IET Information Security

Research Article

IET Journals
The Institution of Engineering and Technology

Logarithmic size ring signatures without random oracles

ISSN 1751-8709
Received on 25th February 2014
Revised on 23rd March 2015
Accepted on 5th April 2015
doi: 10.1049/iet-ifs.2014.0428
www.ietdl.org

Clémentine Gritti ✉, *Willy Susilo*, *Thomas Plantard*

Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, Australia
✉ *E-mail: cjpg967@uowmail.edu.au*

Abstract: Ring signatures enable a user to anonymously sign a message on behalf of group of users. In this study, the authors propose the first ring signature scheme whose size is $O(\log_2 N)$, where N is the number of users in the ring. They achieve this result by improving Chandran *et al.*'s ring signature scheme presented at the International Colloquium on Automata, Languages and Programming 2007. Their scheme uses a common reference string and non-interactive zero-knowledge proofs. The security of their scheme is proven without requiring random oracles.

1 Introduction

The notion of ring signature was put forth by Rivest *et al.* [1] in 2001. In such a scheme, anyone can sign a message on behalf of an *ad-hoc* created group (i.e. the ring) anonymously. In 2007, Chandran *et al.* [2] presented a novel approach to achieve a sub-linear size ring signature scheme without random oracles, with perfect anonymity in the common reference string model. Their scheme is proven secure under the strong Diffie–Hellman and the subgroup decision assumptions, by setting the ring as a $\sqrt{N} \times \sqrt{N}$ matrix for N members. In this work, we aim to further reduce the size of a ring signature, which is a very challenging task.

1.1 Our contributions

In this paper, we provide the first ring signature with logarithmic size without random oracles. To achieve our result, we extend the idea proposed by Chandran *et al.* [2]. We construct our scheme following their techniques using composite order groups with a bilinear map. We prove it secure under the strong Diffie–Hellman and the subgroup decision assumptions. We obtain perfect anonymity in the common reference string model.

The crux of our scheme is that we achieve $\mathcal{O}(\log_2(N))$ for the ring signatures size, where N is the number of members in the ring. In Table 1, we compare our scheme with the one from Chandran *et al.* [2] to highlight the difference in performance.

1.2 Our technique

The novelty of our scheme is to construct the ring as a $\log_2(N)$-dimensional hypercube for N members, which yields a logarithmic size ring signature scheme. A d-dimensional hypercube has $N = 2^d$ vertices and $d2^{d-1}$ edges. Each vertex corresponds to a d-bit binary string and two vertices are linked with an edge if and only if their binary strings differ in precisely one bit. Therefore each vertex is adjacent to $d = \log_2(N)$ other vertices, one for each bit position. We illustrate the hypercubes with N equal to 2, 4 and 8 in Fig. 1.

In [2], a grid is picked such that the diameter is of the square root of the number of the points on the graph, that is, $\sqrt{N}$. In our paper, we consider the hypercube as a graph which has the smallest diameter for a given number of points. Thus, the diameter is of the logarithm of number of the points on the graph, that is, $\log_2(N)$.

In our approach, we use a $\log_2(N)$-dimensional hypercube as a N-member ring to construct the signature. Each verification key v in the ring is indexed by a d-bit string, denoted as $b_1 b_2 \ldots b_d$. To retrieve v, we need to follow the path formed by all the bits, from b_1 to b_d. We obtain v as the vertex corresponding to $b_1 \ldots b_d$. Moreover, the signature related to the verification key v has an equal size to the length of the path between two vertices of the hypercube, that is, between two points of the graph. We illustrate the graph with N equal to 16 in Fig. 2.

In Fig. 3, we compare the paths in the grid and the hypercube. The paths start from point (resp. vertex) 0010 and finish at point (resp. vertex) 1100. We note that in the grid, the path is five-edge long, whereas in the hypercube, it is only three-edge long. This is due to that in a hypercube, we reach intermediate vertices that differ from their direct neighbours in only one bit. Since 0010 and 1100 differ in three bits (only the last bit remains unchanged), we must pass through 1010 and 1110 to reach 1100. In the grid, we pass through 'useless' points 0110 and 1101, increasing the number of edges in the path's length. The diameter of a hypercube is always shorter than the one of the grid and the same property holds for the paths between two given vertices (resp. points). While Chandran *et al.* [2] constructed their scheme based on a $\sqrt{N} \times \sqrt{N}$ matrix, which is treated as a grid, our construction relies on structure of a hypercube. Hence, in our scheme, the verification keys will be the vertices of a hypercube of dimension $\log_2(N)$ and the size of the signature will depend on the path built to reach a targeted verification key.

1.3 Related work

Ring signatures have been found very promising in many practical applications [3, 4]. Rivest *et al.* [1] proved their unconditional anonymous scheme is secure in the random oracle model. Zhang and Kim [5] incorporated the notion of identity-based cryptography to avoid the necessity of incorporating certificates. Subsequently, Au *et al.* constructed a certificate-based ring signature scheme in [6]. Traceable ring signature was proposed by Fujisaki and Suzuki [7]. Liu *et al.* [8] presented the first linkable ring signature scheme satisfying anonymity, linkability and spontaneity. Wang and Liu [9] introduced the notion of signer-admission ring signature, which is a combination of designated confirmer signatures and designated verifier proofs. In most practical applications, the description of the ring is linear to the number of members, but Dodis *et al.* [4] proposed a scheme that is independent of the size of the ring in the random oracle model. Chow *et al.* [10] constructed a scheme that proved secure against the adaptive chosen message attack without random

Table 1 Comparison of the size of the ring signature and the number of elements in the signature between Chandran *et al.*'s work [2] and ours, including the size of the common reference string

Scheme	Size of the common reference string	Size of the ring signature	Number of elements in the signature	Typical values for $k = 128$		
				$N = 1000$	$N = 10\,000$	$N = 100\,000$
Chandran *et al.*'s [2]	$O(k)$	$\mathcal{O}(k\sqrt{N})$	$6 + 6\lceil k\sqrt{N} \rceil$	24292	76806	242869
our approach	$O(k)$	$\mathcal{O}(k\log_2(N))$	$6 + 7\lceil k\log_2(N) \rceil$	8935	11912	14888

Let N be the number of members and k be the security parameter.

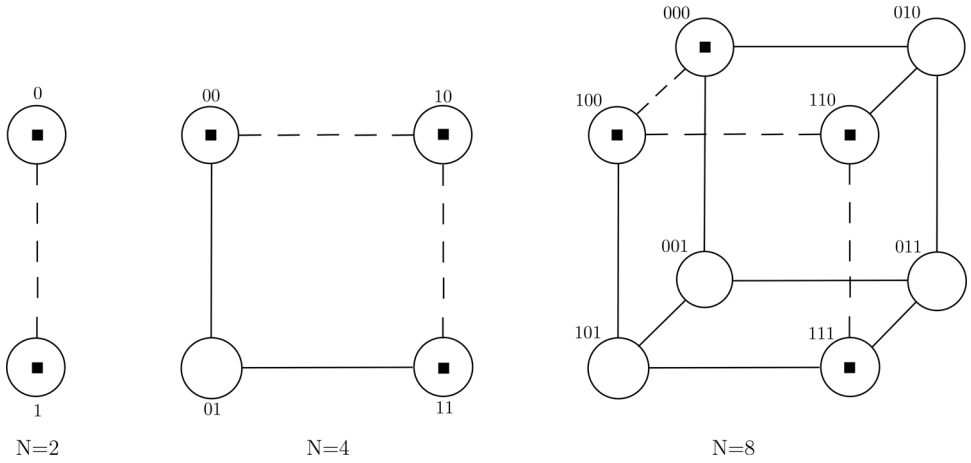

Fig. 1 *N-vertex hypercube for N = 2, 4, 8*

Two vertices are linked with an edge if and only their string differs in precisely one bit position. Diameters are shown in dashed line

oracles. In the same way, Bender *et al.* [11] suggested a scheme using generic ZAPs for non-deterministic polynomial time (NP) in the standard model, but it seems impractical. In addition, Shacham and Waters [12] gave a linear size ring signature, whose security relies on the computational setting of the new definitions of [11], without random oracles. Namely, they proposed a scheme anonymous against full key exposure and unforgeable with respect to insider corruption attacks. Finally, Boyen [13] proposed a construction of linear size in the common random string model with everlasting perfect anonymity. Schäge and Schwenk [14] constructed another ring signature scheme in the standard model using basic assumptions.

