Efficient ID-Based Optimistic Fair Exchange with Provable Security

Zhenfeng Zhang[1,2], Dengguo Feng[1,2], Jing Xu[1,3], and Yongbin Zhou[1,2]

[1] State Key Laboratory of Information Security
[2] Institute of Software, Chinese Academy of Sciences, Beijing 100080, P.R. China
[3] Graduate School of Chinese Academy of Sciences, Beijing 100039, P.R. China
zfzhang@is.iscas.ac.cn

Abstract. The notion of identity based cryptosystem was introduced by Shamir in 1984, and has attracted much interest since it eliminates the need of certificates and simplify the key management. In this paper, we propose an optimistic fair exchange protocol for identity-based signatures. A semi-trust third party (TTP) is still involved in our protocol to ensure fairness. However, there is no need for registrations between users and TTP, and no zero-knowledge proof is needed to provide verifiability. The proposed optimistic fair exchange protocol is much concise and efficient, and can be shown to be secure in the random model with a tight security reduction.

Keywords: Fair exchange, Identity-based Signature, Provable Security.

1 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 [1,5] 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.

S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, pp. 14–26, 2005.

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 [11], 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 [5,2]. The first and only non-interactive verifiably encrypted signature scheme was constructed by Boneh et al. [8], which is provably secure in the random oracle model, and is the first elegant scheme without a special registration between users and TTP and without zero-knowledge proofs.

Recently, Park etc. [17] 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 [14]. Moreover, Dodis and Reyzin [14] proposed a new primitive called *verifiably committed signatures* for constructing fair exchange protocols, and presented a committed signature scheme based on GDH signatures [9]. However, a registration protocol between TTP and users is still needed, and a zero-knowledge proofs of the equality of two discrete logarithms are involved to ensure the fairness.

Motivated by the approaches of verifiably encrypted signatures and verifiably committed signatures, the authors of [21] introduce a paradigm called *verifiable probabilistic signature schemes*, in which the exchanged items are probabilistic signatures. As probabilistic signatures has been studied extensively, this method seems rather natural. In their paradigm, a semi-trusted off-line TTP generates a trapdoor permutation as the system parameter, which can be used to produce verifiable partial signatures, while no further registrations between TTP and users is needed and no zero-knowledge proofs are involved. Thus, their framework has almost-optimal structures as that of [8]. Their approach is generic, and the resulting fair exchange protocol works especially with standard RSA signatures [18]. While being very concise and efficient, the only presented trapdoor permutation is based the factoring problem.

In 1984, Shamir [19] introduced the notion of identity-based cryptography (ID-PKC), in which a user's public-key can be derived form his unique identifier information. The ID-PKC eliminates certificates and greatly simplifies the key management. A breakthrough work in the research of ID-PKC shall owe to Boneh and Franklin [10], who proposed the first efficient identity encryption scheme based on bilinear pairings over elliptic curves. Since then, a great deal of research has been done about the ID-based cryptosystems and protocols. However, as far as we know, no efficient identity based fair exchanges has been proposed. Although the approach proposed by Zhang etc. [21] can be applied to identity based signatures, the specified trapdoor permutation based on factoring may be not desirable in some applications, especially for schemes over elliptic curves.

In this paper, we propose an optimistic protocol for fair exchanges of identity-based signatures. A semi-trusted off-line TTP is still involved, who generates a public-key as the system parameter, while keeps the corresponding private-key secret to settle the dispute. No registration between users and TTP is needed and

no zero-knowledge proofs are involved. The underlying identity-based signatures utilizing bilinear pairings over elliptic curves, and the public-key chosen by TTP is a point over elliptic curves. The proposed protocol is as concise as that in [8,21], and is provably secure in the random oracle model.

It should be noted that, Micali [16] presented a fair electronic exchange protocol for contract signing with an invisible trusted party in PODC 2003, which has a similar framework that does not need registrations between users and TTP. However, Bao et al. [4] showed that Micali's protocol cannot achieve the claimed fairness: the trusted party may face a dilemma situation that no matter what it does, one of the exchanging parties can succeed in cheating, and proposed a revised version that preserves fairness while remaining optimistic. Although the correctness of the revised version is believable, no security proof is provided. Moreover, a semantically secure encryption scheme under adaptive chosen ciphertext attacks is needed both in [16] and [4], and a random number specified by the initial party will be used by both parties for the underlying semantically-secure encryption.

