Efficient and Optimistic Fair Exchanges Based on Standard RSA with Provable Security

ZhenFeng ZHANG, YongBin ZHOU, DengGuo FENG

***Abstract*— In this paper, we introduce a new and natural paradigm for fair exchange protocols, called verifiable probabilistic signature scheme. A security model with precise and formal definitions is presented, and an RSA-based efficient and provably secure verifiable probabilistic signature scheme is proposed. Our scheme works well with standard RSA signature schemes, and the proposed optimistic fair exchange protocol is much concise and efficient, and suitable for practical applications.**

***Index Terms*— Probabilistic signature, RSA, Fair exchange, Provable security**

I. Introduction

With the growth of open networks such as Internet, the problem of fair exchanges has become one of the fundamental problems in secure electronic transactions and digital rights management. Payment systems, contract signing, electronic commerce and certified e-mail are classical examples in which fairness is a relevant security property. Informally, an exchange protocol allows two distributed parties to exchange electronic data in an efficient and fair manner, and it is said to be fair if it ensures that during the exchange of items, no party involved in the protocol can gain a significant advantage over the other party, even if the protocol is halted for any reason.

Protocols for fair exchange have attracted much attention in the cryptographic community in the past few years. The proposed methods mainly include: simultaneous secret exchange, gradual secret releasing, fair exchange using an on-line TTP and fair exchange with an off-line TTP. Among these results, optimistic fair exchange protocols based on an off-line trusted third party [1], [4] are preferable as they offer a more cost-effective use of a trusted third party. An optimistic fair exchange protocol usually involves three parties: users Alice and Bob, as well as an off-line TTP. The off-line TTP does not participate the actual exchange protocol in normal cases, and is invoked only in abnormal cases to dispute the arguments between Alice and Bob to ensure fairness.

Asokan et al. [1] were the first to formally study the problem of optimistic fair exchanges. They present several provably secure but highly interactive solutions, based on the concept of *verifiable encryption of signatures*. Their approach was later generalized by [9], but all these schemes involve expensive and highly interactive zero-knowledge proofs in the exchange phase. Other less formal works on interactive verifiably encrypted signatures include [4], [2]. Ateniese [2] proposed six schemes for fair exchanges, while two of which were shown to be vulnerable to colluding attacks [3]. The first and only non-interactive verifiably encrypted signature scheme was recently constructed by Boneh et al. [7]. While very elegant and provably secure in the random oracle model, the scheme requires special elliptic curve groups with a bilinear map and relies on a form of the computational Diffie-Hellman assumption for such groups.

As for cryptographic engineering practices, it is desirable to propose an efficient fair exchange scheme based on RSA, the most widely used public key cryptosystem. However, it is nontrivial to adopt the existing off-line-TTP ideas to RSA with acceptable efficiency. One of Ateniese's schemes [2] is based on RSA signatures, in which TTP must generate public keys for each participant and then shore secret values per capita, and proofs of equality of two discrete logarithms are used to ensure verifiable encryption. A CEMBS based verifiably encrypted RSA signatures was proposed in [18], which works with non-standard RSA groups and is also less efficient. A simple fair exchange protocol based on mediated-RSA was presented in [17], which relies on the recently proposed identity-based mediated-RSA [14] and no formal proofs provided. Recently, Park etc. [15] proposed an optimistic protocol for fair exchange based on RSA signatures, using a technique of "two-signatures". However, Park's scheme was soon shown to be totally breakable in the registration phase by [13]. Moreover, Dodis and Reyzin [13] proposed a new primitive called *verifiably committed signatures* for constructing fair exchange protocols, and presented a committed signature scheme based on GDH signatures [8]. However, it seems that their method [13] does not work for RSA signatures.

The full domain hash (FDH) signature scheme is popular and provably secure "hash-and-sign" signatures based on trap-door permutations such as RSA. Classically, results of this sort of provable security are asymptotic, and say little about the security of a scheme in practice for a particular choice of key size, as emphasized by Bellare and Rogaway [6]. Thus, for practical considerations it is critical to focus on concrete security reductions. The probabilistic signature scheme (PSS) designed by Bellare and Rogaway [5] is a probabilistic variant of FDH which introduces a random salt to achieve a tight security reduction to, e.g., the RSA problem. The general technique of using a random salt to achieve a tight(er) security reduction has been studied extensively [6], [12], [16].

