Blind Ring Signatures Secure under the Chosen-Target-CDH Assumption

Javier Herranz[1⋆] and Fabien Laguillaumie[2]

[1] Centrum voor Wiskunde en Informatica (CWI)
Kruislaan 413, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands
Javier.Herranz@cwi.nl
[2] Projet TANC - INRIA Futurs
Laboratoire d'informatique (LIX), École polytechnique
91128 Palaiseau cedex - France
laguillaumie@lix.polytechnique.fr

Abstract. Blind signatures are a useful ingredient to design secure sophisticated systems like electronic voting or sensitive applications like e-cash. Multi-users signature schemes, like ring or group signatures, are also a useful tool to provide to such systems some properties like scalability, anonymity, (dynamic) group structure, revocation facilities... We propose in this article a simple blind ring signature scheme based on pairings on algebraic curves. We formally prove the security (anonymity, blindness and unforgeability) of our scheme in the random oracle model, under quite standard assumptions.

Keywords: blind ring signatures, e-cash systems, provable security.

1 Introduction

Blind signatures were introduced by Chaum [13]. They allow a person to get a message signed by another party without revealing any information about the message to this other party. Blind signatures have been intensively studied since their birth. A precise security model is provided in Pointcheval and Stern's paper [20]. Possible applications of blind signatures can be found in electronic auctions and electronic voting systems. However, the original motivation for the use of such signatures came from e-cash and untraceable payments. Roughly speaking, an electronic coin corresponds to a certain amount of money and it is blindly signed by a bank (therefore, the bank does not know the true value of the coin). It is then withdrawn from the bank, spent by a user, and deposited by a shop.

To make this system more scalable by supporting many banks (to fit with real life scenarios), and to possibly add some other properties like strong anonymity of the signing banks, non linkability of two different signatures, revocation facilities, etc., Lysyanskaya and Ramzan introduced the concept of blind group signatures [17], which combines the concepts of blind signatures and group signatures. Group signatures allow any member of a group to sign a document in such a way that a verifier can confirm that the signature comes from the group, but he does not know which member of

⋆ The work of the first author was carried out during the tenure of an ERCIM fellowship.

the group actually signed the document. The protocol allows for the identity of the signer to be discovered, in case of disputes, by a designated group authority that has some auxiliary information. Group signatures have been introduced by Chaum and van Heyst [14]. Like blind signatures, lots of schemes arose in the literature and one can mention Ateniese, Camenisch, Joye and Tsudik's scheme [2] and Boneh, Boyen and Shacham's pairing-based scheme [8] among the most promising and efficient protocols. The security model for group signatures has been finally properly defined by Bellare, Micciancio and Warinschi in their paper [3]. Ring signatures, introduced by Rivest, Shamir and Tauman [21], are somehow similar to group signatures, but with some important differences: (1) the group is not fixed, but chosen by the actual signer in an ad-hoc way, just before computing the signature; (2) there is no group authority who can recover the identity of the author of a ring signature. Ideally, anonymity in ring and group signature schemes should be satisfied in an unconditional way: no information about the author of a signature must be obtained, even if one has unlimited computational resources. In this way, a signer can be sure that his identity as the author of a signature is perfectly protected for the rest of his life. We refer the reader to Wang's on line bibliography on digital signature [22] for a full overview these different signature schemes.

As we have said before, the first proposed group blind signature scheme is Lysyanskaya and Ramzan's one [17], based on Camenisch and Stadler's group signature scheme with constant size signatures [11]. Applied to the scenario of distributed electronic banking, a central bank behaves as the group authority and monitors the group members, which are banks issuing e-cash. Nguyen, Mu and Varadharajan [19] also proposed a blind variant of Camenisch and Stadler's scheme.

Obviously, combining blind and ring signatures also brings solutions to these scenarios of e-banking, e-voting or e-auctions. Indeed ring signatures provide more spontaneity and flexibility to the design of such systems. Namely, suppose that a client wants some bank to sign some electronic coin corresponding to a certain amount of money; the client can choose ad-hoc a set (or *ring*) of potential signing banks, depending on some conditions (for example, the use that the client is going to make of the obtained coin). If some bank in the ring accepts to sign this coin, it starts running the interactive signing protocol with the client. The bank can therefore preserve its anonymity inside the ring of banks, if desired; on the contrary, if it wants to publicly show its identity, it can simply run a standard (not ring) blind signature scheme, or to use a blind ring signature scheme where the considered ring has this bank as the only member. Summing up, the ring can be chosen by the client or by the actual signer, because of the interactive nature of the protocols, and this increases the number of real-life applications of this kind of schemes.

