhu, "New digital signature scheme attaining immunity to adaptive chosen-message attack," *Chinese Journal of Electronics*, vol. 10, no. 4, pp. 484–486, Oct, 2001.

[17] H. Zhu, *A formal proof of Zhu's signature scheme*, available at eprint.iacr.org.

[18] H. Zhu, "Constructing verifiably committed signatures from strong-RSA assumption in the standard complexity model," in *Public Key Cryptography*, pp. 101–114, 2004.

[19] H. Zhu, *Constructing optimistic fair exchange protocols from committed signatures*, available at eprint.iacr.org.

[20] H. Zhu, *Verifiably committed signatures provably secure in the standard complexity model*, available at eprint.iacr.org.

Appendix 1: Cramer-Shoup's Signature Scheme and Its Variations

Cramer-Shoup's trapdoor hash scheme: Cramer and Shoup presented an elegant signature scheme called trapdoor hash function defined below (see [9] for more details):

- Key generation algorithm: Let p, q be two safe primes ($p - 1 = 2p'$ and $q - 1 = 2q'$, where p', q' are two primes) with length l'. Let $n = pq$ and QR_n be the quadratic residue of Z_n^*. Let x, h be two generators of QR_n. Also chosen are a group G of order s, where s is an $(l+1)$-bit prime, and two random generators g_1, g_2 of G. The public key is (n, h, x, g_1, g_2, H) along with an appropriate description of G including s. The private key is (p, q).

- Signature algorithm: To sign a message m, an $(l+1)$-bit prime e and a string $t \in Z_s$ is chosen uniformly at random. The are chosen at random. The equation $y^e = xh^{H(g_1^t g_2^{H(m)})} \bmod n$ 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 first checks that e is an odd $(l+1)$-bit number. Second it checks the validation that $x = y^e h^{-H(g_1^t g_2^{H(m)})} \bmod n$. If the equation is valid, then the verifier accepts, otherwise, it rejects.

Zhu's signature scheme: In Cramer-Shoup's scheme, another extra group G is defined. From the point views of computational complexity it is non-trivial work if one can reduce the computational and communication complexity while its provability and efficiency can be maintained. Based on this observation, Zhu provides a variation scheme below [16]:

- Key generation algorithm: Let p, q be two large safe primes such that $p - 1 = 2p'$ 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 be three generators of QR_n. 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 $y^e = Xg^t h^{H(m)} \bmod n$ 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 first checks that e is an odd $(l+1)$-bit number. Second it checks the validation that $X = y^e g^{-t} h^{-H(m)} \bmod n$. If the equation is valid, then the verifier accepts, otherwise, it rejects.

Camenisch-Lysyanskaya's signature scheme: In SCN'02, Camenisch and Lysyanskaya [7] presented alternative signature scheme. The Camenisch and Lysyanskaya signature is described as follows (see [7] for more details).

- Key generation algorithm: On input 1^k, choose a special RSA modulus $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$ of length $l_n = 2k$. Choose, uniformly at random, $a, b, c \in QR_n$. Output $PK = (n, a, b, c)$, and $SK = p$.

- Message space. Let l_m be a parameter. The message space consist of all binary string of length l_m. Equivalently, it can be thought of as consisting of integers in the range $[0, 2^{l_m})$.

- Signing algorithm: On input m, choose a random prime number $e > 2^{l_m+1}$ of length $l_e = l_m + 2$, and a random number s of length $l_s = l_n + l_m + l$, where l is a security parameter. Compute the value v such that
$$v^e = ca^m b^s \bmod n$$

- Verification algorithm: To verify that the tuple (e, s, v) is a signature on message m in the message space, check that $v^e = ca^m b^s \bmod n$ and check that $2^{l_e} > e > 2^{l_2 - 1}$.

Fischlin's signature scheme: Later a similar modification is presented in PKC'03 by Marc Fischlin. Fischlin's signature scheme is defined as follows [11]:

- Key generation: Generating $n = pq$, where $p = 2p' + 1$ and $q = 2q' + 1$ for primes p, q, p', q'. Also pick three quadratic residue $h_1, h_2, x \in QR_n$. The public key verification key is (n, h_1, h_2, x) and the private key is (p, q).