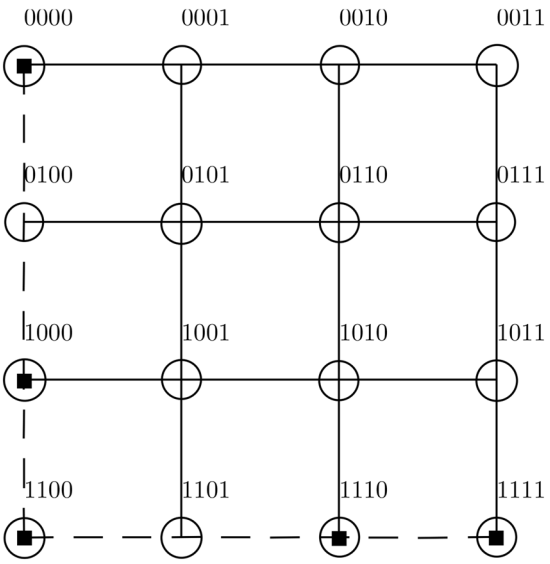

Fig. 2 *Diameter is shown in dashed line and the squares represent the points through which the diameter passes; N = 16*

2 Preliminaries and definitions

2.1 Negligible function

Let $\mathrm{negl}(k)$ be a function in the security parameter k. We say that $\mathrm{negl}(k)$ is a 'negligible function' if for all polynomials $p(k)$, for all sufficiently large k, $\mathrm{negl}(k) < 1/p(k)$.

2.2 Bilinear composite order groups

Let **BMGen** be a randomised algorithm that outputs $(p, q, \mathbb{G}, \mathbb{G}_T, e, g)$ as follows:

- $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups of order $n = pq$,
- g is a generator of $\mathbb{G}$,
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable map such that:
  - *Bilinearity:* $\forall u, v \in \mathbb{G}, \forall a, b \in \mathbb{Z}_n$, $e(u^a, v^b) = e(u, v)^{ab}$,
  - *Non-degeneracy:* $e(g, g)$ is a generator of $\mathbb{G}_T$ whenever g is a generator of $\mathbb{G}$,
- the group operations on $\mathbb{G}$ and $\mathbb{G}_T$ can be performed efficiently.

Let $\mathbb{G}_p$ and $\mathbb{G}_q$ be the unique subgroups of $\mathbb{G}$ of orders p and q, respectively. We recall that $u \mapsto u^q$ maps u into $\mathbb{G}_p$.

2.3 Boneh–Boyen signature scheme

Our approach is inspired by Chandran *et al.* [2], where the main ingredient of the construction is Boneh–Boyen signature scheme

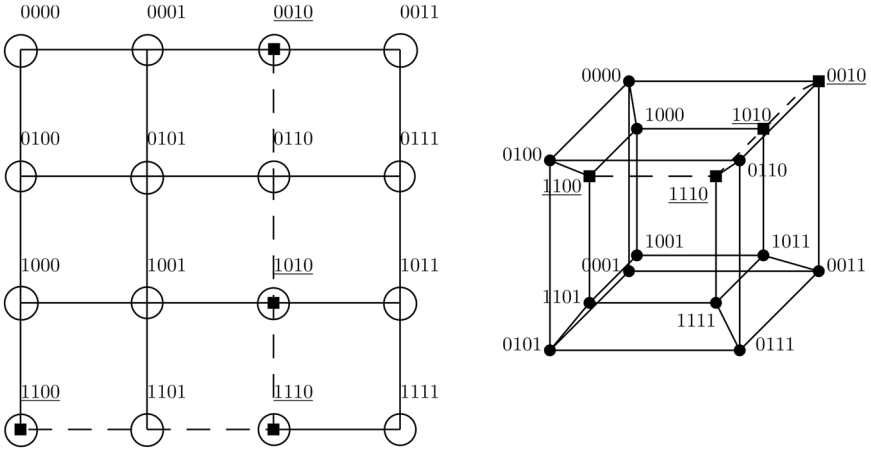

Fig. 3 *In the grid (resp. hypercube), the path is drawn from point (resp. vertex) 0010 to point (resp. vertex) 1100; N = 16. Grid's path is two edges longer than the hypercube's one.*

[15], proved existentially unforgeable under weak chosen message attack based on the strong Diffie–Hellman assumption.

As in [2], one can translate the Boneh–Boyen's scheme into one in the composite group order model such that forging a signature in $\mathbb{G}_p$ under weak chosen message attack is infeasible, based on the strong Diffie–Hellman assumption in $\mathbb{G}_p$. The Boneh–Boyen signature scheme consists of three algorithms:

- **KeyGen**: Given a tuple $(p, q, \mathbb{G}, \mathbb{G}_T, e, g)$, pick at random $sk \in_R \mathbb{Z}_n^*$ and compute $v = g^{sk}$. The key pair is (v, sk).
- **Sign**: Given a secret key $sk \in \mathbb{Z}_n^*$ and a message $M \in \{0, 1\}^l$, output the signature $\delta = g^{(1/(sk+M))}$. By convention, 1/0 is defined to be 0, thus $sk + M = 0 \Rightarrow \delta = 1$. We have $l < |p|$.
- **Verify**: Given a public key v, a message $M \in \{0, 1\}^l$ and a signature $\delta \in \mathbb{G}$, verify that $e(\delta, vg^M) = e(g, g)$. If equality holds, output 'Accept'; otherwise 'Reject'.

2.4 Commitment and encryption schemes

The commitment/encryption scheme based on the subgroup decision assumption proposed in [16] is employed in our construction. The assumption is defined in the next section.

We construct a scheme where a public key v and an element h are description of the composite order group $\mathbb{G}$. This element h is random and of order either n for perfect hiding commitment or q for encryption. It implies that perfect hiding commitment keys look exactly the same as encryption keys.

2.5 Ring signature scheme

We define a ring signature scheme as the following [2, 11].

Definition 1 (Ring signature): A ring signature comprises four probabilistic polynomial-time (PPT) algorithms as follows:
- **Gen**(1^k): on input the security parameter k, outputs a common reference string λ.
- **KeyGen**(λ) is run by the user: on input a common reference string λ, outputs a public verification key v and a private signing key sk.
- **Sign**(λ, sk, M, S): on input a message M and the ring $S = \{v_1, \ldots, v_N\}$, outputs a signature δ along with (M, S). We require that (v, sk) is a valid key pair output by **KeyGen** and that $v \in S$.
- **Verify**(λ, S, M, δ): on input a purported signature δ on a message M with respect to the ring of public keys S, outputs 'Accept' if the signature is correctly verified, otherwise 'Reject'.

Perfect Correctness: A ring signature (**Gen**, **KeyGen**, **Sign**, **Verify**) has perfect correctness if for all PPT adversary $\mathcal{A}$, the probability of

$$\lambda \leftarrow \mathbf{Gen}(1^k); (v, sk) \leftarrow \mathbf{KeyGen}(\lambda);$$
$$(M, S, \delta) \leftarrow \mathbf{Sign}(\lambda, sk, M, S) : \mathbf{Verify}(\lambda, S, M, \delta) = 1 \vee v \notin S$$

is equal to 1.

