Efficient Ring Signatures in the Standard Model

Giulio Malavolta and Dominique Schröder

Friedrich-Alexander-University Erlangen-Nürnberg

Abstract. A ring signature scheme allows one party to sign messages on behalf of an arbitrary set of users, called the ring. The anonymity of the scheme guarantees that the signature does not reveal which member of the ring signed the message. The ring of users can be selected "on-the-fly" by the signer and no central coordination is required. Ring signatures have made their way into practice in the area of privacy-enhancing technologies and they build the core of several cryptocurrencies. Despite their popularity, almost all ring signature schemes are either secure in the random oracle model or in the common reference string model. The only candidate instantiations in the plain model are either impractical or not fully functional.

In this work, we close this gap by proposing a new construction paradigm for ring signatures without random oracles: We show how to efficiently instantiate full-fledged ring signatures from signature schemes with re-randomizable keys and non-interactive zero-knowledge. We obtain the following results:

- The first almost practical ring signature in the plain model from standard assumptions in bilinear groups.
- The first efficient ring signature in the plain model secure under a generalization of the knowledge of exponent assumption.

1 Introduction

Ring signatures were envisioned by Rivest, Shamir, and Tauman [36] as a tool to leak a secret information in an authenticated manner while being anonymous in within a crowd of users. The idea behind this primitive is that a signer can choose a set of users via their public-keys and sign a message on behalf of this set, also called a ring. Signing in the name of the users means that it is infeasible to tell which of the users signed the message. Ring signatures guarantee great flexibility: Rings can be formed arbitrarily and "on-the-fly" by the signer and no trusted authority is required. In fact, users do not even have to be aware of each other. The widely accepted security notions of anonymity against full key exposure and unforgeability with respect to insider corruption were formalized by Bender, Katz, and Morselli [4].

In the past years many applications of ring signatures were suggested, such as the ability to leak secrets while staying anonymous within a certain set [36]. Recently, certain types of ring signatures made it into practice being a building block in the cryptocurrency Monero. In Monero, to spend a certain amount of coins, the user searches for other public-keys sharing the same amount and it issues a ring signature for this set of users. Since the ring signature is anonymous, nobody can tell which of the users in the ring spent their coin.

Despite being one of the classical problems of cryptography and being deployed in practice, almost all ring signatures are either secure in the random oracle model or the common reference string model. Even worse, among the construction without random oracles, the asymptotically most efficient instance is the scheme of Bose, Das, and Rangan [8] with a signature of 95 group elements for a composite order bilinear group. Notable exceptions to what discussed above are the scheme of Bender, Katz, and Morselli [4] and the one of Chow et al. [19]. However, the former is only a feasibility result that relies on generic ZAPs [23], whereas the latter supports only rings of *constant size* and it is secure against a tailored assumption.

In this work, we close this gap presenting a new generic framework to construct efficient ring signatures without random oracles: Our abstraction gives us the first efficient scheme secure in the plain model. As a corollary of our transformation, we also obtain the most efficient scheme with constant size signatures in the common reference string model

1.1 Our Contribution

At the core of our contributions is a novel generic construction of ring signatures from signatures with re-randomizable keys, a property that was recently leveraged by Fleischhacker et al. [26] to build efficient sanitizable signature scheme [12,13,35]. In addition to that, our scheme requires a non-interactive zero-knowledge proof. Our generic transformation is secure in the common reference string model, without random oracles. In the process of instantiating our construction we propose a modification to the signature scheme of Hofheinz and Kiltz [34] that significantly simplifies the statement to be proven in zero-knowledge, thereby boosting the efficiency of our ring signature[1]. A nice feature of our transformation is that the resulting signature size does not depend on the size of the ring, except for the statement to be proven in zero-knowledge. Therefore, instantiating the zero-knowledge proof with SNARKs automatically yields a ring signature of constant size. As an example, we can implement the proof of knowledge with the scheme recently proposed by Groth [31], which adds only three group elements to our signatures.

Ring Signatures in the Plain Model. Our next observation is that, if the zero-knowledge scheme uses a common reference string with some homomorphic properties, then we can include a different string in each user's verification key. The proof for a ring of users is then computed against a combination of all the corresponding strings, that results in a correctly distributed reference string. This effectively lifts the resulting scheme to the plain model, given a suitable zero-knowledge protocol. We show that the scheme of Groth [29] partially satisfies our

[1] We choose Hofheinz-Kiltz signatures for efficiency reasons, although our transformation can also be instantiated from Waters' scheme [41], whose hardness relies on the Computational Diffie-Hellman (CDH) assumption.

constraints and we demonstrate how to integrate it in our system. The resulting ring signature scheme relies on standard assumptions for bilinear groups and it has somewhat efficient signature size, although still large for practical usage. The caveat here is that the scheme achieves only a weak form of anonymity.

Achieving Efficiency and Full Security. The last step towards our main result is a novel instantiation of a zero-knowledge proof system for proving the knowledge of discrete logarithms. The scheme relies on asymmetric bilinear groups and its efficiency is comparable to schemes derived from the Fiat-Shamir heuristic, although it does not use random oracles. We prove the security of our scheme under a generalization of the knowledge of exponent assumption and confirm its hardness in the generic group model [40]. When combined with our variant of the scheme of Hofheinz and Kiltz, the resulting ring signature is fully secure in the standard model (by using a similar trick as described above) and the signatures are composed by roughly 4 group elements per user in the ring.

A comparison of our results with existing schemes is summarized in Table 1. Our construction instantiated with [31] improves the signature size by and order of magnitude with respect to the most efficient scheme without random oracles. The scheme of [4] relies on generic ZAPs and therefore statements need to go through an NP-reduction before being proven, making it hard for us to estimate the real cost of the resulting signature. Although the ring signature of [19] has a very small signature size, it only supports rings of constant size, hindering its the practical deployment. Our two instantiations in the standard model offer a tradeoff between efficiency and assumptions, broadening the landscape of instances without any setup.

Ring Signature	Model	Anon.	Unforg.	Assumptions	Ring Size	Signature Size
[39]	crs	✓	✓	CDH + SubD	$\mathsf{poly}(\lambda)$	$(2n+2)\mathbb{G}$
[9]	crs	✓	✓	$(q,\ell,1)$-Pluri-SDH	$\mathsf{poly}(\lambda)$	$(n+1)\mathbb{G} + (n+1)\mathbb{F}_p$
[37]	crs	Basic	Sub.	CDH	$\mathsf{poly}(\lambda)$	$(n+1)\mathbb{G}$
[16]	crs	✓	✓	SDH + SubD	$\mathsf{poly}(\lambda)$	$O(\sqrt{n})$
[8]	crs	✓	✓	q-SDH + SXDH + SQROOT	$\mathsf{poly}(\lambda)$	$92\mathbb{G} + 3\mathbb{Z}_p$
This work + [31]	crs	✓	✓	q-SDH + GGM	$\mathsf{poly}(\lambda)$	$6\mathbb{G} + \mathbb{Z}_p$
[19]	std	✓	✓	(q,n)-DsjSDH	$O(1)$	$n\mathbb{G} + n\mathbb{Z}_p$
[4]	std	✓	✓	Enc + ZAP	$\mathsf{poly}(\lambda)$	$O(n)$
This work + [29]	std	Basic	✓	q-SDH + DLIN	$\mathsf{poly}(\lambda)$	$\sim 10^3 n\mathbb{G}$
This work	std	✓	✓	q-SDH + L-KEA	$\mathsf{poly}(\lambda)$	$(4n+3)\mathbb{G} + \mathbb{Z}_p$

Table 1: Comparison amongst ring signature schemes without random oracles

On the Knowledge of Exponent Assumption. Although the knowledge of exponent assumption is clearly non-standard, we believe that it is slightly better than assuming the existence of random oracles - at least from a theoretical point of view. The reason is that it is well known that the random oracle is not sound [15], whereas it might be possible that certain assumptions hold in practice. As an example, the SNARKs used in the real-world cryptocurrency Zerocash [3] rely on a variant of the knowledge-of-exponent assumptions.

1.2 Related Work

The notion of ring signatures has been introduced in the visionary work of Rivest, Shamir and Tauman [36], as a way to leak secrets while staying anonymous within the crowd. The authors proposed a construction based on trapdoor permutations and several other schemes have followed based on different assumptions such as discrete logarithms [7], bilinear maps [33], factoring [22], and hybrids [2]. Remarkably, the size of the signatures in the scheme of [22] is independent from the size of the ring. Such a surprising result is achieved with a clever usage of RSA accumulators [14] and the Fiat-Shamir transformation. A practical scheme constructed from a combination of Σ protocols and the Fiat-Shamir heuristic was recently proposed by Groth and Kohlweiss in [32]. The security of all of the aforementioned constructions is based on the existence of random oracles.

There has been a considerable effort in the community in building ring signature schemes from more classical assumptions. In particular, we know how to instantiate ring signatures efficiently admitting the existence of a common reference string model: Shacham and Waters [39] proposed the first efficient scheme in composite order groups with a pairing, whose performance were later on improved in the work of Boyen on mesh signatures [9]. Recently, Schäge and Schwenk [37] have shown a very efficient instantiation based on CDH-hard groups with extremely appealing signatures size, but at the cost of a weaker notion of unforgeability (chosen subring attacks in the terminology of [4]). Derler and Slamanig [21] suggested an efficient linear-size scheme from key-homomorphic signatures and zero-knowledge proofs. The first scheme with sublinear size signatures has been proposed by Chandran, Groth, and Sahai [16], which exhibits signatures that grow linearly with the square root of the size of the ring. To the best of our knowledge, the most (asymptotically) efficient construction without random oracles is due to Bose, Das, and Rangan [8], where a signature accounts for 95 group elements for a composite order bilinear group. We shall mention that in the work on signatures of knowledge [17] the authors claim that one can use their primitive combined with the techniques of Dodis et. al [22] to construct ring signatures of constant size, but this is not supported by any formal analysis. For fairness, we must say that the security model of ring signatures was not yet well established, as the seminal work of Bender, Katz and Morselli [4] has been published concurrently in the same year.

