Forking Lemmas in the Ring Signatures' Scenario

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}@mat.upc.es

Abstract

Pointcheval and Stern introduced in 1996 some forking lemmas useful to prove the security of a family of digital signature schemes. This family includes, for example, Schnorr's scheme and a modification of ElGamal signature scheme.

In this work we generalize these forking lemmas to the ring signatures' scenario. In a ring signature scheme, a signer in a subset (or *ring*) of potential signers produces a signature of a message in such a way that the receiver can verify that the signature comes from a member of the ring, but cannot know which member has actually signed.

We propose a new ring signature scheme, based on Schnorr signature scheme, which provides unconditional anonymity. We use the generalized forking lemmas to prove that this scheme is existentially unforgeable under adaptive chosen-message attacks, in the random oracle model.

1 Introduction

Group-oriented cryptography deals with those situations in which a secret task (signing or decrypting) is performed by a group of entities or on behalf of such a group. Threshold cryptography is an approach to this situation. In a threshold scheme, some participants have shares of the unique secret key of the group. Participation of some determined subset of players is required to perform the corresponding secret task in a correct way.

Two related but different approaches are *ring signatures* and *group signatures*. In a ring signature scheme, an entity signs a message on behalf of a set (or ring) of members that includes himself. The verifier of the signature is convinced that it was produced by some member of the ring, but he does not obtain any information about which member of the ring actually signed. The real signer includes in the signature the identities of the members of the ring that he chooses, depending on his purposes, and probably without their consent.

The idea behind group signature schemes is very similar to that of ring signatures, but with some variations. First of all, there exists a group manager in charge of the join and revocation of the members in the group. Therefore, a user cannot modify the composition of the group. And second, some mechanisms are added in order to allow (only) the group manager to recover the real identity of the signer of a message, for example in the case of a legal dispute. So group signatures are an appropriate tool when members of the group have agreed to cooperate.

Although the formalization and the name of ring signatures schemes have been recently given in [19], first proposals of such schemes can be found in [9, 10, 6]. Furthermore, some efficient proposals of group signature schemes [8, 7, 2] are constructed using as a basis the ring signature scheme that appears in Definition 2 of [6], by adding the necessary elements to achieve revocability of the anonymity on the part of the group manager.

Analogously to individual signature schemes, the highest proposed level of security exigible to ring signature schemes as well as group signature schemes is existential unforgeability under adaptive chosen-message attacks. The recent proposals of ring signature schemes in [19, 5] reach this level of computational security, based on the hardness of the RSA problem, in the random oracle model [3].

In this paper, we extend to the ring signatures' scenario the forking lemmas introduced in [17] to prove the security of the Schnorr signature scheme. We propose a new ring signature scheme based on Schnorr signature scheme [20] which provides unconditional anonymity. We use the extended forking lemmas to prove that this scheme is existentially unforgeable under adaptive chosen-message attacks, in the random oracle model, assuming the hardness of the discrete logarithm problem in subgroups of prime order.

A different but related approach to ring signature schemes for discrete-log settings can be found in [1]. They consider a scenario in which the discrete-log parameters of each participant are different. The resulting scheme is less efficient than ours. However, they also propose a scheme for the particular case where the public parameters of all the participants are equal, which is more efficient than our scheme. The security of this last scheme is not explicitly proved, although the authors asserts that this can be done using reduction techniques similar to those that we use here (i.e. extending to the ring's scenario the techniques appeared in [17, 16] for individual signature schemes).

The paper is organized as follows. In Section 2, we explain the general characteristics of a ring signature scheme, and the security properties that such a scheme must satisfy. In Section 3 we extend to the ring signatures' scenario some techniques introduced by Pointcheval and Stern in [17]. In Section 4 we propose our new ring signature scheme and prove its security (unconditional anonymity and existential unforgeability under adaptive chosen-message attacks). We compare this scheme with the previous proposals of ring signature schemes in Section 5. Finally, we sum up the work in Section 6.

2 Ring Signatures

Following the formalization about ring signatures proposed in [19], we explain in this section the basic definitions and the properties exigible to ring signature schemes, although not rigorously.

Each potential user A_i generates his pair of secret/public keys (sk_i, pk_i) by using a key generation protocol that takes as input a security parameter. The public keys of all the users are certified via a public key infrastructure.

A regular operation of a ring signature scheme consists of the execution of the two following algorithms:

Ring-sign: if a user A_s wants to compute a ring signature on behalf of a ring $A_1, \ldots, A_n$ that contains himself, he executes this probabilistic algorithm with input a message m, the public keys $pk_1, \ldots, pk_n$ of the ring and his secret key sk_s. The output of this algorithm is a ring signature σ for the message m.

Ring-verify: this is a deterministic algorithm that takes as input a message m and a ring signature σ, that includes the public keys of all the members of the corresponding ring, and outputs "True" if the ring signature is valid, or "False" otherwise.

The resulting ring signature scheme must satisfy the following properties:

Correctness: a ring signature generated in a correct way must be accepted by any verifier with overwhelming probability.

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 have greater than $1/(n-1)$.

Unforgeability: among all the proposed definitions of unforgeability (see [14]), we consider the strongest one: any attacker must not have non-negligible probability of success in forging a valid ring signature for some message m on behalf of a ring that does not contain him, even if he knows valid ring signatures for messages, different from m, that he can adaptively choose.

The first proposals of ring signature schemes are previous to the formal definition of this concept. They can be found in [9, 6] 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 [10, 11]).

In [19], 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 [18]. This scheme also uses a symmetric encryption scheme and the notion of combining functions.