3 Security

3.1 Security properties

Intuitively, we require that a ring signature (**Gen**, **KeyGen**, **Sign**, **Verify**) has perfect anonymity if a signature on message M under ring S and key v_{i_0} is indistinguishable from a signature on message M under ring S and key v_{i_1}. The formal definition is as follows.

Definition 2 (Perfect anonymity): Given a ring signature (**Gen**, **KeyGen**, **Sign**, **Verify**), a polynomial $N(\_)$, and a PPT adversary $\mathcal{A}$, we consider the following game:

1. $\mathcal{A}$ chooses the ring of verification keys $S = \{v_1, \ldots, v_{N(k)}\}$, such that $\lambda \leftarrow \mathbf{Gen}(1^k)$ and $(v_i, sk_i) \leftarrow \mathbf{KeyGen}(\lambda)$, where $i \in \{1, \ldots, N(k)\}$.
2. $\mathcal{A}$ is given access (throughout the entire game) to an oracle **OSign**, such that **OSign**(α, M, S) returns **Sign**$(\lambda, sk_\alpha, M, S)$, where $v_\alpha \in S$.
3. $\mathcal{A}$ outputs a message M, distinct indices i_0, i_1, and a ring S for which $v_{i_0}, v_{i_1} \in S$ [i.e. (v_{i_0}, sk_{i_0}) and (v_{i_1}, sk_{i_1}) have been generated by the oracle **KeyGen**(λ)]. A random bit b is chosen, and $\mathcal{A}$ is given the signature $\delta \leftarrow \mathbf{Sign}(\lambda, sk_{i_b}, M, S)$.
4. The adversary outputs a bit b', and succeeds if $b' = b$.

A ring signature scheme achieves perfect anonymity, if for any PPT adversary $\mathcal{A}$ and any polynomial $N(\_)$, the success probability of $\mathcal{A}$ in the above game is equal to 1/2.

We also require that a ring signature (**Gen**, **KeyGen**, **Sign**, **Verify**) is unforgeable (regarding insider corruption) if it is not feasible to forge a ring signature on a message without controlling one of the members in the ring.

Definition 3 (Computational unforgeability): A ring signature (**Gen**, **KeyGen**, **Sign**, **Verify**) is computationally unforgeable if for any PPT adversary $\mathcal{A}$ and any polynomial $N(\_)$, the probability that $\mathcal{A}$ succeeds in the following game is negligible:

1. $\mathcal{A}$ is given the ring of verification keys $S = \{v_1, \ldots, v_{N(k)}\}$, such that $\lambda \leftarrow \mathbf{Gen}(1^k)$ and $(v_i, sk_i) \leftarrow \mathbf{KeyGen}(\lambda)$, where $i \in \{1, \ldots, N(k)\}$.

2. $\mathcal{A}$ is given access to a generator oracle **VKGen**, where **VKGen**(α, w_α) runs (v_α, sk_α) ← **KeyGen**(λ, w_α), such that w_α is randomly selected by the oracle, and outputs v_α.
3. $\mathcal{A}$ is given access to a signing oracle **OSign**, where **OSign**(α, M, S) outputs **Sign**(λ, sk_α, M, S), such that (v_α, sk_α) has been generated by **VKGen**.
4. $\mathcal{A}$ is given access to a corrupt oracle **Corrupt**, where **Corrupt**(α) outputs sk_α.
5. $\mathcal{A}$ outputs (S^*, M^*, δ^*), and succeeds if **Verify**(λ, S^*, M^*, δ^*) = 1. We require that $\mathcal{A}$ never queried (_, M^*, S^*), and S^* only contains verification keys v_α generated by **VKGen**, where α has not been corrupted.

3.2 Assumptions

Definition 4 (Subgroup decision assumption): Given the generator **BMGen**, we define the following distribution

$$(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathbf{BMGen}(1^k), \qquad D = (n = pq, \mathbb{G}, \mathbb{G}_T, e, g)$$

The subgroup decision assumption holds if there is a negligible function ε (in the security parameter k) so for any non-uniform polynomial time adversary $\mathcal{A}$, we have

$$\Pr[r \leftarrow \mathbb{Z}_n^*; h = g^r: \mathcal{A}(D, h) = 1] - \Pr[r \leftarrow \mathbb{Z}_q^*; h = g^{\text{pr}}: \mathcal{A}(D, h) = 1] \leq \varepsilon(k)$$

Definition 5 (Strong Diffie–Hellman assumption): Given the generator **BMGen**, we define the following distribution

$$(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathbf{BMGen}(1^k), \quad D = (p, q, \mathbb{G}, \mathbb{G}_T, e, g)$$

The strong Diffie–Hellman assumption holds in $\mathbb{G}_p$ if there is a negligible function ε (in the security parameter k) so for any non-uniform polynomial time adversary $\mathcal{A}$, we have

$$\Pr[x \leftarrow \mathbb{Z}_p^*: \mathcal{A}(D, g^q, g^{qx}, g^{qx^2}, \ldots) = (c, g^{q/(x+c)}) \in \mathbb{Z}_p \times \mathbb{G}_p] < \varepsilon(k)$$

3.3 Non-interactive zero-knowledge proof

To prove that a statement is true, we can use a non-interactive zero-knowledge (NIZK) proof which is 'complete' and 'sound', such that no interaction is needed between the prover and the verifier.

Using results in [17] providing short common reference string and NIZK proofs for any NP language, Boyen and Waters [18] gave a NIZK proof for the statement $\gamma = (g^{2M-1}h^r)^r$ verified by $e(c, g^{-1}) \stackrel{?}{=} e(h, \gamma)$. For h of order n, the proof has perfect zero knowledge as γ is determined from the verification equation and thus, no information is leaked from the proof. For h of order q, the verification enables us to show that $e(c, g^{-1})$ has order q, that implies $M = 0 \bmod p$ or $1 \bmod p$.

In [19], general methods are presented for constructing simple and efficient NIZK proofs over bilinear groups.

4 Logarithmic-size ring signature scheme

Let ring $S = \{v_1, \ldots, v_N\}$ be fixed and public. A signer knows sk_a corresponding to one of the verification keys in the ring S and wants to sign message M. The verification keys are issued as the ones in Boneh–Boyen signature scheme. The signer creates a signature as follows:

1. The signer selects one-time signature keys (vk_{OT}, sk_{OT}) ← **OTGen** (1^k). The message M is signed following the one-time signature scheme. The verification key vk_{OT} and the one-time signature are published. The signer certifies vk_{OT} by signing it with Boneh–Boyen signature under v_a.
2. The signer needs to hide v_a and the certifying signature on vk_{OT}. Therefore he/she makes two perfectly hiding commitments to v_a and the signature, respectively. Then, the signer makes a NIZK proof that the commitments contain the aforementioned elements.
3. The signer proves that the committed verification key is an element of ring S. The innovation in the scheme is the logarithmic size proof. The signer arranges S in a d-dimensional hypercube, where $N = 2^d$ (we carefully explain the process below). For $i \in \{1, \ldots, d\}$, he/she commits from the first bit b_1 to the last bit b_d of the string a of the hypercube that contains v_a and makes a NIZK proof that the committed verification key appears in this vertex $v_{b1}\ldots b_d$.

4.1 Construction

Gen generates a common reference string that contains the description of a composite order group and a public key for the perfectly hiding commitment scheme.

Gen(1^k): Let the perfectly hiding commitment scheme be as follows. Run (p, q, $\mathbb{G}$, $\mathbb{G}_T$, e, g) ← BMGen(1^k). Set $n = pq$, pick at random $x \in_R \mathbb{Z}_n^*$ and compute $h = g^x$. Output (n, $\mathbb{G}$, $\mathbb{G}_T$, e, g, h).

