Improved ID-based Ring Signature Scheme with Constant-size Signatures

Hongwei Li, Xiao Li and Mingxing He
School of Mathematics and Computer Engineering
Xihua University
E-mail: lhwlihongwei@gmail.com

Shengke Zeng
School of Computer Science and Engineering
University of Electronic Science and Technology of China
E-mail: doris82414@sina.com

Keywords: ring signature, accumulator, constant-size, random oracle

Received: September 7, 2010

Ring signature enable a user to sign a message on behalf of the ring, without revealing the actual signer. Constant-size ring signature is the ring scheme that the size of the signature does not grow with the size of the ring(or group), so it is practical for large rings. In this paper we use the Collision Resistant Accumulator from bilinear pairing to construct an identity-based ring signature scheme with constant-size signature. Our scheme actually is an improvement on the modified version of the scheme proposed by Nguyen, but we greatly improved the efficiency in terms of computational complexity and signature size. To the best of our knowledge, our scheme is the most efficient secure ID-based ring signature with constant-size based on accumulator proposed to date. Our scheme is proven secure in the random oracle model based on a simplified and general Forking Lemma under the k-strong Diffie-Hellman assumption.

Povzetek: Predstavljena je izboljšana metoda podpisa za obroč.

1 Introduction

Ring signature schemes, introduced by Rivest, Shamir and Tauman [1], allow a signer to form a group without a central authority and sign messages on behalf of the group. A user might not even know that he has been included in a group and even a party with unlimited computing resources can not find out the actual signer. In order to remove the need of certification of the public keys, Shamir [2] proposed the concept of ID-based cryptology to simplify public key management. Zhang and Kim [3] extended the ring signature to the ID-based ring signature schemes, where the user's public keys is their identities. The accumulator was introduced by Benaloh and de Mare [4] in order to design distributed protocols without the presence of a trusted central authority. Such a cryptographic primitive is an algorithm allowing the aggregation of a large set of elements into a single value of constant size. So the accumulator could be applied to construct constant size ring signature. Barić and Pfitzmann [5] generalized the definition of accumulators and constructed a collision-free subtype. As an application, they construct a fail-stop signature scheme based on their collision-free accumulator. Camenisch and Lysyanskaya [6] extended the concept of accumulators to dynamic accumulators which allow the addition and deletion of values from the original set of elements. Dodis, Kiayias, Nicolosi and Shoup [7] introduced ad hoc anonymous identification schemes based on the notion of accumulator with one-way domain, an extension of cryptographic accumulators. In 2005, Nguyen [8] proposed a dynamic accumulator based on bilinear pairings to design ID-based ad-hoc anonymous identification schemes and identity escrow protocols with membership revocation. However, Tartary, Zhou, Lin, Wang and Pieprzyk [9] demonstrated that the security model proposed by Nguyen did lead to a cryptographic accumulator which is not collision resistant. later, Nguyen had modified the security model [10] so that collision resistance can be provided. In 2009, Camenisch, Kohlweiss and Soriente proposed a new dynamic accumulator [24] based on bilinear maps for revocation of the authentication credentials. In their construction, however, in the case of accumulating an arbitrary set of size n, the issuer of the accumulator would need to publish a mapping from the set of identities to the elements of destined group. It looks like very difficult to construct ring signature schemes by hiring their accumulator [24].

Since Zhang and Kim [3] proposed the first ID-based ring signature scheme, there are lots of excellent Id-based ring signature schemes have been proposed [11, 12, 13], but all of them the size of ring signatures linearly depend on the group size, thus not practical for large groups. Actually, all the previous proposals had signature size proportional to the size of the ring before the scheme [7] proposed by Dodis, Kiayias, Nicolosi and Shoup. They

provided an ad-hoc anonymous identification scheme and used the Fiat-Shamir heuristics [15] to convert it into the public key which was prime, so an extension supporting ID-based keys seemed to be non-trivial [14]. The first ID-based ring signature scheme with constant-size signatures [8] was proposed by Nguyen. Similar to scheme [7], Nguyen also obtained the constant-size ring signature using the Fiat-Shamir transform [15] from anonymous identification scheme. However, the scheme [8] was found flawed by Zhang and Chen [16]. After that, Nguyen proposed the modified version [10] of the original scheme [8] and shown that the ring signature in their scheme [10] is much more efficient than previous one [7]. So, is there still some room for the computational efficiency of the constant-size ring signature to improve? Is there a way to reduce the size of the constant-size ring signature scheme?

We provide the affirmative answer to these questions and deem that the constant-size ring signature scheme [10] is still not efficient enough. We propose the improved constant-size ring signature scheme which is much more efficient either on computational complexity or signature size than the scheme [10] proposed by Nguyen. Moreover, Nguyen doesn't directly give the security reduction of their ring signature scheme, but we provide the security proof in the random oracle model under the k-strong Diffie-Hellman assumption. To the best of our knowledge, our scheme is the most efficient ID-based constant-size ring signature based on accumulator in the literature.

The rest of this paper is organized as follows. In Section 2, we briefly review some notations and complexity assumptions that will be used throughout this paper. We explain the general characteristics of a ID-based constant-size ring signature scheme, and the security properties that such a scheme must satisfy in Section 3. In Section 4, We present our new ring scheme, and provide security results for it in the section 5. In section 6, we compare its efficiency with previous schemes. Finally, we sum up the work in Section 7.

2 Preliminaries

In this section, we briefly introduce some preliminaries that will be used throughout this paper. A string means a binary one. If $x_1, x_2, \ldots$ are objects, then $x_1||x_2||\ldots$ denotes an encoding of them as strings from which the constituent objects are easily recoverable. If S is a set, $s \in_R S$ denotes the operation of assigning to s an element of S chosen at random. $s \leftarrow s^{'}$ means we let $s = s^{'}$. If A is a randomized algorithm, then $A(x_1, \ldots; \rho)$ denotes its output on inputs $x_1, \ldots$ and ρ, while $y \leftarrow_R A(x_1, \ldots; \rho)$ means that we choose ρ at random and let $y = A(x_1, \ldots; \rho)$.