Bresson, Stern and Szydlo show in [5] that the scheme in [19] can be modified in such a way that the new scheme can be proved to achieve the same level of security, but under the strictly weaker assumption of the random oracle model. Furthermore, they propose 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.

Finally, in [1], Abe, Ohkubo and Suzuki give general constructions of ring signature schemes for a variety of scenarios, including those where signature schemes are based on one-way functions, and those where signature schemes are of the three-move type (for example, Schnorr's signature scheme).

3 Forking Lemmas for Generic Ring Signatures

In this section we prove some lemmas that we will use later to demonstrate the security of our proposal for a Schnorr ring signature scheme.

In [17], Pointcheval and Stern prove the security of a class of signature schemes, that they call *generic*, which includes Schnorr [20] and a modification of ElGamal [12] schemes. They introduce the *forking lemmas*, which are based on a reduction technique that they call *oracle replay attack*.

Our goal is to extend all these results to the ring signatures' scenario.

3.1 Generic Ring Signature Schemes

We define a class of ring signature schemes that we call also *generic*, and for which the results in this section are valid. Consider a security parameter k, a hash function which outputs k-bit long elements, and a ring $A_1, \ldots, A_n$ of n members. Given the input message m, a generic ring signature scheme produces a tuple $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$, where $R_1, \ldots, R_n$ (randomness) take their values randomly in a large set G in such a way that $R_i \neq R_j$ for all $i \neq j$, h_i is the hash value of (m, R_i), for $1 \leq i \leq n$, and the value σ 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. This condition can be achieved by choosing the set G as large as necessary.

The security proofs of this paper are valid in the *random oracle model* [3], in which a cryptographic hash function is supposed to behave as a random and hidden function that outputs values independently of the input (the only restriction is that equal inputs must produce equal outputs). In the framework that we consider, the outputs of the random oracle will be k-bit long elements.

The basic idea of the forking lemmas in [17] and in the ring forking lemmas that we are going to introduce is the following: assuming that an attacker can forge a generic ring signature, another attacker could obtain, by replaying enough times the first attacker with randomly chosen hash functions (i.e. random oracles), two forged ring signatures of the same message and with the same randomness. Then, these two forged signatures could be used to solve some computational problem which is assumed to be intractable. In this way, the corresponding ring signature scheme is proved to be existentially unforgeable under no-message attacks. Some precautions must be taken in order to achieve unforgeability under chosen-message attacks, which is the standard level of security that a signature scheme can achieve (see [14]).

3.2 No-Message Attacks

The following lemma is a generalization of Lemma 1 in [17] to the ring signatures' scenario.

Lemma 1. *Let Σ_{ring} be a generic ring signature scheme with security parameter k, and let n be the number of members of the corresponding 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 $Q \geq n$. We denote as $V_{Q,n}$ the number of n-permutations of Q elements, that is, $V_{Q,n} = Q(Q-1)\cdot \ldots \cdot (Q-n+1)$. We assume that, within time bound T, $\mathcal{A}$ produces, with probability of success $\varepsilon \geq \frac{7\ V_{Q,n}}{2^k}$, a valid ring signature $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$. Then, within time $T' \leq \frac{16 V_{Q,n} T}{\varepsilon}$, and with probability $\varepsilon' \geq \frac{1}{9}$, a replay of this machine outputs two valid ring signatures $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(m, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ 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. The Turing machine $\mathcal{A}$, with random tape ω, can ask Q queries to the random oracle f. We denote by $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$ the Q distinct questions and by $\rho = (\rho_1, \ldots, \rho_Q)$ the list of the Q answers of the random oracle f. So we can see a random choice of the random oracle f as a random choice of such a vector ρ.

Now, for a random choice of (ω, f) and with probability ε, the machine $\mathcal{A}$ outputs a valid ring signature $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$. Since f is a random oracle and its outputs are k-bit long elements, the probability that there exists some index i such that $\mathcal{A}$ has not asked the query (m, R_i) to the random oracle, is less than $n/2^k$, and so negligible. We can assume therefore that $\mathcal{A}$ has asked all the queries (m, R_i) to the oracle, for $1 \leq i \leq n$, and so we have that $Q \geq n$ is necessary.

With probability at least $\varepsilon - \frac{n}{2^k}$, the machine $\mathcal{A}$ is successful in forging a ring signature $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and besides it has asked to the random oracle all the queries (m, R_i), for $i = 1, \ldots, n$. In this case, for all index i there exists an integer $\ell_i \in \{1, 2, \ldots, Q\}$ such that the query $\mathcal{Q}_{\ell_i}$ is precisely (m, R_i). Then, we define $L(\omega, f) = (\ell_1, \ell_2, \ldots, \ell_n)$ and $\beta(\omega, f) = \max\{\ell_i \mid (\ell_1, \ell_2, \ldots, \ell_n) = L(\omega, f)\}$. Note that, since the forged ring signature is a valid generic one, we have that all the R_i's are different, and so the integers ℓ_i are also all different.