The users' key generation algorithm **KeyGen** takes as input a common reference string and outputs a signing public–private key pair (v, sk). In this case, it will output keys for the Boneh–Boyen signature scheme that is secure against weak message attack.

KeyGen(n, $\mathbb{G}$, $\mathbb{G}_T$, e, g, h): Let the Boneh–Boyen signature scheme with public key (g, v) be as follows. Pick at random $sk \in_R \mathbb{Z}_n^*$, and compute $v = g^{sk}$. Output (v, sk).

A user with keys (v_a, sk_a) wants to sign message M under the ring $S = \{v_1, \ldots, v_N\}$ of size N. Then, a is mapped to a d-bit binary string as follows: $a \stackrel{f}{\mapsto} b_1 b_2 \ldots b_d$ such that $f: S \to \{b_1 b_2 \ldots b_d; b_i \in \{0, 1\}, i \in \{1, \ldots, d\}\}$ is public and bijective. It is useful to think S as a d-dimensional hypercube: we assume the existence of a public map from S onto a d-dimensional hypercube that identifies each vk_a with exactly one vertex of the hypercube, labelled with a d-bit binary string. For instance, for $a \stackrel{f}{\mapsto} b_1 b_2 \cdots b_d$, v_a corresponds to the vertex defined as $b_1 b_2 \ldots b_d$. The verification key v is seen as a point in a d-dimensional space, where $d = \log_2(N)$. A ring S contains N elements v indexed by $\log_2(N)$ bits.

Informally, we construct the commitments using the following idea. From the vertex $v_{b1}b_2\ldots b_d$, there are d edges that reach d different vertices. These vertices differ from the vertex $v_{b1}b_2\ldots b_d$ in exactly one bit. For instance, from $v_{000\ldots000}$, we can reach $v_{000\ldots001}$, $v_{000\ldots010}$, …, $v_{010\ldots000}$ and $v_{100\ldots000}$. In particular, from $v_{b1}b_2\ldots b_i\ldots b_d$, we can reach the vertex $v_{b_1 b_2\ldots\bar{b}_i\ldots b_d}$ such that $\bar{b}_i = |b_i - 1| \in \{0, 1\}$. Let * denote the sequence of bits from the jth position until the dth position such that $1 \leq j \leq d$ and for $j \leq i \leq d$, the bit is equal to either b_i or $\bar{b}_i$ (we only consider strings of bit length d). Thus, we can retrieve the verification key $v_a = v_{b_1 b_2\ldots b_d}$ following the path formed by the vertices v_{b_1*}, $v_{b_1 b_2*}$, …, $v_{b_1 b_2\ldots b_{d-1}*}$ and $v_{b_1 b_2\ldots b_{d-1} b_d}$. More precisely, from the verification key v_a such that $a \stackrel{f}{\mapsto} b_1 b_2 \ldots b_d$, we can reach either $v_{\bar{b}_1 b_2\ldots b_d}$, $v_{b_1\bar{b}_2\ldots b_d}$, … or $v_{b_1 b_2\ldots\bar{b}_d}$. Therefore when we want to reach v_a, we first access the first bit b_1 of a, that is, v_{b_1*}. If we find $v_{\bar{b}_1*}$, then we know that one of the direct neighbours is v_{b_1*} that we decide to reach. We then access the second bit b_2 of a, that is, $v_{b_1 b_2*}$. We have already found v_{b_1*}, thus we may meet either $v_{b_1 b_2*}$ or $v_{b_1\bar{b}_2*}$. If we find $v_{b_1 b_2*}$, we remain there. If we find $v_{b_1\bar{b}_2*}$, then we know that one of the direct neighbours is $v_{b_1\bar{b}_2*}$ that we decide to reach. We apply the same methodology for the other bits b_3, …, b_d. Moreover, if we see the hypercube as a graph whose diameter is the smallest for a given number of points, then the resulting signature is of length of the path between two points of the graph. We illustrate the methodology to reach v for hypercubes with N equal to 4 and 8 in Fig. 4

Sign((n, $\mathbb{G}$, $\mathbb{G}_T$, e, g, h, sk_a), M, S)

- First, establish a one-time signature on the message and the ring, such that the pair (vk_{OT}, δ_{OT}) is public. Run (vk_{OT}, sk_{OT})←**OTGen** (1^k), and δ_{OT}←**Sign**(sk_{OT}, M, S). The pair (vk_{OT}, δ_{OT}) is made public.
- Pick at random $r \in_R \mathbb{Z}_n$ and compute $A = v_a h^r$, $\delta_a = g^{1/(sk_a + vk_{\text{OT}})}$. Randomly choose $s \in_R \mathbb{Z}_n$ and $B = \delta_a h^s$, $\gamma_B = g^{(r/(sk_a+vk_{\text{OT}}))+(sk_a+vk_{\text{OT}})s} h^{rs}$.

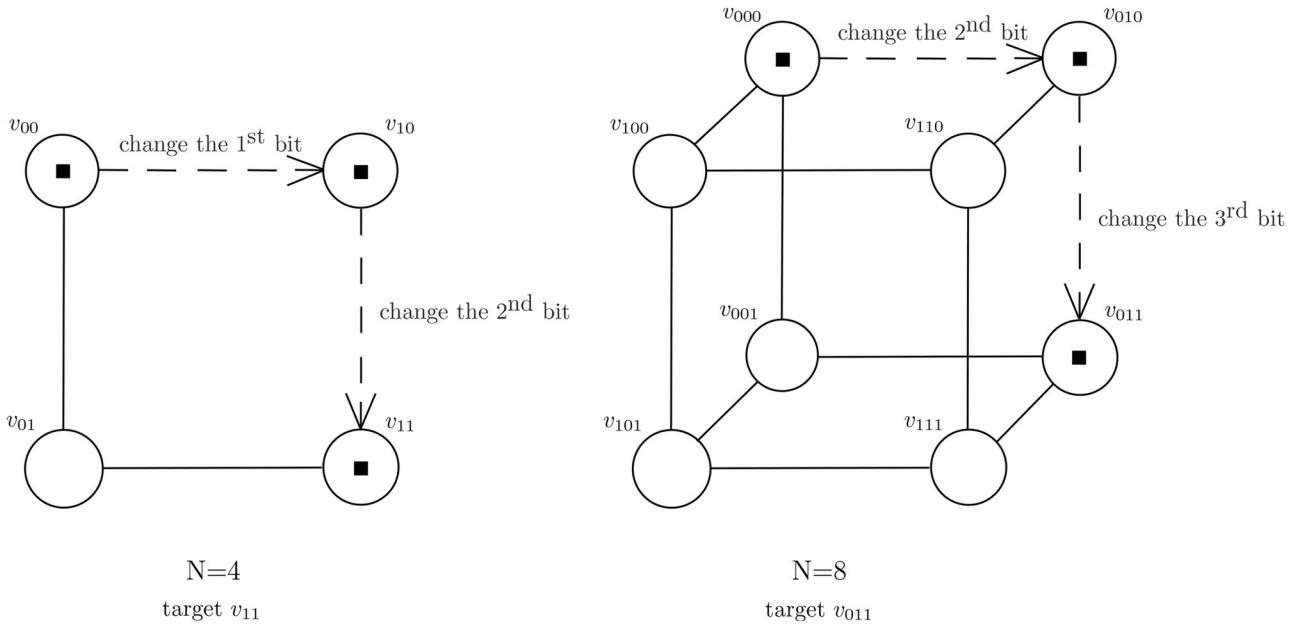

Fig. 4 *N-vertex hypercube for N = 4, 8*

Paths are shown in dashed lines to reach v_{11} and v_{011}We arbitrary start from v_{00} and $v_{000,}$ but we can start anywhere