In contrast to the common reference string settings, ring signature schemes in the plain model (without any setup assumption) have been surprisingly understudied. The first work that considered this problem was [4], where the authors proposed a construction from non-interctive ZAPs [23]. Such a scheme represents the first proof of feasibility of ring signature schemes in the standard model. Concurrently, Chow et al. [19] published a scheme for constant-size rings based on a custom assumption. At today, these two instantiations were the only known candidates for a ring signature scheme in the plain model.

A notion related to ring signatures is that of group signatures, originally envisioned by Chaum and Van Heyst [18]: The main difference here is that a group manager controls the enrolment within the group of users and can revoke anonymity. Efficient realizations are known in the random oracle model [5] and in the standard model [10]. The absence of a trusted authority in ring signatures, makes the two primitives incomparable.

2 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter and by $\mathsf{poly}(\lambda)$ any function that is bounded by a polynomial in λ. We denote any function that is *negligible* in the security parameter by $\mathsf{negl}(\lambda)$. We say that an algorithm is PPT if it is modelled as a probabilistic Turing machine whose running time is bounded by some function $\mathsf{poly}(\lambda)$. Given a set S, we denote by $x \leftarrow S$ the sampling of and element uniformly at random in S.

2.1 Bilinear Maps

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of large prime order p. Let $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ be respective generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $e : \mathbb{G}_1 \times \mathbb{G}_2$ be a function that maps pairs of elements $\in (\mathbb{G}_1, \mathbb{G}_2)$ to elements of some cyclic group $\mathbb{G}_T$ of order p. Throughout the following sections we write all of the group operations mutiplicatively, with identity elements denoted by 1. We further require that:

- The map e and all the group operations in $\mathbb{G}_2$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are efficiently computable.
- The map e is non degenerate, i.e., $e(g_1, g_2) \neq 1$.
- The map e is bilinear, i.e., $\forall u \in \mathbb{G}_1, \forall v \in \mathbb{G}_2, \forall (a,b) \in \mathbb{Z}^2, e(u^a, v^b) = e(u,v)^{ab}$.
- There exists an efficiently computable function $\phi : \mathbb{G}_1 \to \mathbb{G}_2$ such that $\forall (u,v) \in \mathbb{G}_1$ it holds that $\phi(u \cdot v) = \phi(u) \cdot \phi(v)$.

2.2 Ring Signatures

In the following we recall the notion of ring signatures. Our definitions follow the strongest security model proposed by Bender, Katz, and Morselli [4].

Definition 1 (Ring Signature). *A ring signature scheme* $\mathsf{RSig} = (\mathsf{RGen}, \mathsf{RSig}, \mathsf{RVer})$ *is a triple of the following* PPT *algorithms:*

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{RGen}(1^\lambda)$: *On input the security parameter* 1^λ *outputs a key pair* $(\mathsf{sk}, \mathsf{vk})$.

$\sigma \leftarrow \mathsf{RSig}(\mathsf{sk}, m, R)$: *On input a secret key* sk*, a message* m*, and a ring* $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$*, outputs a signature* σ.

$b \leftarrow \mathsf{RVer}(R, \sigma, m)$: *On input a ring* $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$*, a signature* σ *and a message* m *outputs a bit* b.

For completeness we require that for all $\lambda \in \mathbb{N}$, for all $\{(\mathsf{sk}_i, \mathsf{vk}_i)\}_{i \in n}$ output by $\mathsf{RGen}(1^\lambda)$, any $i \in \{1, \ldots, n\}$, and any m, we have that $\mathsf{RVer}(R, \mathsf{RSig}(\mathsf{sk}_i, m, R), m) = 1$, where $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$.

Security of Ring Signatures. The security of a ring signature scheme is captured by the notions of anonymity against full key exposure and unforgeability with respect to insider corruption. We refer to [4] for a comprehensive discussion on the matter. In the following definitions, we assume without loss of generality that the adversary never submits a query where the ring consists only of maliciously generated keys.

Definition 2 (Anonymity). *Let ℓ be a polynomial function, $\mathcal{A}$ a* PPT *adversary, and* $\mathsf{RSig} = (\mathsf{RGen}, \mathsf{RSig}, \mathsf{RVer})$ *a ring signature scheme. Consider the following game:*

1. *For all $i \in \{1, \ldots \ell(\lambda)\}$ the challenger runs $(\mathsf{sk}_i, \mathsf{vk}_i) \leftarrow \mathsf{RGen}(1^\lambda; \omega_i)$, recording each randomness ω_i. The adversary $\mathcal{A}$ is provided with the verification keys $(\mathsf{vk}_1, \ldots, \mathsf{vk}_{\ell(\lambda)})$.*
2. *$\mathcal{A}$ is allowed to make queries of the form (j, R, m), where m is the message to be signed, R is a set of verification keys and j is and index such that $\mathsf{vk}_j \in R$. The challenger responds with $\mathsf{RSig}(\mathsf{sk}_j, m, R)$.*
3. *$\mathcal{A}$ requests a challenge by sending the tuple (i_0, i_1, R, m), where i_0 and i_1 are indices such that $(\mathsf{vk}_{i_0}, \mathsf{vk}_{i_1}) \in R$. The challenger samples a random bit $b \leftarrow \{0, 1\}$ and sends $\mathsf{RSig}(\mathsf{sk}_{i_b}, m, R)$ to $\mathcal{A}$. Additionally, the adversary is provided with the randomnesses $(\omega_1, \ldots, \omega_{\ell(\lambda)})$.*
4. *$\mathcal{A}$ outputs a guess b' and succeeds if $b' = b$.*

A ring signature scheme RSig *achieves anonymity if, for all* PPT *$\mathcal{A}$ and for all polynomial functions ℓ, the success probability of $\mathcal{A}$ in the above experiment is negligibly close to* $1/2$.

Definition 3 (Unforgeability). *Given a ring signature scheme* $\mathsf{RSig} = (\mathsf{RGen}, \mathsf{RSig}, \mathsf{RVer})$*, a polynomial function ℓ, and a* PPT *adversary $\mathcal{A}$, consider the following game:*

1. *For all $i \in \{1, \ldots, \ell(\lambda)\}$ the challenger runs $(\mathsf{sk}_i, \mathsf{vk}_i) \leftarrow \mathsf{RGen}(1^\lambda; \omega_i)$. The adversary $\mathcal{A}$ is provided with the verification keys $\mathbf{vk} := (\mathsf{vk}_1, \ldots, \mathsf{vk}_{\ell(\lambda)})$. Additionally, the challenger initializes an empty list of corrupted users $\mathcal{C}$.*
2. *$\mathcal{A}$ is allowed to make signatures and corruption queries. A signature query is of the form (j, R, m), where m is the message to be signed, R is a set of verification keys and j is and index such that $\mathsf{vk}_j \in R$. The challenger responds with $\mathsf{RSig}(\mathsf{sk}_j, m, R)$. A corruption query is of the form $j \in \{1, \ldots, \ell(\lambda)\}$. The challenger sends sk_j to $\mathcal{A}$ and appends vk_j to $\mathcal{C}$.*
3. *$\mathcal{A}$ outputs a tuple (R^*, m^*, σ^*) and wins if it never made any signing query of the form $(\cdot, R^*, m^*)$ and $R^* \subseteq \mathbf{vk} \setminus \mathcal{C}$ and $\mathsf{RVer}(R^*, \sigma^*, m^*) = 1$.*

A ring signature scheme RSig *achieves unforgeability if, for all* PPT *$\mathcal{A}$ and for all polynomial functions ℓ, the success probability of $\mathcal{A}$ in the above experiment is negligible.*

2.3 Non-Interactive Zero-Knowledge

We recall the definitions and the security properties of non-interactive zero-knowledge proof systems as defined in [29].

Definition 4 (Non-Interactive Zero-Knowledge Proof System). *Let $\mathcal{L}$ be an NP language and let $\mathcal{R}$ be the corresponding relation. A non-interactive zero-knowledge proof system* NIZK *consists of the following* PPT *algorithms:*

$\mathsf{crs} \leftarrow \mathcal{G}(1^\lambda)$: *The setup algorithm takes as input the security parameter 1^λ and generates a random common reference string* crs *and a trapdoor α.*

$\pi \leftarrow \mathcal{P}(\mathsf{crs}, w, x)$: *The prover algorithm takes as input the common reference string* crs*, a statement x, and a witness w and outputs a zero-knowledge proof π.*

$b \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi)$: *The verifier algorithm takes as input the common reference string* crs*, a statement x, and a proof π and returns either 0 or 1.*

Definition 5 (Completeness). *A* NIZK *system is* complete *if for all $\lambda \in \mathbb{N}$ and all $x \in \mathcal{L}$ it holds that*

$$\mathsf{Pr}\big[(\mathsf{crs}, \alpha) \leftarrow \mathcal{G}(1^\lambda), \pi \leftarrow \mathcal{P}(\mathsf{crs}, w, x) : 1 \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi)\big] = 1.$$

Soundness, zero-knowledge and proof of knowledge are defined in the following.

Definition 6 (Soundness). *A* NIZK *system is* computationally sound *if for all $\lambda \in \mathbb{N}$ and all* PPT *adversaries $\mathcal{A}$ there exists a negligible function* negl *such that*

$$\mathsf{Pr}\big[(\mathsf{crs}, \alpha) \leftarrow \mathcal{G}(1^\lambda); (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}); 1 \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi) \mid x \notin \mathcal{L}\big] \leq \mathsf{negl}(n).$$

Definition 7 (Zero-Knowledge). *An* NIZK *system is* statistically zero-knowledge *if there exists a* PPT *simulator $\mathcal{S}$ and a negligible function such that for all $\lambda \in \mathbb{N}$, for all $x \in \mathcal{L}$ with witness w, and for $(\mathsf{crs}, \alpha) \leftarrow \mathcal{G}(1^\lambda)$ it holds that*

$$\mathcal{P}(\mathsf{crs}, w, x) \approx \mathcal{S}(\mathsf{crs}, \alpha, x).$$

where $\approx$ denotes statistical indistinguishability.

