

Shorter Ring Signatures from Standard Assumptions

Alonso González^{1*}

ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.
alonso.gonzalez@ens-lyon.fr

Abstract. Ring signatures, introduced by Rivest, Shamir and Tauman (ASIACRYPT 2001), allow to sign a message on behalf of a set of users while guaranteeing authenticity and anonymity. Groth and Kohlweiss (EUROCRYPT 2015) and Libert et al. (EUROCRYPT 2016) constructed schemes with signatures of size logarithmic in the number of users. An even shorter ring signature, of size independent from the number of users, was recently proposed by Malavolta and Schröder (ASIACRYPT 2017). However, all these short signatures are obtained relying on strong and controversial assumptions. Namely, the former schemes are both proven secure in the random oracle model while the later requires non-falsifiable assumptions.

The most efficient construction under mild assumptions remains the construction of Chandran et al. (ICALP 2007) with a signature of size $\Theta(\sqrt{n})$, where n is the number of users, and security is based on the Diffie-Hellman assumption in bilinear groups (the SXDH assumption in asymmetric bilinear groups).

In this work we construct an asymptotically shorter ring signature from the hardness of the Diffie-Hellman assumption in bilinear groups. Each signature comprises $\Theta(\sqrt[3]{n})$ group elements, signing a message requires computing $\Theta(\sqrt[3]{n})$ exponentiations, and verifying a signature requires $\Theta(n^{2/3})$ pairing operations. To the best of our knowledge, this is the first ring signature based on bilinear groups with $o(\sqrt{n})$ signatures and sublinear verification complexity.

1 Introduction

Ring signatures, introduced by Rivest, Shamir and Tauman, [30], allow to anonymously sign a message on behalf of a ring of users $R = \{P_1, \dots, P_n\}$, only if the signer belongs to that ring. That is, no one outside R can forge a valid signature and an honestly computed signature reveals no information about the actual signer. Unlike other similar primitives such as group signatures [10], ring signatures are not coordinated: each user generates secret/public keys on his own — i.e. no central authorities — and might sign on behalf of a ring without the approval or assistance of the other members.

*This work was funded in part by the French ANR ALAMBIC project (ANR-16-CE39-0006).

The original motivation for ring signatures was anonymous leakage of secrets. Suppose a high rank officer wants to leak some sensitive document to a journalist without revealing its identity. To do so, it signs this document using a ring signature where the ring contains all other high rank officers. The journalist is convinced that some high rank officer signed the document, but it has no clue who, while this leakage might go unnoticed for the rest of officers.

More recently, ring signatures have also found applications in the construction of confidential transactions for cryptocurrencies. In a usual (non-anonymous) transaction the user computes a signature that assesses if is allowed to spend coins. In cryptocurrencies like Monero, a user form a ring from public keys in the blockchain to issue a ring signature on the transaction. Thereby, the anonymity properties of the ring signature guarantee untraceability of the transaction and fungibility, i.e. two coins can be mutually substituted. Given the practical usefulness of ring signatures, it becomes crucial to study and improve its efficiency and security.

1.1 Related Work

The efficiency of a ring signature might be splitted into three parameters: the signature size, the time required for computing a signature, and the time required for verifying a signature. Among these metrics, the signature size has received the most attention and improvements in the size usually imply improvement in the other metrics. In terms of signature size, two of the most efficient constructions have signature size logarithmic in the size of the ring [20, 25]. Both constructions rely on the random oracle model, which is an idealization of hash functions with known theoretical inconsistencies [16]. Malavolta et al. constructed a constant size ring signature without random oracles [26] using SNARKS [14, 11, 19] as a subroutine, which are known to require controversial non-falsifiable assumptions such as the knowledge of exponent assumption [15, 28]. Unlike traditional falsifiable assumptions (e.g. DDH), is not possible to efficiently check whether the adversary effectively breaks the assumption yielding non-explicit security reductions [28]. In practice, random oracles and non-falsifiable assumptions offer great efficiency at the price of less understood security guarantees. Therefore, we believe that it is important and challenging to explore practical constructions from milder assumptions.

Using only standard assumptions like RSA, Chase and Lysyanskaya proposed a ring signature scheme whose size is independent from the number of users [9]. Their ring signature is built on top of signatures of knowledge and accumulators, following Dodis et al. [12]. The scheme description is only sketched and no proof of security is given but, for fairness (as also noted in [26]), their work is previous to the (now standard) formal definition of ring signatures of Bender et al. [4]. Anyway, signatures of knowledge are built on top of simulation sound NIZK which in turn is built from standard NIZK. The underlying statements involve multiplications modulo $\phi(N)$ and exponentiations modulo N , where N an RSA modulus. To the best of our knowledge, no efficient NIZK schemes under

standard assumptions are known for statements of this kind. Thus, the only alternative under standard assumptions seems the NIZK for circuit satisfiability of Groth, Ostrovsky and Sahai [22]. A naive implementation of this protocol would require, at least, perfectly binding bit-by-bit commitments of integers in \mathbb{Z}_N . Typically, N requires 1024 bits so this solution requires at least 1024 elements of a bilinear group. On contrast, our construction is far more efficient than that for any $n < 10^4$. Although it might be possible to avoid committing bit-by-bit, there would be still many challenges. For example, it would require a NIZK proof that $a = b^y \pmod N$, for $a, b \in \mathbb{Z}_N, y \in \mathbb{Z}_{\phi(N)}$, for which the only solution seems to be committing to y bit-by-bit (in order to use binary exponentiation) leading again to proofs of ~ 1024 group elements. Our conclusion is that is not clear how to implement Chase and Lysyanskaya’s ring signature in a practical way.

Despite Chase and Lysyanskaya’s construction, without random oracles or non-falsifiable assumptions all constructions have signatures of size linear in the size of the ring, being the sole exception the $\Theta(\sqrt{n})$ ring signature of Chandran et al. [8]. They construct a simple and elegant ring signature which at its core implements a *set-membership proof*, i.e. a proof that some committed public key belongs to the set of public keys of the ring users. Their set-membership proof is quite strong, in the sense that the verification keys may be even chosen by the adversary. Going a step forward, we will build a more efficient but weaker set-membership proof which is still useful for building ring signatures.

We note that no improvements in the signature size have been made within a decade. In fact, although two previous works claim to construct signatures of constant [7] or logarithmic [18] size, in App. D we show that one construction fails to give a correct proof of security and the other is in fact of size $\Theta(n)$. The only (non-asymptotic) improvements we are aware of are [29, 17].

1.2 Our contribution

In this work we present the first ring signature based on bilinear groups whose signature size is asymptotically smaller than Chandran et al.’s, and whose security is proven under falsifiable assumptions and without random oracles. The signature consists of $\Theta(\sqrt[3]{n})$ group elements, computing a signature requires $\Theta(\sqrt[3]{n})$ exponentiations, and verifying a signature requires $\Theta(n^{2/3})$ pairings. Our ring signature is perfectly anonymous, i.e. it completely hides the identity of the actual signer, and is computationally infeasible to forge signatures for non-members of the ring.

As a first step, we construct a $\Theta(\sqrt[3]{n})$ ring signature whose security relies on a security assumption — the permutation pairing assumption — introduced by Groth and Lu [21] in an unrelated setting: proofs of correctness of a shuffle. While the assumption is “non-standard”, in the sense that is not a “DDH like” assumption, it is a falsifiable assumption and it was proven hard in generic symmetric bilinear groups by Groth and Lu. We work on asymmetric groups (Type III groups [13]) and thus we give a natural translation of the permutation pairing assumption which we also prove secure in generic asymmetric bilinear groups.

We give a second construction which is solely based on the security of the DDH assumption in both base groups (the so called SXDH assumption). The construction is highly inspired in the first construction, but we manage to get rid of the permutation pairing assumption and further shorten the size of the signature. A comparison of our ring signatures and Chandran et al.’s is given in Table 1 which summarizes a more detailed analysis found in Appendix C.

	Chandran et al. [8]	Sect. 3.2	Sect. 4.2
CRS size $\mathbb{G}_1/\mathbb{G}_2$	4/4	4/4	4/8
Verification key size $\mathbb{G}_1/\mathbb{G}_2$	1/0	2/5	10/9
Signature size $\mathbb{G}_1/\mathbb{G}_2$	$12\sqrt{n} + 10/15\sqrt{n} + 8$	$24\sqrt[3]{n} + 36/34\sqrt[3]{n} + 24$	$18\sqrt[3]{n} + 30/34\sqrt[3]{n} + 18$
Signature generation #exps.	$37\sqrt{n} + 23$	$80\sqrt[3]{n} + 71$	$72\sqrt[3]{n} + 61$
Verification #pairings	$2n + 60\sqrt{n} + 38$	$8n^{2/3} + 162\sqrt[3]{n} + 118$	$8n^{2/3} + 122\sqrt[3]{n} + 94$
Assumption	SXDH	PPA	SXDH
Erasures	No	Yes	No

Table 1: Comparison of Chandran et al.’s ring signature and ours for a ring of size n . ‘Signature generation’ is given in number of exponentiations, ‘Verification’ is given in number of pairings, and all other rows are given in number of group elements. The security of the three schemes is proved under the unforgeability of the Boneh-Boyen signature scheme plus the corresponding assumption indicated in the row ‘Assumption’. The last row states if the key generation algorithm erases its random coins after generating the verification and secret keys.

1.3 Technical Overview

Most ring signature constructions have followed the next approach. Given a ring of users, defined by the set of their verification keys, and a message: a) sign the message, b) prove in zero-knowledge knowledge of a signature which can be verified using some committed/randomized verification key, and then c) prove in zero-knowledge that this verification key belongs to the set of public keys in the ring. The most expensive part is c) and is sometimes called a *set-membership proof*.

We observe that, when proving unforgeability, *all the verification keys forming the ring are honestly generated*. Indeed, it only makes sense to guarantee unforgeability when all the members of the ring are honest (otherwise the adversary knows at least one secret key) and thus the set-membership proof might assume that all verification keys were honestly generated. It turns out that all the schemes we are aware of, in particular Chandran et al.’s, obviate this property, meaning that their set-membership proofs work even for adversarially chosen verification keys. We ask the following natural question.

Can we construct more efficient set membership proofs (without random oracles or non-falsifiable assumptions) when verification keys are sampled from a known distribution?

We answer this question in the affirmative constructing a $\Theta(\sqrt[3]{n})$ set membership proof specially tailored to the case when the verification keys are honestly

sampled. In contrast, Chandran et al.’s proof is of size $\Theta(\sqrt{n})$ but it makes no assumption on the verification keys distribution.

Our Construction from the Permutation Pairing Assumption. Our main technical tools are two hash functions compatible with Groth-Sahai proofs.

The first function, h , is *second-preimage resistant* under a slightly different notion of collision. Given $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_m)$ randomly sampled from the domain of h , it is hard to find \mathbf{A}' such that $h(\mathbf{A}') = h(\mathbf{A})$ whenever \mathbf{A}' is not a permutation of \mathbf{A} . We give a simple instantiation of h based on the permutation pairing assumption (PPA). For simplicity, consider a symmetric bilinear group \mathbb{G} of order q and generated by \mathcal{P} (it can be extended to asymmetric bilinear groups as we show in section 2.1). This assumption states that, given $\mathbf{a}_1 = (x_1\mathcal{P}, x_1^2\mathcal{P}), \dots, \mathbf{a}_m = (x_m\mathcal{P}, x_m^2\mathcal{P})$, for $x_1, \dots, x_m \leftarrow \mathbb{Z}_q$, the only way to compute $\mathbf{a}'_1 = (y_1\mathcal{P}, y_1^2\mathcal{P}), \dots, \mathbf{a}'_m = (y_m\mathcal{P}, y_m^2\mathcal{P})$ such that $\sum_{i=1}^m \mathbf{a}'_i = \sum_{i=1}^m \mathbf{a}_i$ is to take \mathbf{A}' as a permutation of the columns of \mathbf{A} . It is straightforward to note that $h(\mathbf{A}) := \sum_{i=1}^m \mathbf{a}_i$ is second-preimage resistant “modulo permutations”, given the hardness of PPA.

Our second function, g , is collision-resistant in the traditional sense. It uses \mathbf{A} as key and returns $g_{\mathbf{A}}(vk_1, \dots, vk_m) = \sum_{i=1}^m e(\mathbf{a}_i, vk_i)$ for $vk_1, \dots, vk_m \in \mathbb{G}$. Groth and Lu conjectured that it is hard to find non-trivial $vk_1, \dots, vk_m \in \mathbb{G}$ such that $\sum_{i=1}^m e(\mathbf{a}_i, vk_i) = 0$ when each \mathbf{a}_i is of the form $(x_i\mathcal{P}, x_i^2\mathcal{P})$ and $x_i \leftarrow \mathbb{Z}_q$ [21]. They give some evidence that this assumption might be true proving its hardness in the generic bilinear group model. It follows that g is collision resistant given the hardness of the aforementioned assumption. In order to be more compatible with Groth-Sahai proofs (say, structure-preserving) we compute g ’s outputs in the base group, instead of the target group \mathbb{G}_T . To render $g_{\mathbf{A}}(\mathbf{vk}) \in \mathbb{G}$ efficiently computable we make $sk_i\mathbf{a}_i$ publicly available, where $vk_i = sk_i\mathcal{P}$, and redefine g as $g_{\mathbf{A}}(\mathbf{vk}) = \sum_i sk_i\mathbf{a}_i$. Note that the discrete logarithm in base $\mathcal{P}_T = e(\mathcal{P}, \mathcal{P})$ of g defined over \mathbb{G}_T and the discrete logarithm in base \mathcal{P} of g defined over \mathbb{G} remain the same.

Each \mathbf{a}_i will be taken from the ring member’s verification key and hence, since all these verification keys are honestly sampled, when proving unforgeability we may assume that \mathbf{A} is honestly sampled from the PPA distribution.

The Basic Construction. In our ring signature, each user possesses an “extended verification key” which contains the verification key of a Boneh-Boyen signature scheme $vk = sk\mathcal{P}$ plus \mathbf{a} and $sk\mathbf{a}$, where sk is the corresponding secret key.¹ We want to show that some commitment c opens to vk and $vk \in \{vk_1, \dots, vk_n\}$. To do so, we arrange the n elements of the ring into $n^{2/3}$ blocks of size $m = \sqrt[3]{n}$. We use the following notation: for $\{s_1, \dots, s_n\}$ define $s_{i,j} := s_{(i-1)m+j}$, where $1 \leq i \leq n^{2/3}, 1 \leq j \leq m$. Assume that $vk = vk_{\mu,\nu}$.

¹Although any signature scheme compatible with Groth-Sahai proofs suffices (e.g. structure preserving signatures), we would rather keep it simple and stick to Boneh-Boyen signature which, since the verification key is just one group element, simplifies the notation and reduces the size of the final signature.

Split $(\mathbf{a}_1, \dots, \mathbf{a}_n)$ into $\mathbf{A}_i := (\mathbf{a}_{i,1}, \dots, \mathbf{a}_{i,m})$ and (vk_1, \dots, vk_n) into $\mathbf{vk}_i = (vk_{i,1}, \dots, vk_{i,m})$, for $1 \leq i \leq n^{2/3}$, and define $H := \{h(\mathbf{A}_1), \dots, h(\mathbf{A}_{n^{2/3}})\}$ and $G := \{g_{\mathbf{A}_1}(\mathbf{vk}_1), \dots, g_{\mathbf{A}_{n^{2/3}}}(\mathbf{vk}_{n^{2/3}})\}$. We use Chandran et al.'s set-membership proof of size $\Theta(\sqrt{n})$ to prove knowledge of some $h(\mathbf{A}_\mu) \in H$. Since $|H| = n^{2/3}$, this proof is of size $\Theta(\sqrt[3]{n})$. Then we prove knowledge of \mathbf{A}' , a preimage of $h(\mathbf{A}_\mu)$ such that $\mathbf{a}'_1 = \mathbf{a}_{\mu,\nu}$. Using Groth-Sahai proofs it requires commitments to the $\sqrt[3]{n}$ columns of \mathbf{A}' plus a $\Theta(1)$ proof that $h(\mathbf{A}') = h(\mathbf{A}_\mu)$. Hence, this part of the proof adds up to $\Theta(\sqrt[3]{n})$ group elements.

We give a second set-membership proof of knowledge of some $g_{\mathbf{A}_{\mu'}}(\mathbf{vk}_{\mu'}) \in G$ such that $\mu' = \mu$ (this is straightforward to do with Chandran et al.'s set-membership proof). We commit to \mathbf{vk}' , a permutation of \mathbf{vk}_μ such that $vk'_1 = vk_{\mu,\nu}$ (and consistent with \mathbf{A}'), and we prove using Groth-Sahai proofs that $g_{\mathbf{A}_{\mu'}}(\mathbf{vk}_{\mu'}) = g_{\mathbf{A}'}(\mathbf{vk}')$. Again, this part of the proof adds $\Theta(\sqrt[3]{n})$ group elements.

The proof that $h(\mathbf{A}') = h(\mathbf{A}_\mu)$ implies that \mathbf{A}' is a permutation of \mathbf{A}_μ , which can be equivalently written as $\mathbf{A}' = \mathbf{A}_\mu \mathbf{P}$, where \mathbf{P} is some permutation matrix. Given that $e(g_{\mathbf{A}'}(\mathbf{vk}'), \mathcal{P}) = e(\mathbf{A}_\mu \mathbf{P}, \mathbf{vk}') = e(g_{\mathbf{A}_\mu}(\mathbf{Pvk}'), \mathcal{P}) = e(g_{\mathbf{A}_\mu}(\mathbf{vk}_\mu), \mathcal{P})$, the collision resistance of g implies that vk'_1, \dots, vk'_m is a permutation of $vk_{\mu,1}, \dots, vk_{\mu,m}$. We conclude that $vk_{\mu,\nu} = vk'_1$ is in the ring.

Getting rid of the permutation pairing assumption. The PPA-based ring signature has the disadvantage that the PPA is not a constant-size assumption and belongs to the class of the so called q -assumptions (such as the Strong Diffie-Hellman assumption among others). It is then desirable to have a similar construction under more standard constant-size assumptions such as the SXDH assumption.