Only few blind ring signature schemes have been proposed up to now. Chan, Fung, Liu and Wei [12] proposed the first one in 2005. This scheme is obscure and it is unclear who actually engages the different protocols. Furthermore, the proofs provided in the paper are not very convincing. All these facts make us suspect that this scheme does not satisfy some required properties such as blindness or anonymity. Finally, Wu, Zhang, Susilo and Mu have recently described an efficient static blind ring signature [23], with constant signature size and efficient algorithms. In this scheme, each user knows the factorization of an RSA modulus $n_i = p_i q_i$. Basically, the underlying ring signature consists for the signer, given $y = g^{n_1 \cdots n_k} \bmod N$ where N is a public RSA modulus of unknown factorization, and g a generator of $(\mathbb{Z}/N\mathbb{Z}^*)^2$, in proving that he knows p_i and $u = g^{q_i \prod_{j \neq i} n_j}$ such that $y = u^{p_i}$, with p_i in a certain range. An external trusted entity is therefore needed, at least in the setup phase of the system, to generate N. Another drawback of the scheme is that anonymity only holds computationally: an adversary with enough computational resources can factorize all the RSA moduli and automatically obtain the identity of the author of each signature. As discussed above, this is not desirable for some applications; maybe a bank does not want its identity to be revealed in the future as the issuer of some (possibly controversial) e-cash. Furthermore, the unforgeability of this scheme relies on strong (and quite debatable) assumptions like the "extended ROS" one, and is proved in the generic group model (which is stronger than the random oracle model). Even if the authors claim that their scheme supports only static groups, we think that this is not true, and that the client who wants to obtain a blind signature can choose the ring of signers in an ad-hoc way. Apparently, authors of [23] consider only static groups to avoid some attacks against blindness. We think that the blindness property definition only makes sense when the two considered signatures involve the same ring of signers; this is independent of the fact that the scheme can be employed for different rings. See more details on this point in Section 3.1, where we propose a formal and quite natural definition for the blindness property of a blind ring signature scheme.

Our Contributions In this article, we extend Boneh, Gentry, Lynn and Shacham's pairing-based ring signatures [9] by adding the feature of blindness. This scheme accepts in essence the pairing-based blindness techniques described by Boldyreva in [7]. We analyze the security of the resulting blind ring signature scheme by providing first a suitable model for the required properties: anonymity, blindness and unforgeability. Then we prove the security of our new scheme in the random oracle model, under quite standard assumptions, without using the generic model or ROS-like assumptions. Our scheme suffers from the drawback that the number of computations and the size of signatures grow linearly with the number of signers in the ring. This problem is recurrent and inherent to ring signatures supporting dynamic rings, because the description of the ring is necessary to verify a signature. This description usually consists in the set of public keys, and so the length of the (blind) ring signature is always linear with respect to the number of users. Techniques based on accumulators allow to obtain constant-size ring signatures, see [15], when the same ring is used for many signatures.

Our scheme is advantageous with respect to the solutions employing group signatures because it is dynamic, in the sense that the group is chosen "ad-hoc" by the client who wants to obtain a blind signature. Furthermore, neither interaction among the set of users nor initialization phase are required: each user generates his own secret/public keys in an independent way. Contrary to Wu *et al.*'s scheme in [23], the anonymity property is obtained in an unconditional way, which means that the identity of the author of a signature is perfectly protected. Finally, it is easily implemented and based on simple operations, due to the spectacular progress of pairing-based tools.

The rest of the paper is organized as follows: in Section 2, we recall the basics about bilinear pairings and give the computational assumptions (of the chosen-target problem family) which underlie our scheme, and especially the chosen-target-inverse-CDH problem that we prove equivalent to the traditional chosen-target CDH, used in

[7] to prove the unforgeability of the blind signature scheme. Then we precisely define in Section 3 a blind ring signature scheme and the security properties that such a scheme should satisfy. In Section 4, we present our new scheme, and formally prove its security. The conclusions of the work and some open problems are given in Section 5.

2 Bilinear Pairings and Computational Assumptions

In this section, we recall some basic facts about bilinear maps and introduce the computational assumptions needed to prove the security of our scheme.

Definition 1. *Let $\mathbb{G}$ be an additive group of prime order q, generated by some element P. Let $\mathbb{H}$ be a multiplicative group with the same order q.*

A symmetric admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{H}$ *satisfies the following three properties:*

i) it is bilinear;
ii) it can be efficiently computed for any possible input pair;
iii) it is non-degenerate, which means that $e(P, P) \neq 1$.

The typical way of obtaining such pairings is by deriving them from the Weil or the Tate pairing on (hyper-)elliptic curves over a finite field (see for instance [1]).

The security of blind signature schemes is based, in general, on the hardness of the *chosen-target* versions of standard computational problems, such as chosen-target RSA problem [4] for the scheme in [13], or the chosen-target CDH problem for the scheme in [7].

The *Chosen-Target-CDH problem* is defined as follows: the solver $\mathcal{S}$ receives as input a pair (P, aP), where P is a generator of $\mathbb{G}_1$ with prime order q, and $a \in \mathbb{Z}_q$ is a random value. The solver $\mathcal{S}$ has adaptive access to two oracles:

– **target oracle:** this oracle outputs a random element $Z_i \in \mathbb{G}_1$,
– **helper oracle:** this oracle takes as input an element $W_i \in \mathbb{G}_1$ and outputs the element aW_i.