Additionally, we say that a NIZK system is *unconditionally* zero-knowledge if the condition above holds for any choice of (crs, α).

Definition 8 (Argument of Knowledge). *A* NIZK *system is an* argument of knowledge *if for all $\lambda \in \mathbb{N}$ and for all* PPT *adversaries $\mathcal{A}$ there exists a* PPT *extractor $\mathcal{E}$ running on the same random tape of $\mathcal{A}$ and a negligible function* negl *such that*

$$\mathsf{Pr}\left[\begin{array}{r} (\mathsf{crs}, \alpha) \leftarrow \mathcal{G}(1^\lambda), (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}), \\ (x, \pi, w) \leftarrow \mathcal{E}(\mathsf{crs}, \alpha) : (x, w) \in \mathcal{R} \end{array} \middle| \mathcal{V}(\mathsf{crs}, x, \pi) = 1\right] \geq (1 - \mathsf{negl}(\lambda)).$$

Additionally, we call a proof *succinct* if the size of the proof is constant, in particular it must be independent from statement to be proven and corresponding witness. In literature this primitive is known as succinct non-interactive argument of knowledge (SNARK) and several efficient realizations are known to exist without random oracles, among the others see [31,30].

2.4 Signatures with Re-Randomizable Keys

We recall the notion of signatures with re-randomizable keys, as defined in [26]. This primitive allows one to consistently re-randomize private and public keys of a signature scheme.

Definition 9 (Signatures with Re-Randomizable Keys). *A digital signature scheme* $\mathsf{Sig} = (\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{RndSK}, \mathsf{RndVK})$ *with perfectly re-randomizable keys is composed by the following* PPT *algorithms:*

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SGen}(1^\lambda)$: *The key generation algorithm takes as input the security parameter* 1^λ *and generates a key pair* $(\mathsf{sk}, \mathsf{vk})$*.*

$\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}, m)$: *The signing algorithm takes as input a signing key* sk *and a message* m *and outputs a signature* σ*.*

$b \leftarrow \mathsf{Ver}(\mathsf{vk}, m, \sigma)$: *The verification algorithm takes as input a verification key* vk*, a message* m*, and a candidate signature* σ *and outputs a bit* b*.*

$\mathsf{sk}' \leftarrow \mathsf{RndSK}(\mathsf{sk}, \rho)$: *The secret key re-randomization algorithm takes as input a signing key* sk *and a randomness* ρ *and outputs a new signing key* sk'*.*

$\mathsf{vk}' \leftarrow \mathsf{RndVK}(\mathsf{vk}, \rho)$: *The public key re-randomization algorithm takes as input a verification key* vk *and a randomness* ρ *and outputs a new verification key* vk'*.*

The scheme is *complete* if for all $\lambda \in \mathbb{N}$, all key-pairs $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SGen}(1^\lambda)$, all messages m we have that $\mathsf{Ver}(\mathsf{vk}, m, \mathsf{Sig}(\mathsf{sk}, m)) = 1$. Additionally we require that for all ρ it holds that $\mathsf{Ver}(\mathsf{RndVK}(\mathsf{vk}, \rho), m, \mathsf{Sig}(\mathsf{RndSK}(\mathsf{sk}, \rho), m)) = 1$. The formal definition of re-randomizable keys follows.

Definition 10 (Re-Randomizable Keys). *A signature scheme* Sig *has* perfectly re-randomizable keys *if for all* $\lambda \in \mathbb{N}$*, all key-pairs* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SGen}(1^\lambda)$*, and a* ρ *chosen uniformly at random we have that the following distributions are identical:*

$$\{(\mathsf{sk}, \mathsf{vk}, \mathsf{sk}', \mathsf{vk}')\} = \{(\mathsf{sk}, \mathsf{vk}, \mathsf{sk}'', \mathsf{vk}'')\}$$

where $(\mathsf{sk}', \mathsf{vk}') \leftarrow \mathsf{SGen}(1^\lambda)$*,* $\mathsf{sk}'' \leftarrow \mathsf{RndSK}(\mathsf{sk}, \rho)$*, and* $\mathsf{vk}'' \leftarrow \mathsf{RndVK}(\mathsf{vk}, \rho)$*.*

The notion of existential unforgeability for signatures with re-randomizable keys is an extension of the standard definition where the attacker is provided with an additional oracle that signs messages under re-randomized keys. The formal definition follows.

Definition 11 (Unforgeability Under Re-Randomizable Keys). *Given a signature scheme* Sig *and a* PPT *adversary* $\mathcal{A}$*, consider the following game:*

1. *The challenger runs* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SGen}(1^\lambda)$ *and provides* $\mathcal{A}$ *with* vk*.*
2. $\mathcal{A}$ *is allowed to make signature and randomized signature queries. A signature query is for the form* $(m, \perp)$*, where* m *is the message to be signed, and the challenger responds with* $\mathsf{Sig}(\mathsf{sk}, m)$*. A randomized signature query is of the form* (m, ρ)*, where* m *is the message to be signed and* ρ *is a randomness. The challenger replies with* $\mathsf{Sig}(\mathsf{RndSK}(\mathsf{sk}, \rho), m)$*.*

3. $\mathcal{A}$ outputs a tuple (m^*, σ^*, ρ^*) and wins the game if it never made a query of the form $(m^*, \cdot)$ and either $\mathsf{Ver}(\mathsf{vk}, m^*, \sigma^*) = 1$ *or* $\mathsf{Ver}(\mathsf{RndVK}(\mathsf{vk}, \rho^*), m^*, \sigma^*) = 1$.

A signature scheme Sig *achieves* unforgeability under re-randomizable keys *if, for all* PPT $\mathcal{A}$ *, the success probability of $\mathcal{A}$ in the above experiment is negligible.*

3 Ring Signatures from Signatures with Re-Randomizable Keys

In this section we describe our framework for designing efficient ring signature schemes without random oracles. For the ease of the exposition and for a more modular presentation we propose a generic construction in the common reference string model. Jumping ahead, we will show later how to upgrade it to the standard model (i.e., without any setup assumptions) for some specific instantiations. The idea behind our construction based on signatures with re-randomizable keys is the following: The signer generates a randomized version of its own public key and signs the message under the secret key re-randomized with the same factor. The signature is then composed by the output of the signing algorithm and a disjunctive argument of knowledge of the randomization factor of the new verification key with respect to a ring of public keys. More formally, our building blocks are a signature scheme with re-randomizable keys Sig and a non-interactive zero-knowledge argument NIZK for the following language $\mathcal{L}$:

$$\mathcal{L} = \{(\mathsf{vk}_1, \ldots, \mathsf{vk}_n, \mathsf{vk}^*) : \exists(\rho, i) : \mathsf{vk}^* = \mathsf{RndVK}(\mathsf{vk}_i, \rho)\}.$$

Our ring signature scheme $\mathsf{RSig} = (\mathsf{RGen}, \mathsf{RSig}, \mathsf{RVer})$ is shown in Figure 1. The completeness of the scheme follows directly from the completeness of the zero knowledge argument and the signature scheme. The security analysis is elaborated below.

Theorem 1. *Let* NIZK *be a statistically zero-knowledge argument and let* $(\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{RndSK}, \mathsf{RndVK})$ *be a signature with perfectly re-randomizable keys, then the construction in Figure 1 is an anonymous ring signature scheme in the common reference string model.*

Proof. Let us consider the following sequence of hybrids:

H_0 : Is the original experiment as defined in Definition 2.

H_1 : Is defined as H_0 except that the proof π in the challenge signature is computed as $\pi \leftarrow \mathcal{S}(\mathsf{crs}, \alpha, x)$, where $x = R^* \| \mathsf{vk}_{i_b}$.

H_2 : Is defined as H_1 except that the challenge signature is substituted with the tuple $(\mathsf{Sig}(\mathsf{sk}', m^* \| R^*), \pi, \mathsf{vk}')$, where $(\mathsf{sk}', \mathsf{vk}') \leftarrow \mathsf{SGen}(1^\lambda)$.

We now argue about the indistinguishability of adjacent experiments.

$H_0 \approx H_1$. The two hybrids are identical except for the proof π that is honestly generated in H_0, while in H_1 it is computed by the simulator $\mathcal{S}$ of NIZK. By the perfect zero knowledge of NIZK, the two simulations are statistically close.

```
$\mathsf{RGen}(1^\lambda)$

$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SGen}(1^\lambda)$
return $(\mathsf{sk}, \mathsf{vk})$
```

```
$\mathsf{RSig}(\mathsf{sk}, m, R)$

parse $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$
if $\nexists i : \mathsf{vk} = \mathsf{vk}_i$
  return $\perp$
$\rho \leftarrow \{0,1\}^\lambda$
$\mathsf{vk}' \leftarrow \mathsf{RndVK}(\mathsf{vk}, \rho)$
$\mathsf{sk}' \leftarrow \mathsf{RndSK}(\mathsf{sk}, \rho)$
$x := R||\mathsf{vk}'$
$\pi \leftarrow \mathcal{P}(\mathsf{crs}, (\rho, i), x)$
$\sigma \leftarrow \mathsf{Sig}(\mathsf{sk}', m||R)$
return $(\sigma, \pi, \mathsf{vk}')$
```

```
$\mathsf{RVer}(R, \sigma, m)$

parse $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$
parse $\sigma = (\sigma', \pi, \mathsf{vk}')$
$x := R||\mathsf{vk}'$
$b \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi)$
$b' \leftarrow \mathsf{Ver}(\mathsf{vk}', (m||R), \sigma')$
return $(b = b' = 1)$
```

Fig. 1: A generic construction of a ring signature

$H_1 \approx H_2$. The two experiments differ only in the sampling procedure of the key pair $(\mathsf{sk}', \mathsf{vk}')$ used to compute the challenge signature. In the former case $(\mathsf{sk}', \mathsf{vk}')$ is the re-randomization of the pair $(\mathsf{sk}_{i_b}, \mathsf{vk}_{i_b})$, while in the latter the pair is freshly sampled. Since the signature scheme has perfectly re-randomizable keys, we can conclude that the two simulations are identical.

