New Distributed Ring Signatures for General Families of Signing Subsets

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
e-mail: {jherranz,german}@ma4.upc.es

Abstract

In a distributed ring signature scheme, a subset of users cooperate to compute a distributed anonymous signature on a message, on behalf of a family of possible signing subsets. The receiver can verify that the signature comes from a subset of the ring, but he cannot know which subset has actually signed.

In this work we use the concept of dual access structures to construct a distributed ring signature scheme which works with general families of possible signing subsets. The length of each signature is linear on the number of involved users, which is desirable for some families with many possible signing subsets. The scheme achieves the desired properties of correctness, anonymity and unforgeability. The reduction in the proof of unforgeability is tighter than the reduction in the previous proposals which work with general families.

We analyze the case in which our scheme runs in an identity-based scenario, where public keys of the users can be derived from their identities. This fact avoids the necessity of digital certificates, and therefore allows more efficient implementations of such systems. But our scheme can be extended to work in more general scenarios, where users can have different types of keys.

1 Introduction

In standard public key cryptosystems, the public keys of the users must be authenticated via a Public Key Infrastructure (PKI) based on digital certificates, which link the identities of the users with their public keys. This fact makes the use of cryptographic protocols less efficient in the real life.

Shamir introduced in 1984 the concept of *identity-based* (from now on, ID-based) cryptography [19]. 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). In this way, digital certificates are not necessary, because anyone can easily verify that some public key PK_U corresponds in fact to user U. Then, the secret key is derived from the public key in a process 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 clear example of cryptographic schemes where the use of digital certificates dramatically decreases the efficiency of the implementation are *ring signature schemes*, because of the number of public keys that can be involved in any basic operation (signature and verification). In a ring signature scheme, an entity signs a message on behalf of a set of members that includes himself. The verifier of the signature is convinced that it was produced by some member of the set, but he does not obtain any information about which specific member actually signed.

The concept of ring signatures was formally introduced in [17]. After that, many proposals of ring signature schemes have been published [4, 1, 24, 11, 8, 13], for both PKI and ID-based scenarios.

We consider in this work an extension of the concept of ring signature, that we call *distributed ring signature schemes*. Suppose that a subset of users A want to sign some message with a certain anonymity. Members of A freely choose the other users to complete the whole set of users $\mathcal{P}$, and then they choose (in an *ad-hoc* way) a family of subsets $\mathcal{U} \subset 2^{\mathcal{P}}$, which will contain the possible signing subsets. Using their secret keys and the public keys of the rest of users, members of A produce a distributed ring signature. The verifier will be convinced that at least *all* the members of some subset in $\mathcal{U}$ have cooperated to compute the signature, but he will not have any information about which subset in $\mathcal{U}$ is the actual author of the signature.

Distributed ring signature schemes were first considered in [4]. Their specific RSA-based scheme runs only when the ad-hoc families $\mathcal{U}$ are necessarily threshold (that is, they contain all the subsets with a specific number of users). Other proposals that only admit threshold can be found in [23], allowing the use of different types of PKI keys (RSA, based on Discrete Logarithm...) and in [7] for an ID-based framework. With respect to schemes running with more general families $\mathcal{U}$, the two only proposals have appeared in [12], for scenarios based on Discrete Logarithm keys, and in [13], for ID-based scenarios. However, these two proposals are not very efficient for some families $\mathcal{U}$, for example if they contain a lot of subsets.

In this work we propose a new scheme for computing distributed ring signatures on behalf of general families of possible signing subsets. With respect to the two aforementioned schemes [12, 13], the length of a signature in the new scheme is linear in the number of involved users, and not linear in the number of possible signing subsets. This is desirable for some families, for example threshold families, multipartite families, etc. The construction uses the combinatorial concept of dual access structure, and generalizes the threshold proposals in [23, 7]. We first explain, for clarity, the particular case where all users have ID-based keys with common parameters. We prove that the resulting scheme achieves anonymity and unforgeability in the random oracle model, assuming that the Computational Diffie-Hellman problem is hard to solve. Finally, we detail how the scheme can be extended to work in more general scenarios, where users can have different types of keys (either PKI-based or ID-based) with different lengths, using techniques similar to those in [1, 23]. The obtained reduction in the proof of the unforgeability of the scheme is tighter than the reductions obtained in [12, 13].

2 Preliminaries

In this section we review some tools and concepts that will be necessary in the design and analysis of our new distributed ring signature scheme.

2.1 Bilinear 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 *bilinear 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 $e(A_1, A_2 + A_3) = e(A_1, A_2) \cdot e(A_1, A_3)$.

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 $e(A_1, A_2) \neq 1_{\mathbb{G}_2}$.

In particular, property 1 implies that $e(aP, bP) = e(P, P)^{ab} = e(P, abP) = e(abP, P)$, for all $a, b \in \mathbb{Z}_q$. This implies $e(A_1, A_2) = e(A_2, A_1)$, for all $A_1, A_2 \in \mathbb{G}_1$.

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 [25] for a complete bibliography of cryptographic works based on pairings.

2.2 The Computational Diffie-Hellman Problem

We consider the following well-known problem in the additive 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 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 Computational Diffie-Hellman 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 distributed ring signature scheme that we propose in this work is based on the Computational Diffie-Hellman Assumption.

2.3 The Splitting Lemma

We first state a well-known lemma that we will use in a security proof of this paper. A proof of this lemma can be found, for example, in [16].

Lemma 1. *(The Splitting Lemma) Let* $A \subset X \times Y$ *be a set verifying that* $\Pr[(x, y) \in A] \geq \epsilon$*. For any* $\alpha < \epsilon$*, let us define*

$$B = \{(x, y) \in X \times Y | \Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon - \alpha\} \text{ and } \bar{B} = (X \times Y) \backslash B.$$

Then the following statements hold:

1. $\Pr[B] \geq \alpha$.
2. *for any* $(x, y) \in B$, $\Pr_{y' \in Y}[(x, y') \in A] \geq \epsilon - \alpha$.
3. $\Pr[B|A] \geq \alpha/\epsilon$.

2.4 The Random Oracle Model

Bellare and Rogaway introduced in [3] a paradigm that makes easier the task of proving the security of some cryptographic schemes. This paradigm is the *random oracle model.* In this model, hash functions are seen as oracles that produce a truly random value for each new input. Obviously, if the same input is asked twice, then the outputs must be identical.

The random oracle model is unreal, because any instantiation of a hash function is in fact a deterministic function. Although there are some theoretical works which criticize the paradigm of the random oracle model [6, 15, 2], it is widely believed that proofs in this model guarantee the security of the overall cryptographic scheme, provided the employed hash function has no weakness.

All the security results that we prove in this work are valid in the random oracle model.

2.5 Access Structures and their Duals

Some of the concepts that we are going to present arise from the theory of secret sharing schemes. For a survey on this field see [22]. Let us suppose that the subset of users is $\mathcal{P}$. We are going to consider digital signatures where a subset of users sign on behalf of a family of subset of users. An *access structure* $\Gamma \subset 2^{\mathcal{P}}$ is a monotone increasing family of subsets of users verifying that, for any $A_1 \in \Gamma$ and $A_2 \subset \mathcal{P}$ such that $A_1 \subset A_2$, then $A_2 \in \Gamma$. Therefore, an access structure can be determined by the family $\Gamma_0 \subset \Gamma$ of minimal subsets in Γ, which is called the *basis* of Γ. For an arbitrary family of subsets $\mathcal{U} \subset 2^{\mathcal{P}}$ the *closure* of $\mathcal{U}$ is the minimum monotone access structure that contains $\mathcal{U}$, that is $cl(\mathcal{U}) = \{A \subset \mathcal{P} :$ there exists $B \in \mathcal{U}$ such that $B \subset A\}$. Of course for a monotone access structure Γ we have $\Gamma = cl(\Gamma_0)$.