Motivated by the approaches of verifiably encrypted signatures and verifiably committed signatures, we introduce a new paradigm for fair exchanges, called *verifiable probabilistic signature schemes*, in which the exchanged items are some

The authors are with State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100080, P. R. China. (E-mail: {zfzhang, zyb, feng}@is.iscas.ac.cn)

variant of probabilistic signatures. As probabilistic signatures can achieve tight security reduction and provide concrete security [6], [12], [16], our method seems rather natural. A semi-trusted off-line TTP is still involved, who generates a key pair and publishes the public key as a system parameter, and no registration is needed. We present a formal model of verifiable probabilistic signatures, and propose an efficient and provably secure RSA-based verifiable probabilistic signature scheme. The resulting optimistic fair exchange protocol works with any standard RSA signature schemes. It is the first concise and efficient RSA-based fair exchange protocol, and much suitable for engineering practices.

II. Verifiable Probabilistic Signature Model

Dodis and Reyzin [13] gave a formal definition of non-interactive fair exchanges via a new primitive called *verifiable committed signature*. A formal definition of non-interactive *verifiable encrypted signature* was also given in [7]. In the following, we would like to precisely present a formal definition of *verifiable probabilistic signature* scheme, by explicitly considering the attack models and security goals, which results in a concrete description for the security against all parties involved in the protocols.

A. Definitions of a Verifiable Probabilistic Signature

A verifiable probabilistic signature scheme involves three entities: a singer Alice, a verifier Bob and an arbitrator TTP, and is given by the following procedures.

Setup: A trapdoor one-way permutation f is first published by TTP as a system parameter, that is, TTP generates a key pair (PK, SK), and makes PK public, and keeps the corresponding trapdoor SK secret. The signer Alice generate her private signing key sk and the public verification key pk, and suppose the underlying standard signing and verification algorithms are Sig and Ver respectively.

Psig and Pver: These are probabilistic signing algorithm and verification algorithm. Given a message m, and keys sk and PK, a signer chooses a random number r, and outputs a probabilistic signature $\sigma = \mathsf{Psig}(sk, PK; m) = (m, r, \delta)$, where $\delta = \mathsf{Sig}(sk; m \| f_{PK}(r))$. The verification algorithm $\mathsf{Pver}(m, \sigma, pk, PK)$ takes as input m, σ and public keys pk and PK, and outputs 1 (accept) or 0 (reject).

VPsig and VPver: These are verifiable partial signing and verification algorithms. The verifiable partial signing algorithm VPsig behaves just like an ordinary probabilistic signing algorithm Psig, except it outputs the value $f_{PK}(r)$ instead of the random number r. Let the output be $\sigma' = \mathsf{VPsig}(sk, PK; m)$. The corresponding verification algorithm VPver is just the standard verification algorithm Ver on $m \| f_{PK}(r)$.

Resolution Algorithm: This is an algorithm run by an arbitrator TTP in case a singer Alice refuses to open her probabilistic signature σ to a verifier, who in turn possesses a valid verifiable partial signature σ'. In this case, $\mathsf{Res}(m, \sigma', SK, pk)$ should output a legal probabilistic signature σ on m.

The correctness of a verifiable probabilistic signature scheme states that

$$\begin{aligned}&\mathsf{Pver}(m, \mathsf{Psig}(sk, PK; m), pk, PK) = 1,\\ &\mathsf{VPver}(m, \mathsf{VPsig}(sk, PK; m), pk) = 1,\\ &\mathsf{Pver}\big(m, \mathsf{Res}\big(m, \sigma', SK, pk\big), pk, PK\big) = 1.\end{aligned}$$

In a verifiable probabilistic signature scheme, TTP only needs to publish a trapdoor one-way permutation as the system parameter. No further registration is needed and no zero-knowledge proofs are involved, which will greatly reduce the communication overhead and managing cost. Recall that in a verifiable committed signature scheme [13] and most of the verifiable encrypted signature schemes, TTP shall maintain a secret-public key pair for each user via a registration phase, and the secret keys will then be used to resolve a dispute.

B. Security of Verifiable Probabilistic Signatures

The security of a verifiable probabilistic signature scheme consists of ensuring fairness from three aspects: security against signer Alice, security against verifier Bob, and security against arbitrator TTP. In the following, we denote by O_{VPsig} an oracle simulating the verifiable probabilistic signing procedure, and O_{Res} an oracle simulating the resolution procedure. Let k be a security parameter, and PPT stand for "probabilistic polynomial time".