Consider the set of binary vectors of size m and the function h defined as the hamming weight of a binary vector $h(\boldsymbol{\beta}) = \sum_{i=1}^m \beta_i$. Analogously as with the PPA, $h(\boldsymbol{\beta}) = h(\boldsymbol{\beta}')$ and $\boldsymbol{\beta}, \boldsymbol{\beta}' \in \{0, 1\}^m$ implies that $\boldsymbol{\beta}'$ is a permutation of $\boldsymbol{\beta}$. (Note that in this case $\boldsymbol{\beta}'$ is a permutation of $\boldsymbol{\beta}$ unconditionally.) We use this property of binary vectors as a replacement of the PPA. Define also $g_{\boldsymbol{\beta}}(\mathbf{vk}) := \sum_i \beta_i vk_i$. Although g is longer collision resistant, it turns out that proofs that $h(\boldsymbol{\beta}') = h(\boldsymbol{\beta})$ and $g_{\boldsymbol{\beta}'}(\mathbf{vk}') = g_{\boldsymbol{\beta}}(\mathbf{vk})$ will still allow us to prove unforgeability.²

Each possible ring member generates a single $\beta \in \{0, 1\}$ and her extended verification key contains commitments $\mathbf{a} = \text{Com}(\beta)$, $\mathbf{d} = \text{Com}(\beta vk)$, and vk . Additionally it contains π , a Groth-Sahai proof that $\beta \in \{0, 1\}$, and θ , a Groth-Sahai proof that $y = \beta vk$ where y is \mathbf{d} 's opening. Although g and h are not efficiently computable from the extended verification keys, it is possible to compute commitments to $h(\boldsymbol{\beta})$ and $g_{\boldsymbol{\beta}}(\mathbf{vk})$ using the homomorphic properties of Groth-Sahai commitments. Indeed $\text{Com}(h(\boldsymbol{\beta})) = \sum_i \mathbf{a}_i$ and $\text{Com}(g_{\boldsymbol{\beta}}(\mathbf{vk})) = \sum_i \mathbf{d}_i$. Using

²Even when the adversary only knows a commitment to $\boldsymbol{\beta}$, as it will be in our case, g is not collision resistant. For small rings, the adversary may guess $\boldsymbol{\beta}$ with non-negligible probability and solve $\sum_i \beta_i (vk_i - vk'_i) = 0$ for some non trivial \mathbf{vk}' . However, this adversary is not even aware that it has found a collision.

this fact together with the re-randomizability of Groth-Sahai proofs (see [3]) we will emulate the ring signature in the PPA setting.

Assume the signer wish to sign on behalf of the ring $R = \{vk_{1,1}, \dots, vk_{n^{2/3},m}\}$ knowing the secret key corresponding to $vk_{\mu,\nu}$. Define $\mathbf{A}_1, \dots, \mathbf{A}_{n^{2/3}}$ as in the PPA construction and let $\beta_1, \dots, \beta_{n^{2/3}}$ the respective openings. In the first part of the signature, the signer proves knowledge of some $\text{Com}(h(\beta_\mu))$ from $H = \{\text{Com}(h(\beta_1)), \dots, \text{Com}(h(\beta_{n^{2/3}}))\}$ and then commits to \mathbf{A}' , a permutation of a re-randomization of \mathbf{A}_μ such that \mathbf{a}'_1 is a re-randomization of $\mathbf{a}_{\mu,\nu}$. Then it shows with a Groth-Sahai proof that a) $\sum_i \mathbf{a}'_i - \text{Com}(h(\beta_\mu)) = \text{Com}(0)$, and b) $\beta'_1 \dots, \beta'_m \in \{0, 1\}$ re-randomizing proofs $\pi_{\mu,1}, \dots, \pi_{\mu,m}$. It follows that β' , the vector of openings of \mathbf{A}' , is a permutation of β_μ , the vector of openings of \mathbf{A}_μ .

In the second part the signer proves knowledge of some $\text{Com}(g_{\beta_\mu}(\mathbf{vk}_\mu))$ from $G = \{\text{Com}(g_{\beta_1}(\mathbf{vk}_1)), \dots, \text{Com}(g_{\beta_{n^{2/3}}}(\mathbf{vk}_{n^{2/3}}))\}$ and computes commitments $\mathbf{c}'_1, \dots, \mathbf{c}'_m$ to $vk'_1 = vk_{\mu,1}, \dots, vk'_m = vk_{\mu,m}$, respectively. In section 4.1 we show that, from $\mathbf{d}_{\mu,1}, \dots, \mathbf{d}_{\mu,m}$ and $\theta_{\mu,1}, \dots, \theta_{\mu,m}$ one can derive a proof that $\sum_i \beta'_i vk'_i = \sum_i \beta_{\mu,i} vk_{\mu,i}$, or equivalently a proof that $g_{\beta'}(\mathbf{vk}') = g_{\beta_\mu}(\mathbf{vk}_\mu)$.

Zero-knowledge of the set-membership proof implies perfect anonymity of the ring signature, and follows from the fact that all proofs are statistically independent of vk when the Groth-Sahai CRS is perfectly hiding. Soundness implies unforgeability, and follows from the following argument.

Without loss of generality, we may assume that \mathbf{vk}_μ has not repeated entries since the verifier might drop all repeated entries in R without changing the statement. Suppose an adversary wish to convince the verifier that $vk = vk'_1$ is in R while in fact $vk \notin R$. In particular, this implies that vk'_1 is different from each of $vk_{\mu,1}, \dots, vk_{\mu,m}$. By the pigeonhole principle, there must be also some $vk_{\mu,i}$ that is different from each of vk'_1, \dots, vk'_m .

Since we can guess such μ, i pair beforehand with non negligible probability $1/Q$, where Q is the maximum number of verification keys. We can jump to a game where we program $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_Q)$ such that its opening $\beta \in \{0, 1\}^Q$ is of hamming weight 1 and $\beta_{\mu,i} = 1$. By the hiding property of the commitment scheme, which is based on the SXDH assumption, the adversary notices such change in \mathbf{A} only with negligible probability. Given that β' is a permutation of β , in this game the equation $\sum_i \beta'_i vk'_i = \sum_i \beta_{\mu,i} vk_{\mu,i}$ is in fact $vk'_j = vk_{\mu,i}$, for some $1 \leq j \leq m$, and hence the adversary has 0 probability of winning.

The erasures assumption. A ring signature must tolerate the adaptive corruption of the verification keys. That is, an adversary may adaptively ask for the random coins used for generating the verification keys. In the PPA-based ring signature, this amounts to reveal x_i and x_i^2 which is incompatible with the PPA (unless one considers a much stronger interactive assumption). The only alternative seems to be assume that the key generation algorithm can erase its random coins.³

But this is not the case for the SXDH-based construction. To avoid erasures, each possible ring member samples the extended verification key with $\beta = 0$.

³We elaborate more on the erasures assumption for ring signatures in App. E.

Thereby, Every answer to a corruption query is of the form $0, sk$ plus all the random coins used to generate the extended verification key.

We can argue as before that an adversary may produce some $vk \notin R$ with roughly the same probability even if \mathbf{A} is computed from a random binary vector β of hamming weight 1 with the unique 1 in the right place. In this case we can answer all corruption queries with the exception of the unique verification key for which $\beta = 1$. But anyway, the probability that the adversary corrupts this verification key is no greater than $1/Q$ so we can safely abort if this is the case. The rest of the argument is exactly as before.

Relation to [17]. Our construction is similar to the set membership proof of González et al. [17, Appendix D.2] also of size $\Theta(\sqrt[3]{n})$. There, the CRS contains a matrix \mathbf{A} of size $2 \times m$ that is used to compute $\sqrt[3]{n}$ hashes of $n^{2/3}$ of subsets of verification keys of size $\sqrt[3]{n}$. Then some hidden hash is shown to belong to the set for $n^{2/3}$ hashes. These hashes are computed as a linear combination of the columns of \mathbf{A} with the verification keys.

One could turn this construction into a ring signature including $vk\mathbf{A}$ in each verification key. However, the fact that \mathbf{A} is fixed implies that signatures of size $\Theta(\sqrt[3]{n})$ can be obtained only when $n \leq m^3$. So, asymptotically, this is not a $\Theta(\sqrt[3]{n})$ signature. Furthermore, the verification key will be of size $\Theta(m)$. In contrast, our ring signature verification keys are of size $\Theta(1)$ and the size of the ring is unbounded.

2 Preliminaries

We write PPT as a shortcut for probabilistic polynomial time Turing machine.

Let Gen_a be some PPT which on input 1^λ , where λ is the security parameter, returns the *group key* which is the description of an asymmetric bilinear group $gk := (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_T = e(\mathcal{P}_1, \mathcal{P}_2), q)$, where $\mathbb{G}_1, \mathbb{G}_2$, and \mathbb{G}_T are groups of prime order q , the element \mathcal{P}_s is a generator of \mathbb{G}_s , and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an efficiently computable and non-degenerated bilinear map. We will use additive notation for the group operation of all groups.

Elements in \mathbb{G}_s are denoted implicitly as $[a]_s := a\mathcal{P}_s$, where $a \in \mathbb{Z}_q, s \in \{1, 2, T\}$. The pairing operation is written as a product \cdot , that is $[a]_1 \cdot [b]_2 = [a]_1[b]_2 = [b]_2[a]_1 = e([a]_1, [b]_2) = [ab]_T$. Vectors and matrices are denoted in boldface. Given a matrix $\mathbf{T} = (t_{i,j})$, $[\mathbf{T}]_s$ is the natural embedding of \mathbf{T} in \mathbb{G}_s , that is, the matrix whose (i, j) th entry is $t_{i,j}\mathcal{P}_s$. Given a matrix \mathbf{S} with the same number of rows as \mathbf{T} , we define $\mathbf{S}|\mathbf{T}$ as the concatenation of \mathbf{S} and \mathbf{T} .

2.1 Hardness Assumptions

We use a natural translation to asymmetric groups of the permutation pairing assumption introduced by Groth and Lu.

Definition 1 (Permutation Pairing Assumption [21]). Let $\mathcal{Q}_m = \overbrace{\mathcal{Q} \dots \mathcal{Q}}^{m \text{ times}}$, where concatenation of distributions is defined in the natural way and $\mathcal{Q} : \mathbf{a} = \begin{pmatrix} x \\ x^2 \end{pmatrix}$, $x \leftarrow \mathbb{Z}_q$. We say that the m -permutation pairing assumption holds relative to Gen_a if for any adversary \mathbf{A}

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}_a(1^\lambda); \mathbf{A} \leftarrow \mathcal{Q}_m; \\ ([\mathbf{Z}]_1, [\mathbf{z}]_2) \leftarrow \mathbf{A}(gk, [\mathbf{A}]_1, [\mathbf{A}]_2) : \\ \text{(i) } \sum_{i=1}^m [\mathbf{z}_i]_1 = \sum_{i=1}^m [\mathbf{a}_i]_1, \\ \text{(ii) } \forall i \in [m] \ [z_{1,i}]_1 [1]_2 = [1]_1 [z_i]_2 \text{ and } [z_{2,i}]_1 [1]_2 = [z_{1,i}]_1 [z_i]_2, \\ \text{and } \mathbf{Z} \text{ is not a permutation of the columns of } \mathbf{A} \end{array} \right],$$

where $[\mathbf{Z}] = [z_1 | \dots | z_m]_1 \in \mathbb{G}_1^{2 \times m}$, $[\mathbf{A}]_1 = [\mathbf{a}_1 | \dots | \mathbf{a}_m]_1 \in \mathbb{G}_1^{2 \times m}$, $[\mathbf{z}]_2 = [(z_1, \dots, z_m)]_2 \in \mathbb{G}_2^{1 \times m}$, is negligible in λ .

Groth and Lu proved the hardness of the PPA in generic symmetric bilinear groups [21]. In Appendix A we show that the m -PPA in generic asymmetric groups is as hard as the PPA in generic symmetric groups.

For constructing the function g in the PPA instantiation we require the assumption that is hard to find $[\mathbf{x}]_2 \in \mathbb{G}_2^m \setminus \{0\}$ such that $[\mathbf{x}^\top]_2 [\mathbf{A}^\top]_1 = 0$, where $\mathbf{A} \leftarrow \mathcal{Q}_m$. Groth and Lu proved the generic hardness of the natural translation of this assumption to symmetric groups [21]. We observe that this assumption corresponds to a kernel assumption [27], the Q_m^\top -KerMDH assumption in symmetric groups.

Definition 2 (Kernel Diffie-Hellman Assumption in \mathbb{G} [27]). Let $gk \leftarrow \text{Gen}_a(1^\lambda)$ and $\mathcal{D}_{\ell,k}$ a distribution over $\mathbb{Z}_q^{\ell \times k}$. The Kernel Diffie-Hellman assumption in \mathbb{G} ($\mathcal{D}_{\ell,k}$ -KerMDH $_{\mathbb{G}_s}$) says that every PPT Algorithm has negligible advantage in the following game: given $[\mathbf{A}]$, where $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$, find $[\mathbf{x}] \in \mathbb{G}^\ell$, $\mathbf{x} \neq \mathbf{0}$, such that $[\mathbf{x}]^\top [\mathbf{A}] = [\mathbf{0}]_T$.

Our assumption is the natural translation of the Q_m^\top -KerMDH assumption to asymmetric groups, where $[\mathbf{A}]_s$ is also given in \mathbb{G}_{3-s} . Such assumption is a weaker variant of a *split* KerMDH assumption, introduced in [17], where the adversary might find an element in $\text{Ker}(\mathbf{A})$ which is splitted between \mathbb{G}_1 and \mathbb{G}_2 .

Definition 3 (Split Kernel Diffie-Hellman Assumption [17]). Let $gk \leftarrow \text{Gen}_a(1^\lambda)$ and $\mathcal{D}_{\ell,k}$ a distribution over $\mathbb{Z}_q^{\ell \times k}$. The Split Kernel Diffie-Hellman assumption ($\mathcal{D}_{\ell,k}$ -SKerMDH) says that every PPT Algorithm has negligible advantage in the following game: given $[\mathbf{A}]_1, [\mathbf{A}]_2$, where $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}$, find $[\mathbf{x}]_1 \in \mathbb{G}_1^\ell$, $[\mathbf{y}]_2 \in \mathbb{G}_2^\ell$, $\mathbf{x} \neq \mathbf{y}$, such that $[\mathbf{x}]_1^\top [\mathbf{A}]_1 = [\mathbf{y}]_2^\top [\mathbf{A}]_2$.

Our weaker variant restricts the adversary to give solutions only in \mathbb{G}_1 (i.e. $[\mathbf{y}]_2 = 0$), while we simply refer to it as the Q_m^\top -SKerMDH. González et al. proved that, in generic asymmetric groups, the $\mathcal{D}_{\ell,k}$ -SKerMDH is as hard as the $\mathcal{D}_{\ell,k}$ -KerMDH assumption in symmetric groups, for any distribution $\mathcal{D}_{\ell,k}$ [17]. We conclude that the Q_m^\top -SKerMDH is hard in generic asymmetric groups (and of course, the weaker variant that we will be using).

Finally, we recall also the definition of the Decisional Diffie-Hellman assumption (in matrix notation).

Definition 4 (Decisional Diffie-Hellman (DDH) in \mathbb{G}_s). Let $gk \leftarrow \text{Gen}_a(1^\lambda)$ and let $\mathbf{A} := (a, 1)^\top$, $a \leftarrow \mathbb{Z}_q$. We say that the DDH assumption holds relative to Gen_a if for all PPT adversaries D

$$\text{Adv}_{\text{DDH}, \text{Gen}_s}(D) := |\Pr[D(gk, [\mathbf{A}]_s, [\mathbf{A}w]_s) = 1] - \Pr[D(gk, [\mathbf{A}]_s, [\mathbf{z}]_s) = 1]|$$

is negligible in λ , where the probability is taken over $gk \leftarrow \text{Gen}_a(1^\lambda)$, $a \leftarrow \mathbb{Z}_q$, $w \leftarrow \mathbb{Z}_q$, $[\mathbf{z}]_2 \leftarrow \mathbb{G}_s^2$, and the coin tosses of the adversary. We say that the Symmetric eXternal Diffie-Hellman (SXDH) assumption holds if the DDH assumption holds in both \mathbb{G}_1 and \mathbb{G}_2 .

2.2 Ring Signature Definition

We follow Chandran et al.'s definitions [8], which extends the original definition of Bender et al. [4] by including a CRS and perfect anonymity. We allow erasures in the key generation algorithm.

Definition 5 (Ring Signature). A ring signature scheme consists of a quadruple of PPT algorithms $(\text{CRSGen}, \text{KeyGen}, \text{Sign}, \text{Verify})$ that respectively, generate the common reference string, generate keys for a user, sign a message, and verify the signature of a message. More formally:

- $\text{CRSGen}(gk)$, where gk is the group key, outputs the common reference string ρ .
- $\text{KeyGen}(\rho)$ is run by the user. It outputs a public verification key vk and a private signing key sk .
- $\text{Sign}_{\rho, sk}(m, R)$ outputs a signature σ on the message m with respect to the ring $R = \{vk_1, \dots, vk_n\}$. We require that (vk, sk) is a valid key-pair output by KeyGen and that $vk \in R$.
- $\text{Verify}_{\rho, R}(m, \sigma)$ verifies a purported signature σ on a message m with respect to the ring of public keys R and reference string ρ . It outputs 1 if σ is a valid signature for m with respect to R and ρ , and 0 otherwise.

The quadruple $(\text{CRSGen}, \text{KeyGen}, \text{Sign}, \text{Verify})$ is a ring signature with perfect anonymity if it has perfect correctness, computational unforgeability and perfect anonymity as defined below.

Definition 6 (Perfect Correctness). We require that a user can sign any message on behalf of a ring where she is a member. A ring signature $(\text{CRSGen}, \text{KeyGen}, \text{Sign}, \text{Verify})$ has perfect correctness if for any unbounded adversary A we have:

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}(1^\lambda); \rho \leftarrow \text{CRSGen}(gk); (vk, sk) \leftarrow \text{KeyGen}(\rho); \\ (m, R) \leftarrow A(\rho, vk, sk); \sigma \leftarrow \text{Sign}_{\rho, sk}(m; R); \\ \text{Verify}_{\rho, R}(m, \sigma) = 1 \text{ or } vk \notin R \end{array} \right] = 1$$

Definition 7 (Computational Unforgeability). A ring signature scheme $(\text{CRSGen}, \text{KeyGen}, \text{Sign}, \text{Verify})$ is unforgeable if it is infeasible to forge a ring signature on a message without controlling one of the members in the ring. Formally, it is unforgeable when for all PPT adversaries A we have that

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}(1^\lambda); \rho \leftarrow \text{CRSGen}(gk); (m, R, \sigma) \leftarrow A^{\text{VKGen}, \text{Sign}, \text{Corrupt}}(\rho) : \\ \text{Verify}_{\rho, R}(m, \sigma) = 1 \end{array} \right]$$

is negligible in λ , where

- VKGen on query number i selects randomness w_i , computes $(vk_i, sk_i) := \text{KeyGen}(\rho; w_i)$ and returns vk_i .
- $\text{Sign}(i, m, R)$ returns $\sigma \leftarrow \text{Sign}_{\rho, sk_i}(m, R)$, provided (vk_i, sk_i) has been generated by VKGen and $vk_i \in R$.
- $\text{Corrupt}(i)$ returns sk_i provided (vk_i, sk_i) has been generated by VKGen . (The fact that w_i is not revealed allows the erasure of the random coins used in the generation of (vk_i, sk_i)).
- A outputs (m, R, σ) such that Sign has not been queried with $(*, m, R)$ and R only contains keys vk_i generated by VKGen where i has not been corrupted.

