# A Provably Secure ID-Based Ring Signature Scheme

Javier Herranz and Germán Sáez

Dept. Matemàtica Aplicada IV, Universitat Politècnica de Catalunya
C. Jordi Girona, 1-3, Mòdul C3, Campus Nord, 08034-Barcelona, Spain
{jherranz,german}@mat.upc.es

**Abstract.** Identity-based (ID) cryptosystems avoid the necessity of certificates to authenticate public keys in a digital communications system. This is desirable, specially for these applications which involve a large number of public keys in each execution. For example, any computation and verification of a ring signature, where a user anonymously signs a message on behalf of a set of users including himself, requires to authenticate the public keys of all the members of the set.
We use bilinear pairings to design a new ID-based ring signature scheme. We extend to the ID-based scenario some known results about the security of generic ring signature schemes. This allows us to formally prove the security of our scheme, under the assumption that the Computational Diffie-Hellman problem is hard to solve.

## 1 Introduction

In a *ring signature scheme*, a user forms a set (or ring) of users which contains himself, and anonymously computes a signature on behalf of the whole ring. Any verifier must be convinced that the signature has been computed by some member of this ring, but he has no information about who is the actual author of the signature.

In real applications, however, the public keys of the users are authenticated via a Public Key Infrastructure (PKI) based on certificates. Therefore, the signer must first verify that the public keys of the ring correspond to the identities of the users that he wants to include on the ring. Later, the verifier must first check the validity of the certificates of all the public keys of the members of the ring.

This necessary management of digital certificates substantially increases the cost of both generation and verification of a ring signature. Thus, any possible alternative which avoids the necessity of digital certificates is welcome in order to design efficient ring signature schemes in particular, and efficient public key cryptosystems in general.

Shamir introduced in 1984 the concept of *identity-based* cryptography (from now on, ID-based) [14]. The idea is that the public key of a user can be publicly computed from his identity (for example, from a complete name, an e-mail or an IP address). Then, the secret key is derived from the public key. In this way, digital certificates are not needed, because anyone can easily verify that some public key $PK_U$ corresponds in fact to user $U$.

The process that generates secret keys from public keys must be executed by an external entity, known as the *master*. Thus, the master knows the secret keys of all the users of the system. A way to relax this negative point could be to consider a set of master entities which share the secret information. A different approach is that of *Certificateless* Public Key Cryptography [2], which avoids this key escrow property, but loosing in some way the ID-based property.

In this work we present a provably secure ID-based ring signature scheme, based on bilinear pairings. Let us do a brief overview of some works related to ring signatures.

The first proposals of ring signature schemes are previous to the formal definition of this concept. They can be found in [6, 5] and they are used as a tool to construct group signature schemes. They use zero-knowledge proofs and witness indistinguishable proofs of knowledge for disjunctive statements (introduced in [7, 8]).

In [13], Rivest, Shamir and Tauman formalize the concept of ring signature schemes, and propose a scheme which they prove existentially unforgeable under adaptive chosen-message attacks, in the ideal cipher model, assuming the hardness of the RSA problem.

Bresson, Stern and Szydlo give in [4] a simpler proof of the security of the scheme in [13], under the strictly weaker assumption of the random oracle model. They propose as well a threshold ring signature scheme, in which a set of $t$ users sign a message on behalf of a ring that contains themselves, in such a way that the verifier is convinced of the participation of $t$ users in the generation of the signature, but he does not obtain any information about which $t$ users have in fact signed the message.

In [1], Abe, Ohkubo and Suzuki design some general ring signature schemes where the public keys of the users can be totally independent: different sizes, different types of keys (RSA keys, discrete logarithm keys...).

Herranz and Saez [11] give some security results for generic ring signature schemes, and they design a new specific scheme based on Schnorr's signature scheme.

Finally, the only ID-based ring signature scheme proposed until now (as far as we know) is the one by Zhang and Kim [15]. Their scheme is also based on pairings. Although the authors do not provide a formal proof of the existential unforgeability of their scheme, such a proof can be found in [10].

We extend to the ID-based scenario the results of Herranz and Sáez in [11] (the so called *ring forking lemmas*). We propose then a new ID-based ring signature scheme. Since this scheme is generic, we can use these results to provide a formal proof of the existential unforgeability of our scheme under chosen message attacks in the ID-based model, assuming that the Computational Diffie-Hellman problem is hard to solve.

The paper is organized as follows. In Section 2 we explain the mathematical background that we need for designing our scheme. In Section 3 we review the properties that a ring signature scheme must satisfy, and we recall some known results about generic ring signature schemes. Then, we present our ID-based ring signature in Section 4. We prove the security of this scheme in Section 5, by using a new ring forking lemma for the ID-based scenario. The conclusions of the work are presented in Section 6.

