A CDH-Based Ring Signature Scheme with Short Signatures and Public Keys

Sven Schäge, Jörg Schwenk

Horst Görtz Institute for IT-Security, Ruhr-Universität Bochum, Germany

Abstract. In this work we present a new CDH-based ring signature scheme with some striking advantages. On the one hand it is secure without random oracles, perfectly anonymous, and unforgeable solely under the CDH assumption in bilinear groups. This makes the security of our ring signature schemes rely on weaker (and less) assumptions than all previous (full) ring signature schemes secure without random oracles. On the other hand the scheme is very space efficient; a public key consists of just a single group element and a ring signature accounts for only $n+1$ group elements, where n is the size of the ring. This is only about half the number of components when compared to other ring signature schemes that do not exploit ring re-use. As all computations are in groups of prime order, we do not need a trusted setup procedure. All these features do not come for free. The main drawback of our scheme is that it only provides security against chosen subring attacks where the attacker is not allowed to query private keys.

Keywords: CDH assumption, bilinear group, ring signature scheme, programmable hash function

1 Introduction

The CDH assumption became practical for standard model signature schemes with the introduction of bilinear pairings into cryptography. In 2005, Waters showed the existence of a hash-and-sign signature scheme that is secure under the CDH assumption in the standard model [31]. Since then several signature schemes, including ring signature schemes [27], sequential aggregate signature schemes, multisignature schemes, and verifiably encrypted signature schemes [23] have been proposed that are secure in the standard model. In this work we develop a new and efficient ring signature schemes without random oracles that is solely based on the CDH assumption in symmetric bilinear groups.

A ring signature scheme allows a signer to sign on behalf of a group of users, the so-called ring; the only condition is that the signer must also be part of this ring. Technically, a ring is represented by the set of public keys that correspond to the identities of the ring members.

Using his private key, the signer can now sign a message such that anyone can check whether the signature has been generated by one of the ring members. At the same time, there exists no possibility to discover the actual signer. Ring signatures provide signer anonymity in a very strong sense. In contrast to group signature schemes [14], the anonymity of the signer cannot be revoked. What makes ring signature schemes very flexible is that no central management is needed and that the signer can freely choose the public keys in the ring even without their owners' consent. Direct applications for ring signature schemes include designated verifier signatures [22] and secret leaking [26], but ring signature schemes are in general useful in applications where signer anonymity is desired.

1.1 Related Work

The first (explicit) ring signature scheme by Rivest, Shamir and Tauman [26] was proven secure in the random oracle/ideal cipher model [2,16]. Since then, several ring signature schemes have been proposed in the random oracle model. In 1998, Canetti, Goldreich and Halevi showed the existence of a signature scheme that is provably secure in the random oracle model but insecure when instantiated with any hash function [12], thus raising serious doubts on the usefulness of the random oracle model for real world protocols. Since then, research on cryptographic primitives that are secure in the standard model has gained much attention. However, today only a handful of ring signature schemes proven secure without random oracles exist.

While the scheme of Chow et al. [15] published in 2006 provides unconditional anonymity, unforgeability is based on a new strong assumption that is given without any evidence for its validity. In the same year, Bender, Katz and Morselli proposed a ring signature scheme based on trapdoor permutations, but since it uses generic ZAPs for NP it is impractical for real world applications [4]. In the same work the authors also presented two 2-user ring signature schemes without random oracles that are secure under the CDH and the LRSW assumption. Disadvantageously, these schemes only allow to issue signatures on rings of maximal size 2. This is security critical since in a ring signature scheme the provided level of signer anonymity is primarily determined by the number of ring members. Thus, dependent on the application and his requirements on an appropriate security level *the user* should decide on the size of the ring. In 2007, Shacham and Waters presented a ring signature scheme [27] that is full key-exposure anonymous, a strong security notion stemming from [4], under the Subgroup Decision assumption [5]. Unfortunately, this assumption relies on groups with composite order such that a trusted setup procedure is necessary in the setup phase. Also, the representation of group elements is rather large (about 1024 bits). Unforgeability is based on the CDH assumption and the signature size is $2n + 2$ group elements, where n is the size of the ring. In the same year, Boyen presented a new signature scheme with perfect anonymity [7]. Unforgeability of this scheme is based on a new complexity assumption, the Pluri-SDH assumption, while evidence for its usefulness is provided by a security analysis in the generic group model. The signature size consist of n group elements and n integers (of 160 bits) while each public key consists of at least three group elements. Recently, Chandran, Groth and Sahai proposed a new signature scheme with perfect anonymity that is secure under the Subgroup Decision assumption and the Strong Diffie-Hellman assumption [13]. Since the above remarks concerning the trusted setup of [27] also hold here, the authors present two variants of their ring signature scheme. The second variant accounts for maliciously generated common reference strings by heuristically guaranteeing (by using a factorization algorithm) that the composite group order output by the setup algorithm is hard to factor.