In the unlikely case where $\mathcal{A}$ has not asked some of the pairs (m, R_i) to the random oracle, then we say that $\ell_i = \infty$, and so $\beta(\omega, f) = \infty$. Now we define the sets

$$\mathcal{S} = \{(\omega, f) \mid \mathcal{A}(\omega, f) \text{ succeeds and } \beta(\omega, f) \neq \infty\}\ ,$$

$$\mathcal{S}_{\vec{\ell}} = \{(\omega, f) \mid \mathcal{A}(\omega, f) \text{ succeeds and } L(\omega, f) = \vec{\ell}\}\ ,$$

for all the vectors $\vec{\ell} \in L_n = \{(\ell_1, \ell_2, \ldots, \ell_n) \mid 1 \leq \ell_i \leq Q \text{ and } \ell_i \neq \ell_j\ \forall i \neq j\}$. Note that the cardinality of this set L_n is the number of n-permutations of Q elements, that is, $V_{Q,n} = Q(Q-1)\cdot \ldots \cdot (Q-n+1)$. Furthermore, the set $\{\mathcal{S}_{\vec{\ell}} \mid \vec{\ell} \in L_n\}$ is a partition of $\mathcal{S}$. The pairs (ω, f) in the set $\mathcal{S}$ are called the successful pairs. We can find the lower bound $\nu = \Pr[\mathcal{S}] \geq \varepsilon - \frac{n}{2^k} \geq \frac{6\varepsilon}{7}$.

Let I be the set formed by the most likely vectors, $I = \{\vec{\ell} \in L_n \mid \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}] \geq \frac{1}{2V_{Q,n}}\}$. Probabilities are taken over the random choice of (ω, f). The following lemma asserts that, in case of success, the corresponding vector of indexes lies in I with probability at least $\frac{1}{2}$.

Lemma 2. $\Pr[L(\omega, f) \in I \mid \mathcal{S}] \geq \frac{1}{2}$.

Proof. Since the sets $\mathcal{S}_{\vec{\ell}}$ are disjoint, we have that $\Pr[L(\omega, f) \in I \mid \mathcal{S}] = \sum_{\vec{\ell} \in I} \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}]$. This probability is equal to $1 - \sum_{\vec{\ell} \notin I} \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}]$. Since the complement of I contains fewer than $V_{Q,n}$ vectors, this probability is at least $1 - V_{Q,n} \cdot \frac{1}{2V_{Q,n}} = \frac{1}{2}$. □

The following lemma will be used in the same way as Pointcheval and Stern do (see [17] for the proof):

Lemma 3. *(The Splitting Lemma) Let* $A \subset X \times Y$ *such that* $\Pr[(x, y) \in A] \geq \epsilon$. *For any* $\alpha < \epsilon$, *define*

$$B = \{(x, y) \in X \times Y | \Pr_{y' \in Y}\left[(x, y') \in A\right] \geq \epsilon - \alpha\}$$

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. $\Pr[B | A] \geq \alpha / \epsilon$.

Now we run $2/\varepsilon$ times the attacker $\mathcal{A}$ with random ω and random f. Since $\nu = \Pr[\mathcal{S}] \geq \frac{6\varepsilon}{7}$, with probability greater than $1 - (1 - 6\varepsilon/7)^{2/\varepsilon} \geq 1 - e^{-12/7} \geq \frac{4}{5}$, we get at least one pair (ω, f) in $\mathcal{S}$.

For each vector $\vec{\ell} \in I$, if we denote by $\beta_{\vec{\ell}}$ the maximum of the coordinates of $\vec{\ell}$, we can apply the Splitting Lemma (Lemma 3). Following the notation of this lemma, and if we see the oracle f as a random vector $(\rho_1, \ldots, \rho_Q)$, then $A = \mathcal{S}_{\vec{\ell}}$, $X = \{(\omega, \rho_1, \ldots, \rho_{(\beta_{\vec{\ell}} - 1)})\}_{\omega, f}$ and $Y = \{(\rho_{\beta_{\vec{\ell}}}, \ldots, \rho_Q)\}_f$. We also refer to $(\rho_1, \ldots, \rho_{(\beta - 1)})$ as f_β, the restriction of f to queries of index strictly less than β. Since $\Pr[\mathcal{S}_{\vec{\ell}}] = \Pr[\mathcal{S}] \cdot \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}] \geq \frac{\nu}{2V_{Q,n}}$, we take $\epsilon = \frac{\nu}{2V_{Q,n}}$ and $\alpha = \frac{\nu}{4V_{Q,n}}$, and the Splitting Lemma proves that there exists a subset $\Omega_{\vec{\ell}}$ of executions (ω, f) such that, for any $(\omega, f) \in \Omega_{\vec{\ell}}$,

$$\Pr_{f'}[(\omega, f') \in \mathcal{S}_{\vec{\ell}} \mid f_{\beta_{\vec{\ell}}} = f'_{\beta_{\vec{\ell}}}] \geq \frac{\nu}{4V_{Q,n}} \tag{1}$$

$$\text{and} \quad \Pr[\Omega_{\vec{\ell}} \mid \mathcal{S}_{\vec{\ell}}] \geq \frac{1}{2}$$

Using again that the sets $\mathcal{S}_{\vec{\ell}}$ are all disjoint, we have that

$$\Pr_{\omega, f}[\exists \vec{\ell} \in I \text{ s.t. } (\omega, f) \in \Omega_{\vec{\ell}} \cap \mathcal{S}_{\vec{\ell}} \mid \mathcal{S}] = \Pr\left[\bigcup_{\vec{\ell} \in I} (\Omega_{\vec{\ell}} \cap \mathcal{S}_{\vec{\ell}}) \mid \mathcal{S}\right] = \sum_{\vec{\ell} \in I} \Pr[\Omega_{\vec{\ell}} \cap \mathcal{S}_{\vec{\ell}} \mid \mathcal{S}] =$$

$$= \sum_{\vec{\ell}\in I} \Pr[\Omega_{\vec{\ell}} \mid \mathcal{S}_{\vec{\ell}}] \cdot \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}] \geq \Big(\sum_{\vec{\ell}\in I} \Pr[\mathcal{S}_{\vec{\ell}} \mid \mathcal{S}]\Big)/2 \geq \frac{1}{4} \ .$$