The rest of the paper is organized as follows. In Section 2, we present a brief description of the formal model and security of identity-based verifiable probabilistic signatures. In Section 3, we propose a concrete identity-based verifiable probabilistic signature scheme and prove its security in the random oracle model. An identity-based optimistic fair exchange protocol is presented in section 4. Section 5 concludes the paper.

2 Verifiable Probabilistic Signature Model

Formal definitions of non-interactive fair exchanges via verifiable committed signatures or verifiable probabilistic signatures are proposed in [14] and [21] respectively, which explicitly consider the attack model and security goals, and result in a concrete description for the security against all parties involved in the protocols. Similar to [14,21], we present a formal model for the identity-based fair exchanges. And, to emphasize the role of *probabilistic* signature, we term it identity-based verifiable probabilistic signature scheme.

2.1 Definitions of ID-Based Verifiable Probabilistic Signatures

In an identity-based cryptosystem, there is a trusted authority called the *private key generator* (PKG) who holds a master key and issues private keys for all users in the system domain, and the public-key of a user can be derived publicly and directly form his unique identifier information. An identity-based verifiable probabilistic signature scheme involves three entities: a singer, a verifier and an arbitrator TTP, and is given by the following procedures.

Setup: System parameters param and a master key s is first generated by the PKG of an identity-based cryptosystem. A trapdoor one-way permutation is also published by a trusted third party (TTP) as a system parameter, that is, TTP

generates a key pair (PK, SK), and makes PK public while keeps the trapdoor SK secret. Note that the TTP may be different from PKG.

Extract: Given a user's identity id, the PKG computes a private-key sk corresponding to id using his master-key s, and transmits it to the user. The user's public-key can be regarded as id and the corresponding private-key is sk.

Psig and Pver: These are probabilistic signing algorithm and verification algorithm. Given a message m, and private key sk, a signer outputs a probabilistic signature $\sigma = \mathsf{Psig}(sk, m)$. The corresponding verification algorithm $\mathsf{Pver}(m, \sigma, id)$ takes as input m, σ and the signer's identity id, outputs 1 or 0.

VPsig and VPver: These are verifiable probabilistic signing and verification algorithms, which are just like an ordinary probabilistic signing and verification algorithms, except they depend on the public key PK. Given a message m, and keys sk and PK, a signer outputs a verifiable partial signature $\sigma' = \mathsf{VPsig}(sk, PK, m)$. The corresponding verification algorithm $\mathsf{VPver}(m, \sigma', id, PK)$ takes as input m, σ' and public keys id and PK, outputs 1 (accept) or 0 (reject).

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

The correctness of a verifiable probabilistic signature scheme states that

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

In a verifiable probabilistic signature model, TTP does not need to store anything except the trapdoor of the published one-way permutation. No further registration between users and TTP is needed, which will greatly reduce the communication overhead and managing cost. While in a verifiable committed signature scheme [14] and most of the verifiable encrypted signature schemes except [8], 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.

2.2 Security of ID-Based Verifiable Probabilistic Signatures

The security of a verifiable probabilistic signature scheme consists of ensuring fairness from three aspects: security against signer, security against verifier, and security against arbitrator. 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, and let O_{Ext} be an oracle simulating the private-key extracting operation. Let k be a suitable security parameter, and PPT stand for "probabilistic polynomial time".

Security against a signer: Intuitively, a signer 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 the signer

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.

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

In the model of considering security against signers, we allow an adversary $\mathcal{A}$ to have the strongest power of extracting the private-key for any identity id.

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

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

where $\mathsf{Query}(\mathcal{A}, O_{\mathsf{Res}})$ is the set of valid queries $\mathcal{A}$ asked to O_{Res}, i.e., the set of (m, σ', id) the adversary $\mathcal{A}$ queried to O_{Res} satisfying $\mathsf{VPver}(m, \sigma', id, PK) = 1$, $\mathsf{Query}(\mathcal{A}, O_{\mathsf{Ext}})$ is the set of queries $\mathcal{A}$ asked to the private-key-extractor O_{Ext}.

Security against the arbitrator: An arbitrator's work is to check the validity of a request and recover the required probabilistic signature in case of dispute. However, a signer does not want the arbitrator to produce a valid probabilistic signature which she did not intend to produce, so we require the arbitrator to be *semi-trusted* in our model. To achieve this goal, we require that any PPT adversary $\mathcal{A}$, associated with verifiable probabilistic signing oracle O_{VPsig}, succeeds with at most negligible probability in the following experiment:

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

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's attack, verifier's attack and arbitrator's attack.*

3 ID-Based Verifiable Probabilistic Signature Scheme

We shall present a verifiable probabilistic signature scheme based on Bellare et al.s [7] modified Sakai-Ogishi-Kasahara signature scheme [20], which was com-

monly called SOK-IBS (for Sakai-Ogishi-Kasahara Identity Based Signature) in literatures [7]. In fact, the SOK-IBS scheme can be regarded as an identity based extension of a randomized version of Boneh et al.'s short signature scheme [9].

