Sub-linear Blind Ring Signatures without Random Oracles

Essam Ghadafi

Dept. Computer Science,
University of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB.
United Kingdom.
ghadafi@cs.bris.ac.uk

Abstract. Ring signatures allow a signer to anonymously sign a message on behalf of a set of arbitrarily chosen signers called a "ring". Blind signatures, on the other hand, allow a user to obtain a signature on a message while maintaining the privacy of the message. Blind ring signatures combine properties of both primitives and hence provide a strong notion of anonymity where the privacy of both the identity of the signer and the message is preserved. Blind ring signatures find applications in various systems; including multi-authority e-voting and distributed e-cash systems.

In this paper we provide the first provably secure blind ring signature construction that does not rely on random oracles, which solves an open problem raised by Herranz and Laguillaumie at ISC 2006. We present different instantiations all of which are round-optimal (i.e. have a two-move signing protocol), yield sub-linear size signatures, and meet strong security requirements. In order to realize our constructions efficiently, we construct a sub-linear size set membership proof which works in the different bilinear group settings, which may be of independent interest.

As a secondary contribution, we show how to generically combine our set membership proof with any secure signature scheme meeting some conditions to obtain ring signatures whose security does not rely on random oracles. All our constructions work over the efficient prime-order bilinear group setting and yield signatures of sub-linear size. In addition, our constructions meet strong security requirements: namely, anonymity holds under full key exposure and unforgeability holds against insider-corruption. Finally, we provide some example instantiations of the generic construction.

1 Introduction

Background. A Ring Signature (RS), introduced by Rivest, Shamir and Tauman [48], allows a signer to choose an arbitrary set of signers called a "ring" and anonymously sign a message on behalf of the ring providing that the signer himself is a member of the ring. Generating the signature does not require the cooperation of other members of the ring and hence they need not even be aware of their inclusion in the ring.

Besides correctness, the security of ring signatures [48, 9] requires anonymity and unforgeability. Informally, anonymity requires that a signature does not reveal the identity of the ring member who produced it. On the other hand, unforgeability requires that an adversary cannot forge new signatures on behalf of an honest ring. In [9], the authors provide various variants of those requirements. We will prove the security of our constructions under the strongest definitions provided in [9].

Like group blind signatures [41], blind ring signatures extend blind signatures to the multi-signer setting. However, in contrast to the former, the latter provide more flexibility in the choice of the group as it is done in an ad hoc manner without requiring prior cooperation or join protocols. In addition, anonymity of the signer is not revocable. Besides the three security properties required from traditional ring signatures, the security of blind ring signatures requires blindness which informally states that members of the ring cannot learn which message is being signed on behalf of the ring. Also, they cannot match a signature to its signing session.

Applications. For applications of ring signatures see, e.g. [48, 42, 24]. They were originally used for anonymous leaking of authoritative secrets. Another application of ring signatures are designated-verifier signatures [39].

Applications of blind ring signatures include distributed e-cash systems [41], where a client's e-coin is signed by a member of a coalition of banks chosen in an ad hoc manner. The choice of the coalition could be specified by either the issuing bank or the client himself. Other applications of the primitive include multi-authority e-voting, e.g. [22], and e-auction systems.

Related Work. BLIND RING SIGNATURES. Only a few blind ring signature schemes [18, 53, 37, 54] were proposed. All of those constructions are secure in the Random Oracle Model (ROM) [8]. The scheme in [18] yields signatures of linear size and its security requires both random oracles and the generic group model [51]. In [53], the authors presented a static blind ring signature scheme that requires both random oracles and the generic group model. This scheme requires that the group (i.e. the ring) is fixed and hence it yields signatures of constant size. The schemes in [37, 54] also yield signatures of linear size. We note here that the blindness requirement of [37] was proven using a different game than the standard definition for blindness [40, 46] where the adversary only interacts once with the challenger and does not get to see the final signature.

RING SIGNATURES. The original construction by Rivest, Shamir and Tauman [48] is based on trapdoor permutations and is secure in the random oracle model. Subsequently, other constructions relying on random oracles followed [5, 13, 38, 24, 43].

A few constructions which do not rely on random oracles were proposed. Bender et al. [9] gave a generic construction requiring generic ZAPs [25], making it inefficient. They also gave two constructions for two-signer rings. Other constructions which do not require random oracles include [50, 19, 15, 49, 16]. The constructions in [49, 16] use a weaker notion of unforgeability than the one we use in this paper.

All existing constructions apart from [19] (which yields sub-linear size signatures in composite-order groups in the Common Reference String (CRS) model) and [24, 43] (which yield signatures of constant size in the ROM) yield signatures of linear size.

Motivation. All the existing blind ring signature schemes [18, 53, 37, 54] require random oracles. Since the random oracle is an assumption that cannot be realized securely in practice [17], many protocol designers strive for constructions in the standard model. Moreover, all existing constructions (apart from [53] in which the ring is static) yield signatures whose size grows linearly with the size of the ring. It is therefore desirable to design more efficient schemes. Furthermore, it is nowadays a common practice to base the security of cryptographic protocols on non-interactive complexity assumptions that are falsifiable [47].

In addition, most of the existing ring signature constructions which do not rely on random oracles are either specific to the (inefficient) composite-order bilinear group setting and/or yield signatures of linear size.

The Challenges. The subtlety one faces when designing blind ring signatures lies in the dual privacy requirement: that is the dilemma of having parts of the witness of the same proof of knowledge coming from different parties who do not trust each other. On the one hand, the signer needs to hide his identity and parts of the signature that could identify him (i.e. the anonymity requirement). On the other hand, the user needs to hide the message and parts of the signature which could reveal the linkage between a signature and its signing session (i.e. the blindness requirement). One might consider addressing such an issue by resorting to secure multiparty computation, however, such an approach would massively degrade the efficiency of the resulting construction.

Due to the nature of random oracles, in the random oracle model this obstacle is easier to tackle by, for instance, using divertible proofs of knowledge e.g. [23, 44]. In the standard model, the issue is more subtle. To get around this issue, we exploit some properties of Groth-Sahai proofs [35], namely: the randomizability of the proofs [7] and the ability to transform some proofs without knowledge of the original witness [32, 30]. This way we obtain the required divertibility needed to achieve the dual privacy requirement.

The second technical challenge is that unforgeability of ring-related signatures requires that the signature is bound to the ring in order to prevent the adversary from transforming a signature by some ring into a signature by a different ring. The scenario is more serious when the construction involves a malleable proof system such as the Groth-Sahai proof system which we use in our constructions. This is because the malleability of the proof system allows one to easily transform a proof for some statement into another proof for a related statement.

When constructing ring signatures, one can easily bind the signature to the ring by simply signing both the message and the ring. For instance, this could be efficiently achieved by signing the hash of the concatenation of both the message and the ring. Unfortunately, this approach does not work in a blind signing protocol. That is because necessitating that the message remains hidden from the signer, one needs to prove that such hashing was applied correctly without revealing the message, which cannot be efficiently realized due to the complex structure of hash functions. To bind the signature to the ring w.r.t. which it was produced, we deploy a different approach. We use a signature scheme that simultaneously signs a pair of messages to construct a partially-blind signature scheme where we hide the actual message from the signer but we include the details of the ring as the public information shared between the user and the signer.

The remaining challenge which is inherent even in traditional ring signatures is the size of the signatures. Almost all previous blind ring signatures e.g. [18, 37, 54] and most existing traditional ring signatures e.g. [9, 50, 15] yield signatures whose size grows linearly with the size of the ring. This limitation is usually inherited from the underlying OR proof used to prove that the signature verifies w.r.t. a verification key contained in the ring without revealing which one it is. In [19], the authors used some techniques from private information retrieval applications to construct a membership proof that has a sub-linear size. Unfortunately, their protocol is limited to the rather inefficient composite-order bilinear group setting. As a part of our contribution, we adapt their technique to the prime-order setting and thus we construct a sub-linear size set membership proof that works in the 3 different settings of prime-order bilinear groups. Although this on its own is not a major contribution, it is of independent interest as we believe it could have further applications beyond the scope of this paper.

Our Contribution. Our main contribution is the first blind ring signature schemes that do not rely on idealized assumptions. This solves a problem that remained open since 2006 [37]. To realize our constructions efficiently, we instantiate the idea used for the membership proof from [19] in the prime-order bilinear group setting. All our constructions yield signatures of sub-linear size and thus are shorter than those of previous constructions. In addition, our schemes meet stronger security requirements than those used for previous constructions, and their security is based solely on falsifiable complexity assumptions. We note here that it is still an open problem to get a ring signature scheme with constant-size signatures in the standard model and hence we believe that our constructions are efficient.

Our final contribution is a generic construction that combines our set membership proof with any signature scheme in the standard model satisfying some properties to get sub-linear size ring signatures without random oracles. Again, our focus is on constructions in the efficient prime-order bilinear group setting.

Paper Organization. The rest of the paper is organized as follows: In Section 2, we give some preliminary definitions. In Section 3, we define blind ring signatures and present their security definitions. In Section 4, we present a new set membership proof. In Section 5 we present our blind ring signature constructions. Finally, in Section 6 we present our ring signature constructions.

2 Preliminaries

Notation. Given a probability distribution S, we denote by $x \leftarrow S$ the operation of selecting an element according to S. If A is a probabilistic machine, we denote by $A(x_1, \ldots, x_n)$ the output distribution of A on inputs $(x_1, \ldots, x_n)$. By p.p.t., we mean running in probabilistic polynomial time in the relevant

security parameter. By $[1, n]$, we denote the set $\{1, 2, \ldots, n\}$. A function $v(.) : \mathbb{N} \rightarrow \mathbb{R}^+$ is negligible in c if for every polynomial $p(.)$ and all sufficiently large values of c, it holds that $v(c) < \frac{1}{p(c)}$.

2.1 Bilinear Groups

A bilinear group is a tuple $\mathcal{P} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, G, \tilde{G}, e)$ where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of a prime order p and G and $\tilde{G}$ generate $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The function e is a non-degenerate bilinear map $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. We use multiplicative notation for all the groups although usually $\mathbb{G}_1$ and $\mathbb{G}_2$ are chosen to be additive. We let $\mathbb{G}_1^\times := \mathbb{G}_1 \setminus \{1_{\mathbb{G}_1}\}$ and $\mathbb{G}_2^\times := \mathbb{G}_2 \setminus \{1_{\mathbb{G}_2}\}$. For clarity, elements from $\mathbb{G}_2$ will be accented with $\tilde{}$.

Following [31], we categorize prime-order bilinear groups into three main types:

- **Type-1**: This is the symmetric pairing setting in which $\mathbb{G}_1 = \mathbb{G}_2$.
- **Type-2**: $\mathbb{G}_1 \neq \mathbb{G}_2$ but there is an efficiently computable isomorphism $\psi : \mathbb{G}_2 \longrightarrow \mathbb{G}_1$.
- **Type-3**: Again $\mathbb{G}_1 \neq \mathbb{G}_2$, but now there is no known efficiently computable isomorphism.

We assume that all three groups are cyclic and that there is an algorithm $\mathsf{BGrpSetup}$ that takes as input a security parameter λ and a type $\mathsf{tp} \in \{1, 2, 3\}$ and outputs a description of bilinear groups of Type-tp.

2.2 Complexity Assumptions

We will use the following assumptions from the literature:

CDH. For a group $\mathbb{G} := \langle G \rangle$ of a prime order p given $(G, G^a, G^b) \in \mathbb{G}^3$ for $a, b \leftarrow \mathbb{Z}_p$, it is hard to compute G^{ab}.

DDH. For a group $\mathbb{G} := \langle G \rangle$ of a prime order p given $(G, G^a, G^b, C) \in \mathbb{G}^4$ for $a, b \leftarrow \mathbb{Z}_p$, it is hard to decide whether or not $C = G^{ab}$.

Co-CDH [20]**.** In Type-2 bilinear groups given $(G, G^a, \tilde{G}, \tilde{G}^b) \in \mathbb{G}_1^2 \times \mathbb{G}_2^2$ for $a, b \leftarrow \mathbb{Z}_p$, it is hard to compute G^{ab}.