We will assume that the families of subsets $\mathcal{U}$ considered in this work are in some way *normalized*: there do not exist two subsets $A, B \in \mathcal{U}$ such that $A \subset B$. In this case, it is easy to see that $(cl(\mathcal{U}))_0 = \mathcal{U}$; that is, $\mathcal{U}$ is the basis of its closure.

For an access structure Γ, the *dual* of Γ is defined as $\Gamma^* = \{\mathcal{P} - A : A \notin \Gamma\}$ and it is also a monotone access structure (see [14] for more details on dual access structures). A basic property of the dual is that $(\Gamma^*)^* = \Gamma$; it is also easy to see, by the definition of Γ^*, that $A \in \Gamma_0$ if and only if $\mathcal{P} - A$ is a maximal subset verifying $\mathcal{P} - A \notin \Gamma^*$.

A useful family of monotone access structures is the vector space access structure due to Brickell [5]. Let Γ be an access structure on a set of participants $\mathcal{P}$ and $D \notin \mathcal{P}$ a special participant called the *dealer.* Γ is said to be a *vector space access structure* if, for some vector space $GF(q)^r$ over a finite field $GF(q)$, there exists a function

$$\psi : \mathcal{P} \cup \{D\} \longrightarrow GF(q)^r$$

such that $A \in \Gamma$ if and only if the vector $\psi(D)$ can be expressed as a linear combination of the vectors in the set $\psi(A) = \{\psi(U) \mid U \in A\}$. An example of vector space access structure are *threshold access structure*, introduced by Shamir in his seminal paper on secret sharing [18]. These access structures are defined as $\Gamma = \{A \subset \mathcal{P} : |A| \geq t\}$ where t is the *threshold.* In effect, threshold access structures are vector space access structures considering $\psi : \mathcal{P} \cup \{D\} \longrightarrow GF(q)^t$ defined by $\psi(D) = (1, 0, \ldots, 0)$ and $\psi(U_i) = (1, i, i^2, \ldots, i^{t-1})$ where $\mathcal{P} = \{U_1, U_2, \ldots, U_\ell\}$.

It is not difficult to prove [14] that, if Γ is a vector space access structure, then Γ^* is also a vector space access structure.

Not all the access structures can be expressed as vector space access structures. Simmons, Jackson and Martin [21] proved that any access structure Γ can be in fact expressed in a similar way where every participant can be associated with more than one vector. The construction that they presented is based on the use of the dual access structure and it is as follows. Let us suppose that the structure Γ is such that $(\Gamma^*)_0 = \{A_1, \ldots, A_d\}$, then ψ assigns vectors in $GF(q)^d$ in the following way: $\psi(D) = (1, 0, \ldots, 0)$ and $\psi(U) = \{(1, i, i^2, \ldots, i^{d-1}) : U \in A_i\}$ for any user $U \in \mathcal{P}$. This assignment ψ realizes the access structure Γ.

3 Distributed Ring Signatures

A distributed ring signature scheme consists of three protocols:

1. **Key generation.** This protocol is executed individually by each user U_i of the system. The input is a security parameter and (possibly) some public parameters, common to all the users of the system. The output consists of a public key PK_i, that the user U_i makes public, and a secret key SK_i, that U_i keeps secret. In ID-based scenarios, this protocol is executed with the help of a master entity.

2. **Distributed ring signature generation.** Suppose users in a subset A want to compute a ring signature on a message m on behalf of a family $\mathcal{U}$ of d subsets, such that $A \in \mathcal{U}$. Then members of A jointly execute this protocol, taking as input the message m, the public keys of all users included in the family $\mathcal{U}$ and their own secret keys $\{SK_j\}_{U_j \in A}$. The output is a signature θ.

3. **Verification of a distributed ring signature.** The recipient of a distributed ring signature checks its validity by running this protocol. It takes as input the message m, the signature θ and the public keys of all the users in $\mathcal{U}$. The output is 1 if the signature is valid, and 0 if it is invalid.

Note that distributed ring signature schemes are related to standard distributed (or threshold) signature schemes [10, 20]. In both cases, the recipient of the signature is convinced that all the users in some subset of a specific family have jointly signed the message, but he does not know which is the signing subset. The two main differences between these two types of signatures are the following: (i) in standard distributed signatures, the family of possible signing subsets is fixed *a priori* for all the life of the system (it is called the access structure of the scheme), whereas in ring signatures it is chosen *ad-hoc* by the signing users, just before signing; (ii) in standard schemes, there is a unique public key for the whole set of users, and the matching secret key is shared among them, whereas in distributed ring schemes, each user has his own public and secret keys, that can be used as well for other purposes.

With respect to the distributed ring signature schemes proposed until now, either they work only for threshold families, which contain all the subsets with a specific number of users [4, 23, 7], or they admit more general families [12, 13] but are not very efficient when the number of subsets in the family is very large.

3.1 Security Requirements

A distributed ring signature scheme must satisfy three properties, that we informally describe below.

1. **Correctness:** if a distributed ring signature is generated by properly following the protocol, then the result of the verification is always 1.

2. **Anonymity:** any verifier should not have probability greater than $1/d$ to guess the identity of the subset which has actually computed a distributed ring signature on behalf of a signing family which contains d subsets.

3. **Unforgeability:** among all the proposed definitions of unforgeability, we consider the strongest one, *existential unforgeability against adaptive chosen message attacks*, adapted to the scenario of distributed ring signatures. Roughly speaking, an attacker should not be able to obtain a valid distributed ring signature for a message m and a family of possible signing users $\mathcal{U}$, unless he has already asked for a valid signature for this pair $(m,\mathcal{U})$ or he has corrupted all the users of some of the subsets in $\mathcal{U}$.

4 The New Proposal

We next propose a new scheme for computing distributed ring signatures, which works with general families of possible signing subsets. The proposal is based on the concept of dual access structure, and extends the scheme designed in [23] for the threshold case. We explain and analyze in this section, for clarity, the particular version where all users have ID-based keys with common parameters. In Section 6, we detail how the scheme can be extended to separable scenarios where users have independent keys (either PKI-based or ID-based, with different sizes, ...).

The protocols of our proposed scheme are described below.

Key generation: 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 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear pairing as defined in Section 2.1. Let $H_1 : \{0,1\}^* \to \mathbb{G}_1^*$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q$ be two hash functions.

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

Secret key extraction: any user U_i of the system, with identity ID_i (which can be an IP or e-mail address, for example), has public key $PK_i = H_1(ID_i)$. When he requests the master for his matching secret key, he obtains the value $SK_i = xPK_i$.

Distributed ring signature generation: assume that a subset of users A want to compute an anonymous signature on behalf of a family $\mathcal{U} \subset 2^{\mathcal{P}}$ of possible signing subsets, taken over a set $\mathcal{P} = \{U_1, \ldots, U_\ell\}$ of ℓ users. Users in A choose the family $\mathcal{U}$ in an ad-hoc way, with the only condition that $A \in \mathcal{U}$. We will consider that any specific set of users can always have access to a private and authenticated broadcast channel; this can be achieved, for example, by using broadcast encryption schemes [9].

