Constructing Committed Signatures From Strong-RSA Assumption In The Standard Complexity Model

Huafei Zhu

Department of Infocomm Security, Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 19613
E-mail: huafei@i2r.a-star.edu.sg

Abstract. In this paper, we provide the first committed signature provably secure in the standard complexity model based on the strong RSA assumption. The idea behind the construction is that given any valid partial signature of message m, if a co-signer with 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 point views of the verifier/arbitrator, then from which a committed signature can be derived.

Keywords: Committed signatures, fair exchange protocols, strong RSA assumption

1 Introduction

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 co-signer store a piece of prime signer's secret key (please refer to [1], [4], [2], [3] 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 committed signatures later. 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 committed signature scheme since the mechanism of the non-interactive fair exchange is the same thing as a committed signature. Unfortunately this 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]. Therefore Dodis and Reyzin's committed signature is the first committed signature provably secure under the Gap Diffie-Hellman assumption in the random oracle paradigm.

Security in the random oracle model does not imply security in the real world. The existence of 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 committed signature. Therefore the challenge problem is to construct a committed signature consistent with a stand-alone signature scheme in the standard complexity model. In this paper, we are able to provide the first committed signature based on the strong RSA assumption. The idea behind the construction is that given any valid partial signature of message m, if a co-signer with 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 point views of the verifier/arbitrator, then from which a committed signature can be derived.

The rest of paper is organized as follows: in Section 2, we formalize the security definition of committed signatures, and a committed signature is fully described in the Subsection 3.1, the proof of its security is presented in Subsection 3.2. In Section 4, we construct committed signatures from the point views of real world by providing two efficient schemes with random strings reusing. Finally the conclusion is presented in Section 5.

2 Notions and definitions

The following definition of committed signatures is formalized the SAME thing as non-interactive fair exchanges introduced by Park, Chong, Siegel and Ray [15] and [10]. Therefore, the committed schemes presented in this report should be viewed as the actual fair exchange protocols working in the real world.

Definition 1 A committed signature involves a primary signer Alice, a verifier Bob and a co-signer (or arbitrator) Charlie, and is given by the following efficient procedures:

-Key generator KG: This is an interactive protocol between a primary signer and a co-signer, by the end of which either one of the parties aborts, or the primary signer learns her secret signing key SK, the co-signer 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 co-signer (arbitrator) in case the primary signer 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 committed signatures states that: (1) $Ver(m, Sig(m, SK), PK)$=1; (2)$PVer(m, PSig(m, SK, PK, APK), PK, APK)$=1;$and$(3)Ver(m, $Res(PSig(m,$ SK, PK, APK), ASK, APK, PK), PK)=1.

2.1 Security of committed signatures

Recall that a committed signature is formalized the same thing as a non-interactive fair exchange. The security of 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-signer/abitrator Charlie.

Security against a primary signer Intuitively, a primary signer Alice should not provide a partial signature which is valid both from the point views of a verifier and a co-signer but which will not be opened into the primary signer's full signature by the honest co-signer. 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 primary signer):

1.1: 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.

1.2: 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 $m \neq m_j$, $1 \leq j \leq p(k)$.

1.3. Success of $Adv := [PVer(m, \sigma', APK, PK) = 1 \wedge Ver(m, \sigma, PK) = 0]$.

Definition 2 A committed signature scheme is secure against primary signer attack, 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 verifier We consider the following scenario: suppose a primary signer Alice and a verifier 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 full signature from the 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 verifier):

2.1 Key generation: $(SK, PK, ASK, APK) \leftarrow KG(1^k)$, where KG is run by the honest primary signer and honest co-signer. Adversary Adv are admitted to make queries to the two orales P and R.

2.2 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 $(m, \sigma) \leftarrow Adv^{P,R}(PK, APK)$.

2.3 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 3 A committed signature scheme is secure against verifier attack, 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 co-signer/arbitrator This property is crucial. Even though the co-signer (arbitrator) is semi-trusted, the primary signer does not want this co-signer 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 co-signer/arbitrator):

3.1 Key generation: $(SK, PK, ASK^*, APK) \leftarrow KG^*(1^k)$, where $KG^*(1^k)$ is run by the dishonest co-signer or arbitrator. Adversary Adv are admitted to make queries to the partial signing orale P.

3.2 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., $(m, \sigma) \leftarrow Adv^P(ASK^*, PK, APK)$.

3.3 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 4 A committed signature scheme is secure against co-signer attack, 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 5 A committed signature scheme is secure if it is secure against primary signer attack, verifier attack and co-signer attack.

3 Constructing committed signatures from strong RSA assumption

3.1 Our committed signature scheme

We utilize Zhu's signature as primary building block to construct 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 purposes. Nevertheless, among the signatures mentioned above, Zhu's signature is the most efficient.