Except for the schemes by Chandran et al. [13] and Dodis et al. [17] (in the random oracle model), all existing ring signature schemes offer signature sizes that are at least linear in the ring size. Both, [13] and [17] provide better (asymptotic) efficiency when several messages are signed using the same ring.

1.2 Contribution

In this work we present a new ring signature scheme for rings of arbitrary size. Anonymity is perfect, unforgeability solely relies on the CDH assumption in bilinear groups. Security is proven in the fully untrusted common reference string model. The signature size is very small, accounting for only $n + 1$ group elements. Since we use programmable hash functions [20], a drawback of our scheme is that we require relatively large global parameters, consisting of around 160 group elements. However, these parameters can be re-used for all instantiations of the scheme that use the same bilinear group. Advantageously, in our ring signature scheme, each public key consists of a *single* group element such that for large groups (e.g. >1000), the public parameters only account for a small portion of the data required for signature generation and verification. Finally we provide a new proof technique for Waters-like signature schemes which is very clean and compact at the same time. The main drawback of our scheme is that it only provides security under chosen subring attacks, where the attacker is not allowed to query secret keys of honest ring members. We stress that our ring signature scheme is much more practical than the CDH-based scheme by Bender, Katz, and Morselli that is also secure under the CDH assumption. First, our scheme can be used to sign messages for rings of arbitrary length, not only for 2-user rings. Second, in our scheme a public key contains only a single group element whereas in the Bender et al. scheme a public key consists of a complete group hash function.

2 Preliminaries

Before presenting our constructions we briefly review the necessary preliminaries.

2.1 Ring Signature Scheme

A ring signature scheme $\mathcal{RSIG}$ consists of three algorithms. Given the security parameter 1^κ, the probabilistic polynomial time (PPT) algorithm ***KeyGen*** generates a secret and public key (SK, PK). The PPT algorithm ***Sign*** takes as input a tuple of public keys $R = (PK_1, \ldots, PK_n)$, a secret key SK_i with $i \in \{1, \ldots, n\}$ and a message m and outputs a signature σ. Finally, the deterministic polynomial time algorithm ***Verify*** processes R, a message m and a signature σ to check whether σ is a legitimate signature on m signed by a holder of a secret key corresponding to one of the public keys in R. Accordingly, the algorithm outputs 1 to indicate a successful verification and 0 otherwise. Note that for simplicity, we do not assume an explicit setup algorithm. In the following, all global parameters depend on 1^κ. We stress that we do not rely on a trusted setup authority.

2.2 Ring Unforgeability

In our paper, we concentrate on unforgeability against chosen subring attacks [4]. This definition has been formalized in the following attack game between a challenger and an adversary.

Setup. The challenger runs ***KeyGen*** n times to obtain the key pairs $(SK_1, PK_1), \ldots, (SK_n, PK_n)$. Next, $R = (PK_1, PK_2, \ldots, PK_n)$ is given to the adversary.

Adaptive signature queries. The adversary adaptively sends q signature queries to the challenger. For $i \in \{1, \ldots, q\}$, each query Q_i

consists of a message m_i, a set $R_i \subseteq R$ of public keys and an index $e_i \in \{1, \ldots, n\}$. When the challenger receives the i'th query $Q_i = (m_i, R_i, e_i)$, he computes $\sigma_i = \boldsymbol{Sign}(R_i, SK_{e_i}, m_i)$ and sends it to the adversary.

Output. The attacker outputs a forgery (m^*, R^*, σ^*) such that $(m^*, R^*, \cdot)$ has not been queried before.

We denote the success probability of an adversary $\mathcal{A}$ (taken over the random coins of the challenger and the adversary) to win the above game as $Adv_{\mathcal{RSIG},\mathcal{A},\mathrm{unf}}$.

Definition 1 (Ring unforgeability). *We say that a ring signature scheme is (t, ϵ, q)-secure, if no t-time attacker has success probability at least ϵ in the above attack game after making q signature queries.*

2.3 Ring Anonymity

The strongest notion of anonymity for ring signature schemes is perfect anonymity. Formally, we consider the following attack game between a challenger and an *unbounded* adversary.

Setup. The challenger runs ***KeyGen*** n times to obtain the key pairs $(SK_1, PK_1), \ldots, (SK_n, PK_n)$. The set of the so computed public keys $R = (PK_1, PK_2, \ldots, PK_n)$ is given to the adversary.

Adaptive signature and corrupt queries. The adversary adaptively sends q signature queries $Q_1, \ldots Q_q$ to the challenger and receives the corresponding answers $\sigma_1, \ldots, \sigma_q$. At the same time, the adversary may adaptively query up to n secret keys SK_i with $i \in \{1, \ldots, n\}$.