We say that $\mathcal{S}$ (q_t, q_h, d)-solves the Chosen-Target-CDH problem, for $q_t \geq d > q_h$, if it makes q_t and q_h queries, respectively, to the target and helper oracles, and after that it outputs d pairs $((V_1, j_1), \ldots, (V_d, j_d))$ such that:

1. all the elements V_i are different,
2. for all $i \in \{1, 2, \ldots, d\}$, the relation $V_i = aZ_{j_i}$ is satisfied, where Z_{j_i} is the element output by the target oracle in the j_i-th query.

To fit our purpose, we define a very similar problem, which in fact is equivalent (see Prop. 1) to the Chosen-Target-CDH problem. This new problem, that we call *Chosen-Target-Inverse-CDH problem*, is defined as follows: the solver $\mathcal{S}'$ receives as input a pair $(P', a'P')$, where P' is a generator of $\mathbb{G}_1$ with primer order q, and $a' \in \mathbb{Z}_q$ is a random value. The solver $\mathcal{S}'$ has adaptive access to two oracles:

– **target oracle:** this oracle outputs a random element $Z_i \in \mathbb{G}_1$,

– **helper oracle:** this oracle takes as input an element $W_i \in \mathbb{G}_1$ and outputs the element $\frac{1}{a'}W_i$.

We say that $\mathcal{S}'$ (q_t, q_h, d)-solves the Chosen-Target-Inverse-CDH problem, for $q_t \geq d > q_h$, if it makes q_t and q_h queries, respectively, to the target and helper oracles, and after that if outputs d pairs $((V_1, j_1), \ldots, (V_d, j_d))$ such that:

1. all the elements V_i are different,
2. for all $i \in \{1, 2, \ldots, d\}$, the relation $V_i = \frac{1}{a'}Z_{j_i}$ is satisfied, where Z_{j_i} is the element output by the target oracle in the j_i-th query.

Lemma 1. *The Chosen-Target-CDH problem and the Chosen-Target-Inverse-CDH problem are equivalent.*

Proof. We show only one of the implications, since the other one can be proved in an identical way. Let us assume, for example, that there exists $\mathcal{S}$ which (q_t, q_h, d)-solves the Chosen-Target-CDH problem, and let us construct from it a solver $\mathcal{S}'$ which (q_t, q_h, d)-solves the Chosen-Target-Inverse-CDH problem.

$\mathcal{S}'$ receives as input a pair $(P', a'P')$, has access to its target and helper oracles, and wants to solve the Chosen-Target-Inverse-CDH problem. To do this, it initializes the (q_t, q_h, d)-solver $\mathcal{S}$ with input pair $(P, aP) = (a'P', P')$. Note that this means $a = 1/a'$. To obtain from $\mathcal{S}$ a solution of the Chosen-Target-CDH problem, $\mathcal{S}'$ must simulate the environment of $\mathcal{S}$, by answering all the queries that $\mathcal{S}$ makes to its oracles:

– **target oracle:** when $\mathcal{S}$ makes a query to this oracle, $\mathcal{S}'$ makes a query to its own target oracle, and sends to $\mathcal{S}$ the obtained random element $Z_i \in \mathbb{G}_1$;
– **helper oracle:** when $\mathcal{S}$ makes a query W_i to this oracle, $\mathcal{S}'$ makes the same query W_i to its own helper oracle. By definition, the helper oracle of $\mathcal{S}'$ returns the element
$$\frac{1}{a'}W_i = aW_i.$$
Therefore, $\mathcal{S}'$ sends to $\mathcal{S}$ this value, which is consistent with the answers of a real helper oracle for $\mathcal{S}$.

After q_t and q_h queries to the respective oracles, $\mathcal{S}$ finally outputs d pairs $((V_1, j_1), \ldots, (V_d, j_d))$ such that:

1. all the elements V_i are different,
2. for all $i \in \{1, 2, \ldots, d\}$, the relation $V_i = aZ_{j_i} = \frac{1}{a'}Z_{j_i}$ is satisfied, where Z_{j_i} is the element output by the target oracle in the j_i-th query.

Note that such a list of pairs is a valid solution for the instance $(P', a'P')$ of the Chosen-Target-Inverse-CDH problem that $\mathcal{S}'$ received. Therefore, $\mathcal{S}'$ has (q_t, q_h, d)-solved the Chosen-Target-Inverse-CDH problem. □

3 Blind Ring Signature Schemes

Given an integer k, a *blind ring signature scheme* BRS with security parameter k consists of the following four algorithms:

- **generation of public parameters:** $\mathsf{BRS.Setup}$ is a probabilistic algorithm which takes as input k and outputs public parameters (which include a description of the signature space, hash functions, etc.);
- **key generation:** $\mathsf{BRS.KeyGen}$ is a probabilistic algorithm which takes as input the public parameters and outputs a signing key pair (pk_j, sk_j) for a user U_j. The value pk_j is made public, whereas the value sk_j is secretly stored by user U_j.
- **blind ring signature generation:** $\mathsf{BRS.Sign}$ is an interactive 2-party protocol which is initialized by a client $\mathcal{C}$. This client chooses a message M and a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of users, and engages an interaction with some of the members U_s of the ring, who can use his secret key sk_j as part of the input. We denote as $\mathcal{I}_{\mathcal{C}}$ the secret inputs that client $\mathcal{C}$ uses, and as $\mathcal{T}_{sig}$ the values that are obtained by the signer, during this interaction.
  At the end, the private output $\mathcal{O}_{\mathcal{C}}$ for the client is a valid ring signature Σ for the message M and the ring of users $\mathcal{U}$.
- **Verification of a blind ring signature:** $\mathsf{BRS.Verify}$ is a deterministic algorithm which takes as input a message M, a ring of users $\mathcal{U} = \{U_1, \ldots, U_n\}$, their public keys $pk_1, \ldots, pk_n$ and bit string Σ. The output is 1 if the signature is valid, and 0 otherwise.

