ID-based Partially Blind Signatures: A Scalable Solution to Multi-Bank E-Cash

Wuping Chen*, Bo Qin†, Qianhong Wu*†, Lei Zhang† and Huanguo Zhang*
*School of Computer, Wuhan University, Wuhan City, China
Email: liss@whu.edu.cn
†Dept. of Comp. Eng. and Maths, Uni. Rovira i Virgili, Tarragona, Spain

Abstract

In this paper, we propose a practical ID-based (partially) blind signature scheme. Our scheme combines the already existing notions of blind signatures and identity-based cryptography. The signatures are shown secure in the random oracle model. We show how to use our ID-based PBSs to construct an electronic cash (e-cash) system in which multiple banks can securely distribute untraceable e-cash.

1. Introduction

In most e-cash systems, only one bank can issue e-cash to users. If the users request e-cash simultaneously, the system may encounter a single-point bottleneck. Hence, this paper focuses on systems allowing multiple banks securely distribute e-cash, and investigates how to realize such systems with ID-based partially blind signatures (PBSs).

1.1. Distributed E-cash Issuing

Consider the following scenario. A country's Central Bank (e.g., the US Treasure or the Federal Reserve) monitors a large group of banks. Each bank is allowed to dispense electronic cash. Usually, the Central Bank is only an administration authority and trusted by civil entities. It will not issue either electronic or paper cash. Such a system should have the following properties:

- Only bank authorized by the Central Bank can issue e-cash. E-cash issued by different banks is distinguishable and can have different face values and expiring dates.
- No bank can issue e-cash on behalf of other banks, i.e., no bank can "frame" another bank.
- No bank can trace any e-cash, no matter whether the e-cash is issued by itself. Therefore, just as with paper money, people can spend their e-cash anonymously.
- A vendor only needs the Central Bank's public key to verify the e-cash issued by each bank. This makes system practical even in a dynamic case where the e-cash issuing banks join and leave (e.g., bankrupted).

In this paper, we propose such a system using ID-based PBSs. Most existing e-cash systems focus on a model with a single bank distributing all the e-cash. In real life, there may more than one bank able to dispense e-cash to avoid the single-point problem. Lysyanskaya *et al* presented a similar scheme using group blind digital signatures [8]. Due to inefficiency of existing group signatures and lack of efficient mechanism to incorporate face value and expiring date into e-cash in their scheme, our solution is more practical to realize multi-bank e-cash.

1.2. Blind Signatures and E-cash

The notion of blind signature was first introduced by Chaum in [4]. In blind signature (BS) schemes a user can ask a signer to blindly sign a (secret) message m. At the end of the (interactive) signing process, the user obtains a valid signature on m, but the signer has no information about the signed message. Chaum remarked that the notion is the only way to implement electronic cash simulating the ones in reality. However, when the original notion of blind signatures is implemented for e-cash systems, as the bank knows nothing about the resulting signature or the signed message, some problems are posed, such as how to prevent misuse of the signature or how to embed the information such as the issuing date, expiration date, face value of e-cash and so on.

To efficiently address such issues in applications, the notion of partially blind signature was introduced by Abe and Fujisaki [1] to allow the signer to explicitly include some pre-agreed information in blind signatures. Using PBSs in e-cash systems, the bank can be relieved from maintaining a unlimitedly growing database. The bank assures that each e-cash contains the information it desires, such as the date information. By embedding an expiration date into each e-cash issued by the bank, all expired e-cash recorded in the bank's database can be removed to reduce storage requirement. At the same time, since face values are embedded into e-cash, the bank knows the value on each e-cash blindly issued.

1.3. Identity-based Signatures

In traditional PKI-based digital signature schemes, certificates generated by a trusted third party are required to 'bind' the user's identity and its public key. In [9], Shamir introduced the new notion of identity-based cryptosystem

978-0-7695-3654-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICSPS.2009.121

IEEE computer society

Authorized licensed use limited to: UNIVERSITAT ROVIRA I VIRGILI. Downloaded on October 29, 2009 at 08:25 from IEEE Xplore. Restrictions apply.

in which the user's public key is indeed its identity such as an email or IP address. In identity-based cryptography, the only secret of each user is its secret identity as a secret key generated by a Key Generation Center (KGC). Hence, in such cryptosystems, the certificates and the intricate management can be avoided.