We observe that in the hybrid H_2 the computation of the challenge signature is completely independent from the bit b. Therefore the success probability of $\mathcal{A}$ in H_2 is exactly $1/2$. By the argument above we have that $H_0 \approx H_2$ and therefore we can conclude that any unbounded $\mathcal{A}$ cannot win the experiment with probability negligibly greater than guessing. □

We now need to show that our ring signature scheme achieves unforgeability against full key exposure. For our formal argument we need to assume the existence of an extractor for the NIZK scheme that is successful even in presence of a signing oracle. For the case of black-box extraction, such a property is trivially guaranteed by any construction. However, as shown in [25], for the case of non-blackbox extraction the definition itself does not necessarily cover this additional requirement. For a more comprehensive discussion on the matter we refer the reader to [25]. In order to be as generic as possible, we are going to explicitly assume the existence of a NIZK that is extractable also in presence of a signing oracle. As discussed in [25], such an assumption as been (implicitly) adopted in several seminal works such as [11,6,27,28].

Theorem 2. *Let* NIZK *be a computationally sound argument of knowledge that is extractable in presence of a signing oracle and let* Sig *be a signature with perfectly re-randomizable keys, then the construction in Figure 1 is an unforgeable ring signature scheme in the common reference string model.*

Proof. Assume towards contradiction that there exists a PPT adversary $\mathcal{A}$ that succeeds in the experiment of Definition 3 with probability $\epsilon(\lambda)$, for some non negligible function ϵ. Then we can construct the following reduction $\mathcal{R}$ against the unforgeability under re-randomizable keys of the signature scheme $(\mathsf{SGen}, \mathsf{Sig}, \mathsf{Ver}, \mathsf{RndSK}, \mathsf{RndVK})$.

$\mathcal{R}(1^\lambda, \mathsf{vk})$. On input the security parameter and the verification key vk the reduction samples an $i \leftarrow \{1, \ldots, \ell(\lambda)\}$ and sets $\mathsf{vk}_i = \mathsf{vk}$. For all $j \in \{1, \ldots, \ell(\lambda)\} \backslash i$, the reduction sets $(\mathsf{sk}_j, \mathsf{vk}_j) \leftarrow \mathsf{SGen}(1^\lambda)$. Upon a corruption query from $\mathcal{A}$ on index $k \neq i$, the reduction sends sk_k to $\mathcal{A}$, if $k = i$ then $\mathcal{R}$ aborts. Signing queries of the form (k, R, m), for $k \neq i$ are handled as specified in the experiment. On the other hand, for queries of the form (i, R, m), the reduction samples a random ρ, and computes $\mathsf{vk}' \leftarrow \mathsf{RndVK}(\mathsf{vk}, \rho)$ and $\pi \leftarrow \mathcal{P}(\mathsf{crs}, (\rho, i), R \| \mathsf{vk}')$. Then it queries the signing oracle on input $(m||R, \rho)$. Let σ be the response of the challenger, $\mathcal{R}$ returns $(\sigma, \pi, \mathsf{vk}')$ to $\mathcal{A}$. At some point of the execution $\mathcal{A}$ outputs a tuple (R^*, m^*, σ^*). $\mathcal{R}$ parses $\sigma^* = (\theta^*, \pi^*, \mathsf{vk}^*)$ and runs $(R^* \| \mathsf{vk}^*, w^*, \pi^*) \leftarrow \mathcal{E}_\mathcal{A}$ on the same inputs and random tape of $\mathcal{A}$. $\mathcal{R}$ parses $w^* = (\rho^*, i^*)$ and aborts if $i^* \neq i$. Otherwise $\mathcal{R}$ returns the tuple $(m^* \| R^*, \theta^*, \rho^*)$ and interrupts the simulation.

Since $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ are PPT machines it follows that $\mathcal{R}$ runs in polynomial time. Let us assume for the moment that $i = i^*$. In this case the successful $\mathcal{A}$ never queries the corruption oracle on i, since $\mathsf{vk}_i \in R^*$ (by the soundness of the NIZK) and $R^* \subseteq \mathbf{vk} \setminus \mathcal{C}$. Therefore the reduction does not aborts. It is also easy to see that all the signing queries on i are answered correctly by the reduction since the signing oracle of the challenger returns some signature $\sigma \leftarrow \mathsf{Sig}(\mathsf{RndSK}(\mathsf{sk}_i, \rho), m||R)$, which is exactly what $\mathcal{A}$ is expecting. It follows that, for the case $i = i^*$, $\mathcal{R}$ correctly simulates the inputs for $\mathcal{A}$. We now show that a successful forgery by $\mathcal{A}$ implies a successful forgery by $\mathcal{R}$. Let (R^*, m^*, σ^*) be the outputs of $\mathcal{A}$, by the winning condition of the unforgeability experiment we have that $\mathsf{RVer}(R^*, m^*, \sigma^*) = 1$. Let $\sigma^* = (\theta^*, \pi^*, \mathsf{vk}^*)$, this implies that $\mathcal{V}(\mathsf{crs}, R^* \| \mathsf{vk}^*, \pi^*) = 1$ and that $\mathsf{Ver}(\mathsf{vk}^*, m^* \| R^*, \theta^*) = 1$. Since the extractor is successful with overwhelming probability we can rewrite $\mathsf{vk}^* = \mathsf{RndVK}(\mathsf{vk}_{i^*}, \rho^*) = \mathsf{RndVK}(\mathsf{vk}_i, \rho^*)$, for the case $i^* = i$. It follows that $\mathsf{Ver}(\mathsf{RndVK}(\mathsf{vk}_i, \rho^*), m^* \| R^*, \theta^*) = 1$. Recall that $\mathsf{vk}_i = \mathsf{vk}$, which means that $(m^* \| R^*, \theta^*, \rho^*)$ is a valid signature under vk. Since $\mathcal{A}$ is required to never query the signing oracle on $(\cdot, R^*, m^*)$, then $\mathcal{R}$ never queried the challenger on some tuple $(m^* \| R^*, \cdot)$. We can conclude that, if $i^* = i$, $\mathcal{R}$ returns a valid forgery with the same probability of $\mathcal{A}$. That is

$$\begin{aligned}
\mathsf{Pr}[\mathcal{R} \text{ wins}] &= \mathsf{Pr}[\mathcal{R} \text{ wins} | i^* = i] \, \mathsf{Pr}[i^* = i] + \mathsf{Pr}\left[\mathcal{R} \text{ wins} \middle| \overline{i^* = i}\right] \mathsf{Pr}\left[\overline{i^* = i}\right] \\
&\geq \mathsf{Pr}[\mathcal{R} \text{ wins} | i^* = i] \, \mathsf{Pr}[i^* = i] \\
&\gtrsim \frac{\epsilon(\lambda)}{\ell(\lambda)},
\end{aligned}$$

which is a non-negligible probability. This represents a contradiction to the existential unforgeability of signatures with re-randomizable keys and concludes our proof. □

4 Bilinear Groups Instantiation

With the goal of an efficient standard model scheme in mind, in this section we show how to efficiently instantiate the construction presented in Section 3 in prime order groups with an efficiently computable pairing. As it was shown in [26], signatures with perfectly re-randomizable keys are known to exist in the standard model due to a construction by Hofheinz and Kiltz [34]. We proceed by describing a slightly modified version of the scheme and then we show how to deploy it in our generic framework.

4.1 A Variation of [34]

We recall the digital signature scheme from Hofheinz and Kiltz [34] as described in [26]. The scheme assumes the existence of a group with an efficiently computable bilinear map and a programmable hash function $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ with domain $\{0,1\}^*$ and range $\mathbb{G}_1$. The common reference string contains the key k of the programmable hash function H, that is assumed to be honestly generated. Signing keys are random elements $\mathsf{sk} \in \mathbb{Z}_p$ and verification keys are of the form $\mathsf{vk} = g_2^{\mathsf{sk}}$. To compute a signature on a message m, the signer returns $\left(s, y = \mathsf{HEval}(k,m)^{\frac{1}{\mathsf{sk}+s}}\right)$, where s is a randomly chosen element of $\mathbb{Z}_p$. The verification of a signature (s,y) consists of checking whether $e\left(y, \mathsf{vk} \cdot g_2^s\right) = e\left(\mathsf{HEval}(k,m), g_2\right)$. Keys can be efficiently re-randomized by computing $\mathsf{sk}' = \mathsf{sk} + \rho \bmod p$ and $\mathsf{vk}' = \mathsf{vk} \cdot g_2^{\rho}$, respectively. The scheme is shown to be existentially unforgeable under re-randomizable key under the q-strong Diffie-Hellman assumption in [26], in the common reference string model.

Towards More Efficient Ring Signatures. We now propose a slight modification of the scheme that allows us for a more efficient instantiation of our generic ring signature scheme. The changes are minimal: We introduce an additional randomness in the signing algorithm that is used in the computation of $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$. Such a randomness is included in the plain signature and it is used in the verification algorithm to check the validity of the output of the algorithm HEval. Our variation of the signature scheme can be found in Figure 2. The correctness of the scheme is trivial to show. Note that the keys of our construction are identical to the ones of the signature scheme in [34], therefore the scheme has also perfectly re-randomizable keys. What is left to be shown is that signatures are still unforgeable.