Co-CDH* [20]**.** In Type-3 bilinear groups given $(G, G^a, G^b, \tilde{G}, \tilde{G}^b) \in \mathbb{G}_1^3 \times \mathbb{G}_2^2$ for $a, b \leftarrow \mathbb{Z}_p$, it is hard to compute G^{ab}.

SXDH. The Decisional Diffie-Hellman (DDH) assumption holds in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$.

DLIN [12]**.** For Type-1 bilinear groups where $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and G generates $\mathbb{G}$, given the tuple $(G^a, G^b, G^{ra}, G^{sb}, G^t)$ where $a, b, r, s, t \in \mathbb{Z}_p$ are unknown, it is hard to tell whether $t = r + s$ or t is random.

q-SDH [11]**.** For a group $\mathbb{G} := \langle G \rangle$ of a prime order p given $(G, G^x, \ldots, G^{x^q}) \in \mathbb{G}^{q+1}$ for $x \leftarrow \mathbb{Z}_p$, it is hard to output a pair $(c, G^{\frac{1}{x+c}}) \in \mathbb{Z}_p \times \mathbb{G}$ for an arbitrary $c \in \mathbb{Z}_p \backslash \{-x\}$.

WFCDH [29]**.** In symmetric bilinear groups, given $(G, G^a, G^b) \in \mathbb{G}^3$ for $a, b \leftarrow \mathbb{Z}_p$, it is hard to output a tuple $(G^r, G^{ra}, G^{rb}, G^{rab}) \in \mathbb{G}^{\times 4}$ for an arbitrary $r \in \mathbb{Z}_p$.

AWFCDH [29]**.** In asymmetric bilinear groups, given $(G, G^a, \tilde{G}) \in {\mathbb{G}_1}^2 \times \mathbb{G}_2$ for $a \leftarrow \mathbb{Z}_p$, it is hard to output a tuple $(G^b, G^{ab}, \tilde{G}^b, \tilde{G}^{ab}) \in {\mathbb{G}_1^\times}^2 \times {\mathbb{G}_2^\times}^2$ for an arbitrary $b \in \mathbb{Z}_p$.

q-DHSDH [29]**.** In symmetric bilinear groups, given $(G, H, K, G^x) \in \mathbb{G}^4$ for $x \leftarrow \mathbb{Z}_p$, and $q-1$ tuples $(W_i := (K \cdot G^{u_i})^{\frac{1}{x+v_i}}, U_{1,i} := G^{u_i}, U_{2,i} := H^{u_i}, V_{1,i} := G^{v_i}, V_{2,i} := H^{v_i})_{i=1}^{q-1}$, where $u_i, v_i \leftarrow \mathbb{Z}_p$, it is hard to output a new tuple $(W^*, U_1^*, U_2^*, V_1^*, V_2^*)$ of this form.

q-ADHSDH [29]**.** In asymmetric bilinear groups, given $(G, F, K, G^x, \tilde{G}, \tilde{G}^x) \in \mathbb{G}_1^4 \times \mathbb{G}_2^2$ for $x \leftarrow \mathbb{Z}_p$, and $q-1$ tuples $(W_i := (K \cdot G^{u_i})^{\frac{1}{x+v_i}}, U_{1,i} := G^{u_i}, \tilde{U}_{2,i} := \tilde{G}^{u_i}, V_{1,i} := F^{v_i}, \tilde{V}_{2,i} := \tilde{G}^{v_i})_{i=1}^{q-1}$ for $u_i, v_i \leftarrow \mathbb{Z}_p$, it is hard to output a new tuple $(W^*, U_1^*, {\tilde{U}_2}^*, V_1^*, {\tilde{V}_2}^*)$ of this form.

2.3 Groth-Sahai (GS) Proofs

Groth and Sahai [35, 36] introduced a proof system in the CRS model that yields Non-Interactive Witness-Indistinguishable (NIWI) and Zero-Knowledge (NIZK) proofs. The system can be instantiated in composite-order or prime-order bilinear groups. The equations one can prove with the system are as follows where in the description $X_1, \ldots, X_m, Y_1, \ldots, Y_n \in \mathbb{G}$, $x_1, \ldots, x_m, y_1, \ldots, y_n \in \mathbb{Z}_p$ are secret variables (hence underlined), whereas $A_i, T \in \mathbb{G}$, $a_i, b_i, k_{i,j}, t \in \mathbb{Z}_p$, $t_T \in \mathbb{G}_T$ are public constants. Note that in the asymmetric setting, there are two types of MSM equations depending on which group the elements belong to.

- **Pairing Product Equation (PPE):** $\prod_{i=1}^{n} e(A_i, \underline{Y_i}) \cdot \prod_{i=1}^{m} \prod_{j=1}^{n} e(\underline{X_i}, \underline{Y_j})^{k_{i,j}} = t_T \cdot$
- **Multi-Scalar Multiplication Equation (MSME):** $\prod_{i=1}^{n} A_i^{\underline{y_i}} \prod_{i=1}^{m} \underline{X_i}^{b_i} \prod_{i=1}^{m} \prod_{j=1}^{n} \underline{X_i}^{k_{i,j}\underline{y_j}} = T \cdot$
- **Quadratic Equation (QE) in $\mathbb{Z}_p$:** $\sum_{i=1}^{n} a_i \underline{y_i} + \sum_{i=1}^{m} \underline{x_i} b_i + \sum_{i=1}^{m} \sum_{j=1}^{n} \underline{x_i}\underline{y_j} = t \cdot$

The system consists of the following algorithms

$$(\mathsf{GSSetup}, \mathsf{GSProve}, \mathsf{GSVerify}, \mathsf{GSExtract}, \mathsf{GSSimSetup}, \mathsf{GSSimProve}) \cdot$$

Algorithm GSSetup takes as input the description of a bilinear group $\mathcal{P}$ and outputs a *soundness* reference string crs and an extraction key xk. GSProve takes as input a reference string crs, a witness and a set of equations, and outputs a proof Ω for the satisfiability of the equations. For clarity, we will underline the elements of the witness in the description of the equations. GSVerify takes as input a reference string crs, a proof Ω and a set of equations, and outputs 1 if the proof is valid or 0 otherwise. In the rest of the paper we will omit the set of equations from the input to the GSVerify algorithm.

GSExtract takes as input a soundness reference string crs, the extraction key xk and a valid proof Ω, and outputs the witness used in the proof. GSSimSetup takes as input a bilinear group $\mathcal{P}$ and outputs a *simulation* string $\mathsf{crs}_{\mathsf{Sim}}$ and a trapdoor key tr that allows to simulate proofs. GSSimProve takes as input $\mathsf{crs}_{\mathsf{Sim}}$ and the simulation trapdoor tr and produces a simulated proof Ω_{Sim} without a witness.

The system works by first committing to the elements of the witness (using the algorithm GSCommit) and then producing a proof of satisfiability for each equation. If a witness component is involved in more than one source equation, the same commitment is re-used when verifying the proofs which makes the proofs correlated.

The proofs come in two flavors: the soundness setting yields extractable proofs, whereas the simulation setting yields simulatable proofs. The system's security requires that the distributions of strings crs and $\mathsf{crs}_{\mathsf{Sim}}$ are indistinguishable and simulated proofs are indistinguishable from real proofs.

As formalized by [7], GS proofs can be rerandomized by rerandomizing the associated GS commitments and updating the proofs accordingly so that we obtain fresh proofs that are unlinkable to the original ones. Rerandomizing a proof requires knowledge of neither the witness nor the associated randomness used in the original GS commitments. We define an algorithm GSRandomize which takes as input a CRS crs and a proof Ω, and outputs a proof Ω' which is a randomized version of the proof Ω.

The proof system has the following properties:

- **Perfect Completeness:** On a correctly generated CRS, crs, and a valid proof Ω, the algorithm GSVerify always accepts the proof.
- **Perfect Soundness:** On a correctly generated CRS, crs, it is impossible to generate a proof unless the equations are satisfiable (i.e. unless the prover knows a witness).
- **Composable Witness-Indistinguishability:** The CRS crs output by GSSetup is computationally indistinguishable from the CRS $\mathsf{crs}_{\mathsf{Sim}}$ output by GSSimSetup. Also, for any p.p.t. adversary $\mathcal{A}$ that

is given $\mathsf{crs}_{\mathrm{Sim}}$ and is allowed to choose any statement y and two distinct witnesses w_0 and w_1 for the statement y, the following probability is negligibly close to 1/2.

$$\Pr\left[b \leftarrow \{0,1\}; \Omega \leftarrow \mathsf{GSProve}(\mathsf{crs}_{\mathrm{Sim}}, w_b, y); b' \leftarrow \mathcal{A}(\Omega) : b = b' \ \wedge \ (w_i, y) \in R \text{ for } i = 0,1\right].$$

– **Composable Zero-Knowledge:** Again, the CRS crs output by $\mathsf{GSSetup}$ is computationally indistinguishable from the CRS $\mathsf{crs}_{\mathrm{Sim}}$ output by $\mathsf{GSSimSetup}$. In addition, we have for any p.p.t. adversary $\mathcal{A}$ that is given $(\mathsf{crs}_{\mathrm{Sim}}, \mathsf{tr})$ and is allowed to choose any $(w, y) \in R$ that

$$\begin{aligned}&\Pr\left[\Omega \leftarrow \mathsf{GSProve}(\mathsf{crs}_{\mathrm{Sim}}, w, y) : \mathcal{A}(\Omega) = 1\right] \\ &= \Pr\left[\Omega_{\mathrm{Sim}} \leftarrow \mathsf{GSSimProve}(\mathsf{crs}_{\mathrm{Sim}}, \mathsf{tr}, y) : \mathcal{A}(\Omega_{\mathrm{Sim}}) = 1\right].\end{aligned}$$

– **Rerandomizability [7]:** We have for all p.p.t. adversaries $(\mathcal{A}_1, \mathcal{A}_2)$, the following probability is negligbly close to 1/2

$$\Pr\left[\begin{array}{l}(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P}); (w, y, \Omega, \mathsf{state}_{\mathsf{find}}) \leftarrow \mathcal{A}_1(\mathsf{crs}); \\ \Omega_0 \leftarrow \mathsf{GSProve}(\mathsf{crs}, w, y); \Omega_1 \leftarrow \mathsf{GSRandomize}(\mathsf{crs}, \Omega); b \leftarrow \{0,1\}; \\ b^* \leftarrow \mathcal{A}_2(\mathsf{state}_{\mathsf{find}}, \Omega_b) : b^* = b \ \wedge \ \mathsf{GSVerify}(\mathsf{crs}, \Omega) = 1 \ \wedge \ (w, y) \in R\end{array}\right].$$

As shown in [7], GS proofs have composable randomizability, where randomizability holds even if we switch to the simulation setting.

For more details on how the proof system can be instantiated in the different settings see [36, 34].

2.4 (Partially) Blind Signatures

Blind Signatures (BS) [21] allow a user to obtain a signature on a message hidden from the signer. Partially Blind Signatures (PBS) [3] are an extension of blind signatures where unlike blind signatures, part of the message to be signed is shared public information info which is known to both parties.

The signing protocol in these schemes is interactive $\langle \mathsf{PBSObtain}(\mathsf{pk}, m, \mathsf{info}), \mathsf{PBSSign}(\mathsf{sk}, \mathsf{info})\rangle$ between a user who knows a message m and a signer who possesses a secret signing key sk and both parties know the public information info. If the protocol is completed successfully, the user obtains a signature Σ on the message m and the information info.

The security of partially blind signatures [6] is similar to that of blind signatures [40, 46] and consists besides correctness of blindness and unforgeability. Intuitively, blindness requires that an adversarial signer does not learn the message being signed and he cannot match a signature to its signing session. In the game, the adversary (modeling an adversarial signer) chooses two messages m_0 and m_1 and common information info and then interacts with the honest user who requests signatures on those messages in an arbitrary order unknown to the adversary. The same information info is used in both interactions. If completed successfully, the adversary gets the two final signatures and wins if it tells the order in which the messages were signed with a probability that is non-negligibly greater than $1/2$.

On the other hand, unforgeability deals with an adversarial user whose goal is to obtain $k+1$ distinct message/signature pairs after only k interactions w.r.t. the public information info with the honest signer.

