Verifiably Committed Signatures Provably Secure in The Standard Complexity Model

Huafei Zhu

Department of Information Science and Electronics Engineering, ZheJiang University,
YuQuan Campus, HangZhou, 310027, PR. China
E-mail: zhuhf@zju.edu.cn

Abstract. In this paper, we study the security notions of verifiably committed signatures by introducing privacy and cut-off time, and then we propose the first scheme which is 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 messages, if a co-signer with its auxiliary input is able to generate variables called the resolution of messages such that the distribution of the variables is indistinguishable from that generated by the primary signer alone from the views of the verifier/arbitrator, a verifiably committed signature can be constructed.

Keywords: Verifiably committed signatures, fair exchange protocols, strong RSA assumption

1 Introduction

An important issue in electronic commerce is how to exchange electronic data between two potentially distributed parties in an efficient and fair manner. Intuitively, fairness allows two parties to exchange items in a way so that either each party gets other's item, or neither party does. Examples of such exchanges include signing of electronic contracts, certificated e-mail delivery and fair purchase of electronic goods over communication network. In such instances, ensuring fairness is crucial if the participants are to be protected from fraud.

Related works The problem of fair exchange has a rich history due to its fundamental importance. In the following, we only briefly mention the body of research most relevant to our results, and refer the reader to [4], [25] and [34] and [1] for further references. Early work on fair exchange of secrets/signatures, focused on the gradual release of secrets to obtain simultaneity and fairness [7], [23], [33] and [20]. The idea is that if each party alternately release a small portion of the secret, then neither party has a considerable advantage over the other. Unfortunately, such a solution has several drawbacks. Apart from being expensive in terms of computation and communication, it has the problem in real situations of uncertain termination.

Alternative approach to achieve fairness makes use of a trusted third party (TTP). A TTP is essentially a judge that can be called in to handle disputes between the participants. The TTP can be on-line in the sense of mediating after every exchange as in [18] and [24], or off-line, meaning that it only gets involved when something goes wrong (e.g., a participant attempts to cheat, or simply crashes, or the communication delays between the participants are intolerably high, etc.). The latter approach has been called optimistic [2].

Fair exchange protocols using verifiable encryption was proposed by Atenies [1] and Bao et al. [6]. These protocols apply ad-hoc techniques to create the fairness primitive via a specific encryption scheme that confirms to a given signature type. Unfortunately, the schemes proposed in [1] and [5] lack any formal security analysis, and consequently, one of the schemes proposed in [6] was shown to be insecure in [10] and [1]. In [3] and [4], Asokan et al. propose an optimistic that uses a cryptographic primitive denoted as verifiably encrypted signatures to produce the fairness. In such schemes, Alice encrypts her signature under TTP's encryption key and proves to Bob that she indeed encrypted her valid signature. After receiving her item from Bob, she proceeds to open the encryption. This approach of [3] and [2] was later generalized by Cachin and Camenisch [12] and Camenisch Damgård [13] but all these schemes involve expensive and highly interactive zero-knowledge proof in the exchange phase.

In PODC 2003, Park, Chong, Siegel and Ray [34] provided alternative 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. Based on Park et.al's study, Dodis and Reyzin [22] 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 [22]. Dodis and Reyzin [22] then presented a remedy scheme by utilizing Boldyreva's non-interactive two-party multi-signature scheme [9]. Therefore Dodis and Reyzin's scheme is the first verifiably 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 verifiably committed signatures provably secure in the standard complexity model are obvious provided the underlying signature schemes are provably secure in the standard complexity model as two signatures with independent keys (pk_1, sk_1), (pk_2, sk_2) are sufficient to build a secure verifiably committed signature if we define $PK = (pk_1, pk_2)$, $SK = (sk_1, sk_2)$ and $\sigma = (\sigma_1, \sigma_2)$. Hence the challenge problem is to construct a verifiably committed signature consistent with a stand-alone signature scheme in the standard complexity model.