# 2 ID-Based Schemes from Pairings

Let $\mathbb{G}_1$ be an additive group of prime order $q$, generated by some element $P$. Let $\mathbb{G}_2$ be a multiplicative group with the same order $q$.

A *pairing* is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ with the following three properties:

1. It is bilinear, which means that given elements $A_1, A_2, A_3 \in \mathbb{G}_1$, we have that $e(A_1 + A_2, A_3) = e(A_1, A_3) \cdot e(A_2, A_3)$ and $e(A_1, A_2 + A_3) = e(A_1, A_2) \cdot e(A_1, A_3)$. In particular, for all $a, b \in \mathbb{Z}_q$, we have $e(aP, bP) = e(P, P)^{ab} = e(P, abP) = e(abP, P)$.
2. The map $e$ can be efficiently computed for any possible input pair.
3. The map $e$ is non-degenerate: there exist elements $A_1, A_2 \in \mathbb{G}_1$ such that $e(A_1, A_2) \neq 1_{\mathbb{G}_2}$.

Combining properties 1 and 3, it is easy to see that $e(P, P) \neq 1_{\mathbb{G}_2}$ and that the equality $e(A_1, P) = e(A_2, P)$ implies that $A_1 = A_2$.

The typical way of obtaining such pairings is by deriving them from the Weil or the Tate pairing on an elliptic curve over a finite field. The interested reader is referred to [16] for a complete bibliography of cryptographic works based on pairings.

Let $H_1 : \{0, 1\}^* \to \mathbb{G}_1 - \{0\}$ be a hash function. The most usual way to design an ID-based cryptosystem is the following. The master has a secret key $x \in \mathbb{Z}_q^*$, and he publishes the value $Y = xP \in \mathbb{G}_1$.

Every user $U$ of the ID-based system has an identifier $ID_U \in \{0, 1\}^*$, that can be an IP address, a telephone number, an e-mail address, etc. The public key of $U$ is then defined to be $PK_U = H_1(ID_U) \in \mathbb{G}_1 - \{0\}$. In this way, everybody can verify the authenticity of a public key without the necessity of certificates.

The user $U$ needs to contact the master to obtain his secret key $SK_U = xPK_U \in \mathbb{G}_1$. The drawback of this approach, as mentioned in the Introduction, is that the master must be completely trusted, because he knows the secret keys of all the users.

## 2.1 The Computational Diffie-Hellman Problem

We consider the following well-known problem in the group $\mathbb{G}_1$ of prime order $q$, generated by $P$:

**Definition 1.** *Given the elements $P$, $aP$ and $bP$, for some random values $a, b \in \mathbb{Z}_q^*$, the Computational Diffie-Hellman (CDH) problem consists of computing the element $abP$.*

The Computational Diffie-Hellman Assumption asserts that, if the order of $\mathbb{G}_1$ is $q \geq 2^k$, then any polynomial time algorithm that solves the CDH problem has a success probability $p_k$ which is negligible in the security parameter $k$. In other words, for all polynomial $f()$, there exists an integer $k_0$ such that $p_k < \frac{1}{f(k)}$, for all $k \geq k_0$.

The security of the ID-based ring signature scheme that we propose in this work is based on the CDH Assumption.

## 3 Ring Signatures

The idea of a ring signature is the following: a user wants to compute a signature on a message, on behalf of a set (or ring) of users which includes himself. He wants the verifier of the signature to be convinced that the signer of the message is in effect some of the members of this ring. But he wants to remain completely anonymous. That is, nobody will know which member of the ring is the actual author of the signature.

These two informal requirements are ensured, if the scheme satisfies the following properties:

1. **Anonymity:** any verifier should not have probability greater than $1/n$ to guess the identity of the real signer who has computed a ring signature on behalf of a ring of $n$ members. If the verifier is a member of the ring distinct from the actual signer, then his probability to guess the identity of the real signer should not be greater than $1/(n-1)$.
2. **Unforgeability:** among all the proposed definitions of unforgeability (see [9]), we consider the strongest one: any attacker must have negligible probability of success in forging a valid ring signature for some message $m$ on behalf of a ring that does not contain himself, even if he knows valid ring signatures for messages, different from $m$, that he can adaptively choose.

### 3.1 Forking Lemmas for Generic Ring Signature Schemes

Herranz and Sáez define in [11] a class of ring signature schemes that they call *generic*. Consider a security parameter $k$, a hash function which outputs $k$-bit long elements, and a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of $n$ members. Given the input message $m$, a generic ring signature scheme produces a tuple $(\mathcal{U}, m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$.