Output. Finally, the attacker outputs a message m^*, a set of public keys $R^* \subseteq R$ and two distinct indices $i_0, i_1 \in \{1, \ldots, n\}$ such that $PK_{i_0}, PK_{i_1} \in R^*$. The challenger randomly chooses $b \in \{0, 1\}$, computes $\sigma^* = \boldsymbol{Sign}(m^*, R^*, SK_{i_b})$, and sends σ^* to the attacker. The attacker then outputs b', indicating his guess for b.

We denote the advantage of an adversary $\mathcal{A}$ (taken over the random coins of the challenger and the adversary) to win the above game as

$$Adv_{\mathcal{RSIG},\mathcal{A},\mathrm{ano}} = \left|\Pr[\mathcal{A} \text{ outputs } b' = b] - \Pr[\mathcal{A} \text{ outputs } b' \neq b]\right| .$$

Definition 2 (Perfect ring anonymity). *We call a ring signature scheme perfectly anonymous, if even an unbounded adversary has no advantage ($Adv_{\mathcal{RSIG},\mathcal{A},ano} = 0$) in winning the above game.*

2.4 Complexity Assumptions

Definition 3 (Computational Diffie-Hellman problem). *Let $\mathbb{G}$ be a group of prime order. The computational Diffie-Hellman problem (CDH) in $\mathbb{G}$ is, given $g, g^a, g^b \in \mathbb{G}$, to compute $g^{ab} \in \mathbb{G}$.*

We say that algorithm $\mathcal{A}$ (t, ϵ)-solves the CDH problem in $\mathbb{G}$ when, in time t, $\mathcal{A}$ has success probability at least ϵ in breaking the CDH problem such that

$$\Pr\left[g^{ab} \leftarrow \mathcal{A}\left(g, g^a, g^b\right)\right] \geq \epsilon,$$

where the probability is over g, a, b and the random coin tosses of $\mathcal{A}$.

Definition 4. *We say that the (t, ϵ)-CDH assumption holds, if no attacker can (t, ϵ)-solve the CDH problem.*

2.5 Bilinear Groups

In the following, we briefly recall some of the basic properties of bilinear groups. Definitions 6 and 7 help to support the intuition behind our security proof. In [6], Boneh, Mironov and Shoup use a similar approach to describe their tree-based signature scheme. However, in contrast to [6], we focus on proving security under the classical CDH assumption, where the challenge and the solution consist of elements from a single group $\mathbb{G}$. We therefore concentrate on symmetric bilinear groups. We stress that, after some minor modifications, we can base our signature schemes on asymmetric bilinear maps $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with an efficient homomorphism $\varPhi : \mathbb{G}_1 \rightarrow \mathbb{G}_2$. However, security is then based on the co-CDH assumption.

Definition 5 (Bilinear group). *Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p. Let g be a generator of $\mathbb{G}$. The function*

$$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$$

is a bilinear map (pairing) if it holds that $\forall a, b \in \mathbb{G}\ \forall x, y \in \mathbb{Z} : e(a^x, b^y) = e(a, b)^{xy}$ (bilinearity), $e(g, g) \neq 1_{\mathbb{G}_T}$ is a generator of $\mathbb{G}_T$ (non-degeneracy), and e is efficiently computable (efficiency). We call $(\mathbb{G}, g, \mathbb{G}_T, p, e)$ a symmetric bilinear group.

Definition 6 (Secure bilinear map). *A bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is (t, ϵ)-secure if for all t-time adversaries $\mathcal{A}$ it holds that*

$$\Pr\left[e(g, g') = e(h, \mathcal{A}(g, g', h)) \mid\ g, g', h \in_R \mathbb{G}, h \neq 1_{\mathbb{G}}\right] \leq \epsilon,$$

where the probability is taken over the random coin tosses of $\mathcal{A}$ and the random choices of g, g', and h.

One can easily see that in symmetric bilinear groups, breaking the security of a bilinear map is equivalent to breaking the CDH assumption.

Lemma 1. *Let $(\mathbb{G}, g, \mathbb{G}_T, p, e)$ be a symmetric bilinear group. Then, e is (t, ϵ)-secure if and only if the (t, ϵ)-CDH assumption holds in $\mathbb{G}$.*

The proof is straight-forward. For completeness, a proof of Lemma 1 can be found in Appendix A. Let again $(\mathbb{G}, g, \mathbb{G}_T, p, e)$ be a bilinear group with a secure bilinear map e.