If we insist on two important items (1) the privacy of exchange message, and (2) the cut-off time of exchange message, integrated with fair-exchange protocols as that have been studied by Asokan et al [4] and Micali [31], then verifiably committed signatures can be classified as follows:

- (1) Verifiably committed signatures without privacy and cut-off time;
- (2) Verifiably committed signatures with privacy but without cut-off time;
- (3) Verifiably committed signatures without privacy but with cut-off time;
- (4) And verifiably committed signatures with privacy and cut-off time.

We remark that Park, Chong, Siegel and Ray [34], and Dodis and Reyzin [22] do not embed two items into their schemes. Recent works of Micali [31], provides a unified model to deal with cut-off time. In Micali's model, Alice chooses cut-off time t and sends $Sig_A(t, m)$ to Bob, where t represents the time after which co-signer or the arbitrator should not help to completing the transaction any more. Thus before executing the response message generation algorithm, Bob should decide whether he has enough time to get the co-signer or arbitrator help if necessary, taking into consideration any possible time discrepancies between his own watch and that of the co-signer or arbitrator. We refer readers to [31] for further references.

Our contributions In this paper, we study the security notions of verifiably committed signatures by introducing privacy and cut-off time, and then we propose the first scheme which is 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 messages, if a co-signer with its auxiliary input is able to generate variables called the resolution of messages such that the distribution of the variables is indistinguishable from that generated by the primary signer alone from the views of the verifier/arbitrator, a verifiably committed signature can be constructed.

The rest of paper is organized as follows: in Section 2, we study the security definitions of verifiably committed signatures by introducing privacy and cut-off time; A verifiably committed signature without privacy and cut-off time is fully described in the Subsection 3.1, from which a verifiably committed signature with privacy and cut-off time can be easily derived. The proof of its security is presented in Subsection 3.2. Finally, the conclusion is presented in section 4.

2 Four notions of verifiably committed signatures

Continuing the works of Dodis and Reyzin [22], we further consider four notions of verifiably committed signatures by introducing privacy and cut-off time.

2.1 Verifiably committed signatures without privacy

Definition 2.1.1 A verifiably committed signature without privacy and cut-off time, involves a primary singer Alice, a verifier Bob and a co-singer (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;

-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);

-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);

-Resolution algorithm Res: This is a resolution algorithm run by the co-singer (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:

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

Verifiably committed signatures with cut-off time If we replace the message m by $(t||m)$, in the above protocol, where t represents the time after which cosigner or the arbitrator should not help to completing the transaction any more, then the correspondent schemes are called verifiably committed signatures with cut-off time. We emphasize that before executing the response message generation algorithm, Bob should decide whether he has enough time to get the co-signer or arbitrator help if necessary, taking into consideration any possible time discrepancies between his own watch and that of the co-signer or arbitrator. We point out here that the function of cut-off time t involved in the partial signature and full signature schemes is to specify the duration of validation of signatures, and it does not affect the security definition of verifiably committed signatures.

Security of verifiably committed signature schemes 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.

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 2.1.2 (security against primary signer):

2.1.2.1: 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.

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

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

Definition 2.1.3 A verifiably 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 2.1.4 (security against verifier):

2.1.4.1 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.

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

2.1.4.3 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 2.1.5 A verifiably 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 2.1.6 (security against co-signer/arbitrator):

2.1.6.1 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.

2.1.6.2 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 signing oracle query, the adversary produces a message-full signature pair (m, σ), i.e., $(m, \sigma) \leftarrow Adv^P(ASK^*, PK, APK)$.

2.1.6.3 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 2.1.7 A verifiably 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 2.1.8 A verifiably committed signature scheme is secure if it is secure against primary signer attack, verifier attack and co-signer attack.

The correspondent fair-exchange protocol A fair-exchange protocol without privacy and cut-off time can be derived by adding response message generation algorithm to the correspondent verifiably committed signature as follows.