The elements $R_1, \ldots, R_n$ (randomness) take their values randomly in some large set in such a way that $R_i \neq R_j$ for all $i \neq j$; $h_i$ is the hash value of $(\mathcal{U}, m, R_i)$, for $1 \leq i \leq n$; and the value $\sigma$ is fully determined by $R_1, \ldots, R_r, h_1, \ldots, h_n$ and the message $m$.

Another required condition is that no $R_i$ can appear with probability greater than $2/2^k$, where $k$ is the security parameter.

These generic ring signature schemes are the natural extension of the generic signature schemes considered by Pointcheval and Stern in [12]. The last authors invented the forking lemmas in order to prove the security of generic signature schemes. In [11], these lemmas are extended to the ring's scenario, in order to show the security of generic ring signature schemes.

We mention here the basic ring forking lemma. This lemma is also extended in [11] to the chosen message attacks' scenario. Analogously, we will extend it, in Section 5.1, to a chosen message ID-based scenario, suitable for proving the security of generic ID-based ring signature schemes.

**Theorem 1.** *(Basic Ring Forking Lemma, [11]) Consider a generic ring signature scheme with security parameter $k$, and let $\hat{n}$ be the maximum number of members of a possible ring. Let the forger $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can ask $Q$ queries to the random oracle, with $Q \geq \hat{n}$. We denote as $V_{Q,\hat{n}}$ the number of $\hat{n}$-permutations of $Q$ elements, that is, $V_{Q,\hat{n}} = Q(Q-1) \cdot \ldots \cdot (Q - \hat{n} + 1)$. We assume that, within time bound $T$, $\mathcal{A}$ produces, with probability of success $\varepsilon \geq \frac{7\ V_{Q,\hat{n}}}{2^k}$, a valid ring signature $(\mathcal{U}, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$.*

*Then, within time $T' \leq \frac{16 V_{Q,\hat{n}} T}{\varepsilon}$, and with probability $\varepsilon' \geq \frac{1}{9}$, a replay of this machine outputs two valid ring signatures $(\mathcal{U}, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(\mathcal{U}, R_1, \ldots \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ such that $h_j \neq h'_j$, for some $j \in \{1, \ldots, n\}$ and $h_i = h'_i$ for all $i = 1, \ldots, n$ such that $i \neq j$.*

The idea in the proof of this theorem is to execute many times the machine $\mathcal{A}$, with the same random tape but with different random oracles for the hash function. After a certain number of executions and with a certain probability, we obtain a new valid ring signature of the same message, with the same randomness but with a different random oracle which has the same outputs than the first one for all the inputs $\{(\mathcal{U}, m, R_i)\}_{1 \leq i \leq n, i \neq j}$ except one, $(\mathcal{U}, m, R_j)$, for which they have different outputs.

In [11] the authors propose a specific generic ring signature scheme. They use the ring forking lemmas to show that this scheme is secure, assuming that the discrete logarithm problem in subgroups of prime order is hard to solve.

In next section, we present an ID-based ring signature scheme which is also generic. Therefore, we could use a new ID-oriented ring forking lemma to show that this new scheme is secure, assuming that the Computational Diffie-Hellman problem is hard to solve.

## 4 Our ID-Based Ring Signature Scheme

In this section we present a new ID-based ring signature scheme. As the one proposed by Zhang and Kim in [15], our scheme is based on bilinear pairings. However, the design of the scheme follows the idea of the Schnorr ring signature scheme, not ID-based, proposed in [11].

*Setup:* let $\mathbb{G}_1$ be an additive group of prime order $q$, generated by some element $P$. Let $\mathbb{G}_2$ be a multiplicative group with the same order $q$. We need $q \geq 2^k$, where $k$ is the security parameter of the scheme. Let $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a pairing as defined in Section 2. Let $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ be two hash functions (in the proof of security, we will assume that they behave as random oracles [3]).

The master entity chooses at random his secret key $x \in \mathbb{Z}_q^*$ and publishes the value $Y = xP$.

*Secret key extraction:* a user $U$, with identity $ID_U$, has public key $PK_U = H_1(ID_U)$. When he requests the master for his matching secret key, he obtains the value $SK_U = xPK_U$.

*Ring Signature:* consider a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of users; for simplicity we denote $PK_i = PK_{U_i} = H_1(ID_{U_i})$. If some of these users $U_s$, where $s \in \{1, \ldots, n\}$, wants to anonymously sign a message $m$ on behalf of the ring $\mathcal{U}$, he acts as follows:

1. For all $i \in \{1, \ldots, n\}$, $i \neq s$, choose $A_i$ uniformly and at random in $\mathbb{G}_1^*$, pairwise different (for example, by choosing $a_i \in \mathbb{Z}_q^*$ at random and considering $A_i = a_iP$). Compute $R_i = e(A_i, P) \in \mathbb{G}_2$ and $h_i = H_2(\mathcal{U}, m, R_i)$, for all $i \neq s$.
2. Choose a random $A \in \mathbb{G}_1$.
3. Compute $R_s = e(A, P) \cdot e(-Y, \sum_{i \neq s} h_i PK_i)$. If $R_s = 1_{\mathbb{G}_2}$ or $R_s = R_i$ for some $i \neq s$, then go to step 2.
4. Compute $h_s = H_2(\mathcal{U}, m, R_s)$.
5. Compute $\sigma = h_s SK_s + A + \sum_{i \neq s} A_i$.
6. Define the signature of the message $m$ made by the ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ to be $(\mathcal{U}, m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$.