Theorem 3. *Let $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ be a programmable hash function, then the construction in Figure 2 is unforgeable under re-randomizable keys under the q-strong Diffie-Hellman assumption.*

Proof. Our proof strategy consists in showing that a forgery in our signature scheme implies a forgery in the scheme of Hofheinz and Kiltz [34]. Since the scheme is proven to be secure against the q-strong Diffie-Hellman assumption, the theorem follows. More formally, assume that there exists an attacker that

```
$\mathsf{SSetup}(1^\lambda)$
    $k \leftarrow \mathsf{HGen}(1^\lambda)$
    return $k$

$\mathsf{SGen}(1^\lambda)$
    $x \leftarrow \mathbb{Z}_p$
    return $(x, g_2^x)$

$\mathsf{Sig}(\mathsf{sk}, m)$
    $s \leftarrow \mathbb{Z}_p$
    $\delta \leftarrow \mathbb{Z}_p$
    $c \leftarrow \mathsf{HEval}(k, m)^\delta$
    $y \leftarrow c^{\frac{1}{\mathsf{sk}+s}}$
    return $(s, y, c, \delta)$

$\mathsf{Ver}(\mathsf{vk}, \sigma, m)$
    parse $\sigma = (s, y, c, \delta)$
    if $\mathsf{HEval}(k, m)^\delta = c$ return
        $e(y, \mathsf{vk} \cdot g_2^s) = e(c, g_2)$

$\mathsf{RndSK}(\mathsf{sk}, \rho)$
    return $\mathsf{sk} + \rho$

$\mathsf{RndVK}(\mathsf{vk}, \rho)$
    return $(\mathsf{vk} \cdot g_2^\rho, k)$
```

Fig. 2: Our variation of the [34] signature scheme

succeeds in the experiment as described in Definition 11 with probability $\epsilon(\lambda)$, for some non-negligible function ϵ. Then we can construct the following reduction against the unforgeability of the scheme [34].

$\mathcal{R}(1^\lambda, \mathsf{vk})$. On input vk the reduction samples $k \leftarrow \mathsf{HGen}(1^\lambda)$ and forwards (vk, k) to $\mathcal{A}$. The adversary is allowed to issue randomized signature queries of the form (m, ρ). $\mathcal{R}$ forwards (m, ρ) to its own oracle and receives (s, y) as a response. Then it samples a random $\delta \leftarrow \mathbb{Z}_p$ and hands over $\left(s, y^\delta, \mathsf{HEval}(k, m)^\delta, \delta\right)$ to $\mathcal{A}$. Standard signature queries are handled analogously. At some point of the execution the adversary outputs a challenge tuple $(m^*, \sigma^* = (s^*, y^*, c^*, \delta^*))$ and $\mathcal{R}$ returns $\left(m^*, \left(s^*, (y^*)^{\frac{1}{\delta^*}}\right)\right)$ to the challenger and terminates the execution.

The reduction is clearly efficient. To see that the queries of $\mathcal{A}$ are correctly answered it is enough to observe that the tuple

$$\begin{aligned}\left(s, y^\delta, \mathsf{HEval}(k, m)^\delta, \delta\right) &= \left(s, \left(\mathsf{HEval}(k, m)^{\frac{1}{s+(\mathsf{sk}+\rho)}}\right)^\delta, \mathsf{HEval}(k, m)^\delta, \delta\right)\\ &= \left(s, \mathsf{HEval}(k, m)^{\frac{\delta}{s+(\mathsf{sk}+\rho)}}, \mathsf{HEval}(k, m)^\delta, \delta\right)\\ &= \left(s, c^{\frac{1}{s+(\mathsf{sk}+\rho)}}, c, \delta\right)\end{aligned}$$

is a correctly distributed key if and only if (s, y) is correctly distributed. To conclude we need to show that $\left(m^*, \left(s^*, (y^*)^{\frac{1}{\delta^*}}\right)\right)$ is a valid forgery if and only if $(m^*, \sigma^* = (s^*, y^*, c^*, \delta^*))$ is a valid forgery. It is easy to see that since m^* was not queried by the adversary then it holds that m^* was not queried by the reduction either. Furthermore, by the winning condition of the experiment we have that

$$e\left(y^*, \mathsf{vk} \cdot g_2^{s^*}\right) = e(c^*, g_2),$$

which we can rewrite as

$$e\left((y^*)^{\frac{1}{\delta^*}}, \mathsf{vk} \cdot g_2^{s^*}\right) = e\left((c^*)^{\frac{1}{\delta^*}}, g_2\right).$$

Since it must be the case that $c^* = \mathsf{HEval}(k, m^*)^{\delta^*}$, then we have that

$$e\left((y^*)^{\frac{1}{\delta^*}}, \mathsf{vk} \cdot g_2^{s^*}\right) = e(\mathsf{HEval}(k, m^*), g_2),$$

which implies that $\left(m^*, \left(s^*, (y^*)^{\frac{1}{\delta^*}}\right)\right)$ is a valid message-signature tuple for the scheme in [34]. By initial assumption this happens with non-negligible probability $\epsilon(\lambda)$, which is a contradiction to the existential unforgeability of the scheme in [34] under re-randomizable keys and it concludes our proof. $\square$

4.2 A Ring Signature Scheme

Now we show how to deploy our scheme as described above in our framework for the construction of an efficient ring signature scheme. As our ultimate goal is to remove the common reference string, the first challenge in using the scheme of Figure 2 in our generic construction of Section 3 is that it assumes a trusted setup of a key k for a programable hash function. The natural strategy to solve this issue is to include the key k in the verification key and show a re-randomized version of k in the ring signature. However, this comes at the price of proving the knowledge of a randomization factor of k, which is typically very expensive. In fact, the most efficient instance of a programmable hash function is due to Waters [41] and the size of the key is linear in the security parameter. Instead, we leverage some non-blackbox properties of the scheme presented in Section 4.1 to re-randomize the output of the programmable hash function. Loosely speaking, this allows us to remove the evaluation of H from the statement to be proven in zero knowledge thus greatly improving the efficiency of the resulting scheme. A complete description of the resulting ring signature can be found in Figure 3. Here the language $\mathcal{L}$ of our proof system is defined as

$$\mathcal{L} = \left\{ \begin{array}{l} ((\mathsf{vk}_1, \ldots, \mathsf{vk}_n, \mathsf{vk}^*), (k_1, \ldots, k_n), c, m) \in \mathbb{G}_2^{n+1} \times \mathbb{G}_1^{\lambda \cdot n+1} \times \{0,1\}^* : \\ \quad \exists(\delta, \rho, i) : \frac{\mathsf{vk}^*}{\mathsf{vk}_i} = g_2^{\rho} \wedge c = \mathsf{HEval}(k_i, m)^{\delta} \end{array} \right\},$$

which is essentially a disjunctive proof of two discrete logarithms over two vectors of group elements.

Since the scheme in Figure 3 is not a direct instantiation of the construction discussed in Section 3, we shall prove that the construction satisfies the requirements for a ring signature scheme. First we show that our scheme is statistically anonymous.

Theorem 4. *Let* NIZK *be a statistically zero-knowledge argument, and let* $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ *be a programmable hash function, then the construction in Figure 3 is an anonymous ring signature scheme in the common reference string model.*

```
$\mathsf{RGen}(1^\lambda)$
$x \leftarrow \mathbb{Z}_p$
$k \leftarrow \mathsf{HGen}(1^\lambda)$
return $(x, (g_2^x, k))$
```

```
$\mathsf{RSig}(\mathsf{sk}, m, R)$
parse $R = (\mathsf{vk}_1, \dots, \mathsf{vk}_n)$
if $\nexists i : \mathsf{vk} = \mathsf{vk}_i$
    return $\perp$
parse $\mathsf{vk} = (z, k)$
$(s, \rho, \delta) \leftarrow \mathbb{Z}_p^3$
$z' \leftarrow z \cdot g_2^\rho$
$x' \leftarrow \mathsf{sk} + \rho$
$c \leftarrow \mathsf{HEval}(k, m||R)^\delta$
$x := R||z'||c||(m, R)$
$\pi \leftarrow \mathcal{P}(\mathsf{crs}, (\rho, \delta, i), x)$
$y \leftarrow c^{\frac{1}{x'}}$
$\sigma = (s, y, c)$
return $(\sigma, \pi, z')$
```

```
$\mathsf{RVer}(R, \sigma, m)$
parse $R = (\mathsf{vk}_1, \dots, \mathsf{vk}_n)$
parse $\mathsf{vk}_i = (z_i, k_i)$
parse $\sigma = (\sigma', \pi, z')$
parse $\sigma' = (s, y, c)$
$x := R||z'||c||(m, R)$
$b \leftarrow \mathcal{V}(\mathsf{crs}, x, \pi)$
$b' = 1$ if
    $e(y, \mathsf{vk}' \cdot g_2^s) = e(c, g_2)$
return $(b = b' = 1)$
```

Fig. 3: A ring signature scheme in the common reference string model

Proof. The first steps of the proof are identical to the ones of the proof of Theorem 1. We only need to introduce an extra hybrid H_3 where we substitute c^* with a random element in $\mathbb{G}_1$. For the indistinguishability $H_2 \approx H_3$ we argue that for all $m \in \{0,1\}^*$, for a random key $k \leftarrow \mathsf{HGen}(1^\lambda)$, and for a random $\delta \in \mathbb{Z}_p$, the element $\mathsf{HEval}(k, m)^\delta$ is uniformly distributed in $\mathbb{G}_1$, except with negligible probability. This is clearly the case as long as $\mathsf{HEval}(k, m) \neq 1$, which, in the construction of [41], happens only with negligible probability over the random choice of k. With this in mind, the argument follows along the same lines. □

Finally, we show that our scheme is unforgeable against full key exposure.

Theorem 5. *Let* NIZK *be a computationally sound argument of knowledge that is extractable in presence of a signing oracle and let* $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ *be a programmable hash function, then the construction in Figure 3 is an unforgeable ring signature scheme under the q-strong Diffie-Hellman assumption in the common reference string model.*

Proof. Our proof strategy is to reduce against the unforgeability of the scheme described in Figure 2. The proof outline is identical to the proof of Theorem 2, but there are some subtleties to address due to the non-blackbox usage of the signature scheme. More formally, assume towards contradiction that there exists a PPT adversary $\mathcal{A}$ that succeeds in the experiment of Definition 3 with probability $\epsilon(\lambda)$, for some non negligible function ϵ. Then we can construct the following reduction $\mathcal{R}$ against the unforgeability under re-randomizable keys of the signature scheme in Figure 2.