Security against a signer. Intuitively, a signer Alice should not be able to produce a verifiable probabilistic signature which is valid from a verifier's point of view, but which will not be extracted into a probabilistic signature of Alice by an honest arbitrator TTP. More precisely, we require that any PPT adversary $\mathcal{A}$ succeeds with at most negligible probability in the following experiment.

$$\begin{aligned}\mathsf{Setup}^*(1^k) &\rightarrow (sk^*, pk, SK, PK)\\ (m, \sigma') &\leftarrow \mathcal{A}^{O_{\mathsf{Res}}}(sk^*, pk, PK)\\ \sigma &\leftarrow \mathsf{Res}(m, \sigma', SK, pk)\\ \text{Success of}\mathcal{A} &= \big[\mathsf{VPver}(m, \sigma', pk) = 1 \wedge\\ &\qquad \mathsf{Pver}(m, \sigma, pk, PK) = 0\big].\end{aligned}$$

where Setup^* denotes the run of Setup with dishonest Alice (run by the adversary $\mathcal{A}$) and sk^* is $\mathcal{A}$'s state after this run.

Security Against Verifier. Verifier Bob should not be able to transfer any of the verifiable probabilistic signatures σ' that he got from Alice into a probabilistic signature σ, without explicitly asking TTP to do that. More precisely, we require that any PPT adversary $\mathcal{A}$ succeeds with at most negligible probability in the following experiment:

$$\begin{aligned}\mathsf{Setup}(1^k) &\rightarrow (sk, pk, SK, PK)\\ (m, \sigma) &\leftarrow \mathcal{A}^{O_{\mathsf{VPsig}}, O_{\mathsf{Res}}}(pk, PK)\\ \text{Success of}\mathcal{A} &= \big[\mathsf{Pver}(m, \sigma, pk, PK) = 1 \wedge\\ &\qquad m \notin \mathsf{Query}(\mathcal{A}, O_{\mathsf{Res}})\big],\end{aligned}$$

where $\mathsf{Query}(\mathcal{A}, O_{\mathsf{Res}})$ is the set of valid queries $\mathcal{A}$ asked to the resolution oracle O_{Res}, i.e., the set of (m, σ') the adversary $\mathcal{A}$ queried to O_{Res} satisfying $\mathsf{VPver}(m, \sigma', pk) = 1$.

Security against arbitrator. This property is crucial. Even though the arbitrator is semi-trusted, the primary signer Alice does not want the arbitrator to produce a valid probabilistic signature which she did not intend on producing. To achieve this goal, we require that any PPT adversary $\mathcal{A}$ associated with verifiable probabilistic signing oracle O_{VEsig}, succeeds with at most negligible probability in the following experiment:

$$\begin{aligned} \mathsf{Setup}^*(1^k) &\rightarrow (sk, pk, SK^*, PK) \\ (m, \sigma) &\leftarrow \mathcal{A}^{O_{\mathsf{VPsig}}}(SK^*, pk, PK) \\ \text{Success of}\mathcal{A} &= \big[\mathsf{Pver}(m, \sigma, pk, PK) = 1 \wedge \\ &\quad m \notin \mathsf{Query}(\mathcal{A}, O_{\mathsf{VPsig}})\big], \end{aligned}$$

where $\mathsf{Setup}^*(1^k)$ denotes the run of Setup with the dishonest arbitrator $\mathcal{A}$, and SK^* is her state after this run, and $\mathsf{Query}(\mathcal{A}, O_{\mathsf{VPsig}})$ is the set of queries $\mathcal{A}$ asked to the verifiable probabilistic signing oracle O_{VPsig}.

Definition 1. *A verifiable probabilistic signature scheme is secure if it is secure against signer attack, verifier attack and arbitrator attack.*

III. Efficient Verifiable Probabilistic Signature Scheme based on RSA

We shall present a verifiable probabilistic signature scheme based on the standard RSA-FDH (Full Domain Hash) signature scheme. As usual, let n be an RSA-modulus, which is a product of two distinct large primes, let $e \in \mathbf{Z}_n^*$ be a randomly chosen public exponent and d be a secret exponent satisfying $ed \equiv 1 \bmod \varphi(n)$. Let H be a collision-free hash function. The RSA-FDH signing algorithm gets inputs (n, d) and a message m, outputs a signature

$$\delta = \mathsf{Sig}(sk, m) = H(m)^d \bmod n.$$

The verifying algorithm $\mathsf{Ver}(pk, m, \delta)$ gets inputs (m, δ) and the public key (n, e), and accepts it if $\delta^e = H(m) \bmod n$ holds.