-Alice runs the partially signing algorithm $PSig$ on input message m, and sends the partial signature σ' to Bob;

-Bob runs the response message generation algorithm Rmg as follows: If Bob receives the properly signed partial signature σ' of message m, he generates a proper message $Rmg(m)$ related to the message m, and digitally signs it and sends $Sig_B(Rmg(m))$ to Alice (otherwise, the response message generation algorithm outputs a null string indicating termination of the protocol);

-If Alice receives a properly signed message $Sig_B(Rmg(m))$, she runs $Sig(m, SK)$ and the output of the algorithm $Sig(m, SK)$ is a full signature σ on m (otherwise, the response message generation algorithm outputs a null string indicating termination of the protocol);

-If the primary singer refuses to open her signature σ to the verifier, Bob sends the record $(\sigma', Sig_B(Rmg(m))$ to the arbitrator together with a proof that he fulfilled his obligation to the primary signer. The co-signer then runs the resolution algorithm on input σ' which outputs a valid full signature σ of m and then sends σ to Bob.

2.2 Verifiably committed signatures with privacy

We now present a formal definition on verifiably committed signatures with privacy but without cut-off time in which no one else including co-signer/ arbitrator may learn partial information of message which is sent to Bob by Alice. Essentially, we properly replace message m by $E_B(m)$, where by $E_B(m)$, we denote the encryption of a message m with the public key of Bob.

Definition 2.2.1 A verifiably committed signature with privacy but without cut-off time, involves a primary singer Alice, a verifier Bob and a co-singer (or arbitrator) Charlie, and is given by the following efficient procedures:

-Key generator KG: The algorithm KG 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;

-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. $PSig(E_B(m), SK, PK, APK)$, run by the primary signer, outputs a partial signature σ', while $PVer(E_B(m), \sigma', PK, APK)$, run by any verifier, outputs 1 (accept) or 0 (reject);

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

-Resolution algorithm Res: This is a resolution algorithm run by the co-singer (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 $E_B(m)$ and a proof that he fulfilled his obligation to the primary signer. In this case, $Res(E_B(m), \sigma', ASK, PK)$ should output a valid full signature of $E_B(m)$.

Correctness of committed signatures states that:

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

Verifiably committed signatures with privacy and cut-off time If we replace the cipher-text $E_B(m)$ by $t||E_B(m)$ in the above protocol, where t represents the time after which cosigner or the arbitrator should not help to completing the transaction any more, then the correspondent schemes are called verifiably committed signatures with privacy and cut-off time. Again we point out here that the function of cut-off time t involved in the partial signature and full signature schemes is to specify the duration of validity of signatures in the protocol, and it does not affect the security definition of verifiably committed signatures indeed.

Security of verifiably committed signature schemes with privacy The security of verifiably committed signature scheme with privacy 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.

Security against a primary signer Experiment 2.2.2 (security against primary signer):

2.2.2.1 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.

2.2.2.2 *Res* oracle query: In this phase, for each adaptively chosen message m_j (we view m_j as the cipher-text of a plain-text message m_j under the encryption of public key of Bob E_B, i.e., $m_j \leftarrow E_B(m_j)$ since the security definition against primary signer could ignore the privacy), 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 m 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 $m \neq m_j$, $1 \leq j \leq p(k)$.

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

Definition 2.2.3 A verifiably 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 Experiment 2.2.4 (security against verifier):

2.2.4.1 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..2.4.2 P and R oracle query: For each adaptively chosen message m_j (since the plain-text m_j is encrypted with Bob's public key by Alice, therefore both

Alice and Bob know the transmitted message m_j, consequently we can assume that m_j is a cipher-text of a message m_j or simply assume that it is plain-text an adaptively chosen 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 $(m, \sigma) \leftarrow Adv^{P,R}(PK, APK)$.