2.1 Bilinear map groups and related computational problems [25]

Let l be a security parameter and p be a l-bit prime. Let us consider groups G_1, G_2 and G_T of the same prime order p and let P, Q be the generators of G_1 and G_2 respectively. We say that (G_1, G_2, G_T) are bilinear map groups if there exists a bilinear map $e : G_1 \times G_2 \rightarrow G_T$ satisfying the following properties:

- Bilinearity: $\forall (S,T) \in G_1 \times G_2, \forall a, b \in \mathbb{Z}_p^*, e(aS, bT) = e(S,T)^{ab}$.
- Non-degeneracy: $\forall S \in G_1, e(S,T) = 1$ for all $T \in G_2$ iff $S = \mathcal{O}$.
- Computability: $\forall (S,T) \in G_1 \times G_2, e(S,T)$ is efficiently computable.
- There exits an efficient, publicly computable (but not necessarily invertible) isomorphism $\psi : G_2 \rightarrow G_1$ such that $\psi(Q) = P$

Such bilinear map groups are known to be instantiate with ordinary elliptic curves such as that suggested in [21]. In this case, the trace map can be used as an efficient isomorphic ψ as long as G_2 is properly chosen [22]. With supersingular curves, symmetric pairings(i.e.$G_1 = G_1$) can be obtained and ψ is the identity. The computational assumption for the security of our scheme was proposed by Boneh and Boyen [27] and is recalled in the following definition.

Definition 1. *Let us consider the bilinear map groups (G_1, G_2, G_T) and the generators $P \in G_1$ and $Q \in G_2$. The k-strong Diffie-Hellman problem in the groups G_1, G_2 is defined as follows: given a (k+2)-tuple $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^k Q)$ as input, where $P = \psi(Q)$, output a pair $(c, \frac{1}{c+\alpha} P)$ with $c \in Z_p^*$.*

2.2 Collision Resistant Accumulator

Here we present the definition of accumulators and the collision resistance property as set by Nguyen [10].

Definition 2. ***(Accumulator[10])*** *Accumulator is a tuple $(\{X_l\}_{l \in N}, \{F_l\}_{l \in N})$, where $\{X_l\}_{l \in N}$ is called the value domain of the accumulator and $\{F_l\}_{l \in N}$ is a sequence of families of pairs of functions such that each $(f, g) \in F_l$ is defined as $f : U_f \times X_f^{ext} \rightarrow U_f$ for some $X_1 \subseteq X_f^{ext}$, and $g : U_f \rightarrow U_g$ is a bijective function. In addition, the following properties are satisfied:*

- *(efficient generation) There exists an efficient algorithm $\mathcal{G}$ that takes as input a security parameter 1^l and outputs a random element $(f, g) \in_R F_l$, possibly together with some auxiliary information a_f.*
- *(quasi commutativity) For every $l \in N$, $(f, g) \in F_l$, $u \in U_f$, $x_1, x_2 \in X_l$, $f(f(u, x_1), x_2) = f(f(u, x_2), x_1)$. For any $l \in N$, $(f, g) \in F_l$, and $X = \{x_l, ..., x_k\} \subset X_l$, we call $g(f(...f(u, x_1)..., x_k))$ the*

accumulated value of the set X over u. Due to quasi commutativity, the value $g(f(...f(u,x_1)...,x_k))$ is independent of the order of the x_i's and is denoted by $f(u,X)$.

- *(efficient evaluation) For every $(f,g) \in F_l$, $u \in U_f$ and $X \subset X_l$ with size bound by a polynomial of l : $g(f(u,X))$ is computable in time polynomial in l, even without the knowledge of a_f .*

Definition 3. *(Collision Resistant Accumulator [9] [10]). An accumulator is defined as collision resistant if for every PPT algorithm A, the following function $Adv_A^{col.acc}(l) = Pr[(f,g) \in_R F_l; u \in_R U_f; (x,w,X) \leftarrow A(g \circ f, U_f, u) | (X \subset X_l) \wedge (w \in U_g) \wedge (x \in X_f^{ext} \setminus X) \wedge (f(g^{-1}(w),x) = f(u,X))]$ is negligible as a function of l. We say that w is a witness for the fact that $x \in X_l$ has been accumulated in $v \in U_g$ whenever $g(f(g^{-1}(w),x) = v$.*

To generate an instance of the accumulator [10] from the security parameter l, run the algorithm $\mathcal{G}$ with 1^l to obtain a tuple $t = (p, G_1, G_2, G_T, e(\cdot,\cdot), P, Q)$ and a uniformly chosen element s from Z_p^*. We construct a tuple $t^{'} = (P, Q, sQ, \ldots, s^qQ)$ where q is the an upper bound on the number of elements to be accumulated. The corresponding functions (f,g) for this instance $(t, t^{'})$ are defined as:

$$f : Z_p \times Z_p \to Z_p \qquad g : Z_p \to G_2$$
$$(v,x) \mapsto (x+s)v \qquad v \mapsto vQ$$

This construction involves that we have:

$$U_f = X_f^{ext} = Z_p \quad U_g = G_2 \quad X_l = Z_p \setminus \{-s\}$$

It is clear that f is quasi-commutative. In addition, for $u \in Z_p$ and a set $X = \{x_1, ..., x_k\} \subseteq Z_p \setminus \{-s\}$ where $k \leq p$, the accumulated value $g(f(u,X)) = \prod_{i=1}^{k}(x_i + s)uQ$ is computable in time polynomial in l from the tuple $t^{'}$ and without the knowledge of the auxiliary information s. The accumulator proposed by Nguyen [10] has been proven secure by [9] under the k-strong Diffie-Hellman assumption.