For simplicity, we denote by $\vec{\ell}$ the vector $L(\omega, f)$ corresponding to the successful pair (ω, f) obtained in the first $2/\varepsilon$ repetitions of the attack $\mathcal{A}$ with probability at least $4/5$, and by β the corresponding index $\beta(\omega, f)$. As we have seen, with probability at least $1/4$, we have that $\vec{\ell} \in I$ and $(\omega, f) \in \Omega_{\vec{\ell}} \cap \mathcal{S}_{\vec{\ell}}$. Therefore, with probability greater than $1/5$, the $2/\varepsilon$ repetitions of the attack have provided a successful pair (ω, f), with $\vec{\ell} = L(\omega, f) \in I$ and $(\omega, f) \in \Omega_{\vec{\ell}} \cap \mathcal{S}_{\vec{\ell}}$.

If now we replay the attack, with fixed random tape ω but randomly chosen oracle f' such that $f'_\beta = f_\beta$, we can use inequality (1) and thus we obtain that

$$\Pr_{f'}[(\omega, f') \in \mathcal{S}_{\vec{\ell}} \text{ and } \rho_\beta \neq \rho'_\beta \mid f'_\beta = f_\beta] \geq \Pr_{f'}[(\omega, f') \in \mathcal{S}_{\vec{\ell}} \mid f'_\beta = f_\beta] - \Pr_{f'}[\rho_\beta = \rho'_\beta] \geq$$

$$\geq \frac{\nu}{4V_{Q,n}} - \frac{1}{2^k} \geq \frac{\varepsilon}{14V_{Q,n}} \ ,$$

where $\rho_\beta = f(\mathcal{Q}_\beta)$ and $\rho'_\beta = f'(\mathcal{Q}_\beta)$. If we now replay the attack $14V_{Q,n}/\varepsilon$ times with fixed ω and random oracle f' such that $f'_\beta = f_\beta$, we will get another success (or forking) $(\omega, f') \in \mathcal{S}_{\vec{\ell}}$ with probability greater than $3/5$.

Summing up, after less than $\frac{2}{\varepsilon} + \frac{14V_{Q,n}}{\varepsilon} \leq \frac{16V_{Q,n}}{\varepsilon}$ executions of the machine $\mathcal{A}$, and with probability greater than $\frac{1}{5} \cdot \frac{3}{5} \geq \frac{1}{9}$, we obtain two valid ring signatures $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(m, R'_1, \ldots, R'_n, h'_1, \ldots, h'_n, \sigma')$ from two executions of $\mathcal{A}$ with the same random tape ω (that is, with the same randomness $R_1 = R'_1, \ldots, R_n = R'_n$), but with two different random oracles f and f', that we can see as two different vectors $\rho = (\rho_1, \ldots, \rho_Q) = (f(\mathcal{Q}_1), \ldots, f(\mathcal{Q}_Q))$ and $\rho' = (\rho'_1, \ldots, \rho'_Q) = (f'(\mathcal{Q}_1), \ldots, f'(\mathcal{Q}_Q))$. These two oracles verify, furthermore, that $\rho_t = \rho'_t$, for all $t = 1, \ldots, \beta - 1$, and $\rho_\beta \neq \rho'_\beta$, where β is the index $\beta(\omega, f)$ corresponding to the first successful forgery performed by $\mathcal{A}$.

Therefore, if we denote as j the index such that (m, R_j) was the query $\mathcal{Q}_\beta$, then we have that $h_j = f(\mathcal{Q}_\beta) \neq f'(\mathcal{Q}_\beta) = h'_j$. However, the rest of pairs (m, R_i), for $i = 1, \ldots, n$, $i \neq j$, have been asked before the query $\mathcal{Q}_\beta$, and so the values obtained from the oracles f and f' have been the same. That is, $h_i = h'_i$, for all $i = 1, \ldots, n$ with $i \neq j$.

□

Theorem 1. *(The No-Message Ring Forking Lemma). Let Σ_{ring} be a generic ring signature scheme with security parameter k, and let n be the number of members of the corresponding ring. Let the forger $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by Q the number of queries that $\mathcal{A}$ can ask to the random oracle. Assume that, within time bound T, $\mathcal{A}$ produces, with probability of success $\varepsilon \geq \frac{7\ V_{Q,n}}{2^k}$, a valid ring signature $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$. Then there is another probabilistic polynomial time Turing machine which uses $\mathcal{A}$ and produces two valid ring signatures $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(m, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$*

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 $i \neq j$, in expected time $T' \leq \frac{84480 T V_{Q,n}}{\varepsilon}$.

Proof. The idea is to construct a specific expected polynomial time Turing machine $\mathcal{B}$ that uses the Turing machine $\mathcal{A}$ as a sub-routine in order to obtain two valid ring signatures of the same message and with the desired properties (same randomness and all the h_i's but one equal). And then we must calculate the expectation of the random variable that counts the number of times that the machine $\mathcal{A}$ is invoked by $\mathcal{B}$. The design of $\mathcal{B}$ and the computation of this expectation can be performed exactly in the same way as in [17], changing their value Q by our value $V_{Q,n}$. The resulting expectation is less than $84480 V_{Q,n}/\varepsilon$. If T is a time bound for the machine $\mathcal{A}$, then a time bound for the machine $\mathcal{B}$ will be $\frac{84480 T V_{Q,n}}{\varepsilon}$. □

3.3 Chosen-Message Attacks

Now we consider another kind of attacks against a ring signature scheme, the chosen-message attacks. They are the strongest ones usually considered, so if a ring signature scheme is proved to be unforgeable against them, then we can say that the scheme achieves the standard level of security.