$\mathcal{R}(1^\lambda, (\mathsf{vk}, k))$. On input the security parameter and the key (vk, k) the reduction samples an $i \leftarrow \{1, \ldots, \ell(\lambda)\}$ and sets $\mathsf{vk}_i = (\mathsf{vk}, k)$. For all $j \in \{1, \ldots, \ell(\lambda)\} \backslash i$, the reduction executes $(\mathsf{sk}'_j, \mathsf{vk}'_j) \leftarrow \mathsf{SGen}(1^\lambda)$ (as defined in Figure 2) and $k_i \leftarrow \mathsf{HGen}(1^\lambda)$. $\mathcal{R}$ sets $\mathsf{vk}_j = (\mathsf{vk}'_j, k_j)$. Upon a corruption query from $\mathcal{A}$ on index $j \neq i$, the reduction sends sk_j to $\mathcal{A}$, if $j = i$ then $\mathcal{R}$ aborts. Signing queries of the form (j, R, m), for $j \neq i$ are handled as specified in the experiment. Upon queries of the form (i, R, m), the reduction sends $(m||R, \rho)$ to the signing oracle, for a random $\rho \in \mathbb{Z}_p$. $\mathcal{R}$ parses the response as (s, y, c, δ), sets $x := (R, \mathsf{vk} \cdot g_2^\rho, c, m\|R)$ and computes $\pi \leftarrow \mathcal{P}(\mathsf{crs}, (\rho, \delta, i), x)$. $\mathcal{A}$ is provided with the tuple $((s, y, c), \pi, \mathsf{vk} \cdot g_2^\rho)$. At some point of the execution $\mathcal{A}$ outputs a tuple (R^*, m^*, σ^*). $\mathcal{R}$ parses $\sigma^* = (\theta^*, \pi^*, z^*)$ and $\theta^* = (s^*, y^*, c^*)$ and runs $(x^*, w^*, \pi^*) \leftarrow \mathcal{E}_\mathcal{A}$ on the same inputs and random tape of $\mathcal{A}$. $\mathcal{R}$ parses $w^* = (\rho^*, \delta^*, i^*)$ and aborts if $i^* \neq i$. Otherwise $\mathcal{R}$ returns the tuple $(m^*\|R^*, (s^*, y^*, c^*, \delta^*), \rho^*)$ and interrupts the simulation.

Since it runs only PPT machines, it follows that $\mathcal{R}$ terminates in polynomial time. For the case $i = i^*$ it is easy to see that the queries are answered correctly: π is clearly a correct argument of knowledge and for a tuple $((s, y, c), \pi, \mathsf{vk} \cdot g_2^\rho)$ it holds that $e(y, \mathsf{vk} \cdot g_2^\rho \cdot g_2^s) = e(c, g_2)$, since $y = c^{\frac{1}{\mathsf{sk}+\rho+s}}$, as expected. We now have to argue that a successful forgery by $\mathcal{A}$ implies a successful forgery of $\mathcal{R}$. First we note that, in order to win the experiment, $\mathcal{A}$ must not have queried the signing oracle on input $(\cdot, R^*, m^*)$, it follows that a valid signature on $m^*||R^*$ by $\mathcal{R}$ must be a forgery. Let $\sigma^* = (\theta^*, \pi^*, z^*)$ be the challenge signature of $\mathcal{A}$ and let $\theta^* = (s^*, y^*, c^*)$. Since the extractor is successful with overwhelming probability we have that $c = \mathsf{HEval}(k_i, m^*||R^*)^{\delta^*}$ and that $z^* = \mathsf{vk} \cdot g_2^{\rho^*}$, and by the winning condition of the game we have that $e\left(y^*, z^* \cdot g_2^{s^*}\right) = e\left(y^*, \mathsf{vk} \cdot g_2^{\rho^*} \cdot g_2^{s^*}\right) = e(c^*, g_2)$. It follows that $(m^*\|R^*, (s^*, y^*, c^*, \delta^*), \rho^*)$ is a valid message-signature pair for the scheme of Figure 2. Since $i^* = i$ with probability at least $\frac{1}{\ell(\lambda)}$, this represents a contradiction to the q-strong Diffie-Hellman assumption and it concludes our proof. □

CONSTANT SIZE SIGNATURES. An interesting feature of our construction is that the computation of a signature under a re-randomized key is completely independent from the size of the ring. The only element that potentially grows with the size of the ring is therefore the proof π. However, if we implement the NIZK as a SNARK (such as in [31]) then we obtain a ring signature scheme with constant size signatures without random oracles. For the particular instantiation of [31], the proof adds only three group elements to the signature, independently of the size of the ring.

5 Efficient Non-Interactive Zero-Knowledge without Random Oracles

In this section we put forward a novel NIZK system for languages of the class of the discrete logarithms in groups that admit an efficient bilinear map and a homomorphism $\phi : \mathbb{G}_1 \to \mathbb{G}_2$. Our constructions enjoy very simple algorithms and extremely short proofs (of size comparable to proofs derived from the Fiat-Shamir heuristic [24]) and do not rely on random oracles.

5.1 Complexity Assumptions

To prove the existence of an extractor for our main protocol, we need to assume that the knowledge of the exponent assumption holds in bilinear groups of prime order. On a high level, the knowledge of exponent ensures that, for a random $h \in \mathbb{G}_1$ and for all $x \in \mathbb{Z}_p$ it is hard to compute (h^x, g^x) without knowing x. This assumption was introduced by Damgård in [20] and proven to be secure in the generic group model by Abe and Fehr in [1].

Assumption 1 (Knowledge of Exponent (KEA)) *For all* PPT *adversaries $\mathcal{A}$ there exists a non-uniform* PPT *algorithm $\mathcal{E}_\mathcal{A}$ running on the same random tape of $\mathcal{A}$ and a negligible function* negl *such that*

$$\Pr\left[\begin{array}{r} (Y,Z) \leftarrow \mathcal{A}(p,e,g_1,g_2,g_2^x), \\ (y,Y,Z) \leftarrow \mathcal{E}_\mathcal{A}(p,e,g_1,g_2,g_2^x) \end{array} \middle| Z = Y^x \wedge g^y \neq Y\right] \leq \mathsf{negl}(\lambda).$$

We define a new variant of the knowledge of exponent assumption to guarantee the existence of an extractor for disjunctive statements. This modified version of the assumption allows the adversary to choose multiple bases $h_1, \dots, h_n$ as long as they satisfy the constraint $\prod_{i \in n} h_i = h$. Then $\mathcal{A}$ has to output $\{h_i^{x_i}, g^{x_i}\}_{i \in n}$ without knowing any x_i. The intuition why we believe this assumption to be as hard as the standard knowledge of exponent is that there must be at least one h_i that is not sampled independently from h, and therefore we conjecture that a tuple of the form (h^x, g^x) can be recovered from the code of the adversary. To back up this intuition, we show that such an assumption holds against generic algorithms in Section 5.3.

Assumption 2 (Linear Knowledge of Exponent (L-KEA)) *For all $n \in$* $\mathsf{poly}(\lambda)$ *and for all* PPT *adversaries $\mathcal{A}$ there exists a non-uniform* PPT *algorithm $\mathcal{E}_\mathcal{A}$ running on the same random tape of $\mathcal{A}$ and a negligible function* negl *such that*

$$\Pr\left[\begin{array}{r} (\{Y_i,V_i,Z_i\}_{i\in n}) \leftarrow \mathcal{A}(p,e,g_1,g_2,g_2^x), \\ (y,\{Y_i,V_i,Z_i\}_{i\in n}) \leftarrow \mathcal{E}_\mathcal{A}(p,e,g_1,g_2,g_2^x) \end{array} \middle| \begin{array}{l} \forall i \in \{1,\dots,n\}: \\ \quad \mathsf{Dlog}_{g_1}(Y_i) \cdot \mathsf{Dlog}_{g_1}(V_i) \\ \qquad = \mathsf{Dlog}_{g_1}(Z_i) \\ \wedge \prod_{i\in n} V_i = g_2^x \\ \wedge \forall i : g_1^y \neq Y_i \end{array}\right] \leq \mathsf{negl}(\lambda).$$

5.2 Our Construction

Here we introduce our construction for efficient NIZK (without random oracles) for proving the knowledge of discrete logarithms. Although we sample statements from $\mathbb{G}_1$ it is easy to extend our techniques to handle the same languages in $\mathbb{G}_2$.

Base Protocol. The starting point for our system is the proof suggested by Abe and Fehr [1] for the knowledge of discrete logarithms: To prove the knowledge of an x such that $g_1^x = h$ the prover computes $\pi = \phi(T^x)$, where T is a randomly sampled group element and it is part of the common reference string. Such a proof is publicly verifiable by computing $e(h, T) = e(\pi, g_2)$. Note that the proving algorithm is deterministic and the proofs are uniquely determined by the statements and the common reference string. In some application, such as anonymous credentials, we would like to be able to re-randomize the proof. To address this issue, we present our first protocol. More formally, we describe a NIZK scheme for the language $\mathcal{L}_B$ defined as follows:

$$\mathcal{L}_B = \{A \in \mathbb{G}_1 : \exists(a) : g_1^a = A\}.$$

Clearly every element of a cyclic group have a well defined discrete logarithm when the base is the generator of the group, therefore the validity of such a statement can be simply verified by checking that A is a valid encoding of an element of $\mathbb{G}_1$. For groups of prime order this is a trivial task. However, it is unclear whether the party that outputs A *knows* the value of its discrete logarithm with respect to g_1. The scheme is depicted in Figure 4.

```
$\mathcal{G}(1^\lambda)$
    $\alpha \leftarrow \mathbb{Z}_p$
    $\mathsf{crs} \leftarrow g_2^\alpha$
    return $(\mathsf{crs}, \alpha)$
```

```
$\mathcal{P}(\mathsf{crs}, x, w)$
    parse $\mathsf{crs} = T \in \mathbb{G}_2$
    $r \leftarrow \mathbb{Z}_p$
    $R \leftarrow \phi(g_2^r)$
    $P_R \leftarrow T^r$
    $P_A \leftarrow \phi(T^{r \cdot w})$
    return $(P_A, R, P_R)$
```

```
$\mathcal{V}(\mathsf{crs}, x, \pi)$
    parse $\mathsf{crs} = T \in \mathbb{G}_2$
    parse $x = A \in \mathbb{G}_1$
    parse $\pi = (P_A, R, P_R) \in \mathbb{G}_1^2 \times \mathbb{G}_2$
    return 1 iff
        $e(R, T) = e(\phi(P_R), g_2) \wedge$
        $e(A, P_R) = e(P_A, g_2)$
```

Fig. 4: The base NIZK protocol.