For simplicity, we will assume that $cl(\mathcal{U})$ is a vector space access structure. In this case, we consider $\Gamma = (cl(\mathcal{U}))^*$, which is also a vector space access structure: there exist a positive integer r and a mapping $\psi : \mathcal{P} \cup \{D\} \to \mathbb{Z}_q^r$ such that $B \in \Gamma$ if and only if $\psi(D) \in \langle \{\psi(U_i)\}_{U_i \in B} \rangle$. Our construction can be easily extended to the case of more general access structures, where the mapping ψ assigns possibly more than one vector to some users. For example, a generic solution would be to use the construction of Simmons et al. [21]: if $\mathcal{U} = \{A_1, \ldots, A_d\}$, then ψ assigns vectors in $GF(q)^d$ in the following way: $\psi(D) = (1, 0, \ldots, 0)$ and $\psi(U) = \{(1, i, i^2, \ldots, i^{d-1}) : U \in A_i\}$ for any user $U \in \mathcal{P}$. This assignment ψ realizes the access structure $\Gamma = (cl(\mathcal{U}))^*$.

We assume that the family $\mathcal{U}$ is normalized, so $\mathcal{U} = (cl(\mathcal{U}))_0$. Therefore, we have that $A \in (cl(\mathcal{U}))_0$. This means that $\mathcal{P} - A \notin \Gamma$, and is maximal with respect to the inclusion, meaning that $(\mathcal{P} - A) \cup \{U_j\} \in \Gamma$ for any user $U_j \in A$.

The signing users in A execute the following protocol to compute a valid distributed signature on a message $m \in \{0,1\}^*$:

1. They consider a basis of the subspace $\langle \psi(\mathcal{P} - A) \rangle$. This basis corresponds to some subset of users $C \subset \mathcal{P} - A$; that is, vectors in $\psi(C)$ are linearly independent and $\langle \psi(C) \rangle = \langle \psi(\mathcal{P} - A) \rangle$.

2. For every user $U_i \in C$, the signing users choose uniformly at random $c_i \in \mathbb{Z}_q$

and $R_i \in \mathbb{G}_1$; they compute and broadcast the value

$$z_i = e(R_i, P) \cdot e(Y, c_i PK_i).$$

3. For users $U_t \in (\mathcal{P} - A) - C$, we have that $\psi(U_t) = \sum_{U_i \in C} \lambda_{it} \psi(U_i)$, for some $\lambda_{it} \in \mathbb{Z}_q$, because $\psi(C)$ is a basis of $\langle \psi(\mathcal{P} - A) \rangle$. The signing users choose uniformly at random $R_t \in \mathbb{G}_1$ and consider $c_t = \sum_{U_i \in C} \lambda_{it} c_i$; they compute and broadcast the value

$$z_t = e(R_t, P) \cdot e(Y, c_t PK_t).$$

4. Each signing user $U_j \in A$ chooses uniformly at random $T_j \in \mathbb{G}_1$; he computes and broadcasts the value

$$z_j = e(T_j, P).$$

5. The signing users compute then the value $c = H_2(\mathcal{U}, m, z_1, \ldots, z_\ell)$.

6. They choose uniformly at random one of the vectors $\mathbf{v} \in \mathbb{Z}_q^r$ that verifies:

   (i) $\mathbf{v}\psi(D) = c$, and

   (ii) $\mathbf{v}\psi(U_i) = c_i$, for all $U_i \in C$.

   Note that this vector $\mathbf{v}$ exists because $C \notin \Gamma$ and so the vectors $\{\psi(D), \{\psi(U_i)\}_{U_i \in C}\}$ are linearly independent.

7. Every signing user $U_j \in A$ individually computes $c_j = \mathbf{v}\psi(U_j)$; then he computes and broadcasts the value

$$R_j = T_j - c_j SK_j.$$

   Note that the rest of users in A can verify that this value R_j is consistent with the value z_j broadcast in step 4, by checking if $z_j = e(R_j, P) \cdot e(Y, c_j PK_j)$. In this way, they detect dishonest users who try to boycott the process.

8. The resulting signature is $(\mathcal{U}, m, \mathbf{v}, R_1, \ldots, R_\ell, \psi)$.

Note that the length of the signature is linear with respect to the number ℓ of users.

Verification of a distributed ring signature: the recipient of the message first computes $c_i = \mathbf{v}\psi(U_i)$, for every user $U_i \in \mathcal{P}$ and then computes the values

$$z_i = e(R_i, P) \cdot e(Y, c_i PK_i).$$

The signature is valid if $\mathbf{v}\psi(D) = H_2(\mathcal{U}, m, z_1, \ldots, z_\ell)$.

5 Analysis of the Scheme

In this section we prove that our new scheme satisfies the three required properties for distributed ring signature schemes: correctness, anonymity and unforgeability. The two last properties are proved to be achieved in the random oracle model.

5.1 Correctness of the Scheme

We show that a signature that has been generated following the above method is always valid. The vector $\mathbf{v}$ in the signature satisfies

(i) $\mathbf{v}\psi(D) = c$, and

(ii) $\mathbf{v}\psi(U_i) = c_i$, for all $U_i \in C$.

Therefore, for users U_i in the set C, we have that $c_i = \mathbf{v}\psi(U_i)$ and $z_i = e(R_i, P) \cdot e(Y, c_i PK_i)$.

For users $U_t \in (\mathcal{P} - A) - C$, we have that $\psi(U_t) = \sum_{U_i \in C} \lambda_{it}\psi(U_i)$, by definition of the set C. This implies that

$$c_t = \sum_{U_i \in C} \lambda_{it} c_i = \sum_{U_i \in C} \lambda_{it} \mathbf{v}\psi(U_i) = \mathbf{v}\psi(U_t).$$

And $z_t = e(R_t, P) \cdot e(Y, c_t PK_t)$ for these users, as well.

Finally let us consider users $U_j \in A$. By construction, the equality $c_j = \mathbf{v}\psi(U_j)$ is also satisfied. Note that these values are independent of the choice of the vector $\mathbf{v}$, as long as it satisfies the two required conditions. In effect, as far as $\langle\psi(C)\rangle = \langle\psi(\mathcal{P} - A)\rangle$ and $\mathcal{P} - A$ is maximal verifying $\mathcal{P} - A \notin \Gamma$, then $C \cup \{U_j\} \in \Gamma$, for any user $U_j \in A$. So there exist coefficients λ_j and $\{\lambda_{ji}\}_{U_i \in C}$ satisfying

$$\psi(D) = \sum_{U_i \in C} \lambda_{ji}\psi(U_i) \ + \ \lambda_j \psi(U_j)$$

where $\lambda_j \neq 0$. From this equality we can derive

$$c_j = \mathbf{v}\psi(U_j) = \lambda_j^{-1}\left(\mathbf{v}\psi(D) - \sum_{U_i \in C} \lambda_{ji}\mathbf{v}\psi(U_i)\right) = \lambda_j^{-1}\left(c - \sum_{U_i \in C} \lambda_{ji} c_i\right),$$

which does not depend on the specific vector $\mathbf{v}$.

Furthermore, for users $U_j \in A$ we have that

$$z_j = e(T_j, P) = e(R_j + c_j SK_j, P) = e(R_j, P) \cdot e(c_j x PK_j, P) = e(R_j, P) \cdot e(c_j PK_j, Y),$$

as desired.

Therefore, for all users U_i in $\mathcal{P}$ we have that $c_i = \mathbf{v}\psi(U_i)$ and $z_i = e(R_i, P) \cdot e(Y, c_i PK_i)$, and so the correctness of the signature is verified because $\mathbf{v}\psi(D) = c = H_2(\mathcal{U}, m, z_1, \ldots, z_\ell)$.