Recently, Bellare *et al* [2] demonstrated that identity-based signature schemes can be constructed from any PKI-based signature scheme. In [7] at Asiacrypt 2006, following the framework in [2], Galindo *et al* proposed generic constructions of identity-based signature schemes with additional properties such as identity-based blind signatures or identity-based PBSs from PKI-based signature schemes with the same properties. Their scheme requires four moves to generate an identity-based blind signature. For practical reasons such as implementation costs, user-friendliness and etc, more efficient identity-based (partially) blind signature schemes are demanding with a middle level of security.

2. A New Multi-bank E-cash Framework

2.1. ID-based Partially Blind Signatures

By combining the properties of PBSs and ID-based signatures, we obtain the so-called ID-based PBSs. In such signatures, a KGC has a public/secret key pair and generates signing keys for each signer. Then each signer can produce PBSs for requestors.

The first identity-based blind signature (IBBS) schemes were proposed in [12], [13]. They employ bilinear pairings, but their security is not formally analyzed. Subsequent schemes were proposed in [6] but security is only provided in a weaker model (i.e. against sequential adversaries). In [5], the stronger security definition under parallel attacks is proposed and applied to privacy enhanced software registration systems.

We propose an efficient partially blind ID-based signature scheme. By following the security definition of [5], the scheme is shown secure under a slight variation of the chosen-target computation Diffie-Hellman assumption in the random oracle model. Compared with the state-of-the-art generic construction [7] which requires four moves, our schemes require only two moves and the final signatures consist of two group elements and hence are very practical.

2.2. Realizing E-cash with ID-based PBSs

We now show how to use ID-based PBSs to achieve system in which multiple banks can securely distribute anonymous and untraceable e-cash. By inputting the identity of each bank, the country's Central Bank generates private signing key. When a user withdraws e-cash from her bank, she first creates an electronic coin and tell her bank the face value of the coin. Her bank applies an ID-based PBS to the coin with the specified face value and it withdraws the appropriate amount from the user's bank. When the user purchases some services from a vendor, she gives the coin and her bank's signature on the coin to the vendor. If the coin is valid, the vendor gives the user her merchandize, and gives the coin to the vendor's bank. The vendor's bank double checks the coin's validity, adds it to a global list of coins that have already been spent to prevent the user double spending, and credits the vendor's account accordingly. Although the user's bank is signing blindly, it is impossible for the user to cheat her bank by having it sign something other than what it was supposed to sign. This is because the output is a PBS which contains commonly agreed information to specify the signed content must be a coin with agreed face value (and expiring date).

3. Proposed ID-based PBS Scheme

In this section, we present an efficient PBS schemes. Then the scheme is analyzed by following the strong security definition in [5].

3.1. Computational Assumptions

Our schemes are derived from bilinear parings. We shortly review some general concepts of pairing groups. Let $\mathbb{G}$ denote an finite cyclic group of prime order q and $\mathbb{G}_T$ be a multiplicative group of the same order. Let g be a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear mapping with the following properties:

- The map e is bilinear: $e(g^a, h^b) = e(g, h)^{ab}$ for all $g, h \in \mathbb{G}, a, b \in \mathbb{Z}_p$.
- The map e is non-degenerate: $e(g, g) \neq 1 \in \mathbb{G}_T$.
- The map e is efficiently computable.

Similarly to the standard Computational Diffie-Hellman assumption, we introduce a variation referred to as Accompanied Computational Diffie-Hellman (ACDH) assumption.

Definition 1. *(Accompanied Computational Diffie-Hellman (ACDH) assumption) Let $\mathbb{G}$ be a finite cyclic group of prime order p. The Accompanied Computational Diffie-Hellman assumption states that for any PPT algorithm, given a random tuple $(g, g^x, g^y, g^\alpha, g^\beta, g^\gamma) \in \mathbb{G}^6$, the probability to output $(c, g^a, g^b) \in \mathbb{Z}_p \times \mathbb{G}^2$ such that $ab = \gamma + \alpha x + c\beta y$ is negligible.*

Our schemes require the following Chosen-Target Accompanied Computational Diffie-Hellman (CT-ACDH) assumption, which is a variation of Chosen-Target Computational Diffie-Hellman (CT-CDH) assumption [3].