In a chosen-message attack, an adversary is given the public data of the scheme (including the members of the ring and their public keys). Then he can ask to some real signers of the ring for valid ring signatures of a polynomial number of messages of his choice. A same message can be asked more than once, and the choice of the messages is adaptive, in the sense that the attacker can adapt his queries according to previous message-signature pairs.

If the attacker obtains, after this interaction with the signers, with non-negligible probability and in polynomial time a valid ring signature on a message that has not been previously signed by the real signers, then we say that the attack is successful, or that the ring signature scheme is *existentially forgeable under chosen-message attacks.*

If we want to prove the security of a ring signature scheme in the random oracle model, then the considered attacker will be able to ask a polynomial number of queries to the random oracle model, too.

The following theorem is a variation of Theorem 1 considering chosen-message attacks. It can also be seen as an adaptation of Theorem 3 in [17] to the ring signatures' scenario.

Theorem 2. *(The Chosen-Message Ring Forking Lemma). Let Σ_{ring} be a generic ring signature scheme with security parameter k, and let n be the number of members of the corresponding ring. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by Q and W the number of queries that $\mathcal{A}$ can ask to the random oracle and to some real signers of the ring, respectively. Assume that, within time bound T, $\mathcal{A}$ produces, with probability of success $\varepsilon \geq \frac{12\ V_{Q,n} + 6(Q+Wn)^2}{2^k}$, a valid ring signature $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$. Suppose that valid ring signatures can be simulated with an indistinguishable distribution of probability, with time bound T_s*

and without knowing any of the secret keys of the ring. Then there is another probabilistic polynomial time Turing machine which has control over the machine obtained from $\mathcal{A}$ by replacing interactions with the real signers by simulation, and which produces two valid ring signatures $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(m, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ 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 $i \neq j$, in expected time $T' \leq \frac{144823 V_{Q,n}(T+WT_s)}{\varepsilon}$.

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 oracle and with real signers. Then we could see $\mathcal{B}$ as a machine performing a no-message attack against the ring signature scheme.

We denote by $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$ the Q distinct queries of $\mathcal{A}$ to the random oracle, and by $m^{(1)}, \ldots, m^{(W)}$ the W queries (possibly repeated) to the real signers. Using the simulator, $\mathcal{B}$ can simulate the answers of the real ring of signers without knowing any of the secret keys of the ring. For a message $m^{(j)}$, the simulator answers a tuple $(m^{(j)}, R_1^{(j)}, \ldots, R_n^{(j)}, h_1^{(j)}, \ldots, h_n^{(j)}, \sigma^{(j)})$. Then $\mathcal{B}$ constructs a random oracle f by storing in a "random oracle list" the relations $f(m^{(j)}, R_i^{(j)}) = h_i^{(j)}$. The attacker $\mathcal{A}$ receives this signature, assumes that $f(m^{(j)}, R_i^{(j)}) = h_i^{(j)}$, where f is the random oracle, for all $1 \leq i \leq n$ and $1 \leq j \leq W$, and stores all these relations. When $\mathcal{A}$ makes a query $\mathcal{Q}_t = (m, R)$ to the random oracle, $\mathcal{B}$ looks for the value (m, R) in the random oracle list. If the value is already in the list, then $\mathcal{B}$ returns to $\mathcal{A}$ the corresponding $f(m, R)$. Otherwise, $\mathcal{B}$ chooses a random value h, sends it to $\mathcal{A}$ and stores the relation $f(m, R) = h$ in the list.

There is some risk of "collisions" of queries to the random oracle. In the definition of generic ring signature schemes, we made the assumption that no R_i can appear with probability greater than $2/2^k$ in a 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 $1/2^k$.

Then, three kinds of collisions can occur:

- A pair $(m^{(j)}, R_i^{(j)})$ that the simulator outputs, as part of a simulated ring signature, has been asked before to the random oracle by the attacker. In this case, it is quite unlikely that the relation $f(m^{(j)}, R_i^{(j)}) = h_i^{(j)}$ corresponding to the values output by the simulator coincides with the relation previously stored in the random oracle list. The probability of such a collision is, however, less than $Q \cdot nW \cdot \frac{2}{2^k} \leq \frac{\varepsilon}{6}$.

- A pair $(m^{(j_1)}, R_{i_1}^{(j_1)})$ that the simulator outputs, as part of a simulated ring signature, is exactly equal to another pair $(m^{(j_2)}, R_{i_2}^{(j_2)})$ also output by the simulator. The probability of this collision is less than $\frac{(nW)^2}{2} \cdot \frac{2}{2^k} \leq \frac{\varepsilon}{6}$.

- Two answers h_1 and h_2 of the random oracle chosen at random by $\mathcal{B}$ are exactly equal, while the two corresponding inputs $(m^{(1)}, R_1)$ and $(m^{(2)}, R_2)$ are different. The probability of such an event is less than $\frac{(Q+nW)^2}{2} \cdot \frac{1}{2^k} \leq \frac{\varepsilon}{12}$.