5.2 Anonymity of the Scheme

Given a valid distributed ring signature $Sig = (\mathcal{U}, m, \mathbf{v}, R_1, \ldots, R_\ell, \psi)$ on behalf of a family of subsets of users $\mathcal{U}$, the probability that a particular subset $B \in \mathcal{U}$ is the author of this signature can be exactly computed. If the full set of users is $\mathcal{P}$, we know that $\psi : \mathcal{P} \cup \{D\} \rightarrow \mathbb{Z}_q^r$ is a mapping which defines the access structure

$\Gamma = (cl(\mathcal{U}))^*$. Since $B \in \mathcal{U} = (cl(\mathcal{U}))_0$, we have that $\mathcal{P} - B \notin \Gamma$. Let $C \subset \mathcal{P} - B$ be a subset of users such that $\langle \psi(C) \rangle = \langle \psi(\mathcal{P} - B) \rangle$ and such that the vectors in $\{\psi(U_i)\}_{U_i \in C}$ are linearly independent. Since $C \notin \Gamma$, we have that the set of vectors $\{\psi(D), \{\psi(U_i)\}_{U_i \in C}\}$ are linearly independent in $\mathbb{Z}_q^r$. Therefore, the number of users in C is $\omega = |C| \leq r - 1$.

Consider the values $c = \mathbf{v}\psi(D)$ and $c_i = \mathbf{v}\psi(U_i)$, for all users $U_i \in C$. The probability that users in B choose these values $\{c_i\}_{U_i \in C}$ in step 2 of the signing protocol is exactly $1/q^\omega$. Later, the value c is the output of the hash function H_2. If we assume that this hash function behaves as a random oracle, then the probability that users in B obtain this value c in step 5 of the protocol is exactly $1/q$, independently of the inputs taken by the hash function.

After that, users in B would choose at random one vector among the solutions of the system of equations $M\mathbf{x} = \mathbf{b}$, where

$$M = \begin{pmatrix} \cdots & \psi(D) & \cdots \\ \cdots & \psi(U_{i_1}) & \cdots \\ \vdots & \vdots & \vdots \\ \cdots & \psi(U_{i_\omega}) & \cdots \end{pmatrix} \qquad b = \begin{pmatrix} c \\ c_{i_1} \\ \vdots \\ c_{i_\omega} \end{pmatrix},$$

if we denote $C = \{U_{i_1}, \ldots, U_{i_\omega}\}$.

The number of different vectors in $\mathbb{Z}_q^r$ which are solution of this system is q^γ, where $\gamma = \dim(\ker M) = r - \dim(\mathrm{Im} M) = r - (\omega + 1)$. Therefore, the probability that users in B choose in step 6 of the protocol the vector $\mathbf{v}$ that appears in Sig is exactly $1/q^\gamma$.

The probability that members of B choose, in steps 2 and 3 of the signing protocol, the values $\{R_i\}_{U_i \notin B}$ that appear in Sig and, in step 4, the values $\{T_j\}_{U_j \in B}$ that lead to the values $\{R_j\}_{U_j \in B}$ in Sig is exactly equal to $1/q^\ell$.

Summing up, the probability that users in B obtain the signature Sig when they execute the signing protocol is exactly

$$\frac{1}{q^\omega} \cdot \frac{1}{q} \cdot \frac{1}{q^\gamma} \cdot \frac{1}{q^\ell} = \frac{1}{q^{\omega+1+\gamma+\ell}} = \frac{1}{q^{r+\ell}},$$

which does not depend on B and so is the same for all the subsets in the family $\mathcal{U}$. This proves that the scheme is unconditionally anonymous, in the random oracle model for the hash function H_2.

5.3 Unforgeability of the Scheme

We will analyze the exact unforgeability of our scheme, that measures all the resources and performances of an adversary against it. The analysis is done in the random oracle model.

Such an adversary is allowed to adaptively corrupt up to Q_e users, obtaining their secret keys. The adversary can also make Q_1 queries to the random oracle H_1 and Q_2 queries to the random oracle H_2. Finally, the adversary can require the execution of the signing algorithm for Q_s pairs of messages and families of subsets that it adaptively chooses, obtaining a valid distributed ring signature for each query.

We say that this adversary is $(T, \varepsilon, Q_1, Q_2, Q_e, Q_s)$-*successful* if it obtains in time T and with probability ε a valid ring signature for some message m and some family of subsets $\mathcal{U}$, such that:

(i) the pair formed by the message m and the family $\mathcal{U}$ has not been asked to the signing oracle during the attack; and

(ii) all the subsets in the family $\mathcal{U}$ contain at least one user who has not been corrupted by the adversary.

Finally, we say that a distributed ring signature scheme is $(T, \varepsilon, Q_1, Q_2, Q_e, Q_s)$-*unforgeable* if there does not exist any $(T, \varepsilon, Q_1, Q_2, Q_e, Q_s)$-successful adversary against it.

In the following theorem, we relate the unforgeability of our scheme to the difficulty of solving the Computational Diffie-Hellman problem.

Theorem 1. *Let $\mathcal{A}$ be a $(T, \varepsilon, Q_1, Q_2, Q_e, Q_s)$-successful adversary against the proposed ID-based ring signature scheme, such that the success probability ε of $\mathcal{A}$ is non-negligible in the security parameter $k \geq 9$, and such that $Q_s \leq \frac{2^{k/2}}{4}$ and $Q_2 \leq \frac{2^{k/2}}{3}$.*

Then the Computational Diffie-Hellman problem in $\mathbb{G}_1$ can be solved with probability $\varepsilon' \geq \frac{\varepsilon^2}{385 Q_e Q_2}$ and in time $T' \leq 2T + 2Q_1 + 2Q_2 + 2T_\psi Q_s$, where T_ψ is the expected time to perform some computations related to the access structure defined by the assignment of vectors ψ.

Proof. We are going to construct a probabilistic polynomial time Turing machine $\mathcal{F}$ which will use the attacker $\mathcal{A}$ as a sub-routine in order to solve the given instance of the Computational Diffie-Hellman problem. Therefore, $\mathcal{F}$ must perfectly simulate the environment of the attacker $\mathcal{A}$.

The machine $\mathcal{F}$ receives the public data (P, aP, bP), and its goal is to compute the value abP. The public key of the master entity is defined to be $Y = aP$. Then $\mathcal{F}$ runs the attacker $\mathcal{A}$ against the threshold ID-based ring signature scheme, answering to all the queries that $\mathcal{A}$ makes. The public key $Y = aP$ is also sent to the attacker $\mathcal{A}$.

Without loss of generality, we can assume that $\mathcal{A}$ asks the random oracle H_1 for the value $H_1(ID)$ before asking for the secret key of ID.

Let us define $\mu = (5/6)^{1/Q_e}$ (we assume $Q_e \geq 1$; otherwise, we would take $\mu = 0$).

The machine $\mathcal{F}$ constructs a table TAB_{H_1} to simulate the random oracle H_1. Every time an identity ID_i is asked by $\mathcal{A}$ to the oracle H_1, the machine $\mathcal{F}$ acts as follows: first $\mathcal{F}$ checks if this input is already in the table; if this is the case, then $\mathcal{F}$ sends to $\mathcal{A}$ the corresponding relation $H_1(ID_i) = PK_i$. Otherwise, with probability μ, the machine $\mathcal{F}$ chooses the bit $d_i = 0$ and a different $x_i \in \mathbb{Z}_q^*$ at random, and defines $PK_i = x_i P$ and $SK_i = x_i Y$; the new entry $(ID_i, PK_i, x_i, SK_i, d_i)$ is stored in the table TAB_{H_1}. On the other hand, with probability $1-\mu$, the machine $\mathcal{F}$ chooses the bit $d_i = 1$ and a different $\alpha_i \in \mathbb{Z}_q^*$ at random, and defines $PK_i = (\alpha_i)bP$ (in this case $\mathcal{F}$ does not know the secret key for this identity). The values $(ID_i, PK_i, \alpha_i, d_i)$ are stored in a new entry of TAB_{H_1}, and the relation $H_1(ID_i) = PK_i$ is sent to $\mathcal{A}$.