In fact, the values $h_i$ can be publicly computed from the ring $\mathcal{U}$, the message $m$ and the values $R_i$. We include them in the signature for clarity in the treatment of the security of the scheme.

*Verification:* the validity of the signature is verified by the recipient of the message by checking that $h_i = H_2(\mathcal{U}, m, R_i)$ and that

$$e(\sigma, P) = R_1 \cdot \ldots \cdot R_n \cdot e(Y, \sum_{i=1}^{n} h_i PK_i) \ .$$

### 4.1 Correctness and Unconditional Anonymity

The property of correctness is satisfied. In effect, if the ring signature has been correctly generated, then:

$$R_1 \cdot \ldots \cdot R_n \cdot e(Y, \sum_{i=1}^{n} h_i PK_i) = e(A + \sum_{i \neq s} A_i, P) \cdot e(-Y, \sum_{i \neq s} h_i PK_i) \cdot e(Y, \sum_{i=1}^{n} h_i PK_i) =$$

$$= e(A + \sum_{i \neq s} A_i, P) \cdot e(Y, h_s PK_s) = e(A + \sum_{i \neq s} A_i, P) \cdot e(P, h_s x PK_s) =$$
$$= e(A + \sum_{i \neq s} A_i + h_s SK_s, P) = e(\sigma, P) \ .$$

The unconditional anonymity of the scheme is also easy to prove. Intuitively, the scheme is completely symmetric. Therefore, for a valid ring signature on behalf of a ring $\mathcal{U}$, the probability that a specific user in $\mathcal{U}$ is the actual author of the signature is the same for all the members of $\mathcal{U}$. Specifically, if the ring has $n$ members, this probability is

$$\frac{1}{q-1} \cdot \frac{1}{q-2} \cdot \ldots \cdot \frac{1}{q-n+1} \cdot \frac{1}{q-n}$$

which does not depend on the considered member of $\mathcal{U}$.

So we can conclude that any attacker outside a ring of $n$ possible users has probability $1/n$ to guess which member of the ring has actually computed a given signature on behalf of this ring.

# 5 Security Analysis

We must consider the most powerful attack against an ID-based ring signature scheme, that we call *chosen message and identities attack*. Such an attacker $\mathcal{A}$ is allowed to:

- make $Q_1$ queries to the random oracle $H_1$ and $Q_2$ queries to the random oracle $H_2$;
- ask for the secret key of $Q_e$ identities of its choice (extracting oracle);
- ask $Q_s$ times for valid ring signatures, on behalf of rings of its choice, of messages of its choice (signing oracle).

The total number of queries must be polynomial in the security parameter. The attacker is successful if it outputs, in polynomial time and with non-negligible probability, a valid ring signature $(\mathcal{U}, m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ for some message $m$ and some ring of users $\mathcal{U} = \{U_1, \ldots, U_n\}$ such that:

- the attacker has not asked for the secret key of any of the members of the ring $\mathcal{U}$;
- the attacker has not asked for a valid ring signature, on behalf of the ring $\mathcal{U}$, of message $m$.

## 5.1 A New Ring Forking Lemma

In this section we introduce a new ring forking lemma that considers attackers, with the capabilities listed above, against ID-based generic ring signature schemes. We consider that public key of any user $U$ is $PK_U = H_1(ID_U)$. Now the hash function inherent to the generic ring signature scheme is $H_2$. In the security proof, we consider that both $H_1$ and $H_2$ behave as random oracles.