2.2.4.3 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 2.2.5 A verifiably 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 Experiment 2.2.6 (security against co-signer/arbitrator):

2.2.6.1 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 and decryption oracle queries to D_B, where by $D_B(m)$, we denote the decryption algorithm with secret key of Bob;

2.2.6.2A 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 $m_j \leftarrow E_B(m_j)$, $1 \leq j \leq p(k)$, and $p(\cdot)$ is a polynomial. At the end of the 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 $Adv(Sig) := [Ver(m, \sigma, PK) = 1 \wedge m \notin Query(Adv, P)]$, where $Query(Adv, P)$ is the set of valid queries Adv asked to the partial oracle P, i.e., (m, σ') such that $PVer(m, \sigma') = 1$;

2.2.6.2B D_B oracle query (security against privacy in the sense of Rackoff and Simons definition [36]): First, the encryption scheme's key generation algorithm is run, with a security parameter as input. Next the adversary makes arbitrary queries to the decryption oracle D_B, decrypting the cipher-texts of it choice. Next the adversary chooses two message m_0, m_1, and sends these to the encryption oracle E_B. The encryption oracle chooses a bit $b \in \{0, 1\}$, at random and encrypts m_b. The correspondent cipher-text is given to the adversary (the internal coin tosses of the encryption oracle, in particular b, are not in the adversary's view). After receiving the cipher-text from the decryption oracle, the adversary continues to query the decryption oracle, subject only to the restriction that the query must be different than the output of the encryption oracle. At the end of game, the adversary outputs $b' \in \{0, 1\}$, which is supposed to be the adversary's guess of the value b. If the probability that $b' = b$ is $1/2 + \epsilon$, the adversary's advantage $Adv(Enc) := \epsilon$;

Definition 2.2.7 A verifiably committed signature scheme is secure against co-signer attack, if any probabilistic polynomial time adversary Adv associated with partial signing oracle P and decryption oracle D_B, succeeds with at most negligible probability of both $Adv(Sig)$ and $Adv(Enc)$, where the probability takes over coin tosses in $KG(\cdot)$, $KG_B(\cdot)$, $P(\cdot)$.

Definition 2.2.8 A verifiably committed signature scheme is secure if it is secure against primary signer attack, verifier attack and co-signer attack.

The correspondent fair-exchange protocol A fair-exchange protocol with privacy and cut-off time can be derived by adding response message generation algorithm to the correspondent verifiably committed signature as follows.

-Alice runs the partially signing algorithm $PSig$ on input message $t||E_B(m)$, and sends the partial signature σ' to Bob, where E_B is Cramer-Shoup's encryption algorithm which is provably secure under assumption of hardness of decisional Diffie-Hellman problem in the standard complexity model [17];

-Bob runs the response message generation algorithm Rmg as follows: If Bob receives the properly signed partial signature σ' of message $t||E_B(m)$, he generates a proper message $Rmg(m)$ related to the message m, and digitally signs it and sends $Sig_B(E_A(Rmg(m))$ to Alice (otherwise, the response message generation algorithm outputs a null string indicating termination of the protocol);

-If Alice receives a properly signed message $Sig_B(E_A(Rmg(m))$, she runs $Sig(t||E_B(m), SK)$ and the output of the algorithm $Sig(t||E_B(m), SK)$ is a full signature σ on $t||E_B(m)$ (otherwise, the response message generation algorithm outputs a null string indicating termination of the protocol);

-If the primary singer refuses to open her signature σ to the verifier, Bob sends the record $(\sigma', Sig_B(E_A(Rmg(m)))$ to the arbitrator together with a proof that he fulfilled his obligation to the primary signer. The co-signer then runs the resolution algorithm on input σ' which outputs a valid full signature σ of $t||E_B(m)$ and then sends σ to Bob.