The condition $PK_i \neq PK_j$ must be satisfied for all the different entries $i \neq j$ of the table; if this is not the case, the process is repeated for one of these users.

Since we are assuming that H_1 behaves as a random function, and the values PK_i are all randomly chosen, this simulation of the hash function H_1 is consistent.

Later, every time $\mathcal{A}$ asks for the secret key corresponding to an identity ID_i, the machine $\mathcal{F}$ looks for ID_i in the table TAB_{H_1}. If $d_i = 0$, then $\mathcal{F}$ sends $SK_i = x_i Y$ to $\mathcal{A}$. If $d_i = 1$, the machine $\mathcal{F}$ cannot answer and halts. The probability that $\mathcal{F}$ halts in this process is less than $1 - \mu^{Q_e} = 1/6$.

As well, $\mathcal{F}$ constructs a table TAB_{H_2} to simulate the random oracle H_2. Every time $\mathcal{A}$ makes a query to this oracle, $\mathcal{F}$ looks for this value in the table. If it is already there, then $\mathcal{F}$ sends the corresponding relation to $\mathcal{A}$; if not, $\mathcal{F}$ chooses at random an output of the random oracle for the queried input, different from the outputs which are already in the table, sends the relation to $\mathcal{A}$ and stores it in the table TAB_{H_2}.

Finally, the attacker $\mathcal{A}$ can ask Q_s times for valid distributed ring signatures for messages m' and families of subsets $\mathcal{U}'$, where the full set of ℓ' users is $\mathcal{P}'$. To answer such queries, the machine $\mathcal{F}$ proceeds as follows:

1. Define $\Gamma = (cl(\mathcal{U}))^*$; then find a mapping $\psi' : \mathcal{P}' \cup \{D\} \to \mathbb{Z}_q^{r'}$ such that $B \in \Gamma$ if and only if $\psi'(D) \in \langle \psi'(B) \rangle$. Then choose a subset $A \in \mathcal{U}$; consider a basis of the subspace $\langle \psi'(\mathcal{P}' - A) \rangle$. This basis corresponds to some subset of users $C \subset \mathcal{P}' - A$.

2. For every user $U_i \in C$, choose uniformly at random $c'_i \in \mathbb{Z}_q$. Choose uniformly at random a value $c' \in \mathbb{Z}_q$.

3. Choose at random a vector $\mathbf{v}'$ among the set of vectors $\mathbf{v}$ satisfying $\mathbf{v}\psi'(D) = c'$ and $\mathbf{v}\psi'(U_i) = c'_i$ for all users $U_i \in C$.

4. For users $U_j \in \mathcal{P}' - C$, compute the values $c'_j = \mathbf{v}'\psi'(U_j)$.

5. Choose at random ℓ' values $R'_1, \dots, R'_{\ell'} \in \mathbb{G}_1$, one for each user in $\mathcal{P}'$.

6. Compute, for $i = 1, \dots, \ell'$, the values $z'_i = e(R'_i, P) \cdot e(Y, c'_i PK_i)$.

7. Impose and store in the table TAB_{H_2} the new relation $H_2(\mathcal{U}', m', z'_1, \dots, z'_{\ell'}) = c'$.

8. Define the signature to be $(\mathcal{U}', m', \mathbf{v}', R'_1, \dots, R'_{\ell'}, \psi')$.

In each simulation, the machine $\mathcal{F}$ must find a suitable assignment ψ, choose at random some values, then choose a vector $\mathbf{v}$, perform ℓ' evaluations of the bilinear pairing, etc. We denote as T_ψ a bound for the expected time necessary for performing all these tasks.

The process results in a valid distributed ring signature, because we are assuming that H_2 behaves as a random function, and c' is taken uniformly at random in $\mathbb{Z}_q$. However, the assignment $H_2(\mathcal{U}', m', z'_1, \dots, z'_{\ell'}) = c'$ can produce some collisions in the management of the table TAB_{H_2} that simulates the random oracle H_2.

A first possible collision occurs if a tuple $(\mathcal{U}', m', z_1', \dots, z_{\ell'}')$ produced in the simulation of a signature has been already queried to the random oracle H_2. The probability of this event is less than $\frac{Q_s Q_2}{q} \leq 1/12$.

A second possible collision occurs when the same tuple $(\mathcal{U}', m', z_1', \dots, z_{\ell'}')$ is produced in two different signature simulations. The probability of this event is less than $\frac{Q_s^2}{2q} \leq 1/12$.

We denote by ω the whole set of random tapes that take part in an attack by $\mathcal{A}$, with the environment simulated by $\mathcal{F}$, but excluding the randomness related to the oracle H_2. The success probability of $\mathcal{A}$ in forging a valid ring signature scheme is then taken over the space (ω, H_2).

In an execution of the attacker $\mathcal{A}$, we use the notation $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_{Q_2}$ for the different queries that $\mathcal{A}$ makes to the random oracle H_2. If $\mathcal{A}$ produces a valid forged signature $(\mathcal{U}, m, \mathbf{v}, R_1, \dots, R_\ell, \psi)$, by the ideal randomness of the oracle H_2, the probability that $\mathcal{A}$ has not asked to this oracle for the corresponding tuple $(\mathcal{U}, m, z_1, \dots, z_\ell)$, and so $\mathcal{A}$ must have guessed the corresponding output, is less than $\frac{1}{q}$. We define $\beta = \infty$ in this case; otherwise, β denotes the index of the query where the tuple above was asked. That is, $\mathcal{Q}_\beta = (\mathcal{U}, m, z_1, \dots, z_\ell)$.

We denote by $\mathcal{S}$ the set of successful executions of $\mathcal{A}$, with $\mathcal{F}$ simulating its environment, and such that $\beta \neq \infty$. We also define the following subsets of $\mathcal{S}$: for every $i = 1, 2, \dots, Q_2$, the set $\mathcal{S}_i$ contains the successful executions such that $\beta = i$.

This gives us a partition $\{\mathcal{S}_i\}_{i=1,\dots,Q_2}$ of $\mathcal{S}$ in exactly Q_2 classes.

The probability that an execution (ω, H_2) of $\mathcal{A}$ with the environment simulated by $\mathcal{F}$ results in a valid forgery with $\beta \neq \infty$ is

$$\tilde{\varepsilon} = \Pr[(\omega, H_2) \in \mathcal{S}] \geq \varepsilon \left(1 - \frac{1}{q}\right) \left(1 - (1 - \mu^{Q_e}) - \frac{Q_s Q_2}{q} - \frac{Q_s^2}{2q}\right) \geq$$

$$\geq \varepsilon \cdot \frac{3}{4} \cdot \left(1 - \frac{1}{3}\right) = \frac{\varepsilon}{2}.$$

Now we define the set of indexes which are more likely to appear as

$$I = \{i \text{ s.t. } \Pr[(\omega, H_2) \in \mathcal{S}_i \mid (\omega, H_2) \in \mathcal{S}] \geq \frac{1}{2Q_2}\}.$$

And the corresponding subset of successful executions as $\mathcal{S}_I = \{(\omega, H_2) \in \mathcal{S}_i \text{ s.t. } i \in I\}$.

For a specific index $i \in I$, we have that

$$\Pr[(\omega, H_2) \in \mathcal{S}_i] = \Pr[(\omega, H_2) \in \mathcal{S}] \cdot \Pr[(\omega, H_2) \in \mathcal{S}_i \mid (\omega, H_2) \in \mathcal{S}] \geq$$

$$\geq \tilde{\varepsilon} \cdot \frac{1}{2Q_2}.$$

Lemma 2. *It holds that* $\Pr[(\omega, H_2) \in \mathcal{S}_I \mid (\omega, H_2) \in \mathcal{S}] \geq 1/2$.