A blind ring signature scheme must satisfy 4 requirements:

1. *Correctness* means that a verifier always accepts as valid a signature that has been properly generated by a honest client and a honest signer in the corresponding ring of users.
2. *Anonymity* means that the client has no information about which member of the ring has actually participated in the interactive blind ring signature generation.
3. *Blindness* intuitively means that the users in the ring obtain no information about the message that they are actually signing for the client.
4. *Unforgeability* means that a client is not able to produce $\ell + 1$ valid and different ring signatures if he has queried for at most ℓ executions of the blind ring signature protocol.

We now recall the formal definition of the two last properties.

3.1 Blindness

Blindness of a blind ring signature scheme is defined by a game played between a challenger and an adversary. This adversary $\mathcal{B}$ models the dishonest behaviour of a ring of users who try to distinguish which message (between two messages chosen by them) is being signed in an interactive execution of the signing protocol with a client. The game is as follows:

1. **Setup:** the adversary $\mathcal{B}$ chooses a universe $\mathcal{U}^*$ of users and a security parameter k. The challenger runs the setup protocol of the blind signature scheme with input k, as well as the key generation protocol for each user $U_j \in \mathcal{U}^*$. The adversary $\mathcal{B}$ is given all the resulting information: the public common parameters, the public and secret keys of all users in the universe.

2. **Challenge:** the adversary chooses a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of users, and two messages M_0 and M_1. The challenger chooses at random one bit $b \in \{0, 1\}$ and initializes the interactive blind ring signature protocol with message M_b and ring $\mathcal{U}$ as inputs. The adversary $\mathcal{B}$ chooses some user $U_s \in \mathcal{U}$ and plays the role of the signer in the protocol (note that $\mathcal{B}$ knows the secret key of U_s). At the end, the adversary obviously obtains $\mathcal{T}_{sig}$.
3. **Guess:** the adversary $\mathcal{B}$ finally outputs its guess b'.

We say that such an adversary $\mathcal{B}$ succeeds if $b' = b$. A scheme has the blindness property if, for all adversary $\mathcal{B}$, its probability of success in this game is only negligibly bigger than 1/2.

If this probability is exactly 1/2, for any adversary $\mathcal{B}$, then the blindness of the scheme is unconditional. A standard way of proving that a (ring) blind signature scheme enjoys unconditional (or perfect) blindness is by showing that the information $\mathcal{T}_{sig}$, that the signer obtains from an execution of the signing protocol, follows the same probability distribution for any possible message. If this is proved, then in the challenge phase of the game defined above the adversary cannot obtain from $\mathcal{T}_{sig}$ any information about which message M_b is actually being signed, and therefore its success probability (random guess) is limited to 1/2. This is the argument that will be use to analyze the blindness of our blind ring signature scheme.

As opposed to what is claimed by the authors of [23], where they present some "attacks" on the scheme in [12] and they consider only static groups for their scheme to avoid exactly this kind of attacks, we think that a natural definition for blindness in a blind ring signature scheme must consider only one ring of signers. Otherwise, suppose that a member of a ring executes the protocol for two pairs $(m_1, \mathcal{U}_1)$ and $(m_2, \mathcal{U}_2)$ of message/ring, with $\mathcal{U}_1 \neq \mathcal{U}_2$, such that he is in both rings. Later, when seeing the resulting valid signature for some of the two messages, this signature will in particular contain the involved ring, and so he will be trivially able to distinguish which of the two passed executions was indeed the one corresponding to this message. In this way, such an adversary would break this weak notion of blindness. For this reason, we think that our definition is the good one (in particular, in step 2 of the game above, we only consider one ring and not two rings $\mathcal{U}_0$ and $\mathcal{U}_1$). This fact does not imply that a scheme with this blindness property should be used also with one ring (as suggested in [23]). The only point is that the client will only be sure that a blind signature obtained from a ring $\mathcal{U}$ is perfectly hidden and untraceable with respect to all the blind signatures obtained from this particular ring $\mathcal{U}$.

3.2 Unforgeability

Unforgeability for blind ring signatures is adapted from the concept of $(\ell, \ell + 1)$-unforgeability, introduced in [20] and maintained in [4, 7] for standard blind signatures. A $(\ell, \ell + 1, q_i)$-forger $\mathcal{A}$ against a blind ring signature scheme is thus defined by means of the following game that it plays against a challenger:

1. **Setup:** the adversary $\mathcal{A}$ chooses a universe $\mathcal{U}^*$ of users and a security parameter k. The challenger runs the setup protocol of the blind signature scheme with input k, as well as the key generation protocol for each user $U_j \in \mathcal{U}^*$. It gives to the

adversary $\mathcal{A}$ the resulting common parameters and the public keys pk_j, and keeps secret the secret keys sk_j.