2.3 New General Forking Lemma

The security proof of our ID-based constant-size Ring Signature scheme relies on a generalization of the Forking Lemma [18] proposed by Bellare and Neven instead of the Forking Lemma in ring scenario [20] proposed by Herranz and Sáez.

Lemma 1. *(General Forking Lemma [18] [23]). Fix an integer $Q \geq 1$ and a set H of size $\mid H \mid \geq 2$. Let B be a randomized algorithm that on input $x, h_1, \ldots, h_Q$ returns a pair (J,σ) where $J \in \{0, \ldots, Q\}$ and σ is referred as side output. Let IG be a randomized algorithm called the input generator. Let $acc_B = Pr[J \geq 1 : x \leftarrow_R IG, h_1, \ldots, h_Q \leftarrow_R H; (J,\sigma) \leftarrow_R B(x, h_1, \ldots, h_Q)]$ be the accepting probability of B. The forking algorithm F_B associated to B is the randomized algorithm that takes in input x and proceeds as follows:*

Algorithm $F_B(x)$
Pick random coins ρ for B
$h_1, \ldots, h_Q \leftarrow_R H$
$(J,\sigma) \leftarrow B(x, h_1, \ldots, h_Q; \rho)$
If $J = 0$ then return $(0, \perp, \perp)$
$h_1^{'}, \ldots, h_Q^{'} \leftarrow_R H$
$(J^{'}, \sigma^{'}) \leftarrow B(x, h_1, \ldots, h_{J-1}, h_J^{'}, \ldots, h_Q^{'}; \rho)$
If $(J = J^{'}$ and $h_J \neq h_J^{'})$ then return $(1, \sigma, \sigma^{'})$
Else return $(0, \perp, \perp)$.

Let $frk = Pr[b = 1 : x \leftarrow_R IG; (b, \sigma, \sigma^{'} \leftarrow_R F_B(x)]$. Then $frk \geq acc_B(\frac{acc_B}{Q} - \frac{1}{|H|})$.

The exactly proof of this lemma could be found in [18]. Roughly says that if an algorithm B accepts with some non-negligible probability, then a "rewind" of B is likely to accept roughly with the probability squared[23]. The intuitions are that: $(1) h_1, \ldots, h_Q$ can be seen as the set of replies to random oracle queries made by the original adversary and (2) the forking algorithm implements the rewinding. Moreover, it is important that in F_B the two executions of B are run with the same random coins.

3 The Model of ID-based Constant-size Ring Signature

3.1 ID-based Constant-size Ring Signature Schemes

Here we give the definition of ID-based constant-size ring signature schemes, which is quite the same as the definition in [10].

Definition 4. *An ID-based constant-size ring signature scheme is as a tuple IR =(Setup, KeyGen, MakeGPK, MakeGSK, Sign, Verify) of PT algorithms, which are described as follows.*

- ***Setup:** takes as input a security parameter 1^l and returns the public parameters $params$ and a master key mk. The master key is only known to the Private Key Generator (PKG).*
- ***KeyGen:** run by the PKG, takes as input $params$, mk and an arbitrary identity id_i and outputs a private key s_{id_i}. The identity is used as the corresponding public key.*
- ***MakeGPK:** takes as input $params$ and a set of identities and deterministically outputs a single group public key gpk which is used in the ID-based ring signature scheme described below. Its cost linearly depends on the number of identities being aggregated. The algorithm is order invariant that means the order of aggregating the identities does not matter.*

- ***MakeGSK:*** *takes as input $params$, a set of identities R, a pair of an identity id_i and the corresponding private key s_{id_i} and deterministically outputs a single group secret key gsk_{id_i} which is used in the ID-based ring signature scheme described below. It should be noted that each user has its own group secret key gsk_{id_i} which is different from the others. Its cost linearly depends on the number of identities being aggregated. It can be observed that a group secret key $gsk_{id_i} \leftarrow MakeGSK(params, S^{'}, (s_{id_i}, id_i))$ corresponds to a group public key $gpk \leftarrow MakeGPK(params, S)$ if and only if $S = S^{'} \cup id_i$. More than one group secret key might correspond to the same group public key.*

- ***Sign:*** *takes as input the public parameter $params$, a user private key s_{id_i}, the user's group secret key gsk_{id_i}, group public key gpk which includes the identity corresponding to id_i, and a message m, outputs a signature σ for m.*

- ***Verify:*** *The deterministic polynomial time(DPT) algorithm takes as input a set of identities R, group public key gpk, a message m and a ring signature σ, and outputs either $accept$ or $reject$.*

3.2 Security Requirements

There are two preliminary security requirements for ID-based ring signature schemes: Anonymity and Unforgeability.

- **Anonymity:** the anonymity requires, informally, that an adversary should not be able to tell which member of a ring generated the particular signature.

- **Unforgeability:** the intuitive notion of unforgeability is that an adversary should be unable to output (m, σ) such that Verify$(m, \sigma) = 1$. However, there are lots of security definitions about unforgeability of ring signature [20]. We use the unforgeability definition [26] proposed by Herranz. It should be noted that [26]'s unforgeability definition is very similar to the strongest unforgeability definition(unforgeability w.r.t. insider corruption) proposed in [20].