Definition 7 (Collision generator for bilinear groups). *A collision generator for e is a polynomial time algorithm that on input two elements $g, h \in \mathbb{G}$ outputs a collision $(g', h') \in \mathbb{G}$ such that*

$$e(g, g') = e(h, h').$$

For symmetric pairings there exists an efficient collision generator that can output *all* possible collisions: given g, h randomly choose $r \in \mathbb{Z}_p$ and compute $g' = h^r$ and $h' = g^r$.

2.6 Multi-Generator Programmable Hash Function

In our (ring) signature schemes we use the multi-generator programmable hash function by Hofheinz and Kiltz in groups with known prime order [20] which in turn is based on the CDH-based signature scheme by Waters [31].

Definition 8 (Multi-Generator PHF). *The multi-generator programmable hash function consists of four algorithms.*

1. *Given 1^κ, $l = l(\kappa)$ and a group $\mathbb{G}$ of prime order p, **GHGen** returns $l + 1$ random group generators $u_0, u_1, \ldots, u_l \in \mathbb{G}$.*
2. *Given the u_i and a message $m \in \{0, 1\}^l$, **GHEval** outputs*

$$u(m) = u_0 \prod_{i=1}^{l} u_i^{m_i},$$

*where $(m_l, \ldots, m_1)$ is the binary representation of m: $m = \sum_{i=1}^{l} m_i 2^{i-1}$. The pair (**GHGen**, **GHEval**) is called a group hash function.*

3. *On input 1^κ, l and generators $g, h \in \mathbb{G}$, the algorithm **PHTrapGen** randomly chooses $a_0', a_1, \ldots, a_l \in \{-1, 0, 1\}$ and $b_0, b_1, \ldots, b_l \in \mathbb{Z}_p$. Next, it sets $a_0 = a_0' - 1$ and outputs $l + 1$ group elements $u_i = g^{a_i} h^{b_i}$ for $i = 0, 1, \ldots, l$ and the trapdoor $(a_0, a_1, \ldots, a_l, b_0, b_1, \ldots, b_l)$.*

4. *Now, given $(a_0, a_1, \ldots, a_l, b_0, b_1, \ldots, b_l)$ and a message m, the algorithm* ***PHTrapEval*** *outputs $a(m) = a_0 + \sum_{i=1}^{l} a_i m_i$ and $b(m) = b_0 + \sum_{i=1}^{l} b_i m_i$. Note that when the u_i have been computed by* ***PHTrapGen*** *it clearly holds that $u(m) = u_0 \prod_{i=1}^{l} u_i^{m_i} = g^{a(m)} h^{b(m)}$.*

Hofheinz and Kiltz showed that for every fixed polynomial $q = q(\kappa)$ the multi-generator programmable hash function is $(1, q, 0, P_{q,l})$-programmable where $P_{q,l} = \mathcal{O}\left(\frac{1}{q\sqrt{l}}\right)$. This means, that the group elements output by ***GHGen*** and ***PHTrapGen*** are *equally* distributed. Furthermore it holds for all possible input parameters to ***PHTrapGen*** and all $M_1, \ldots, M_{q+1} \in \{0,1\}^l$ with $M_{q+1} \neq M_i$ for $i \leq q$ that

$$\Pr[a(M_{q+1}) = 0 \ \wedge \ a(M_1), \ldots, a(M_q) \neq 0] \geq P_{q,l}.$$

The corresponding proof and further details on programmable hash functions can be found in the original paper [20]. A similar but weaker result ($P_{q,l} = \frac{1}{8(l+1)q}$) was implicitly given by Waters in [31].

3 Efficient Ring Signature Scheme $\mathcal{RS}$

In this section we present our ring signature scheme $\mathcal{RS}$ that allows for very short public keys and signatures. In $\mathcal{RS}$, the global parameters consist of $l + 2$ random elements $h, u_0, u_1, \ldots, u_l \in \mathbb{G}$ that give rise to a group hash function $u(m) = u_0 \prod_{j=1}^{l} u_j^{m_j}$ and a symmetric bilinear group $(\mathbb{G}, g, \mathbb{G}_T, p, e)$ with a secure bilinear map.

KeyGen(1^κ)**.** Each user i chooses a random element $x_i \in \mathbb{Z}_p$ as his secret key SK_i. The corresponding public key is $PK_i = g^{x_i}$.

Sign$(PK_1, \ldots, PK_n, SK_t, m)$**.** Given a ring of n public keys, the holder of secret key SK_t with $t \in \{1, \ldots, n\}$ can sign a message $m \in \{0,1\}^l$ in the following way: for all $i \in \{1, \ldots, n+1\} \setminus \{t\}$ he chooses $r_i \in_R \mathbb{Z}_p$ and sets

$$s_i = g^{r_i}.$$

Then, he computes

$$s_t = \left(h \cdot \prod_{\substack{i=1 \\ i \neq t}}^{n} PK_i^{-r_i} \cdot \left(u_0 \prod_{j=1}^{l} u_j^{m_j} \right)^{-r_{n+1}} \right)^{1/x_t}.$$

The final signature is $\sigma = (s_1, \ldots, s_{n+1})$.