A Partially Blind Signature Scheme [29, 30]. In [29, 2], the authors gave a blind signature scheme whose security reduces to the DLIN, WFCDH and q-DHSDH/q-ADHSDH assumptions in the symmetric setting or the SXDH, AWFCDH and q-ADHSDH assumptions in the asymmetric setting. The message space of the scheme is $\mathcal{M} := \{(G^m, \tilde{G}^m) | m \in \mathbb{Z}_p\}$. To get a partially blind scheme, we use a variant of their blind scheme based on the modified signature scheme from [30] whose message space is $\mathcal{M} \times \mathbb{Z}_p$ as highlighted in [30].

The high-level idea behind the scheme is that the user commits to his message and sends the commitment along with GS proofs to prove that it is well-formed to the signer. The signer uses his secret key

to produce a signature on the commitment and the public information info. When the user receives the signature, he uses the randomness he used in the commitment to modify the signature from one on the commitment to one on the message itself. The final signature is a set of GS proofs of knowledge of such a signature. The blindness of the scheme is ensured by the NIWI/NIZK properties of GS proofs and the fact that the first-round commitment is information-theoretically hiding. The scheme in the asymmetric setting is in given Figure 1.

$\mathsf{PBSSetup}(1^\lambda)$:	$\mathsf{PBSKeyGen}(\mathsf{param}_{\mathsf{PBS}})$:
$-\ (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, G, \tilde{G}, e) \leftarrow \mathsf{BGrpSetup}(1^\lambda, 3)$.	$-$ Choose $a \leftarrow \mathbb{Z}_p$ and set $A := G^a$ and $\tilde{A} := \tilde{G}^a$.
$-\ \mathcal{P} := (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, G, \tilde{G}, e)$.	$-\ \mathsf{sk} := a$, $\mathsf{pk} := (A, \tilde{A})$. Return $(\mathsf{sk}, \mathsf{pk})$.
$-\ (\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P})$.	$\mathsf{PBSVerify}(\mathsf{pk}, (M, \tilde{M}), \Sigma, \mathsf{info})$:
$-\ F, K, L, T \leftarrow \mathbb{G}_1$.	$-$ Parse Σ as Ω_{sig}.
$-$ Return $\mathsf{param}_{\mathsf{PBS}} := (\mathcal{P}, \mathsf{crs}, F, K, L, T)$.	$-$ Return 1 if $\mathsf{GSVerify}(\mathsf{crs}, \Omega_{\mathsf{sig}}) = 1$ Else Return 0.

The signing protocol $\langle \mathsf{PBSObtain}(\mathsf{pk}, (M, \tilde{M}), \mathsf{info}), \mathsf{PBSSign}(\mathsf{sk}, \mathsf{info}) \rangle$

$\underline{\mathsf{PBSObtain} \rightarrow \mathsf{PBSSign}}$

Choose $s \leftarrow \mathbb{Z}_p$ and compute $S := G^s$, $\tilde{S} := \tilde{G}^s$ and $C := M \cdot T^s$.

$-\ \Omega \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (M, \tilde{M}, S, \tilde{S}), \Big\{ e(\underline{M}, \tilde{G}) = e(G, \underline{\tilde{M}}) \ \wedge\ e(\underline{S}, \tilde{G}) = e(G, \underline{\tilde{S}}) \ \wedge\ e(\underline{M}, \tilde{G}) e(T, \underline{\tilde{S}}) = e(C, \tilde{G}) \Big\}\Big)$.

Send (C, Ω) to $\mathsf{PBSSign}$.

$\underline{\mathsf{PBSSign} \rightarrow \mathsf{PBSObtain}}$

Abort if $\mathsf{GSVerify}(\mathsf{crs}, \Omega) \neq 1$. Otherwise, choose $u, v \leftarrow \mathbb{Z}_p$ and compute:

$U' := G^u$, $V := F^v$, $W := (K \cdot T^u \cdot C \cdot L^{\mathsf{info}})^{\frac{1}{a+v}}$, $\tilde{U}' := \tilde{G}^u$, $\tilde{V} := \tilde{G}^v$.

Send $\sigma := (W, U', \tilde{U}', V, \tilde{V})$ to $\mathsf{PBSObtain}$.

$\underline{\mathsf{PBSObtain}}$

Compute $U := U' \cdot S$ and $\tilde{U} := \tilde{U}' \cdot \tilde{S}$.

Abort if $e(U, \tilde{G}) \neq e(G, \tilde{U})$, $e(F, \tilde{V}) \neq e(V, \tilde{G})$ or $e(W, \tilde{A} \cdot \tilde{V}) \neq e(K \cdot M \cdot L^{\mathsf{info}}, \tilde{G}) e(T, \tilde{U})$.

$-\ \Omega_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (V, \tilde{V}, W, U, \tilde{U}), \Big\{ e(\underline{V}, \tilde{G}) = e(F, \underline{\tilde{V}}) \ \wedge\ e(\underline{U}, \tilde{G}) = e(G, \underline{\tilde{U}}) \ \wedge\ e(\underline{W}, \tilde{A} \cdot \underline{\tilde{V}}) e(T^{-1}, \underline{\tilde{U}}) = e(K \cdot M \cdot L^{\mathsf{info}}, \tilde{G}) \Big\}\Big)$.

Output $\Sigma := \Omega_{\mathsf{sig}}$.

Fig. 1. The partially blind signature scheme (in the asymmetric setting) [29, 30]

2.5 Ring Signatures

A ring signature is a tuple $\mathsf{RS} := (\mathsf{RSSetup}, \mathsf{RSKeyGen}, \mathsf{RSSign}, \mathsf{RSVerify})$ of p.p.t. algorithms. Those algorithms are defined as follows; where to aid notation all algorithms (bar RSSetup and RSKeyGen) take as implicit input $\mathsf{param}_{\mathsf{RS}}$ (output by RSSetup):

- $\mathsf{RSSetup}(1^\lambda)$ takes as input a security parameter λ and outputs common public parameters $\mathsf{param}_{\mathsf{RS}}$.
- $\mathsf{RSKeyGen}(\mathsf{param}_{\mathsf{RS}})$ takes as input $\mathsf{param}_{\mathsf{RS}}$ and outputs a pair of secret/public keys $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{RSSign}(\mathsf{sk}_i, m, \mathcal{R})$ takes as input a secret key sk_i, a message $m \in \mathcal{M}$ (where $\mathcal{M}$ is the message space) and a ring $\mathcal{R} := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$ with the condition that $\mathsf{pk}_i \in \mathcal{R}$ and outputs a signature Σ on the message m.
- $\mathsf{RSVerify}(m, \Sigma, \mathcal{R})$ takes as input a message m, a ring signature Σ and a ring $\mathcal{R}$ and outputs 1 if the signature is on the message m w.r.t. ring $\mathcal{R}$ or 0 otherwise.

The security properties required from ring signatures are informally as follows:

- **Correctness:** All honestly generated signatures are accepted by the RSVerify algorithm.
- **Anonymity:** An adversary cannot tell which ring member produced a signature.

– **Unforgeability:** An adversary cannot output a valid signature Σ^* on a message m^* and w.r.t. an honest ring $\mathcal{R}^*$ unless the adversary obtained such a signature by querying the sign oracle on $(m^*, \mathcal{R}^*)$.

For detailed definitions and variants of those properties, we refer the reader to [9]. We use the strongest variants from [9], namely: anonymity under full key exposure and unforgeability against insider-corruption.

3 Blind Ring Signatures

Definition 1 (Blind Ring Signatures). *A Blind Ring Signature (*BRS*) is a tuple of p.p.t. algorithms* $(\mathsf{BRSSetup}, \mathsf{BRSKeyGen}, \langle \mathsf{BRSObtain}, \mathsf{BRSSign} \rangle, \mathsf{BRSVerify})$*. Those algorithms are defined as follows; where to aid notation all algorithms (bar* $\mathsf{BRSSetup}$ *and* $\mathsf{BRSKeyGen}$*) take as implicit input* $\mathsf{param}_{\mathsf{BRS}}$ *(output by* $\mathsf{BRSSetup}$*):*

- $\mathsf{BRSSetup}(1^\lambda)$ *takes as input a security parameter* λ *and outputs public parameters* $\mathsf{param}_{\mathsf{BRS}}$.
- $\mathsf{BRSKeyGen}(\mathsf{param}_{\mathsf{BRS}})$ *is run by a signer* Signer_i *to generate his pair of secret/public keys* $(\mathsf{sk}, \mathsf{pk})$.
- $\langle \mathsf{BRSObtain}(m, \mathcal{R}), \mathsf{BRSSign}(\mathsf{sk}_i, \mathcal{R}) \rangle$ *is an interactive two-party protocol between a user* User *and a signer in the ring* $\mathcal{R}$ *where* $\mathsf{pk}_i \in \mathcal{R}$*. If the protocol completes successfully,* User *obtains a blind ring signature* Σ *on the message* m*. If any of the parties abort,* User *outputs* $\perp$*. This protocol is initiated by a call to* $\mathsf{BRSObtain}$*. The choice of the ring could be influenced by either the signer or the user.*
- $\mathsf{BRSVerify}(m, \Sigma, \mathcal{R})$ *verifies if the blind ring signature* Σ *is on the message* m *w.r.t. the ring* $\mathcal{R}$*.*

A tuple $\mathsf{BRS} := (\mathsf{BRSSetup}, \mathsf{BRSKeyGen}, \langle \mathsf{BRSObtain}, \mathsf{BRSSign} \rangle, \mathsf{BRSVerify})$ is a secure blind ring signature if it has correctness, anonymity, unforgeability and blindness which are defined as follows:

Definition 2 (Correctness). *A blind ring signature* BRS *is correct if for any* $\lambda \in \mathbb{N}$*, any polynomial* $n(\cdot)$*, any* $\{(\mathsf{pk}_i, \mathsf{sk}_i)\}_{i=1}^{n(\lambda)}$ *output by* $\mathsf{BRSKeyGen}$*, any message* m *in the message space* $\mathcal{M}$ *and any index* $i \in [1, n(\lambda)]$ *if* Σ *is the output of the honest interaction* $\langle \mathsf{BRSObtain}(m, \mathcal{R}), \mathsf{BRSSign}(\mathsf{sk}_i, \mathcal{R}) \rangle$ *where* $\mathcal{R} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_{n(\lambda)}\}$ *then* $\mathsf{BRSVerify}$ *accepts the signature* Σ*.*

ANONYMITY. We use a strong definition for anonymity where we allow the adversary to use corrupt keys as well as obtaining the secret keys for the two challenge signers i_0 and i_1 and hence capturing security against *full key exposure* and *adversarially-chosen keys* [9].