- Signing: To sign a message m calculate the l-bit hash value $H(m)$ with a collision-intractable hash function $H(\cdot)$. Pick a random $(l+1)$-bit prime e, and a random l-bit string α and compute a representation $(-\alpha, -(\alpha \oplus H(m)), y)$ of x with respect to h_1, h_2, e, n, i.e.,
$$y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \bmod n.$$
Computing this e-th root y from $x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)}$ is easy given the factorization of n. The signature is (e, α, y).

- Verification algorithm: On upon receiving a triple (e, α, y), one checks that e is an odd $(l+1)$-bit integer and α is l bits long string, finally it checks the validity of the equation $y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \bmod n$. It is valid, then it output "ACCEPT", otherwise, it outputs "REJECT"

The relationship between Zhu's signature and Camenisch-Lysyanskaya's signature scheme is obvious. Here we remark the relationship between Zhu's signature schemes and Fischlins's scheme therefore.

- It is clear that the algebraic structures of Zhu's and Fischlin's signature are same;

- If there is no collision hash function involved in the above two schemes, then it is not hard to show that the above two signature schemes are equivalent in the same security level. More precisely, if Zhu's scheme can be broken by an adversary A with non-negligible probability then there exists an adversary B^A so that Fischlin's signature scheme can be broken with the same probability. The statement is also true by means of vis-a-vis argument.

- In case of a collision free hash function involved in both schemes, suppose Zhu's signature scheme can be broken with non-negligible probability, i.e., there is an adversary A is able to forge a faking message m in Zhu's signature scheme, denoted by $\sigma(m) = (e, y, t)$ with non-negligible probability. Then there exists an adversary B^A in Fischlin's signature scheme so that it is able to produce a valid signature $\sigma(m') = (e, y, t)$ for any message in the set $S := \{m' | H(m) \oplus H(m') = t\}$, where t is a component of faking signature $\sigma(m)$ correspondent to Zhu's signature scheme. The statement is also true by means of vis-a-vis argument.

Appendix 2: Security Proof of Zhu's Signature Scheme (A Refined Version of the Previous Work [16, 17])

Main result: Zhu's signature scheme is immune to adaptive chosen-message attack under the strong RSA assumption and the assumption that H is a collision resistant.

Proof: Assume that the signature scheme is NOT secure against adaptive chosen message attack. That is, there is an adversary, who is able to forge the signature (e, t, y) of a message $m (m \neq m_i,\ 1 \leq i \leq f)$ with non-negligible probability after it has queried correspondent signature of each message $m_1, \cdots, m_f$, which is chosen adaptively by the adversary. Let $(e_1, t_1, y_1), \cdots, (e_f, t_f, y_f)$ be signatures provided by the signing oracle corresponding to a set of messages $m_1, \cdots, m_f$. We consider three types of forgeries: 1) for some $1 \leq j \leq f$, $e = e_j$ and $t = t_j$; 2) for some $1 \leq j \leq f$, $e = e_j$ and $t \neq t_j$; 3) for all $1 \leq j \leq f$, $e \neq e_j$. We should show that any forgery scheme of the three types will lead to a contradiction to the assumptions of the theorem. This renders any forgery impossible.

Type 1 - Forger:

We consider an adversary who chooses a forgery signature such that $e = e_j$ for a fixed j: $1 \leq j \leq f$, where f is the total number of the queries to the signing oracle. If the adversary succeeds in a signature forgery as type1 with non-negligible probability then given n, we are able to compute $z^{1/r}$ with non-negligible probability, where r is a $(l+1)$-bit prime. This contradicts to the assumed hardness of the standard RSA problem. We state the attack in details as follows: given $z \in Z_n^*$ and r, we choose a set of total $f - 1$ primes with length $(l+1)$-bit $e_1, ...e_{j-1}, e_{j+1}, ..., e_f$ uniformly at random. We then create the correspondent public key (X, g, h) of the simulator as follows: given $z \in Z_n^*$ and r, we choose a set of total $f - 1$ primes with length $(l+1)$-bit $e_1, ...e_{j-1}, e_{j+1}, ..., e_f$ uniformly at random. We choose $w, v \in Z_n$ uniformly at random, and compute $h = z^{2e_1 \ldots e_{j-1} e_{j+1} \ldots e_f}$, $g = v^{2e_1 \cdots e_f} z^{2e_1 \ldots e_{j-1} e_{j+1} \ldots e_f}$ and $X = w^{2\beta e_1 \cdots e_f} z^{2e_1 \ldots e_{j-1} e_{j+1} \ldots e_f (-\alpha)}$, where $\alpha \in \{0,1\}^{l+1}$ and $\beta \in Z_n$ are chosen uniformly at random.

Since the simulator knows each e_i, therefore it is easy to compute the i-th signing query. What we need to show is how to simulate the j-th signing query. This can be done as follows:

$$\begin{aligned} y_j^{e_j} &= X g^{t_j} h^{H(m_j)} \\ &= (w^{\beta} v^{t_j})^{2e_1 \cdots e_f} z^{2e_1 \ldots e_{j-1} e_{j+1} \ldots e_f (-\alpha + t_j + H(m_j))} \end{aligned}$$

Now we set $-\alpha + t_j + H(m_j) = 0$, i.e, $t_j = \alpha - H(m_j)$.

To show the simulation above is non-trivial, we should show t_i is uniformly distributed over $\{0,1\}^l$ with non-negligible amount. Since $\alpha \in \{0,1\}^{l+1}$ is chosen uniformly at random, i.e., $0 \leq \alpha \leq 2^{l+1} - 1$, the probability t_j belongs to the correct interval and it does so with the correct uniform distribution can be computed as follows:

$$\frac{(2^{l+1}-1-H(m_j)-2^l+1)+H(m_j)}{(2^{l+1}-1-H(m_j))-(-H(m_j))+1}=1/2$$

Suppose the adversary is able to forge a faking signature of message m, denoted by (e, y, t), such that $e_j = e(= r)$, $t_j = t$. Notice that one can not assume that $e_j = e$, $t_j = t$ and $y_j = y$, since H is a collision free hash function. Now we have two equations: $y_j^e = Xg^th^{H(m_j)}$ and $y^e = Xg^th^{H(m)}$. Consequently, we obtain the equation:

$$(\frac{y_j}{y})^e = h^{H(m_j)-H(m)} = z^{2e_1,\ldots e_{j-1},e_{j+1},\ldots,e_f(H(m_j)-H(m))}$$

It follows that one can extract the e-th root of z with non-negligible probability. Therefore, we arrive at the contradiction of the standard hardness of RSA assumption.

Type 2 - Forger:
We consider an adversary who succeed in forging a valid signature such that $e = e_j$, $t \neq e_j$ for a fixed j: $1 \leq j \leq f$, where f is the total number of the queries to the signing oracle. If the adversary succeeds in a signature forgery as type1 with non-negligible probability then given n, we are able to compute $z^{1/r}$ with non-negligible probability for a given z and r, where r is a $(l + 1)$-bit prime. This contradicts to the assumed hardness of the standard RSA problem. We state the attack in details as follows: given $z \in Z_n^*$ and r, we choose a set of total $f - 1$ primes with length $(l + 1)$-bit $e_1, ...e_{j-1}, e_{j+1}, ..., e_f$ at random. We then create the correspondent public key (X, g, h) of the simulated signature scheme as follows: $g = z^{2e_1\ldots e_{j-1}e_{j+1}\ldots e_f}$, $h = v^{2e_1\ldots e_f}$ and $X = g^{-\alpha}w^{2e_1\ldots e_f}$, where $w, v \in Z_n$ and α is a l-bit random string. Since QR_n is a cyclic group, we can assume that g, h are generators of QR_n with overwhelming probability. To sign the i-th message $m_i(i \neq j)$, the signing oracle selects a random string $t_i \in \{0, 1\}^l$, and computes:

$${y_i}^{e_i} = ((wv^{H(m_i)})^{2e_1\ldots e_{i-1}e_{i+1}\ldots e_f} z^{2(t_i-\alpha)\Pi_{s\neq i,s\neq j}e_s})^{e_i}$$

The output of the signing oracle is a signature of message m_i, denoted by $\sigma(m_i) = (e_i, y_i, t_i)$.