Definition 2. *(Chosen-Target Accompanied Computational Diffie-Hellman (CT-ACDH) assumption) Let $\mathbb{G}$ be a finite cyclic group of prime order p. Let $(g, g^x, g^y, g^\alpha, g^\beta) \in \mathbb{G}^5$. The adversary $\mathcal{A}$ is allowed to access two oracles:*

Authorized licensed use limited to: UNIVERSITAT ROVIRA I VIRGILI. Downloaded on October 29, 2009 at 08:25 from IEEE Xplore. Restrictions apply.

- *Target Oracle $\mathcal{O}_T$, which randomly samples a random element $g^{\gamma_i} \in \mathbb{G}$ whenever it is invoked for the i-th time.*
- *Helper Oracle $\mathcal{O}_H$, which takes input $(c, M) \in \mathbb{Z}_p \times \mathbb{G}$, outputs $((g^{\alpha x + c\beta y} M)^z, g^z, g^{\frac{1}{z}}) \in \mathbb{G}^3$.*

The adversary $\mathcal{A}$ wins if it outputs a common string c' and $k+1$ tuples $(g^{a_1}, g^{b_1}), \cdots, (g^{a_{k+1}}, g^{b_{k+1}})$ such that $a_j b_j = \gamma_{\pi(j)} + \alpha x + c\beta y$, where $\pi(\cdot)$ is a permutation on $\{1, \cdots, q_T\}$, q_T is the number of queries to $\mathcal{O}_T$ and the adversary $\mathcal{A}$ queries the Helper Oracle $\mathcal{O}_H$ with c' at most k times. The CT-ACDH assumption states that for any PPT (probabilistic polynomial-time) adversary, the probability to win the above game is negligible.

3.2. Efficient ID-based Partially Blind Signature

In an ID-based PBS scheme, a requester can ask a signer with identity ID to blindly sign a (secret) message m and a common agreed string c. At the end of the (interactive) signing process, the user obtains a valid signature on m and c, but the signer has no information about the signed message.

- Setup. The Key Generation Center (KGC) generates parameters and master keys as follows:
  1) Generates groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p with bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$;
  2) Choose random generators $g, h \in \mathbb{G}$;
  3) Set $X = g^x, Y = g^y$ for random $x, y \in \mathbb{Z}_p$;
  4) Choose cryptographic hash functions H_1, H_2, $H_3 : \{0,1\}^* \to \mathbb{G}$. The KGC's public key is mpk = ($\mathbb{G}$, $\mathbb{G}_T$, e, p, g, h, X, Y, H_1, H_2, H_3); its master secret is msk = (x, y) $\in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$.
- Key Extract. The signer with identity ID receives the secret key $sk[id] = (u, v) = (H_2(ID)^x, H_3(ID)^y)$.
- Signing. Let the signer and the requester agree with the common information c. This Signing procedure consists of the following sub-procedures.
  1) Blind. The user randomly chooses a number $r \in \mathbb{Z}_p$ as the blinding factor, computes $M = h^r H_1(m)$ and sends it to the signer. Here, h^r can be pre-computed.
  2) BSign. The signer returns $(S_1, S_2, S_3) = ((Muv^c)^z, g^{\frac{1}{z}}, h^z)$ to the requester for a random $z \in \mathbb{Z}_p^*$ and. Here, $g^{\frac{1}{z}}, h^z$ can be pre-computed.
  3) Output. The requester computes $S_1' = S_1 / S_3^r$. The requester randomly selects another blind factor $w \in \mathbb{Z}_p$ and outputs the final signature $s = (s_1, s_2) = ((S_1')^w, S_2^{\frac{1}{w}})$.
- Verify. Given a signature (s_1, s_2) on the message m, the verifier checks that

$$e(s_1, s_2) = e(H_1(m), g) e(H_2(ID), X) e(H_3(ID), Y)^c.$$

If the equation does not hold, the verifier refuses the signature. Else, it accepts it. Here, $e(H_2(ID), X) e(H_3(ID), Y)^c$ can be pre-computed.