Proof. Since the sets $\mathcal{S}_i$ are disjoint, we have

$$\Pr[(\omega, H_2) \in \mathcal{S}_I \mid (\omega, H_2) \in \mathcal{S}] = \sum_{i \in I} \Pr[(\omega, H_2) \in \mathcal{S}_i \mid (\omega, H_2) \in \mathcal{S}] =$$

$$1 - \sum_{i \notin I} \Pr[(\omega, H_2) \in \mathcal{S}_i \mid (\omega, H_2) \in \mathcal{S}].$$

Since the complement of I contains at most Q_2 indexes, we have that this probability is greater than $1 - Q_2 \cdot \frac{1}{2Q_2} = 1/2$. □

We come back to the execution of $\mathcal{A}$ with the environment simulated by $\mathcal{F}$. With probability at least $\tilde{\varepsilon}$, such an execution (ω, H_2) results in a valid forgery with $\beta \neq \infty$. In this case, applying Lemma 2, we know that this successful execution belongs to $\mathcal{S}_I$ with probability at least $1/2$.

Now we split H_2 as (H_2', c), where H_2' corresponds to the answers of all the queries to H_2 except the query $\mathcal{Q}_\beta$, whose answer is denoted as c.

We apply the Splitting Lemma (Lemma 1), taking $X = (\omega, H_2')$, $Y = c$, $A = \mathcal{S}_\beta$, $\delta = \frac{\tilde{\varepsilon}}{2Q_2}$ and $\alpha = \frac{\tilde{\varepsilon}}{4Q_2}$. The lemma says that there exists a subset of executions Ω_β such that

$$\Pr[(\omega, H_2) \in \Omega_\beta \mid (\omega, H_2) \in \mathcal{S}_\beta] \geq \frac{\alpha}{\delta} = \frac{1}{2}$$

and such that, for any $(\omega, H_2) \in \Omega_\beta$:

$$\Pr_{\tilde{c}}[(\omega, H_2', \tilde{c}) \in \mathcal{S}_\beta] \geq \delta - \alpha = \frac{\tilde{\varepsilon}}{4Q_2}.$$

With probability at least $\frac{\tilde{\varepsilon}}{2}$, the first execution (ω, H_2', c) of $\mathcal{A}$ simulated by $\mathcal{F}$ is successful and the index β belongs to the set I. Furthermore, in this case we have that $(\omega, H_2', c) \in \Omega_\beta$ with probability at least $1/2$. If we now repeat this simulated execution of $\mathcal{A}$ with fixed (ω, H_2') and randomly chosen $\tilde{c} \in \mathbb{Z}_q$, we know that $(\omega, H_2', \tilde{c}) \in \mathcal{S}_\beta$ and furthermore $\tilde{c} \neq c$ with probability at least $\frac{\tilde{\varepsilon}}{4Q_2}\left(1 - \frac{1}{q}\right) \geq \frac{\tilde{\varepsilon}}{5Q_2}$ (because, in particular, we know that $q \geq 5$).

Now consider the two successful executions of the attack, (ω, H_2', c) and $(\omega, H_2, \tilde{c})$, that the algorithm $\mathcal{F}$ has obtained by executing the attack $\mathcal{A}$. We denote by $(\mathcal{U}, m, \mathbf{v}, R_1, \ldots, R_\ell, \psi)$ and $(\tilde{\mathcal{U}}, \tilde{m}, \tilde{\mathbf{v}}, \tilde{R}_1, \ldots, \tilde{R}_\ell, \tilde{\psi})$, respectively, the two forged distributed ring signatures. Since the random tapes and H_1 are identical, and the answers of the random oracle H_2 are the same until the query $\mathcal{Q}_\beta = (\mathcal{U}, m, z_1, \ldots, z_\ell)$, we have in particular that $\tilde{\mathcal{U}} = \mathcal{U}$, $\tilde{\psi} = \psi$, $\tilde{m} = m$ and $\tilde{z}_i = z_i$, for $i = 1, \ldots, \ell$ (the whole set of ℓ users is denoted by $\mathcal{P}$).

Let us define the subset $B = \{U_i \in \mathcal{P} : \mathbf{v}\psi(U_i) = \tilde{\mathbf{v}}\psi(U_i)\}$. Since $\mathbf{v}\psi(D) = c \neq \tilde{c} = \tilde{\mathbf{v}}\psi(D)$ then B cannot be in Γ. Otherwise, if $B \in \Gamma$ then there would exist coefficients $\lambda_i \in \mathbb{Z}_q$ for users $U_i \in B$ satisfying $\psi(D) = \sum_{U_i \in B} \lambda_i \psi(U_i)$. This would imply

$$c = \mathbf{v}\psi(D) = \sum_{U_i \in B} \lambda_i \mathbf{v}\psi(U_i) = \sum_{U_i \in B} \lambda_i \tilde{\mathbf{v}}\psi(U_i) = \tilde{\mathbf{v}} \sum_{U_i \in B} \lambda_i \psi(U_i) = \tilde{\mathbf{v}}\psi(D) = \tilde{c},$$

a contradiction. Therefore we must have $B \notin \Gamma$, and so $\mathcal{P} - B \in \Gamma^* = cl(\mathcal{U})$; in other words, $A = \mathcal{P} - B = \{U_j \in \mathcal{P} : \mathbf{v}\psi(U_j) \neq \tilde{\mathbf{v}}\psi(U_j)\} \in cl(\mathcal{U})$.

By definition of successful forgery, there must exist some user $U_j \in A$, satisfying $c_j = \mathbf{v}\psi(U_j) \neq \tilde{\mathbf{v}}\psi(U_j) = \tilde{c}_j$, whose secret key has not been asked by the attacker $\mathcal{A}$. In this case, with probability $1 - \mu$ we have $d_j = 1$ and so $PK_j = \alpha_j bP$.

The equality $z_j = \tilde{z}_j$ becomes $e(R_j, P) \cdot e(Y, c_j PK_j) = e(\tilde{R}_j, P) \cdot e(Y, \tilde{c}_j PK_j)$. This is equivalent to

$$e(R_j - \tilde{R}_j, P) = e(Y, (\tilde{c}_j - c_j)PK_j) = e(aP, (\tilde{c}_j - c_j)\alpha_j bP) = e(a(\tilde{c}_j - c_j)\alpha_j bP, P).$$

This implies that $R_j - \tilde{R}_j = a(\tilde{c}_j - c_j)\alpha_j bP$. Therefore, the machine $\mathcal{F}$ obtains the solution of the given instance of the Computational Diffie-Hellman problem as

$$abP = \frac{1}{(\tilde{c}_j - c_j)\alpha_j}(R_j - \tilde{R}_j).$$

The inverse can be taken modulo q, since $\alpha_j \in \mathbb{Z}_q^*$ and $c_j \neq \tilde{c}_j$.