3.1 The Bilinear Pairing

Let $\mathcal{G}_1$ be a cyclic additive group generated by P, whose order is a prime q, and $\mathcal{G}_2$ be a cyclic multiplicative group of the same order. Let $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$ be a pairing which satisfies the following conditions:

1. Bilinearity: For any $P, Q, R \in \mathcal{G}_1$, we have $e(P + Q, R) = e(P, R)e(Q, R)$ and $e(P, Q + R) = e(P, Q)e(P, R)$. In particular, for any $a, b \in \mathbf{Z}_q$,

$$e(aP, bP) = e(P, P)^{ab} = e(P, abP) = e(abP, P).$$

2. Non-degeneracy: There exists $P, Q \in \mathcal{G}_1$, such that $e(P, Q) \neq 1$.
3. Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in \mathcal{G}_1$.

The typical way of obtaining such pairings is by deriving them from the Weil-pairing or the Tate-pairing on an elliptic curve over a finite field. We refer to [9,10] for a more comprehensive description on how these groups, pairings and other parameters should be selected for efficiency and security.

Computation Diffie-Hellman (CDH) Problem: Given P, aP, $bP \in \mathcal{G}_1$ for randomly chosen $a, b \in_{\mathcal{R}} \mathbf{Z}_q^*$, to compute abP.

3.2 The Proposed Scheme

Since 2001, all kinds of identity-based cryptosystems have been proposed based on the bilinear maps, such as Weil-pairing or Tate-pairing on an elliptic curve over a finite field. The following is a brief overview of the identity-based setting. We refer to [10] for a detailed description.

• **Setup:** Given a security parameter k, the PKG chooses groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of prime order $q > 2^k$, a generator P of $\mathcal{G}_1$, a bilinear map $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$, a randomly chosen master key $s \in \mathbf{Z}_q^*$ and the associated public key $P_{pub} = sP$. It also picks cryptographic hash functions of same domain and range $H_1, H_2 : \{0, 1\}^* \to \mathcal{G}_1$. The system's public parameters are $\texttt{params} = (\mathcal{G}_1, \mathcal{G}_2, e, P, P_{pub}, H_1, H_2)$.

The TTP chooses $x \in \mathbf{Z}_q^*$ at random, generates a public key $PK = xP$ and publishes it as a system parameter, and keeps $SK = x$ secret.

• **Extract:** Suppose the identity of a user is ID. Given an identity ID, the PKG computes $Q_{ID} = H_1(ID) \in \mathcal{G}_1$ and $d_{ID} = sQ_{ID} \in \mathcal{G}_1$ that is transmitted to the user. The user's private-key is d_{ID}, which satisfies $e(d_{ID}, P) = e(Q_{ID}, P_{pub})$.

• **Psig and Pver:** These are the signing and verification algorithms of the SOK-IBS scheme. In order to sign a message m, Psig perform as following:

– Pick $r \in_{\mathcal{R}} \mathbf{Z}_q$, compute $U = rP \in \mathcal{G}_1$ and $H = H_2(ID, m, U) \in \mathcal{G}_1$.
– Compute $V = d_{ID} + rH \in \mathcal{G}_1$.

The signature on m is the pair $\sigma = \langle U, V \rangle \in \mathcal{G}_1 \times \mathcal{G}_1$.

To verify a signature $\sigma = \langle U, V \rangle \in \mathcal{G}_1 \times \mathcal{G}_1$ on a message m for an identity ID, the algorithm Pver first takes

$$Q_{ID} = H_1(ID) \in \mathcal{G}_1 \quad \text{and} \quad H = H_2(ID, m, U) \in \mathcal{G}_1,$$

and then accepts the signature if and only if

$$e(P, V) = e(P_{pub}, Q_{ID}) \cdot e(U, H). \tag{1}$$

The signature scheme constituted of (Psig, Pver) is actually the SOK-IBS scheme, which has been proved [7] to be non-existentially forgeable against adaptive chosen message attacks in the random oracle model [6], first by Bellare etc. [7] through a general framework applying to a large family of schemes, and then by Libert and Quisquater [15], who showed that SOK-IBS scheme has a much tighter security proof under the CDH assumption.