Zhu's signature scheme Zhu's signature scheme is defined as follows [16]:

- 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 $y^e = Xg^t h^{H(m)} \bmod n$ 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 $X = y^e g^{-t} h^{-H(m)} \bmod n$. 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:

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

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

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. Please refer to the appendix for details. Based on Zhu's signature scheme, we are ready to describe the new 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 $i \in I$ $(1 \leq i \leq s)$. 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:

$$X \leftarrow x^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N$$

$$g_1 \leftarrow {h_1}^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N$$

$$g_2 \leftarrow {h_2}^{e_{1,2}e_{2,2}\cdots e_{s-1,2}e_{s,2}} \bmod N$$

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

The APK of the co-signer 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

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

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:

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

and

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

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 co-signer 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 co-signer 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 co-signer then computes:

$$X_i \leftarrow x^{e_{1,2} \cdots e_{i-1,2} e_{i+1,2} \cdots e_{s,2}}$$

$$g_{i,1} \leftarrow {h_1}^{e_{1,2} \cdots e_{i-1,2} e_{i+1,2} \cdots e_{s,2}}$$

and

$$g_{i,2} \leftarrow {h_2}^{e_{1,2} \cdots e_{i-1,2} e_{i+1,2} \cdots e_{s,2}}$$

Finally, the co-signer 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 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.

-We also remark that the scheme requires both the signer and co-signer to be stateful to keep count $i \in I_{used}$ and so never reuse primes. And the used index set I_{used} updated after each signature generation is apparently assumed to be accessible to the verifier and co-signer.

3.2 The proof of security

Theorem 6: The 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 co-signer holds ASK in the protocol.

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 $1 \leq j \leq f$, $e_{k,2} = e_{j,2}$ and $t'_{k,2} = t'_{j,2}$, where $k \notin \{1, \cdots, f\}$; 2) for some $1 \leq j \leq f$, $e_{k,2} = e_{j,2}$ and $t'_{k,2} \neq t'_{j,2}$, where $k \notin \{1, \cdots, f\}$; 3) for all $1 \leq j \leq f$, $e_{k,2} \neq e_{j,2}$, 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

$$g_1 \leftarrow {z_1}^{2e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}} z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}}$$

$$g_2 \leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}}$$

$$X \leftarrow {z_2}^{2\beta e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}} z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}(-\alpha)}$$

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

Since the simulator knows each $e_{i,1}$ ($1 \leq i \leq f$), 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:

$${y_{j,2}}^{e_{j,2}} = X {g_1}^{t'_{j,2}} {g_2}^{H(t_{j,1}||m_j)}$$

$$= {z_2}^{2\beta \prod_{1,\cdots f}(e_{i,1}e_{i,2})} {z_1}^{2t'_{j,2}\prod_{1,\cdots f}(e_{i,1}e_{i,2})} \times$$

$$z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}(-\alpha+t'_{j,2}+H(t_{j,1}||m_j))}$$

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 of message m_k, denoted by $(k, e_{k,1}, e_{k,2}, t'_{k,1}, t'_{k,2}, y_{k,1}, y_{k,2})$, 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:

$${y_{k,2}}^{e_{k,2}} = X {g_1}^{t'_{k,2}} {g_2}^{H(t_{k,1}||m_k)}$$

And

$${y_{j,2}}^{e_{j,2}} = X {g_1}^{t'_{j,2}} {g_2}^{H(t_{j,1}||m_j)}$$

It follows that

$$(\frac{y_{j,2}}{y_{k,2}})^{e_{j,2}} = {g_2}^{H(t_{j,1}||m_j)-H(t_{k,1}||m_k)}$$

$$= z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}(H(t_{j,1}||m_j)-H(t_{k,1}||m_k))}$$

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 $1 \le i \ne j \le f$. The j-th prime pair is defined by $(e_{j,1}, e)$. We compute PK and APK by choosign $z_1, z_2 \in Z_N^*$ uniformly at random and computing

$$g_1 \leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}}$$

$$g_2 \leftarrow {z_1}^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j,2}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}}$$

$$X \leftarrow g_1^{-\alpha} {z_2}^{2e_{1,1}e_{1,2}\cdots e_{j-1,1}e_{j-1,2}e_{j,1}e_{j,2}e_{j+1,1}e_{j+1,2}\cdots e_{f,1}e_{f,2}}$$

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 signing 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:

$$y_{i,2}^{e_{i,2}} = X {g_1}^{t'_{i,2}} {g_2}^{H(t_{i,1}||m_i)}$$

$= ((z_1{}^{H(t_{i,1}||m_i)} z_2)^{2e_{1,1}e_{1,2}\cdots e_{i-1,1}e_{i-1,2}e_{i,1}e_{i+1,1}e_{i+1,2}\cdots e_{f,1}e_{f,2}} z^{2e_{i,1}(t'_{i,2}-\alpha)\prod_{s\neq i,j} e_{s,1}e_{s,2}})^{e_{i,2}}$

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:

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

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 $m_k \neq m_i$ for all $1 \leq i \leq f$. By the assumption, we know that

$$y_{k,2}{}^{e_{k,2}} = X g_1{}^{t'_{k,2}} g_2{}^{H(t'_{k,1}||m_k)}$$

and

$$y_{j,2}{}^{e_{j,2}} = X g_1{}^{t'_{j,2}} g_2{}^{H(t'_{j,1}||m_j)}$$

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,

$$z^{2(\alpha-t'_{k,2})e_{j,1}\prod_{i\neq j} e_{i,1}e_{i,2}} = (z_1{}^{2e_{j,1}(H(t'_{j,1}||m_j)-H(t'_{k,1}||m_k))\prod_{i\neq j} e_{i,1}e_{i,2}})^{e_{j,2}}$$