The RSA-FDH signature scheme has been proved [11] to be existentially unforgeable against adaptive chosen message attacks in the random oracle model [5], assuming that inverting RSA is hard. Now we present our scheme as following.

• **Setup.** TTP generates a public key $PK = N$ and publishes it as a system parameter, and keeps $SK = (P, Q)$ secret, where $N = PQ$ and P, Q are distinct strong primes of length k, i.e., $P = 2P' + 1$ and $Q = 2Q' + 1$, while P' and Q' are also primes. Denote by $H'(\cdot) = H(\cdot)\|1$, which maps any string to an odd integer, here H may be taken as SHA-1. Considering $\varphi(N) = 4P'Q'$, the probability of the output from H' being co-prime to $\varphi(N)$ is overwhelming, because finding an odd integer not co-prime with $4P'Q'$ is equivalent to find P' or Q' or $P'Q'$ and consequently factoring N.

Alice randomly chooses two primes p and q of length k, and sets $n = pq$. Then she generates two exponents e and d satisfying $ed \equiv 1 \bmod \varphi(n)$. Her private signing key is $sk = (p, q, d)$ and the public verification key is $pk = (n, e)$.

• **Psig and Pver:** To probabilistically sign a message m, Alice first randomly chooses a number $r \in \mathbf{Z}_N^*$ and computes

$$y = f_{PK}(r) = r^{H'(ID\|pk)} \bmod N, \tag{1}$$

where ID is Alice's identity. Then Alice computes

$$\delta = \mathsf{Sig}(sk, m\|y) = H(m\|y)^d \bmod n.$$

The probabilistic signature for message m is $\sigma = (m, r, \delta)$.

The corresponding verification algorithm Pver takes as input σ, computes y as (1), and verify $H(m\|y) = \delta^e \bmod n$.

• **VPsig and VPver:** For a message m, VPsig first runs $\mathsf{Psig}(m, sk, PK)$. Let $\sigma = (m, r, \delta)$ be the output of Psig, and y be the value satisfying (1). Then the verifiable probabilistic signature generated by Alice for a message m is

$$\sigma' = \mathsf{VPsig}(m, sk, PK) = (m, y, \delta).$$

On inputs $\sigma' = (m, y, \delta)$ and Alice's public key (n, e), the algorithm VPver checks

$$H(m\|y) = \delta^e \bmod n,$$

and accepts $\sigma' = (m, y, \delta)$ as a valid verifiable probabilistic signature only if the above equation holds.

• **Res:** Given a verifiable partial signature $\sigma' = (m, y, \delta)$, the arbitrator TTP first verifies its validity by checking $H(m\|y) = \delta^e \bmod n$. If valid, TTP computes

$$r = y^{H'(ID\|pk)^{-1} \bmod \varphi(N)} \bmod N \tag{2}$$

and returns $\sigma = (m, r, \delta) = \mathsf{Res}(m, \sigma', SK, pk)$ as a probabilistic signature of m to the verifier.

Note that, TTP actually specifies a family of one-way trapdoor permutations by publishing N, for which the common trapdoor is (P, Q). Although the encryption exponents $H'(ID\|pk)$ are different for distinct signers, TTP can always extract a number $r \in \mathbf{Z}_N^*$ satisfying (2), for any m and y. And for a valid verifiable partial signature (m, y, δ), we have $H(m\|y) = \delta^e \bmod n$, thus the output $\sigma = (m, r, \delta)$ of Res is a valid probabilistic signature on m.

Remark 1: (a) For a particular signer with identity ID and public key pk, $H'(ID\|pk)$ is a fixed encryption exponent. Thus $y = f_{PK}(r)$ is a permutation on $\mathbf{Z}_N^*$ and y is uniformly distributed as r. Therefore the probabilistic signature scheme Psig is actually a RSA-PFDH signature scheme proposed by Coron [12], which is provably secure in the random oracle with a tight security reduction. (b) The FDH signature scheme can be replaced by any other secure signature scheme such as RSA-PSS [16]. (c) Although a common modulus N is used as a system parameter, the common modulus attack does not work here, since the encryption exponent is fixed for each signer, and the "plaintext" r is chosen at random.