• **VPsig and VPver:** To generate a partial signature a message m, VPsig performs as following:

– First choose $r \in \mathbf{Z}_q$ at random and compute $U = rP \in \mathcal{G}_1$, and then let $H = H_2(ID, m, U) \in \mathcal{G}_1$.
– Compute $V' = d_{ID} + rH + rPK \in \mathcal{G}_1$.

The verifiable partial signature on m is the pair $\sigma' = \langle U, V' \rangle \in \mathcal{G}_1 \times \mathcal{G}_1$.

The corresponding verification algorithm VPver takes as input σ', ID and PK. It first computes $Q_{ID} = H_1(ID) \in \mathcal{G}_1$ and $H = H_2(ID, m, U) \in \mathcal{G}_1$, and then accepts the signature if

$$e(P, V') = e(P_{pub}, Q_{ID}) \cdot e(U, H + PK), \tag{2}$$

and rejects it otherwise.

• **Res:** Given a verifiable partial signature $\sigma' = (U, V')$ on a message m for an identity ID, the arbitrator TTP first verifies its validity by checking (2). If valid, TTP computes

$$V = V' - xU \tag{3}$$

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

Note that, TTP holds the trapdoor $SK = x$, and if $U = rP$, then we have

$$r \cdot PK = r \cdot xP = x \cdot rP = xU.$$

That is, if $U = rP$ and $V' = d_{ID} + rH + rPK$, then $V = V' - xU = d_{ID} + rH$. Thus $\langle U, V \rangle$ is a regular SOK-IBS signature on message m for the identity ID.

Remark: The following facts shall be noted.

- Similar to the proofs [7,15] of SOK-IBS signature scheme, the verifiable partial signature scheme constituted of $(\mathsf{VPsig}, \mathsf{VPver})$ can be shown to be non-existential forgeable under adaptive chosen message attacks in the random oracle model, assuming the CDH problem in $\mathcal{G}_1$ is hard.
- The signer's identity ID is explicitly included in the signature as $H = H_2(ID, m, U)$, thus the colluding attacks proposed by Bao [3] will not work.
- This approach also works for other identity-based signatures, such as [12].
- In our scheme, the TTP is semi-trusted and can be any party different from PKG. This makes our scheme more flexible. In fact, if we designate PKG as the arbitrator, then this TTP must be fully trusted since every user's private key is escrowed by it.

3.3 Security of Our Scheme

Theorem 1. *Under the formal model described in section 2, the verifiable probabilistic signature scheme based on SOK-IBS is provably secure in the random oracle model, provided that the CDH problem is hard.*

Proof. According to Definition 1, we shall show that the proposed verifiable probabilistic signatures is secure against signer, verifier and arbitrator. As we will see, the security against a signer follows unconditionally, and an arbitrator's attack can be converted into a forger for the SOK-IBS signature, while a verifier's attack can be related to the CDH problem. The major difficulty for the proof of security against a verifier comes from the dealing with the resolution queries.

Security against signer's attack: With the help of the oracles O_{Res} and O_{Ext}, a malicious signer's goal is to produce a valid verifiable partial signature $\sigma' = (U, V')$, which cannot be extracted into a valid probabilistic signature $\sigma = (U, V)$. However, this is always not the case. In fact, for any $\sigma' = (U, V')$ satisfying $e(P, V') = e(P_{pub}, Q_{ID}) \cdot e(U, H + PK)$, and $V = V' - xU$, we have

$$e(P, V) = e(P, V')e(P, -xU) = e(P, V')e(PK, U)^{-1} = e(P_{pub}, Q_{ID}) \cdot e(U, H).$$

Thus the (U, V) extracted by TTP is definitely a valid SOK-IBS signature on m, and the signer 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 xU, which was already known to her as $rPK = rxP = xU$.

Security against verifier's attack: Making use of the oracles O_{VPsig}, O_{Ext} and O_{Res}, an adversarial verifier wins if he forges a valid probabilistic signature $\sigma = (U, V)$ for an entity with identity ID, for which the corresponding verifiable partial signature $\sigma' = (U, V')$ has not been queried to O_{Res}, and ID has not been queried to O_{Ext}. We shall show that such an attack can be used by a probabilistic polynomial time algorithm $\mathcal{F}$ to solve the CDH problem.