**Theorem 2.** *(ID-Oriented Chosen Message Ring Forking Lemma). Consider a generic, and ID-based, ring signature scheme with security parameter $k$, and let $\hat{n}$ be the maximum number of members of a ring. Suppose that both valid ring signatures and consistent pairs of secret-public keys can be simulated, with a polinomially indistinguishable distribution of probability and without knowing the master secret key.*

*Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine. We denote by $Q_1, Q_2, Q_e$ and $Q_s$ the number of queries that $\mathcal{A}$ can ask to the random oracles $H_1$ and $H_2$ and to the extracting and signing oracles, respectively. Assume that, within time bound $T$, $\mathcal{A}$ produces, with probability of success $\varepsilon \geq \frac{12\ V_{Q_2,\hat{n}} +\ 10(Q_1+Q_e)^2 +\ 10(Q_2+\hat{n}Q_s)^2}{2^k}$, a valid ring signature $(\mathcal{U}, m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$, such that $\mathcal{A}$ has not asked for the secret key of any of the members of $\mathcal{U}$, and has not asked for a valid ring signature of $m$ on behalf of the ring $\mathcal{U}$.*

*Then there is another probabilistic polynomial time Turing machine $\mathcal{B}$ which produces, with probability $\varepsilon' \geq \frac{1}{9}$ and in time $T' \leq \frac{27\ V_{Q_2,\hat{n}} T}{\varepsilon}$, two valid ring signatures $(\mathcal{U}, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(\mathcal{U}, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ such that $h_j \neq h'_j$, for some $j \in \{1, \ldots, n\}$ and $h_i = h'_i$ for all $i = 1, \ldots, n$ such that $i \neq j$.*

*Proof.* We consider a machine $\mathcal{B}$ that executes the machine $\mathcal{A}$, in such a way that $\mathcal{B}$ simulates all the environment of $\mathcal{A}$. Therefore, $\mathcal{B}$ must simulate the interactions of $\mathcal{A}$ with the random, extracting and signing oracles. Then we could see $\mathcal{B}$ as a machine performing a no-message attack against the ring signature scheme.

When $\mathcal{A}$ asks the extracting oracle for the secret key corresponding to some identity $ID$, the machine $\mathcal{B}$ runs the extracting simulator to obtain consistent pairs $SK_{ID}$ and $PK_{ID}$. He sends the value $SK_{ID}$ to $\mathcal{A}$. Furthermore, $\mathcal{B}$ constructs a random oracle $H_1$ by storing in a "random oracle list" the relations $H_1(ID) = PK_{ID}$.

When $\mathcal{A}$ makes a query $ID$ to the random oracle $H_1$, $\mathcal{B}$ looks for the value $ID$ in this random oracle list. If the value is already in the list, then $\mathcal{B}$ returns to $\mathcal{A}$ the corresponding $H_1(ID)$. Otherwise, $\mathcal{B}$ chooses a random value $PK$, sends it to $\mathcal{A}$ and stores the relation $H_1(ID) = PK$ in the list.

There is some risk of "collisions" of queries to the random oracle $H_1$. Here we must have that the outputs of $H_1$ belong to a group with order greater than $2^k$, where $k$ is the security parameter. As well, the public keys $PK$ obtained from the extracting simulator must be uniformly distributed in this group. Therefore, a determined $PK$ appears with probability less than $1/2^k$.

Three kinds of collisions can happen:

- A value $PK_{ID}$ that the extracting simulator outputs, has been already given to $\mathcal{A}$ as the answer of some previous query to the random oracle $H_1$. In this case, it is quite unlikely that the relation $H_1(ID) = PK_{ID}$ corresponding to the values output by the simulator matches with the relation previously stored in the random oracle list. The probability of such a collision is, however, less than $Q_1 \cdot Q_e \cdot \frac{1}{2^k} \leq \frac{\varepsilon}{20}$.

- A value $PK_{ID_1}$ that the extracting simulator outputs is exactly equal to another value $PK_{ID_2}$ also output by this simulator. The probability of this collision is less than $\frac{Q_e^2}{2} \cdot \frac{1}{2^k} \leq \frac{\varepsilon}{20}$.
- Two answers $PK_1$ and $PK_2$ of the random oracle $H_1$ chosen at random by $\mathcal{B}$ are exactly equal, while the two corresponding inputs $ID_1$ and $ID_2$ are different. The probability of such an event is less than $\frac{(Q_1+Q_e)^2}{2} \cdot \frac{1}{2^k} \leq \frac{\varepsilon}{20}$.

On the other hand, when $\mathcal{A}$ asks for a valid ring signature for some message $m^{(j)}$ and some ring of $n_j \leq \hat{n}$ users $\mathcal{U}^{(j)}$ (possibly repeated), $\mathcal{B}$ uses the signature simulator and sends to $\mathcal{A}$ the resulting tuple $(\mathcal{U}^{(j)}, m^{(j)}, R_1^{(j)}, \ldots, R_{n_j}^{(j)}, h_1^{(j)}, \ldots, h_{n_j}^{(j)}, \sigma^{(j)})$. Then $\mathcal{B}$ constructs another random oracle $H_2$ by storing in a different list the relations $H_2(\mathcal{U}^{(j)}, m^{(j)}, R_i^{(j)}) = h_i^{(j)}$.