Since $t'_{j,2} = \alpha$ and $t_{k,2} \neq t'_{j,2}$, it follows that $\alpha - t'_{k,2} \neq 0$. 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 $1 \leq i \leq f$ and compute the PK and ASK as follows:

$$g_1 \leftarrow z^{2e_{1,1}e_{1,2}\cdots e_{f,1}e_{f,2}}$$

and

$$g_2 \leftarrow g_1^a, X \leftarrow g_1^b$$

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 $m_k \neq m_i$ for all $1 \leq i \leq f$. It yields the equation

$$y_{k,2}{}^{e_{k,2}} = X g_1^{t'_{k,2}} g_2{}^{H(t_{k,1}||m_k)} = z^E$$

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 $b + t + aH(m) \equiv 0 \mathrm{mod} e$ is about $1/e$.

Security against the co-signer/arbitrator Charlie: Even though the co-signer (arbitrator) is semi-trusted, the primary signer does not want this co-signer to produce valid signature which the primary signer did not intend on producing. In other words, if the co-signer 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 Bob holds the correspondent ASK, therefore we can assume that Bob 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 Conclusion

In this report, we provide the first committed signature from the strong RSA assumption based on Zhu's signature scheme. As the committed signature formalized the same thing as the fair exchange protocol, our scheme is actually a fair exchange protocol which is provably secure in the standard complexity model. We should admit that the scheme does not quite achieve the consistency with Zhu's signature scheme with a stand-alone signature fully. How to construct a compactly specified one is our further research.

References

1. G. Ateniese, Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures. In 6th ACM Conference on Computer and Communications Security (ACM CCS'99), 138-146.
2. N. Asokan, M. Schunter, M. Waidner: Optimistic Protocols for Fair Exchange. ACM Conference on Computer and Communications Security 1997: 7-17.
3. N. Asokan, V. Shoup, M. Waidner: Optimistic Fair Exchange of Digital Signatures (Extended Abstract). EUROCRYPT 1998: 591-606.
4. F. Bao, R. Deng, W. Mao, Efficient and Practical Fair Exchange Protocols, Proceedings of 1998 IEEE Symposium on Security and Privacy, Oakland, pp. 77-85, 1998.
5. A. Boldyreva. Efficient threshold signatures, multisigntaures and blind signatures based on the Gap Diffie Helman group signature scheme. PKC 2003, LNCS 2567.
6. N. Braic and B. Pfitzmann. Collision free accumulators and fail-stop signature scheme without trees. Eurocrypt'97, 480-494, 1997.
7. J.Camenisch, A. Lysyanskaya. A Signature Scheme with Efficient Protocols. SCN 2002: 268-289.
8. Jan Camenisch, Markus Michels: Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes. EUROCRYPT 1999:107-122
9. R. Cramer and V. Shoup. Signature scheme based on the Strong RAS assumption. 6th ACM Conference on Computer and Communication Security, Signapore, ACM Press, November 1999.

10. Y.Dodis, L. Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003, ACM Workshop on Digital Rights Management (DRM), October 2003.
11. Marc Fischlin: The Cramer-Shoup Strong-RSASignature Scheme Revisited. Public Key Cryptography, 2003: 116-129.
12. E. Fujisaki, T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial relations. Crypto'97, LNCS 1294, Springer-verlag, 1997.
13. S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2): 281-308, 1988.
14. L. Guillou, J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory. Eurocrypto'88, 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, PODC 2003, 172-181.
16. Huafei 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.

Appendix: A Formal Proof of Zhu's Signature Scheme

Claim: 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 $h = z^{2e_1...e_{j-1}e_{j+1}...e_f}$, $g = v^{2e_1 \cdots e_f} z^{2e_1...e_{j-1}e_{j+1}...e_f}$ and $X = w^{2\beta e_1 \cdots e_f} z^{2e_1...e_{j-1}e_{j+1}...e_f(-\alpha)}$, 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:

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

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., $0 \le \alpha \le 2^{l+1} - 1$, 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: $y_j^e = X g^t h^{H(m_j)}$ and $y^e = X g^t h^{H(m)}$. Consequently, we obtain the equation:

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

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: $1 \le j \le f$, 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: $g = z^{2e_1 \cdots e_{j-1}e_{j+1} \cdots e_f}$, $h = v^{2e_1...e_f}$ and $X = g^{-\alpha} w^{2e_1...e_f}$, 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:

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

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 $y^e = Xg^t h^{H(m)}$. Consequently, we have the following equation:

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

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 $g = z^{2e_1 \ldots e_f}$ 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 $y^e = Xg^t h^{H(m)} = z^E$, where $E = (b + t + aH(m))2e_1 \ldots e_f$. 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 $\gcd(e, e_1 e_2 \cdots e_f) = 1$, 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 $b+t+aH(m) \equiv 0 mod e$ is about $1/e$.