  Definition 5. *(Unforgeability against chosen messages/identies attacks). A ring signature scheme (Setup, KeyGen, MakeGPK, MakeGSK, Sign, Verify) is unforgeable with chosen-subring attacks if for any PPT adversary A and for any polynomial $n(\cdot)$, the probability that A succeeds in the following game is negligible:*

  - *the challenger takes a security parameter k and runs the Setup algorithm of the scheme. He gives the resulting parameter to adversary.*
  - *A is given access to a signing oracle $OSign(\cdot, \cdot, \cdot)$, $OSign(id_s, m, R)$ returns $Sign_{s_{id_s}}(m, R)$, where we require $id_s \in R$, where R is a set of identities.*
  - *A is also given access to extraction oracle $Extraction(\cdot)$, where $Extraction(ID_i)$ outputs corresponding secret key sk_i.*
  - *at the end of the above execution, A outputs (R^*, m^*, σ^*) and succeeds if $Verify_{R^*}(m^*, \sigma^*) = 1$, A never queried $(R^*, m^*, \cdot)$ to its signing oracle, and for all $ID_i \in R$, the adversary has not requested an extraction query for ID_i.*

4 The Proposed Constant-size Ring Signature Scheme

In this section, we present our ID-based constant-size ring signature scheme, Our scheme is the modified version of the scheme [10] proposed by Nguyen, we describe our scheme as the following algorithms: Setup, KeyGen, MakeGPK, MakeGSK, Sign, Verify.

- **Setup:** on a security parameter l, chooses $s \in_R Z_p^*$, $u \in_R Z_p^*$ and generates an collision resistant accumulator as in section 2.2, including functions (f, g) and tuples $t \leftarrow (p, G_1, G_2, G_T, e(\cdot, \cdot), \psi)$ and $t^{'} \leftarrow (P, Q, sQ, \ldots, s^qQ)$, where q is the upper bound on the number of identities to be aggregated. It sets $Q_{pub} = sQ, P_{pub} = \psi(Q_{pub})$. Let H_0, H_1 be collision-free hash function $H_0 : \{0,1\}^* \rightarrow Z_p^*$, $H_1 : \{0,1\}^* \rightarrow Z_p^*$. Then, public parameter $params \leftarrow (l, t, t^{'}, u, H_0, H_1, f \circ g)$ and the master key is $mk \leftarrow s$.

- **KeyGen:** extracts a private key $s_{id_i} \leftarrow \frac{1}{H_0(id_i)+s}P$ for an identity id_i. The identity is used as the corresponding public key. The user can verify the private key by checking $e(H_0(id_i)Q + Q_{pub}, s_{id_i}) \stackrel{?}{=} e(Q, P)$.

- **MakeGPK:** given a set of identities $R = \{id_i\}_{i=1}^k$, computes the set X = $\{H_0(id_i)\}_{i=1}^k$ and generates the group public key for the set $gpk = V \leftarrow g(f(u, X))$.

- **MakeGSK:** generates the group secret key gsk for a user $id_s \in R$, $R = \{id_i\}_{i=1}^k$ by just computing the set $X^{'} \leftarrow \{H_0(id_i)\}_{i=1, i \neq s}^k, h_{id_s} \leftarrow H_0(id_s)$ and the witness $W \leftarrow g(f(u, X^{'}))$. Note that $X = X^{'} \cup h_{id_s}$. The group secret key is $gsk = (h_{id_s}, s_{id_s}, W)$.

- **Sign:** given a message m, a set of identities $R = \{id_i\}_{i=1}^k$ which includes the signer's identity id_s, the signer's private key s_{id_s}.

  - Given a message $m \in_R \{0,1\}^*$, choose $r_1, r_2, k_1, k_2, k_3, k_4, k_5 \in_R Z_p^*$.

- Compute $U_1 \leftarrow s_{id_s} + r_1 P$; $U_2 \leftarrow W + r_2 Q$; then compute $\prod_1 \leftarrow e(Q, U_1)^{-k_5} \cdot e(Q, P)^{k_2} \cdot e(Q_{pub}, P)^{k_1}$; $\prod_2 \leftarrow e(P, U_2)^{-k_5} \cdot e(P, Q)^{k_4} \cdot e(P_{pub}, Q)^{k_3}$.
- Get $c \leftarrow H_1(m||U_1||U_2||\prod_1||\prod_2||R)$.
- Then compute $s_1 \leftarrow k_1 + cr_1$; $s_2 \leftarrow k_2 + cr_1 h_{id_s}$; $s_3 \leftarrow k_3 + cr_2$; $s_4 \leftarrow k_4 + cr_2 h_{id_s}$; $s_5 \leftarrow k_5 + ch_{id_s}$.
- The signature is $\sigma = (U_1, U_2, \prod_1, \prod_2, s_1, s_2, s_3, s_4, s_5)$.

– **Verify:** Given signature σ, message m, a set of identities $R = \{id_i\}_{i=1}^{k}$.

- Get $c' \leftarrow H_1(m||U_1||U_2||\prod_1||\prod_2||R)$.
- Check $\prod_1 \stackrel{?}{=} e(Q, U_1)^{-s_5} \cdot e(Q, P)^{s_2} \cdot e(Q_{pub}, P)^{s_1} \cdot e(Q_{pub}, U_1)^{-c'} \cdot e(P, Q)^{c'}$; $\prod_2 \stackrel{?}{=} e(P, U_2)^{-s_5} \cdot e(P, Q)^{s_4} \cdot e(P_{pub}, Q)^{s_3} \cdot e(P_{pub}, U_2)^{-c'} \cdot e(P, V)^{c'}$.

If above condition holds, the verifier accept the ring signature, else reject. It's easy to see that our ring signature scheme can be converted into an ad-hoc anonymous identification scheme [7], where a user can form ad-hoc groups and anonymously prove membership in such group. Although we use the collision-free accumulator in our scheme, the dynamic accumulators are also available such that the addition and deletion of members from the original group(or ring) are allowed.