2. **Queries:** the forger $\mathcal{A}$ makes different queries to the challenger:
   - q_i hash queries: if the scheme involves some hash function H_i which is assumed to behave as a random oracle [5] in the security proof, then the challenger must answer q_i queries of the adversary to this oracle, providing it with consistent and totally random values.
   - ℓ blind ring signature queries $(M, \mathcal{U})$, where $\mathcal{U} \subset \mathcal{U}^*$: the challenger must answer with a valid blind ring signature Σ for this pair message/ring of users.

   All these queries can be made in an adaptive way; that is, each query may depend on the answers obtained to the previous queries.
3. **Forgery:** the adversary $\mathcal{A}$ outputs a list of $\ell + 1$ tuples $\{(M_i, \mathcal{U}_i, \Sigma_i)\}_{1 \leq i \leq \ell+1}$. We say that $\mathcal{A}$ *succeeds* if:
   - The $\ell + 1$ ring signatures are valid; and
   - $(M_{i_1}, \mathcal{U}_{i_1}) \neq (M_{i_2}, \mathcal{U}_{i_2})$, for all indices $1 \leq i_1, i_2 \leq \ell + 1$ such that $i_1 \neq i_2$.

Note that we require the adversary to output valid blind ring signatures for different pairs message/ring of users. That is, we do not consider as successful, for example, a forger which asks for a valid blind ring signature for the pair $(M, \mathcal{U})$ and later outputs as forgery two valid signatures $(M, \mathcal{U}, \Sigma)$ and $(M, \mathcal{U}, \Sigma')$. Even if we do not consider, with this restriction, all the kinds of adversaries against a blind ring signature scheme, we believe that our model captures the most powerful attacks that such a scheme can suffer in practice. In effect, consider for example the application of blind ring signatures to electronic payments: a message (a coin) is signed by a ring of banks, and later this coin is spent in some electronic transaction. The coin usually contains the date, a serial number, etc., and sellers are assumed to maintain a database with the received pairs coin/ring of banks. Therefore, an attacker which would try to spent two times the same coin, signed by the same ring of banks, should be easily detected.

4 The New Scheme

In this section we propose a blind ring signature scheme quite simple and efficient. It combines the ideas of the ring signature scheme which appears in [9] and the blind signature scheme which appears in [7]. The protocols of the new scheme are described below.

Setup and key generation. On input a security parameter k, an additive group $\mathbb{G}_1$ of prime order $q > 2^k$, generated by some element P, and a multiplicative group $\mathbb{G}_2$ with the same order q are chosen, such that they admit a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ as defined in Section 2. A hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1^*$ is also chosen. All these parameters are common and public.

Each user U_i chooses his secret key $x_i \in \mathbb{Z}_q$ at random; the matching public key is $Y_i = x_i P \in \mathbb{G}_1$.

Blind ring signature generation. The client who wants to obtain a blind ring signature on a message M with respect to a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ of users, proceeds as follows: he chooses at random $r_1, \ldots, r_n \in \mathbb{Z}_q$ and computes the value

$$\bar{M} = H(M,\mathcal{U}) + \sum_{i=1}^{n} r_i Y_i.$$

This value, along with the ring $\mathcal{U}$, is sent to the members of the ring. Then some of these members, say U_s, where $s \in \{1, \ldots, n\}$, acts as follows:

1. For all $i \in \{1, \ldots, n\}$, $i \neq s$, choose a_i uniformly at random in $\mathbb{Z}_q$, and compute $\bar{\sigma}_i = a_i P$.
2. Compute
$$\bar{\sigma}_s = \frac{1}{x_s}\left(\bar{M} - \sum_{i \neq s} a_i Y_i\right).$$
3. Send to the client the tuple $(\bar{\sigma_1}, \ldots, \bar{\sigma_n})$.

The client verifies if

$$e(\bar{M}, P) = \prod_{i=1}^{n} e(\bar{\sigma}_i, Y_i).$$

If so, he computes the values

$$\sigma_i = \bar{\sigma_i} - r_i P, \text{ for all } i = 1, \ldots, n$$

and defines the signature of the message M made by the ring $\mathcal{U} = \{U_1, \ldots, U_n\}$ to be $(M,\mathcal{U}, \sigma_1, \ldots, \sigma_n)$.

Following the notation introduced in Section 3, we have $\mathcal{I_C} = (M, r_1, \ldots, r_n)$, $\mathcal{T}_{sig} = (\mathcal{U}, \bar{M}, \{a_i\}_{i \neq s}, \bar{\sigma_1}, \ldots, \bar{\sigma_n})$ and $\mathcal{O_C} = (M,\mathcal{U}, \Sigma)$, where $\Sigma = (\sigma_1, \ldots, \sigma_n)$.