Verify$(PK_1, \ldots, PK_n, m, \sigma)$**.** Given a set of n public keys, a message m, and a ring signature $\sigma = (s_1, \ldots, s_{n+1})$, verify the following equation:

$$\prod_{i=1}^{n} e(s_i, PK_i) \cdot e\left(s_{n+1}, u_0 \prod_{j=1}^{l} u_j^{m_j}\right) \overset{?}{=} e(g, h) \ .$$

4 Security

In this section, we show that $\mathcal{RS}$ provides ring unforgeability and perfect ring anonymity according to Definition 1 and 2 (correctness can easily be verified by inspection).

4.1 Ring Unforgeability

Theorem 1. *Suppose the* $(t_{CDH}, \epsilon_{CDH})$*-CDH assumption holds in the group* $\mathbb{G}$*. Then the ring signature scheme* $\mathcal{RS}$ *is* (t, ϵ, q)*-secure against chosen subring attacks provided that*

$$\epsilon \leq \epsilon_{CDH}/P_{q,l}, \quad t \approx t_{CDH}.$$

Proof. By contradiction. Assume there exists an adversary $\mathcal{A}$ that breaks the security of the ring signature scheme in time t with probability ϵ after q signature queries. Then, one can construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ as a black box to solve the CDH assumption. We assume that $\mathcal{B}$ is given a random challenge for the CDH-problem: $\left(\bar{g}, \bar{g}^a, \bar{g}^b\right) \in \mathbb{G}^3$. The main idea behind our proof is the following. Recall Definition 7 and Lemma 1. Given two group elements $g, h \in \mathbb{G}$, it is easy to generate all pairs $(g', h') \in \mathbb{G}^2$ such that $e(g, g') = e(h, h')$. On the other hand, given three group elements g, g', h, the problem of finding a corresponding h' is as hard as solving the CDH problem. Our aim is to transfer this situation to the unforgeability game of our ring signature scheme: the simulator has to choose the input parameters for the attacker such that answering signature queries is as easy as computing collisions and computing a forgery is as hard as breaking the CDH assumption.

In the following, we provide a proof of security that proceeds in a sequence of games [3,28]. Let $\Pr[S_i]$ denote the success probability for an attacker to successfully forge signatures in Game i.

Game$_0$. This is the original attack game. By assumption, attacker $\mathcal{A}$ (t, ϵ, q)-breaks $\mathcal{RS}$ when interacting with the challenger. We have,

$$\Pr[S_0] = \epsilon \tag{1}$$

Game$_1$. This game is like the previous one except that $\mathcal{B}$ constructs the global parameters and the secret and public keys using the algorithms of the programmable hash function and the CDH challenge. First, $\mathcal{B}$ randomly chooses: n elements $x_i \in_R \mathbb{Z}_p$ for $i = 1, \ldots, n$, $l + 1$ elements $a'_0, a_1, \ldots, a_l \in_R \{-1, 0, 1\}$, and $l + 1$ elements $b_0, b_1, \ldots, b_l \in_R \mathbb{Z}_p$. Let $a_0 = a'_0 - 1$. Then, for all $i \in \{1, \ldots, n\}$ and $j \in \{0, \ldots, l\}$ $\mathcal{B}$ computes

$$g := \bar{g}^a,\ h := \bar{g}^b,\ PK_i := \bar{g}^{x_i},\ u_j := h^{a_j} \bar{g}^{b_j}.$$

using the CDH challenge. Due to the properties of the multi-generator programmable hash function the distribution of the so computed values is equal to the distribution in the previous game. Thus,

$$\Pr[S_1] = \Pr[S_0]\ . \tag{2}$$

Game$_2$. Now, $\mathcal{B}$ simulates the challenger in the attack game by answering $\mathcal{A}$'s signature queries (m_j, R_j, e_j). For convenience, let $a(m) = a_0 + \sum_{i=1}^{l} a_i m_i$ and $b(m) = b_0 + \sum_{i=1}^{l} b_i m_i$. Let $I[j] \subset \{1, \ldots, n\}$ be the set of all indices $i \in \{1, \ldots, n\}$ such that PK_i is a component of R_j. When receiving a signature query, $\mathcal{B}$ at first tests whether $a(m_j) = 0$. In this case, $\mathcal{B}$ outputs the failure signal $\mathtt{F}_1$ and aborts. Otherwise $\mathcal{B}$ chooses $r \in_R \mathbb{Z}_p$ and computes a collision $(d_{\bar{g}}, d_h)$ as $d_{\bar{g}} = h^r$ and $d_h = \bar{g}^r$. Note that by construction $e(d_{\bar{g}}, \bar{g}) = e(d_h, h)$.