Definition 8 (Perfect Anonymity). A ring signature scheme $(\text{CRSGen}, \text{KeyGen}, \text{Sign}, \text{Verify})$ has perfect anonymity, if a signature on a message m under a ring R and key vk_{i_0} looks exactly the same as a signature on the message m under the ring R and key vk_{i_1} , where $vk_{i_0}, vk_{i_1} \in R$. This means that the signer's key is hidden among all the honestly generated keys in the ring. Formally, we require that for any unbounded adversary A :

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}(1^\lambda); \rho \leftarrow \text{CRSGen}(gk); \\ (m, i_0, i_1, R) \leftarrow A^{\text{KeyGen}(\rho)}(\rho); \sigma \leftarrow \text{Sign}_{\rho, sk_{i_0}}(m, R) : \\ A(\sigma) = 1 \end{array} \right] =$$

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}(1^\lambda); \rho \leftarrow \text{CRSGen}(gk); \\ (m, i_0, i_1, R) \leftarrow A^{\text{KeyGen}(\rho)}(\rho); \sigma \leftarrow \text{Sign}_{\rho, sk_{i_1}}(m, R) : \\ A(\sigma) = 1 \end{array} \right]$$

where A chooses i_0, i_1 such that $(vk_{i_0}, sk_{i_0}), (vk_{i_1}, sk_{i_1})$ have been generated by the oracle $\text{KeyGen}(\rho)$.

2.3 Groth-Sahai Proofs in the SXDH Instantiation

The Groth Sahai (GS) proof system is a non-interactive witness indistinguishable proof system (and in some cases also zero-knowledge) for the language of quadratic equations over a bilinear group. The admissible equation types must be in the following form:

$$\sum_{j=1}^{m_y} f(\alpha_j, y_j) + \sum_{i=1}^{m_x} f(x_i, \beta_i) + \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} f(x_i, \gamma_{i,j} y_j) = t, \quad (1)$$

where $\alpha \in A_1^{m_y}$, $\beta \in A_2^{m_x}$, $\Gamma = (\gamma_{i,j}) \in \mathbb{Z}_q^{m_x \times m_y}$, $t \in A_T$, and $A_1, A_2, A_T \in \{\mathbb{Z}_q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$ are equipped with some bilinear map $f : A_1 \times A_2 \rightarrow A_T$.

The GS proof system is a *commit-and-prove* proof system, that is, the prover first commits to solutions of equation (1) using the GS commitments, and then computes a proof that the committed values satisfies equation (1).

GS proofs are perfectly sound when the CRS is sampled from the perfectly binding distribution, and perfectly witness-indistinguishable when sampled from the perfectly hiding distribution. Computational indistinguishability of both distributions implies either perfect soundness and computational witness indistinguishability or computational soundness and perfect witness-indistinguishability.

Further, Belenky et al. noted that Groth-Sahai proofs can be *re-randomized* [3]. This means that, given commitments and proofs showing the satisfiability of some equation, one can compute new proofs which look exactly as fresh proofs (i.e. computed with fresh randomness) for the same equation, even without knowing the commitment openings nor the randomness. In this work we compute such proofs for integer equations $\beta(\beta - 1) = 0$ and $\beta x = y$. For completeness, in App. B we show how to construct and re-randomize such proofs.

2.4 Groth-Sahai Commitments.

Following Groth and Sahai's work [23], in asymmetric groups and using the SXDH assumption, GS commitments are vectors in \mathbb{G}_γ^2 , $\gamma \in \{1, 2\}$, the form

$$\begin{aligned} \text{GS.Com}_{ck_\gamma}([x]_\gamma; \mathbf{r}) &:= \begin{pmatrix} [0]_\gamma \\ [x]_\gamma \end{pmatrix} + r_\gamma \left[\mathbf{u}_1 - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]_\gamma + r_2 [\mathbf{u}_2]_\gamma \\ \text{GS.Com}_{ck_\gamma}(x; \mathbf{r}) &:= x[\mathbf{u}_1]_\gamma + r[\mathbf{u}_2]_\gamma \end{aligned}$$

where $ck_\gamma := [\mathbf{u}_1 | \mathbf{u}_2]_\gamma$, and \mathbf{u}_2 are sampled from the same distribution as \mathbf{A} , the matrix from definition 4. The GS reference string is formed by the commitment keys ck_1, ck_2 and $\mathbf{u}_1 := w\mathbf{u}_2 + \mathbf{e}_2$ in the perfectly binding setting, and $\mathbf{u}_1 := w\mathbf{u}_2$ in the perfectly hiding setting, for $w \leftarrow \mathbb{Z}_q$.

We define commitments to row vectors as the horizontal concatenation of commitments to each of the coordinates. That is, for $\mathbf{x} \in \mathbb{Z}_q^m$ and $\mathbf{r} \in \mathbb{Z}_q^m$

$$\text{GS.Com}_{ck_\gamma}(\mathbf{x}^\top; \mathbf{r}^\top) := [\mathbf{u}_1]_\gamma \mathbf{x}^\top + [\mathbf{u}_2]_\gamma \mathbf{r}^\top \in \mathbb{G}_\gamma^{2 \times m}.$$

Given a Groth-Sahai commitment $[c]_\gamma$, we will say that $[c']_\gamma$ is a re-randomization of $[c]_\gamma$ if $[c']_\gamma = [c]_\gamma + \text{GS.Com}_{ck_s}(0; \delta)$, for $\delta \leftarrow \mathbb{Z}_q$.

2.5 Boneh-Boyen Signatures

Boneh and Boyen introduced a short signature — each signature consists of only one group element — which is secure against existential forgery under weak chosen message attacks without random oracles [5]. The verification of the validity of any signature-message pair can be written as a set of pairing product equations. Thereby, using Groth-Sahai proofs one can show the possession of a valid signature without revealing the actual signature.

We construct our ring signature using Boneh-Boyen signatures, but we could replace the Boneh-Boyen signature scheme with any structure preserving signature scheme secure under milder assumptions (e.g. [24]). We rather keep it simple and stick to Boneh-Boyen signature which, since the verification key is just one group element, simplifies the notation and reduces the size of the final signature.

Definition 9 (weak Existential Unforgeability (wUF-CMA)). We say that a signature scheme $\Sigma = (\text{KGen}, \text{Sign}, \text{Ver})$ is wUF-CMA if for any PPT adversary A

$$\Pr \left[\begin{array}{l} gk \leftarrow \text{Gen}_\alpha(1^\lambda), (m_1, \dots, m_{q_{\text{sig}}}) \leftarrow A(gk), (sk, vk) \leftarrow \text{KGen}(1^\lambda), \\ (m, \sigma) \leftarrow A(\text{Sign}_{sk}(m_1), \dots, \text{Sign}_{sk}(m_{q_{\text{sig}}})) : \\ \text{Ver}_{vk}(m, \sigma) = 1 \text{ and } m \notin \{m_1, \dots, m_{q_{\text{sig}}}\} \end{array} \right]$$

is negligible in λ .

The Boneh-Boyen signature described below is wUF-CMA under the *m-strong Diffie-Hellman* assumption.

BB.KeyGen: Given a group key gk , pick $x \leftarrow \mathbb{Z}_q$. The secret/public key pair is defined as $(sk, vk) := (x, [x]_{3-s})$.

BB.Sign: Given a secret key $sk \in \mathbb{Z}_q$ and a message $m \in \mathbb{Z}_q$, output the signature $[\sigma]_s := \left[\frac{1}{x+m} \right]_s$. In the unlikely case that $x + m = 0$ we let $[\sigma]_s := [0]_s$.

BB.Ver: On input the verification key $[vk]_{3-s}$, a message $m \in \mathbb{Z}_q$, and a signature $[\sigma]_s$, verify that $[m + x]_{3-s}[\sigma]_s = [1]_T$.

It is direct to prove knowledge of a Boneh-Boyen signature for some message m under some committed verification key with a Groth-Sahai proof for the verification equation. In our SXDH based ring signature we need to prove a slightly different statement. Since we have a commitment to the secret key $[c]_2 = \text{Com}_{ck_2}(x; s) = x[\mathbf{w}_1]_2 + s[\mathbf{w}_2]_2$ we need to show that

$$e([\sigma]_1, m[\mathbf{w}_1]_2 + [c]_2) - [\mathbf{w}_1]_T = e([\tilde{s}]_1, [w_2]_2), \quad (2)$$

for some $[\tilde{s}]_1 \in \mathbb{G}_1$.

2.6 Chandran et al.'s Set-Membership Proof

The core of Chandran et al.'s ring signature is a set-membership proof of size $\Theta(\sqrt{n})$ for a set $S \subset \mathbb{G}_\gamma$, $\gamma \in \{1, 2\}$, of size n . Assume that $S = \{[s_1]_\gamma, \dots, [s_n]_\gamma\}$. The proof arranges elements of the set in a matrix of size $m \times m$, where $m := \sqrt{n}$,

$$[\mathbf{S}]_\gamma := \begin{pmatrix} [s_{1,1}]_\gamma & \cdots & [s_{1,m}]_\gamma \\ \vdots & \ddots & \vdots \\ [s_{m,1}]_\gamma & \cdots & [s_{m,m}]_\gamma \end{pmatrix} \text{ where } s_{i,j} := s_{(i-1)m+j} \text{ for } 1 \leq i, j \leq m.$$

Let $[s_\alpha]_\gamma$ the element for which the prover wants to show that $[s_\alpha]_\gamma \in S$ and let i_α, j_α such that $s_\alpha = s_{i_\alpha, j_\alpha}$. The prover selects the j_α th column of $[\mathbf{S}]_\gamma$ and then the i_α th element of that column. To do so, the prover commits to

1. $b_1, \dots, b_m \in \{0, 1\}$ such that $b_j = 1$ iff $j = j_\alpha$,
2. $b'_1, \dots, b'_m \in \{0, 1\}$ such that $b'_i = 1$ iff $i = i_\alpha$,
3. $[\kappa_1]_\gamma := [s_{1,j_\alpha}]_\gamma, \dots, [\kappa_m]_\gamma := [s_{m,j_\alpha}]_\gamma$.

Using Groth-Sahai proofs, the prover proves that

- i. $b_1(b_1 - 1) = 0, \dots, b_m(b_m - 1) = 0, b'_1(b'_m - 1) = 0, \dots, b'_m(b'_m - 1) = 0$,
- ii. $\sum_{i=1}^m b_i = 1$ and $\sum_{i=1}^m b'_i = 1$,
- iii. $[\kappa_1]_\gamma = \sum_{j=1}^m b_j [s_{1,j}]_\gamma, \dots, [\kappa_m]_\gamma = \sum_{j=1}^m b_j [s_{m,j}]_\gamma$,
- iv. $[s_\alpha]_\gamma = \sum_{i=1}^m b'_i [\kappa_i]_\gamma$.

Equations i and ii prove that (b_1, \dots, b_m) and (b'_1, \dots, b'_m) are unitary vectors, equation iii proves that $([\kappa_1]_\gamma, \dots, [\kappa_m]_\gamma)^\top$ is a column of $[\mathbf{S}]_\gamma$, and equation iv proves that $[s_\alpha]_\gamma$ is an element of $([\kappa_1]_\gamma, \dots, [\kappa_m]_\gamma)$.

In our SXDH based ring signature we need this set-membership to show that some vector $[\mathbf{s}]_\gamma$ is the re-randomization of one of the elements of the set of commitments $S = \{[\mathbf{s}]_\gamma, \dots, [\mathbf{s}_n]_\gamma\} \subseteq \mathbb{G}_\gamma^2$. That is, there exists some $\delta \in \mathbb{Z}_q$ such that $[\mathbf{s}]_\gamma - \text{GS.Com}_{ck_\gamma}(0; \delta) \in S$. The proof remains the same but now the prover computes re-randomizations

$$3'. [\kappa_1]_\gamma := [s_{1,j_\alpha}]_\gamma + \text{GS.Com}_{ck_\gamma}(0; \delta_1), \dots, [\kappa_m]_\gamma := [s_{m,j_\alpha}]_\gamma + \text{GS.Com}_{ck_\gamma}(0; \delta_m),$$

and Groth-Sahai proofs that

- iii'. $[\kappa_1]_\gamma - \sum_{j=1}^m b_j [s_{1,j}]_\gamma = \text{GS.Com}_{ck_\gamma}(0; \delta_1), \dots, [\kappa_m]_\gamma - \sum_{j=1}^m b_j [s_{m,j}]_\gamma = \text{GS.Com}_{ck_\gamma}(0; \delta_m)$,
- iv'. $[\mathbf{s}]_\gamma - \sum_{i=1}^m b'_i [\kappa_i]_\gamma = \text{GS.Com}_{ck_\gamma}(0; \delta - \delta_{i_\alpha})$.

2.7 Hash Functions

We recall the definition of a hash function plus a weaker notion where the adversary needs to find a second preimage (see [31]). We consider a function $h : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ and an algorithm KGen which on input a group key randomly samples an element from \mathcal{K} .

Definition 10 (Collision Resistance). *We say that h is a hash-function family with collision resistance if for all PPT adversary \mathbf{A}*

$$\text{Adv}_h^{\text{Col}}(\mathbf{A}) := \Pr[k \leftarrow \text{KGen}(1^\lambda), (x, x') \leftarrow \mathbf{A}(k) : x \neq x' \text{ and } h_k(x) = h_k(x')]$$

is negligible in λ .

We use a weaker variant of collision resistance for our hash function based on the PPA assumption.

Definition 11 (Second-Preimage Resistance). *We say that h is a hash-function family with always second-preimage resistance if for all PPT adversary \mathbf{A}*

$$\text{Adv}_h^{\text{Sec}}(\mathbf{A}) := \Pr \left[k \leftarrow \text{KGen}(gk), x \leftarrow \mathcal{M}, x' \leftarrow \mathbf{A}(k, x) : x \neq x' \text{ and } h_k(x) = h_k(x') \right]$$

is negligible in λ .

3 Our Construction in the PPA setting

The high level description of our PPA based ring signature was already given in sect. 1.3. Next we proceed to formally define the hash functions h and g and then we give the formal description and security proof of the protocol.

3.1 The hash functions h and g

We instantiate definition 10 with the function g and 11 with h defined as follows. For h , $\mathcal{M} = Q_m$, $\mathcal{Y} = \mathbb{G}_1^2$, $\text{KGen} = \text{Gen}_a$, and

$$h(A) := \sum_{([\mathbf{a}]_1, [\mathbf{a}]_2) \in A} [\mathbf{a}]_1, \text{ where}$$

$$Q_m := \{\mathbf{A} \in \mathbb{Z}_q^{2 \times m} : A = (\mathbf{a}_1 | \dots | \mathbf{a}_m) \text{ and } \mathbf{a}_i = (a_{i,1}, a_{i,2})^\top \text{ s.t. } a_{i,2} = a_{i,1}^2\} \text{ and}$$

$$Q_m = \{A : \exists \mathbf{A} \in Q_m \text{ s.t. } A' = \cup_{i=1}^m ([\mathbf{a}_i]_1, [\mathbf{a}_i]_2)\}.$$

It might seem odd to define Q_m as sets of vectors in both groups while h only require elements in one group. However, this will be crucial in the security proof of our ring signature, where we need to compute $[vk\mathbf{a}]_2$, for some $vk \in \mathbb{Z}_q$, without knowledge of \mathbf{a} . For simplicity, we may just write $h(A)$ for $A \subseteq \mathbb{G}_1^2$ (which is still well defined).

Given a second preimage h , it is trivial to construct an adversary breaking the m -PPA assumption. Indeed, Let $[\mathbf{A}]_1, [\mathbf{A}]_2$ the challenge of the m -PPA assumption and let A the set of columns of $[\mathbf{A}]_1$ and $[\mathbf{A}]_2$, which is clearly uniformly distributed in Q_m . Then given any $A' \in Q_m$ such that $A' \neq A$ and $h(A) = h(A')$, it holds that $[\mathbf{A}']_1$, the matrix whose columns are the first components of the elements of A' , is not a permutation of $[\mathbf{A}]_1$ and hence breaks m -PPA assumption. Then for any adversary \mathbf{A} there is an adversary \mathbf{B} such that $\text{Adv}^{\text{aPre}_g}(\mathbf{A}) = \text{Adv}_{m\text{-PPA}}(\mathbf{B})$.

In the case of g , $\mathcal{M} = \mathbb{G}_2^m$, $\mathcal{Y} = \mathbb{G}_2^2$, and $\text{KGen}_{\text{global}}$ picks a group description $gk \leftarrow \text{Gen}_a(1^\lambda)$, while $\text{KGen}_{\text{local}}$ picks $[\mathbf{a}]_1 \in \mathbb{G}_1^{2 \times m}$, where $\mathbf{a} \leftarrow Q_1$, and the function is defined as

$$g_{[\mathbf{A}]_1}([\mathbf{x}]_2) := [\mathbf{A}\mathbf{x}]_2.$$

Although not efficiently computable, one can efficiently check if $g_{[\mathbf{A}]_1}([\mathbf{x}]_2) = g_{[\mathbf{A}]_1}([\mathbf{x}']_2)$ using the pairing operation. Further, in our scheme we will publish values of the form $[\mathbf{a}_i x_i]_2$ which will render g efficiently computable.

Given a collision $[\mathbf{x}]_2, [\mathbf{x}']_2$ for g , then $([\mathbf{x}]_2 - [\mathbf{x}']_2) \neq [\mathbf{0}]$ is in the kernel of $[\mathbf{A}]_1$. Therefore, is trivial to prove that for any adversary \mathbf{A} against static collision resistance there is an adversary \mathbf{B} such that $\text{Adv}^{\text{Col}_g}(\mathbf{A}) = \text{Adv}_{Q_m^\top\text{-SKerMDH}}(\mathbf{B})$, whenever $\mathbf{A} \leftarrow Q_m$.

We note that given $A \in Q_m$, $[\mathbf{A}]_1 \in \mathbb{G}_1^{2 \times m}$, $[\mathbf{x}]_2 \in \mathbb{G}_2^m$, $[\mathbf{y}]_1 \in \mathbb{G}_2^2$ and $[\mathbf{y}']_1 \in \mathbb{G}_2^1$ one can express the statements $A \in Q_m$, $g_{[\mathbf{A}]_1}([\mathbf{x}]_2) = [\mathbf{y}]_2$, and $h(A) = [\mathbf{y}']_1$

as (3),(4), and (5), respectively.

$$\begin{aligned} e([a_1]_1, [1]_2) &= e([1]_1, [b_1]_2) \text{ and} \\ e([a_2]_1, [1]_2) &= e([a_1]_1, [b_1]_2) \text{ for each } ([\mathbf{a}]_1, [\mathbf{b}]_2) \in A \end{aligned} \quad (3)$$

$$\sum_{j=1}^m e([a_{i,j}]_1, [x_i]_1) = e([1]_1, [y_i]_2) \text{ for each } i \in \{1, 2\} \quad (4)$$

$$\sum_{([\mathbf{a}]_1, [\mathbf{a}]_2) \in A} [a_i]_1 = [y'_i]_1 \text{ for each } i \in \{1, 2\}. \quad (5)$$

Hence, one can compute Groth-Sahai proofs of size $\Theta(m), \Theta(1)$, and $\Theta(1)$, respectively, for the satisfiability of each statement.