The total success probability ε' of the attack performed by $\mathcal{F}$ is

$$\varepsilon' \geq (1-\mu)\frac{\tilde{\varepsilon}}{2} \cdot \frac{1}{2} \cdot \frac{\tilde{\varepsilon}}{4Q_2} \cdot \frac{q-1}{q} \geq (1-\mu)\frac{\tilde{\varepsilon}^2}{16Q_2} \cdot \frac{q-1}{q} \geq$$

$$\geq (1-\mu)\frac{\varepsilon^2}{64Q_2} \cdot \frac{q-1}{q} \geq \frac{\varepsilon^2}{384Q_eQ_2} \cdot \frac{q-1}{q} \geq \frac{\varepsilon^2}{385Q_eQ_2}.$$

We have used the fact that $1 - \mu = 1 - (5/6)^{1/Q_e} \geq 1/6Q_e$ (applying Taylor's series methodology to the function $f(x) = 1 - (1-x)^{1/q_e}$ and then fixing $x = 1/6$). We have also assumed that $q \geq 385$, which happens if the security parameter k is $k \geq 9$. Note that in the case where $Q_e = 0$, the obtained result would be $\varepsilon' \geq \frac{\varepsilon^2}{33Q_2}$.

The total execution time T' of the machine $\mathcal{F}$ consists of running two times the machine $\mathcal{A}$, simulating its environment. We have that $T' \leq 2(T + Q_1 + Q_2 + T_\psi Q_s)$. □

The reduction shown in the proof above is tighter than the reduction in the security theorems of the two previous proposals of distributed ring signature schemes for general access structures [12, 13].

6 Different Types of Keys

The distributed ring signature scheme proposed in Section 4 for ID-based scenarios can be extended to the case where users have different types of keys, of different lengths, etc. This fits in with a more real situation where each user generates his keys in an independent way. We consider three possibilities: RSA keys, Disc-Log keys and ID-based keys. The construction follows some ideas of the works [1, 23].

If a user U_i has RSA keys, then there exist a public key (n_i, e_i) such that user U_i knows the matching public key: the primes p_i and q_i such that $n_i = p_i q_i$, and the value d_i such that $d_i e_i = 1 \bmod \phi(n_i)$. There exists a public hash function $\hat{H}_i : \{0,1\}^* \to \mathbb{Z}_{n_i}^*$.

If a user U_i has a Disc-Log pair of keys, then there exists a pair of prime numbers p_i and q_i, and an element $g_i \in \mathbb{Z}_{p_i}$ such that $q_i | p_i - 1$ and g_i has order q_i in $\mathbb{Z}_{p_i}$.

The secret key of user U_i is a value $x_i \in \mathbb{Z}_{q_i}^*$, whereas the matching public key is $y_i = g_i^{x_i} \bmod p_i$.

Finally, if a user has ID-based keys, this means that there exist an additive group $\mathbb{G}_{1,i}$, generated by some element P_i, and a multiplicative group $\mathbb{G}_{2,i}$, both with the same prime order q_i. There exist a bilinear pairing $e_i : \mathbb{G}_{1,i} \times \mathbb{G}_{1,i} \to \mathbb{G}_{2,i}$ and a public hash function $H_i : \{0,1\}^* \to \mathbb{G}_{1,i} - \{0\}$. User U_i is under the control of a master entity whose secret key is $x_i \in \mathbb{Z}_{q_i}$ and whose public key is $Y_i = x_i P \in \mathbb{G}_{1,i}$. The public key of a user U_i, with identity ID_i is $PK_i = H_i(ID_i)$, whereas his secret key is $SK_i = x_i PK_i$.

Distributed ring signature generation. Assume that a subset of users A want to compute an anonymous signature on behalf of a family $\mathcal{U}$ of possible signing subsets, taken over a set $\mathcal{P} = \{U_1, \ldots, U_n$ of n users.

Let k be twice the length of the largest q_i or n_i, among the n users in $\mathcal{P}$. Let $H : \{0,1\}^* \to \{0,1\}^k$ be a public hash function.

For simplicity, we will assume that both $cl(\mathcal{U})$ and $\Gamma = (cl(\mathcal{U}))^*$ are vector space access structures, and that there exist an integer r and a mapping $\psi : \mathcal{P} \cup \{D\} \to GF\left(2^k\right)^r$ such that $B \in \Gamma \Leftrightarrow \psi(D) \in \langle \{\psi(U_i)\}_{U_i \in B} \rangle$.

Since $\mathcal{U} = (cl(\mathcal{U}))_0$, we have that $A \in (cl(\mathcal{U}))_0$. This means that $\mathcal{P} - A \notin \Gamma$, and is maximal in the sense that $(\mathcal{P} - A) \cup \{U_j\} \in \Gamma$ for any user $U_j \in A$.

The signing users in A execute the following protocol to compute a valid distributed signature on a message $m \in \{0,1\}^*$:

1. They consider a basis of the subspace $\langle \psi(\mathcal{P} - A) \rangle$. This basis corresponds to some subset of users $C \subset \mathcal{P} - A$; that is, vectors in $\psi(C)$ are linearly independent and $\langle \psi(C) \rangle = \langle \psi(\mathcal{P} - A) \rangle$.

2. For every user $U_i \in C$, the signing users choose uniformly at random $c_i \in \{0,1\}^k$, and then proceed as follows:

   (a) If U_i has RSA keys, they choose uniformly at random $s_i \in \mathbb{Z}_{n_i}$; then they compute and broadcast the value $z_i = \hat{H}_i(c_i) + s_i^{e_i} \bmod n_i$.

   (b) If U_i has Disc-Log keys, they choose uniformly at random $s_i \in \mathbb{Z}_{q_i}$; then they compute and broadcast the value $z_i = g_i^{s_i} y_i^{c_i} \bmod p_i$.

   (c) If U_i has ID-based keys, they choose uniformly at random $s_i \in \mathbb{G}_{1,i}$; they compute and broadcast the value $z_i = e_i(s_i, P_i) \cdot e(Y_i, c_i PK_i)$.

3. For users $U_t \in (\mathcal{P} - A) - C$, we have that $\psi(U_t) = \sum_{U_i \in C} \lambda_{it} \psi(U_i)$, for some $\lambda_{it} \in GF(2^k)$, because $\psi(C)$ is a basis of $\langle \psi(\mathcal{P} - A) \rangle$. The signing users consider $c_t = \sum_{U_i \in C} \lambda_{it} c_i$, then they proceed as follows:

   (a) If U_t has RSA keys, they choose uniformly at random $s_t \in \mathbb{Z}_{n_t}$; then they compute and broadcast the value $z_t = \hat{H}_t(c_t) + s_t^{e_t} \bmod n_t$.

   (b) If U_t has Disc-Log keys, they choose uniformly at random $s_t \in \mathbb{Z}_{q_t}$; then they compute and broadcast the value $z_t = g_t^{s_t} y_t^{c_t} \bmod p_t$.

(c) If U_t has ID-based keys, they choose uniformly at random $s_t \in \mathbb{G}_{1,t}$; they compute and broadcast the value $z_t = e_t(s_t, P_t) \cdot e(Y_t, c_t PK_t)$.

4. Each signing user $U_j \in A$ acts as follows:

   (a) If U_j has RSA keys, he chooses uniformly at random $z_j \in \mathbb{Z}_{n_j}$ and computes this value.

   (b) If U_j has Disc-Log keys, he chooses uniformly at random $a_j \in \mathbb{Z}_{q_j}$; then he computes and broadcasts the value $z_j = g_j^{a_j} \bmod p_j$.

   (c) If U_j has ID-based keys, he chooses uniformly at random $T_j \in \mathbb{G}_{1,j}$; he computes and broadcasts the value $z_j = e(T_j, P_j)$.

5. The signing users compute then the value $c = H(\mathcal{U}, m, z_1, \ldots, z_n)$.

6. They choose uniformly at random one of the vectors $\mathbf{v} \in GF\left(2^k\right)^r$ that verifies:

   (i) $\mathbf{v}\psi(D) = c$, and

   (ii) $\mathbf{v}\psi(U_i) = c_i$, for all $U_i \in C$.