- δ_a is the signer's certifying signature on vk_{OT}, and A and B are perfectly hiding commitments to v_a, δ_a, respectively. γ_B is a NIZK proof that A and B contain, respectively, a verification key and a signature on vk_{OT}, using results from [17].
- The rest of the protocol is a NIZK proof that A contains $v_a \in S$ without revealing which one, using results from [17, 18].
- For $a = b_1 b_2 \ldots b_d$, start the NIZK proof from the first bit b_1 of a, then the second bit b_2, and so on until the last bit b_d. Let $v_{b_i} = v_{b_1 \ldots b_{i-1} b_i *}$ and $v_{\bar{b}_i} = v_{b_1 \ldots b_{i-1} \bar{b}_i *}$, where * denotes the sequence of bits from the $i+1$th position until the dth position such that, for $i+1 \le j \le d$, the bit is equal to either b_j or $\bar{b}_j$. Randomly choose $r_{b_i} \in_R \mathbb{Z}_n$, and set $C_{b_i} = g h^{r_{b_i}}$ and $\gamma^C_{b_i} = (g h^{r_{b_i}})^{r_{b_i}}$. Set $r_{\bar{b}_i} = -r_{b_i}$, $C_{\bar{b}_i} = h^{r_{\bar{b}_i}}$ and $\gamma^C_{\bar{b}_i} = (g^{-1} h^{r_{\bar{b}_i}})^{r_{\bar{b}_i}}$.
- More precisely, for $i \in \{1, \ldots, d\}$, the commitments C_{b_i}, $C_{\bar{b}_i}$ are chosen so that C_{b_i} is a commitment to g whereas $C_{\bar{b}_i}$ is a commitment to 1, that is, $C_{b_i} C_{\bar{b}_i} = g$. The proofs γ_{b_i}, $\gamma_{\bar{b}_i}$ are NIZK proofs such that each C_{b_i}, $C_{\bar{b}_i}$ contains either g or 1. $C_{b_i} C_{\bar{b}_i} = g$ tells the verifier that there is exactly one C_{b_i} that contains g, while the other commitment contains 1. Compute $E_{b_i} = e(C_{b_i}, v_{b_i}) e(C_{\bar{b}_i}, v_{\bar{b}_i}) = e(g, v_{b_i}) \prod_{j \in \{b_j, \bar{b}_i\}} e(h^{r_j}, v_j)$, which is a commitment to $e(g, v_{b_i})$.
- Pick at random $s_{b_i}, s_{\bar{b}_i} \in_R \mathbb{Z}_n$, and compute $D_{b_i} = v_{b_i} h^{s_{b_i}}$, $D_{\bar{b}_i} = v_{\bar{b}_i} h^{s_{\bar{b}_i}}$ and $\gamma^D_{b_i} = g^{-s_{b_i}} v_{b_i}^{r_{b_i}} v_{\bar{b}_i}^{r_{\bar{b}_i 1}}$. Specifically, the D_{b_i} are commitments to verification keys $v_{b_i} \in S$ such that the ith bit of a is b_i, for $i = 1, \ldots, d$. In particular, D_{b_d} is the commitment to verification key $v_a = v_{b_d}$. The E_{b_i} contain the bit b_i of S paired with g. $\gamma^D_{b_1}, \ldots, \gamma^D_{b_d}$ are NIZK proofs that $D_{b_1}, \ldots, D_{b_d}$ contain elements that paired with g give the contents of $E_{b_1}, \ldots, E_{b_d}$. This demonstrates to the verifier that D_{b_i} contain the bit b_i in the indices of the verification keys in S.
- Compute $\gamma_A = g^{s_{b_d} - r} \prod_{j \in \{b_d, \bar{b}_d\}} v_j^{r_j} h^{s_j r_j}$ for b_d as the last bit of a. Here, $E = e(D_{b_d}, C_{b_d}) e(D_{\bar{b}_d}, C_{\bar{b}_d})$ is a commitment to $e(g, v_{b_d})$. We recall that $v_{b_d} = v_a$. γ_A is a NIZK proof that the content of A paired with g corresponds to the content in E.
- Output the signature

$$\delta = (vk_{\mathrm{OT}}, \delta_{\mathrm{OT}}, A, B, \gamma_B, \{C_{b_i}, C_{\bar{b}_i} : i \in \{1, \ldots, d\}\}, \{\gamma^C_{b_i}, \gamma^C_{\bar{b}_i} : i \in \{1, \ldots, d\}\}, \{D_{b_i}, D_{\bar{b}_i} : i \in \{1, \ldots, d\}\}, \{\gamma^D_{b_i} : i \in \{1, \ldots, d\}\}, \gamma_A)$$

Verify$((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), M, \delta)$:

1. Verify that δ_{OT} is a one-time signature of M, S under v_{OT}.
2. Verify $e(B, A g^{vk_{\mathrm{OT}}}) \stackrel{?}{=} e(g, g) e(h, \gamma_B)$.
3. Verify $e(C_{b_i}, C_{b_i} g^{-1}) \stackrel{?}{=} e(h, \gamma^C_{b_i})$ and $e(C_{\bar{b}_i}, C_{\bar{b}_i} g^{-1}) \stackrel{?}{=} e(h, \gamma^C_{\bar{b}_i})$ for all $1 \le i \le d$ and $C_{b_i} C_{\bar{b}_i} \stackrel{?}{=} g$.
4. Compute $E_{b_i} = e(C_{b_i}, v_{b_i}) e(C_{\bar{b}_i}, v_{\bar{b}_i})$ and verify $E_{b_i} \stackrel{?}{=} e(g, D_{b_i}) e(h, \gamma^D_{b_i})$ for all $1 \le i \le d$.
5. Compute $E = e(D_{b_d}, C_{b_d}) e(D_{\bar{b}_d}, C_{\bar{b}_d})$ and verify $E \stackrel{?}{=} e(A, g) e(h, \gamma_A)$.
6. If all the above steps verify correctly, then output 'Accept'; otherwise, output 'Reject'.

4.2 Security proofs

Theorem 1: The quadruple (**Gen**, **KeyGen**, **Sign**, **Verify**) is a ring signature scheme with perfect correctness, perfect anonymity and computational unforgeability under the subgroup decision assumption, the strong Diffie–Hellman assumption and given that the one-time signature is unforgeable.

Proof (Perfect correctness): For $\lambda \leftarrow$ **Gen**(1^k), for $(v, sk) \leftarrow$ **KeyGen** (λ), for any message M with respect to a ring S, we prove the perfect correctness by showing that the equalities in the algorithm **Verify** hold.

- *Point 2:* Verify the following equality $e(B, A g^{vk_{\mathrm{OT}}}) \stackrel{?}{=} e(g, g) e(h, \gamma_B)$

$$\begin{aligned} e(B, A g^{vk_{\mathrm{OT}}}) &= e(\delta_a h^s, v_a h^r g^{vk_{\mathrm{OT}}}) = e(g^{(1/(sk_a + vk_{\mathrm{OT}}))} h^s, g^{sk_a} h^r g^{vk_{\mathrm{OT}}}) \\ &= e(g^{(1/(sk_a + vk_{\mathrm{OT}}))} h, g^{sk_a + vk_{\mathrm{OT}}} h)^{rs} \\ &= e(g^{(1/(sk_a + vk_{\mathrm{OT}}))}, g^{sk_a + vk_{\mathrm{OT}}})^{rs} e(h, h)^{rs} \\ &= e(g^{(r/(sk_a + vk_{\mathrm{OT}}))}, g^{(sk_a + vk_{\mathrm{OT}})s}) e(h^{rs}, h) \\ &= e(g^{(r/(sk_a + vk_{\mathrm{OT}}))} h^{rs}, g^{(sk_a + vk_{\mathrm{OT}})s} h) e(g, g) \\ &= e(g, g) e(h, \gamma_B) \end{aligned}$$