Altogether, the probability of collisions is less than $5\varepsilon/12$. Now we can compute:

$$\Pr_{(\omega,f)}[\mathcal{B} \text{ succeeds}] = \Pr_{(\omega,f)}[\text{no-collisions in the simulations and } \mathcal{A} \text{ succeeds}] \geq$$

$$\geq \Pr_{(\omega,f)}[\mathcal{A} \text{ succeeds} \mid \text{no-collisions in the simulations }] - \Pr_{(\omega,f)}[\text{collisions in the simulations}]$$

$$\geq \varepsilon - \frac{5\varepsilon}{12} = \frac{7\varepsilon}{12} \ .$$

Summing up, we have a machine $\mathcal{B}$ that performs a no-message attack against the ring signature scheme with time bound $T + WT_s$ and with probability of success greater than $\frac{7\varepsilon}{12} \geq \frac{7V_{Q,n}}{2^k}$. So we can use Theorem 1 applied to the machine $\mathcal{B}$, and we will obtain two valid ring signatures in expected time bounded by $\frac{84480(T+WT_s)V_{Q,n}}{7\varepsilon/12} \leq \frac{144823V_{Q,n}(T+WT_s)}{\varepsilon}$.

□

In the next section we show an example of how these extended forking lemmas can be applied to prove the security of a specific ring signature scheme. In fact, we think that the these Ring Forking Lemmas are of independent interest, because they could be useful to prove the security of future ring signature schemes.

4 Schnorr Ring Signature Scheme

In this section we present a ring signature scheme based on Schnorr signature scheme [20]. The proposed scheme is unconditionally anonymous, and it achieves the highest level of computational security, in the random oracle model. Namely, we prove that an existential forgery of our scheme under an adaptive chosen-message attack is equivalent to solving the discrete logarithm problem in subgroups of prime order.

4.1 Schnorr Signature Scheme

Let p and q be large primes such that $q|p-1$ and $q \geq 2^k$, where k is the security parameter of the scheme. Let g be an element of $\mathbb{Z}_p^*$ with order q, and let $H()$ be a collision resistant hash function which outputs elements in $\mathbb{Z}_q$.

A signer has a private key $x \in \mathbb{Z}_q^*$ and the corresponding public key $y = g^x \mod p$. To sign a message m, the signer:

1. Chooses a random $a \in \mathbb{Z}_q^*$.
2. Computes $R = g^a \bmod p$ and $\sigma = a + xH(m, R) \mod q$.
3. Defines the signature on m to be the pair (R, σ).

The validity of the signature is verified by the recipient by checking that $g^\sigma = Ry^{H(m,R)} \bmod p$.

In [17], Pointcheval and Stern proved that, in the random oracle model, an existential forgery under an adaptive chosen-message attack of Schnorr's scheme is equivalent to solving the discrete logarithm problem in the subgroup $< g >$ generated by the element g. The input of this problem is a tuple (p, q, g, y) such that $y \in < g >$, where g is an element of order q in $\mathbb{Z}_p$, and q is a prime that divides $p-1$ (where p is also a prime number). The solution of the problem is the only element $x \in \mathbb{Z}_q$ such that $y = g^x \bmod p$. The discrete logarithm problem in subgroups of prime order is supposed to be computationally intractable.

4.2 The Proposed Scheme

As in the Schnorr signature scheme, let p and q be large primes with $q|p-1$ and such that $q \geq 2^k$, where k is the security parameter of the scheme. Let g be a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q and $H()$ a collision resistant hash function that outputs elements in $\mathbb{Z}_q$.

Consider a set, or ring, of potential signers $A_1, \ldots, A_n$. Every potential signer A_i has a private key $x_i \in \mathbb{Z}_q^*$ and the corresponding public key $y_i = g^{x_i} \bmod p$.

Ring-sign: to sign a message m on behalf of the ring $A_1, \ldots, A_n$, a signer A_s, where $s \in \{1, \ldots, n\}$, acts as follows:

1. For all $i \in \{1, \ldots, n\}$, $i \neq s$, choose a_i at random in $\mathbb{Z}_q^*$, pairwise different. Compute $R_i = g^{a_i} \bmod p$, for all $i \neq s$.
2. Choose a random $a \in \mathbb{Z}_q$.
3. Compute $R_s = g^a \prod_{i \neq s} y_i^{-H(m,R_i)} \bmod p$. If $R_s = 1$ or $R_s = R_i$ for some $i \neq s$, then go to step 2.
4. Compute $\sigma = a + \sum_{i \neq s} a_i + x_s H(m, R_s) \bmod q$.
5. Define the signature of the message m made by the ring $A_1, \ldots, A_n$ to be $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$, where $h_i = H(m, R_i)$, for all $1 \leq i \leq n$.

Ring-verify: the validity of the signature is verified by the recipient of the message by checking that $h_i = H(m, R_i)$ and that

$$g^\sigma = R_1 \cdot \ldots \cdot R_n \cdot y_1^{h_1} \cdot \ldots \cdot y_n^{h_n} \bmod p.$$

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

$$R_1 \cdot \ldots \cdot R_n \cdot y_1^{h_1} \cdot \ldots \cdot y_n^{h_n} = g^a y_s^{h_s} \prod_{i \neq s} R_i = g^{a + x_s h_s + \sum_{i \neq s} a_i} = g^\sigma \bmod p.$$

4.3 Anonymity of the Scheme

In order to prove that our ring signature scheme is unconditionally anonymous, it is enough to prove that any ring signature produced with the method described in Section 4.2 could have been computed by any of the n members of the ring with the same probability.

Let $Sig = (m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ a valid ring signature of a message m. That is, $h_i = H(m, R_i)$ and $g^\sigma = R_1 \cdot \ldots \cdot R_n \cdot y_1^{h_1} \cdot \ldots \cdot y_n^{h_n}$. Let A_s be a member of the ring. We now find the probability that A_s computes exactly the ring signature Sig, when he produces a ring signature of message m by following the method explained in Section 4.2.