Verification of a blind ring signature. The validity of the signature $(M,\mathcal{U}, \sigma_1, \ldots, \sigma_n)$ is verified by checking if

$$e(H(M,\mathcal{U}), P) = \prod_{i=1}^{n} e(\sigma_i, Y_i).$$

Correctness and anonymity of the resulting scheme directly infer from the properties satisfied by the aforementioned schemes in [9, 7]. In particular, the anonymity property holds unconditionally: even if a client has unlimited computational resources (which means for example that he can obtain the secret keys of all the members of a ring) he cannot obtain any information about which member has actually participated in the interactive protocol to compute a blind ring signature.

Note that unconditional anonymity directly implies a different property, *unlinkability* [16], which means that nobody (including the client) will be able to distinguish if two different interactive executions of the blind ring signature protocol have been performed by the same member of the ring or not. In effect, if a scheme is linkable, then there exists a polynomial-time linking algorithm which takes as input two executions of the blind ring signature protocol and outputs 1 if and only if the same member of the ring has participated in both executions. If this holds, then a client with unlimited resources who tries to break the anonymity of some execution of the protocol can act as follows: (1) he obtains all the secret keys of the members of the ring; (2) for each member U_i of the ring, the client uses the obtained secret key to run by himself a new interactive execution of the blind ring signature protocol; (3) the client applies the linking algorithm to this last execution and to the initial execution whose anonymity he is trying to break; (4) if the output of the linking algorithm is 1 for user U_i, then this user was the one who participated in the initial (target) execution.

We now prove that the scheme also satisfies the properties of blindness and unforgeability.

4.1 Blindness of the Scheme

As stated in Section 3.1, we can prove that the proposed scheme achieves unconditional blindness if we prove that the probability distribution of the information $\mathcal{T}_{sig}$ that the signer (the adversary in the blindness game) obtains in an execution of the signing protocol is exactly the same for any possible message. In the case of our scheme, we have $\mathcal{T}_{sig} = (\mathcal{U}, \bar{M}, \{a_i\}_{i \neq s}, \bar{\sigma_1}, \ldots, \bar{\sigma_n})$, where $U_s \in \mathcal{U} = \{U_1, \ldots, U_n\}$ is a user chosen by the adversary.

The value $\bar{M} = H(M,\mathcal{U}) + \sum_{i=1}^{n} r_i Y_i$ follows a completely random and uniform distribution in $\mathbb{G}_1$, independently of the message M, because all the integers $r_i \in \mathbb{Z}_q$ are chosen uniformly and at random. For the rest of values in $\mathcal{T}_{sig}$, either they are chosen by the adversary or they depend on $\bar{M}$. In any case, their probability distribution does not depend on the signed message M.

Summing up, during the challenge phase of the blindness game (see Section 3.1), the information that the adversary obtains if the challenger chooses M_0 is perfectly indistinguishable from the information that the adversary obtains if the challenger chooses M_1. Therefore, the scheme achieves perfect blindness.

4.2 Unforgeability of the Scheme

We are going to prove that our scheme is $(\ell, \ell+1)$-unforgeable in the random oracle model, and under the assumption that the Chosen-Target-Inverse-CDH problem is hard to solve. We denote as q_1 the number of queries that an adversary $\mathcal{A}$ against the unforgeability of our scheme can make to the (random) oracle which models the behaviour of the hash function $H : \{0,1\}^* \to \mathbb{G}_1^*$.

Theorem 1. *If there exists a $(\ell, \ell+1, q_1)$-forger $\mathcal{A}$ against the unforgeability of our blind ring signature scheme, which succeeds with probability ε, then there exists a (q_t, q_h, d)-solver $\mathcal{S}'$ of the Chosen-Target-Inverse-CDH problem, which also succeeds with probability $\varepsilon' \geq \varepsilon - \frac{\ell+1}{q}$, where q is the order of the group $\mathbb{G}_1$, $q_t =$, $d = \ell + 1$ and $q_h = \ell$.*

Proof. Assuming the existence of such a forger $\mathcal{A}$, let us construct a solver $\mathcal{S}'$ of the Chosen-Target-Inverse-CDH problem. First of all, $\mathcal{S}'$ initializes $\mathcal{A}$, which chooses a security parameter k and a universe of users $\mathcal{U}^*$. Solver $\mathcal{S}'$ chooses a group $\mathbb{G}_1$ with primer order $q > 2^k$ which admits a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

After that, solver $\mathcal{S}'$ asks for an instance of the Chosen-Target-Inverse-CDH problem in the group $\mathbb{G}_1$. It receives a pair (P', Y'), where $Y' = a'P'$ for some random and secret value $a' \in \mathbb{Z}_q$; it is also provided with access to the target and the helper oracles.

For each user $U_j \in \mathcal{U}^*$, solver $\mathcal{S}'$ defines his public key to be $Y_j = \alpha_j Y'$, for some random value $\alpha_j \in \mathbb{Z}_q^*$. At this point, $\mathcal{S}'$ sends to $\mathcal{A}$ all the common parameters q, $\mathbb{G}_1 = \langle P' \rangle$, $\mathbb{G}_2$, e, the public keys Y_j of all the users U_j in the universe, and provides it with access to a random oracle for a hash function $H : \{0,1\}^* \to \mathbb{G}_1^*$.