- *Point 3:* For $i \in \{1, \ldots, d\}$

$$\begin{aligned} e(C_{b_i}, C_{b_i} g^{-1}) &= e(g h^{r_{b_i}}, g h^{r_{b_i}} g^{-1}) = e(g h^{r_{b_i}}, h^{r_{b_i}}) \\ &= e((g h^{r_{b_i}})^{r_{b_i}}, h) = e(h, \gamma^C_{b_i}) \\ e(C_{\bar{b}_i}, C_{\bar{b}_i} g^{-1}) &= e(h^{r_{\bar{b}_i}}, h^{r_{\bar{b}_i}} g^{-1}) \\ &= e(g^{-1} h^{r_{\bar{b}_i}}, h^{r_{\bar{b}_i}}) = e((g^{-1} h^{r_{\bar{b}_i}})^{r_{\bar{b}_i}}, h) = e(h, \gamma^C_{\bar{b}_i}) \end{aligned}$$

• *Point 4:* Verify the following equality $E_{b_i} \stackrel{?}{=} e(g, D_{b_i})e(h, \gamma^D_{b_i})$ for all $i \in \{1, \ldots, d\}$

$$\begin{aligned} e(g, D_{b_i})e(h, \gamma^D_{b_i}) &= e(g, v_{b_i}h^{s_{b_i}})e(h, g^{-s_{b_i}})e(h, v_{b_i}^{r_{b_i}}v_{\bar{b}_i}^{r_{\bar{b}_i}}) \\ &= e(g, v_{b_i})e(h, v_{b_i}^{r_{b_i}})e(h, v_{\bar{b}_i}^{r_{\bar{b}_i}}) \\ &= e(g, v_{b_i}) \prod_{j \in \{b_i, \bar{b}_i\}} e(h^{r_j}, v_j) = E_{b_i} \end{aligned}$$

• *Point 5:* Verify the following equality $E \stackrel{?}{=} e(A, g)e(h, \gamma_A)$

$$\begin{aligned} e(A, g)e(h, \gamma_A) &= e(v_a h^r, g)e\left(h, g^{s_{b_d}-r} \prod_{j \in \{b_d, \bar{b}_d\}} v_j^{r_j} h^{s_j r_j}\right) \\ &= e(v_a h^r, g)e(h, g^{s_{b_d}-r})e\left(h, \prod_{j \in \{b_d, \bar{b}_d\}} v_j^{r_j} h^{s_j r_j}\right) \\ &= e(v_a h^r, g)e(h, g^{s_{b_d}-r}) \prod_{j \in \{b_d, \bar{b}_d\}} e(v_j^{r_j} h^{s_j r_j}, h) \\ &= e(v_{b_d} h^{s_{b_d}}, g)e(v_{b_d} h^{s_{b_d}}, h^{r_{b_d}})e(v_{\bar{b}_d} h^{s_{\bar{b}_d}}, h^{r_{\bar{b}_d}}) \\ &= e(v_{b_d} h^{s_{b_d}}, gh^{r_{b_d}})e(v_{\bar{b}_d} h^{s_{\bar{b}_d}}, h^{r_{\bar{b}_d}}) = E \end{aligned}$$

Perfect anonymity: Following [17–19], we will prove that our scheme is secure in the anonymity game against adaptively chosen message attacks. Informally, the perfect anonymity comes from two intuitive arguments. First, for $sk_{\mathrm{OT}} \in_R \mathbb{Z}^*_n$, $vk_{\mathrm{OT}} = g^{sk_{\mathrm{OT}}}$, and for some message M, $\delta_{\mathrm{OT}} = g^{(1/(sk_{\mathrm{OT}}+M))}$, meaning that vk_{OT} and δ_{OT} are similarly generated, regardless which signing key is used. Second, all the commitments are perfectly hiding and the proofs are perfectly zero knowledge, when h has order n. In addition, an adversary can tell whether h is a random generator of $\mathbb{G}_q$ or $\mathbb{G}$ with negligible probability using a reduction proof based on the subgroup decision problem.

We assume there exist a simulator $\mathcal{B}$ that plays the subgroup decision problem with probability $\mathrm{Adv}_{\mathcal{B}}$ and an adversary $\mathcal{A}$ that wants to break the anonymity of the above ring signature scheme. In the game G_0, the simulator computes h as an element in $\mathbb{G}$ and in the game G_1, it computes h as an element in $\mathbb{G}_q$. We denote the adversary's advantage in these games as $\mathrm{Adv}_{\mathcal{A}}$ and $\mathrm{Adv}_{\mathcal{A},G_1}$, respectively.

We consider a simulator $\mathcal{B}$ receiving the subgroup decision challenge $\lambda = (n, \mathbb{G}, \mathbb{G}_T, e, g, h)$. It then creates the public parameters as in the real scheme, and sends the parameters to an adversary $\mathcal{A}$ and plays the anonymity game with it. If $h \in \mathbb{G}$, then $\mathcal{A}$ plays the normal game G_0. If $h \in \mathbb{G}_q$, then $\mathcal{A}$ plays the hybrid game G_1. We assume that $\mathcal{B}$ is able to reply all the adaptively chosen message queries, that is, it is able to issue the signing keys for any user and to sign any message by any user, since it knows the challenge λ. At some point, $\mathcal{A}$ chooses one message M and two identities i_0 and i_1 it wishes to be challenged on. We assume that the adversary had not previously made a signing key query on i_x. $\mathcal{B}$ creates a challenge signature on M, and $\mathcal{A}$ guesses the identity of the signer. If $\mathcal{A}$ answers correctly, then the simulator outputs $b = 1$, meaning that h is guessed to be in $\mathbb{G}$. Otherwise, it outputs $b = 0$, meaning that h is guessed to be in $\mathbb{G}_q$.

We denote the simulator's advantage as $\mathrm{Adv}_{\mathcal{B}}$ in the subgroup decision game. Since $\Pr[h \in \mathbb{G}] = \Pr[h \in \mathbb{G}_q] = \frac{1}{2}$, we obtain that

$$\begin{aligned} \mathrm{Adv}_{\mathcal{A}} - \mathrm{Adv}_{\mathcal{A},G_1} &= \Pr[b = 1 | h \in \mathbb{G}] - \Pr[b = 1 | h \in \mathbb{G}_q] \\ &= 2\Pr[b = 1 \wedge h \in \mathbb{G}] - 2\Pr[b = 1 \wedge h \in \mathbb{G}_q] \\ &= 2\mathrm{Adv}_{\mathcal{B}} \leq 2\varepsilon \end{aligned}$$

The result that comes from $\mathrm{Adv}_{\mathcal{B}}$ must be smaller than ε because of the hardness of the assumption.

Next, in the real scheme, when h belongs to $\mathbb{G}_q$ instead of $\mathbb{G}$, the challenge signature is statistically independent of the signer's identity in the adversary's view: we will determine what the adversary may deduce from δ.

First, we observe that $vk_{\mathrm{OT}}, \delta_{\mathrm{OT}}, A, B, \gamma_B$ do not depend on the signer identity. However, since $\mathcal{A}$ is computationally unbounded, we assume that these values reveal some information relative to the exponents. Second, we consider $C_{b_i}, C_{\bar{b}_i}$, and the corresponding $\gamma^C_{b_i}, \gamma^C_{\bar{b}_i}$ for each $i \in \{1, \ldots, d\}$. There are two hypotheses that may be formulated by $\mathcal{A}$: $b_i = 0$ or $b_i = 1$. For either hypothesis, there is a solution. Since h is a generator of $\mathbb{G}_q$, there are $\eta_{i,0}, \eta_{i,1} \in \mathbb{Z}_q$ for each $i \in \{1, \ldots, d\}$, such that $C_{b_i=1} = gh^{\eta_{i,1}} = h^{\eta_{i,0}} = C_{b_i=0}$. Thus, we obtain that $\gamma^C_{b_i=0} = (gh^{\eta_{i,1}})^{\eta_{i,1}} = (h^{\eta_{i,0}})^{\eta_{i,1}} = (h^{\eta_{i,1}})^{\eta_{i,0}} = (g^{-1}h^{\eta_{i,0}})^{\eta_{i,0}} = \gamma^C_{b_i=1}$. This means that the knowledge of $C_{b_i}, C_{\bar{b}_i}, \gamma^C_{b_i}, \gamma^C_{\bar{b}_i}$ for each $i \in \{1, \ldots, d\}$ does not reveal any information about the bit b_i, and therefore, it does not reveal the identity of the signer.