Finally, we prove a simple lemma that relates both functions

Lemma 1. *Let $A \leftarrow Q_m, A' \in Q_m, [\mathbf{x}]_2, [\mathbf{x}']_2 \in \mathbb{G}_2^m$, and $[\mathbf{A}]_1, [\mathbf{A}']_1$ the matrices whose columns are the first component of the elements of A and A' , respectively. Then $h(A) = h(A')$ and $g_{[\mathbf{A}]_1}([\mathbf{x}]_2) = g_{[\mathbf{A}']_1}([\mathbf{x}']_2)$ implies that A' is a second preimage of $h(A)$ or there exists a permutation matrix \mathbf{P} such that $g_{[\mathbf{A}]_1}([\mathbf{x}]_2) = g_{[\mathbf{A}]_1}([\mathbf{P}\mathbf{x}']_2)$.*

Proof. If $A \neq A'$, then A' is a second preimage of $h(A)$. Else, there is a permutation matrix \mathbf{P} such that $[\mathbf{A}']_1 = [\mathbf{A}\mathbf{P}]_1$. Then

$$g_{[\mathbf{A}']_1}([\mathbf{x}']_2) = g_{[\mathbf{A}']_1}([\mathbf{x}']_2) \iff g_{[\mathbf{A}]_1}([\mathbf{x}]_2) = g_{[\mathbf{A}\mathbf{P}]_1}([\mathbf{x}']_2) = g_{[\mathbf{A}]_1}([\mathbf{P}\mathbf{x}']_2).$$

3.2 Our Ring Signature

In the following let $n := |R|, m := \sqrt[3]{n}$, and for $1 \leq \alpha \leq n$ define $1 \leq \mu \leq n^{2/3}$ and $1 \leq \nu \leq m$ such that $\alpha = (\mu - 1)m + \nu$. For a sequence $\{s\}_{1 \leq i \leq n}$ we define $s_{\mu, \nu} := s_{(\mu-1)m+\nu}$. Consider $\text{OT} = (\text{OT.KeyGen}, \text{OT.Sign}, \text{OT.Ver})$ a one-time signature scheme.

CRSGen(gk): Pick a perfectly hiding CRS for the Groth-Sahai proof system crs_{GS} and define $(ck_1, ck_2) := \text{crs}_{\text{GS}}$. Note that crs_{GS} can be also used for the $\Theta(\sqrt{n})$ set-membership of Chandran et al. The CRS is $\rho := (gk, \text{crs}_{\text{GS}})$.

KeyGen(ρ): Pick $\mathbf{a} \leftarrow \mathcal{Q}$ and $(sk, [vk]_2) \leftarrow \text{BB.KeyGen}(gk)$, compute $[\mathbf{a}]_1, [\mathbf{a}]_2$ and then erase \mathbf{a} (but if not erased we prove security under the (ℓ, m) -PPA). The secret key is sk and the extended verification key is $vk := ([vk]_2, [\mathbf{a}]_1, [\mathbf{a}]_2, \mathbf{a}[vk]_2)$.

Sign $_{\rho, sk}(m, R)$: Let α the index of the signer with respect to R .

1. Compute $(sk_{\text{ot}}, vk_{\text{ot}}) \leftarrow \text{OT.KeyGen}(gk)$ and $\sigma_{\text{ot}} \leftarrow \text{OT.Sign}_{sk_{\text{ot}}}(m, R)$.
2. Compute $[c]_2 := \text{GS.Com}_{ck_2}([vk_{\alpha}]_2; \mathbf{r}), \mathbf{r} \leftarrow \mathbb{Z}_q^2, [\sigma]_1 \leftarrow \text{BB.Sign}_{sk_{\alpha}}(vk_{\text{ot}}), [\mathbf{d}]_1 := \text{GS.Com}_{ck_1}([\sigma]_1; \mathbf{s}), \mathbf{s} \leftarrow \mathbb{Z}_q^2$, and a GS proof π_{BB} that $\text{BB.Ver}_{[vk]_2}([\sigma]_1, vk_{\text{ot}}) = 1$.
3. For $1 \leq i \leq n^{2/3}$, let $[\kappa_i]_2 = ([vk_{i,1}]_2, \dots, [vk_{i,m}]_2)^\top, A_i = \{([\mathbf{a}_{i,1}]_1, [\mathbf{a}_{i,1}]_2), \dots, ([\mathbf{a}_{i,m}]_1, [\mathbf{a}_{i,m}]_2)\}$, and $[\mathbf{A}_i]_1 := [\mathbf{a}_{i,1}] \cdots [\mathbf{a}_{i,m}]_1$. Define the sets $H = \{h(A_1), \dots, h(A_{n^{2/3}})\}$ and $G = \{g_{[\mathbf{A}]_1}([\kappa_1]_2) \cdots, g_{[\mathbf{A}_{n^{2/3}}]_1}([\kappa_{n^{2/3}}]_2)\}$.

4. Let $[\mathbf{x}]_1 := h(A_\mu)$ and $[\mathbf{y}]_2 = g_{[\mathbf{A}_\mu]_1}([\kappa_\mu]_2)$. Compute GS commitments to $[\mathbf{x}]_1$ and $[\mathbf{y}]_2$ and compute proofs π_G and π_H that they belong to G and H , respectively. It is also proven that they appear in the same positions reusing the commitments to b_1, \dots, b_m and b'_1, \dots, b'_m , used in the set-membership proof of Chandran et al., which define $[\mathbf{x}]_1$'s and $[\mathbf{y}]_2$'s position in H and G respectively.
 5. Let $[\kappa']_2 := ([vk_\alpha]_2, [vk_{\mu,1}]_2, \dots, [vk_{\alpha-1}]_2, [vk_{\alpha+1}]_2, \dots, [vk_{\mu,m}]_2)^\top \in \mathbb{G}_2^m$, $[\mathbf{A}']_1 := [\mathbf{a}_\alpha | \mathbf{a}_{\mu,1} | \dots | \mathbf{a}_{\alpha-1} | \mathbf{a}_{\alpha+1} | \dots | \mathbf{a}_{\mu,m}]_1 \in \mathbb{G}_1^{2 \times m}$ and $A' = \{([\mathbf{a}_{\mu,1}]_1, [\mathbf{a}_{\mu,1}]_2), \dots, ([\mathbf{a}_{\mu,1}]_1, [\mathbf{a}_{\mu,1}]_2)\}$. Compute GS commitments to all but the first element of $[\kappa']_2$ (note that $[\mathbf{c}]_2$ is a commitment to the first element of $[\kappa']_2$). Compute also a GS proof π_g that $g_{[\mathbf{A}']_1}([\kappa']_2) = [\mathbf{y}]_2$, a GS proof π_h that $h(A') = [\mathbf{x}]_1$, and a GS proof π_{Q_m} that $A' \in Q_m$.
 6. Return the signature $\sigma := (vk_{\text{ot}}, \sigma_{\text{ot}}, [\mathbf{c}]_2, [\mathbf{d}]_1, \pi_{\text{BB}}, \pi_G, \pi_H, \pi_g, \pi_h, \pi_{Q_m})$. (GS proofs include commitments to variables).
- Verify $_{\rho,R}(m, \sigma)$: Verify the validity of the one-time signature and of all the proofs. Return 0 if any of these checks fails and 1 otherwise.

We prove the following theorem which states the security of our construction.

Theorem 1. *The scheme presented in this section is a ring signature scheme with perfect correctness, perfect anonymity and computational unforgeability under the Q_{gen} -permutation pairing assumption, the $Q_{Q_{\text{gen}}}^\top$ -SKerMDH assumption, the SXDH assumption, and the assumption that the one-time signature and the Boneh-Boyen signature are unforgeable. Concretely, for any PPT adversary A against the unforgeability of the scheme, there exist adversaries B_1, B_2, B_3, B_4, B_5 such that*

$$\begin{aligned} \mathbf{Adv}(A) \leq & \mathbf{Adv}_{\text{SXDH}}(B_1) + \mathbf{Adv}_{Q_{\text{gen}}\text{-PPA}}(B_2) + \mathbf{Adv}_{Q_{Q_{\text{gen}}}^\top\text{-SKerMDH}}(B_3) + \\ & Q_{\text{gen}}(Q_{\text{sign}} \mathbf{Adv}_{\text{OT}}(B_4) + \mathbf{Adv}_{\text{BB}}(B_5)), \end{aligned}$$

where Q_{gen} and Q_{sign} are, respectively, upper bounds for the number of queries that A makes to its VKGen and Sign oracles.

Proof. Perfect correctness follows directly from the definitions. Perfect anonymity follows from the fact that the perfectly hiding Groth-Sahai CRS defines perfectly hiding commitments and perfect witness-indistinguishable proofs, information theoretically hiding any information about vk .

We say that an unforgeability adversary is “eager” if makes all its queries to the VKGen oracle at the beginning. Note that any non-eager adversary A' can be perfectly simulated by an eager adversary that makes Q_{gen} queries to VKGen and answers A' queries to VKGen “on demand”. This is justified by the fact that the output of VKGen is independent of all previous outputs.

W.l.o.g. we assume that A is an eager adversary. Computational unforgeability follows from the indistinguishability of the following games

Game₀: This is the real unforgeability experiment. **Game₀** returns 1 if the adversary A produces a valid forgery and 0 if not.

Game₁: This is game exactly as **Game₀** with the following differences:

- The Groth-Sahai CRS is sampled together with its discrete logarithms from the perfectly binding distribution. Note that the discrete logarithms of the CRS allow to open the Groth-Sahai commitments.
- At the beginning, variables err_2 and err_3 are initialized to 0 and a random index i^* is chosen from $\{1, \dots, Q_{\text{gen}}\}$.
- On a query to **Corrupt** with argument i , if $i = i^*$ set $\text{err}_3 \leftarrow 1$ and proceed as in **Game₀**.
- Let (m, R, σ) the purported forgery output by **A**. If $[vk]_2$, the opening of commitment $[c_{\mu, \nu}]_2$ from σ , is not equal to $[vk_{i^*}]_2$, set $\text{err}_3 \leftarrow 1$. If $[vk]_2 \notin R$, then set $\text{err}_2 = 1$.

Game₂: This is game exactly as **Game₁** except that, if err_2 is set to 1, **Game₂** aborts.

Game₃: This is game exactly as **Game₂** except that, if err_3 is set to 1, **Game₃** aborts.

Since in **Game₁** variables err_2 and err_3 are just dummy variables, the only difference with **Game₀** comes from the Groth-Sahai CRS distribution. It follows that there is an adversary \mathbf{B}_1 against SXDH such that $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_1 = 1]| \leq \mathbf{Adv}_{\text{SXDH}}(\mathbf{B}_1)$.

Lemma 2. *There exist adversaries \mathbf{B}_2 and \mathbf{B}_3 against the Q_{gen} -permutation pairing assumption and against the Q_{gen}^\top -KerMDH assumption, respectively, such that*

$$|\Pr[\text{Game}_2 = 1] - \Pr[\text{Game}_1 = 1]| \leq \mathbf{Adv}_{Q_{\text{gen}}\text{-PPA}}(\mathbf{B}_2) + \mathbf{Adv}_{Q_{\text{gen}}^\top\text{-SKerMDH}}(\mathbf{B}_3).$$

Proof. Note that

$$\begin{aligned} \Pr[\text{Game}_1 = 1] &= \Pr[\text{Game}_1 = 1 | \text{err}_2 = 0] \Pr[\text{err}_2 = 0] + \\ &\quad \Pr[\text{Game}_1 = 1 | \text{err}_2 = 1] \Pr[\text{err}_2 = 1] \\ &\leq \Pr[\text{Game}_2 = 1] + \Pr[\text{Game}_1 = 1 | \text{err}_2 = 0] \\ &\implies |\Pr[\text{Game}_2 = 1] - \Pr[\text{Game}_1 = 1]| \leq \Pr[\text{Game}_1 = 1 | \text{err}_2 = 1]. \end{aligned}$$

We proceed to bound this last probability constructing two adversaries against collision resistance of g and preimage resistance of h . Let $1 \leq \mu \leq n^{2/3}$ the index defined in π_G and π_S .

Consider an adversary \mathbf{A}_h that finds a second preimage of h when $\mathcal{M} = Q_{Q_{\text{gen}}}$. \mathbf{A}_h receives as challenge $B \in Q_{Q_{\text{gen}}}$ and honestly simulates **Game₁** with the following exception. On the i th query of **A** to **VKGen** picks $(sk, [vk]) \leftarrow \text{BB.KeyGen}(1^\lambda)$ and sets $(sk_i, \tilde{vk}_i) := (sk, ([vk]_2, [\mathbf{b}_i]_1, [\mathbf{b}_i]_2, sk[\mathbf{b}_i]_2))$, where $([\mathbf{b}_i]_1, [\mathbf{b}_i]_2)$ is the i th element of B . When **A** corrupts the i th party, it returns sk_i but it might also request \mathbf{a}_i to its oracle if we are proving security under the (ℓ, m) -PPA assumption. When **A** outputs and π_{Q_m} , \mathbf{A}_h extracts $A' = \{([\mathbf{a}'_1]_1, [\mathbf{a}'_1]_2), \dots, ([\mathbf{a}'_m]_1, [\mathbf{a}'_m]_2)\}$ and returns $A' \cup \bar{A}_\mu$, where $\bar{A}_\mu := B \setminus A_\mu$.

Consider another adversary \mathbf{A}_g against the collision resistance of g when $\mathcal{M} = \mathbb{G}^{Q_{\text{gen}}}$. **B** receives as challenge $[\mathbf{B}]_1 \in \mathbb{G}_1^{2 \times Q_{\text{gen}}}$ and $[\mathbf{B}]_2 \in \mathbb{G}_2^{2 \times Q_{\text{gen}}}$ and

honestly simulates Game_1 embedding $[\mathbf{B}]_1, [\mathbf{B}]_2$ in the user keys in the same way as A_h . When A outputs $[c]_2, \text{GS.Com}_{ck_2}([\kappa'_2]_2), \dots, \text{GS.Com}_{ck_2}([\kappa'_m]_2)$, A_g extracts $[vk], [\kappa'_2], \dots, [\kappa'_m]$. W.l.o.g. assume that $\mathbf{B} = \mathbf{A}_\mu \bar{\mathbf{A}}_\mu$, where \mathbf{A}_μ is some matrix whose rows are the discrete logs of the elements of \bar{A}_μ . A_g attempts to extract a permutation matrix \mathbf{P} such that $[\mathbf{A}']_1 = [\mathbf{A}_\mu]_1 \mathbf{P}$. If there is no such permutation matrix, then A_g aborts. Else, A_g returns $\begin{pmatrix} [\kappa_\mu]_2 \\ [\mathbf{0}]_2 \end{pmatrix}, \begin{pmatrix} \mathbf{P}[\kappa']_2 \\ [\mathbf{0}]_2 \end{pmatrix} \in \mathbb{G}_2^{Q_{\text{gen}}}$, where $[\kappa'_1]$ is the opening of $[c]$.

Perfect soundness of proof π_g (recall that the Groth-Sahai CRS is perfectly binding) implies that

$$g_{[\mathbf{A}']_1}([\kappa']_2) = [\mathbf{y}]_2.$$

Perfect soundness of proof π_g and π_{Q_m} implies that

$$h(A') = [\mathbf{x}]_1 \text{ and } A' \in Q_m.$$

Given perfect soundness of proofs π_G, π_H , it holds that that

$$\begin{aligned} g_{[\mathbf{A}']_1}([\kappa']_2) &= g_{[\mathbf{A}_\mu]_1}([\kappa_\mu]_2) \\ h(A') &= h(A_\mu). \end{aligned}$$

By Lemma 1 we get that either $A' \neq A_\mu$ is a second preimage for $h(A_\mu)$, thus $A' \cup \bar{A}_\mu \neq B$ and A_h is successful, or there exists a permutation matrix \mathbf{P} , which is the one that A_g searches, such that $g_{[\mathbf{A}_\mu]_1}(\mathbf{P}[\kappa']_2) = g_{[\mathbf{A}_\mu]_1}([\kappa_\mu]_2)$. $\text{err}_2 = 1$ implies that $[vk]_2 = [\kappa'_1]_2 \neq [\kappa_{\mu,i}]_2$, for all $1 \leq i \leq m$, and thus $\mathbf{P}[\kappa']_2 \neq [\kappa_\mu]_2$ and, since $[\mathbf{B}]_1 = [\mathbf{A}_\mu]_1 \bar{\mathbf{A}}_\mu$,

$$g_{[\mathbf{A}_\mu]_1}(\mathbf{P}[\kappa']_2) = g_{[\mathbf{B}]_1} \begin{pmatrix} \mathbf{P}[\kappa']_2 \\ [\mathbf{0}]_2 \end{pmatrix} = g_{[\mathbf{A}_\mu]_1}([\kappa_\mu]_2) = g_{[\mathbf{B}]_1} \begin{pmatrix} [\kappa_\mu]_2 \\ [\mathbf{0}]_2 \end{pmatrix}$$

and A_g is successful.

As stated in Section 2.7, from A_h we can construct an adversary B_2 that breaks the Q_{gen} -PPA assumption and from A_g we can construct an adversary B_3 that breaks the Q_m^\top -SKerMDH assumption, with the same advantages. We conclude that

$$\Pr[\text{Game}_1 = 1 | \text{err}_2 = 1] \leq \mathbf{Adv}_{Q_{\text{gen}}\text{-PPA}}(B_2) + \mathbf{Adv}_{Q_m^\top\text{-SKerMDH}}(B_3)$$

Lemma 3.

$$\Pr[\text{Game}_3 = 1] \geq \frac{1}{Q_{\text{gen}}} \Pr[\text{Game}_2 = 1].$$

Proof. It holds that

$$\begin{aligned} \Pr[\text{Game}_3 = 1] &= \Pr[\text{Game}_3 = 1 | \text{err}_3 = 0] \Pr[\text{err}_3 = 0] \\ &= \Pr[\text{Game}_2 = 1 | \text{err}_3 = 0] \Pr[\text{err}_3 = 0] \\ &= \Pr[\text{err}_3 = 0 | \text{Game}_2 = 1] \Pr[\text{Game}_2 = 1]. \end{aligned}$$

The probability that $\text{err}_3 = 0$ given $\text{Game}_2 = 1$ is the probability that the Q_{cor} calls to **Corrupt** do not abort and that $[vk]_2 = [vk_{i^*}]_2$. Since A is an eager adversary, at the i th call to **Corrupt** the index i^* is uniformly distributed over the $Q_{\text{gen}} - i + 1$ indices of uncorrupted users. Similarly, when A outputs its purported forgery, the probability that $[vk]_2 = [vk_{i^*}]_2$ is $1/(Q_{\text{gen}} - Q_{\text{cor}})$, since $[vk]_2 \in R$ (or otherwise Game_2 would have aborted). Therefore

$$\Pr[\text{err}_2 = 1 | \text{Game}_2 = 1] = \frac{Q_{\text{gen}} - 1}{Q_{\text{gen}}} \frac{Q_{\text{gen}} - 2}{Q_{\text{gen}} - 1} \cdots \frac{Q_{\text{gen}} - Q_{\text{cor}}}{Q_{\text{gen}} - Q_{\text{cor}} + 1} \frac{1}{Q_{\text{gen}} - Q_{\text{cor}}} = \frac{1}{Q_{\text{gen}}}.$$

Lemma 4. *There exist adversaries B_4 and B_5 against the unforgeability of the one-time signature scheme and the weak unforgeability of the Boneh-Boyer signature scheme such that*

$$\Pr[\text{Game}_3 = 1] \leq Q_{\text{sig}} \mathbf{Adv}_{\text{OT}}(B_4) + \mathbf{Adv}_{\text{BB}}(B_5)$$

Proof. We construct adversaries B_4 and B_5 as follows.