A. Security of Our Scheme

Theorem 1. *Under the formal model described in section 3, the verifiable probabilistic signature scheme based on RSA is provably secure in the random oracle model, provided that inverting RSA function is hard.*

Proof. According to Definition 1, we shall show that the proposed verifiable probabilistic signature schemes is secure against signer, verifier and arbitrator. Note that the underlying RSA-FDH signature scheme is existentially unforgeable against adaptive chosen message attack in the random oracle.

Hence the probability of a valid forgery for the RSA-FDH signature scheme is negligible.

Secure against signer's attack: For a malicious signer, with the help of the oracle O_{Res}, her goal is to produce a valid verifiable probabilistic signature $\sigma' = (m, y, \delta)$, which cannot be extracted into a valid probabilistic signature $\sigma = (m, r, \delta)$. However, this is always not the case. For any y and m, the number $r \in \mathbf{Z}_N^*$ satisfying $y = r^{H'(ID\|pk)} \bmod N$ can always be extracted as (2) using the trapdoor $SK = (P, Q)$. For this extracted r there holds $y = r^{H'(ID\|pk)} \bmod N$. And for a valid verifiable partial signature (m, y, δ), we have $H(m\|y) = \delta^e \bmod n$. Thus the resulting triplet (m, r, δ) is definitely a valid probabilistic signature on m, and Alice cannot deny it. In fact, the oracle O_{Res} cannot give any help to a malicious signer: what O_{Res} extracted is exactly the number r she used to compute the value y, which was already known to her.

Secure against verifier's attack: An adversarial verifier's goal, making use of oracles O_{VPsig} and O_{Res}, is to forge a valid probabilistic signature $\sigma = (m, r, \delta)$, for which the corresponding verifiable partial signature $\sigma' = (m, y, \delta)$ has not been queried to O_{Res}. We shall convert such an attack into a forger $\mathcal{F}$ against the RSA-FDH signature scheme. Note that $\mathcal{F}$ takes as input $pk = (e, n)$ and has access to the signing oracle O_{Sig} of RSA-FDH signature scheme. While Bob accepts pk and PK as inputs, and has access to oracles O_{VPsig} and O_{Res}, and wins if he forges a probabilistic signature σ for some message m without making a query (m, σ') to O_{Res}.

To invoke Bob, $\mathcal{F}$ shall answer Bob's O_{VPsig}-queries and O_{Res}-queries by himself. For an O_{VPsig} query on message m_i, $\mathcal{F}$ chooses a number $r_i \in \mathbf{Z}_N^*$ at random, and computes $y_i = {r_i}^{H'(ID\|pk)} \bmod N$, and then queries its own signing oracle O_{Sig} on message $m_i\|y_i$ to get a signature $\delta_i = \mathsf{Sig}(sk, m_i\|y_i)$. Now $\mathcal{F}$ produces a verifiable partial signature $\sigma_i' = (m_i, y_i, \delta_i)$, and sends σ_i' to Bob as the answer. $\mathcal{F}$ keeps a list of $L = \{(m_i, \sigma_i' = (m_i, y_i, \delta_i), r_i)\}$. To simulate a valid O_{Res}-query on (m', σ'), $\mathcal{F}$ just looks up the list L, answers Bob with r_i if (m', σ', r_i) is in the list, and halts otherwise. Note that, for a valid O_{Res}-query (m', σ') where $\sigma' = (m', y', \delta')$, there must hold $\delta' = \mathsf{Sig}(sk, m'\|y')$. Hence the probability that m' has not been queried to O_{VPsig} (which means that $(m'\|y', \delta')$ is a valid RSA-FDH forgery) is negligible, and so is it with $\mathcal{F}$ halts in answering O_{Res}-queries.