Finally, we focus on $D_{b_i}, D_{\bar{b}_i}, \gamma^D_{b_i}$ for $i \in \{1, \ldots, d\}$, and γ_A. These values are redundant in the adversary's view since the $\mathcal{A}$ already knows all the values that determine them.

Therefore the identity is statistically independent of the entire signature δ, that means $\mathrm{Adv}_{\mathcal{A},G_1} = 0$. Thus, we obtain that $\mathrm{Adv}_{\mathcal{A}} \leq 2\varepsilon$.

Computational unforgeability: Following [17–19], our scheme is proved computationally unforgeable with relation to insider corruption. Informally, under the subgroup decision assumption, the probability that the forgery happens when we switch from h of order n in a common reference string to h of order q is negligible. The commitments are now perfectly binding in $\mathbb{G}_p$ and the NIZK proofs are perfectly sound in $\mathbb{G}_p$, and therefore some uncorrupted $v_a \in S$ is contained in A and a signature δ_a on vk_{OT} under v_a is contained in B. We carefully develop this part in the proof below. Next, by the properties of the one-time signature scheme, vk_{OT} has not been used in any other signature and thus, δ_a is a forged Boneh–Boyen signature on vk_{OT}. We omit this part since the proof is quite straightforward: Boneh and Boyen [15] showed that this probability is negligible under the strong Diffie–Hellman assumption.

We assume there exists a simulator $\mathcal{B}$ that plays the subgroup decision problem with probability $\mathrm{Adv}_{\mathcal{B}}$ and an adversary $\mathcal{A}$ that wants to break the unforgeability of the above ring signature scheme. $\mathcal{B}$ receives the subgroup decision challenge $\lambda = (n = pq, \mathbb{G}, \mathbb{G}_T, e, g, h)$, where $(p, q, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathbf{BMGen}(1^k)$ and h is either equal to g^r for $r \in \mathbb{Z}_n$ or to g^{pr} for $r \in \mathbb{Z}_q$. More precisely, in the game G_0, $\mathcal{B}$ computes h as an element in $\mathbb{G}$ and in the game G_1, it computes h as an element in $\mathbb{G}_q$. $\mathcal{B}$ runs $\mathcal{A}$ with input the verification keys $S = \{v_1, \ldots, v_N\}$ that $\mathcal{B}$ generates as in the real scheme. $\mathcal{B}$ also selects a user $\tilde{a} \in \{1, \ldots, N\}$ at random. If $h \in \mathbb{G}$, then $\mathcal{A}$ plays the normal game G_0. Otherwise, if $h \in \mathbb{G}_q$, then $\mathcal{A}$ plays the hybrid game G_1.

$\mathcal{B}$ proceeds to simulate the oracle queries of $\mathcal{A}$ as follows:

• When $\mathcal{A}$ requests a signature on a message M, with respect to ring S (S might contain some verification keys generated in an arbitrary manner by $\mathcal{A}$), to be signed by user $a \neq \tilde{a}$, then $\mathcal{B}$ can easily generate the response to this query by running the **Sign** algorithm in a honest manner.

• When $\mathcal{A}$ requests a signature on message M, with respect to ring S (S might contain some verification keys generated in an arbitrary manner by $\mathcal{A}$), to be signed by user $\tilde{a}$, then $\mathcal{B}$ cannot directly respond to this query since it does not have the appropriate secret key for $\tilde{a}$ (we recall that $v_{\tilde{a}} = g^{sk_{\tilde{a}}}$ for some unknown $sk_{\tilde{a}}$). Instead, $\mathcal{B}$ submits M to its signing oracle and obtains in return a signature for $\tilde{a}$. The remainder of the signature is calculated as in the real scheme using h.

• Any corruption query made by $\mathcal{A}$ for user $a \neq \tilde{a}$ can be accurately answered by $\mathcal{B}$. However, if $\mathcal{A}$ ever makes a corruption query for $\tilde{a}$, then $\mathcal{B}$ simply aborts.

At some point, $\mathcal{A}$ outputs a forgery

$$\begin{aligned} \delta^* = (&vk_{\mathrm{OT}}, \delta_{\mathrm{OT}}, A^*, B^*, \gamma^*_B, \{C^*_{b_i}, C^*_{\bar{b}_i} : i \in \{1, \ldots, d\}\}, \\ &\{\gamma^{C*}_{b_i}, \gamma^{C*}_{\bar{b}_i} : i \in \{1, \ldots, d\}\} \\ &\{D^*_{b_i}, D^*_{\bar{b}_i} : i \in \{1, \ldots, d\}\}, \{\gamma^{D*}_{b_i} : i \in \{1, \ldots, d\}\}, \gamma^*_A) \end{aligned}$$

on a message M^* regarding some ring of honest user verification keys $S' \subseteq S$. If $v_{\tilde{a}} \notin S'$, then $\mathcal{B}$ aborts. If **Verify**$((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), M, \delta) \rightarrow$ 'Accept', then the adversary wins the game. We denote the adversary's advantage in the games G_0 and G_1 as $\mathrm{Adv}_{\mathcal{A}}$ and $\mathrm{Adv}_{\mathcal{A},G_1}$, respectively.

If $h \in \mathbb{G}$ as in the game G_0, then $\mathcal{B}$ provides a perfect simulation for the adversary $\mathcal{A}$ since the signature given to $\mathcal{B}$ is as in the real game. Otherwise (i.e. $h \in \mathbb{G}_q$ as in the game G_1), then the forgery is uniformly distributed in $\mathbb{G}_q$ and independent of the random choices made by $\mathcal{B}$. We recall that the simulator's advantage is $\mathrm{Adv}_{\mathcal{B}} \leq \varepsilon_1$ in the subgroup decision game. Since $\Pr[h \in \mathbb{G}] = \Pr[h \in \mathbb{G}_q] = \frac{1}{2}$, we obtain the following

$$\begin{aligned}
\mathrm{Adv}_{\mathcal{A}} - \mathrm{Adv}_{\mathcal{A},G_1} &= \Pr[\mathcal{A} \text{ wins the game } G_0] \\
&\quad - \Pr[\mathcal{A} \text{ wins the game } G_1] \\
&= \Pr[\mathrm{Verify}((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), \\
&\qquad M, \delta) \rightarrow \text{'Accept'} | h \in \mathbb{G}] \\
&\quad - \Pr[\mathrm{Verify}((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), \\
&\qquad M, \delta) \rightarrow \text{'Accept'} | h \in \mathbb{G}_q] \\
&= 2\Pr[\mathrm{Verify}((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), \\
&\qquad M, \delta) \rightarrow \text{'Accept'} \wedge h \in \mathbb{G}] \\
&\quad - 2\Pr[\mathrm{Verify}((n, \mathbb{G}, \mathbb{G}_T, e, g, h, S), \\
&\qquad M, \delta) \rightarrow \text{'Accept'} \wedge h \in \mathbb{G}_q] \\
&= 2\mathrm{Adv}_{\mathcal{B}} \leq 2\varepsilon_1
\end{aligned}$$