B_4 receives vk_{ot}^\dagger and simulates Game_3 honestly but with the following differences. It chooses a random $j^* \in \{1, \dots, Q_{\text{sig}}\}$ and answer the j^* th query to $\text{Sign}(i, m^\dagger, R^\dagger)$ honestly but computing $\sigma_{\text{ot}}^\dagger$ querying on (m^\dagger, R^\dagger) its oracle and setting vk_{ot}^\dagger as the corresponding one-time verification key. Finally, when A outputs its purported forgery $(m, R, (\sigma_{\text{ot}}, vk_{\text{ot}}, \dots))$, B_4 outputs the corresponding one-time signature.

B_5 receives $[vk]_2$ and simulates Game_3 honestly but with the following differences. Let $i := 0$. B_5 computes $(sk_{\text{ot}}^i, vk_{\text{ot}}^i) \leftarrow \text{OT.KeyGen}(gk)$, for each $1 \leq i \leq Q_{\text{sig}}$ and queries its signing oracle on $(vk_{\text{ot}}^1, \dots, vk_{\text{ot}}^{Q_{\text{sig}}})$ obtaining $[\sigma_1]_1, \dots, [\sigma_{Q_{\text{sig}}}]_1$. On the i^* th query of A to the key generation algorithm, B_5 picks $\mathbf{a} \leftarrow \mathcal{Q}$ and outputs $\tilde{vk} := ([vk]_2, [\mathbf{a}]_1, [\mathbf{a}]_2, \mathbf{a}[vk]_2)$. When A queries the signing oracle on input (i^*, m, R) , B_5 computes an honest signature but replaces vk_{ot} with vk_{ot}^i and $[\sigma]_1$ with $[\sigma_i]_2$, and then adds 1 to i . Finally, when A outputs its purported forgery $(m, R, (\sigma_{\text{ot}}, vk_{\text{ot}}, [c]_2, [\mathbf{d}]_1, \dots))$, it extracts $[\sigma]_1$ from $[\mathbf{d}]_1$ as its forgery for vk_{ot} .

Let E be the event where vk_{ot} , from the purported forgery of A , has been previously output by Sign . We have that

$$\Pr[\text{Game}_3 = 1] \leq \Pr[\text{Game}_3 = 1 | E] + \Pr[\text{Game}_3 = 1 | \neg E].$$

Since (m, R) has never been signed by a one-time signature and that, conditioned on E , the probability of $vk_{\text{ot}} = vk_{\text{ot}}^\dagger$ is $1/Q_{\text{sig}}$, then

$$Q_{\text{sig}} \mathbf{Adv}_{\text{OT}}(B_4) \geq \Pr[\text{Game}_3 = 1 | E]$$

Finally, if $\neg E$ holds, then $[\sigma]$ is a forgery for vk_{ot} and thus

$$\mathbf{Adv}_{\text{BB}}(B_5) \geq \Pr[\text{Game}_3 = 1 | \neg E]$$

4 Our Construction in the SXDH setting

Our construction follow the high-level description depicted in section 1.3 with the only difference that we do not use the verification key of the Boneh-Boyen signature vk , but a commitment to the secret key x . The only reason is efficiency since in this way we use Groth-Sahai proofs for integer equations instead of equations involving group elements.

For $\beta \in \{0, 1\}^m$ we define $h(\beta) := \sum_{i=1}^m \beta_i$ and $g_\beta(\mathbf{x}) := \sum_{i=1}^m \beta_i x_i$. Unlike the PPA-based construction, we do not prove collision resistance of h or g (g is not collision resistant). Instead, these functions are only used as shorthand and to keep an intuitive link with the PPA-based construction.

In the high level description of our ring signature in the SXDH setting from section 1.3 it was left to show how to derive a proof that $g_{\beta'}(\mathbf{x}') = g_\beta(\mathbf{x})$, which is described in following section. For completeness, in App. B.1 and App. B.2 we describe how to re-randomize proofs for the equations $\beta \in \{0, 1\}$ and $y = \beta x$ following [3].

4.1 NIZK proof that $g_{\beta'}(\mathbf{x}') = g_\beta(\mathbf{x})$

Let $[\mathbf{U}]_1$ and $[\mathbf{W}]_2$ Groth-Sahai commitment keys. Consider $[\mathbf{a}_i]_1 = \text{Com}(\beta_i; r_i)$, $[\mathbf{c}_i]_2 = \text{Com}_{[\mathbf{W}]_2}(x_i; s)$, and $[\mathbf{d}_i]_1 = \text{Com}_{[\mathbf{U}]_1}(y_i; t)$, where $y_i = \beta_i x_i$, $\beta \in \{0, 1\}$, $r, s, t \in \mathbb{Z}_q$, and $1 \leq i \leq m$. Consider also $[\mathbf{g}]_1$, a re-randomization of $\sum_{i=1}^m [\mathbf{d}_i]_1 = \text{Com}(g_\beta(\mathbf{x}))$, and $[\mathbf{A}']_1$ and $[\mathbf{C}']_2$ permutations of re-randomizations of $[\mathbf{A}]_1 := ([\mathbf{a}_1] \cdots [\mathbf{a}_m])$ and $[\mathbf{C}]_2 := ([\mathbf{c}_1]_2 \cdots [\mathbf{c}_m]_2)$, respectively. We want to construct a proof that $g_{\beta'}(\mathbf{x}') = g_\beta(\mathbf{x})$, or equivalently $\sum_{i=1}^m \beta'_i x'_i = \sum_{i=1}^m \beta_i x_i$, only from the extended verification keys and the random coins used in the re-randomizations.

Apart from $[\mathbf{a}_i]_1, [\mathbf{c}_i]_2, [\mathbf{d}_i]_1$, the extended verification key contains Groth-Sahai proofs $[\psi_i]_2, [\omega_i]_1$ for the equation $\beta_i x_i = y_i$ as defined in App. B.2. Each of these proofs satisfy the verification equation

$$[\mathbf{a}_i]_1 [\mathbf{c}_i^\top]_2 - [\mathbf{d}_i]_1 [\mathbf{w}_1^\top]_2 = [\mathbf{u}_2]_1 [\psi_i^\top]_2 + [\omega_i]_1 [\mathbf{w}_2^\top]_2.$$

$[\mathbf{A}']_1, [\mathbf{C}']_2$ and $[\mathbf{g}]_1$ are computed as $[\mathbf{A}']_1 = [\mathbf{A}]_1 \mathbf{P} + [\mathbf{u}_2]_1 \delta_a^\top$, $[\mathbf{C}']_2 = [\mathbf{C}]_2 \mathbf{P} + [\mathbf{w}_2]_2 \delta_c^\top$, and $[\mathbf{g}]_1 = \sum_{i=1}^m [\mathbf{d}_i]_1 + [\mathbf{u}_2]_1 \delta_g$, where \mathbf{P} is a permutation matrix and $\delta_a, \delta_c \in \mathbb{Z}_q^m$ and $\delta_g \in \mathbb{Z}_q$. The right side of the verification equation for equation $\sum_{i=1}^m \beta'_i x'_i - y = 0$, where $y = \sum_{i=1}^m \beta_i x_i$ is the opening of $[\mathbf{d}']_1$ and β', \mathbf{x}' are the openings of $[\mathbf{A}']_1$ and $[\mathbf{C}']_2$ respectively, is equal to

$$\begin{aligned} & [\mathbf{A}']_1 [\mathbf{C}'^\top]_2 - [\mathbf{d}']_1 [\mathbf{w}_1^\top]_2 \\ &= [\mathbf{A}]_1 \mathbf{P} \mathbf{P}^\top [\mathbf{C}^\top]_2 + [\mathbf{A}]_1 \mathbf{P} \delta_c [\mathbf{w}_2^\top]_2 + [\mathbf{u}_2]_1 \delta_a^\top [\mathbf{C}'^\top]_2 - [\mathbf{d}']_1 [\mathbf{w}_2^\top]_2 \\ &= \sum_{i=1}^m ([\mathbf{a}_i]_1 [\mathbf{c}_i^\top]_2 - [\mathbf{d}_i]_1 [\mathbf{w}_1^\top]_2) + [\mathbf{A}]_1 \mathbf{P} \delta_c [\mathbf{w}_2^\top]_2 + [\mathbf{u}_2]_1 (\delta_a^\top [\mathbf{C}'^\top]_2 - \delta_g [\mathbf{w}_1^\top]_2) \\ &= [\mathbf{u}_2]_1 \left(\sum_{i=1}^m [\psi_i]_1 + [\mathbf{C}']_2 \delta_a - \delta_g [\mathbf{w}_1]_2 \right)^\top + \left(\sum_{i=1}^m [\omega_i]_1 + [\mathbf{A}]_1 \mathbf{P} \delta_c \right) [\mathbf{w}_2^\top]_2. \end{aligned}$$

The last equation indicates that the proof must be the terms multiplying $[\mathbf{u}_2]_1$ and $[\mathbf{w}_2^\top]_2$ plus randomization terms. That is, for $\xi \leftarrow \mathbb{Z}_q$

$$\begin{aligned} [\boldsymbol{\psi}']_2 &= \sum_{i=1}^m [\boldsymbol{\psi}_i]_1 + [\mathbf{C}']_2 \boldsymbol{\delta}_a - \delta_g [\mathbf{w}_1]_2 + \xi [\mathbf{w}_2]_2 \\ [\boldsymbol{\omega}']_1 &= \sum_{i=1}^m [\boldsymbol{\omega}_i]_1 + [\mathbf{A}]_1 \mathbf{P} \boldsymbol{\delta}_c - \xi [\mathbf{u}_2]_1. \end{aligned} \quad (6)$$

Assuming $[\mathbf{d}']_1$ is correctly computed, the proof is sound because it satisfy the Groth-Sahai verification equation for $\sum_{i=1}^m \beta'_i x'_i - \sum_{i=1}^m \beta_i x_i = 0$. Furthermore, the proof is uniformly distributed conditioned on satisfying the verification equation and thus follows exactly the same distribution as a fresh Groth-Sahai proof.

4.2 Our Ring Signature

In the following let $n := |R|$, $m := \sqrt[3]{n}$, and for $1 \leq \alpha \leq n$ define $1 \leq \mu \leq n^{2/3}$ and $1 \leq \nu \leq m$ such that $\alpha = (\mu - 1)m + \nu$. For a sequence $\{s\}_{1 \leq i \leq n}$ we define $s_{\mu, \nu} := s_{(\mu-1)m+\nu}$. Consider $\text{OT} = (\text{OT.KeyGen}, \text{OT.Sign}, \text{OT.Ver})$ a one-time signature scheme. We assume that ring descriptions don't contain repeated elements.

CRSGen(gk): Pick three perfectly hiding CRS for the Groth-Sahai proof system ck_1, ck_2, ck'_2 , where $ck_1 := [\mathbf{U}]_1, ck_2 := [\mathbf{V}]_2, ck'_2 := [\mathbf{W}]_2$. We use ck_1, ck_2 for the $\Theta(\sqrt{n})$ set-membership of Chandran et al. The CRS is $\rho := (gk, ck_1, ck_2, ck'_2)$.

KeyGen(ρ): Pick $(x, [x]_2) \leftarrow \text{BB.KeyGen}(gk)$, compute $[\mathbf{a}]_1 := \text{Com}_{[\mathbf{U}]_1}(\beta = 0; r)$, where $r \leftarrow \mathbb{Z}_q$, plus a Groth-Sahai proof π that $\beta(\beta - 1) = 0$ (see App. B.1). Compute also $[\mathbf{c}]_2 = \text{GS.Com}_{ck'_2}(x; s)$, $[\mathbf{d}]_1 := \text{GS.Com}_{ck_1}(y; t)$, where $s, t \leftarrow \mathbb{Z}_q$, and a proof $[\boldsymbol{\psi}]_2, [\boldsymbol{\omega}]_1$ that $\beta x = y$ (see App. B.2). The secret key is x and the extended verification key is $\widetilde{vk} := ([x]_2, [\mathbf{a}]_1, [\mathbf{c}]_2, [\mathbf{d}]_1, \pi, [\boldsymbol{\psi}]_2, [\boldsymbol{\omega}]_1)$.

Sign $_{\rho, x}(m, R)$: Let $\alpha = (\mu - 1)m + \nu$ the index of the signer with respect to R .

1. Compute $(sk_{\text{ot}}, vk_{\text{ot}}) \leftarrow \text{OT.KeyGen}(gk)$ and $\sigma_{\text{ot}} \leftarrow \text{OT.Sign}_{sk_{\text{ot}}}(m, R)$.
2. For $1 \leq i \leq n^{2/3}$, let $[\mathbf{A}_i]_1 := [\mathbf{a}_{i,1}] \dots [\mathbf{a}_{i,m}]_1$, $[\mathbf{h}_i]_1 := \sum_{j=1}^m [\mathbf{a}_{i,j}]_1$ and $[\mathbf{g}_i]_1 := \sum_{j=1}^m [\mathbf{d}_{i,j}]_1$. Define the sets $H = \{[\mathbf{h}_1]_2, \dots, [\mathbf{h}_{n^{2/3}}]_2\}$ and $G = \{[\mathbf{g}_1]_2, \dots, [\mathbf{g}_{n^{2/3}}]_2\}$.
3. Let $[\mathbf{h}]_1 := [\mathbf{h}_\mu]_1 + \delta_h [\mathbf{u}_1]_1$ and $[\mathbf{g}]_1 = [\mathbf{g}_\mu]_1 + \delta_g [\mathbf{u}_2]_1$, $\delta_g, \delta_h \leftarrow \mathbb{Z}_q$. Compute proofs π_G and π_H that they belong to G and H , respectively. It is also proven that they appear in the same positions reusing the commitments to b_1, \dots, b_m and b'_1, \dots, b'_m , used in the set-membership proof of Chandran et al., which define $[\mathbf{h}]_1$'s and $[\mathbf{g}]_2$'s positions in H and G respectively.

4. Let $[\mathbf{C}']_2 := [\mathbf{c}_{\mu,\nu} | \mathbf{c}_{\mu,1} | \dots | \mathbf{c}_{\mu,m}]_2 + [\mathbf{w}_2]_2 \delta_c^\top$ and $[\mathbf{A}']_1 := [\mathbf{a}_{\mu,\nu} | \mathbf{a}_{\mu,1} | \dots | \mathbf{a}_{\mu,m}]_1 + [\mathbf{u}_2]_1 \delta_a^\top \in \mathbb{G}_1^{2 \times m}$, where $\delta_a, \delta_c \leftarrow \mathbb{Z}_q^m$ (the ν -th row is moved to the front of each matrix). Use $[\mathbf{A}_\mu]_1, [\mathbf{C}']_2, \mathbf{P}$ the permutation matrix that swaps the first element with the ν -th element, and $[\psi_{\mu,i}]_2, [\omega_{\mu,i}]_1$ plus $\delta_a, \delta_c, \delta_g$ to derive $\pi_g = ([\psi']_2, [\omega']_1)$, a proof that $g_{\beta'}(\mathbf{x}') = g_\beta(\mathbf{x})$, as in equation (6).
 5. Compute a proof π_h that $h(\beta') = h(\beta_\mu)$ as the GS proof that $\sum_{i=1}^m [\mathbf{a}'_i]_1 - [\mathbf{h}]_1 = \tilde{\delta}_h [\mathbf{u}_2]_1$, where $\tilde{\delta}_h = \sum_{i=1}^m \delta_{a,i} - \delta_h$.
 6. Compute a GS proof π_{bits} that β' , the vector of openings of \mathbf{A}' , belongs to $\{0, 1\}^m$ re-randomizing proofs $\pi_{\mu,\nu}, \pi_{\mu,1}, \dots, \pi_{\mu,m}$.
 7. Compute $[\sigma]_1 \leftarrow \text{BB.Sign}_{x_{\mu,\nu}}(vk_{\text{ot}})$, $[\mathbf{f}]_1 \leftarrow \text{GS.Com}_{ck_1}([\sigma]_1)$, and a GS proof π_{BB} of satisfiability of equation (2) with $[\mathbf{c}_{\mu,\nu}]_2$ the commitment to the secret key.
 8. Return the signature $\sigma := (vk_{\text{ot}}, \sigma_{\text{ot}}, [\mathbf{f}]_1, [\mathbf{A}']_2, [\mathbf{C}']_2, [\mathbf{g}]_1, [\mathbf{h}]_1, \pi_G, \pi_H, \pi_g, \pi_h, \pi_{\text{bits}}, \pi_{\text{BB}})$. (GS proofs include commitments to variables).
- Verify $_{\rho,R}(m, \sigma)$: Verify the validity of the one-time signature and of all the proofs. Return 0 if any of these checks fails and 1 otherwise.

We prove the following theorem which states the security of our construction.

Theorem 2. *The scheme presented in this section is a ring signature scheme with perfect correctness, perfect anonymity and computational unforgeability under the SXDH assumption, and the assumption that the one-time signature and the Boneh-Boyen signature are unforgeable. Concretely, for any PPT adversary \mathbf{A} against the unforgeability of the scheme, there exist adversaries $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ such that*

$$\text{Adv}(\mathbf{A}) \leq (Q_{\text{gen}}^2 + 1) \text{Adv}_{\text{SXDH}}(\mathbf{B}_1) + Q_{\text{gen}} Q_{\text{sig}} \text{Adv}_{\text{OT}}(\mathbf{B}_2) + Q_{\text{gen}} \text{Adv}_{\text{BB}}(\mathbf{B}_3),$$

where Q_{gen} and Q_{sig} are, respectively, upper bounds for the number of queries that \mathbf{A} makes to its VKGen and Sign oracles.

Proof. Perfect correctness follows directly from the definitions. Perfect anonymity follows from the fact that the perfectly hiding Groth-Sahai commitment keys defines perfectly hiding commitments and perfect witness-indistinguishable proofs, information theoretically hiding any information about \widetilde{vk} and x . Further, the re-randomized commitments are random elements \mathbb{G}_2^1 or \mathbb{G}_2^2 , and hence independent of the original commitments, and the re-randomized proofs follows the same distribution of the honest proofs and hence, they don't reveal any information about \widetilde{vk} and x .

We say that an unforgeability adversary is “eager” if makes all its queries to the VKGen oracle at the beginning. Note that any non-eager adversary \mathbf{A}' can be perfectly simulated by an eager adversary that makes Q_{gen} queries to VKGen and answers \mathbf{A}' queries to VKGen “on demand”. This is justified by the fact that the output of VKGen is independent of all previous outputs.