The correctness of the above scheme follows from the following fact.

$$\begin{aligned} e(s_1, s_2) &= e((S_1')^w, S_2^{\frac{1}{w}}) = e(S_1', S_2) \\ &= e(S_1 / S_3^r, S_2) = e((Muv^c)^z / h^{zr}, g^{\frac{1}{z}}) \\ &= e((h^r H_1(m) H_2(ID)^x H_3(ID)^{cy})^z / h^{zr}, g^{\frac{1}{z}}) \\ &= e(H_1(m) H_2(ID)^x H_3(ID)^{cy}, g) \\ &= e(H_1(m), g) e(H_2(ID), X) e(H_3(ID), Y)^c. \end{aligned}$$

3.3. Security Analysis

We analyze the security of scheme following the security definition in [5].

Theorem 1. (Blindness) *The above ID-based PBS is unconditionally blind.*

Proof: It is sufficient to prove that, for any view M of the adversary signer $\mathcal{S}^*$ with identity id^* and any signature pair (s_1, s_2, m), there exists a blind factor pair (r, w) that maps the view and the signature pair.

For $i = 0, 1$, let M_i be views of $\mathcal{S}^*$, $(S_{1,i}, S_{2,i}, S_{3,i})$ be the transcripts from $\mathcal{S}^*$ to the user, and $(s_{1,j}, s_{2,j}, m_j)$ be two valid PBSs from user $j = 0, 1$, respectively. Since g is a generator, we can assume that $h = g^\delta, H_1(m_j) = g^{\gamma_j}$ for some values $\delta, \gamma_j \in \mathbb{Z}_p^*$. Notice that both signatures are valid. We obtain a forged blind factor pair (r, w) where $r = \frac{\gamma_i - \gamma_j}{\delta} + r_i$, $w = \log_{S_{1,i}/S_{3,i}^r} s_{1,j} = 1/\log_{S_{2,i}} s_{2,j}$. It can be used to convert $(S_{1,i}, S_{2,i}, S_{3,i})$ into a signature pair $(s_{1,j}, s_{2,j}, m_j)$.

Hence, M_i and $(S_{1,i}, S_{2,i}, S_{3,i})$ for $i \in \{0, 1\}$ have the same relation with $(s_{1,j}, s_{2,j}, m_j, c)$ defined by the signing protocol. Therefore, given a signature message tuple $(s_{1,j}, s_{2,j}, m_j, c)$, an infinitely powerful $\mathcal{S}^*$ can guess j correctly with probability exactly 1/2. □

Let us denote the original unforgeability game by $\mathcal{G}_0$ and the adversary $\mathcal{F}_0$ can win with probability ε_0 in time t_0. We consider an attacker $\mathcal{F}_1$ which can win in a modified unforgeability game $\mathcal{G}_1$ with probability ε_1 in time t_1, where $\mathcal{G}_1$ is the same as $\mathcal{G}_0$ except that the forgery is for a given identity ID^* and the corresponding secret key $sk[ID^*]$ cannot be queried for. We have the following two claims.

Lemma 1. *If there is an adversary $\mathcal{F}_0$ winning game $\mathcal{G}_0$ with probability ε_0 in time t_0, there exists an adversary $\mathcal{F}_1$ winning game $\mathcal{G}_1$ with probability $\varepsilon_1 \geq \varepsilon_0(1 - \frac{1}{p})(1 - \frac{1}{\min\{q_{H_2}, q_{H_3}\}}) \frac{1}{\min\{q_{H_2}, q_{H_3}\}}$ in time $t_1 = t_0$, where q_{H_2}, q_{H_3} are the query numbers to the random oracles H_2 and H_3.*

Proof: We use the adversary $\mathcal{F}_0$ in the game $\mathcal{G}_0$ to construct an adversary $\mathcal{F}_1$ in the game $\mathcal{G}_1$. Let the given identity for $\mathcal{F}_1$ is ID^*. $\mathcal{F}_1$ simulates the challenger in $\mathcal{G}_0$ to answer all the queries from $\mathcal{F}_0$ as follows.