Let $(X = aP, Y = bP) \in \mathcal{G}_1 \times \mathcal{G}_1$ be a random instance of the CDH problem taken as input by $\mathcal{F}$. $\mathcal{F}$ takes $z \in \mathbf{Z}_q^*$ at random and sets $PK = zY$, and

then initializes Bob with $P_{pub} = X$ and PK as system's overall public keys. The algorithm $\mathcal{F}$ then starts performing queries such as those required by an identity based setting and the security model described as in section 2. Without loss of generality, we assume that, for any key extraction query or signature query involving an identity, a H_1 oracle query was previously issued for the same identity. Then these queries are answered by $\mathcal{F}$ as follows.

– *Queries on oracle* H_1: When an identity ID is submitted to the H_1 oracle, as in Coron's proof technique [13], $\mathcal{F}$ flips a coin $T \in \{0, 1\}$ that yields 0 with probability δ and 1 with probability $1 - \delta$. $\mathcal{F}$ then picks $w \in \mathbf{Z}_q^*$. If $T = 0$ then the hash value $H_1(ID)$ is defined as being $wP \in \mathcal{G}_1$. If $T = 1$, then $\mathcal{F}$ returns $wY \in \mathcal{G}_1$. In both cases, $\mathcal{F}$ inserts a tuple (ID, w, T) in a list L_1 to keep track of the way it answered the query.

– *Key extraction queries*: When Bob requests the private key associated to an identity ID, $\mathcal{F}$ recovers the corresponding (ID, w, T) from L_1 (recall that such a tuple must exist because of the aforementioned assumption). If $T = 1$, then $\mathcal{F}$ outputs "failure" and halts because it is unable to coherently answer the query. Otherwise, it means that $H_1(ID)$ was previously defined to be $wP \in \mathcal{G}_1$ and $wP_{pub} = wX$ is then returned to Bob as a private key associated to ID.

– *Queries on oracle* H_2: When a tuple (ID, m, U) is submitted to the H_2 oracle, $\mathcal{F}$ first scans a list L_2 to check whether H_2 was already defined for that input. If it was, the previously defined value is returned. Otherwise, $\mathcal{F}$ picks a random $v \in \mathbf{Z}_q^*$, stores the tuple (ID, m, U, v) in the list L_2 and returns $H = vP \in \mathcal{G}_1$ as a hash value to Bob.

– *Partial signature queries* O_{VPsig}: When Bob queries the partial signature oracle O_{VPsig} on a message m_i for an identity ID, $\mathcal{F}$ first recovers the previously defined value $Q_{ID} = H_1(ID)$ from L_1. (1) If $Q_{ID} = wP$, $\mathcal{F}$ randomly chooses $v_i \in \mathbf{Z}_q^*$ and sets $U_i = v_i P$ and $V_i' = wP_{pub} + v_i(H + PK)$. (2) If $Q_{ID} = wY$, it chooses numbers $t_i, v_i \in \mathbf{Z}_q^*$ at random, and then sets $V_i' = t_i P_{pub}$, $U_i = v_i P_{pub}$, and defines the hash value $H_2(ID, m_i, U_i)$ as $H = v_i^{-1}(t_i P - Q_{ID}) - PK \in \mathcal{G}_1$ ($\mathcal{F}$ halts and outputs "failure" if H_2 turns out to be already defined for the input (ID, m_i, U_i)). The (U_i, V_i') is returned to Bob and appears as a valid verifiable partial signature from the latter's point of view, since

$$(1)\ e(P, V_i') = e(wP, P_{pub})e(v_i P, H + PK) = e(P_{pub}, Q_{ID})e(U_i, H + PK);$$

$$\begin{aligned}(2)\ e(P_{pub}, Q_{ID})e(U_i, H + PK) &= e(P_{pub}, Q_{ID})e(v_i P_{pub}, v_i^{-1}(t_i P - Q_{ID})) \\ &= e(P_{pub}, Q_{ID})e(P_{pub}, (t_i P - Q_{ID})) \\ &= e(P_{pub}, t_i P) = e(t_i P_{pub}, P) = e(P, V_i').\end{aligned}$$

$\mathcal{F}$ keeps a list of $L_3 = \{(ID, m_i, U_i, V_i', v_i)\}$.

– *Resolution queries* O_{Res}: When Bob queries the resolution oracle O_{Res} on a partial signature (m, U, V') for an identity ID, $\mathcal{F}$ first check its validity and recovers the previously defined value $Q_{ID} = H_1(ID)$ from L_1. If $T = 1$, it halts and outputs "failure". Otherwise, $\mathcal{F}$ looks up the list L_3, finds out v_i and answers

Bob with $V = V' - v_i PK$ if (m, U, V') is in the list, and halts otherwise. Note that, if (U, V') is a valid partial signature and if $V' = v_i P$, then (U, V) is a valid SOK-IBS signature. And since the partial signature scheme (VPsig, VPver) is non-existential forgeable under adaptive chosen message attacks, assuming the CDH problem is hard, the probability that m has not been queried to O_{VPsig} (which means that (m, U, V') is a valid forgery) is negligible, and so is it with $\mathcal{F}$ halts in answering O_{Res}-queries if $T = 0$.