W.l.o.g. we assume that \mathbf{A} is an eager adversary. Computational unforgeability follows from the indistinguishability of the following games

- Game₀**: This is the real unforgeability experiment. **Game₀** returns 1 if the adversary **A** produces a valid forgery and 0 if not.
- Game₁**: This is game exactly as **Game₀** with the following differences:
- The commitment key ck'_2 is sampled together with its discrete logarithms from the perfectly binding distribution. Note that the discrete logarithms of ck'_2 allow to open commitments $[c_i]_2$ and $[c_j]_2$ for $i \in [Q_{\text{gen}}]$ and $j \in [m]$.
 - At the beginning, variables $\text{err}_1, \text{err}_2, \text{err}_3$ and err_4 are initialized to 0 and random index i^* from $\{1, \dots, Q_{\text{gen}}\}$ is chosen.
 - On a query to **Corrupt** with argument i , if $i = i^*$ set $\text{err}_3 \leftarrow 1$.
 - Let (m, R, σ) the purported forgery output by **A**.
 - * If $[x]_2 \notin R$, then set $\text{err}_1 = 1$.
 - * If $i^* \neq (m-1)\mu + i$ for all $i \in [m]$, where μ is the index defined in π_G and π_H , or there is some $j \in [m]$ such that $[x_{i^*}]_2 = [x'_j]_2$, then set $\text{err}_2 \leftarrow 1$.
 - * If $[x'_1]_2$, the opening of commitment $[c'_1]_2$ from σ , is not equal to $[x_{i^*}]_2$, set $\text{err}_4 \leftarrow 1$.
- Game₂**: This is game exactly as **Game₁** except that, if err_1 is set to 1, **Game₂** aborts.
- Game_{2,1}**: This game is exactly as **Game₁** except that, if at the onset $\text{err}_1 = 0$ or $\text{err}_2 = 1$, **Game_{2,1}** aborts.
- Game_{2,2}**: This game is exactly as **Game_{2,1}** except that in the i^* th query to **VKGen** commitment $[a_{i^*}]_1$ is set to $\text{Com}_{[\mathbb{U}]_1}(\beta_{i^*} = 1; r_{i^*})$, $r_{i^*} \leftarrow \mathbb{Z}_q$. Additionally, if err_3 is set to 1 abort.
- Game_{2,3}**: This game is exactly as **Game_{2,2}** except that ck_1 and ck_2 are sampled from the perfectly binding distribution.
- Game₃**: This is game exactly as **Game₂** except that, if err_3 or err_4 are set to 1, **Game₃** aborts.
- Game₄**: This is game exactly as **Game₃** except that, if err_3 is set to 1, **Game₄** aborts.

Since in **Game₁** variables $\text{err}_1, \text{err}_2$ and err_3 are just dummy variables, the only difference with **Game₀** comes from ck'_2 distribution. Similarly, the only difference between **Game_{2,2}** and **Game_{2,3}** comes from ck_1 and ck_2 distribution. It follows that there are adversaries $\mathbf{B}_1, \mathbf{B}_2$ against **SXDH** such that $|\Pr[\text{Game}_0 = 1] - \Pr[\text{Game}_1 = 1]| \leq \mathbf{Adv}_{\text{SXDH}}(\mathbf{B}_1)$ and $|\Pr[\text{Game}_{2,2} = 1] - \Pr[\text{Game}_{2,3} = 1]| \leq \mathbf{Adv}_{\text{SXDH}}(\mathbf{B}_2)$.

Lemma 5.

$$\Pr[\text{Game}_1 = 1] \leq \Pr[\text{Game}_2 = 1] + Q_{\text{gen}} \Pr[\text{Game}_{2,1} = 1]$$

Proof.

$$\begin{aligned} \Pr[\text{Game}_1 = 1] &= \Pr[\text{Game}_1 = 1 | \text{err}_1 = 0] \Pr[\text{err}_1 = 0] + \\ &\quad \Pr[\text{Game}_1 = 1 | \text{err}_1 = 1] \Pr[\text{err}_1 = 1] \\ &\leq \Pr[\text{Game}_2 = 1] + \Pr[\text{Game}_1 = 1 | \text{err}_1 = 1] \Pr[\text{err}_1 = 1] \end{aligned}$$

Now we proceed to bound $\Pr[\text{Game}_1 = 1 | \text{err}_1 = 1] \Pr[\text{err}_1 = 1]$. It holds that

$$\begin{aligned} \Pr[\text{Game}_{2,1} = 1] &= \Pr[\text{Game}_1 = 1, \text{err}_1 = 1, \text{err}_2 = 0] \\ &= \Pr[\text{err}_2 = 0 | \text{Game}_1 = 1, \text{err}_1 = 1] \Pr[\text{Game}_1 = 1, \text{err}_1 = 1] \\ &\geq \frac{1}{Q_{\text{gen}}} \Pr[\text{Game}_1 = 1 | \text{err}_1 = 1] \Pr[\text{err}_1 = 1]. \end{aligned}$$

where the last inequality follows from the fact that $\text{err}_1 = 1$ implies that $[x'_1]_2 \notin R$ and then $x'_i \neq x_{\mu,k}$ for all $k \in [m]$. Given that all entries of \mathbf{x}_μ must be different, there is least one $j \in [m]$ such that $x_{\mu,j} \neq x'_k$ for all $k \in [m]$. Since j^* is completely hidden to the adversary, it follows that $\Pr[\text{err}_2 = 0 | \text{Game}_1 = 1, \text{err}_1 = 1] \geq \Pr[j^* = (m-1)\mu + j] = 1/Q_{\text{gen}}$.

Lemma 6. $\Pr[\text{Game}_{2,1} = 1] \leq Q_{\text{gen}} \Pr[\text{Game}_{2,2} = 1]$

Proof. Since ck_1 and ck_2 are perfectly hiding there is no information revealed about β through the extended verification keys or the signatures. Then, it holds that $\Pr[\text{Game}_{2,2} = 1] = \Pr[\text{err}_3 = 0 | \text{Game}_{2,1} = 1] \Pr[\text{Game}_{2,1} = 1]$ and $\Pr[\text{err}_3 = 0 | \text{Game}_{2,1} = 1]$ is the probability that the Q_{corr} calls to **Corrupt** do not abort. Since \mathbf{A} is an eager adversary, the probability that i^* doesn't hit any of the Q_{corr} corrupted users is $(Q_{\text{gen}} - Q_{\text{corr}})/Q_{\text{gen}} \geq 1/Q_{\text{gen}}$ and then $\Pr[\text{Game}_{2,2} = 1] \geq 1/Q_{\text{gen}} \Pr[\text{Game}_{2,1} = 1]$.

Lemma 7. $\Pr[\text{Game}_{2,3} = 1] = 0$

Proof. Since ck_1, ck_2 and ck'_2 are perfectly binding, all Groth-Sahai proofs are perfectly sound. If π_{bits} and π_h are valid proofs, then β' , the opening of $[\mathbf{A}']$, is a permutation of β_μ . Since $\text{err}_1 = 1$ and $\text{err}_2 = 0$, it holds that $x_{i^*} = x_{\mu,i^*}$, for some $i^* \in [m]$, and $x_{\mu,i^*} \neq x'_j$ for all j . Furthermore, since $\beta_{i^*} = \beta_{\mu,i^*} = 1$, then $\beta_{j^*} = 1$ for some unique $j^* \in [m]$.

Finally, equation $\sum_{i=1}^m \beta'_i x'_i = \sum_{i=1}^m \beta_{\mu,i} x_{\mu,i}$ becomes $x'_{j^*} = x_{\mu,i^*}$, and therefore can't be satisfied. We conclude that π_{bits}, π_h , and π_g can't be valid proofs simultaneously and thus $\Pr[\text{Game}_{2,3} = 1] = 0$.

Lemma 8.

$$\Pr[\text{Game}_2 = 1] \leq Q_{\text{gen}} \Pr[\text{Game}_3 = 1].$$

Proof. It holds that

$$\begin{aligned} \Pr[\text{Game}_3 = 1] &= \Pr[\text{Game}_3 = 1 | \text{err}_3 = 0, \text{err}_4 = 0] \Pr[\text{err}_3 = 0, \text{err}_4 = 0] \\ &= \Pr[\text{Game}_2 = 1 | \text{err}_3 = 0, \text{err}_4 = 0] \Pr[\text{err}_3 = 0, \text{err}_4 = 1] \\ &= \Pr[\text{err}_3 = 0, \text{err}_4 = 0 | \text{Game}_2 = 1] \Pr[\text{Game}_2 = 1]. \end{aligned}$$

The probability that $\text{err}_3 = 0$ and $\text{err}_4 = 0$ given $\text{Game}_3 = 1$ is the probability that the Q_{corr} calls to **Corrupt** do not abort and that $[x'_1]_2 = [x_{i^*}]_2$. Since \mathbf{A} is an eager adversary, the probability that i^* doesn't hit any of the Q_{corr} corrupted users is $Q_{\text{gen}} - Q_{\text{corr}}/Q_{\text{gen}}$. Similarly, when \mathbf{A} outputs its purported forgery, the

probability that $[x'_1]_2 = [x_{i^*}]_2$ is $1/(Q_{\text{gen}} - Q_{\text{corr}})$, since $[x'_1]_2 \in R$ (or otherwise Game_3 would have aborted). Therefore

$$\Pr[\text{err}_3 = 0, \text{err}_4 = 0 | \text{Game}_2 = 1] = \frac{Q_{\text{gen}} - Q_{\text{corr}}}{Q_{\text{gen}}} \frac{1}{Q_{\text{gen}} - Q_{\text{corr}}} = \frac{1}{Q_{\text{gen}}}.$$

Lemma 9. *There exist adversaries \mathbf{B}_3 and \mathbf{B}_4 against the unforgeability of the one-time signature scheme and the weak unforgeability of the Boneh-Boyen signature scheme such that*

$$\Pr[\text{Game}_3 = 1] \leq Q_{\text{sig}} \mathbf{Adv}_{\text{OT}}(\mathbf{B}_3) + \mathbf{Adv}_{\text{BB}}(\mathbf{B}_4)$$

Proof. We construct adversaries \mathbf{B}_3 and \mathbf{B}_4 as follows.

\mathbf{B}_3 receives vk_{ot}^\dagger and simulates Game_3 honestly but with the following differences. It chooses a random $j^* \in \{1, \dots, Q_{\text{sig}}\}$ and answer the j^* th query to $\text{Sign}(i, m^\dagger, R^\dagger)$ honestly but computing $\sigma_{\text{ot}}^\dagger$ querying on (m^\dagger, R^\dagger) its oracle and setting vk_{ot}^\dagger as the corresponding one-time verification key. Finally, when \mathbf{A} outputs its purported forgery $(m, R, (\sigma_{\text{ot}}, vk_{\text{ot}}, \dots))$, \mathbf{B}_3 outputs the corresponding one-time signature.

\mathbf{B}_4 receives $[x]_2$ and simulates Game_3 honestly but with the following differences. Let $i := 0$. \mathbf{B}_4 computes $(sk_{\text{ot}}^i, vk_{\text{ot}}^i) \leftarrow \text{OT.KeyGen}(gk)$, for each $1 \leq i \leq Q_{\text{sig}}$ and queries its signing oracle on $(vk_{\text{ot}}^1, \dots, vk_{\text{ot}}^{Q_{\text{sig}}})$ obtaining $[\sigma_1]_1, \dots, [\sigma_{Q_{\text{sig}}}]_1$. On the i^* th query of \mathbf{A} to the key generation algorithm, \mathbf{B}_4 it computes $[\mathbf{a}]_1 := \beta[\mathbf{u}_1]_1 + r[\mathbf{u}_2]$, for $\beta = 0$, $[\mathbf{c}]_2 = [x]_2 \mathbf{w}_1 + s[\mathbf{w}_2]_2$ and $[\mathbf{d}]_1 = y[\mathbf{u}_1]_1 + t[\mathbf{u}_2]_1$ and $[\psi]_2, [\omega]_1$ as a Groth-Sahai proof for equation $\beta x = y$, for $\beta = y = 0$. The proof π_{bits} that $\beta \in \{0, 1\}$ is honestly computed and \mathbf{A} outputs $\mathbf{vk} := ([x]_2, [\mathbf{a}]_1, [\mathbf{c}]_2, [\mathbf{d}]_1, [\psi]_2, [\omega]_1, \pi)$. When \mathbf{A} queries the signing oracle on input (i^*, m, R) , \mathbf{B}_4 computes an honest signature but replaces vk_{ot} with vk_{ot}^i and $[\sigma]_1$ with $[\sigma_i]_2$, and then adds 1 to i . Finally, when \mathbf{A} outputs its purported forgery $(m, R, (\sigma_{\text{ot}}, vk_{\text{ot}}, [\mathbf{f}]_2, [\mathbf{A}']_1, \dots))$, it extracts $[\sigma]_1$ from $[\mathbf{f}]_1$ as its forgery for vk_{ot} .

Let E be the event where vk_{ot} , from the purported forgery of \mathbf{A} , has been previously output by Sign . We have that

$$\Pr[\text{Game}_4 = 1] \leq \Pr[\text{Game}_4 = 1 | E] + \Pr[\text{Game}_4 = 1 | \neg E].$$

Since (m, R) has never been signed by a one-time signature and that, conditioned on E , the probability of $vk_{\text{ot}} = vk_{\text{ot}}^\dagger$ is $1/Q_{\text{sig}}$, then

$$Q_{\text{sig}} \mathbf{Adv}_{\text{OT}}(\mathbf{B}_4) \geq \Pr[\text{Game}_4 = 1 | E]$$

Finally, if $\neg E$ holds, then $[\sigma]_1$ is a forgery for vk_{ot} and thus

$$\mathbf{Adv}_{\text{BB}}(\mathbf{B}_4) \geq \Pr[\text{Game}_4 = 1 | \neg E].$$

5 Acknowledgments

We thank to the anonymous reviewers for the feedback, which helped to simplify the SXDH-based construction. We also thanks Carla Ràfols and Mojtaba Khalili for their comments on earlier versions of this work. This work was funded in part by the French ANR ALAMBIC project (ANR-16-CE39-0006).

References

1. M. Abe, K. Haralambiev, and M. Ohkubo. Group to group commitments do not shrink. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 301–317, Cambridge, UK, Apr. 15–19, 2012. Springer, Heidelberg, Germany.
2. M. Abe, M. Kohlweiss, M. Ohkubo, and M. Tibouchi. Fully structure-preserving signatures and shrinking commitments. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 35–65, Sofia, Bulgaria, Apr. 26–30, 2015. Springer, Heidelberg, Germany.
3. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 108–125, Santa Barbara, CA, USA, Aug. 16–20, 2009. Springer, Heidelberg, Germany.
4. A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 60–79, New York, NY, USA, Mar. 4–7, 2006. Springer, Heidelberg, Germany.
5. D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.
6. D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.
7. P. Bose, D. Das, and C. P. Rangan. Constant size ring signature without random oracle. In E. Foo and D. Stebila, editors, *ACISP 15*, volume 9144 of *LNCS*, pages 230–247, Brisbane, QLD, Australia, June 29 – July 1, 2015. Springer, Heidelberg, Germany.
8. N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In L. Arge, C. Cachin, T. Jurdzinski, and A. Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 423–434, Wroclaw, Poland, July 9–13, 2007. Springer, Heidelberg, Germany.
9. M. Chase and A. Lysyanskaya. On signatures of knowledge. In C. Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 78–96, Santa Barbara, CA, USA, Aug. 20–24, 2006. Springer, Heidelberg, Germany.
10. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *EUROCRYPT’91*, volume 547 of *LNCS*, pages 257–265, Brighton, UK, Apr. 8–11, 1991. Springer, Heidelberg, Germany.
11. G. Danezis, C. Fournet, J. Groth, and M. Kohlweiss. Square span programs with applications to succinct NIZK arguments. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550, Kaoshiung, Taiwan, R.O.C., Dec. 7–11, 2014. Springer, Heidelberg, Germany.
12. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 609–626, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.
13. S. Galbraith, K. Paterson, and N. Smart. Pairings for cryptographers. Cryptology ePrint Archive, Report 2006/165, 2006. <http://eprint.iacr.org/2006/165>.

14. R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct NIZKs without PCPs. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.
15. C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 99–108, San Jose, CA, USA, June 6–8, 2011. ACM Press.
16. S. Goldwasser and Y. T. Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115, Cambridge, MA, USA, Oct. 11–14, 2003. IEEE Computer Society Press.
17. A. González, A. Hevia, and C. Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629, Auckland, New Zealand, Nov. 30 – Dec. 3, 2015. Springer, Heidelberg, Germany.
18. C. Gritti, W. Susilo, and T. Plantard. Logarithmic size ring signatures without random oracles. *IET Information Security*, 10(1):1–7, 2016.
19. J. Groth. On the size of pairing-based non-interactive arguments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.
20. J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 253–280, Sofia, Bulgaria, Apr. 26–30, 2015. Springer, Heidelberg, Germany.
21. J. Groth and S. Lu. A non-interactive shuffle with pairing based verifiability. In K. Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, Dec. 2–6, 2007. Springer, Heidelberg, Germany.
22. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.
23. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In N. P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, Istanbul, Turkey, Apr. 13–17, 2008. Springer, Heidelberg, Germany.
24. C. S. Jutla and A. Roy. Improved structure preserving signatures under standard bilinear assumptions. Cryptology ePrint Archive, Report 2017/025, 2017. <http://eprint.iacr.org/2017/025>.
25. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 1–31, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.
26. G. Malavolta and D. Schröder. Efficient ring signatures in the standard model. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 128–157, Hong Kong, China, Dec. 3–7, 2017. Springer, Heidelberg, Germany.
27. P. Morillo, C. Ràfols, and J. L. Villar. The kernel matrix Diffie-Hellman assumption. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758, Hanoi, Vietnam, Dec. 4–8, 2016. Springer, Heidelberg, Germany.

28. M. Naor. On cryptographic assumptions and challenges (invited talk). In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109, Santa Barbara, CA, USA, Aug. 17–21, 2003. Springer, Heidelberg, Germany.
29. C. Ràfols. Stretching groth-sahai: NIZK proofs of partial satisfiability. In Y. Dodis and J. B. Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276, Warsaw, Poland, Mar. 23–25, 2015. Springer, Heidelberg, Germany.
30. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In C. Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565, Gold Coast, Australia, Dec. 9–13, 2001. Springer, Heidelberg, Germany.
31. P. Rogaway and T. Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 371–388, New Delhi, India, Feb. 5–7, 2004. Springer, Heidelberg, Germany.
32. V. Shoup. Lower bounds for discrete logarithms and related problems. In W. Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.