In the following we prove that our scheme is a secure NIZK protocol.

Theorem 6. *The protocol in Figure 4 is a non-interactive statistically sound unconditionally zero-knowledge argument of knowledge for the language $\mathcal{L}_B$ under the knowledge of exponent assumption.*

Proof. To check whether and element A belongs to an group of order p (for some prime p) one can simply test whether A is relatively prime to p. Therefore soundness is trivial to prove. For the zero-knowledge property consider the following simulator:

$\mathcal{S}(\mathsf{crs}, x, \alpha)$. The simulator parses crs as $T \in \mathbb{G}_2$ and the statement as $A \in \mathbb{G}_1$, then it samples a random $r \leftarrow \mathbb{Z}_p$ and computes $R \leftarrow \phi(g_2^r)$, $P_R \leftarrow T^r$, and $P_A \leftarrow A^{(\alpha \cdot r)}$, The output of the simulator is (P_A, R, P_R).

The simulator is clearly efficient. It is easy to show that the proof $\pi = (P_A, R, P_R)$ is a correctly distributed proof for $x = A$, therefore the protocol is perfectly zero-knowledge. Note that this holds for any choice of the common reference string crs, as long as it is an element of $\mathbb{G}_2$ (which can be efficiently checked). It follows that the proof is unconditionally zero-knowledge. Argument of knowledge is proven by constructing an extractor for any valid proof output by any (possibly malicious) algorithm $\mathcal{A}$. Consider the following algorithm $\mathcal{B}$:

$\mathcal{B}(\mathsf{crs})$. The algorithm runs the adversary $\mathcal{A}$ on crs, which returns a tuple of the form $(x = A, \pi = (P_A, R, P_R))$ and it runs the corresponding extractor the retrieve $(x, \pi, r) \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{crs})$. $\mathcal{B}$ returns the tuple $\left(A, \pi = (P_A)^{r^{-1}}\right)$.

Let $\mathsf{extract}$ be the event such that $R = g_2^r$. Since we consider only valid arguments, it must be the case that $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$, and therefore we have that $P_R = R^\alpha$. Thus, by the knowledge of exponent assumption, we have that

$$\mathsf{Pr}[\mathsf{extract}] \geq (1 - \mathsf{negl}(\lambda)).$$

Let us now consider the following algorithm extractor:

$\mathcal{E}(\mathsf{crs})$. The extractor exexutes $\mathcal{B}$ (defined as describe above) on input crs to receive the tuple $(x = A, \pi = P_A)$. Concurrently it runs the corresponding extractor on the same input $(x, \pi, w) \leftarrow \mathcal{E}_{\mathcal{B}}(\mathsf{crs})$ and it returns w.

The algorithm $\mathcal{E}$ is obviously efficient. We now have to show that the extraction is successful with overwhelming probability, i.e., $g_1^w = A$. We split the probability as follows:

$$\mathsf{Pr}[g_1^w = A] = \mathsf{Pr}[g_1^w = A|\mathsf{extract}]\,\mathsf{Pr}[\mathsf{extract}] + \mathsf{Pr}[g_1^w = A|\overline{\mathsf{extract}}]\,\mathsf{Pr}[\overline{\mathsf{extract}}]\,.$$

By the argument above we have that

$$\mathsf{Pr}[g_1^w = A] \geq \mathsf{Pr}[g_1^w = A|\mathsf{extract}]\,(1 - \mathsf{negl}(\lambda)) + \mathsf{Pr}[g_1^w = A|\overline{\mathsf{extract}}]\,\mathsf{negl}(\lambda).$$

It follows that

$$\mathsf{Pr}[g_1^w = A] \gtrsim \mathsf{Pr}[g_1^w = A|\mathsf{extract}]\,.$$

Note that when $\mathsf{extract}$ happens it holds that $P_A = A^\alpha$. Therefore, by the knowledge of exponent assumption, the extraction is successful with probability negligibly close to 1. This concludes our proof. $\square$

Logical Conjunctions. We now describe a protocol to prove conjunction of discrete-logarithm proofs. Specifically we define the following language:

$$\mathcal{L}_C = \left\{ \begin{array}{c} (A_0, \ldots, A_n, h_1, \ldots, h_n) \in \mathbb{G}_1^{n+1} \times \mathbb{G}_2^n : \\ \exists (a) : g_1^a = A_0 \wedge \forall i \in \{1, \ldots, n\} : h_i^a = A_i \end{array} \right\}.$$

We show how to modify the previous scheme to handle statements in this language in Figure 5. We omit the description of the setup algorithm, since it is unchanged.

```
$\mathcal{P}(\mathsf{crs}, x, w)$
parse $\mathsf{crs} = T \in \mathbb{G}_2$
$r \leftarrow \mathbb{Z}_p$
$R \leftarrow \phi(g_2^r)$
$P_R \leftarrow T^r$
$P_A \leftarrow \phi(T^{r \cdot w})$
return $(P_A, R, P_R)$
```

```
$\mathcal{V}(\mathsf{crs}, x, \pi)$
parse $\mathsf{crs} = T \in \mathbb{G}_2$
parse $x = (A_0, \ldots, A_n) \in \mathbb{G}_1^{n+1}$,
          $(h_1, \ldots, h_n) \in \mathbb{G}_2^n$
parse $\pi = (P_A, R, P_R) \in \mathbb{G}_1^2 \times \mathbb{G}_2$
return 1 iff
  $e(R, T) = e(\phi(P_R), g_2) \wedge$
  $e(A_0, P_R) = e(P_A, g_2) \wedge$
  $\forall i \in \{1, \ldots, n\}$ :
    $e(A_i, P_R) = e(P_A, h_i)$
```

Fig. 5: NIZK for conjunctive statements.

Theorem 7. *The protocol in Figure 5 is a non-interactive statistically sound unconditionally zero-knowledge argument of knowledge for the language $\mathcal{L}_C$ under the knowledge of exponent assumption.*

Proof. The proof is the simple observation that the algorithm $\mathcal{P}$ is the same algorithm as the one for the language $\mathcal{L}_B$. The formal argument follows along the same lines of the one for Theorem 6. $\square$

Logical Disjunctions. We now show a protocol that handles disjunctive statements over the family of discrete logarithms. We formally define the corresponding language $\mathcal{L}_D$ as follows:

$$\mathcal{L}_D = \{(A_1, \ldots, A_n) \in \mathbb{G}_1^n : \exists (a, i) : g_1^a = A_i\}.$$

The protocol can be found in Figure 6. The basic idea here is to let the adversary cheat on $n-1$ statements by letting it choose the corresponding value of T_i (which can be seen as a temporary reference string). However, since we require that the relation $\prod_{i \in n} T_i = T$ is satisfied, at least one T_{i^*} must depend on the common T. By the linear knowledge of exponent assumption, computing a proof over such a T_{i^*} is as hard as computing it over T. In the following we show that our protocol is a secure NIZK.

Theorem 8. *The protocol in Figure 6 is a non-interactive statistically sound unconditionally zero-knowledge argument of knowledge for the language $\mathcal{L}_D$ under the linear knowledge of exponent assumption.*

Proof. As argued in the proof of Theorem 6, soundness is trivial to show for this class of statements. In order to prove zero-knowledge we construct the following simulator:

$\mathcal{S}(\mathsf{crs}, x, \alpha)$. The simulator parses crs as $T \in \mathbb{G}_2$ and x as $(A_1, \ldots, A_n) \in \mathbb{G}_1^n$. Then it samples some $i \in \{1, \ldots, n\}$ and for all $j \in \{1, \ldots, n\} \backslash i$ it samples a

```
$\mathcal{P}(\mathsf{crs}, x, w)$

parse $\mathsf{crs} = T \in \mathbb{G}_2$
parse $w = (a, i)$
$\forall j \in \{1, \dots, n\} \backslash i$ :
    $t_j \leftarrow \mathbb{Z}_p$
    $T_j \leftarrow g_2^{t_j}$
    $P_j \leftarrow (A_j)^{t_j}$
$T_i \leftarrow T \cdot (\prod_{j \in n \backslash i} g_2^{t_j})^{-1}$
$P_i \leftarrow \phi(T_i^a)$
return $(T_1, \dots, T_n, P_1, \dots, P_n)$
```

```
$\mathcal{V}(\mathsf{crs}, x, \pi)$

parse $\mathsf{crs} = T \in \mathbb{G}_2$
parse $x = (A_1, \dots, A_n) \in \mathbb{G}_1^n$
parse $\pi = (T_1, \dots, T_n) \in \mathbb{G}_2^n$
            $(P_1, \dots, P_n) \in \mathbb{G}_1^n$
return 1 iff
    $\prod_{i \in n} T_i = T \wedge$
    $\forall i \in \{1, \dots, n\}$ :
        $e(A_i, T_i) = e(P_i, g_2)$
```

Fig. 6: NIZK for disjunctive statements.

random $t_j \leftarrow \mathbb{Z}_p$ and sets $P_j \leftarrow (A_j)^{t_j}$ and $T_j \leftarrow (g_2)^{t_j}$. Then it computes $T_i = T \cdot (\prod_{j \in n \backslash i} g_2^{t_j})^{-1}$ and $P_i \leftarrow A_i^{\frac{\alpha}{\sum_{j \in n \backslash i} t_j}}$. The algorithm returns the tuple $(T_1, \dots, T_n, P_1, \dots, P_n)$.