Suppose Bob outputs a probabilistic signature forgery $\tilde{\sigma} = (\tilde{m}, \tilde{r}, \tilde{\delta})$ in the ultimate. Let $\tilde{y} = \tilde{r}^{H'(ID\|pk)} \bmod N$. If $(\tilde{m}, \tilde{y}, \tilde{\delta}) \neq (m_i, y_i, \delta_i)$ for all i, $\mathcal{F}$ outputs $(\tilde{m}\|\tilde{y}, \tilde{\delta})$, which is a valid forgery for the RSA-FDH signature scheme, since $\tilde{m}\|\tilde{y}$ have never been queried to O_{Sig}. Otherwise $\mathcal{F}$ halts. In the latter case, $\tilde{\sigma}' = (\tilde{m}, \tilde{y}, \tilde{\delta})$ is an output of O_{VPsig}-query, but $(\tilde{m}, \tilde{\sigma}')$ has not been queried to O_{Res}. Bob may try to extract the number $\tilde{r} \in \mathbf{Z}_N^*$ the signer used to computes $\tilde{y}$. Note that

$$\tilde{r}^{H'(ID\|pk)} = \tilde{y} \bmod N,$$

extracting $\tilde{r}$ from the above equation is actually the intractable problem of inverting RSA functions. Hence the probability of the latter case occurs is negligible. As a result, if Bob can success with non-negligible probability, then $\mathcal{F}$ can succeed in producing a valid forgery of the RSA-FDH signature scheme with non-negligible probability.

Secure against arbitrator's attack: Now we consider an adversarial TTP's attack. Holding the trapdoor $SK = (P, Q)$ of the one-way permutation, TTP can extract any y into a pair of (m, r) satisfying (2). We shall also show a reduction of converting an arbitrator's attack into a valid forgery for the RSA-FDH signature scheme. As before, a forger $\mathcal{F}$ accepts $pk = (e, n)$ as input and has oracle access to the signing oracle O_{Sig} of RSA-FDH signature scheme. TTP holds (PK, SK) and has access to the O_{VPsig}-oracle, and wins if he forges a probabilistic signature $\tilde{\sigma} = (\tilde{m}, \tilde{r}, \tilde{\delta})$, for which

$$H(\tilde{m}\|\tilde{r}^{H'(ID\|pk)} \bmod N) = \delta^e \bmod n$$

holds, while $\tilde{m}$ has not been queried to the oracle O_{VPsig}.

Here is how $\mathcal{F}$ invokes TTP. For an O_{VPsig}-query on message m, $\mathcal{F}$ randomly chooses $r \in \mathbf{Z}_N^*$ and computes $y = r^{H'(ID\|pk)} \bmod N$, and then makes a O_{Sig}-query on $m\|y$ to obtain a signature $\delta = \mathsf{Sig}(sk, m\|y)$. $\mathcal{F}$ answers TTP with (m, y, δ) as a valid verifiable partial signature. When TTP outputs a forgery $(\tilde{m}, \tilde{\sigma})$ as described above, then $\tilde{\delta}$ is a valid forgery on $m' = \tilde{m}\|\tilde{r}^{H'(ID\|pk)} \bmod N$ under $pk = (e, n)$, since $\tilde{m}$ has not been queried to O_{VPsig}. $\mathcal{F}$ just outputs (m', δ). We see that the simulation is perfect, and $\mathcal{F}$ succeeds in generating a valid forgery if TTP succeeds.

The above arguments show that, if an adversary can attack our verifiable probabilistic signature scheme with non-negligible probability, then one can break the existential unforgeability of RSA-FDH signatures under adaptive chosen-message attacks, with almost the same success probability. Thus the security of our scheme follows from the security of RSA-FDH scheme, which in turn relys on the well-known RSA-assumption that inverting RSA function is hard. □

Remark 2: **Security against colluding attacks**. Another powerful attack we must take into consideration is a colluding attack proposed recently by Bao [3]. If an adversary can manage to extract r, then she get a valid probabilistic signature. However, the adversary cannot extract r from y by herself, since it is a intractable problem of inverting the RSA function. Moreover, since r is explicitly bound with a signer's ID and pk as $y = r^{H'(ID\|pk)} \bmod N$, it is infeasible for an adversary to generate $y' = r^{H'(ID'\|pk')} \bmod N$ for a different ID' and pk' from y, as shown in Lemma 2. Therefore, the colluding attack [3] doesn't work here.