Randomly choose $I \in \{1, \cdots, \min\{q_{H_2}, q_{H_3}\}\}$. Let the identities queried by $\mathcal{F}_0$ be ID_i for $i \in$

Authorized licensed use limited to: UNIVERSITAT ROVIRA I VIRGILI. Downloaded on October 29, 2009 at 08:25 from IEEE Xplore. Restrictions apply.

$\{1, \cdots, \max\{q_{H_2}, q_{H_3}\}\}$. If ID_I is queried to q_{H_2}, q_{H_3}, $\mathcal{F}_0$ replaces it with ID^* to query the challenger in $\mathcal{G}_1$ and forwards the outputs to $\mathcal{F}_0$. If $sk[ID_I]$ is queried for by $\mathcal{F}_0$, the simulation fails. For all the other queries from $\mathcal{F}_0$, $\mathcal{F}_1$ forwards them to the challenger in $\mathcal{G}_1$ and returns the outputs from the challenger in $\mathcal{G}_1$ to $\mathcal{F}_0$. Clearly, the simulation is perfect except a failure probability $\frac{1}{\min\{q_{H_2}, q_{H_3}\}}$.

Finally, if the above game does not fails, $\mathcal{F}_0$ will output a valid forgery regarding identity ID'. Since q_{H_2}, q_{H_3} are modeled as random oracles, the forgery is valid but $\mathcal{F}_0$ has never queried ID' to H_2, H_3 with probability at most $\frac{1}{p}$.

If $ID' = ID_I$, $\mathcal{F}_1$ can directly use this forgery to answer the challenger in $\mathcal{G}_0$. Since I has been independently chosen, $ID' = ID_I$ happens with probability at least $\frac{1}{\min\{q_{H_2}, q_{H_3}\}}$. Hence $\mathcal{F}_1$ wins with probability $\varepsilon_1 \geq \varepsilon_0(1 - \frac{1}{p})(1 - \frac{1}{\min\{q_{H_2}, q_{H_3}\}})\frac{1}{\min\{q_{H_2}, q_{H_3}\}}$ in the same time $t_1 = t_0$. □

The following lemma illustrates that a successful adversary $\mathcal{F}_1$ can be used to break the CT-ACDH assumption in $\mathbb{G}$. Hence, combined with Lemma 1, our ID-based blind signatures are secure under the CT-ACDH assumption.

Lemma 2. *If there is a one-more forger $\mathcal{F}_1$ in the game $\mathcal{G}_1$ with running time t_1 and advantage ε_1, then there exists an adversary $\mathcal{F}_2$ breaking the CT-ACDH assumption in $\mathbb{G}$, which has running time $t_2 \leq O(t_1 + t_{H_1} + t_{H_2} + t_{H_3} + t_S + t_E)$ and advantage $\varepsilon_2 \geq \varepsilon_1$, where t_{H_j} , t_E, t_S are time to simulate the hash functions H_j, and the* Key Extraction *and* Signing *procedures.*

Proof: Suppose that $\mathcal{F}_1$ is an adversary in the game $\mathcal{G}_1$. We describe the algorithm $\mathcal{F}_2$ which will simulate the challenger for $\mathcal{F}_1$ in order to break the CT-ACDH assumption in $\mathbb{G}$. The adversary $\mathcal{F}_2$ is given $(e, \mathbb{G}, \mathbb{G}_T, p, g, g^x, g^y, g^\alpha, g^\beta)$, the target oracle and the helper oracle. $\mathcal{F}_2$ simulates the challenger and interacts with forger $\mathcal{F}_1$ as follows.

Setup. $\mathcal{F}_2$ computes $h = g^\delta$ for a random $\delta \in \mathbb{Z}_p^*$. $\mathcal{F}_2$ randomly selects three hash functions $H_1, H_2, H_3 : \{0, 1\}^* \to \mathbb{G}$ modeled as random oracles, and provides $\mathcal{F}_1$ with the master public key $mpk = (\mathbb{G}, \mathbb{G}_T, e, p, g, h, X, Y, H_1, H_2, H_3)$ and the fixed identity ID^*.

Queries. $\mathcal{F}_1$ is allowed to access to the following oracles.