Hash queries: the forger $\mathcal{A}$ makes q_h queries $\mathcal{Q}_i = (M_i, \mathcal{U}_i)$ to the random oracle. Solver $\mathcal{S}'$ maintains a table TAB where it stores the relations $H(\mathcal{Q}_i) = Z_i$ that it computes as follows: if a received query $\mathcal{Q}_i = (M_i, \mathcal{U}_i)$ is already in the table, $\mathcal{S}'$ sends to $\mathcal{A}$ the stored value Z_i. If not, $\mathcal{S}'$ makes a query to its target oracle; it receives as answer a random element $Z_i \in \mathbb{G}_1$. Then it stores the new relation $H(\mathcal{Q}_i) = Z_i$ in TAB and sends Z_i to the forger $\mathcal{A}$.

Blind ring signature queries: the forger $\mathcal{A}$ is assumed to initialize ℓ times the interactive blind ring signature protocol, playing the role of the client. Solver $\mathcal{S}'$ must play the role of the signers and simulate the information that $\mathcal{A}$ should obtain in a real execution of this protocol. The forger $\mathcal{A}$ sends a message $\bar{M} \in \mathbb{G}_1$ to be signed by a ring $\mathcal{U} = \{U_1, \ldots, U_n\}$. Then solver $\mathcal{S}'$ acts as follows:

1. it chooses at random a user $U_s \in \mathcal{U}$. For $i \in \{1, \ldots, n\}$, $i \neq s$, the solver chooses random values $a_i \in \mathbb{Z}_q$ and computes $\bar{\sigma}_i = a_i P'$;
2. it sends to the helper oracle the value
$$W_i = \frac{1}{\alpha_s} \left(\bar{M} - \sum_{i \neq s} a_i Y_i \right),$$
and obtains as answer the value $\bar{\sigma}_s = \frac{1}{a'} W_i$;
3. $\mathcal{S}'$ sends to $\mathcal{A}$ the tuple $(\bar{\sigma_1}, \ldots, \bar{\sigma_n})$.

In effect, this tuple perfectly simulates the information that $\mathcal{A}$ would have obtained in a real execution of the protocol, since

$$\prod_{i=1}^{n} e(\bar{\sigma}_i, Y_i) = e(\bar{\sigma}_s, Y_s) \prod_{i \neq s} e(a_i P', Y_i) =$$
$$= e\left(\frac{1}{a'\alpha_s} \left(\bar{M} - \sum_{i \neq s} a_i Y_i \right), \alpha_s a' P' \right) \prod_{i \neq s} e(a_i P', Y_i) =$$
$$= e(\bar{M}, P') \prod_{i \neq s} e(-a_i Y_i, P') e(a_i P', Y_i) = e(\bar{M}, P').$$

The environment of $\mathcal{A}$ is thus perfectly simulated by $\mathcal{S}'$, so with probability ε the forger $\mathcal{A}$ outputs $\ell+1$ tuples $\{(M_i, \mathcal{U}_i, \Sigma_i)\}_{1 \le i \le \ell+1}$ of valid ring signatures such that all the pairs $(M_i, \mathcal{U}_i)$ in these tuples are different. Since the hash function H is assumed to behave as a random function, the probability that $\mathcal{A}$ obtains a valid ring signature for

$(M_i, \mathcal{U}_i)$ without asking for the value $H(M_i, \mathcal{U}_i)$ is $1/q$. Therefore, we have that with probability $1 - \frac{\ell+1}{q}$ the forger $\mathcal{A}$ has queried the random oracle with $(M_i, \mathcal{U}_i)$, for the $\ell + 1$ forged pairs. This means that, for $i = 1, \ldots, \ell + 1$, we have that $H(M_i, \mathcal{U}_i) = Z_{j_i}$ where Z_{j_i} are elements given to $\mathcal{S}'$ by its target oracle. The signatures are valid, so

$$e(H(M_i, \mathcal{U}_i), P') = \prod_{U_j \in \mathcal{U}_i} e(\sigma_{ij}, Y_j) = e\left(\sum_{U_j \in \mathcal{U}_i} \alpha_j a' \sigma_{ij}, P'\right)$$

For $i = 1, \ldots, \ell + 1$, solver $\mathcal{S}'$ outputs the pair (V_i, j_i), where

$$V_i = \sum_{U_j \in \mathcal{U}_i} \alpha_j \sigma_{ij}$$

satisfies $V_i = \frac{1}{a'} H(M_i, \mathcal{U}_i) = \frac{1}{a'} Z_{j_i}$, as desired. Furthermore, since all the pairs $(M_i, \mathcal{U}_i)$ are assumed to be different, we have that all the values V_i are also different.