Definition 3 (Anonymity). *A blind ring signature* BRS *satisfies anonymity if for any* $\lambda \in \mathbb{N}$ *and polynomial* $n(\cdot)$ *the success probability of any p.p.t. adversary* $\mathcal{A}$ *in the following game is negligibly close to* $1/2$*:*

1. *The challenger generates* $\mathsf{param}_{\mathsf{BRS}}$ *and key pairs* $\{(\mathsf{pk}_i, \mathsf{sk}_i)\}_{i=1}^{n(\lambda)}$ *using* $\mathsf{BRSKeyGen}(\mathsf{param}_{\mathsf{BRS}})$*.* $\mathcal{A}$ *is given* $\mathsf{param}_{\mathsf{BRS}}$ *and* $\mathcal{S} := \{\mathsf{pk}_i\}_{i=1}^{n(\lambda)}$*.*
2. *Throughout the game,* $\mathcal{A}$ *has access to a sign oracle* OSign *with which it interacts to obtain signatures on messages and by signers in rings of its choice (providing that the signer's public key is in* $\mathcal{R}$ *and* $\mathcal{S}$*).* $\mathcal{A}$ *can also ask for the secret key of any signer to be revealed at any stage of the game.*
3. $\mathcal{A}$ *outputs two distinct indices* i_0 *and* i_1 *and a ring* $\mathcal{R}$ *with the only condition that* $\mathsf{pk}_{i_0}, \mathsf{pk}_{i_1} \in \mathcal{R}$*. It then interacts with the challenger to get a signature by signer* Signer_{i_b} *where* $b \leftarrow \{0, 1\}$*.*
4. $\mathcal{A}$ *outputs a bit* b' *and succeeds if* $b = b'$*.*

UNFORGEABILITY. Informally, a blind ring signature is unforgeable if the adversary cannot output a blind ring signature w.r.t. to a ring $\mathcal{R}$ of honest signers that was never produced by the sign oracle. Due to the blind nature of the signing protocol and as in standard blind signatures, we follow the $(k, k+1)$-unforgeability definition [40, 46]. The following definition also protects against insider corruption [9]:

Definition 4 (Unforgeability). *A blind ring signature* BRS *is unforgeable if for any* $\lambda \in \mathbb{N}$*, and polynomial* $n(\cdot)$ *the success probability of any p.p.t. adversary* $\mathcal{A}$ *in the following game is negligible:*

1. *The challenger generates* $\mathsf{param}_{\mathsf{BRS}}$ *and key pairs* $\{(\mathsf{pk}_i, \mathsf{sk}_i)\}_{i=1}^{n(\lambda)}$ *using* $\mathsf{BRSKeyGen}(\mathsf{param}_{\mathsf{BRS}})$*.* $\mathcal{A}$ *is given* $\mathsf{param}_{\mathsf{BRS}}$ *and* $\mathcal{S} := \{\mathsf{pk}_i\}_{i=1}^{n(\lambda)}$*.*
2. *Throughout the game,* $\mathcal{A}$ *has access to the same oracles as in the anonymity game (Definition 3).*
3. $\mathcal{A}$ *outputs* $k+1$ *pairs of message/signature* $\{(m_i, \Sigma_i)\}_{i=1}^{k+1}$*, and a ring* $\mathcal{R}^*$*.* $\mathcal{A}$ *wins if all the following conditions hold:*
   (a) *All* $k+1$ *signatures verify correctly (w.r.t. ring* $\mathcal{R}^*$*) and all the messages are distinct.*
   (b) *All members of the ring* $\mathcal{R}^*$ *are honest.*
   (c) $\mathcal{A}$ *engaged in at most* k *interactions with the sign oracle w.r.t. ring* $\mathcal{R}^*$*.*

Note that our definition above is not of strong unforgeability, i.e., we do not require that the adversary cannot output a new signature on an old message.

BLINDNESS. Informally, a blind ring signature is blind if an adversary (modeling a dishonest behavior of signers in the ring) does not learn the message it is signing. Moreover, it cannot link a signature to its signing session.

Definition 5 (Blindness). *A blind ring signature* BRS *satisfies blindness if for any* $\lambda \in \mathbb{N}$ *and polynomial* $n(\cdot)$ *the success probability of any p.p.t. adversary* $\mathcal{A}$ *in the following game is negligibly close to* $1/2$*:*

1. *The challenger generates* $\mathsf{param}_{\mathsf{BRS}}$ *and sends it to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *outputs a ring* $\mathcal{R}$ *(the keys of which are possibly adversarially chosen) and two messages* m_0 *and* m_1*.*
3. *The honest user interacts with the adversary concurrently to get signatures on those two messages in an arbitrary order unknown to the adversary by choosing a bit* $b \leftarrow \{0, 1\}$*.* $\mathcal{A}$ *is sent the signatures* Σ_b, Σ_{1-b}*. If any of the interactions did not finish or any of the signatures do not verify w.r.t.* $\mathcal{R}$*,* $\mathcal{A}$ *is not informed about the other signature.*
4. $\mathcal{A}$ *outputs a bit* b' *and wins if* $b = b'$*.*

As noted by [37], since the ring is public, it is a natural requirement that the two challenge signatures are signed w.r.t. the same ring. Otherwise, blindness can be trivially broken if different rings were used.

Unlike the blindness definition used in [37], which does not give the final challenge signatures to the adversary, we give the adversary the two final signatures. This is important because blindness should capture the case that a blind signature is not linkable to its signing session. Take, for example, the e-cash application where the issuing bank eventually gets to see the coins when they are deposited. Also, unlike [37], our definition allows the adversary to use corrupt keys of its choice which again provides a strong definition of blindness [1, 45].

4 Sub-linear Size Set Membership Proof over Prime-Order Groups

In this section we construct a non-interactive set membership proof. The proof allows a prover to prove that a value X_γ is contained in some set $\{X_1, \ldots, X_N\} \in \mathbb{G}^N$. Our construction is based on the underlying idea of the proof in [19] which is specific to the composite-order bilinear group setting and is based on the subgroup decision assumption [14]. Unlike the proof in [19], our proof is more general and works in both composite-order and prime-order bilinear groups. However, for efficiency reasons our focus is on the prime-order setting. We provide different instantiations over the 3 main types of the prime-order setting as summarized in Table 1. We note that it might also be possible to use the recent techniques, e.g. [28], for translating composite-order based protocols to the prime-order setting to obtain a variant of the original protocol in [19] in the prime-order setting.

The idea from [19] is to represent the set by a square $n \times n$ matrix where $n = \lceil \sqrt{N} \rceil$. If N is not square, we can repeat X_1 as many times as required to obtain a set whose size is square. As we will see, this does not affect the complexity of the proof. The prover knows a secret value X_γ and wants to produce a non-interactive proof that such a value is contained in a square $n \times n$ matrix $\mathbf{X}$ without revealing the secret value. Thus, we construct a proof for the statement $\Omega_{\mathsf{mem}} := \mathrm{PoK}\{(X_\gamma \in \mathbb{G}) : X_\gamma \in \mathbf{X}\}$, where the matrix is

$$\mathbf{X} = \begin{pmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \dots & X_{n,n} \end{pmatrix}$$

For ease of composition, the proof we give here is in the symmetric setting and is based on the DLIN assumption and can be instantiated in the asymmetric setting as summarized in Table 1. Let $(\mathbb{G}, \mathbb{G}_T, p, G, e)$ be a description of symmetric bilinear groups. We summarize the proof in the following steps where we assume that $X_\gamma = X_{\alpha,\beta}$ (i.e. X_γ lies in row α and column β) and crs is the reference string used for the proof system:

1. The prover chooses 2 secret binary vectors $\boldsymbol{y}, \boldsymbol{z} \in \{0,1\}^n$ as follows

$$y_i = \begin{cases} 1 & \text{if } i = \alpha, \\ 0 & \text{if } i \neq \alpha \end{cases} \qquad z_i = \begin{cases} 1 & \text{if } i = \beta, \\ 0 & \text{if } i \neq \beta \end{cases}$$

   The vectors $\boldsymbol{y}, \boldsymbol{z}$ will be used to anonymously single out the row and the column containing the secret value, respectively. The prover first needs to prove that each element of those vectors has a value $\in \{0,1\}$ which is done by the following QE proofs

$$\Omega_{y_i} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (y_i), \left\{\underline{y_i}(\underline{y_i} - 1) = 0\right\}\right),$$
$$\Omega_{z_i} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (z_i), \left\{\underline{z_i}(\underline{z_i} - 1) = 0\right\}\right).$$

   Additionally, the prover needs to prove that each vector contains only a single value of 1. This could be achieved by generating two extra proofs for the equations $\sum_{i=1}^n y_i = 1$ and $\sum_{i=1}^n z_i = 1$, respectively, which only adds two linear QE proofs and no extra commitments to the complexity. Alternatively, if witness-indistinguishability is sufficient, one can prove this for free by exploiting the homomorphic property of GS commitments (which are ElGamal ciphertexts [26] in the SXDH instantiation and Linear ciphertexts [12] in the DLIN instantiation). Thus, by choosing the *GS randomness* used for committing to one of those GS commitments to be the inverse of the sum of the corresponding randomness used for committing to the remaining values in the vector and then by multiplying the GS commitments to the values in each vector, the randomness cancels out and we can recover the sum of the values in the vector which allows the verifier to verify such a claim. Let $\mathcal{C}_{y_i} \leftarrow \mathsf{GSCommit}(y_i, \tau_{y_i})$ and $\mathcal{C}_{z_i} \leftarrow \mathsf{GSCommit}(z_i, \tau_{z_i})$ be the GS commitments used in committing to y_i and z_i, respectively. We set $\tau_{y_n} := -\sum_{i=1}^{n-1} \tau_{y_i}$ and $\tau_{z_n} := -\sum_{i=1}^{n-1} \tau_{z_i}$.
   Note that since the randomness $\tau_{y_1}, \dots, \tau_{y_{n-1}}$ and $\tau_{z_1}, \dots, \tau_{z_{n-1}}$ is chosen uniformly, the randomness τ_{y_n} and τ_{z_n} is also uniform.
   The verifier can verify that indeed each vector contains only a single value of 1 by checking that $\prod_i^n \mathcal{C}_{y_i} = \prod_i^n \mathcal{C}_{z_i} = \mathsf{GSCommit}(1, 0)$, i.e. the product is equal to a trivial GS commitment to 1.
   In total, this step requires $2n$ GS commitments and $2n$ QE proofs.
2. The prover anonymously singles out the row containing his secret value. To do so, for each column j in the matrix compute $X_{\alpha,j} := \prod_{i=1}^n X_{i,j}^{y_i}$. Note that $\boldsymbol{X_\alpha}$ contains the messages in row α of the matrix $\mathbf{X}$. The prover generates the following MSME proofs to prove that each $X_{\alpha,j}$ was computed correctly

$$\Omega_{X_{\alpha,j}} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (X_{\alpha,j}, y_1, \dots, y_n), \left\{\prod_{i=1}^n X_{i,j}^{y_i} \cdot \underline{X_{\alpha,j}^{-1}} = 1\right\}\right).$$

3. Finally, the prover proves that the value X_γ is contained in the secret vector $\boldsymbol{X_\alpha}$. This is achieved by the following MSME proof

$$\Omega_{X_\gamma} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (X_\gamma, X_{\alpha,1}, \ldots, X_{\alpha,n}, z_1, \ldots, z_n), \left\{\prod_{i=1}^{n} \underline{X_{\alpha,i}^{z_i}} \cdot \underline{X_\gamma^{-1}} = 1\right\}\right).$$

The membership proof is $\Omega_{\mathsf{mem}} := \left(\left(\boldsymbol{\mathcal{C}_y}, \boldsymbol{\mathcal{C}_z}, \boldsymbol{\mathcal{C}_{X_\alpha}}, \mathcal{C}_{X_\gamma}\right), \left(\boldsymbol{\Omega_y}, \boldsymbol{\Omega_z}, \boldsymbol{\Omega_{X_\alpha}}, \Omega_{X_\gamma}\right)\right)$.

To verify the proof, the verifier verifies the proofs $\Omega_{y_i}, \Omega_{z_i}, \Omega_{X_{\alpha,i}}$ for all $i \in [1, n]$ and Ω_{X_γ} and additionally checks that $\prod_i^n \mathcal{C}_{y_i} = \prod_i^n \mathcal{C}_{z_i} = \mathsf{GSCommit}(1, 0)$.

Note that when the proof is instantiated over asymmetric bilinear groups we need to commit to the vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and provide a proof of their equality.

Theorem 1. *The set membership proof is correct, sound and zero-knowledge.*

Proof. Correctness and soundness follow from the correctness and soundness of GS proofs and the fact that by checking that $\prod_i^n \mathcal{C}_{y_i} = \mathsf{GSCommit}(1, 0)$ and $\prod_i^n \mathcal{C}_{z_i} = \mathsf{GSCommit}(1, 0)$, the verifier ensures that only one non-zero value is contained in each vector. The witness-indistinguishability of the membership proof also follows from that of GS proofs.

When zero-knowledge is required, all the equations we prove (which are of types QE and MSME) are simulatable at no extra cost. Simply by using trivial witnesses (i.e. 0 for exponent values and 1 for group elements), we can simulate all the proofs. Thus, the proof is zero-knowledge.

Complexity of the Proof. We summarize in Table 1 the size (in group elements) of the proof in the different GS instantiations.