3 Constructing verifiably committed signatures from strong RSA assumption in the standard complexity model

In this section, we provide the first verifiably committed signatures from strong RSA assumption in the standard complexity model. The core technique is that by specifying a proper index set, we can convert partial signature of a message to the full signature of the message. The technique is not new and it has been used by Rabin [35], Blum and Zhu [8], Deng, Lee and Zhu [21], and Camenish and Koprowski [19], for constructing secure protocols.

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 $2s$ 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:

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

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

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

By I_{used}, we denote a subset of index set in which each index i has been used to sign some message m_i. 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 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:

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

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:

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

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:

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

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:

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

and

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

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:

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

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

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

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,

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

-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.

3.1 The proof of security

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

Theorem 3.1: 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 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 [16]:

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

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

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

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

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:

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

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 $1 \leq i \neq j \leq f$. 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

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

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

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

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 $1 \leq i \leq f$ are known therefore, the partial singing oracle is perfect from the point views of the adversary. To simulate the i-th message m_i $(i \neq j)$ 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

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

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 [32], 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 $b + t + aH(m) \equiv 0 \text{mod} e$ 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, please refer to the appendix for details.

We have presented a basic verifiably committed signatures from strong RSA assumption without privacy and cut-off time. However, one can easily transform the above scheme to verifiably committed signatures with privacy and cut-off time under the same assumption if one replaces m by $t||E_B(m)$. The proof is straight forward and is very similar to the above argument, therefore omitted.

4 Conclusion

In this report, we provide the first 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.

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. N. Asokan, Victor Shoup, Michael Waidner: Optimistic Fair Exchange of Digital Signatures. IEEE JOurnal on Selected Areas in Communications, Vol 18, No.4, 2000, 593-610.
5. F. Bao. An efficient verifiable encryption scheme for encryption of discrete logarithms. CARDIS'98, 213-220.
6. 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.
7. M. Blum. Three Applications of the Oblivious Transfer: Part I: Coin Flipping by the telephone; Part II: How to Exchange Secrets; Part III: How to Send Certified Electronic Mail. Manuscript. University of California, Berkeley, 1981. (Also described in Cryptographers gather to discuss Research. By G. Kolata. Science, Vol. 214, November 6, 1981.)
8. M. Blum, H.Zhu. Deniable authentication schemes. Manuscript, in honour of Professor Manuel Blum's 60th Birthday April 20-24, 1998, City University of Hong Kong.
9. A. Boldyreva. Efficient threshold signatures, multisigntaures and blind signatures based on the Gap Diffie Helman group signature scheme. PKC 2003, LNCS 2567.
10. C. Boyd, E. Foo: Off-Line Fair Payment Protocols Using Convertible Signatures. ASIACRYPT 1998: 271-285
11. N. Braic and B. Pfitzmann. Collision free accumulators and fail-stop signature scheme without trees. Eurocrypt'97, 480-494, 1997.
12. C. Cachin, J. Camenisch: Optimistic Fair Secure Computation. CRYPTO 2000: 93-111.
13. Jan Camenisch, Ivan Damgård: Verifiable Encryption, Group Encryption, and Their Applications to Separable Group Signatures and Signature Sharing Schemes. ASIACRYPT 2000: 331-345.
14. J.Camenisch, A. Lysyanskaya. A Signature Scheme with Efficient Protocols. SCN 2002: 268-289.
15. Jan Camenisch, Markus Michels: Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes. EUROCRYPT 1999:107-122
16. R. Cramer and V. Shoup. Signature scheme based on the Strong RAS assumption. 6th ACM Conference on Computer and Communication Security, Singapore, ACM Press, November 1999.
17. R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In Crypto '98, LNCS 1462, pages 13-25, Springer-Verlag, Berlin, 1998.
18. B.Cox, J.D.Tygar and M.Sirbu, NetBill Security and Transaction Protocol. In first USENIX workshop on Electronic Commerce, 1995, 77-88.
19. Camenish, M. Koprowski. Fine-grained forward-secure signature scheme without random oracle. Workshop on coding and cryptography 2003 (France), 71-80.
20. Ivan Damgård: Practical and Provably Secure Release of a Secret and Exchange of Signatures. Journal of Cryptology 8(4): 1995, 201-222.
21. X. Deng, C.H. Lee and Zhu. Denaible authentication protocols. Computers and Digital Techniques, IEE Proceedings. Volume 148, Issue 2, 2001, page 101-104.
22. Y.Dodis, L. Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003, ACM Workshop on Digital Rights Management (DRM), October 2003.