It should be noted that Nguyen' scheme and our ring signature scheme actually are both non-interactive proof of the knowledge of (s_{id}, h_{id}, W) satisfying $e(h_{id}Q + Q_{pub}, S_{id}) = e(Q, P)$ and $e(h_{id}P + P_{pub}, W) = e(P, V)$, although Nguyen' scheme is much more complex than ours. The exact reason why our scheme could cut down the computation and size of the signature will be given in section 6.

5 Security Analysis

5.1 Unforgeability

theorem 1. *Assume that an adversary F has an advantage ε against our scheme when running in time t, asking q_{H_i} queries to random oracles $H_i (i = 0, 1)$, q_s signature queries to signature oracle. Then there is an adversary B to solve the k-strong Diffie-Hellman problem with probability $\varepsilon' \geq \frac{\varepsilon^2}{q_{H_1}^2} + \frac{(q_{H_0}+q_{H_1}+q_s)^2 - 4\varepsilon(q_{H_0}+q_{H_1}+q_s)}{q_{H_1}^2 \cdot 2^{l+1}} - \frac{1}{2^l}$ within a time $t' \leq 2t + q_s[10t_{exp} + 9t_p + (n+1)q_{mult}] + \mathcal{O}[(q_s(n+1)+q_H)(1+q_s+q_H)]$, where t_p denotes the require time for a paring evaluation, t_{exp} denotes the costs of an exponentiation in G_T, t_{mult} denotes the costs of a multiplication in G_2 and q_H denotes the maximum total number of queries to all random oracles.*

Proof: Here, we are ready to present the actual proof. On a security parameter l, Algorithm B takes as input $(p, G_1, G_2, G_T, \psi, P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^k Q)$ and aims to find a pair $(w^*, \frac{1}{w^*+\alpha} P)$ where $w^* \in Z_p^*$. We first show how to provide the adversary with a consistent view. In setup phase, it builds a generator $G \in G_1$ such that it knows $k-q (k > q)$ pairs $(w_i, \frac{1}{w_i+\alpha} G)$ for $w_1, \ldots, w_{k-q} \in_R Z_p^*$. It should be noted that q is the an upper bound on the number of elements to be accumulated and $k-q$ is the an upper bound on the number of extraction queries. To do so,

– It picks $w_1, \ldots, w_{k-q} \in_R Z_p^*$ and expands $f(z) \leftarrow \prod_{i=1}^{k-q}(w_i + z)$ to obtain $e_0, \ldots, e_{k-q} \in_R Z_p^*$ so that $f(z) \leftarrow \sum_{i=0}^{k-q} e_i z^i$.

– It sets generators $H \leftarrow \sum_{i=0}^{k-q} e_i(\alpha^i Q) = f(\alpha)Q \in G_2$ and $G \leftarrow \psi(H) = f(\alpha)P \in G_1$.

– For $1 \leq i \leq k-q$, B expands $f_i(z) \leftarrow \frac{f(z)}{z+w_i}$ to obtain $d_{i_0}, \ldots, d_{i_{k-q-1}} \in_R Z_p^*$ so that $f_i(z) = \sum_{j=0}^{k-q-1} d_{i_j} z^j$. Then compute $\sum_{j=0}^{k-q-1} d_{i_j} \psi(\alpha^j Q) = f_i(\alpha)P = \frac{f(\alpha)}{\alpha+w_i} P = \frac{1}{\alpha+w_i} G$. Then all pairs $(w_i, \frac{G}{\alpha+w_i})$ for $1 \leq i \leq k-q$ could be available.

– It sets $\alpha H \leftarrow \sum_{i=0}^{k-q} e_i(\alpha^{i+1}Q), \alpha^2 H \leftarrow \sum_{i=0}^{k-q} e_i(\alpha^{i+2}Q), \ldots, \alpha^q H \leftarrow \sum_{i=0}^{k-q} e_i(\alpha^{i+q}Q)$. let $t = \{p, G_1, G_2, G_T, e(\cdot,\cdot), \psi, G, H, \alpha H, \ldots, \alpha^q H\}$.

Now, B first chooses a random $u \in_R Z_p^*$ and generates an collision resistant accumulator $f \circ g$ as in section 2.2, then send $(l, t, u, f \circ g)$ to adversary F. To handle the oracle queries, B maintains two lists L_{H_0} and L_{H_1}. For simplicity, we assume that adversary F asks q_{H_0} distinct queries for q_{H_0} distinct identities. Simulates adversary's environment as follows:

– H_0 queries on an identity $ID \in \{0,1\}^*$: B selects a random index γ, where $1 \leq \gamma \leq q_{H_0}$ and fixes ID_γ as target identity. B first initializes a counter $index$ to 1 and answers $w \leftarrow w_{index} \in Z_p^*$ and increments $index$ if $ID \neq ID_\gamma$, else B returns a random $w_\gamma \in_R Z_p^*$. Add the tuple (ID, w_{index}) to L_{H_0}.

– H_1 queries on a tuple $\mu = (m||U_1||U_2||\prod_1||\prod_2||R)$: If μ has been defined in L_{H_1}, retrieves c from L_{H_1} and returns c to F, else chooses a new random $c \in_R Z_p^*$ and adds (μ, c) into L_{H_1}.

– Key extraction queries on ID: B recovers the matching pair (ID, w) from L_0 and returns the previously computed $\frac{1}{\alpha+w} G$ if $ID \neq ID_\gamma$, else B sets $bad_1 = true$, then aborts.