To sign the j-th message m_j, the signing oracle, sets $t_j \leftarrow \alpha$ and computes:

$${y_j}^{e_j} = ((wv^{H(m_j)})^{2\Pi_{s\neq j}e_s})^{e_j}$$

The output of the signing oracle is a signature of message m_j, denoted by $\sigma(m_j) = (e_j, y_j, t_j)$.

Let $\sigma(m) = (e, y, t)$ be a valid signature forged by the adversary of message m. By assumption, we know that $y^e = Xg^th^{H(m)}$. Consequently, we have the following equation:

$$g^{t_j}h^{H(m_j)}{y_j}^{e_j} = g^th^{H(m)}y^e$$

Equivalently

$$z^{2(\alpha-t)\Pi_{i\neq j}e_i} = (v^{2(H(m)-H(m_j))\Pi_{i\neq j}e_i}\frac{y}{y_j})^{e_j}$$

Since $t_j = \alpha$ and $t \neq t_j$ by assumption, it follows that $t \neq \alpha$. We then apply Guillou-Quisquater lemma to extract the r-th root of z, where $r = e_j$.

Type 3 - Forger:
We consider the third type of the attack: the adversary forgery is that for all $1 \leq j \leq f$, $e \neq e_j$. If the adversary succeeds in forgery with non-negligible probability, then given n, a random $z \in Z_n^*$, we are able to compute $z^{1/d}$ $(d > 1$) with non-negligible probability, which contradicts to the assumed hardness of strong RSA assumption. We state our attack in details as follows: we generate g and h with the help of z. We define $g = z^{2e_1\cdots e_f}$ and $h = g^a$, where $a \in (1, n^2)$, is a random element. We can assume that g is a generator of QR_n with overwhelming probability. Finally, we define $X = g^b$, where $b \in (1, n^2)$. Since the simulator knows the all e_j, the signature oracle can be perfectly simulated. Let (e, t, y) be a forgery signature of message m. It yields the equation $y^e = Xg^th^{H(m)} = z^E$, where $E = (b + t + aH(m))2e_1...e_f$.

Since we are able to compute (e/E)-th root of z provided e is a not a divisor of E according to the lemma of Guillou and Qusiquater, it is sufficient to show that e is not a divisor of E with non-negligible probability. Due to the the fact that $\gcd(e, e_1e_2 \cdots e_f) = 1$, it is sufficient to show that e is not a divisor of $b + t + aH(m)$ with non-negligible probability. Since $b \in (1, n^2)$, it follows that one can write $b = b'p'q' + b''$. Therefore, the probability that $b + t + aH(m) \equiv 0 \bmod e$ is about $1/e$.

Remark on Type 3- Forger: To show that $e|(b + t + aH(m)$ with negligible probability, one may make use of randomness of $a \in (1, n^2)$. That is one can write a as $a = a'p'q' + a''$. It follows a' is a random element from the adversary's view. Hence the probability that $b + t + aH(m) \equiv 0 \bmod e$ is about $1/e$. Thus, with non-negligible probability, e is not a divisor of $b + t + aH(m)$. We point out that since the adversary may find $H(m) = 0$, the term $aH(m)$ may be cancelled in the formula in the equation. Thus the random argument must be done in term b instead of $aH(m)$ since collision-resistance does not imply zero-finder intractability in general. This remark also suitable for Cramer-Shoup's argument.

Huafei Zhu got his PhD degree at Institute of Information Security, University of Electronics Science and Technology (XIDIAN University, Supervised by Professor Guozhen Xiao) at May 1997. Now he is associate professor at zhejiang university (YuQuan Campus) mainly interested in cryptology and network security.