23. S. Even, O. Goldreich and A. Lempel. A Randomized Protocol for Signing Contracts. Tech. Rep. 233, CS Dept., Technion, Haifa, February 1982. Subsequent version: Communications of the ACM, 28(6) pp. 637–647, 1985.
24. M. K. Franklin and M. Reiter: Fair exchange with a semi-trusted third party. IN ACM Security,1-5.
25. J.A.Garay, M. Jakobsson, P.D.MacKenzie: Abuse-Free Optimistic Contract Signing. CRYPTO, 1999: 449-466.
26. O. Goldreich: A Simple Protocol for Signing Contracts. CRYPTO 1983: 133-136
27. E. Fujisaki, T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial relations. Crypto'97, LNCS 1294, Springer-verlag, 1997.
28. Marc Fischlin: The Cramer-Shoup Strong-RSASignature Scheme Revisited. Public Key Cryptography, 2003: 116-129.
29. E. Fujisaki, T. Okamoto. Statistical zero-knowledge protocols to prove modular polynomial relations. Crypto'97, LNCS 1294, Springer-verlag, 1997.
30. S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2): 281-308, 1988.
31. S. Micali: Simple and fast optimistic protocols for fair electronic exchange. PODC 2003: 12-19.
32. L. Guillou, J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessors minimizing both transmission and memory. Eurocrypto'88, 123-128, 1988.
33. M. Ben-Or, M. Goldreich, S. Micali and R. Rivest. A Fair Protocol for Signing Contracts. IEEE Transactions on Information Theory 36/1 (1990) 40-46.
34. 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.
35. M.Rabin. Efficient Deniable Authentication of Long Messages,International conference on theoretical computer science,in honour of Professor Manuel Blum's 60th Birthday April 20-24, 1998, City University of Hong Kong.
36. C.Rackoff,D.Simon. Non-interactive zero-knowledge proof of knowledge and chosen cipher-text attacks. Cryptology-Crypto'91. 433-444, Springer-Verlag, 1992.
37. H. Zhu. New Digital Signature Scheme Attaining Immunity to Adaptive Chosen-message attack. Chinese Journal of Electronics, Vol.10, No.4, Page 484-486, Oct, 2001.

Appendix: A variation of Cramer-Shoup's signature scheme

Cramer-Shoup's trapdoor hash scheme Cramer and Shoup presented an elegant signature scheme called trapdoor hash function defined below (see [16] 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 [37]:

- 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 [14] presented alternative signature scheme. The Camenisch and Lysyanskaya signature is described as follows (see [14] 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 $v^e = ca^m b^s \mathrm{mod} n$ 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 [28]:

- 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.,
$$y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \mathrm{mod} n.$$
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 $y^e = x{h_1}^{\alpha}{h_2}^{\alpha \oplus H(m)} \mathrm{mod} n$. It is valid, then it output "ACCEPT", otherwise, it outputs "REJECT"

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-Forgery: 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:

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

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: $y_j^e = X g^t h^{H(m_j)}$ and $y^e = X g^t h^{H(m)}$. Consequently, we obtain the equation:

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

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-Forgery: 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 $y^e = Xg^t h^{H(m)}$. 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-Forgery: 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 $y^e = Xg^t h^{H(m)} = z^E$, 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 $b + t + aH(m) \equiv 0 \mathrm{mod} e$ is about $1/e$.

Remark on Type 3- Forgery: 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 $b + t + aH(m) \equiv 0 \mathrm{mod} e$ 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.