The aim of $\mathcal{B}$ is to compute $s_{n+1} \in \mathbb{G}$ and $|I[j]|$ values $s_i \in \mathbb{G}$ (for all $i \in I[j]$) such that

$$\prod_{i \in I[j]} e(s_i, PK_i) \cdot e(s_{n+1}, u(m_j)) = e(g, h)$$

or equivalently

$$e\left(s_{n+1}^{b(m_j)} \cdot \prod_{i \in I[j]} s_i^{x_i}, \bar{g}\right) = e\left(g s_{n+1}^{-a(m_j)}, h\right).$$

In the next step, $\mathcal{B}$ chooses $y \in_R I[j]$ and for all $i \in I[j] \setminus \{y\}$ $s_i \in_R \mathbb{G}$. The values s_y and s_{n+1} are computed in the following way:

$$s_{n+1} = \left(g d_h^{-1}\right)^{1/a(m_j)},\ s_y = \left(d_{\bar{g}} \cdot s_{n+1}^{-b(m_j)} \cdot \prod_{i \in I[j] \setminus \{y\}} s_i^{-x_i}\right)^{1/x_y}.$$

$\mathcal{B}$ outputs the ring signature $\sigma = (s_1, s_2, \ldots, s_n, s_{n+1})$. The probability for $\mathcal{B}$ to win this game is

$$\Pr[S_2] = \Pr[S_1 \wedge \bar{\mathtt{F}_1}] \ . \tag{3}$$

Game₃. In this game $\mathcal{B}$ uses $\mathcal{A}$'s forgery $(m^*, R^*, \sigma^* = (s_1^*, s_2^*, \ldots, s_{n+1}^*))$ to break the CDH assumption. Adversary $\mathcal{B}$ at first checks whether $a(m^*) = 0$. If not, $\mathcal{B}$ outputs the failure signal $\mathtt{F}_2$ and aborts. We get that

$$\Pr[S_3] = \Pr[S_2 \wedge \bar{\mathtt{F}_2}] \ . \tag{4}$$

Otherwise, $\mathcal{B}$ computes the solution to the CDH problem as follows. Since $a(m^*) = 0$, we get that

$$e\left((s_{n+1}^*)^{b(m^*)} \cdot \prod_{i \in I[*]} (s_i^*)^{x_i}, \bar{g}\right) = e(g, h) \ \Leftrightarrow \ \bar{g}^{ab} = (s_{n+1}^*)^{b(m^*)} \cdot \prod_{i \in I[*]} (s_i^*)^{x_i}$$

what constitutes a solution to the CDH challenge.
We finally have

$$\Pr[S_3] = \epsilon_{\mathrm{CDH}} \ . \tag{5}$$

Now, let us analyze the probabilities for an abort, i.e. for one of the events $\mathtt{F}_1$ or $\mathtt{F}_2$ to occur. Surely, the probability that both failure events do not occur depends on the number of signature queries q and the bit size l of the messages. Since, $u(m)$ is generated by the multi-generator programmable hash function as defined in Sect. 2.6, we can directly apply the results from [20] to show that

$$\Pr[\bar{\mathtt{F}_1} \wedge \bar{\mathtt{F}_2}] \geq P_{q,l} \ .$$

Putting (1-5) together, we get that

$$\epsilon_{\mathrm{CDH}} = \Pr[S_0 \wedge \bar{\mathtt{F}_1} \wedge \bar{\mathtt{F}_2}] = \Pr[S_0 | \bar{\mathtt{F}_1} \wedge \bar{\mathtt{F}_2}] \cdot \Pr[\bar{\mathtt{F}_1} \wedge \bar{\mathtt{F}_2}] \geq \epsilon \cdot P_{q,l}$$

which proves Theorem 1.

4.2 Ring Anonymity

Theorem 2. *The ring signature scheme $\mathcal{RS}$ is perfectly secure.*

We give an information theoretic argument. Given a ring signature, we have to show that each ring member could possibly have created it. Consider a ring signature on message m, that has been created using SK_z. We show that with the same probability it could have been created using SK_y with $y \neq z$. The proof is straight-forward.