When $\mathcal{A}$ makes a query $(\mathcal{U}, m, R)$ to the random oracle $H_2$, $\mathcal{B}$ looks for the value $(\mathcal{U}, m, R)$ in the random oracle list. If the value is already in the list, then $\mathcal{B}$ returns to $\mathcal{A}$ the corresponding $H_2(\mathcal{U}, m, R)$. Otherwise, $\mathcal{B}$ chooses a random value $h$, sends it to $\mathcal{A}$ and stores the relation $H_2(\mathcal{U}, m, R) = h$ in the list.

Analogously to what happens with random oracle $H_1$, there can be collisions in the management of the random oracle $H_2$ by the machine $\mathcal{B}$. Recall that we made the assumption that no $R_i$ can appear with probability greater than $2/2^k$ in a generic ring signature. If the simulator outputs ring signatures which are indistinguishable of the ones produced by a real signer of the ring, then we have that no $R_i^{(j)}$ can appear with probability greater than $2/2^k$ in a simulated ring signature, too. Since the values $h_i^{(j)}$ are the outputs of the random oracle, then we have that a determined $h_i^{(j)}$ appears in a ring signature (real or simulated) with probability less than $1/2^k$.

Again, three kinds of collisions can occur:

- A tuple $(\mathcal{U}^{(j)}, m^{(j)}, R_i^{(j)})$ that the signing simulator outputs, as part of a simulated ring signature, has been asked before to the random oracle $H_2$ by $\mathcal{A}$. The probability of such a collision is less than $Q_2 \cdot \hat{n}Q_s \cdot \frac{2}{2^k} \leq \frac{\varepsilon}{10}$.
- A pair $(\mathcal{U}^{(j_1)}, m^{(j_1)}, R_{i_1}^{(j_1)})$ that the simulator outputs, as part of a simulated ring signature, is exactly equal to another pair $(\mathcal{U}^{(j_2)}, m^{(j_2)}, R_{i_2}^{(j_2)})$ also output by the simulator. The probability of this collision is less than $\frac{(\hat{n}Q_s)^2}{2} \cdot \frac{2}{2^k} \leq \frac{\varepsilon}{10}$.
- Two answers $h_1$ and $h_2$ of the random oracle $H_2$ chosen at random by $\mathcal{B}$ are exactly equal, while the two corresponding inputs $(\mathcal{U}^{(1)}, m^{(1)}, R_1)$ and $(\mathcal{U}^{(2)}, m^{(2)}, R_2)$ are different. The probability of such an event is less than $\frac{(Q_2+\hat{n}Q_s)^2}{2} \cdot \frac{1}{2^k} \leq \frac{\varepsilon}{20}$.

Altogether, the probability of collisions is less than $8\varepsilon/20 = 2\varepsilon/5$. Now we can compute:

$$\Pr[\mathcal{B} \text{ succeeds}] = \Pr[\text{no-collisions in the simulations and } \mathcal{A} \text{ succeeds}] \geq$$

$$\geq \Pr[\mathcal{A} \text{ succeeds} \mid \text{no-collisions}] \; - \; \Pr[\text{collisions}] \geq \varepsilon - \frac{2\varepsilon}{5} = \frac{3\varepsilon}{5} \; .$$

Summing up, we have a machine $\mathcal{B}$ that performs a no-message attack against the ring signature scheme with time bound $T$ (plus the execution time of the extracting and signing simulators, that we consider negligible with respect to $T$) and with probability of success greater than $\frac{3\varepsilon}{5} \geq \frac{7V_{Q_2,\hat{n}}}{2^k}$. So we can use Theorem 1 applied to the machine $\mathcal{B}$, and we will obtain, with probability greater than 1/9 and in time bounded by $\frac{16V_{Q_2,\hat{n}}T}{3\varepsilon/5} \leq \frac{27V_{Q_2,\hat{n}}T}{\varepsilon}$, the two desired valid ring signatures. □

### 5.2 Unforgeability of the Scheme

We prove that the existence of a successful attack against our scheme could be used to solve the Computational Diffie-Hellman problem in $\mathbb{G}_1$ (a proof by reduction). Since this problem is assumed to be hard, we conclude that there does not exist such an attack. In this way, our scheme is proved to be existentially unforgeable under chosen message and identities attacks.