Lemma 2. *Let n be an RSA modulus. Given y and h such that $y = r^h \bmod n$ for a unknown r, if one can generate y' and h' such that $y' = r^{h'} \bmod n$, where both h and h' are odd integers, then there exists an efficient algorithm to compute x and z such that $y = z^x \bmod n$.*

Proof. Let $\tilde{h}$ be the least common multiple of h and h', and $\tilde{h} = th$, $\tilde{h} = t'h'$. Then we have

$$y^t = r^{ht} = r^{\tilde{h}} = r^{t'h'} = y'^{t'} \bmod n.$$

Let $c = \gcd(t, t')$. Then $2 \nmid c$ and $\gcd(c, \varphi(n)) = 1$, otherwise one can efficiently factor n. There exists a and b such that $a\frac{t'}{c} + b\frac{t}{c} = 1$. Set $z = y^a y'^b$ and $x = \frac{t'}{c}$. Then we have

$$z^x = z^{\frac{t'}{c}} = y^{a\frac{t'}{c}} y'^{b\frac{t'}{c}} = y^{a\frac{t'}{c}} y^{b\frac{t}{c}} = y.$$

The Strong RSA Assumption [10] states that, on input an RSA modulus n and an element $y \in \mathbf{Z}_n^*$, it is infeasible to computes

values $x > 1$ and z such that $z^x = y \bmod n$. Then, according to Lemma 2, any adversary is infeasible to find a y' from y, which encrypt the same r under different public exponents. Thus the proposed verifiable probabilistic signatures are secure against colluding attacks.

IV. Fair Exchanges based on Probabilistic Signatures

Now we present an optimistic fair exchange protocol based on the probabilistic signatures described as in section 3. The construction is similar to [7], [13].

Assume the public key of Alice is $pk_A = (n_A, e_A)$ and the private key is $sk_A = (p_A, q_A, d_A)$, and Bob's public key is $pk_B = (n_B, e_B)$ and private key is $sk_B = (p_B, q_B, d_B)$. The public key of a TTP is $PK = N$ while the private key is $SK = (P, Q)$. Here $n_A = p_A \cdot q_A$, $n_B = p_B \cdot q_B$, $N = P \cdot Q$, p_A, q_A, p_B, q_B are primes and P, Q are strong primes.

1. Alice chooses $r_A \in \mathbf{Z}_N^*$ at random, computes $y_A = r_A^{H'(ID_A \| pk_A)} \bmod N$, and generates $\delta_A = H(m \| y_A)^{d_A} \bmod n_A$. Then Alice sends a verifiable probabilistic signature $\sigma'_{Alice} = (m, y_A, \delta_A)$ to Bob.

2. Bob first checks $H(m \| y_A) = {\delta_A}^{e_A} \bmod n_A$. If it is valid, Bob chooses $r_B \in \mathbf{Z}_N^*$ at random, and then computes $y_B = r_B^{H'(ID_B \| pk_B)} \bmod N$, $\delta_B = H(m \| y_B)^{d_B} \bmod n_B$. Bob sends his probabilistic signature $\sigma_{Bob} = (m, r_B, \delta_B)$ to Alice.

3. After receiving Bob's probabilistic signature $\sigma_{Bob} = (m, r_B, \delta_B)$, Alice computes $y_B = r_B^{H'(ID_B \| pk_B)} \bmod N$, and verifies ${\delta_B}^{e_B} = H(m \| y_B) \bmod n_B$. If valid, she sends $\sigma_{Alice} = (m, r_A, \delta_A)$ to Bob.

4. If Bob does not receive anything in step 3, or if σ_{Alice} is invalid, then he sends the verifiable partial signature $\sigma'_{Alice} = (m, y_A, \delta_A)$ and his probabilistic signature $\sigma_{Bob} = (m, r_B, \delta_B)$ to TTP. This protocol provides a vehicle for TTP to understand whether the protocol was correctly carried out. TTP first computes $y_B = r_B^{H'(ID_B \| pk_B)} \bmod N$. If both

$$ {\delta_B}^{e_B} = H(m \| y_B) \bmod n_B $$

and

$$ {\delta_A}^{e_A} = H(m \| y_A) \bmod n_A, $$

hold, TTP extracts

$$ r'_A = {y_A}^{H'(ID_A \| pk_A)^{-1} \bmod \varphi(N)} \bmod N. $$

Then TTP sends $\sigma_{Alice} = (m, r_A, \delta_A)$ to Bob and sends $\sigma_{Bob} = (m, r_B, \delta_B)$ to Alice.

Security of the protocol follows directly from Theorem 1 and Remark 2. The proposed protocol is concise and efficient, and works with standard RSA signature schemes.