Component/Instantiation	DLIN	$\mathrm{DDH}_{\mathbb{G}_1}$ + $\mathrm{DLIN}_{\mathbb{G}_2}$		SXDH	
		$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_1$	$\mathbb{G}_2$
GS Commitments	$9n+3$	$4n$	$9n+3$	$4n$	$6n+2$
GS proofs	$21n+9$	$12n+4$	$15n+3$	$12n+4$	$10n+2$
Total	$30n+12$	$16n+4$	$24n+6$	$16n+4$	$16n+4$

Table 1. Complexity of the proof

We note here that in the asymmetric setting, the total size of the proof is irrespective of whether the set is in $\mathbb{G}_1^N$ or $\mathbb{G}_2^N$. Although the size of the commitments and proofs are swapped between the two cases, the total size remains the same.

To speed up verification, one can apply batch verification techniques [33, 10] to Groth-Sahai proofs.

5 Blind Ring Signature Construction

Overview of the Construction. Some of the recent round-optimal blind signature constructions e.g. [4, 29] are instantiations of Fischlin's generic construction [27], and combine the GS proof system with commitment and signature schemes that are compatible with one another. The latter is referred to as structure-preserving signatures [2]. In Fischlin's construction, the user sends a commitment to the message to the signer who in turn returns a signature on the commitment. The user then constructs the final blind signature by encrypting both the signature and the commitment and providing a NIZK proof of knowledge that the signature is valid on the commitment and that the commitment is to the message in question.

We exploit some properties of GS proofs. First, the rerandomizability of the proofs [7]. Second, that they are independent of the public terms in the equations being proven [32, 30], which as shown by [32],

allows transforming GS NIWI proofs into new NIWI/NIZK proofs by adding some/all of those public terms to the witness without knowledge of the original witness. The latter property was used by [32] to construct a group blind signature scheme.

Additionally, we require that:

1. The verification equations of the signature scheme has the form that all the monomials (e.g. the pairing in the case of PPE equations) involving the message are independent of the signing key, i.e. they involve neither the verification key nor any signature component that depends on the signing key so that we can exploit the second property above. An example scheme satisfying this condition is the automorphic scheme from [29].
2. The signature scheme signs $n+1$ group elements or n group elements and an integer where n is the size (in group elements) of the commitment so that we bind the signature to the ring. To this end, we require a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathcal{M}_{\mathsf{SIG}}$ to map the ring into the message space of the signature scheme.

The high-level idea of our generic construction is as follows: The user sends a commitment to the message to the signer. The signer signs the commitment along with the ring information and instead of sending the signature to the user in the clear, sends a GS proof of knowledge Ω'_{sig} of his public key and the signature σ such that the signature is on the public commitment to the message and that it verifies w.r.t. to the signer's public key. In order to reduce the communication overhead, one does not need to hide the whole signature and it is sufficient to just hide the components which depend on the secret key. One might additionally need to hide other parts of the signature to ensure that the proof is in a transformable form. In addition, using the set membership proof from Section 4, the signer generates a proof Ω'_{mem} to prove that his public key is in the ring. The signer sends proofs Ω'_{sig} and Ω'_{mem} plus any remaining public components of the signature to the user.

If the proofs are valid, the user first rerandomizes the proofs Ω'_{sig} and Ω'_{mem} (and their GS commitments) into Ω_{sig} and Ω_{mem}, respectively. The new proofs are now unlinkable to the original ones. Additionally, he transforms the NIWI proof Ω_{sig} into a NIZK proof by adding the commitment to the message and the remaining public components of the signature (if any) to the witness of the proof. Finally, the user adds a NIZK proof Ω_{com} to prove that the commitment is indeed to the message. The final blind ring signature is a set of GS proofs $(\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}}, \Omega_{\mathsf{com}})$ and their associated GS commitments. It is vital that the proofs are correlated, i.e. proofs Ω_{sig} and Ω_{mem} involve the same public key, and proofs Ω_{sig} and Ω_{com} involve the same commitment. Thanks to the nature of GS proofs, in our instantiations this correlation is directly realized by sharing the same GS commitment for those shared components of the witness between the proofs.

Anonymity is ensured by the NIWI/NIZK properties of the proofs and the fact that any remaining public components of the signature the user sees are independent of the signer's key. Blindness follows from the properties required by Fischlin's generic construction plus the composable rerandomizability [7] of the GS proof system. Finally, unforgeability is reduced to the unforgeability of the underlying signature scheme, the soundness of the proofs, the binding property of the commitment scheme, and the collision-resistant property of the hash function.

Efficient Instantiation. In order to get an efficient construction, we will base our signing protocol on a variant of the blind signing protocol from [29] using the signature scheme from [30] which has a short public key and is capable of signing a pair of group elements and an integer. Thus, obtaining a partially blind signing protocol as illustrated in Figure 1. We note here that the blind signature in [29] deviates from Fischlin's generic construction [27] for blind signatures in that the final signature is on the message itself rather than its commitment and it requires proofs of knowledge in the signature request protocol.

To obtain a blind ring signature on the message $(M, \tilde{M}) \in \mathbb{G}_1 \times \mathbb{G}_2$, the user commits to the message using Pedersen commitment $C := M \cdot T^s$ for some random $s \leftarrow \mathbb{Z}_p$ and computes $S := G^s$ and $\tilde{S} := \tilde{G}^s$. He then sends the commitment C along with GS proofs of knowledge Ω to prove that:

Setting	Signature Size	Assumptions
Type-1	$\mathbb{G}^{30n+42}$	DLIN, q-ADHSDH and WFCDH
Type-2	$\mathbb{G}_1^{16n+22} + \mathbb{G}_2^{24n+30}$	$\text{DDH}_{\mathbb{G}_1}$, $\text{DLIN}_{\mathbb{G}_2}$, q-ADHSDH and AWFCDH
Type-3	$\mathbb{G}_1^{16n+22} + \mathbb{G}_2^{16n+20}$	SXDH, q-ADHSDH and AWFCDH

Table 2. Size of the blind ring signature in the different instantiations

the commitment C is indeed to the message M and that the message and the randomness pairs are well-formed.

The signer first verifies the proofs and if they are valid, produces a signature $\sigma := (U', \tilde{U}', V, \tilde{V}, W)$ on the commitment C and the public integer $\mathcal{H}(\mathcal{R})$ (for some collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_p$) using the variant of the automorphic signature scheme [29] as in [30]. However, instead of sending the signature in the clear, the signer sends a GS proof of knowledge Ω'_{sig} of his public verification key $(A, \tilde{A})$ and the signature σ such that the signature verifies w.r.t. his key. Since the components U' and $\tilde{U}'$ of σ are independent of the signing key, we need not hide them. Additionally, the signer generates a proof of membership Ω'_{mem} to prove that his key is in the ring. The signer's response is $(\Omega'_{\mathsf{sig}}, \Omega'_{\mathsf{mem}}, U', \tilde{U}')$.

The user first verifies the GS proofs Ω'_{sig} and Ω'_{mem}, and that the pair $(U', \tilde{U}')$ is well-formed. If they are valid, the user rerandomizes those proofs into Ω_{sig} and Ω_{mem}, respectively, using the algorithm $\mathsf{GSRandomize}$. The new proofs are unlinkable to the original ones. The user then transforms the proof Ω_{sig} by making the signature verifiy w.r.t. to the message itself rather than its commitment: he computes $U := U' \cdot S$ and $\tilde{U} := \tilde{U}' \cdot \tilde{S}$, and transforms the last equation in Ω_{sig} from $e(\underline{W}, \underline{\tilde{A}} \cdot \underline{\tilde{V}}) = e(K \cdot C \cdot L^{\mathcal{H}(\mathcal{R})}, \tilde{G}) e(T, \tilde{U}')$ into $e(\underline{W}, \underline{\tilde{A}} \cdot \underline{\tilde{V}}) e(T^{-1}, \underline{\tilde{U}}) = e(K \cdot M \cdot L^{\mathcal{H}(\mathcal{R})}, \tilde{G})$. In addition, he hides the components $(U, \tilde{U})$ by adding a proof for the equation $e(\underline{U}, \tilde{G}) = e(G, \underline{\tilde{U}})$. The final blind ring signature is $\Sigma := (\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}})$.

The detailed construction in the asymmetric setting is as follows:

- $\underline{\mathsf{BRSSetup}(1^\lambda)}$:
  - Run $\mathcal{P} \leftarrow \mathsf{BGrpSetup}(1^\lambda, 3)$ and $(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P})$. Parse $\mathcal{P}$ as $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, G, \tilde{G}, e)$.
  - Choose a suitable collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_p$ and $F, K, L, T \leftarrow \mathbb{G}_1$.
  - Set $\mathsf{param}_{\mathsf{BRS}} := (\mathcal{P}, \mathsf{crs}, \mathcal{H}, F, K, L, T)$. Return $\mathsf{param}_{\mathsf{BRS}}$.
- $\underline{\mathsf{BRSKeyGen}(\mathsf{param}_{\mathsf{BRS}})}$:
  - Choose $a \leftarrow \mathbb{Z}_p$ and set $A := G^a$ and $\tilde{A} := \tilde{G}^a$. Set $\mathsf{sk} := a$, $\mathsf{pk} := (A, \tilde{A})$. Return $(\mathsf{sk}, \mathsf{pk})$.
- The signing protocol $\langle \mathsf{BRSObtain}((M, \tilde{M}), \mathcal{R}), \mathsf{BRSSign}(\mathsf{sk}, \mathcal{R}) \rangle$ is in Figure 2.
- $\underline{\mathsf{BRSVerify}((M, \tilde{M}), \Sigma, \mathcal{R})}$
  - Parse Σ as $(\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}})$.
  - Return 1 if $\mathsf{GSVerify}(\mathsf{crs}, \Omega_{\mathsf{mem}}) = 1$ and $\mathsf{GSVerify}(\mathsf{crs}, \Omega_{\mathsf{sig}}) = 1$. Otherwise, return 0.

Theorem 2. *The construction is a secure blind ring signature scheme.*

The full proof is in Appendix A.

Efficiency of the Construction. As mentioned earlier, our construction is the first realization in the standard model and also the first to offer sub-linear signatures instead of linear ones. Table 2 summarizes the size of the signature as well as the required assumptions for the different instantiations. Type-1 instantiation uses the DLIN instantiation of GS proofs, Type-2 uses GS proofs based on DDH in $\mathbb{G}_1$ and DLIN in $\mathbb{G}_2$ as used in [34], and Type-3 uses the SXDH instantiation of the proofs.

To give example concrete figures, we consider a security level equivalent to 128-bit symmetric key security. For a ring consisting of 10,000 members, the Type-1 instantiation, where the size of elements of group $\mathbb{G}$ is 512 bits, yields signatures of size of approximately 190 kB. At the same security level in the asymmetric setting where elements of $\mathbb{G}_1$ are of size 256 bits and those of $\mathbb{G}_2$ are of size 512 bits, the signature size is 203 kB and 152 kB in the Type-2 and Type-3 instantiations, respectively. Again, the verification of the signature can be made more efficient by batch verifying Groth-Sahai proofs [33, 10].

$\underline{\mathsf{BRSObtain} \rightarrow \mathsf{BRSSign}}$
– Choose $s \leftarrow \mathbb{Z}_p$ and compute $S := G^s$, $\tilde{S} := \tilde{G}^s$ and $C := M \cdot T^s$.
– $\Omega \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (M, \tilde{M}, S, \tilde{S}), \Big\{e(\underline{M}, \tilde{G}) = e(G, \underline{\tilde{M}}) \ \wedge \ e(\underline{S}, \tilde{G}) = e(G, \underline{\tilde{S}}) \ \wedge \ e(T, \underline{\tilde{S}})e(\underline{M}, \tilde{G}) = e(C, \tilde{G})\Big\}\Big)$.
– Send (C, Ω) to BRSSign.