Now in the game G_1, the commitments are perfectly binding in $\mathbb{G}_p$ and the NIZK proofs are perfectly sound in $\mathbb{G}_p$. We show these results for the commitments C_{b_i}, the other commitments following a similar demonstration. We recall that $C_{b_i} = gh^{r_{b_i}}$, for $i = 1, \ldots, d$. The corresponding NIZK proof for the statement $\gamma_{b_i}^C = (gh^{r_{b_i}})^{r_{b_i}}$ is verified by checking $e(C_{b_i}, C_{b_i}g^{-1}) \stackrel{?}{=} e(h, \gamma_{b_i}^C)$. When $h \in \mathbb{G}_n$ and since $\gamma_{b_i}^C$ is uniquely determined from the verification equation, the proof has perfectly zero knowledge. When $h \in \mathbb{G}_q$, the verification shows that $e(C_{b_i}, C_{b_i}g^{-1})$ has order q. Since this happens for all the commitments and the corresponding NIZK proofs, there is a honest user a with uncorrupted signing public key $vk_a \in S$ such that $A^* = vk_a h^r$ and so there is a signer's certifying signature δ_a on vk_{OT} such that $B^* = \delta_a h^s$. In other words, if $\mathcal{A}$ outputs a valid forgery, then with all but negligible probability ε_2 by soundness of NIZK, it holds that δ^* is a valid signature of M^* regarding v_a for some a. From this, with probability $1/N$, we get that the event [$\mathcal{B}$ did not abort $\wedge \delta^*$ is a valid signature of M^* regarding $v_{\tilde{a}}$ for $\tilde{a}$] occurs. Therefore the advantage of the adversary in the game G_1 is

$$\mathrm{Adv}_{\mathcal{A},G_1} \leq N \cdot \varepsilon_2$$

Later, the forgery δ^* on M^* implies a forgery of the Boneh–Boyen signature. More precisely, A contains a verification key that is not corrupted and B contains a signature on vk_{OT} under this verification key. We recall that the probability of the event (vk_{OT} has not been used in any other signature) is negligible based on the properties of the one-time signature scheme and the strong Diffie–Hellman assumption. For simplicity, we do not count this part in our security analysis.

We conclude that the adversary succeeds with probability $\mathrm{Adv}_{\mathcal{A}} \leq \varepsilon_1 + N \cdot \varepsilon_2$. □

4.3 Working in prime order groups

We work in composite order groups in our construction. The anonymity relies on the hardness of the subgroup decision assumption. This assumption is as follows: given a group $\mathbb{G}$ of composite order $n = pq$, it is hard to decide whether a given element $g \in \mathbb{G}$ is in the subgroup of order p without knowing p and q. It has to be infeasible to factor n to achieve this hardness. This results in very large parameter sizes, for example, $\log_2 n = 3072$ or 3248 for a 128-bit security level, according to the National Institute of Standards and Technology (NIST) or the European Network of Excellence in Cryptology II (ECRYPT II) recommendations [20].

Extending our scheme in prime order groups would be an interesting challenge to gain in efficiency. In addition, the pairing computation seems to be much slower in the composite order setting than in the prime order setting. We reckon that there are useful properties for bilinear composite order models to design protocols; however, the latter is not very competitive compared with the protocols relying on other assumptions such that prime order models with asymmetric pairings.

Recently, Groth and Sahai [19] have shown that their non-interactive witness-indistinguishable (NIWI) and NIZK techniques can be realised in prime order groups under the decision linear problem. We could apply these results in our ring signature protocol to obtain a scheme in prime order groups.

5 Conclusion

In this paper, we constructed 'the first' ring signature scheme of logarithmic size in the number of users in the ring, improving the sub-linear size result obtained in [2]. Inspired by Chandran *et al.*'s work [2], our scheme requires a common reference string and the NIZK proofs and is proved secure without relying on random oracles.

6 References

1 Rivest, R., Shamir, A., Tauman, Y.: 'How to leak a secret: theory and applications of ring signatures'. In Essays in Memory of Shimon Even, 2006 (*LNCS*, **3895**), pp. 164–186
2 Chandran, N., Groth, J., Sahai, A.: 'Ring signatures of sub-linear size without random oracles'. Proc. of ICALP'07, Wrocław, Poland, 2007 (*LNCS*, **4596**), pp. 423–434
3 Naor, M.: 'Deniable ring authentication'. Proc. of CRYPTO'02, 2002 (*LNCS*, **2442**), pp. 481–498
4 Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: 'Anonymous identification in ad hoc groups'. Proc. of EUROCRYPT'04, Interlaken, Switzerland, 2004 (*LNCS*, **3027**), pp. 609–626
5 Zhang, F., Kim, K.: 'ID-based blind signature and ring signature from pairings'. Proc. of ASIACRYPT'02, Queenstown, New Zealand, 2002 (*LNCS*, **2501**), pp. 533–547
6 Au, M.H., Liu, J.K., Susilo, W., Yuen, T.H.: 'Certificate based (linkable) ring signature'. Proc. of ISPEC'07, Hong Kong, China, 2007 (*LNCS*, **4464**), pp. 79–92
7 Fujisaki, E., Suzuki, K.: 'Traceable ring signature'. Proc. of PKC'07, Beijing, China, 2007 (*LNCS*, **4450**), pp. 181–200
8 Liu, J.K., Wei, V.K., Wong, D.S.: 'Linkable and anonymous signature for ad hoc groups'. ACISP'04, 2004 (*LNCS*, **3108**), pp. 325–335
9 Wang, C.-H., Liu, C.-Y.: 'A new ring signature scheme with signer-admission property', *Inf. Sci.*, 2007, **177**, (3), pp. 747–754
10 Chow, S.S.M., Wei, V.K., Liu, J.K., Yuen, T.H.: 'Ring signatures without random oracles'. Proc. of ASIACCS'06, Taipei, Taiwan, 2006 (*CCS*), pp. 297–302
11 Bender, A., Katz, J., Morselli, R.: 'Ring signatures: stronger definitions and construction without random oracles'. *J. Cryptol.*, 2008, **22**, pp. 114–138
12 Shacham, H., Waters, B.: 'Efficient ring signatures without random oracles'. Proc. PKC'07, Beijing, China, 2007 (*LNCS*, **4450**), pp. 166–180
13 Boyen, X.: 'Mesh signatures'. Proc. of EUROCRYPT'07, Barcelona, Spain, 2007 (*LNCS*, **4515**), pp. 210–227
14 Schäge, S., Schwenk, J.: 'A CDH-based ring signature scheme with short signatures and public keys'. Proc. of FC'10, Tenerife, Spain, 2010 (*LNCS*, **6052**), pp. 129–142
15 Boneh, D., Boyen, X.: 'Short signatures without random oracles'. Proc. of EUROCRYPT'04, Interlaken, Switzerland, 2004 (*LNCS*, **3027**), pp. 56–73
16 Boneh, D., Goh, E.-J., Nissim, K.: 'Evaluating 2-DNF formulas on ciphertexts'. Proc. of TCC'05, Cambridge, MA, 2005 (*LNCS*, **3378**), pp. 325–341
17 Groth, J., Ostrovsky, R., Sahai, A.: 'Perfect non-interactive zero-knowledge for NP'. Proc. of EUROCRYPT'06, St. Petersburg, Russia, 2006 (*LNCS*, **4004**), pp. 339–358
18 Boyen, X., Waters, B.: 'Compact group signatures without random oracles'. Proc. of EUROCRYPT'06, St. Petersburg, Russia, 2006 (*LNCS*, **4004**), pp. 427–444
19 Groth, J., Sahai, A.: 'Efficient non-interactive proof systems for bilinear groups'. Proc. of EUROCRYPT'08, Istanbul, Turkey, 2008 (*LNCS*, **4965**), pp. 415–432
20 Guillevic, A.: 'Comparing the pairing efficiency over composite-order and prime-order elliptic curves'. Cryptology ePrint Archive, Report 2013/218 (2013)