A Security of the m -PPA assumption in Asymmetric Groups

We use the natural generalization of Shoup’s generic group model [32] to the asymmetric bilinear setting, as it was used for instance by Boneh et al. [6]. In such a model an adversary can only access elements of $\mathbb{G}_1, \mathbb{G}_2$ or \mathbb{G}_T via a query to a group oracle, which gives him a randomized encoding of the queried element. The group oracle must be consistent with the group operations (allowing to query for the encoding of constants in either group, for the encoding of the sum of previously queried elements in the same group and for the encoding of the product of pairs in $\mathbb{G}_1 \times \mathbb{G}_2$).

We prove the following theorem which states generic security of the m -PPA assumption.

Theorem 3. *If the m -PPA assumption holds in generic symmetric bilinear groups, then the m -PPA holds in generic asymmetric bilinear groups.*

Proof. Suppose there is an adversary A in the asymmetric generic bilinear group model against the m -PPA assumption. We show how to construct an adversary B against the m -PPA assumption in the symmetric generic group model.

Adversary B has oracle access to the randomized encodings $[\cdot] : \mathbb{Z}_q \rightarrow \mathcal{E}(\mathbb{G})$, and $[\cdot]_T : \mathbb{Z}_q \rightarrow \mathcal{E}(\mathbb{G}_T)$, where $\mathcal{E}(\mathbb{G}), \mathcal{E}(\mathbb{G}_T)$ is the set of all strings that encodes elements of \mathbb{G}, \mathbb{G}_T , respectively. It also has access to oracles $+$: $\mathcal{E}(\mathbb{G}) \times \mathcal{E}(\mathbb{G}) \rightarrow \mathcal{E}(\mathbb{G})$, $+_T$: $\mathcal{E}(\mathbb{G}_T) \times \mathcal{E}(\mathbb{G}_T) \rightarrow \mathcal{E}(\mathbb{G}_T)$, and e : $\mathcal{E}(\mathbb{G}) \times \mathcal{E}(\mathbb{G}) \rightarrow \mathcal{E}(\mathbb{G}_T)$. At the beginning, B receives as a challenge $\{[a_i], [a_i^2] : 1 \leq i \leq m\}$.

Adversary B simulates the generic hardness game for A as follows. It defines encodings

$$\widetilde{[\cdot]}_s : \mathbb{Z}_q \rightarrow \mathcal{E}(\mathbb{G}_s) \text{ and also oracles } \widetilde{+}_s : \mathcal{E}(\mathbb{G}_s) \times \mathcal{E}(\mathbb{G}_s), \quad \widetilde{e} : \mathcal{E}(\mathbb{G}_1) \times \mathcal{E}(\mathbb{G}_2) \rightarrow \mathcal{E}(\mathbb{G}_T)$$

where, $s \in \{1, 2, T\}$ and $\widetilde{[\cdot]}_s$ are random injective functions conditioned on satisfying the group laws, for which it suffices that for all $s \in \{1, 2, T\}$:

1. $a\widetilde{[x]}_s \widetilde{+}_s b\widetilde{[y]}_s = \widetilde{[ax + by]}_s$.
2. $\widetilde{e}(\widetilde{[x]}_1, \widetilde{[y]}_2) = \widetilde{[xy]}_T$

As usual in the generic group model, we can assume that any call to $\widetilde{+}_s$ and \widetilde{e} receives as arguments two polynomials $p \in \mathbb{Z}_q[X_1, \dots, X_m], q \in \mathbb{Z}[Y_1, \dots, Y_n]$, where n and m is the number of elements known by the adversary in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, as appropriate. Thereby, $p\widetilde{+}_s q$ and $\widetilde{e}(p, q)$ are randomly chosen from $\mathcal{E}(\mathbb{G}_s)$ if $\widetilde{[p + q]}_s$ was not previously queried, or equal to the same encoding answered previously. From this, is not hard to see that such $\widetilde{[\cdot]}_s$ can be efficiently constructed keeping track of the elements that the adversary knows.

We abuse of notation and omit the $\widetilde{\cdot}$ (“tilde”) for the simulated oracles, as long as we will no refer anymore to B’s oracles. Initially, B picks $\tilde{a}_i \leftarrow \mathbb{Z}_q$, for

$1 \leq i \leq m$, and runs **A** on input $\{[\tilde{a}_i]_s, [\tilde{a}_i^2]_s : 1 \leq i \leq m, s \in \{1, 2\}\}$. At the onset of the simulation **A** will output as a solution to the challenge a pair

$$\tilde{\mathbf{Z}} = \begin{pmatrix} \tilde{z}_{1,1} \cdots \tilde{z}_{1,m} \\ \tilde{z}_{2,1} \cdots \tilde{z}_{2,m} \end{pmatrix}, \tilde{\mathbf{z}} = (\tilde{z}_1, \dots, \tilde{z}_m)$$

such that they are, respectively, the result of applying the encoding functions $[\cdot]_1$ to degree 2 polynomials $p_{1,1}, \dots, p_{2,m} \in \mathbb{Z}_q[A_1, \dots, A_m]$ evaluated on random $\tilde{a}_1, \dots, \tilde{a}_m \in \mathbb{Z}_q$, and the result of applying the encoding function $[\cdot]_2$ to degree 2 polynomials $\underline{p}_1, \dots, \underline{p}_m \in \mathbb{Z}_q[\underline{A}_1, \dots, \underline{A}_m]$ evaluated on $\underline{a}_1 = \tilde{a}_1, \dots, \underline{a}_m = \tilde{a}_m$. If the challenge is successful it must also hold that

$$\begin{aligned} [p_{1,i}(\tilde{a}_1, \dots, \tilde{a}_m)]_1 [1]_2 &= [1]_1 [p_i(\tilde{a}_1, \dots, \tilde{a}_m)]_2 \\ \iff p_{1,i}(\tilde{a}_1, \dots, \tilde{a}_m) &= \underline{p}_i(\tilde{a}_1, \dots, \tilde{a}_m) \text{ for each } 1 \leq i \leq m, \end{aligned} \quad (7)$$

$$\begin{aligned} [p_{2,i}(\tilde{a}_1, \dots, \tilde{a}_m)]_1 [1]_2 &= [p_i(\tilde{a}_1, \dots, \tilde{a}_m)]_1 [p_i(\tilde{a}_1, \dots, \tilde{a}_m)]_2 \\ \iff p_{2,i}(\tilde{a}_1, \dots, \tilde{a}_m) &= p_{1,i}(\tilde{a}_1, \dots, \tilde{a}_m) \underline{p}_i(\tilde{a}_1, \dots, \tilde{a}_m), \end{aligned} \quad (8)$$

$$\begin{aligned} \text{and } \sum_{i=1}^m [p_{1,i}(\tilde{a}_1, \dots, \tilde{a}_m)]_1 &= \sum_{i=1}^m [\tilde{a}_i]_1, \quad \sum_{i=1}^m [p_{2,i}(\tilde{a}_1, \dots, \tilde{a}_m)]_1 = \sum_{i=1}^m [\tilde{a}_i^2]_1 \\ \iff \sum_{i=1}^m p_{1,i}(\tilde{a}_1, \dots, \tilde{a}_m) &= \sum_{i=1}^m \tilde{a}_i, \quad \sum_{i=1}^m p_{2,i}(\tilde{a}_1, \dots, \tilde{a}_m) = \sum_{i=1}^m \tilde{a}_i^2 \end{aligned} \quad (9)$$

since $[a]_1 [b]_2 = [ab]_T$ and $[c]_s = [d]_s$ iff $c = d$.

Given that $\tilde{a}_1, \dots, \tilde{a}_m$ remain statistically hidden to **A**, it must choose

$$\begin{aligned} p_{i,1} &\equiv \underline{p}_i, \quad p_{2,i} \equiv p_{1,i} \cdot \underline{p}_i, \text{ such that} \\ \sum_{i=1}^m p_{1,i} &= \sum_{i=1}^m A_i \text{ and } \sum_{i=1}^m p_{2,i} = \sum_{i=1}^m A_i^2 \end{aligned}$$

since otherwise, by the Schwartz-Zippel lemma, equations (7), (8) and (9) only hold with negligible probability. We conclude that $p_{2,i} \equiv p_{1,i}^2$.

Finally, **B** computes its own response to its challenge. Since equations (7), (8) and (9) imply equality for polynomials, they must hold for any assignment of the variables. In particular they hold when A_i is assigned to a_i , the discrete logarithms of **B**'s challenge. Therefore, **B** computes its own response as

$$\mathbf{Z} = \begin{pmatrix} z_{1,1} \cdots z_{1,m} \\ z_{2,1} \cdots z_{2,m} \end{pmatrix},$$

where $z_{i,j} := [p_{i,j}(a_1, \dots, a_m)]$, for $i \in \{1, 2\}$ and $1 \leq j \leq m$, which can be computed from **B**'s challenge replacing each occurrence of A_i with $[a_i]$, and each occurrence of A_i^2 with $[a_i^2]$. It holds that

$$\sum_{j=1}^m z_{i,j} = \sum_{j=1}^m z_i = \sum_{j=1}^m [a_j].$$

Given, that $z_{2,j} = p_{2,j}(a_1, \dots, a_m) = p_{1,j}^2(a_1, \dots, a_m)$, and thus $e([z_{1,j}], [z_{1,j}]) = [p_{1,j}^2(a)]_T = e([z_{2,j}], [1])$, we conclude that \mathbf{Z} which is a solution of the m -PPA assumption.

B Re-randomizable Groth-Sahai NIZK proofs

In this section we detail the Groth-Sahai proofs used by the SXDH-based ring signature. We also describe how these proofs can be re-randomized, as shown by Belenky et al. [3].

B.1 Groth-Sahai proofs for $\beta(\beta - 1) = 0$

To prove membership in \mathcal{Q}_1 we construct Groth-Sahai proofs for

$$\beta(1 - \beta) = 0, \quad (10)$$

for $\beta = \beta_1, \dots, \beta_m$. To do so we compute an additional commitment to β' , $[\mathbf{b}]_2 = \beta'[\mathbf{v}_1]_2 + \rho[\mathbf{v}_2]_2$ and proofs

$$\begin{aligned} [\boldsymbol{\theta}]_2 &= r([\mathbf{v}_1]_2 - [\mathbf{b}]_2) + \delta[\mathbf{v}_2]_2 & [\boldsymbol{\pi}]_1 &= \beta\rho[\mathbf{u}_1]_1 - \delta[\mathbf{u}_2]_1 \\ [\boldsymbol{\xi}]_2 &= r[\mathbf{v}_1]_2 + \delta'[\mathbf{v}_2]_2 & [\boldsymbol{\phi}]_2 &= \rho[\mathbf{u}_1]_1 - \delta'[\mathbf{u}_2]_1, \end{aligned} \quad (11)$$

that $\beta(1 - \beta') = 0$ and $\beta = \beta'$. These proofs satisfy the following verification equations

$$[\mathbf{a}]_1([\mathbf{v}_1]_2 - [\mathbf{b}]_2)^\top = [\mathbf{u}_2]_1[\boldsymbol{\theta}^\top]_2 + [\boldsymbol{\pi}]_1[\mathbf{v}_2^\top]_2 \text{ and} \quad (12)$$

$$[\mathbf{a}]_1[\mathbf{v}_1^\top]_2 - [\mathbf{u}_1]_1[\mathbf{b}^\top]_2 = [\mathbf{u}_2]_1[\boldsymbol{\xi}^\top]_2 + [\boldsymbol{\phi}]_1[\mathbf{v}_2^\top]_2. \quad (13)$$

Further, these proofs can be re-randomized as noted in [3]. That is, given only $[\mathbf{a}]_1, [\mathbf{b}]_2$ and $[\boldsymbol{\theta}]_2, [\boldsymbol{\pi}]_1, [\boldsymbol{\xi}]_2, [\boldsymbol{\phi}]_2$ (and not its openings nor randomness) satisfying (12) and (13), we can compute new commitments $[\mathbf{a}']_1$ and $[\mathbf{b}']_2$ and proofs $[\boldsymbol{\theta}']_2, [\boldsymbol{\pi}']_1, [\boldsymbol{\xi}']_2, [\boldsymbol{\phi}']_2$ for the satisfiability of equation $\beta(1 - \beta) = 0$. For $\gamma, \epsilon, \zeta, \zeta' \leftarrow \mathbb{Z}_q$, the re-randomized commitments and proofs are computed as

$$\begin{aligned} [\mathbf{a}']_1 &= [\mathbf{a}]_1 + \gamma[\mathbf{u}]_2, & [\mathbf{b}']_2 &= [\mathbf{b}]_2 + \epsilon[\mathbf{v}_2]_2 \\ [\boldsymbol{\theta}']_2 &= [\boldsymbol{\theta}]_2 + \gamma([\mathbf{v}_1]_2 - [\mathbf{b}']_2) + \zeta[\mathbf{v}_2]_2 & [\boldsymbol{\pi}']_1 &= [\boldsymbol{\pi}]_1 + \epsilon[\mathbf{a}]_1 - \zeta[\mathbf{u}_2]_1 \\ [\boldsymbol{\xi}']_2 &= [\boldsymbol{\xi}]_2 + \gamma[\mathbf{v}_1]_2 + \zeta'[\mathbf{v}_2]_2 & [\boldsymbol{\phi}']_1 &= [\boldsymbol{\phi}]_1 + \epsilon[\mathbf{u}_1]_1 - \zeta'[\mathbf{u}_2]_1, \end{aligned} \quad (14)$$

The following Lemma formally states the security of the previous proofs.

Lemma 10. *Consider the quadratic equation $\beta(1 - \beta) = 0$ whose variable is β . The proof system whose crs consists Groth-Sahai perfectly binding commitment keys $[\mathbf{U}]_1, [\mathbf{V}]_2$, whose prover computes the proofs as in (11), and the verifier verifies equations (12) and (13), is perfectly complete and sound, and computationally zero-knowledge under the SXDH assumption. Further, the re-randomized proofs from (14) follow exactly the same distribution as the proofs computed by the prover.*

Proof. Completeness follows by inspection. Soundness follows from the fact that, whenever \mathbf{U}, \mathbf{V} come from the perfectly binding distribution, $\mathbf{u}_1 \mathbf{v}_1^\top, \mathbf{u}_1 \mathbf{v}_2^\top, \mathbf{u}_2 \mathbf{v}_1^\top, \mathbf{u}_2 \mathbf{v}_2^\top$ form basis of $\mathbb{Z}_q^{2 \times 2}$. Since in equations (12) and (13) the right sides have no components in $\mathbf{u}_1 \mathbf{v}_1$ and left sides components are, respectively, $\beta(1 - \beta')$ and $\beta - \beta'$, we conclude that $\beta(1 - \beta') = 0$ and $\beta = \beta'$.

Computational zero-knowledge follows from the following argument. When $[\mathbf{U}]_1, [\mathbf{V}]_2$ are sampled from the perfectly hiding distribution, $\mathbf{u}_1 = \mu \mathbf{u}_2$ and $\mathbf{v}_1 = \nu \mathbf{v}_2$, for some random μ and ν . In this setting we can sample $\mathbf{a} = r \mathbf{u}_2$ without changing \mathbf{a} 's distribution, and we can simulate the proofs for any $[\mathbf{b}]_2$ as

$$\begin{aligned} [\boldsymbol{\theta}]_2 &= r([\mathbf{v}_1]_1 - [\mathbf{b}]_2) + \delta[\mathbf{v}_2]_2 & [\boldsymbol{\pi}]_1 &= -\delta[\mathbf{u}_2]_1 \\ [\boldsymbol{\xi}]_2 &= r[\mathbf{v}_1]_2 - \mu[\mathbf{b}]_2 + \delta'[\mathbf{v}_2]_2 & [\boldsymbol{\phi}]_2 &= -\delta'[\mathbf{u}_2]_1. \end{aligned} \quad (15)$$

In particular, our simulator considers $[\mathbf{b}]_2 = \rho[\mathbf{v}_2]_2$, for $\rho \leftarrow \mathbb{Z}_q$, which follows exactly the same distribution as in the honest proof.

Note that both, the honest and the simulated proofs, follows exactly the same distribution. Indeed in both cases, \mathbf{a} and \mathbf{b} are uniformly distributed in, respectively, $\text{Span}(\mathbf{u}_2)$ and $\text{Span}(\mathbf{v}_2)$, and the proofs are uniformly chosen among those that satisfy the respective verification equation.

Finally, is direct that the re-randomized proofs follow the same distribution as the real proofs. Define $\tilde{r} := r + \gamma, \tilde{\rho} := \rho + \epsilon, \tilde{\delta} := \delta + \zeta - \epsilon(r + \gamma), \tilde{\delta}' := \zeta'$, which are uniformly distributed over \mathbb{Z}_q . It follows by inspection that the re-randomized proof can be obtained as a real proof using $\tilde{r}, \tilde{\rho}, \tilde{\delta}$ and $\tilde{\delta}'$ as random coins.

B.2 Groth-Sahai proofs for $\beta x = y$

We construct a Groth-Sahai proof that $\beta x = y$ and then show how can be re-randomized. Let $r, s, t, \delta \leftarrow \mathbb{Z}_q$, these proofs consists of two vectors

$$[\boldsymbol{\psi}]_2 = r[\mathbf{c}]_2 - t[\mathbf{w}_1]_2 - \delta[\mathbf{w}_2]_2 \quad [\boldsymbol{\omega}]_1 = \beta s[\mathbf{u}]_1 + \delta[\mathbf{u}_2]_1 \quad (16)$$

and commitments to β, x and to y

$$\begin{aligned} [\mathbf{a}]_1 &= \beta[\mathbf{u}_1]_1 + r[\mathbf{u}]_2, & [\mathbf{c}]_2 &= x[\mathbf{w}_1]_2 + s[\mathbf{w}_2]_2, \\ [\mathbf{d}]_1 &= \beta x[\mathbf{u}_1]_2 + t[\mathbf{u}_2]_2, \end{aligned}$$

satisfying the following verification equation

$$[\mathbf{a}]_1[\mathbf{c}^\top]_2 - [\mathbf{d}]_1[\mathbf{w}_1]_2 = [\mathbf{u}_2]_1[\boldsymbol{\psi}^\top]_2 + [\boldsymbol{\omega}]_1[\mathbf{w}_2^\top]_2. \quad (17)$$

These proofs and commitments can be re-randomized as follows

$$\begin{aligned} [\mathbf{a}']_1 &= [\mathbf{a}]_1 + \gamma[\mathbf{u}_2]_1 & [\mathbf{d}']_1 &= [\mathbf{d}]_1 + \eta[\mathbf{u}_2]_1 \\ [\mathbf{c}']_2 &= [\mathbf{c}]_2 + \alpha[\mathbf{w}_2]_2, & [\boldsymbol{\omega}']_1 &= [\boldsymbol{\omega}]_1 + \alpha[\mathbf{a}]_1 + \tau[\mathbf{u}_2]_1, \\ [\boldsymbol{\psi}']_2 &= [\boldsymbol{\psi}]_2 + \gamma[\mathbf{c}']_2 - \eta[\mathbf{w}_1]_2 - \tau[\mathbf{w}_2]_2, \end{aligned} \quad (18)$$

for $\gamma, \alpha, \eta, \tau \leftarrow \mathbb{Z}_q$.

We prove the following Lemma.