The probability that A_s computes the correct $R_i \neq 1$ of Sig, pairwise different for $1 \leq i \leq n$, $i \neq s$, is $\frac{1}{q-1} \cdot \frac{1}{q-2} \cdot \ldots \cdot \frac{1}{q-n+1}$. Then, the probability that A_s chooses exactly the only value $a \in \mathbb{Z}_q$ that leads to the value R_s of Sig, among all possible values for R_s different to 1 and different to all R_i with $i \neq s$, is $\frac{1}{q-n}$.

Summing up, the probability that A_s generates exactly the ring signature Sig is

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

and this probability does not depend on A_s, so it is the same for all the members of the ring. This fact proves the unconditional anonymity of the scheme.

4.4 Unforgeability of the Scheme

Proposition 1. *The ring signatures produced by the scheme proposed in Section 4.2 can be simulated in polynomial time, without knowing any of the secret keys of the ring, and with distribution of probability indistinguishable of ring signatures produced by a legitimate signer, in the random oracle model.*

Proof. The simulation of a Schnorr ring signature for a message m 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 $\mathbb{Z}_q^*$, pairwise different. Compute $R_i = g^{a_i} \mod p$, for all $i \neq s$.
3. Choose at random $h_1, h_2, \ldots, h_n$, pairwise different in $\mathbb{Z}_q$.
4. Choose at random $\sigma \in \mathbb{Z}_q$.
5. Compute $R_s = g^{\sigma - \sum_{i \neq s} a_i} y_1^{-h_1} \ldots y_n^{-h_n}$. If $R_s = 1$ or $R_s = R_i$ for some $i \neq s$, then go to step 4.
6. Return the tuple $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$.

It is easy to see that this simulation runs in polynomial time. We denote by T_s the time bound for an execution of each simulation. Note that, if we assume $H(m, R_i) = h_i$ (we are in the random oracle model), for all $i \in \{1, \ldots, n\}$, then the returned tuple is a valid Schnorr ring signature of the message m.

The distribution corresponding to ring signatures generated by using the protocol explained in Section 4.2, and the distribution corresponding to ring signatures simulated with the method explained in this section can be proved to be statistically indistinguishable, and so polynomially indistinguishable, as desired (the complete proof will appear in the extended version of this paper).

□

Now we prove the existential unforgeability of our proposed ring signature scheme under adaptive chosen-message attacks. The proof is valid in the random oracle model, and the security of the scheme is reduced to the intractability of the discrete logarithm problem in subgroups of prime order.

Theorem 3. *Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine which obtains an existential forgery of the Schnorr ring signature scheme presented in Section 4.2, within a time bound T and under an adaptive chosen-message attack, with probability of success ε. Let n be the number of members of the ring. We denote respectively by Q and W the number of queries that $\mathcal{A}$ can ask to the random oracle and to the signing oracle. Assuming that $\varepsilon \geq \frac{12\ V_{Q,n} + 6(Q+Wn)^2}{2^k}$ (otherwise, ε would be negligible), then the discrete logarithm problem in subgroups of prime order can be solved within expected time less than $\frac{144823 V_{Q,n}(T+WT_s)}{\varepsilon}$.*

Proof. Let (p, q, g, y) the input of an instance of the discrete logarithm problem in the subgroup $< g >$ of $\mathbb{Z}_p$ of order q, where q is a prime that divides $p-1$.

We choose at random $\alpha_i \in \mathbb{Z}_q^*$ pairwise different, for $1 \leq i \leq n$, and define $y_i = y^{\alpha_i} \bmod p$. Then we initialize the attacker $\mathcal{A}$ with a ring of members $A_1, \ldots, A_n$ and corresponding public keys $y_1, \ldots, y_n$. Since our Schnorr ring signature scheme can be simulated, we can apply Theorem 2, and so we can obtain from a replay of attacker $\mathcal{A}$ two valid ring signatures $(m, R_1, \ldots, R_n, h_1, \ldots, h_n, \sigma)$ and $(m, R_1, \ldots, R_n, h'_1, \ldots, h'_n, \sigma')$ 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 $i \neq j$. Then we have that

$$g^{\sigma} = R_1 \cdot \ldots \cdot R_n \cdot y_1^{h_1} \cdot \ldots \cdot y_j^{h_j} \cdot \ldots \cdot y_n^{h_n}$$

$$g^{\sigma'} = R_1 \cdot \ldots \cdot R_n \cdot y_1^{h'_1} \cdot \ldots \cdot y_j^{h'_j} \cdot \ldots \cdot y_n^{h'_n}$$

Dividing these two equations, we obtain that $g^{\sigma - \sigma'} = y_j^{h_j - h'_j} = y^{\alpha_j (h_j - h'_j)}$, and so we have that

$$y = g^{\frac{\sigma - \sigma'}{\alpha_j (h_j - h'_j)}} \bmod p.$$

Therefore, we have found the discrete logarithm of y in base g, which is $\log_g y = (\sigma - \sigma')\alpha_j^{-1}(h_j - h'_j)^{-1} \bmod q$.

□

5 Comparing Ring Signature Schemes

The ring signature scheme proposed in Section 4 runs in a discrete-log scenario in which there are some public parameters (the primes p and q, and the element g) common to all the entities involved in the scheme. This fact also happens in some previous proposals of ring and group signature schemes [6, 8, 7, 2] defined in a discrete-log setting. In [1], the authors propose ring signature schemes (for different scenarios) that avoid this problem, allowing each entity of the scheme to have different public parameters. The proposals in [19, 5] for an RSA scenario also enjoy this property.