– Signature queries on a pair (m, ID, R): If $ID \neq ID_\gamma$, B proceeds according to the sign algorithm. This is possible for B knows the private key of ID. If $ID = ID_\gamma$, then:

- Chooses $U_1 \in_R G_1$, $U_2 \in_R G_2$, chooses pairwise different $s_1, \ldots, s_5 \in_R Z_p^*$.
- Selects a random $c \in_R Z_p^*$, then computes $\prod_1 \leftarrow e(H,U_1)^{-s_5} \cdot e(H,G)^{s_2} \cdot e(H_{pub},G)^{s_1} \cdot e(H_{pub},U_1)^{-c} \cdot e(G,H)^c$; $\prod_2 \leftarrow e(G,U_2)^{-s_5} \cdot e(G,H)^{s_4} \cdot e(G_{pub},H)^{s_3} \cdot e(G_{pub},U_2)^{-c} \cdot e(G,V)^c$. Then add the new tuple $(m||U_1||U_2||\prod_1||\prod_2||R,c_i)$ in L_{H_1}(if (μ,c_i) had already been defined in L_{H_1}, set $bad_2 \leftarrow true$, aborts).
- Returns $\sigma = (U_1, U_2, \prod_1, \prod_2, s_1, s_2, s_3, s_4, s_5)$ as the signature.

We have explained how to simulate F's environment in chosen-messages and chosen-identities attack. So, B runs the algorithm F_B(t) as described in section 2.3. In this way we get two forgeries σ_0 and σ_1 together with a set of identities R and message m. Let c_0 be the answer from the random oracle H_1 given to F in the first execution, i.e., h_J in F_B(t), and let c_1 be the second answer $h_J^{'}$. The forged signature $\sigma_0 = (U_1, U_2, \prod_1, \prod_2, s_1, s_2, s_3, s_4, s_5)$ and another signature is $\sigma_1 = (U_1^{'}, U_2^{'}, \prod_1^{'}, \prod_2^{'}, s_1^{'}, s_2^{'}, s_3^{'}, s_4^{'}, s_5^{'})$. Let $f_i \leftarrow \frac{s_i - s_i^{'}}{c_0 - c_1}$ for $i \in \{1, \ldots, 5\}$, then we get a tuple $(f_5, U_1 - f_1 G)$ satisfying $e(f_5 H + H_{pub}, U_1 - f_1 G) = e(H,G)$. It implies that $w^* = f_5$ and $\frac{1}{\alpha + w^*} G = (U_1 - f_1 G)$. We note that $w^* \neq w_1, \ldots, w_{k-q}$ with probability at least $1 - \frac{k-q}{2^l}$. If both forgeries satisfy the verification equation, B can proceed as in [27] to extract $\frac{1}{\alpha + w^*} P$ from $\frac{1}{\alpha + w^*} G$:

- Writes $f(z) = \prod_{i=1}^{k-q}(w_i + z) = \gamma(z)(z + w^*) + \gamma_{-1}$, where $\gamma_{-1} \in Z_p$ and $\gamma(z) = \sum_{i=0}^{k-q-1} \gamma_i z^i$.
- Then $\frac{f(z)}{w^* + z} = \frac{\gamma_{-1}}{w^* + z} + \sum_{i=0}^{k-q-1} \gamma_i z^i$. Since $G = \psi(H) = f(\alpha) P \in G_1$, as thus $\frac{1}{\alpha + w^*} G = \frac{f(\alpha)}{\alpha + w^*} P = \frac{\gamma_{-1}}{\alpha + w^*} P + \sum_{i=0}^{k-q-1} \gamma_i (\alpha^i P)$.
- It's easy to get $\frac{1}{\alpha + w^*} P = \frac{1}{\gamma_{-1}} [\frac{1}{w^* + \alpha} G - \sum_{i=0}^{k-q-1} \gamma_i (\alpha^i P)]$, then the tuple $(w^*, \frac{1}{\alpha + w^*} P)$ will be the answer of the k-strong Diffie-Hellman problem.

Let Pr[bad_i] denote the probability of the event that flag bad_i set to be true(fails in providing a consistent simulation). We bound the accepting probability acc as follows:

$$\begin{aligned} acc &\geq \varepsilon - Pr[bad_1] - Pr[bad_2] \\ &\geq \varepsilon - \frac{q_{H_0}}{2^l} - \frac{q_{H_1} + q_s}{2^l} \end{aligned}$$

The probability that algorithm B succeeds in getting the answer of the k-strong Diffie-Hellman problem is given by

$$\begin{aligned} \varepsilon^{'} &\geq \frac{frk}{q_{H_1}} \\ &\geq \frac{acc^2}{q_{H_1}^2} - \frac{1}{2^l} \end{aligned}$$

The running time $t^{'}$ is twice that of once execution in $F_B(t)$ plus the time needed to compute the solution of the k-strong Diffie-Hellman problem. The running time of once execution in $F_B(t)$ is the running time t of F plus the time needed to answer q_H random oracle queries and q_s signature queries, where q_H denotes the maximum total number of queries to all random oracles. We assume that t_p denotes the require time for a paring evaluation, t_{exp} and t_{mult} respectively denotes the costs of an exponentiation in G_T and a multiplication in G_2, and all other operations take unit time. Each random oracle query at most cause B to perform $\mathcal{O}(1 + q_H + q_s)$ unit-time operations. Each signature query involves at most $10t_{exp} + 9t_p + (n+1)q_{mult} + (n+1)\mathcal{O}(1 + q_H + q_s)$ operations, where n is the maximum number of identities of each signature query. Therefore, we have $t^{'} \leq 2t + q_s[10t_{exp} + 9t_p + (n+1)q_{mult}] + \mathcal{O}[(q_s(n+1) + q_H)(1 + q_s + q_H)]$.

5.2 Anonymity

In order to give the proof for anonymity, we present the proofs of our scheme's perfect zero-knowledge is enough. The simulator randomly chooses $U_1, U_2 \in_R \mathbb{G}_1$, $c, s_1, s_2, s_3, s_4, s_5 \in_R \mathbb{Z}_p$, then computes $\prod_1 = e(Q,U_1)^{-s_5} \cdot e(Q,P)^{s_2} \cdot e(Q_{pub},P)^{s_1} \cdot e(Q_{pub},U_1)^{-c} \cdot e(P,Q)^c$; $\prod_2 = e(P,U_2)^{-s_5} \cdot e(P,Q)^{s_4} \cdot e(P_{pub},Q)^{s_3} \cdot e(P_{pub},U_2)^{-c} \cdot e(P,V)^c$. We can see that the distribution of the simulation is the same as the real transcript. This completes the proof.