Proof. Fix an arbitrary ring R of n public keys and choose two indices $y, z \in_R \{1, \ldots, n\}$. Next, fix a random $m \in \{0,1\}^l$ and $n-1$ random values r_i with $i \in \{1, \ldots, n+1\} \setminus \{y, z\}$. We show that for any r_y there exists an r_z such that the final signatures generated by ***Sign*** with either (r_y, SK_z) or (r_z, SK_y) are equal. Since $\mathbb{G}$ is a cyclic group with prime order p, there exists $t \in \mathbb{Z}_p$ and $b(M) = b_0 + \sum_{i=1}^{l} M_i b_i$ with $b_i \in \mathbb{Z}_p$ such that $h = g^t$ and $u(M) = g^{b(M)}$ for all $M \in \{1, \ldots, n\}$.
Let the ring signature consist of all $s_i = g^{r_i}$ with $i \in \{1, \ldots, n\} \setminus \{y, z\}$. Then, the remaining s_y, s_z are computed using SK_z and the ***Sign*** algorithm as

$$s_y = g^{r_y},\ s_z = \left(h \cdot \prod_{\substack{i=1\\ i\neq z}}^{n} PK_i^{-r_i} \cdot \left(u_0 \prod_{j=1}^{l} u_j^{m_j} \right)^{-r_{n+1}} \right)^{1/x_z} .$$

Now, let $r_z = \frac{t - \sum_{i=1, i\neq z}^{n} r_i x_i - r_{n+1} b(m)}{x_z}$. Using SK_y we get $s_z = g^{r_z}$ and

$$s_y = \left(h \cdot \prod_{\substack{i=1\\ i\neq y}}^{n} PK_i^{-r_i} \cdot \left(u_0 \prod_{j=1}^{l} u_j^{m_j} \right)^{-r_{n+1}} \right)^{1/x_y} = g^{r_y}$$

with $r_y = \frac{t - \sum_{i=1, i\neq y}^{n} r_i x_i - r_{n+1} b(m)}{x_y}$ what concludes the proof of Theorem 2.

4.3 Digital Signature Schemes

Our new proof technique can also be applied to other CDH based signature schemes. For example, we can surprisingly easy obtain as a special case ($n = 1$) of our ring signature scheme a variant $\mathcal{S}$ of the Waters signature scheme that has distinct setup and sign algorithms but the same verification equation. We briefly compare it with the original scheme by Waters in Table 1. For completeness, we also describe a third variant $\mathcal{S}_0$ where the group hash function constitutes the public key of the user. Both schemes can easily be proven secure under the standard notion of security for digital signatures by Goldwasser, Micali and Rivest [18] by adapting the proof of Theorem 1.

5 Conclusion

In this work, we presented an efficient and perfectly anonymous ring signature scheme that is secure under chosen subring attacks in symmetric

Table 1. Comparison of the Waters signature scheme and $\mathcal{S}$ and $\mathcal{S}_0$. Unless not stated otherwise, all values are elements of $\mathbb{G}$. We set $u(m) = u_0 \prod_{i=1}^{l} u_i^{m_i}$ and $x(m) = x_0 + \sum_{i=1}^{l} x_i m_i$.

	Waters [31]	$\mathcal{S}$	$\mathcal{S}_0$
publ. params.	$g_0, h, u_0, \dots, u_l$	$g, h, u_0, \dots, u_l$	g_0, g, h
SK	h^x	$x \in \mathbb{Z}_p$	$x_0, \dots, x_l \in \mathbb{Z}_p$
PK	$g = g_0^x$	$g_0 = g^x$	$u_0 = g^{x_0}, \dots, u_l = g^{x_l}$
s_1	$h^x \cdot (u(m))^r$	$(h \cdot (u(m))^r)^{\frac{1}{x}}$	g^{-r}
s_2	g_0^{-r}	g^{-r}	$(hg_0^r)^{\frac{1}{x(m)}}$
verification	$e(s_1, g_0) \cdot e(s_2, u(m)) \overset{?}{=} e(g, h)$		

bilinear groups with a secure bilinear map. Additionally, we developed an new technique for proving Waters-like signature schemes secure that uses $(1, poly)$-programmable hash functions and results in very clean and compact security proofs. In our ring signature scheme, each public key consists of a single group element, while the signature size only accounts for $n + 1$ group elements, where n is the size of the ring. When compared to all other ring signature schemes that are proven secure in the standard model and do not assume ring re-use, this is extremely efficient. Finally, we stress that using the generic transformation by Huang, Wong and Zhao [21] all presented schemes can be made strongly unforgeable, meaning that we also consider new signatures on previously queried messages as forgeries in the attack game. The overhead of this transformation is very small; the signature is extended by just a public key and an one-time signature, while no additional key material is required.

References

1. Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-TAA. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 111–125. Springer, 2006.
2. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
3. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Vaudenay [29], pages 409–426.
4. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Halevi and Rabin [19], pages 60–79.
5. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In Joe Kilian, editor, *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341. Springer, 2005.