   Note that this vector $\mathbf{v}$ exists because $C \notin \Gamma$ and so the vectors $\{\psi(D), \{\psi(U_i)\}_{U_i \in C}\}$ are linearly independent.

7. Every signing user $U_j \in A$ individually computes $c_j = \mathbf{v}\psi(U_j)$; then he proceeds as follows:

   (a) If U_j has RSA keys, he computes and broadcasts the value $s_j = \left(z_j - \hat{H}_j(c_j)\right)^{d_j} \bmod n_j$.

   (b) If U_j has Disc-Log keys, he computes and broadcasts the value $s_j = a_j - c_j x_j \bmod q_j$.

   (c) If U_j has ID-based keys, he computes and broadcasts the value $s_j = T_j - c_j SK_j$.

   Note that the rest of users in A can verify if the broadcast value s_j is consistent with the value z_j broadcast in step 4, by using the public key of user U_j. In this way, they detect dishonest users who try to boycott the process.

8. The resulting signature is $(\mathcal{U}, m, \mathbf{v}, s_1, \ldots, s_n, \psi)$.

Verification of a distributed ring signature. The recipient of the message first computes $c_i = \mathbf{v}\psi(U_i)$, for every user $U_i \in \mathcal{P}$ and then computes the following values:

(a) If U_i has RSA keys, compute $z_i = \hat{H}_i(c_i) + s_i^{e_i} \bmod n_i$.

(b) If U_i has Disc-Log keys, compute $z_i = g_i^{s_i} y_i^{c_i} \bmod p_i$.

(c) If U_i has ID-based keys, compute $z_i = e_i(s_i, P_i) \cdot e(Y_i, c_i PK_i)$.

The signature is valid if $\mathbf{v}\psi(D) = H(\mathcal{U}, m, z_1, \ldots, z_n)$.

The correctness of the scheme is easy to verify. With respect to anonymity and unforgeability, it can be proved using a combination of the techniques that appear in the proof of Theorem 1 and in the security proofs of the papers [1, 23]. To show that the scheme is unforgeable, one proves that if there would exist a successful adversary against it, then one could solve either the RSA problem, or the Discrete Logarithm problem, or the Computational Diffie-Hellman problem.

7 Conclusion

In this work we have dealt with distributed ring signature schemes in identity-based scenarios. Such schemes provide anonymity to a subset of users who want to sign a message on behalf of a family of possible signing subset. In identity-based scenarios, public keys of the users are derived from publicly verifiable data (for example, an e-mail address), and so digital certificates are not necessary to authenticate the validity of public keys.

We have proposed a distributed ring signature scheme which works with general families of possible signing subsets. In the design, we use as a primitive the concept of dual access structures. We have formally proved the unconditional anonymity and the existential unforgeability of our scheme, in the random oracle model, assuming that the Computational Diffie-Hellman problem is intractable. With respect to previous proposals working with general families, the new scheme provides two improvements: the reduction in the proof of unforgeability is tighter, and the length of each signature is linear in the number of involved users.

Although we have analyzed, for clarity, the version for ID-based scenarios with common parameters, the scheme can be extended (using the techniques in [1, 23]) to work in a framework where users have independent keys: either PKI-based or ID-based, and with different public parameters, lengths, etc.

References

[1] M. Abe, M. Ohkubo and K. Suzuki, $1\text{--}\mathrm{out}\text{--}\mathrm{of}\text{--}n$ signatures from a variety of keys. *Proceedings of Asiacrypt'02*, Springer-Verlag, Lecture Notes in Computer Science 2501: 415–432 (2002).

[2] M. Bellare, A. Boldyreva and A. Palacio, An uninstantiable random-oracle-model scheme for a hybrid-encryption problem. *Proceedings of Eurocrypt'04*, Springer-Verlag, Lecture Notes in Computer Science 3027: 171–188 (2004).

[3] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of 1st Conference on Computer and Communications Security*, ACM: 62–73 (1993).

[4] E. Bresson, J. Stern and M. Szydlo, Threshold ring signatures for ad-hoc groups. *Proceedings of Crypt0'02*, Springer-Verlag, Lecture Notes in Computer Science 2442: 465–480 (2002).

[5] E.F. Brickell, Some ideal secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, Vol. 9: 105–113 (1989).

[6] R. Canetti, O. Goldreich and S. Halevi, The random oracle methodology, revisited. *Proceedings of STOC'98*, ACM: 209–218 (1998).

[7] S. Chow, L. Hui and S.M. Yiu, Identity based threshold ring signatures. *Proceedings of ICISC'04*, Springer-Verlag, Lecture Notes in Computer Science (to appear, 2004).

[8] Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup, Annonymous identification in ad hoc groups. *Proceedings of Eurocrypt'04*, Springer-Verlag, Lecture Notes in Computer Science 3027: 609–626 (2004).

[9] A. Fiat and M. Naor, Broadcast encryption. *Proceedings of Crypto'93*, Springer-Verlag, Lecture Notes in Computer Science 773: 480–491 (1993).

[10] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, Robust threshold DSS signatures. *Proceedings of Eurocrypt'96*, Springer-Verlag, Lecture Notes in Computer Science 1070: 354–371 (1996).

[11] J. Herranz and G. Sáez, Forking lemmas for ring signature schemes. *Proceedings of Indocrypt'03*, Springer-Verlag, Lecture Notes in Computer Science 2904: 266–279 (2003).

[12] J. Herranz and G. Sáez, Ring signature schemes for general access structures. *Proceedings of ESAS'04*, Springer-Verlag, Lecture Notes in Computer Science 3313: 54–65 (2005).

[13] J. Herranz and G. Sáez, New ID-based ring signature schemes. *Proceedings of ICICS'04*, Springer-Verlag, Lecture Notes in Computer Science 3269: 27–39 (2004).

[14] W.A. Jackson and K.M. Martin, Geometric secret sharing schemes and their duals. *Designs, Codes and Cryptography*, Vol. 4: 83–95 (1994).

[15] J.B. Nielsen, Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. *Proceedings of Crypto'02*, Springer-Verlag, Lecture Notes in Computer Science 2442: 111–126 (2002).

[16] D. Pointcheval and J. Stern, Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13 (3): 361–396 (2000).

[17] R. Rivest, A. Shamir and Y. Tauman, How to leak a secret. *Proceedings of Asiacrypt'01*, Springer-Verlag, Lecture Notes in Computer Science 2248: 552–565 (2002).

[18] A. Shamir, How to share a secret. *Communications of the ACM*, 22: 612–613 (1979).

[19] A. Shamir, Identity-based cryptosystems and signature schemes. *Proceedings of Crypto'84*, Springer-Verlag, Lecture Notes in Computer Science 196: 47–53 (1984).

[20] V. Shoup, Practical threshold signatures. *Proceedings of Eurocrypt'00*, Springer-Verlag, Lecture Notes in Computer Science 1807: 207–220 (2000).

[21] G.J. Simmons, W.A. Jackson and K.M. Martin, The geometry of secret sharing schemes. *Bulletin of the ICA* Vol. 1: 71–88 (1991).

[22] D.R. Stinson, *Cryptography: Theory and Practice.* CRC Press Inc., Boca Raton (1995).

[23] J.K. Sui Liu, V.K. Wei and D.S. Wong, A separable threshold ring signature scheme. *Proceedings of ICISC'03*, Springer-Verlag, Lecture Notes in Computer Science 2971: 12–26 (2004).

[24] F. Zhang and K. Kim, ID-based blind signature and ring signature from pairings. *Proceedings of Asiacrypt'02*, Springer-Verlag, Lecture Notes in Computer Science 2501: 533–547 (2002).

[25] The Pairing-Based Crypto Lounge, Web page maintained by Paulo Barreto: `http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html`