6 Some Remarks and Efficiency Comparison

In many scenarios, as pointed in [7], the group doesn't change for a long time or has a short description. So an appropriate measurement of ring signature-size does not need to include the group description. In this situation, both the signer and verifier need to perform a one-time computation proportional to the size of the ring, and get the gpk and gsk which allow them to produce/verify many subsequent signatures in constant time. It's obvious that the constant-size ring signature scheme will be much more efficient than the previous schemes which signature size proportional to the size of the ring in such scenarios. We note that even in large ad-hoc groups, the size of our signature scheme is much smaller than that of schemes which the size of signature linearly depends on the group size. To the best of our knowledge, Chow et al.'s scheme [14] is the most efficient one among all the ID-based ring signature schemes which the size of signature linearly depends on the group size. For the sake of comparison and concreteness, we fix $|G_1| = |Z_p| = 256$ bits for a security level equivalent to a 128-bit symmetric key for AES(cf.[28]). We conclude that our scheme has smaller size than Chow et al.'s scheme when the number of identities of the ring over 8.

The first constant-size ring signature scheme (DKNS04)

had been proposed by Dodis, Kiayias, Nicolosi and Shoup [7], after that, no more efficient constant-size ring signature scheme was found until the first secure ID-based ring signature scheme with constant-size signatures proposed by Nguyen [10]. Nguyen had compared his constant-size ID-based ring signature scheme with DKNS04 at the same level of security. The conclusion is that the signature size is very much smaller than that of constant-size ring signature scheme DKNS04 [7]. He also pointed out that in the future, when higher levels of security are required, this difference even grows much larger.

We now make a specific comparison between our scheme and that of Nguyen's. Due to our scheme actually is an improvement on the modified version of the scheme proposed by Nguyen [10], it seems that our scheme and Nguyen's scheme are implemented by the same elliptic curve or hyperelliptic curve over a finite field is reasonable. As shown in [10], we assume p is a 160-bit Jacobian of a hyperelliptic curve over a finite field with order p and compression techniques are used. G_T is a subgroup of a finite field of size approximately 2^{1024}. A possible choice of these parameters is that $G_1(G_1 = G_2)$ is derived from the curve $E/GF(3^l)$ defined by $y^2 = x^3 - x + 1$.

We summarize the result in Table 1. It's obvious that we greatly reduce the size of the signature, although the computational efficiency is improved slightly. It should be noted that our scheme has the same keys(GSK, GPK) with Nguyen's, so we don't list them(i.e. computation of keys and size of keys) in the table 1. Here we give our analysis of why our scheme could cut down the computation and size of the signature. As we mentioned is section 4, our ring signature scheme and Nguyen's scheme actually are both non-interactive proof of the knowledge of (s_{id}, h_{id}, W) satisfying $e(h_{id}Q + Q_{pub}, S_{id}) = e(Q, P)$ and $e(h_{id}P + P_{pub}, W) = e(P, V)$. In our signature $\sigma = (U_1, U_2, \prod_1, \prod_2, s_1, s_2, s_3, s_4, s_5)$, $\sigma_1 \stackrel{def}{=} (U_1, \prod_1, s_1, s_2, s_5)$ are used to prove the knowledge (s_{id}, h_{id}) satisfying $e(h_{id}Q + Q_{pub}, S_{id}) = e(Q, P)$, and $\sigma_2 \stackrel{def}{=} (U_2, \prod_2, s_3, s_4, s_5)$ are used to prove the knowledge (s_{id}, h_{id}) satisfying $e(h_{id}P + P_{pub}, W) = e(P, V)$. It looks like that there should be some extra "useful" data to set up a tough relation between σ_1 and σ_2 to build up resistance to attack whereby the adversary "tricked" generate a new valid signature use several valid signatures. Actually, this is what Nguyen's scheme did. However, it's easy to see that σ_1 and σ_2 share the same element $s_5 = k_5 + ch_{id}$ which is used to prove the relationships of (s_{id}, h_{id}, W). In our scheme, s_5, $\prod_1$ and $\prod_2$ share the same random number k_5. It implies that each signature has a unique random number, and it doesn't leave open the possibility of an attack whereby the adversary "tricked" generate a new valid signature use several valid signatures. So, the extra "useful" data is really redundancy. Our constant-size ring signature actually is the essence of the Nguyen's scheme, i.e. the remaining part of the Nguyen's scheme after cut extra "useful" data down. Then, there is a question: is there a possibility that cut something down from our signature scheme? Due to the way we construct the private key of user's, it looks like paring operation is necessary. If paring operation is necessary, it's really very hard to cut something down from our signature scheme.

Table 1: Efficiency comparison

scheme	signature size	mul	padd	pmul
Nguyen's	2,240 bits	7	15	20
Ours	1,440 bits	5	2	2

*mul, $padd$ and $pmul$ respectively indicate the number of multiplications, point additions and point scalar multiplications.

7 Conclusions

We have proposed an improved ID-based constant-size ring signature scheme based on Nguyen's scheme, which will be useful for implementation in large ring scenario. Our scheme outperforms in size of signature the previously proposed constant-size ring signatures and admits proofs of secure in the random oracle model based on a simplified and general Forking Lemma under the k-strong Diffie-Hellman assumption.

Acknowledgments

This work is supported by National Natural Science Foundation of China (60773035), The fund of Key Disciplinary of Computer Software and Theory(SZD0802-09-1), The research fund of key disciplinary of application mathematics (XZD0910-09-1).