Lemma 11. *Consider the the quadratic equation $\beta x = y$, whose variables are β, x, y . The proof system whose crs consists of perfectly binding Groth-Sahai commitment keys $[\mathbf{U}]_1, [\mathbf{W}]$, whose prover computes the proofs as in (16), and the verifier verifies equation (17), is perfectly complete and sound, and computationally zero-knowledge under the SXDH assumption. Further, the proofs from (18) follows exactly the same distribution as the proofs computed by the prover.*

Proof. Completeness follows by inspection. Soundness follows from the fact that, whenever \mathbf{U}, \mathbf{W} come from the perfectly binding distribution, $\mathbf{u}_1 \mathbf{w}_1^\top, \mathbf{u}_1 \mathbf{w}_2^\top, \mathbf{u}_2 \mathbf{w}_1^\top, \mathbf{u}_2 \mathbf{w}_2^\top$ form basis of $\mathbb{Z}_q^{2 \times 2}$. The right side of equation (17) has no component in $\mathbf{u}_1 \mathbf{w}_1^\top$, while the left side component is $\beta x - y$. Hence, we conclude that $\beta x = y$.

Computational zero-knowledge follows from the following argument. When $[\mathbf{U}]_1, [\mathbf{W}]_2$ are sampled from the perfectly hiding distribution, $\mathbf{u}_1 = \mu \mathbf{u}_2$ and $\mathbf{w}_1 = \nu \mathbf{w}_2$, for some random μ and ν . In this setting we can sample $\mathbf{a} = r \mathbf{u}_2$ without changing \mathbf{a} 's distribution, and we can simulate the proofs for any $[\mathbf{c}]_2, [\mathbf{d}]_1$ as

$$[\psi]_2 = r[\mathbf{c}]_2 - \delta[\mathbf{w}_2]_2 \text{ and } [\omega]_1 = [\mathbf{d}]_1 \nu + \delta[\mathbf{u}_2]_1, \text{ for } \delta \leftarrow \mathbb{Z}_q. \quad (19)$$

In particular, our simulator sets $[\mathbf{c}]_2 = s[\mathbf{w}_2]_2, [\mathbf{d}]_1 = t[\mathbf{u}_2]_1$ for $s, t \leftarrow \mathbb{Z}_q$.

Note that both, the honest and the simulated proofs, follows exactly the same distribution. Indeed in both cases, \mathbf{a} and \mathbf{c}, \mathbf{d} are uniformly distributed in, respectively, $\text{Span}(\mathbf{u}_2)$ and $\text{Span}(\mathbf{w}_2)$, and the proofs are uniformly chosen among those that satisfy the respective verification equation.

Finally, is direct that the re-randomized proofs follow the same distribution as the real proofs. Define $\tilde{r} := r + \gamma, \tilde{s} := s + \alpha, \tilde{t} := t + \eta, \tilde{\delta} := \delta + \tau + r\alpha$, which are uniformly distributed over \mathbb{Z}_q . It follows by inspection that the re-randomized proof can be obtained as a real proof using $\tilde{r}, \tilde{s}, \tilde{t}$ and $\tilde{\delta}$ as random coins.

C Efficiency Analysis

In this section we calculate the proof size, in number of group elements, signer time, in number of exponentiations, and verifier time, in number of exponentiations, values which are shown in Table 1. For simplicity, we ignore the influence of the one-time signature as it affects all schemes in the same way. Further, we note that one-time signatures can be constructed from symmetric-key primitives and thus, its costs are negligible in comparison to bilinear groups based primitives. We just remark that the three schemes require a hash function $H : \mathcal{VK}_{\text{ots}} \rightarrow \mathbb{Z}_q$, which hashes one-time verification keys into integers modulo q .

C.1 Chandran et al.'s Ring Signature

Signature Size: The number of elements of \mathbb{G}_1 required for the signature is

- 2 for committing to the Boneh-Boyen signature,
- 4 for the proof that the Boneh-Boyen signature is correct,
- $4\sqrt{n}$ for commitments to $b_1, \dots, b_m, b'_1, \dots, b'_m$,

- $8\sqrt{n}$ for proof i,

- 4 for proof iv,

which adds up to $12\sqrt{n} + 10$.

The number of elements of \mathbb{G}_2 required for the signature is:

- 2 for committing to the Boneh-Boyen verification key,

- 4 for the proof that the Boneh-Boyen signature is correct,

- $4\sqrt{n}$ for commitments to $b_1, \dots, b_m, b'_1, \dots, b'_m$,

- $2\sqrt{n}$ for commitments to $[\kappa_1]_2, \dots, [\kappa_m]_2$,

- $8\sqrt{n}$ for proof i,

- \sqrt{n} for proof iii,

- 2 elements for proof iv,

which adds up to $15\sqrt{n} + 8$. Additionally, it requires \sqrt{n} elements of \mathbb{Z}_q .

Signature Time: The number of exponentiations computed by the prover is:

- 1 for the Boneh-Boyen signature

- 8 for computing $[c]_2$ and $[d]_1$,

- 8 for proof that the Boneh-Boyen signature is correct,

- $16\sqrt{n}$ for commitments to $b_1, \dots, b_m, b'_1, \dots, b'_m$,

- $4\sqrt{n}$ for commitments to $[\kappa_1]_2, \dots, [\kappa_m]_2$,

- $16\sqrt{n}$ for proof i,

- \sqrt{n} for proof iii,

- 6 for proof iii,

which adds up to $37\sqrt{n} + 23$ exponentiations.

Verification Time: The number of pairing operations for verifying a signature is:

- 24 for verifying the proof that the Boneh-Boyen signature is correct,

- $48\sqrt{n}$ for proof i,

- $2n + 4\sqrt{n}$ for proof iii,

- $8\sqrt{n} + 14$ for proof iv,

which adds up to $2n + 60\sqrt{n} + 38$ exponentiations.

C.2 Our Ring Signature in the PPA setting

Signature Size: The number of elements of \mathbb{G}_1 required for the signature is

- 2 for committing to the Boneh-Boyen signature,

- 4 for the proof that the Boneh-Boyen signature is correct,

- 4 for commitment to $[x]_1$,

- $18\sqrt[3]{n} + 12$ for membership in H and G ,

- $4\sqrt[3]{n}$ for commitments to A' .

- $8\sqrt[3]{n}$ for proof π_{Q_m} ,

- 2 for proof π_h ,

- 8 for π_g ,

which adds up to $30\sqrt[3]{n} + 32$.

The number of elements of \mathbb{G}_2 required for the signature is:

- 2 for committing to the Boneh-Boyen verification key,

- 4 for the proof that the Boneh-Boyen signature is correct,

- 4 for commitment to $[y]_2$,

- $18\sqrt[3]{n} + 8$ for membership in H and G ,
- $2\sqrt[3]{n} - 2$ for commitments to $[\kappa']_2$.
- $8\sqrt[3]{n}$ for proof π_{Q_m} ,
- 8 for π_g ,

which adds up to $28\sqrt[3]{n} + 28$. Additionally, it requires $\sqrt[3]{n}$ elements of \mathbb{Z}_q .

Signature Time: The number of exponentiations computed by the prover is:

- 1 for the Boneh-Boyen signature
- 8 for computing $[c]_2$ and $[d]_1$,
- 8 for proof that the Boneh-Boyen signature is correct,
- 16 for commitments to $[x]_1$ and $[y]_2$,
- $53\sqrt[3]{n}$ for membership in G and H ,
- $12\sqrt[3]{n} - 4$ for commitments to A' and $[\kappa']_2$,
- $16\sqrt[3]{n}$ for proof π_{Q_m} ,
- 2 for proof π_{Q_h} ,
- 16,

which adds up to $80\sqrt[3]{n} + 71$.

Verification Time: The number of paring operations for verifying a signature is:

- 24 for verifying the proof that the Boneh-Boyen signature is correct,
 - $8n^{2/3} + 96\sqrt[3]{n} + 56$ for membership in H and G ,
 - $46\sqrt[3]{n}$ for proof π_{Q_m} ,
 - $4\sqrt[3]{n} + 8$ for proof π_h ,
 - $16\sqrt[3]{n} + 36$ for proof π_g ,
- which adds up to $8n^{2/3} + 162\sqrt[3]{n} + 118$.

C.3 Our Ring Signature in the SXDH setting

Signature Size: The number of elements of \mathbb{G}_1 required for the signature is

- 2 for committing to the Boneh-Boyen signature,
- 4 for the proof that the Boneh-Boyen signature is correct,
- 4 for $[h]_1$ and $[g]_1$,
- $12\sqrt[3]{n} + 16$ for membership in H and G ,
- $2\sqrt[3]{n}$ for $[A']_1$,
- 2 for $[\omega']_1$,
- $4\sqrt[3]{n}$ for proof π_{bits} ,
- 2 for proof π_h ,
- 8 for π_g ,

which adds up to $18\sqrt[3]{n} + 30$.

The number of elements of \mathbb{G}_2 required for the signature is:

- 4 for the proof that the Boneh-Boyen signature is correct,
- $28\sqrt[3]{n} + 10$ for membership in H and G ,
- $2\sqrt[3]{n} - 2$ for $[C']_2$,
- 2 for $[\psi']_2$,
- $4\sqrt[3]{n}$ for proof π_{bits} ,
- 2 for π_h ,

which adds up to $34\sqrt[3]{n} + 18$. Additionally, it requires $\sqrt[3]{n}$ elements of \mathbb{Z}_q .

Signature Time: The number of exponentiations computed by the prover is:

- 5 for the Boneh-Boyer signature and its commitment,
- 8 for proof that the Boneh-Boyer signature is correct,
- 4 for computing $[g]_1$ and $[h]_1$,
- $52\sqrt[3]{n} + 36$ for membership in G and H ,
- $8\sqrt[3]{n}$ for $[A']_1$ and $[C']_2$,
- $4\sqrt[3]{n} + 6$ for $[\psi]_2$ and $[\omega]_1$,
- $8\sqrt[3]{n}$ for proof π_Q ,
- 6 for proof π_h ,

which adds up to $72\sqrt[3]{n} + 61$.

Verification Time: The number of paring operations for verifying a signature is:

- 24 for verifying the proof that the Boneh-Boyer signature is correct,
 - $8n^{2/3} + 96\sqrt[3]{n} + 56$ for membership in H and G ,
 - $24\sqrt[3]{n}$ for proof π_Q ,
 - 8 for proof π_h ,
 - $2\sqrt[3]{n} + 12$ for $g_{A'}([C]_2, [\psi']_2, [\omega']_1) = [g]_1[w_1^\top]_2$,
- which adds up to $8n^{2/3} + 122\sqrt[3]{n} + 94$.

D The Ring Signatures of Bose et al. and of Gritti et al.^{4,5}

Bose et al. claim to construct a constant-size ring signature in the standard model [7]. However, we believe its security can not correctly be assessed. Our first observation is that they use a computational assumption, the SQROOT assumption, in the “exponent” of elements of a composite order bilinear group. More formally, they assume that, given composite order bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order N , there are hard problems in the set of quadratic residues modulo N , $Q_N \subset \mathbb{Z}_N$. Although, not necessarily false, this is at least odd as, similarly, one might assume the hardness of DDH in the set of quadratic residues of \mathbb{Z}_q when $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are bilinear groups of prime order order p . One might expect that, at least, both N and p must be quite large integers.

Even assuming the security of SQROOT, is still difficult to asses the security of their protocol from the security proof they provide. Specifically, on page 18 they construct an EUF-CMA adversary \mathcal{A}_3 which “hands over the tuple $(rParam, \{PK_1, PK_2, \dots, PK_{i^*}, \dots, PK_k\})$ to ring adversary \mathcal{D} ”. However, \mathcal{A}_3 must also hand the extended public keys (i.e. those containing the squares of the secret keys). Adversary \mathcal{A}_3 clearly can do this for parties other than P_{i^*} , while this is not true for P_{i^*} since sk_{i^*} is not known. In general, the problem is that the underlying signature scheme might not be secure if the integer $sk_{i,j}^2$ is public, as done on page 9 in the description of the **RKeyGen** algorithm. Note that one can compute forgeries in \mathbb{G}_2 for the full BB, given a^2, b^2 one can

⁴Both flaws were communicated to the respective authors.

⁵In this section we use multiplicative notation for the group operations to keep the expressions as they appear in the original works.

compute $(g_2^a g_2^b)^{1/(a^2-b^2)} = g_2^{1/(a-b)}$ which is a signature (in \mathbb{G}_2) for message 0 and $r = -1$. Although we don't know if actual forgeries exist, the important conclusion is that unforgeability is not guaranteed.

It seems that they explicitly tried to solve this issue by considering adversaries \mathcal{A}_4 and \mathcal{A}_5 “attempting to recover secret keys from the knowledge of public keys”. They discard such adversaries because \mathcal{A}_4 should solve SQROOT (why not also \mathcal{A}_5 ? the descriptions of forgery type IV and V are equal) and for \mathcal{A}_5 they say “We remark that such an adversary are no stronger than \mathcal{A}_3 type of adversary and advantage due to it is encompassed by $\text{Adv}_{\text{RSig}, \mathcal{A}_3}^{U^n, \text{forg}}(\delta)$ ”. However, this is not addressing the real problem: an adversary might construct a forgery without knowing sk but only sk^2 .

Gritti et al. claim to construct a logarithmic ring signature in the standard model [18]. However, their signatures are in fact of linear size as explained below. In page 12, Gritti et al. define $v_{b_i} := v_{b_1 \dots b_i^*}$, where $b_1 \dots b_i^*$ is the set of all bit-strings of size $d := \log n$ whose prefix is $b_1 \dots b_i$. From this, one has to conclude that v_{b_i} is a set (or vector) of group elements of size 2^{d-i} . In the same page they define the commitment $D_{b_i} := v_{b_i} h^{s_{b_i}}$, for random $s_{b_i} \in \mathbb{Z}_q$, which, according to the previous observation, is the multiplication of a set (or vector) of group elements with a group element. Given that length reducing group to group commitments are known to not exist [1], its representation requires at least 2^{d-i} group elements.⁶ Since commitments D_{b_0}, \dots, D_{b_d} are part of the signature, the actual signature size is $\Theta(2^d) = \Theta(n)$, rather than $\Theta(d) = \Theta(\log n)$ as claimed by Gritti et al.

E The Erasures Assumption in Ring Signatures

In the security proof of our PPA-based ring signature we need to embed a random preimage $A = \{\mathbf{a}_1, \dots, \mathbf{a}_{q_{\text{gen}}}\}$ of h in the verification keys, where q_{gen} is the total number of verification keys. On the other hand, the adversary may adaptively corrupt parties obtaining all the random coins used to generate the verification key. That is, we need to reveal $\log_{\mathcal{P}_S} \mathbf{a}_i$ (the discrete logs of \mathbf{a}_i) to the adversary, which is incompatible with the permutation pairing assumption and thus with the security of h . Since is not clear how to obliviously sample $(a_i \mathcal{P}, a_i^2 \mathcal{P})$ and we can only guess the set of corrupted parties with negligible probability, we are forced to use erasures: after sampling a_i and computing \mathbf{a}_i , the key generation algorithm erases a_i and a_i^2 .

Erasures were considered by Bender et al. [4] but only with respect to anonymity. Our signature achieves the stronger notion of perfect anonymity of Chandran et al. [8], meaning that, information theoretically, there is nothing in the signature that binds a signer to a signature. Since the random coins of our key generation algorithm are completely determined by the public key, in the in-

⁶In fact, there exists length reducing group to group commitments [2] with a weaker binding property, but is far from clear how to use these commitments in the Gritti et al.'s work

formation theoretic setting it is irrelevant if parties erase or not part of their random coins.

On the other hand, erasures in the unforgeability experiment seems to have not received much attention. In fact, the definition of Bender et al. doesn't prevent erasures, since, after corrupting a party, the adversary receives only the secret key but not the random coins used by the key generation algorithm. Chandran et al.'s unforgeability definition explicitly includes the random coins in the adversary view, preventing any erasure. However, this is not discussed and further, erasures are not even mentioned in their work.

We would also like to point out that other schemes may also require erasures. This is the case Malavolta et al.'s CRS-less ring signature. In order to get rid of the CRS, which is a pair of Groth-Sahai commitment keys, each party appends its own Groth-Sahai commitment keys to its public key. The signer and the verifier combines all the commitment keys by simply adding them and, as long as at least one verification key was honestly generated, the combined commitment keys are correctly distributed.

Nevertheless, when proving unforgeability one needs to move from perfectly hiding commitment keys to perfectly binding commitment keys. This implies that the reduction must change itself the verification keys of all the users to perfectly binding ones. In the perfectly hiding setting commitment keys are chosen as $\mathbf{u}_1 = \lambda \mathbf{u}_2$ and $\mathbf{u}_2 = (a\mathcal{P}, \mathcal{P})^\top$, for a random λ and random a . On the other hand, perfectly binding commitments keys have the only difference that $\mathbf{u}_1 = (\mathcal{P}, 0)^\top + \lambda \mathbf{u}_2$. An adversary that dynamically corrupts parties will eventually gets access to λ and a , which clearly allows him to detect any change on the commitment keys. We believe is an interesting open question if this problem can be fixed.

F Getting rid of the CRS

Malavolta et al. showed how to get rid of the CRS distributing it among the users public keys [26]. To eliminate the CRS, which is a pair of Groth-Sahai commitment keys, each party appends its own Groth-Sahai commitment keys to its public key. The signer and the verifier combines all the commitment keys by simply adding them and, as long as at least one verification key was honestly generated, the combined commitment keys are correctly distributed. We can easily apply this approach to our construction. That is, each participant's verification is appended with $[\mathbf{u}_{i,1}]_1, [\mathbf{u}_{i,2}], [\mathbf{v}_{i,1}]_2, [\mathbf{v}_{i,2}]$, perfectly hiding Groth-Sahai commitments keys, and the Groth-Sahai proofs are computed using the following commitment keys: $[\mathbf{u}_{i,j}]_1 := \sum_{i=1}^n [\mathbf{u}_{i,j}]_1, [\mathbf{v}_{i,j}]_2 := \sum_{i=1}^n [\mathbf{v}_{i,j}]_2$, for $j = 1, 2$ and the ring of verification keys is $\{\mathbf{vk}_1, \dots, \mathbf{vk}_n\}$.

Nevertheless, as noted on Section E, this approach requires erasures. Indeed, when proving unforgeability one needs to move from perfectly hiding commitment keys to perfectly binding commitment keys. This implies that the reduction must change itself the verification keys of all the users to perfectly binding ones (as done by adversary A_h on Lemma 2). In the perfectly hiding setting commit-

ment keys are chosen as $\mathbf{u}_1 = \lambda \mathbf{u}_2$ and $\mathbf{u}_2 = (a, 1)^\top$, for a random λ and random a . On the other hand, perfectly binding commitments keys have the only difference that $\mathbf{u}_1 = (1, 0)^\top + \lambda \mathbf{u}_2$. An adversary that dynamically corrupts parties will eventually get access to λ and a , which clearly allows him to detect any change on the commitment keys. Hence, also these values must be erased after the verification key is computed.