All the ring signature schemes in the discrete-log scenario require in the generation of a ring signature a linear number of exponentiations with respect to the size of the ring of possible signers. The general proposal in [1] produces ring signatures shorter than the ones produced by our scheme. However, the $O(n)$ exponentiations (in the signature generation as well as in the verification) of their scheme depend on the previous ones, and so cannot be computed in a parallel way, as it happens in our scheme.

The authors of [1] also propose a more efficient scheme for the particular case in which the discrete-log parameters are common to all the entities. This scheme is more efficient than ours, in terms of the length of the signatures and the number of exponentiations, which can be parallelized in their scheme, too. No explicit proof of the security of this particular scheme is given in [1], although the authors remark that this security could be proved extending to the ring signatures' scenario (supposedly in a similar way as we have done in this work) the techniques introduced in [17, 16] in the context of individual signatures.

The signature scheme in Definition 2 of [6] can be seen as a Schnorr ring signature scheme, too. It derives from the witness indistinguishable proofs of knowledge introduced in [10, 11]. But this fact does not ensure, to the best of our knowledge, that the resulting ring scheme would be existentially unforgeable under chosen-message attacks (similarly to what happens in the case of individual Schnorr's scheme). Therefore, an explicit and rigorous proof of security for this ring signature scheme would be desirable. We suppose that the idea is to extend to the ring's scenario the results of [16], as we have done in this work with the results of [17]. Anyway, this ring signature scheme is less efficient than our scheme, because the resulting signatures are longer, and twice the amount of exponentiations is needed in the verification of a ring signature.

The proposals for an RSA scenario [19, 5, 1] are in some way more efficient because the ring signature generation protocol requires only one modular exponentiation, and the verification protocol requires no modular exponentiations. However, this is true only if the public RSA keys e_i of all the entities A_i are very small numbers (for example, $e_i = 3$ for all A_i).

Finally, in [5] the authors propose a threshold ring signature scheme. In such a scheme, t entities take part in the generation of the ring signature for a certain ring containing these t entities. The verifier of the signature must be convinced that t different entities of the ring took part in the generation of the signature, but must have no information about which t users actually signed the message.

As it has been pointed out in [4], the ring signature scheme in Definition 2 of [6], as well as all the group signature schemes [6, 8, 7, 2], which are based on it, can be extended to enable generation of threshold ring/group signatures. How to do this extension in the case of our scheme or the schemes proposed in [1] remains as an open problem.

6 Conclusions

We have proposed a new ring signature scheme for the discrete-log setting, which we have proved to be unconditionally anonymous and existentially unforgeable, in the random oracle model, under adaptive chosen-message attacks, assuming the hardness of the discrete logarithm problem in subgroups of prime order. For proving these results, we have extended to the ring signatures' scenario some security lemmas introduced by Pointcheval and Stern in [17] to prove the security of some generic signature schemes.

The forking lemmas in [17] can be applied in any signature scheme obtained from a honest-verifier zero-knowledge identification protocol (also known as three-move signature schemes), for example the ones by Schnorr [20], Fiat-Shamir [13], or Guillou-Quisquater [15]. Analogously, our extension of the forking lemmas to the ring signatures' scenario, that we have applied to a particular Schnorr ring signature scheme, could be used to prove the security of future ring signature schemes constructed from these three-move signature schemes.

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] G. Ateniese, J. Camenisch, M. Joye and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. *Advances in Cryptology-Crypto'00*, LNCS **1880**, Springer-Verlag, pp. 255–270 (2000).

[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 and J. Stern. Proofs of knowledge for non-monotone discrete-log formulae and applications. *Proceedings of ISC'02*, LNCS **2433**, Springer-Verlag, pp. 272–288 (2002).

[5] 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).

[6] J. Camenisch. Efficient and generalized group signatures. *Advances in Cryptology-Eurocrypt'97*, LNCS **1233**, Springer-Verlag, pp. 465–479 (1997).

[7] J. Camenisch and M. Michels. A group signature scheme with improved efficiency. *Advances in Cryptology-Asiacrypt'98*, LNCS **1514**, Springer-Verlag, pp. 160–174 (1998).

[8] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. *Advances in Cryptology-Crypto'97*, LNCS **1294**, Springer-Verlag, pp. 410–424 (1997).

[9] D. Chaum and E. van Heyst. Group signatures. *Advances in Cryptology-Eurocrypt'91*, LNCS **547**, Springer-Verlag, pp. 257–265 (1991).

[10] 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).

[11] 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).

[12] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory* **31**, pp. 469–472 (1985).

[13] A. Fiat and A. Shamir. How to prove yourself: practical solutions of identification and signature problems. *Advances in Cryptology-Crypto'86*, LNCS **263**, Springer-Verlag, pp. 186–194 (1986).

[14] 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).

[15] L.C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. *Advances in Cryptology-Eurocrypt'88*, LNCS **330**, Springer-Verlag, pp. 123–128 (1988).

[16] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. *Advances in Cryptology-Crypto'98*, LNCS **1462**, Springer-Verlag, pp. 354–369 (1998).

[17] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, Vol. **13** (3), Springer-Verlag, pp. 361–396 (2000).

[18] R.L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, **21**, pp. 120–126 (1978).

[19] 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).

[20] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, Vol. **4**, Springer-Verlag, pp. 161–174 (1991).