References

[1] R.Rivest,A.Shamir, and Y.Tauman. How to Leak a Secret: Theory and Applications of Ring Signatures. in: Theoretical Computer Science. LNCS, vol.3895, Springer Berlin, 2006, pp.164-186.

[2] A.Shamir. Identity-Based Cryptosystems and Signature Schemes. in: Advances in Cryptology. LNCS, vol.196, Springer Berlin, 1985, pp.47-53.

[3] F.Zhang, and K.Kim. ID-Based Blind Signature and ring from pairings. in: Advances in Cryptology - ASIACRYPT 2002. LNCS, vol.2501, Springer Berlin, 2002, pp.629-637.

[4] J.Benaloh and M.de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. in: Advances in Cryptology - EUROCRYPT'93. LNCS, vol.765, Springer Berlin, 1994, pp.274-285.

[5] N.Barić and B.Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. in: Advances in Cryptology - EUROCRYPT'97. LNCS, vol.1233, Springer Berlin, 1997, pp.480-494.

[6] J.Camenisch, and A.Lysyanskaya. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. in: Advances in Cryptology - CRYPTO 2002. LNCS, vol.2442, Springer Berlin, 2002, pp.101-120.

[7] Y.Dodis, A.Kiayias, A.Nicolosi, and V.Shoup. Anonymous Identification in Ad Hoc Groups. in: Advances in Cryptology - EUROCRYPT 2004. LNCS, vol.3027, Springer Berlin, 2004, pp.609-626.

[8] Lan Nguyen. Accumulators from Bilinear Pairings and Applications. in: Topics in Cryptology - CT-RSA 2005. LNCS, vol.3376, Springer Berlin, 2005, pp.275-292.

[9] C Tartary, S Zhou, D Lin, H Wang and J Pieprzyk. Analysis of bilinear pairing-based accumulator for identity escrowing. Information Security, IET 2(4)(2008), pp.99-107.

[10] Lan Nguyen. Accumulators from Bilinear Pairings and Applications to ID-based Ring Signatures and Group Membership Revocation. http://eprint.iacr.org/2005/123.

[11] Chih-Yin Lin and Tzong-Chen Wu. An Identity-based Ring Signature Scheme from Bilinear Pairings. in: AINA'04, Also appear in http://eprint.iacr.org/2003/117.

[12] Javier Herranz and Germán Sáez. New Identity-Based Ring Signature Schemes. in: Information and Communications Security. LNCS, vol.3269, Springer Berlin, 2004, pp. 269-274.

[13] Sherman S.M.Chow, S.M.Yiu, and Lucas C.K.Hui. Efficient Identity Based Ring Signature. in: Applied Cryptography and Network Security. LNCS, vol.3531, Springer Berlin, 2005, pp.499-512.

[14] S.S.M. Chow,R.W.C. Lui, L.C.K.Hui, and S.M.Yiu. Identity Based Ring Signature: Why, How and What Next. in: Public Key Infrastructure. LNCS, vol.3545, Springer Berlin, 2005, pp.144-161.

[15] A.Fiat, and A.Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. in: Advances in Cryptology - CRYPTO'86. LNCS, vol.263, Springer Berlin, 1987, pp. 186-194.

[16] F.Zhang and Xiaofeng Chen. Cryptanalysis and improvement of an ID-based ad-hoc anonymous identification scheme at CT-RSA 05. Information Processing Letters 105(15)(2009) pp.846-849.

[17] D.Boneh, and X.Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups . Journal of Cryptology 21(2)(2008), pp.149-177.

[18] M.Bellare and G.Neven. Multi-signatures in the plain public-key model and a generalized forking lemma. in: Conference on Computer and Communications Security: Proceedings of the 13th ACM conference on Computer and communications security, ACM, New York, 2006, PP.390-399.

[19] J.Herranz and G.Sáez. Forking lemmas for ring signature schemes. in: Progress in Cryptology - INDOCRYPT 2003. LNCS, vol.2904, Springer Berlin, 2003, pp.266-279.

[20] Adam Bender, Jonathan Katz, Ruggero Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. Journal of Cryptology 22(1)(2009), pp.114-138.

[21] A.Miyaji, M.Nakabayashi, and S.Takano. New explict conditions of elliptic curve traces for FR-reduction. IEICE Transactions on Fundamentals E84-A(5)(2001), pp.1234-1243.

[22] N.P.Smart and F.Vercauteren. On computable isomorphisms in efficient pairing based systems. http://eprint.iacr.org/2005/116.

[23] Fiore,D., Gennaro,R. Making the Diffie-Hellman Protocol Identity-Based. in: Topics in Cryptology - CT-RSA 2010. LNCS, vol.5985, Springer Berlin, 2010, pp.165-178.

[24] J.Camenisch, M.Kohlweiss and C.Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. in: Public Key Cryptography - PKC 2009. LNCS, vol.5443, Springer Berlin, 2009, pp.481-500.

[25] Paulo S.L. Barreto, B.Libert, N.McCullagh, J.J.Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. in: Advances in Cryptology - ASIACRYPT 2005. LNCS, vol.3788, Springer Berlin, 2005, pp.515-532.

[26] J.Herranz. Identity-based ring signatures from RSA. Theoretical Computer Science 389(1-2)(2007) pp.100-117.

[27] D.Boneh and X.Boyen. Short Signatures Without Random Oracles. in: Advances in Cryptology - EUROCRYPT 2004. LNCS, vol.3027, Springer Berlin, 2004, pp.56-73.

[28] ECRYPT II Yearly Report on Algorithms and Key Lengths (2010), http://www.ecrypt.eu.org/documents/D.SPA.13.pdf (revision 1.0, 30 March 2010).