Suppose Bob outputs a fake signature $\tilde{\sigma} = (\tilde{m}, \tilde{U}, \tilde{V})$ for an identity $\tilde{ID}$ eventually. $\mathcal{F}$ then recovers the triple $(\tilde{ID}, \tilde{w}, \tilde{T})$ from L_1. If $\tilde{T} = 0$, then $\mathcal{F}$ outputs "failure" and stops. Otherwise, it goes on and finds out whether $(\tilde{ID}, \tilde{m}, \tilde{U}, \cdot, \cdot)$ appears in the list L_3. Suppose it does not appear in the list L_3, then the list L_2 must contain an entry $(\tilde{ID}, \tilde{m}, \tilde{U}, \tilde{v})$ with overwhelming probability (otherwise, B stops and outputs "failure"). Then, since $\tilde{H} = H_2(\tilde{ID}, \tilde{m}, \tilde{U})$ was defined to be $\tilde{v}P \in \mathcal{G}_1$, if Bob succeeded in the game with the view it was provided with, $\mathcal{F}$ knows that

$$e(P, \tilde{V}) = e(X, Q_{\tilde{ID}}) e(\tilde{U}, \tilde{H})$$

with $\tilde{H} = \tilde{v}P$ and $Q_{\tilde{ID}} = \tilde{w}Y$ for known elements $\tilde{w}$, $\tilde{v} \in \mathbf{Z}_q^*$. Then, it is also known that

$$e(P, \tilde{V} - \tilde{v}\tilde{U}) = e(X, \tilde{w}Y),$$

and thus $\tilde{w}^{-1}(\tilde{V} - \tilde{v}\tilde{U})$ is the solution to the CDH instance $(X, Y) \in \mathcal{G}_1 \times \mathcal{G}_1$.

If $(\tilde{ID}, \tilde{m}, \tilde{U}, \cdot, \cdot)$ does appear in the list L_3, then $(\tilde{ID}, \tilde{m}, \tilde{U}, \cdot)$ most not have been queried to the oracle O_{Res}. $\mathcal{F}$ goes through the list L_3 to find out the $\tilde{v}$, for which $\tilde{U} = \tilde{v}P_{pub} = \tilde{v}X$. Note that $(\tilde{U}, \tilde{V})$ and $(\tilde{U}, \tilde{V}')$ are verifiable partial signature and SOK-IBS signature on $\tilde{m}$ respectively. From (1) and (2) we have

$$e(\tilde{V}' - \tilde{V}, P) = e(\tilde{U}, PK) = e(\tilde{v}X, zY) = e(X, Y)^{\tilde{v}z},$$

and thus $(\tilde{v}z)^{-1}(\tilde{V}' - \tilde{V})$ is the solution to the CDH instance $(X, Y) \in \mathcal{G}_1 \times \mathcal{G}_1$.

Assume that a PPT verifier Bob has an advantage ε in forging a signature in an attack modelled by the game of section 2, when running in a suitable time and asking q_{H_i} queries to random oracles $H_i (i = 1, 2)$, q_E queries to the key extraction oracle, q_S queries to the verifiable partial signature oracle, and q_{Res} queries to the resolution oracle.

When assessing $\mathcal{F}$'s probability of failure, one readily checks that its probability to fail in handling a signing query because of a conflict on H_2 is at most $q_S(q_{H_2} + q_S)/2^k$ (as L_2 never contains more than $q_{H_2} + q_S$ entries) while the probability for Bob to output a valid forgery $(\tilde{U}, \tilde{V})$ on $\tilde{M}$ without asking the corresponding $H_2(\tilde{ID}, \tilde{m}, \tilde{U})$ query is at most $1/2^k$. Finally, by an analysis similar to Coron's one [13], the probability $\delta^{q_E + q_{Res}}(1 - \delta)$ for $\mathcal{F}$ not to fail in a key extraction query or a resolution query or because Bob produces its forgery on a 'bad' identity $\tilde{ID}$ is greater than $1 - 1/e(q_E + q_{Res} + 1)$ when the optimal probability $\delta_{opt} = (q_E + q_{Res})/(q_E + q_{Res} + 1)$ is taken. Eventually, it comes that $\mathcal{F}$'s advantage in solving the CDH problem in $\mathcal{G}_1$ is at least

$$\left(\varepsilon - (q_S(q_{H_2} + q_S) + 1)/2^k\right)/e(q_E + q_{Res} + 1).$$