- H_1-oracle. $\mathcal{F}_1$ is allowed to query the oracle H_1 with its chosen message $m_i \in \{0,1\}^*$. To respond to these queries, $\mathcal{F}_2$ maintains an H_1-list of pairs $(m_i, H_1(m_i))$ which is initially empty. On receiving $\mathcal{F}_1$'s query m_i, if the query m_i appears on the H_1-list in a tuple $(m_i, H_1(m_i))$, then $\mathcal{F}_2$ responds with $H_1(m_i)$. Otherwise, $\mathcal{F}_2$ queries the Target oracle $\mathcal{O}_T$ and obtains the output $g^{\gamma_i} = \mathcal{O}_T(i)$. $\mathcal{F}_2$ sends $H_1(m_i) = g^{\gamma_i}$ to $\mathcal{F}_1$ and adds the pair $(m_i, H_1(m_i))$ to the H_1-list.
- H_2-oracle. $\mathcal{F}_1$ is allowed to query the oracle H_2 with its chosen identity ID_i. To respond to these queries, $\mathcal{F}_2$ maintains an H_2-list of tuples $(ID_i, H_2(ID_i), r_{i,2})$ which is initially empty. If the query ID_i appears on the H_2-list in a tuple $(ID_i, H_2(ID_i), r_{i,2})$ or $(ID_i, H_1(ID_i), *)$, $\mathcal{F}_2$ responds with $H_2(ID_i)$. If $ID_i \neq ID^*$, $\mathcal{F}_2$ sets $H_2(ID_i) = g^{r_{i,2}}$ for a random $r_{i,2}$ and sends it to $\mathcal{F}_1$. $\mathcal{F}_2$ adds the tuple $(ID_i, H_2(ID_i), r_{i,2})$ to the H_2-list. If $ID_i = ID^*$, $\mathcal{F}_2$ sends $H_2(ID_i) = g^\alpha$ to $\mathcal{F}_1$. $\mathcal{F}_2$ appends the tuple $(ID_i, H_2(ID_i), *)$ to the H_2-list.
- H_3-oracle. $\mathcal{F}_1$ is allowed to query the oracle H_2 with its chosen identity ID_i. To respond to these queries, $\mathcal{F}_2$ maintains an H_3-list of tuples $(ID_i, H_3(ID_i), r_{i,3})$ which is initially empty. If the query ID_i appears on the H_3-list in a tuple $(ID_i, H_3(ID_i), r_{i,3})$ or $(ID_i, H_3(ID_i), *)$, then $\mathcal{F}_2$ responds with $H_3(ID_i)$. If $ID_i \neq ID^*$, $\mathcal{F}_2$ sets $H_3(ID_i) = g^{r_{i,3}}$ for a random $r_{i,3}$ and sends it to $\mathcal{F}_1$. $\mathcal{F}_2$ adds the tuple $(ID_i, H_3(ID_i), r_{i,3})$ to the H_3-list. If $ID_i = ID^*$, $\mathcal{F}_2$ sends $H_3(ID_i) = g^\alpha$ to $\mathcal{F}_1$. $\mathcal{F}_2$ appends the tuple $(ID_i, H_3(ID_i), *)$ to the H_3-list.
  Since H_2, H_3 are random oracles, $\mathcal{F}_1$ obtains no information on $H_1(ID)$ before he queries the H_2 and H_3 oracles on ID. Hence, we assume that $\mathcal{F}_1$ has already queried the H_2, H_3 oracle on an identity ID before it makes the Key Extraction query or Signing query with respect to the identity ID.
- Key Extraction oracle. $\mathcal{F}_1$ is allowed to query for the secret key $sk[ID_i]$ with its chosen identity $ID_i \neq ID^*$. $\mathcal{F}_2$ responds with $sk[ID_i] = (X^{r_{i,2}}, Y^{r_{i,3}})$. Note that $X^{r_{i,2}} = H_2(ID_i)^x$ and $Y^{r_{i,3}} = H_3(ID_i)^y$. $sk[ID_i]$ is the secret key with respect to identity $ID_i \neq ID^*$ as the real scheme.
- Signing oracle. Assume that $\mathcal{F}_1$ queries this oracle with (c, ID_i, M_i), where $M_i \in \mathbb{G}$. If the same pair (ID_i, M_i) is queried in different query, $\mathcal{F}_2$ replies with the previous answer. Note that the signer has only one move in the Signing protocol. If $ID_i \neq ID^*$, $\mathcal{F}_2$ works as the real scheme as it knows the secret key $sk[ID_i]$ with its chosen identity ID_i. Else if $ID_i = ID^*$, $\mathcal{F}_2$ queries the Helper Oracle $\mathcal{O}_H$ with (c, M_i) and obtains $\mathcal{O}_H(M_i) = ((g^{\alpha x + c\beta y} M_i)^{z_i}, g^{z_i}, g^{\frac{1}{z_i}}) \in \mathbb{G}^3$. $\mathcal{F}_2$ forwards $((g^{\alpha x + c\beta y} M_i)^{z_i}, g^{\delta z_i}, g^{\frac{1}{z_i}})$ to $\mathcal{F}_1$.