$\underline{\mathsf{BRSSign} \rightarrow \mathsf{BRSObtain}}$
– If $\mathsf{GSVerify}(\mathsf{crs}, \Omega) \neq 1$ Then Abort().
– Choose $u, v \leftarrow \mathbb{Z}_p$ and set $U' := G^u$, $V := F^v$, $W := (K \cdot T^u \cdot C \cdot L^{\mathcal{H}(\mathcal{R})})^{\frac{1}{a+v}}$, $\tilde{U}' := \tilde{G}^u$, $\tilde{V} := \tilde{G}^v$.
– $\Omega'_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (V, \tilde{V}, W, \tilde{A}), \Big\{e(\underline{V}, \tilde{G}) = e(F, \underline{\tilde{V}}) \ \wedge \ e(\underline{W}, \underline{\tilde{A}} \cdot \underline{\tilde{V}}) = e(K \cdot C \cdot L^{\mathcal{H}(\mathcal{R})}, \tilde{G})e(T, \tilde{U}')\Big\}\Big)$.
– Compute the membership proof $\Omega'_{\mathsf{mem}} \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (\tilde{A}), \Big\{\underline{\tilde{A}} \in \mathcal{R}_{\tilde{A}}\Big\}\Big)$. [1]
– Send $(\Omega'_{\mathsf{sig}}, \Omega'_{\mathsf{mem}}, U', \tilde{U}')$ to BRSObtain.

$\underline{\mathsf{BRSObtain}}$
– Abort if $e(U', \tilde{G}) \neq e(G, \tilde{U}')$, $\mathsf{GSVerify}(\mathsf{crs}, \Omega'_{\mathsf{sig}}) \neq 1$ or $\mathsf{GSVerify}(\mathsf{crs}, \Omega'_{\mathsf{mem}}) \neq 1$.
– Compute $U := U' \cdot S$ and $\tilde{U} := \tilde{U}' \cdot \tilde{S}$.
– $\Omega_{\mathsf{sig}} \leftarrow \mathsf{GSRandomize}(\Omega'_{\mathsf{sig}})$, $\Omega_{\mathsf{mem}} \leftarrow \mathsf{GSRandomize}(\Omega'_{\mathsf{mem}})$ and transform Ω_{sig} as follows:
– $\Omega_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\Big(\mathsf{crs}, (V, \tilde{V}, W, \tilde{A}, U, \tilde{U}), \Big\{e(\underline{V}, \tilde{G}) = e(F, \underline{\tilde{V}}) \ \wedge \ e(\underline{U}, \tilde{G}) = e(G, \underline{\tilde{U}}) \ \wedge \ e(\underline{W}, \underline{\tilde{A}} \cdot \underline{\tilde{V}})e(T^{-1}, \underline{\tilde{U}}) = e(K \cdot M \cdot L^{\mathcal{H}(\mathcal{R})}, \tilde{G})\Big\}\Big)$.
– Output $\Sigma := (\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}})$.

Fig. 2. The signing protocol

6 Generic Construction of Ring Signatures over Prime-Order Bilinear Groups

Here we provide a generic construction for ring signatures without random oracles by combining the set membership proof from Section 4 with any compatible signature scheme.

Let $\mathsf{Sig} := ([\mathsf{SigSetup}], \mathsf{SigKeyGen}, \mathsf{SigSign}, \mathsf{SigVerify})$ be an existentially unforgeable signature scheme secure against adaptive chosen-message attack that works in any of the 3 main types (cf. Section 2.1) of prime-order bilinear groups. Let $\mathcal{M}_{\mathsf{Sig}}$ be its message space, $(\mathsf{sk}, \mathsf{pk})$ be its key pair and $\sigma := (\sigma_1, \ldots, \sigma_n)$ be its signatures for some positive integer n with the condition that for any $i \in [1, n]$, σ_i is a group element if it depends on sk. [2] Our construction is as follows:

- $\mathsf{RSSetup}(1^\lambda)$: Run $\mathcal{P} \leftarrow \mathsf{BGrpSetup}(1^\lambda, \mathsf{tp})$ for $\mathsf{tp} \in [1, 3]$, $(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P})$. Choose a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathcal{M}_{\mathsf{Sig}}$. The public parameters is then $\mathsf{param}_{\mathsf{RS}} := (\mathcal{P}, \mathsf{crs}, \mathcal{H})$. Note that if Sig requires setup, then the output of SigSetup is also added to $\mathsf{param}_{\mathsf{RS}}$.
- $\mathsf{RSKeyGen}(\mathsf{param}_{\mathsf{RS}})$: Run SigKeyGen to obtain $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{RSSign}(\mathsf{sk}_i, m, \mathcal{R})$: To sign a message $m \in \{0,1\}^*$ w.r.t. a ring $\mathcal{R} := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ with the condition that $\mathsf{pk}_i \in \mathcal{R}$, run $\sigma \leftarrow \mathsf{SigSign}(\mathsf{sk}_i, \mathcal{H}(m, \mathcal{R}))$. Then generate the following two Groth-Sahai proofs where $\bar{\sigma}$ is the subset of σ which depends on the secret key sk.

$$\Omega_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\{\mathsf{crs}, (\mathsf{pk}_i, \bar{\sigma}), \{\mathsf{SigVerify}(\underline{\mathsf{pk}_i}, \mathcal{H}(m, \mathcal{R}), \sigma) = 1\}\},$$
$$\Omega_{\mathsf{mem}} \leftarrow \mathsf{GSProve}\{\mathsf{crs}, (\mathsf{pk}_i), \{\underline{\mathsf{pk}_i} \in \mathcal{R}\}\}.$$

  The ring signature is then $\Sigma := (\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}}, \{\sigma\} \setminus \{\bar{\sigma}\})$. Again, the two proofs must be correlated, i.e. they involve the same public key pk_i. This is checked by ensuring that both proofs use the same GS commitment to pk_i when verifying the signature.

[1] We only prove membership for one component of the key. The verifier can verify that all keys in the ring are well-formed. Alternatively, one can add a proof for the equation $e(\underline{A}, \tilde{G}) = e(G, \underline{\tilde{A}})$. It is a matter of trade-off between communication and computation.

[2] Unlike structure-preserving signatures [2], we do not require that the messages are group elements.

Instantiation	Setting	Signature Size	Complexity Assumptions
Waters	Type-1	$\mathbb{G}^{30n+19}$	CDH + DLIN
	Type-2	$\mathbb{G}_1^{16n+10} + \mathbb{G}_2^{24n+13}$	Co-CDH + $\text{DDH}_{\mathbb{G}_1}$ + $\text{DLIN}_{\mathbb{G}_2}$
	Type-3	$\mathbb{G}_1^{16n+11} + \mathbb{G}_2^{16n+9}$	Co-CDH* + SXDH
FBB	Type-1	$\mathbb{G}^{42n+39} + \mathbb{Z}_p^4$	q-SDH + DLIN
	Type-2	$\mathbb{G}_1^{20n+14} + \mathbb{G}_2^{30n+21} + \mathbb{Z}_p^4$	q-SDH + $\text{DDH}_{\mathbb{G}_1}$ + $\text{DLIN}_{\mathbb{G}_2}$
	Type-3	$\mathbb{G}_1^{20n+14} + \mathbb{G}_2^{20n+14} + \mathbb{Z}_p^3$	q-SDH + SXDH

Table 3. Example instantiations of the generic ring signatures construction

– $\mathsf{RSVerify}(m, \Sigma, \mathcal{R})$: To verify that the ring signature Σ is a valid signature on the message m w.r.t. the ring $\mathcal{R}$, verify the two proofs Ω_{mem} and Ω_{sig}.

Theorem 3. *The generic construction is a secure ring signature scheme for message space* $\{0,1\}^*$.

The full proof is in Appendix B.

Next, we provide some example instantiations of the construction and compare their efficiency in the different bilinear group settings in Table 3.

6.1 Instantiation-1 (Using the Full Boneh-Boyen (FBB) Signature [11])

Here we instantiate the signature scheme Sig with the full Boneh-Boyen signature scheme [11]. The following instantiation is in Type-1 setting:

– $\mathsf{RSSetup}(1^\lambda)$: Run $\mathcal{P} \leftarrow \mathsf{BGrpSetup}(1^\lambda, 1)$, $(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P})$. Choose a suitable collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{Z}_p$. Set $\mathsf{param_{RS}} := (\mathcal{P}, \mathsf{crs}, \mathcal{H})$.
– $\mathsf{RSKeyGen}(\mathsf{param_{RS}})$: Select $a, b \leftarrow \mathbb{Z}_p$ and set $A := G^a$ and $B := G^b$. Set $\mathsf{sk} := (a, b)$ and $\mathsf{pk} := (A, B)$.
– $\mathsf{RSSign}((a, b), m, \mathcal{R})$: Select $r \leftarrow \mathbb{Z}_p$ and compute $h := \mathcal{H}(m, \mathcal{R})$. Generate the full Boneh-Boyen signature $\sigma := G^{\frac{1}{a+r\cdot b+h}}$. Let $A_\gamma := A$ and $B_\gamma := B$. Generate the following proofs:

$$\Omega_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (\sigma, A_\gamma, B_\gamma, B'), \left\{\underline{B_\gamma}^r = \underline{B'} \ \wedge \ e(\underline{\sigma}, \underline{A_\gamma})e(\underline{\sigma}, \underline{B'})e(\underline{\sigma}, G^{\mathcal{H}(m,\mathcal{R})}) = e(G, G)\right\}\right),$$

$$\Omega_{\mathsf{mem}} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (A_\gamma, B_\gamma), \left\{\underline{A_\gamma} \in \mathcal{R}_A \ \wedge \ \underline{B_\gamma} \in \mathcal{R}_B\right\}\right)^{3}.$$

The final ring signature is $\Sigma := (\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}}, r)$.

The proof of the following Theorem follows easily from that of Theorem 3.

Theorem 4. *The construction is secure if the DLIN and q-SDH assumptions hold in group* $\mathbb{G}$ *and the hash function* $\mathcal{H}$ *is collision-resistant.*

6.2 Instantiation-2 (Using Waters Signature Scheme [52])

Using Waters signature scheme [52] instantiated over prime-order groups, we get a more efficient construction (yielding sub-linear size signatures) than the Waters signature based construction [50] (yielding linear size signatures in composite-order groups).

Unlike the instantiation in Section 6.1, the public key only contains one group element and hence it requires a simpler membership proof. We give the instantiation below in Type-1 setting:

[3] In producing the proof, we will use the same vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ for both proofs since those elements are in symmetric positions in their respective matrices.

- $\mathsf{RSSetup}(1^\lambda)$: $\mathcal{P} \leftarrow \mathsf{BGrpSetup}(1^\lambda, 1)$ and parse $\mathcal{P}$ as $(\mathbb{G}, \mathbb{G}_T, p, G, e)$. Run $(\mathsf{crs}, \mathsf{xk}) \leftarrow \mathsf{GSSetup}(\mathcal{P})$ and choose a suitable collision-resistant hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \{0,1\}^k$. Select $a \leftarrow \mathbb{Z}_p$, $U_i \leftarrow \mathbb{G}$ for all $i \in [0,k]$ then set $A := G^a$, the public parameter is then set to $\mathsf{param_{RS}} := (\mathcal{P}, \mathsf{crs}, \mathcal{H}, A, U_0, U_1, \ldots, U_k)$.
- $\mathsf{RSKeyGen}(\mathsf{param_{RS}})$: Choose $b \leftarrow \mathbb{Z}_p$ and compute $B := G^b$. Set $\mathsf{sk} := A^b$ and $\mathsf{pk} := B$.
- $\mathsf{RSSign}(\mathsf{sk}, m, \mathcal{R})$: To sign a message $m \in \{0,1\}^*$ using the secret key $\mathsf{sk} = A^b$, compute $h := \mathcal{H}(m, \mathcal{R})$. Parse h as $(h_1, \ldots, h_k) \in \{0,1\}^k$ and then compute $H := U_0 \prod_{i=1}^k U_i^{h_i}$. Choose $r \leftarrow \mathbb{Z}_p$ and compute the signature $\sigma_1 := G^r$ and $\sigma_2 := \mathsf{sk} \cdot H^r$. Let $\sigma := (\sigma_1, \sigma_2)$ and $B_\gamma := B$. Generate the following proofs:

$$\Omega_{\mathsf{sig}} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (\sigma_2, B), \left\{ e(\underline{B}, A^{-1}) e(\underline{\sigma_2}, G) = e(\sigma_1, U_0 \prod_{i=1}^k U_i^{h_i}) \right\}\right),$$
$$\Omega_{\mathsf{mem}} \leftarrow \mathsf{GSProve}\left(\mathsf{crs}, (B), \{\underline{B} \in \mathcal{R}_B\}\right).$$

The final ring signature is $\Sigma := (\Omega_{\mathsf{sig}}, \Omega_{\mathsf{mem}}, \sigma_1)$.