**Theorem 3.** *Let $k$ be a security parameter, and let the order of $\mathbb{G}_1$ be $q \geq 2^k$. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine attacking our ID-based ring signature scheme. We denote by $Q_1, Q_2, Q_e$ and $Q_s$ the number of queries that $\mathcal{A}$ can ask to the random oracles $H_1$ and $H_2$ and to the extracting and signing oracles, respectively. We denote by $\hat{n}$ the maximum cardinality of the rings for which $\mathcal{A}$ asks for a valid signature.*

*Assume that $\mathcal{A}$ produces, within time bound $T$ and probability of success $\varepsilon \geq \frac{12\ V_{Q_2,\hat{n}} +\ 10(Q_1+Q_e)^2 +\ 10(Q_2+\hat{n}Q_s)^2}{2^k}$, a valid ring signature $(\mathcal{U}, m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$, such that $\mathcal{A}$ has not asked for the secret key of any of the members of $\mathcal{U}$, and has not asked for a valid ring signature of $m$ on behalf of the ring $\mathcal{U}$.*

*Then the Computational Diffie-Hellman problem in $\mathbb{G}_1$ can be solved with probability $\varepsilon' \geq \frac{1}{9Q_1}$ and in time $T' \leq \frac{27\ V_{Q_2,\hat{n}}T}{\varepsilon}$.*

*Proof.* The first comment to be noted is that, since we are in the random oracle model, with overwhelming probability, the attacker $\mathcal{A}$ has asked the random oracle $H_1$ for the identities of all the members of the ring $\mathcal{U}$.

Note also that our ID-based scheme is a suitable generic ring signature scheme, satisfying that any randomness value $R_i \in \mathbb{G}_1^*$ appears in a ring signature with probability less than $\frac{1}{q-1} \leq \frac{2}{2^k}$, as required.

The simulation of an ID-based ring signature for a message $m$ and a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ goes as follows:

1. Choose at random an index $s \in \{1, \ldots, n\}$.
2. For all $i \in \{1, \ldots, n\}$, $i \neq s$, choose $A_i$ at random in $\mathbb{G}_1^*$, pairwise different. Compute $R_i = e(A_i, P)$, for all $i \neq s$.
3. Choose independently and at random $h_1, h_2, \ldots, h_n$ in $\mathbb{Z}_q$.
4. Choose at random $\sigma \in \mathbb{G}_1$.

5. Compute $R_s = e(\sigma - \sum_{i \neq s} A_i, P) \cdot e(-Y, \sum_{i=1}^{n} h_i PK_i)$. If $R_s = 1$ or $R_s = R_i$ for some $i \neq s$, then go to step 4.
6. Return the tuple $(\mathcal{U}, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$.

It is easy to see that this simulation runs in polynomial time. If we impose $H_2(m, R_i) = h_i$ (we are in the random oracle model), for all $i \in \{1, \ldots, n\}$, then the returned tuple is a valid ring signature of the message $m$.

Furthermore, the distribution of ring signatures generated by using the protocol explained in Section 4, and the distribution of ring signatures simulated as above are polynomially indistinguishable.

On the other hand, to simulate consistent pairs of secret-public keys to answer the $Q_e$ extraction queries $\{ID_j\}_{1 \leq j \leq Q_e}$ of the attacker, one chooses random elements $x_j \in \mathbb{Z}_q^*$ and computes, for all $1 \leq j \leq Q_e$, the values $PK_j = x_j P$ and $SK_j = x_j Y$, where $Y$ is the master public key. If we impose $H_1(ID_j) = PK_j$, then the resulting pairs are consistent. Furthermore, the values $PK_j$ are uniformly distributed in $G_1^*$, which has order $\geq 2^k$, as required in the proof of Theorem 2.

Recall that we are assuming that our scheme is not secure: there exists some chosen message and identities attack $\mathcal{A}$ with non-negligible probability of success. Then we can apply Theorem 2 to our ID-based ring signature scheme. The idea is to guess which will be the identity $ID_j$ corresponding to the only member of the ring $\mathcal{U}$ such that $h_j \neq h'_j$ in the two ring signatures obtained from applying Theorem 2. If the guess is correct, we will be able to solve the CDH problem.

In effect, let $(P, aP, bP)$ be the input of an instance of the CDH problem in $\mathbb{G}_1$. We set $Y = aP$. We choose at random a value $\ell \in \{1, 2, \ldots, Q_1\}$. When the attacker $\mathcal{A}$ makes his $\ell$-th query to the random oracle $H_1$, with some identity $ID_\ell$, we impose $PK_\ell = H_1(ID_\ell) = bP$, and send this value to the attacker.

Later, if the attacker would ask for the secret key of $ID_\ell$, then the algorithm solving the CDH problem outputs "fail". For the rest of identities, one can simulate consistent answers to the secret key extraction queries, as explained above.

With probability $1/Q_1$, our guess is correct, and the public key $PK_\ell$ corresponds to the only member $U_j \in \mathcal{U}$ such that $h_j \neq h'_j$ in the two obtained ring signatures. In particular, this means that the attacker has not asked for the secret key matching with $PK_\ell$, and so the CDH-solver has not output "fail".