V. Conclusion

We introduce a formal definition of verifiable probabilistic signature for constructing optimistic fair exchange protocols, and present an efficient and provably secure verifiable probabilistic signature scheme based on RSA signatures. The proposed fair exchange protocol works with standard RSA signature schemes. No further registration is needed and no zero-knowledge proofs are involved. This is the first concise and efficient RSA-based fair-exchange protocol suitable for cryptographic engineering practices. It is very interesting to explore other probabilistic signatures to construct efficient and practical fair exchanges as well as other electronic commerce protocols.

Acknowledgement

The work is supported by National Natural Science Foundation of China (60373039), and National Grand Fundamental Research Project of China (G1999035802). We have applied for Chinese Patents regarding the scheme proposed in the paper.

References

[1] N.Asokan, V.Shoup, M.Waidner. Optimistic fair exchange of digital signatures. Advances in Cryptology - EUROCRYPT'98, LNCS 1403, pages 591-606, Springer-Verlag, 1998; IEEE J. on Selected Areas in Communication, 18(4): 593-610, 2000.

[2] G.Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. Sixth ACM Conference on Computer and Communication Security, pages 138-146. ACM, 1999; Verifiable encryption of digital signatures and applications, ACM Transactions on Information and System Security, Vol. 7, No. 1, pages 1-20, 2004.

[3] F. Bao. Colluding Attacks to a Payment Protocol and Two Signature Exchange Schemes. Advances in Cryptology-ASIACRYPT 2004, LNCS 3329, pages 417-429, Springer-Verlag, 2004.

[4] F. Bao, R.H. Deng, W. Mao. Efficient and practical fair exchange protocols with off-line TTP. IEEE Symposium on Security and Privacy, pages 77-85, 1998.

[5] M. Bellare and P. Rogaway: Random oracles are practical: a paradigm for designing efficient protocols. Proceedings of the First Annual Conference on Computer and Commmunications Security, ACM, 1993.

[6] M. Bellare and P. Rogaway. The exact security of digital signatures - how to sign with RSA and Rabin. Eurocrypt'96.

[7] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. Advances in Cryptology-EUROCRYPT 2003, LNCS 2656, pp. 416-432. Springer-Verlag, 2003.

[8] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. Advances in Cryptology-ASIACRYPT 2001, LNCS 2248, pages 514-532, Springer-Verlag, 2001.

[9] J. Camenisch and I. B. Damgard. Verifiable encryption, group encryption, and their applications to group signatures and signature sharing schemes. Advances in Cryptology-ASIACRYPT 2000, LNCS 1976, pages 331-345, Kyoto, Japan, 3.7 Dec. 2000. Springer-Verlag.

[10] R. Cramer, V. Shoup. Signature Schemes Based on the Strong RSA Assumption. ACM Transactions on Information and System Security, Vol. 3, No. 3, pages 161-185, August 2000.

[11] J.S. Coron. On the exact security of Full Domain Hash. Advances in Cryptology-Crypto 2000, LNCS 1880, pp.229-235, Springer-Verlag, 2000.

[12] J.S. Coron. Optimal Security Proofs for PSS and Other Signature Schemes, Advances in Cryptology - EUROCRYPT 2002, LNCS 2332, pages 272-287, Springer-Verlag, 2002.

[13] Y. Dodis and L. Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003. ACM Workshop on Digital Rights Management (DRM), pages 47-54, October 2003.

[14] X. Ding, G. Tsudik. Simple Identity-Based Encryption with Mediated-RSA. Topics in Cryptology-CT-RSA 2003, LNCS 2612, pages 193-210, Springer-Verlag 2003.

[15] J. M. Park, E. Chong, H. Siegel, I. Ray. Constructing fair exchange protocols for E-commerce via distributed computation of RSA signatures. In 22^{th} ACM Symp. on Principles of Distributed Computing, pages 172-181, 2003.

[16] RSA Labs: RSA Cryptography Standard: EMSAPSS-PKCS♯1 v2.1.

[17] Z. F. Zhang and D. G. Feng. Simple fair exchange based on mediated-RSA and factoring representation. PreProceeding of 4rd Workshop on Information Security Application, Jeju Island, Korea, pp. 689-696, 2003.

[18] Y. B. Zhou, Z. F. Zhang, S. H. Qing, J. Liu. A New CEMBS Based on RSA Signatures and Its Application in Constructing Fair Exchange Protocol. Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04), pages 560-564, 2004.