The proof of the following Theorem follows easily from that of Theorem 3.

Theorem 5. *The construction is secure if the DLIN assumption (which implies the CDH assumption) holds in group $\mathbb{G}$ and the hash function $\mathcal{H}$ is collision-resistant.*

6.3 Instantiating the Construction in [19] over Prime-Order Groups

Combining our set membership proof with the weakly secure Boneh-Boyen signature scheme [11] and one-time signatures instantiated over prime-order groups, we get efficient instantiations of the composite-order construction given in [19]. The anonymity of the construction follows from the hardness of the assumption on which we base GS proofs, whereas the unforgeability of the constructions follows from the q-SDH assumption and the soundness of GS proofs.

7 Acknowledgements

This work was supported by ERC Advanced Grant ERC-2010-AdG-267188-CRIPTO and EPSRC via grant EP/H043454/1.

References

1. M. Abdalla, C. Namprempre and G. Neven. On the (im)possibility of blind message authentication codes. In *CT-RSA 2006*, Springer LNCS 3860, 262–279, 2006.
2. M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Crypto 2010*, Springer LNCS 6223, 209–236, 2010.
3. M. Abe, and E. Fujisaki. How to date blind signatures. In *Advances in Cryptology – Asiacrypt 1996*, Springer LNCS 1163, 244–251, 1996.
4. M. Abe, K. Haralambiev and M. Ohkubo. Signing on Elements in Bilinear Groups for Modular Protocol Design. *Cryptology ePrint Archive, Report 2010/133*. `http://eprint.iacr.org/2010/133`.
5. M. Abe, M. Ohkubo and K. Suzuki. 1-out-of-n Signatures from a Variety of Keys. In *Advances in Cryptology – Asiacrypt 2002*, Springer LNCS 2501, 415–432, 2002.
6. M. Abe, T. Okamoto. Provably Secure Partially Blind Signatures. In *Advances in Cryptology – Crypto 2000*, Springer LNCS 1880, 271–286, 2000.
7. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya and H. Shacham. Randomizable proofs and delegatable anonymous credentials. In *Advances in Cryptology – CRYPTO 2009*, Springer LNCS 5677, 108–125, 2009.
8. M. Bellare and P. Rogaway. Random oracles are practical: A Paradigm for Designing Efficient Protocols. In: *ACM Conference on Computer and Communications Security 1993*, ACM, pp. 62–73.
9. A. Bender, J. Katz and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of cryptography– TCC 2006*, Springer-Verlag LNCS 3876, 60–79, 2006.

10. O. Blazy, G. Fuchsbauer, M. Izabach'ene, A. Jambert, H. Sibert and D. Vergnaud. Batch Groth-Sahai. *Cryptology ePrint Archive, Report 2010/040*. http://eprint.iacr.org/2010/040.
11. D. Boneh and X. Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. *Journal of Cryptology*, **21(2)**, 149–177, 2008.
12. D. Boneh, X. Boyen and H. Shacham. Short group signatures. In *Advances in Cryptology – CRYPTO 2004*, Springer LNCS 3152, 41–55, 2004.
13. D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology – EUROCRYPT 2003*, Springer LNCS 2656, 416–432, 2003.
14. D. Boneh, E. Goh and K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In *Theory of Cryptography Conference – TCC 2005*, Springer LNCS 3378, 325–341, 2005.
15. X. Boyen. Mesh Signatures. In *Advances in Cryptology – EUROCRYPT 2007*, Springer LNCS 4515, 210–227, 2007.
16. Z. Brakerski and Y.T. Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. In *Cryptology ePrint Archive, Report 2010/086*, http://eprint.iacr.org/2010/086.pdf.
17. R. Canetti, O. Goldreich and S. Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC* ACM Press, 209–218, 1998.
18. T.K. Chan, K. Fung, J.K. Liu and V.K. Wei Blind Spontaneous Anonymous Group Signatures for Ad Hoc Groups. In *ESAS 2004*, Springer LNCS 3313, 82–94, 2004.
19. N. Chandran, J. Groth and A. Sahai. Ring Signatures of Sub-linear Size Without Random Oracles. In *ICALP 2007*, Springer LNCS 4596, 423–434, 2007.
20. S. Chatterjee and D. Hankerson and E. Knapp and A. Menezes. Comparing Two Pairing-Based Aggregate Signature Schemes. In *Cryptology ePrint Archive, Report 2009/060*, http://eprint.iacr.org/2009/060.pdf.
21. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO 1982*, Plenum Press, 199–203, 1983.
22. R. Cramer, R. Gennaro and B. Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *Advances in Cryptology – EUROCRYPT 1997*, Springer LNCS 1233, 103–118, 1997.
23. Y. Desmedt, C. Goutier and S. Bengio. Special Uses and Abuses of the Fiat-Shamir Passport Protocol. In *CRYPTO 1987*, Springer LNCS 293, 21–39, 1988.
24. Y. Dodis, A. Kiayias, A. Nicolosi and Victor Shoup. Anonymous Identification in Ad Hoc Groups. In *Advances in Cryptology – EUROCRYPT 2004*, Springer LNCS 3027, 609–626, 2004.
25. C. Dwork and M. Naor. Zaps and their applications. In *FOCS 2000*, 283–293, 1999.
26. T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology – CRYPTO 1984*, Springer LNCS 196, 10–18, 1985.
27. M. Fischlin. Round-optimal composable blind signatures in the common reference string model. In *Advances in Cryptology – CRYPTO 2006*, Springer LNCS 4117, 60–77, 2006.
28. D.M. Freeman. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In *Advances in Cryptology – EUROCRYPT 2010*, Springer LNCS 6110, 44–61, 2010.
29. G. Fuchsbauer. Automorphic Signatures in Bilinear Groups and an Application to Round-Optimal Blind Signatures. In *Cryptology ePrint Archive, Report 2009/320*, http://eprint.iacr.org/2009/320.pdf.
30. G. Fuchsbauer. Commuting signatures and verifiable encryption. In *Advances in Cryptology – EUROCRYPT 2011*, Springer LNCS 6632, 224–245, 2011.
31. S. Galbraith, K. Paterson and N.P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, **156**, 3113–3121, 2008
32. E. Ghadafi. Formalizing group blind signatures and practical constructions without random oracles. In *Cryptology ePrint Archive, Report 2011/402*, http://eprint.iacr.org/2011/402.pdf.
33. E. Ghadafi, N.P. Smart and B. Warinschi. Practical zero-knowledge proofs for circuit evaluation. In *Coding and Cryptography: IMACC 2009*, Springer LNCS 5921, 469–494, 2009.
34. E. Ghadafi, N.P. Smart and B. Warinschi. Groth-Sahai proofs revisited. In *PKC 2010*, Springer LNCS 6056, 177–192, 2010.
35. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Advances in Cryptology – EUROCRYPT 2008*, Springer LNCS 4965, 415–432, 2008.
36. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups (full version). http://www.brics.dk/~jg/WImoduleFull.pdf
37. J. Herranz and F. Laguillaumie. Blind Ring Signatures Secure Under the Chosen-Target-CDH Assumption. In *Information Security – ISC 2006*, Springer LNCS 4176, 117–130, 2006.
38. J. Herranz and G. Sez. Forking Lemmas for Ring Signature Schemes. In *Progress in Cryptology – INDOCRYPT 2003*, Springer LNCS 2904, 266-279, 2003.
39. M. Jakobsson, K. Sako and R. Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology – EUROCRYPT 1996*, Springer LNCS 1070, 143–154, 1996.
40. A. Juels, M. Luby and R. Ostrovsky. Security of blind digital signatures. In *Advances in Cryptology – CRYPTO '97*, Springer LNCS 1294, 150–164, 1997.
41. A. Lysyanskaya and R. Zulfikar. Group blind digital signatures: A scalable solution to electronic cash. In *Financial Cryptography 1998*, Springer LNCS 1465, 184–197, 1998.

42. M. Naor. Deniable Ring Authentication. In *Advances in Cryptology – CRYPTO 2002*, Springer LNCS 2442, 481–498, 2002.
43. L. Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA 2005*, Springer LNCS 3376, 275–292, 2005.
44. K. Q. Nguyen, Y. Mu, and V. Varadharajan. Divertible Zero-Knowledge Proof of Polynomial Relations and Blind Group Signature. In *ACISP 1999*, Springer LNCS 1587, 117–128, 1999.
45. T. Okamoto. Efficient blind and partially blind signatures without random oracles. In *TCC 2006*, Springer LNCS 3876, 80–99, 2006.
46. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, **13(3)**, 361–396, 2000.
47. M. Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology – Crypto 2003*, Springer LNCS 2729, 96–109, 2003.
48. R.L. Rivest, A. Shamir and Y. Tauman. How to leak a secret . In *Advances in Cryptology – ASIACRYPT 2001*, Springer LNCS 2248 , 552–565, 2001.
49. S. Schäge and J. Schwenk. A CDH-Based Ring Signature Scheme with Short Signatures and Public Keys. In *Financial Cryptography 2010*, Springer LNCS 6052, 129–142, 2010.
50. H. Shacham and B. Waters. Efficient ring signatures without random oracles. In *PKC 2007*, Springer LNCS 4450, 166–180, 2007.
51. V. Shoup. Lower bounds for discrete logarithms and related problems. In *Eurocrypt 1997*, Springer LNCS 1233, 256–266, 1997.
52. B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology – EUROCRYPT 2005*, Springer LNCS 3494, 114–127, 2005.
53. Q. Wu, F. Zhang, W. Susilo and Yi Mu. An Efficient Static Blind Ring Signature Scheme. In *Information Security and Cryptology – ICISC 2005*, Springer LNCS 3935, 410–423, 2006.
54. J. Zhang, H. Chen, X. Liu and C. Liu. An Efficient Blind Ring Signature Scheme without Pairings. In *WAIM Workshops 2010*, Springer LNCS 6185, 177–188, 2010.

A Proof of Theorem 2

Proof. Correctness follows from the perfect completeness of GS proofs and the correctness of the underlying signature scheme and hence it is trivial to verify.

Lemma 1. *The construction is anonymous (against full key exposure) providing that GS proofs are hiding (i.e. witness-indistinguishable/zero-knowledge) and that the SXDH assumption holds.*

Proof. We prove that

$$\mathsf{Adv}^{\text{Anon}}_{\mathsf{BRS},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\text{Hiding}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) + \mathsf{Adv}^{\text{SXDH}}_{\mathcal{A}_2}(\lambda).$$

By the perfect NIWI/NIZK of GS proofs in the simulation setting, we have that $\mathsf{Adv}^{\text{Hiding}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) = 0$ and hence an adversary has a zero advantage in breaking the NIWI/NIZK properties of the proofs.

What is left is to show that the adversary cannot distinguish which of the two types of the CRS, it was given. By the security of the SXDH assumption in this instantiation (as shown in [36]), the adversary has a negligible advantage in telling apart a simulation CRS from a soundness one and therefore this only negligibly changes $\mathcal{B}$ success probability. The reduction to the security of SXDH assumption was proven in [36].

Note that the values U' and $\tilde{U}'$ the user sees are independent of the signer's key and thus reveal no information about the signer. Thus, we have that $\mathsf{Adv}^{\text{Anon}}_{\mathsf{BRS},\mathcal{B}}(\lambda) \leq v(\lambda)$ and therefore the construction is anonymous.

Lemma 2. *The construction is unforgeable if the GS proof system is sound, the hash function $\mathcal{H}$ is collision-resistant, and the signature scheme used in the blind signing protocol is existentially unforgeable.*

Proof. We show that

$$\mathsf{Adv}^{\text{Unforg}}_{\mathsf{BRS},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\text{Sound}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) + \mathsf{Adv}^{\text{Coll}}_{\mathcal{H},\mathcal{A}_2}(\lambda) + n(\lambda) \cdot \mathsf{Adv}^{\text{Unforg}}_{\mathsf{SIG},\mathcal{A}_3}(\lambda) \cdot$$

We distinguish between 3 different types of adversaries:

– **Type I:** This type forges signatures by producing proofs for false statements.
– **Type II:** This type forges signatures by finding two different rings $\mathcal{R} \neq \mathcal{R}'$ but $\mathcal{H}(\mathcal{R}) = \mathcal{H}(\mathcal{R}')$.
– **Type III:** This type forges signatures by breaking the unforgeability of the underlying signature scheme and thus forging signatures for new messages that the signer did not sign.