Security against arbitrator's attack: Now we consider an adversarial TTP's attack. Holding the trapdoor $SK = x$, TTP can extract any U and V' into a pair of (U, V) satisfying (1). We shall also show a reduction of converting an arbitrator's attack into a valid forgery for the SOK-IBS signature scheme. A forger $\mathcal{F}$ accepts ID and PK as input. The TTP holds (PK, SK) and has access to the O_{VPsig}-oracle and the random oracle H_2, and wins if he forges a SOK-IBS signature $\tilde{\sigma} = (\tilde{m}, \tilde{U}, \tilde{V})$, while $\tilde{m}$ has not been queried to O_{VPsig}-oracle.

Here is how $\mathcal{F}$ invokes TTP. For an O_{VPsig}-query on message m, $\mathcal{F}$ chooses $t_i, v_i \in \mathbf{Z}_q^*$ at random, and then sets $V_i' = t_i P_{pub} \in \mathcal{G}_1$, $U_i = v_i P_{pub} \in \mathcal{G}_1$, and defines the hash value $H_2(ID, m_i, U_i)$ as $H = v_i^{-1}(t_i P - Q_{ID}) - PK \in \mathcal{G}_1$ ($\mathcal{F}$ halts and outputs "failure" if H_2 turns out to be already defined for the input (ID, m_i, U_i)). The (U_i, V_i') is returned to Bob as a valid verifiable partial signature. When TTP outputs a forgery $(\tilde{m}, \tilde{\sigma})$ as described above, where $\tilde{m}$ has not been queried to O_{VPsig}, $\mathcal{F}$ just outputs $(\tilde{m}, \sigma)$. We see that the simulation is perfect, and $\mathcal{F}$ succeeds in generating a valid forgery if TTP succeeds.

From a TTP's point of view, a O_{VPsig}-oracle is essentially a SOK-IBS signing oracle, since she holds the trapdoor $SK = x$. Therefore, what TTP is trying to do is to forge a valid SOK-IBS signature under adaptive chosen message attacks. Her advantage is negligible as the SOK-IBS scheme is non-existential forgeable against adaptive chosen-message attacks, under the CDH-assumption.

The above arguments show that, our scheme is provably secure under the well-known CDH assumption, of course, in the random oracle model. □

4 ID-Based Optimistic Fair Exchanges

Now we present an optimistic fair exchange protocol based on the probabilistic signatures described as in section 3. The construction is similar to [8], [14], [21].

Assume Alice's identity is ID_A and the private key is $d_A = sQ_A = sH_1(ID_A)$, and Bob's identity is ID_B and his private key is $d_B = sQ_B = sH_1(ID_B)$. The public key of a TTP is $PK = xP$ while the private key is $SK = x$.

1. Alice first randomly chooses $r_A \in \mathbf{Z}_q^*$ and computes $U_A = r_A P \in \mathcal{G}_1$ and $H_A = H_2(ID_A, m, U_A) \in \mathcal{G}_1$, then computes $V' = d_A + r_A H_A + r_A PK \in \mathcal{G}_1$. Alice sends a verifiable partial signature $\sigma'_{Alice} = (m, U_A, V'_A)$ to Bob.

2. Bob first computes $Q_A = H_1(ID_A) \in \mathcal{G}_1$ and $H_A = H_2(ID_A, m, U_A) \in \mathcal{G}_1$. He then checks

$$e(P, V'_A) = e(P_{pub}, Q_A) \cdot e(U_A, H_A + PK).$$

If it is valid, Bob randomly chooses $r_B \in \mathbf{Z}_q^*$ and computes $U_B = r_B P \in \mathcal{G}_1$ and $H_B = H_2(ID_B, m, U_B) \in \mathcal{G}_1$, and then computes $V'_B = d_B + r_B H_B \in \mathcal{G}_1$. Bob sends his signature $\sigma_{Bob} = (m, U_B, V_B)$ to Alice.

3. After receiving Bob's signature $\sigma_{Bob} = (m, U_B, V_B)$, Alice first computes $Q_B = H_1(ID_B) \in \mathcal{G}_1$ and $H_B = H_2(ID_B, m, U_B) \in \mathcal{G}_1$, and then verifies

$$e(P, V_B) = e(P_{pub}, Q_B) \cdot e(U_B, H_B).$$

If valid, she computes $V_A = V'_A - r_A PK$ and sends $\sigma_{Alice} = (m, U_A, V_A)$ to Bob.