6. Dan Boneh, Ilya Mironov, and Victor Shoup. A secure signature scheme from bilinear maps. In Marc Joye, editor, *CT-RSA*, volume 2612 of *Lecture Notes in Computer Science*, pages 98–110. Springer, 2003.
7. Xavier Boyen. Mesh signatures. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 2007.
8. Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *ACM Conference on Computer and Communications Security*, pages 132–145. ACM, 2004.
9. Jan Camenisch and Els Van Herreweghen. Design and implementation of the *demix* anonymous credential system. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security*, pages 21–30. ACM, 2002.
10. Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002.
11. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.
12. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *STOC*, pages 209–218, 1998.
13. Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434. Springer, 2007.
14. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.
15. Sherman S. M. Chow, Victor K.-W. Wei, Joseph K. Liu, and Tsz Hon Yuen. Ring signatures without random oracles. In Ferng-Ching Lin, Der-Tsai Lee, Bao-Shuh Lin, Shiuhpyng Shieh, and Sushil Jajodia, editors, *ASIACCS*, pages 297–302. ACM, 2006.
16. Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In Wagner [30], pages 1–20.
17. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.
18. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.
19. Shai Halevi and Tal Rabin, editors. *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*. Springer, 2006.
20. Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In Wagner [30], pages 21–38.
21. Qiong Huang, Duncan S. Wong, and Yiming Zhao. Generic transformation to strongly unforgeable signatures. In Jonathan Katz and Moti Yung, editors, *ACNS*, volume 4521 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2007.
22. Markus Jakobsson, Kazue Sako, and Russell Impagliazzo. Designated verifier proofs and their applications. In *EUROCRYPT*, pages 143–154, 1996.

23. Steve Lu, Rafail Ostrovsky, Amit Sahai, Hovav Shacham, and Brent Waters. Sequential aggregate signatures and multisignatures without random oracles. In Vaudenay [29], pages 465–485.
24. Tatsuaki Okamoto. Efficient blind and partially blind signatures without random oracles. In Halevi and Rabin [19], pages 80–99.
25. Giuseppe Persiano and Ivan Visconti. An efficient and usable multi-show non-transferable anonymous credential system. In Ari Juels, editor, *Financial Cryptography*, volume 3110 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 2004.
26. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2001.
27. Hovav Shacham and Brent Waters. Efficient ring signatures without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180. Springer, 2007.
28. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs, manuscript, nov. 30, 2004. revised version from jan. 18, 2006., 2004.
29. Serge Vaudenay, editor. *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*. Springer, 2006.
30. David Wagner, editor. *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*. Springer, 2008.
31. Brent Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2005.

A Proof of Lemma 1

Proof. By contradiction. Let $(\mathbb{G}, g, \mathbb{G}_T, p, e)$ be our bilinear group. First, assume attacker $\mathcal{A}$ can break the security of the bilinear map in time t with advantage at least ϵ. Then, algorithm $\mathcal{B}$ can solve the CDH assumption in $\mathbb{G}$ in time t with advantage ϵ by using $\mathcal{A}$ as a black-box. Let $\bar{g}, \bar{g}^a, \bar{g}^b$ be $\mathcal{B}$'s CDH challenge in group $\mathbb{G}$. $\mathcal{B}$ sets $\tilde{g} = \bar{g}^a$, $\tilde{g}' = \bar{g}^b$, and $\tilde{h} = \bar{g}$ and runs attacker $\mathcal{A}$ on $(\tilde{g}, \tilde{g}', \tilde{h})$. As a result, $\mathcal{A}$ outputs $\tilde{h}'$ such that $e(\tilde{g}, \tilde{g}') = e(\tilde{h}, \tilde{h}')$. Since equivalently $e(\bar{g}^a, \bar{g}^b) = e(\bar{g}, \tilde{h}')$, $\tilde{h}'$ is a solution to the CDH problem.

Now, assume adversary $\mathcal{A}$ (t, ϵ)-breaks the CDH assumption in $\mathbb{G}$. Let $\tilde{g}, \tilde{g}', \tilde{h} \in \mathbb{G}$, $\tilde{h} \neq 1_{\mathbb{G}}$ be $\mathcal{B}$'s challenge against the security of the bilinear map. Since $\tilde{h}$ is a generator, there exist $a, b \in \mathbb{Z}_p$ such that $\tilde{h}^a = \tilde{g}$, and $\tilde{h}^b = \tilde{g}'$. $\mathcal{B}$ runs $\mathcal{A}$ on $\tilde{h}, \tilde{g}, \tilde{g}'$. Because $\mathcal{A}$ outputs $\tilde{h}^{ab}$, we have that $e(\tilde{g}, \tilde{g}') = e(\tilde{h}, \tilde{h}^{ab})$, and thus $\mathcal{A}$'s output is a correct solution to $\mathcal{B}$'s challenge.