We start by instantiating GS proofs in the soundness setting so that the resulting proofs are perfectly sound and therefore this eliminates the case that the adversary (i.e. Type I) forges signatures by faking proofs for false statements. Thus, we have that $\mathsf{Adv}^{\text{Sound}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) = 0$.

By the collision-resistant property of the hash function, the adversary (i.e. Type II) has a negligible advantage in finding two different rings $\mathcal{R} \neq \mathcal{R}'$ such that $\mathcal{H}(\mathcal{R}) = \mathcal{H}(\mathcal{R}')$. If this is the case, we can construct an adversary $\mathcal{A}_2$ that breaks the collision-resistant property of the hash function. Thus, we have that $\mathsf{Adv}^{\text{Coll}}_{\mathcal{H},\mathcal{A}_2}(\lambda) \leq v(\lambda)$.

Finally, we show that by the existential unforgeability of the underlying signature scheme, the adversary has a negligible advantage in forging blind ring signatures by forging signatures for messages/rings he did not ask the sign oracle to sign for him. We show that if $\mathcal{B}$ can forge a blind ring signature with a non-negligible probability, we can construct an adversary $\mathcal{A}_3$ that breaks the unforgeability of the underlying signature scheme used in the blind signing protocol with a non-negligible probability.

Adversary $\mathcal{A}_3$ generates $\mathsf{param}_{\mathsf{BRS}}$ where it instantiates GS proofs in the soundness setting. It also generates n pairs of keys, where $n = n(\lambda)$ is an upper bound on the number of honest signers $\mathcal{B}$ uses in its game. $\mathcal{A}_3$ randomly chooses $i^* \in [1, n]$ and generates all the keys for $i \in [1, n] \setminus \{i^*\}$ normally with the exception that it sets pk_{i^*} to be the public key it gets from its own unforgeability game and hence $\mathcal{A}_3$ does not know the corresponding secret key sk_{i^*}. Let $\mathcal{S} := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$. $\mathcal{A}_3$ passes $\mathsf{param}_{\mathsf{BRS}}$ and $\mathcal{S}$ to $\mathcal{B}$.

When $\mathcal{B}$ asks for a corrupt query on any $i \neq i^*$, $\mathcal{A}_3$ simply reveals the corresponding secret key sk_i. If $\mathcal{B}$ asks a corrupt query on $i = i^*$, $\mathcal{A}_3$ aborts. This happens with probability $1/n$.

For sign queries made by $\mathcal{B}$ for any signer Signer_i where $i \neq i^*$, $\mathcal{A}_3$ generates the signature itself by using the corresponding secret key and constructing the rest of the signature and then returning $(\Omega'_{\mathsf{mem}}, \Omega'_{\mathsf{sig}}, U', \tilde{U}')$ as the answer. On the other hand, when $\mathcal{B}$ asks for signatures by Signer_{i^*}, $\mathcal{A}_3$ first extracts the message pair $(M, \tilde{M})$ from the proofs the adversary sends in the first round and then forwards the message and $\mathcal{H}(\mathcal{R})$ to its own sign oracle. After it receives the answer from its oracle, it generates the rest of the blind ring signature itself and returns $(\Omega'_{\mathsf{mem}}, \Omega'_{\mathsf{sig}}, U', \tilde{U}')$ as the answer; note this does not require knowledge of sk_{i^*} because all the proofs of knowledge involve pk_{i^*} and not sk_{i^*}.

Eventually, $\mathcal{B}$ terminates and outputs $k+1$ pairs of message/signature $\{((M_j, \tilde{M}_j), \Sigma_j)\}_{j=1}^{k+1}$ by ring $\mathcal{R}^* \subseteq \mathcal{S}$. At this stage, $\mathcal{A}_3$ first verifies that all members in $\mathcal{R}^*$ are indeed all honest, that it never participated in more than k interactions with $\mathcal{B}$ w.r.t. ring $\mathcal{R}^*$ and that all messages in the forgery are distinct. If all conditions are satisfied, $\mathcal{A}_3$ uses the GS system extraction key to extract the public keys and the signatures σ_j. With probability $1/n$ the extracted public key will be for the signer we have guessed. If this is not the case, $\mathcal{A}_3$ aborts.

$\mathcal{A}_3$ now identifies the extra message pair $(M^*, \tilde{M}^*)$ that it did not query its sign oracle on and returns it with $\mathcal{H}(\mathcal{R}^*)$ and σ^* as its answer in its unforgeability game. Clearly, if $\mathcal{B}$ wins its game with non-negligble probability, $\mathcal{A}_3$ breaks the unforgeability of the underlying signature scheme. By the unforgeability of the underlying signature scheme (see [30]) we have that $\mathsf{Adv}^{\text{Unforg}}_{\mathsf{SIG},\mathcal{A}_3}(\lambda) \leq v(\lambda)$.

Summing up, we have that $\mathsf{Adv}^{\text{Unforg}}_{\mathsf{BRS},\mathcal{B}}(\lambda) \leq v(\lambda)$ and thus our construction is unforgeable.

Lemma 3. *The construction is blind if the GS proof system is hiding and rerandomizable, the* Pedersen *commitment (used in the first round of the signing protocol) is hiding and the SXDH assumption holds.*

Proof. We show that

$$\mathsf{Adv}^{\text{Blind}}_{\mathsf{BRS},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\text{Hiding}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) + \mathsf{Adv}^{\text{Hiding}}_{\mathsf{Pedersen},\mathcal{A}_2}(\lambda) + \mathsf{Adv}^{\text{SXDH}}_{\mathcal{A}_3}(\lambda).$$

By the perfect NIWI/NIZK of GS proofs in the simulation setting, we have that $\mathsf{Adv}^{\text{Hiding}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) = 0$. Also, the commitment used in the first round of the signing protocol is perfectly hiding, and therefore we also have $\mathsf{Adv}^{\text{Hiding}}_{\mathsf{Pedersen},\mathcal{A}_2}(\lambda) = 0$. Furthermore, we have by the security of GS proofs (i.e. the security of the SXDH assumption [36]) that an adversary has a negligible advantage in telling apart a simulation CRS from a soundness one and therefore this only negligibly changes $\mathcal{B}$'s success probability. Also, by the rerandomizability of GS proofs [7], we have that the rerandomized proofs Ω_{mem} and Ω_{sig} are independent of the original proofs Ω'_{mem} and Ω'_{sig} the signer generated and thus they cannot be linked. Thus, the construction is blind.

B Proof of Theorem 3

Proof. **Correctness**. This follows from the perfect completeness of the GS proof system and the correctness of the Sig scheme.

Lemma 4. *If the GS proof system is hiding (i.e. witness-indistinguishable/zero-knowledge), then the construction is anonymous under full key exposure.*

Proof. We instantiate the proof system in the simulation setting which results in proofs that are perfectly hiding. By the security of the GS proof system (see [36]), the adversary has a negligible advantage in distinguishing between the simulation setting and the soundness setting.

Since in our proofs within the final signature, we hide all the components of the signature σ that depend on the key and the public key itself w.r.t. which the signature σ verifies, all the public values in the signature are independent of the signer's public key. Thus, if the adversary learns about the witness used in the proofs, we can reduce anonymity to breaking the hiding properties of GS proofs and hence break the witness-indistinguishability/zero-knowledge of the system. See [36] for more details.

Lemma 5. *If the GS proof system is sound, the hash function $\mathcal{H}$ is collision-resistant, and the signature scheme* Sig *is existentially unforgeable against adaptive chosen-message attack, the construction is unforgeable in the presence of insider corruption.*

Proof. Instantiating the proof system in the soundness setting yields proofs that are perfectly sound and hence the adversary against unforgeability has zero advantage in succeeding by faking proofs for false statements. We thus exclude this case and we are left with two types of forgers as follows, where we assume that the forger aganist the ring signature makes q sign queries:

- **Type-I:** The forgery is on message m^* and ring $\mathcal{R}^*$ where $\mathcal{H}(m^*, \mathcal{R}^*) = \mathcal{H}(m_i, \mathcal{R}_i)$ for some previous query $(m_i, \mathcal{R}_i)$ and $(m^*, \mathcal{R}^*) \neq (m_i, \mathcal{R}_i)$ for some $i \in [1, q]$.
- **Type-II:** We have that $\mathcal{H}(m^*, \mathcal{R}^*) \notin \{\mathcal{H}(m_i, \mathcal{R}_i)\}_{i=1}^{q}$.

We reduce Type-I forgeries to breaking the collision-resistant property of the hash function. Thus, by the security of the hash function, we have that Type-I forgery can only happen with a negligible probability.

We now focus on Type-II forgeries and reduce it to the existential unforgeability of the signature scheme Sig. Let $\mathcal{B}$ be an adversary against the unforgeability of the ring signature, using $\mathcal{B}$, we can construct an adversary $\mathcal{A}_3$ that successfully breaks the unforgeability of the underlying signature scheme Sig.

$\mathcal{A}_3$ starts by generating the public parameters $\mathsf{param}_{\mathsf{RS}}$ where it instantiates the GS CRS in the soundness setting. Adversary $\mathcal{A}_3$ also generates n pairs of keys, where $n = n(\lambda)$ is an upper bound on the number of honest signers $\mathcal{B}$ uses in its game. $\mathcal{A}_3$ randomly chooses $i^* \in [1, n]$ and generates all the keys for $i \in [1, n] \setminus \{i^*\}$ normally with the exception that it sets pk_{i^*} to be the public key it gets from its own unforgeability game and hence $\mathcal{A}_3$ does not know the corresponding secret key sk_{i^*}. $\mathcal{A}_3$ passes $\mathsf{param}_{\mathsf{RS}}$ and $\mathcal{S} := \{\mathsf{pk}_1, \ldots, \mathsf{pk}_n\}$ to $\mathcal{B}$.

When $\mathcal{B}$ asks for a corrupt query on any $i \neq i^*$, $\mathcal{A}_3$ simply reveals the corresponding secret key sk_i. If $\mathcal{B}$ asks a corrupt query on $i = i^*$, $\mathcal{A}_3$ aborts. This happens with probability $1/n$.

For sign queries made by $\mathcal{B}$ for any signer Signer_i where $i \neq i^*$, $\mathcal{A}_3$ generates the ring signature itself by using the corresponding secret key and constructing the rest of the ring signature. On the other hand, if the query is for Signer_{i^*}, $\mathcal{A}_3$ queries its sign oracle on $\mathcal{H}(m, \mathcal{R})$ and then generates the rest of the ring signature itself.

Eventually, $\mathcal{B}$ terminates by outputting a message m^*, a ring signature Σ^* and a ring $\mathcal{R}^*$. $\mathcal{A}_3$ first verifies that all members in $\mathcal{R}^*$ are indeed all honest, that it never answered a sign query on message m^* w.r.t. the ring $\mathcal{R}^*$. If all conditions are satisfied, $\mathcal{A}_3$ uses the GS extraction key to extract the public key pk^* and the signature σ^* from the proof. With probability $1/n$, the extracted public key will be for the signer $\mathcal{A}_3$ has guessed. If this is not the case, $\mathcal{A}_3$ aborts. Otherwise, $\mathcal{A}_3$ returns $\mathcal{H}(m^*, \mathcal{R}^*)$ and σ^* as a successful forgery in its game.

Thus, we have

$$\mathsf{Adv}^{\text{Unforg}}_{\mathsf{RS},\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\text{Sound}}_{\mathsf{GS},\mathcal{A}_1}(\lambda) + \mathsf{Adv}^{\text{Coll}}_{\mathcal{H},\mathcal{A}_2}(\lambda) + n(\lambda) \cdot \mathsf{Adv}^{\text{Unforg}}_{\mathsf{Sig},\mathcal{A}_3}(\lambda),$$

which concludes the proof.