Summing up, solver $\mathcal{S}'$ makes $q_t \leq q_1$ queries to its target oracle, makes $q_h = \ell$ queries to its helper oracle, and with probability $\varepsilon' \geq \varepsilon - \frac{\ell+1}{q}$ outputs $d = \ell + 1$ valid pairs (V_i, j_i). $\square$

5 Conclusions

We proposed a simple and quite efficient pairing-based ring signature scheme. It is based on Boneh *et al.* ring signatures and on Boldyreva's blind signature, and naturally inherits the advantages and drawbacks of both constructions: the number of scalar multiplications to compute a signature grows linearly with the number of members in the ring, as well as the number of pairing evaluations for the verification, and the size of the signature itself. The scheme remains practical anyway, for rings of reasonable size. Furthermore, it achieves unconditional blindness and anonymity, as opposed to previous blind ring signature schemes. Unforgeeability of the scheme is proved in the random oracle, under some quite standard assumptions.

An open problem would be to build a practical scheme whose unforgeability could be proved in the standard model. Blind signatures and ring signatures without random oracles have been recently proposed [10, 6], so maybe it is possible to combine them and obtain blind ring signatures in the standard model. Another open question deals with the possibility of modifying our scheme so that the size of the signatures becomes constant, independent of the number of signers in the ring. A possible strategy to achieve this could be the use of accumulators based on pairings [18].

References

1. Advances in elliptic curve cryptography, edited by I. F. Blake, G. Seroussi and N. P. Smart, LMS **317**, Cambridge University Press (2005).
2. G. Ateniese, J. Camenisch, M. Joye and G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. *Proceedings of Crypto'00*, LNCS **1880**, Springer-Verlag, pp. 255–270 (2000).

3. M. Bellare, D. Micciancio and B. Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. *Proceedings of Eurocrypt'03*, LNCS **2656**, Springer-Verlag, pp. 614–629 (2003).
4. M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *Journal of Cryptology*, Vol. **16** (3), Springer-Verlag, pp. 185–215 (2003).
5. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. *Proceedings of CCS'93*, ACM Academic Press, pp. 62–73 (1993).
6. A. Bender and J. Katz and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *Proceedings of TCC'06*, LNCS **3876**, Springer-Verlag, pp. 60–79 (2006).
7. A. Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. *Proceedings of PKC'03*, LNCS **2567**, Springer-Verlag, pp. 31–46 (2003).
8. D. Boneh, X. Boyen and H. Shacham. Short Group Signatures. *Proceedings of Crypto'04*, LNCS **3152**, Springer-Verlag, pp. 41–55 (2004).
9. D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. *Proceedings of Eurocrypt'03*, LNCS **2656**, Springer-Verlag, pp. 416–432 (2003).
10. J. Camenisch, M. Koprowski and B. Warinschi. Efficient Blind Signatures Without Random Oracles. *Proceedings of SCN'04*, LNCS **3352**, Springer-Verlag, pp. 134–148 (2005).
11. J. Camenisch and M. Stadler. Efficient Group Signature Schemes for Large Groups. *Proceedings of Crypto'97*, LNCS **1294**, Springer-Verlag, pp. 410–424 (1997).
12. T. K. Chan, K. Fung, J. K. Liu and V. K. Wei. Blind Spontaneous Anonymous Group Signatures for Ad Hoc Groups. *Proceedings of ESAS'04*, LNCS **3313**, Springer-Verlag, pp. 82–84 (2005)
13. D. Chaum. Blind Signatures for Untraceable Payments. *Proceedings of Crypto'82*, pp. 199–203 (1982).
14. D. Chaum, E. van Heyst. Group Signatures. *Proceedings of Eurocrypt'91*, LNCS **547**, Springer-Verlag, pp. 257–265 (1991).
15. Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup. Anonymous Identification in Ad-Hoc Groups. *Proceedings of Eurocrypt'04*, LNCS **3027**, Springer-Verlag, pp. 609–626 (2004).
16. J.K. Liu, V.K. Wei and D.S. Wong. Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups. *Proceedings of ACISP'04*, LNCS **3108**, Springer-Verlag, pp. 325–335 (2004).
17. A. Lysyanskaya and Z. Ramzan. Group Blind Digital Signatures: a Scalable Solution to Electronic Cash. *Proceedings of FC'98*, LNCS **1465**, Springer-Verlag, pp. 184-197 (1998)
18. L. Nguyen. Accumulators from Bilinear Pairings and Applications. *Proceedings of CT-RSA'05*, LNCS **3376**, Springer-Verlag, pp. 275–292 (2005).
19. K. Q. Nguyen, Y. Mu and V. Varadharajan. Divertible Zero-Knowledge Proof of Polynominal Relations and Blind Group Signature. *Proceedings of ACISP'99*, LNCS **1587**, Springer-Verlag, pp. 117–128 (1999).
20. 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).
21. R. L. Rivest, A. Shamir and Y. Tauman. How to Leak a Secret. *Proceedings of Asiacrypt'01*, LNCS **2248**, Springer-Verlag, pp. 552–565 (2001)
22. G. Wang. Bibliography on Digital Signatures. http://www.i2r.a-star.edu.sg/icsd/staff/guilin/bible.htm
23. Q. Wu, F. Zhang, W. Susilo and Y. Mu. An Efficient Static Blind Ring Signature Scheme. *Proceedings of ICISC'05*, LNCS, Springer-Verlag, to appear (2006).