4. If Bob does not receive anything in step 3, or if Alice's signature σ_{Alice} is invalid, then he sends the verifiable partial signature $\sigma'_{Alice} = (m, U_A, V'_A)$ and his probabilistic signature $\sigma_{Bob} = (m, U_B, V_B)$ to TTP. This protocol provides a vehicle for TTP to understand whether the protocol was correctly carried out. TTP first computes $Q_A = H_1(ID_A)$, $H_A = H_2(ID_A, m, U_A)$, and $Q_B = H_1(ID_B)$, $H_B = H_2(ID_B, m, U_B)$, and then checks

$$e(P, V_B) = e(P_{pub}, Q_B) \cdot e(U_B, H_B),$$

and

$$e(P, V'_A) = e(P_{pub}, Q_A) \cdot e(U_A, H_A + PK).$$

If both are valid, TTP extracts $V_A = V'_A - xU_A$, and sends $\sigma_{Alice} = (m, U_A, V_A)$ to Bob and sends $\sigma_{Bob} = (m, U_B, V_B)$ to Alice.

5 Conclusion

We propose an efficient and optimistic fair exchange protocol of identity-based signatures and give a security proof with tight reduction in the random model. Similar to [21], a semi-trust third party (TTP) is still involved in our protocol to ensure fairness, while it is not required to store any information except its private-key. There is no need for registrations between users and TTP, and no zero-knowledge proof is involved. This is the first identity-based optimistic fair exchange protocol with such a concise framework.

Acknowledgement

The work is supported by National Natural Science Foundation of China under Granted No.60373039, and National Grand Fundamental Research Project of China under Granted No.G1999035802.

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. In ASIACRYPT 2004, LNCS 3329, pages 417-429, Springer-Verlag, 2004.
4. F. Bao, G.L. Wang, J.Y. Zhou, H.F. Zhu. Analysis and Improvement of Micali's Fair Contract Signing Protocol, ACISP 2004, LNCS 3108, pp. 176-187, 2004.
5. 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.

6. 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.
7. M. Bellare, C. Namprempre and G. Neven. Security Proofs for Identity-Based Identification and Signature Schemes, Advances in Cryptology-Eurocrypt'04, LNCS 3027, pages 268-286, Springer-Verlag, 2004.
8. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. Advances in Cryptology-EUROCRYPT 2003, LNCS 2656, pages 416-432. Springer-Verlag, 2003.
9. D. Boneh, B. Lynn, H. Shacham. Short signatures from the weil pairing. Advances in Cryptology-ASIACRYPT 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
10. D. Boneh, M. Franklin: Identity-based encryption from the Weil Pairing. In Crypto'2001, LNCS 2139, Springer-Verlag, pages 213-229, 2001.
11. 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, Springer-Verlag, 2000.
12. J.C. Cha, J.H. Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups, Proc. of PKC 2003. Springer-Verlag, LNCS 2567, pp.18-30, Springer, 2003.
13. J.S. Coron. On the exact security of Full Domain Hash. Advances in Cryptology-Crypto 2000, LNCS 1880, pp.229-235, Springer-Verlag, 2000.
14. Y. Dodis and L. Reyzin. Breaking and Repairing Optimistic Fair Exchange from PODC 2003. ACM Workshop on Digital Rights Management, pages 47-54, 2003.
15. B. Libert, J.-J. Quisquater. The Exact Security of an Identity Based Signature and its Applications, IACR Cryptology ePrint Archive, Report 2004/102, 2004.
16. S. Micali. Simple and fast optimistic protocols for fair electronic exchange. 2003 ACM Symposium on Principles of Distributed Computing, pages 12-19, 2003.
17. 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.
18. RSA Labs: RSA Cryptography Standard: EMSAPSS-PKCS♯1 v2.1, 2002.
19. A. Shamir, Identity based cryptosystems and signature schemes, Advances in Cryptology-Crypto'84, LNCS 196, Springer-Verlag, pages 47-53.
20. R. Sakai, K. Ohgishi, M. Kasahara. Cryptosystems based on pairing, In 2000 Sympoium on Cryptography and Information Security, Okinawa, Japan, 2000.
21. Z. F. Zhang, Y. B. Zhou and D. G. Feng. Efficient and Optimistic Fair Exchange based on Standard RSA with Provable Security, IACR Cryptology ePrint Archive, Report 2004/351, 2004.