Challenge. Finally, $\mathcal{F}_1$ outputs $\ell_{c,*} + 1$ message-signature pairs $(m'_1, s_{1,1}, s_{2,1}), \cdots, (m'_{\ell_{c,*}+1}, s_{1,\ell_{c,*}+1}, s_{2,\ell_{c,*}+1})$ regarding identity ID^* and common strings c, where $\ell_{c,*}$ is the number of queries to the Signing oracle regarding identity ID^* and common string c. Since $s_{1,i}, s_{2,i}$ satisfies $e(s_{1,i}, s_{2,i}) = e(H_1(m'_i), g)e(H_2(ID^*), X)e(H_3(ID^*), Y)^c = e(g, g)^{\gamma_{\pi(i)} + \alpha x + c\beta y}$ where $H_1(m'_i) = g^{\gamma_{\pi(i)}} \in \{g^{\gamma_j}\}$, it follows that $\{(s_{1,i}, s_{2,i})\}$ for $1 \leq i \leq \ell_{c,*} + 1$ is a solution to the CT-ACDH assumption.

Clearly, the above simulation is perfect without failure and $\mathcal{F}_2$ succeeds provided that $\mathcal{F}_1$ is successful. Therefore, we have that $\mathcal{F}_2$ succeeds with probability $\varepsilon_2 \geq \varepsilon_1$. $\mathcal{F}_2$'s running time is the same as $\mathcal{F}_1$'s running time plus the time t_{H_j} , t_E, t_S to respectively answer the H_j-hash $(j = 1, 2, 3)$,

Authorized licensed use limited to: UNIVERSITAT ROVIRA I VIRGILI. Downloaded on October 29, 2009 at 08:25 from IEEE Xplore. Restrictions apply.

Key Extraction and Signing queries. □

From the above lemmas, we obtain the following claim.

Theorem 2. *The proposed ID-based (partially) blind signature scheme are unforgeable in the random oracle model assuming the CT-ACDH assumption.*

3.4. Performance Comparison

We compare our schemes with the state-of-the-art proposals in the following table, where λ is the bitlength of elements in $\mathbb{G}$, Exp represents a multi-exponentiation or a single exponentiation and P represents a multi-pairing or a single pairing. We do not discriminate a multi-pairing from a single pairing (or a multi-exponentiation from a single exponentiation) as they require similar overhead. The ROM is the random oracle model and ROS represents the ROS assumption introduced in [10].

Table 1. Comparison of state-of-the-art ID-based blind signatures.

	Moves	Size	Sign	Request	Verify	Security
[12]	3	2λ	2Exp	2Exp+1P	1Exp+1P	ROM+ROS
[13]	3	2λ	2Exp	3Exp	1Exp+1P	ROM+ROS
[6]	3	2λ	3Exp	4Exp	1Exp+1P	ROM+ROS
Ours	2	2λ	3Exp	2Exp	1Exp+1P	ROM+CT-ACDH

The existing schemes [6], [12], [13] are derived from the blind Schnorr signatures relying on the ROS assumption [10]. The analysis in [11] shows that it only requires subexponential running time $O(2^{2\sqrt{\log p}})$ to break the ROS assumption with the generalized birthday attack. Hence, to achieve a security level of 2^{80}, λ has to be more than 1600 or $p > 2^{1600}$ for the existing schemes [6], [12], [13], while for our schemes, the security level can be achieved with $\lambda = 170$. For pairings of an order $p > 2^{1600}$, the computation is rather inefficient and the schemes relies on ROS are impractical. For round efficiency, our schemes are optimal since any blind signature require at least two moves, i.e., the requester submits a blinded message and the signer returns a string such that the requester can extract the final signature.