Summing up, with probability $\varepsilon' \geq \frac{1}{9} \cdot \frac{1}{Q_1}$, we obtain two valid ring signatures $(\mathcal{U}, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(\mathcal{U}, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ such that $h_j \neq h'_j$, for some $j \in \{1, \ldots, n\}$ and $h_i = h'_i$ for all $i = 1, \ldots, n$ such that $i \neq j$. Furthermore, the public key $PK_j$ of user $U_j$ is equal to $PK_\ell = bP$. Then we have that

$$e(\sigma, P) = R_1 \cdot \ldots \cdot R_n \cdot e(Y, h_1 PK_1) \cdot \ldots \cdot e(Y, h_n PK_n)$$

$$e(\sigma', P) = R_1 \cdot \ldots \cdot R_n \cdot e(Y, h'_1 PK_1) \cdot \ldots \cdot e(Y, h'_n PK_n)$$

Dividing these two equations, we obtain $e(\sigma - \sigma', P) = e(Y, (h_j - h'_j)PK_j) = e(aP, (h_j - h'_j)bP) = e(ab(h_j - h'_j)P, P)$. Since the pairing is non-degenerate, this implies that $\sigma - \sigma' = ab(h_j - h'_j)P$. Therefore, we find a solution of the CDH problem by computing

$$abP = \frac{1}{h_j - h'_j}(\sigma - \sigma') \ .$$

□

## 6 Conclusions and Future Work

We have proposed in this work a new ID-based ring signature scheme, based on bilinear pairings. Our scheme is a generic ring signature scheme, according to the definition given in [11]. This allows us to use some security results provided in [11] for this kind of ring schemes.

More specifically, we adapt their results (known as ring forking lemmas, following the work of [12]) to an ID-based scenario. Therefore, we can apply this extended result to our proposed scheme. In this way we prove that it is existentially unforgeable under chosen message and identities attacks, assuming that the Computational Diffie-Hellman problem is hard to solve.

The reduction of this proof is not quite efficient; that is, the relation between both the success probabilities and the execution times of the forger and the CDH-solver algorithms is far to be tight. This is a consequence of the use of the ring forking lemmas. Any improvement in this direction will be very positive from both theoretical and practical points of view.

Another possible line of future research is to design and analyze ring signature schemes in the Certificateless Public Key model, recently introduced in [2].

## References


1. M. Abe, M. Ohkubo and K. Suzuki. $1-$out$-$of$-n$ signatures from a variety of keys. *Advances in Cryptology-Asiacrypt'02*, LNCS **2501**, Springer-Verlag, pp. 415–432 (2002).
2. S. Al-Riyami and K. G. Patterson. Certificateless public key cryptography. *Advances in Cryptology-Asiacrypt'03*, LNCS **2894**, Springer-Verlag, pp. 452-473 (2003).
3. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. *First ACM Conference on Computer and Communications Security*, pp. 62–73 (1993).
4. E. Bresson, J. Stern and M. Szydlo. Threshold Ring Signatures for Ad-hoc Groups. *Advances in Cryptology-Crypto'02*, LNCS **2442**, Springer-Verlag, pp. 465–480 (2002).
5. J. Camenisch. Efficient and generalized group signatures. *Advances in Cryptology-Eurocrypt'97*, LNCS **1233**, Springer-Verlag, pp. 465–479 (1997).
6. D. Chaum and E. van Heyst. Group signatures. *Advances in Cryptology-Eurocrypt'91*, LNCS **547**, Springer-Verlag, pp. 257–265 (1991).

7. R. Cramer, I. Damgård and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. *Advances in Cryptology-Crypto'94*, LNCS **839**, Springer-Verlag, pp. 174–187 (1994).
8. A. De Santis, G. Di Crescenzo, G. Persiano and M. Yung. On monotone formula closure of SZK. *Proceedings of FOCS'94*, IEEE Press, pp. 454–465 (1994).
9. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptative chosen-message attacks. *SIAM Journal of Computing*, **17 (2)**, pp. 281–308 (1988).
10. J. Herranz. A formal proof of security of Zhang and Kim's ID-based ring signature scheme. Technical report (2003).
11. J. Herranz and G. Sáez. Forking lemmas for ring signature schemes. *Proceedings of Indocrypt'03*, LNCS **2904**, Springer-Verlag, pp. 266-279 (2003).
12. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. **13** (3), pp. 361–396 (2000).
13. R. Rivest, A. Shamir and Y. Tauman. How to leak a secret. *Advances in Cryptology-Asiacrypt'01*, LNCS **2248**, Springer-Verlag, pp. 552–565 (2001).
14. A. Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology-Crypto'84*, LNCS **196**, pp. 47–53 (1984).
15. F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. *Advances in Cryptology-Asiacrypt'02*, LNCS **2501**, Springer-Verlag, pp. 533–547 (2002).
16. The Pairing-Based Crypto Lounge. Web page maintained by Paulo Barreto: `http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html`