4. Conclusion

In this paper, we showed how to realize secure multi-bank e-cash systems with ID-based partially blind signatures. Then we proposed an efficient ID-based partially blind signature scheme. The scheme was shown secure under a variation of the chosen-target computational Diffie-Hellman assumption in the random oracle model. Compared with the state-of-the-art existing schemes and the recent generic construction at Asiacrypt 2006 which requires four moves, our proposal is efficient and practical.

Acknowledgment and Disclaimer

This paper is partly supported by the Chinese NSF project 60673071 and by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES" and TSI2007-65406-C03-01 "E-AEGIS". The views of those authors with the UNESCO Chair in Data Privacy do not necessarily reflect the position of UNESCO nor commit that organization.

References

[1] M. Abe, E. Fujisaki. How to Date Blind Signatures. Asiacrypt'96, LNCS 1163, pp. 244-251. Springer-Verlag, 1996.

[2] M. Bellare, C. Namprempre and G. Neven. Security Proofs for Identity-based Identification and Signature Schemes. Eurocrypt'04, LNCS 3027, pp. 268-286. Springer-Verlag, 2004.

[3] A. Boldyreva, Efficient Threshold Signature, Multisignature and Blind Signature Schemes Based on the Gap-Diffie-Hellman-group Signature Scheme. PKC'03, LNCS 2139, pp.31-46. Springer-Verlag, 2003.

[4] D. Chaum. Blind Signatures for Untraceable Payments. Crypto'82, pp. 199-204. Prenum Publishing Corporation, 1982.

[5] W. Chen, B. Qin, Q. Wu and H. Zhang. Efficient Privacy Enhanced Software Registration with ID-based Blind Signatures. Wuhan University Journal of Natural Sciences. Vol. 13, No. 6, pp. 733-738, 2008.

[6] S. S. M. Chow, L. C.K. Hui, S. M. Yiu, and K. P. Chow. Two Improved Partially Blind Signature Schemes from Bilinear Pairings. ACISP'05, LNCS 3574, pp. 316-328. Springer-Verlag, 2005.

[7] D. Galindo1, J. Herranz, and E. Kiltz. On the Generic Construction of Identity-Based Signatures with Additional Properties. Asiacrypt'06, LNCS 4284, pp. 178-193. Springer-Verlag, 2006.

[8] A. Lysyanskaya, Z. Ramzan. Group Blind Digital Signatures: A Scalable Solution to Electronic Cash. FC'98, LNCS 1465, pp. 184-197. Springer-Verlag, 1998.

[9] A. Shamir. Identity-based Cryptosystems and Signature Schemes. Crypto'84, LNCS 196, pp. 47-53. Springer-Verlag, 1985.

[10] C. Schnorr. Security of Blind Discrete Log Signatures against Interactive Attacks. ICICS'01, LNCS 2229, pp. 1-12. Springer-verlag, 2001.

[11] D. Wagner. A Generalized Birthday Problem. Crypto'02, LNCS 2442, pp. 288-304, 2002. Springer-Verlag, 2002.

[12] F. Zhang and K. Kim. ID-based Blind Signature and Ring Signature from Pairings. Asiacrypt'02, LNCS 2501, pp. 533-547. Springer-Verlag, 2002.

[13] F. Zhang and K. Kim. Efficient ID-based blind Signature and Proxy Signature from Bilinear Pairings. ACISP'03, pp. 312-323. Springer-Verlag, 2003.

Authorized licensed use limited to: UNIVERSITAT ROVIRA I VIRGILI. Downloaded on October 29, 2009 at 08:25 from IEEE Xplore. Restrictions apply.