The simulation is clearly efficient. To show that the proof is correctly distributed it is enough to observe that the tuple $(T_1, \dots, T_n)$ is sampled identically to $\mathcal{P}$ and that for all $i \in \{1, \dots, n\} : P_i = A_i^{\mathsf{Dlog}_{g_1}(T_i)}$. Hence the scheme is a perfect zero-knowledge proof. As before, one can easily show that the proof is correctly distributed for all crs as long as crs is an element of $\mathbb{G}_2$. Therefore the scheme achieves unconditional zero-knowledge. The formal argument to show that our protocol is an argument of knowledge consists of the following extractor for any PPT $\mathcal{A}$:

$\mathcal{E}(\mathsf{crs})$. The extractor runs $\mathcal{A}$ on the common reference string and it receives the tuple $(x = (A_1, \dots, A_n), \pi = (T_1, \dots, T_n, P_1, \dots, P_n))$. Then it executes the extractor $\mathcal{E}_{\mathcal{A}}$ on the same random tape to obtain (x, π, w). The extractor checks for all $i \in \{1, \dots, n\}$ whether $A_i = g_1^w$, if this is the case it returns (w, i).

The algorithm runs in polynomial time. To show that the extraction is successful whenever the proof correctly verifies, we observe that it must hold that $\prod_{i \in n} T_i = T = g_1^\alpha$ and that for all $i \in \{1, \dots, n\} : \mathsf{Dlog}_{g_1}(P_i) = \mathsf{Dlog}_{g_1}(T_i) \cdot \mathsf{Dlog}_{g_1}(A_i)$. Let fail be the event such that there exists no i such that $A_i = g_1^w$, where w is defined as above. Assume towards contradiction that

$$\Pr[\mathsf{fail}] \geq \epsilon(\lambda)$$

for some non-negligible function ϵ. Then it is easy to see that $\mathcal{E}$ contradicts the linear knowledge of exponent assumption with the same probability. It follows that

$$\Pr[\mathsf{fail}] \leq \mathsf{negl}(\lambda)$$

which implies that the extraction is successful with overwhelming probability and it concludes our proof. □

5.3 The Hardness of L-KEA Against Generic Attacks

In the following we show that L-KEA holds against generic algorithms. We model the notion of generic group algorithms as introduced by Shoup [40]. In this abstraction, algorithms can solve hard problems only by using operations and the structure of the group. In particular, generic algorithms cannot exploit the encoding of the elements. Although a hardness proof in the generic group model shall not be interpreted as a comprehensive proof of security, is also states that an algorithm that solves that specific problem will have to necessarily use the encoding of the group in some non-trivial way.

Our generic model for groups with a bilinear map is taken almost in verbatim from [5]: In such a model elements of $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are encoded as unique random strings. Given such an encoding one can only test whether two strings correspond to the same element. The group operations between elements are substituted by oracle queries to five oracles: Three oracles to compute the group operation in each of the three groups $(\mathbb{G}_2, \mathbb{G}_2, \mathbb{G}_T)$, one for the homomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, and one for the bilinear map $e : \mathbb{G}_1, \times\mathbb{G}_2 \to \mathbb{G}_T$. We denote such set of oracles by $\mathcal{O}$. The encoding of elements of $\mathbb{G}_1$ is defined as an injective function $\delta_1 : \mathbb{Z}_p \to S_1$, where $S_1 \subset \{0,1\}^*$. Mappings $\delta_2 : \mathbb{Z}_p \to S_2$ for $\mathbb{G}_2$ and $\delta_T : \mathbb{Z}_p \to S_T$ for $\mathbb{G}_T$ are defined analogously.

The main result of this section is the proof of the following theorem.

Theorem 9. *Let $(\delta_1, \delta_2, \delta_T)$ be random encoding functions for $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$, respectively. For all $n \in \mathsf{poly}(\lambda)$ and for all generic algorithms $\mathcal{A}$ with oracle access to $\mathcal{O}$ there exists a non-uniform* PPT *algorithm $\mathcal{E}_{\mathcal{A}}$ running on the same random tape of $\mathcal{A}$ and a negligible function* negl *such that*

$$\mathsf{Pr}\left[\begin{array}{l|c} (\{\delta_1(y_i), \delta_2(v_i), \delta_1(z_i)\}_{i\in n}) \leftarrow \mathcal{A}(p, \delta_1(1), \delta_2(1), \delta_2(x)), & \forall i \in \{1,\dots,n\} : \\ & y_i \cdot v_i = z_i \\ (y, \{\delta_1(y_i), \delta_2(v_i), \delta_1(z_i)\}_{i\in n}) \leftarrow \mathcal{E}_{\mathcal{A}}(p, \delta_1(1), \delta_2(1), \delta_2(x)) & \wedge \sum_{i\in n} v_i = x \\ & \wedge \forall i : y \neq y_i \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Proof. In the following we describe an extractor $\mathcal{E}$ that simulates the oracle set $\mathcal{O}$ to extract a well-formed y. At the beginning of the simulation, $\mathcal{E}$ initializes three empty lists $(\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_T)$, then it samples three random strings $(r_1, r_2, r_x) \in \{0,1\}^*$ and it appends $(1, r_1)$ and (x, r_x) to $\mathcal{Q}_1$ and $(1, r_2)$ to $\mathcal{Q}_2$. Note that we can express all of the entries in all of the lists as (F, r), where r is a random string and $F \in \mathbb{Z}_p[x]$ is a polynomial in the indeterminate x with coefficients in $\mathbb{Z}_p$. We assume without loss of generality that $\mathcal{A}$ makes oracle queries only on encodings previously received by $\mathcal{E}$, since we can set the range $\{0,1\}^*$ to be arbitrarily large. $\mathcal{E}$ provides $\mathcal{A}$ with the tuple (p, r_1, r_2, r_x) and simulates the queries of $\mathcal{A}$ to the different oracles as follows.

- Group Operation: On input two strings (r_i, r_j), $\mathcal{E}$ parses $\mathcal{Q}_1$ to retrieve the corresponding polynomials F_i and F_j, then it computes $F_k = F_i \pm F_j$ (depending on whether a multiplication or a division is requested). If an entry (F_k, r_k) is present in $\mathcal{Q}_1$ then $\mathcal{E}$ returns r_k, else samples a random $r'_k \in \{0,1\}^*$, adds (F_k, r'_k) to $\mathcal{Q}_1$ and returns r'_k. Group operation queries for $\mathbb{G}_2$ and $\mathbb{G}_T$ are treated analogously.
- Homomorphism: On input a string r_i, $\mathcal{E}$ fetches the corresponding F_i from $\mathcal{Q}_2$. If there exists an entry $(F_i, r_j) \in \mathcal{Q}_1$, then it returns r_j. Otherwise it samples an $r'_j \in \{0,1\}^*$, appends (F_i, r'_j) to $\mathcal{Q}_1$, and returns r'_j.
- Pairing: On input two strings (r_i, r_j), $\mathcal{E}$ retrieves the corresponding F_i and F_j from $\mathcal{Q}_1$ and $\mathcal{Q}_2$, respectively. Then it computes $F_k = F_i \cdot F_j$. If $(F_k, r_k) \in \mathcal{Q}_T$ then $\mathcal{E}$ returns r_k. Otherwise it samples a random $r'_k \in \{0,1\}^*$, adds (F_k, r'_k) to $\mathcal{Q}_T$ and returns r'_k.

At some point of the execution, $\mathcal{A}$ outputs a list of encodings of the form $((y_1, v_1, z_1), \ldots, (y_n, v_n, z_n))$. As argued above, we can assume that such a list is composed only by valid encodings, i.e., returned by $\mathcal{E}$ as a response to some oracle query. For all $i \in \{1, \ldots, n\}$ $\mathcal{E}$ parses $\mathcal{Q}_1$ to retrieve the F_{y_i} corresponding to y_i. If there exists an F_{y_i} that is a constant (a polynomial of degree 0 in x), then $\mathcal{E}$ returns such an F_{y_i}. Otherwise it aborts. The simulation is clearly efficient. Note that whenever $\mathcal{E}$ does not abort, its output is an element o such that $\delta_1(o) = y_i$, for some $i \in \{1, \ldots, n\}$. What is left to be shown is that $\mathcal{E}$ does not abort with all but negligible probability. First we introduce the following technical lemma from Schwarz [38].

Lemma 1. *Let $F(x_1, \ldots, x_m)$ be a polynomial of total degree $d \geq 1$. Then the probability that $F(x_1, \ldots, x_m) = 0$ mod n for randomly chosen values $(x_1, \ldots, x_m) \in \mathbb{Z}_n^m$ is bounded above by d/q, where q is the largest prime dividing n.*

Observe that at any point of the execution, for all elements $(F_i, r_i) \in \mathcal{Q}_1$, F_i is a polynomial of degree at most 1 and the same holds for the elements of $\mathcal{Q}_2$. Consequently, for each element $(F_j, r_j) \in \mathcal{Q}_T$, F_j is a polynomial of degree at most 2 in x. We can now show the following:

Lemma 2. *For all $i \in \{1, \ldots, n\}$:*

$$\mathsf{Pr}[\textit{degree of } F_{y_i} \textit{ in } x \textit{ is } \geq 1 \wedge \textit{degree of } F_{v_i} \textit{ in } x \textit{ is } \geq 1] \leq \mathsf{negl}(\lambda)$$

where F_{y_i} and F_{v_i} are the polynomials corresponding to y_i in $\mathcal{Q}_1$ and v_i in $\mathcal{Q}_2$, respectively.

Proof (Lemma 2). Let F_{z_i} be the polynomial corresponding to z_i in $\mathcal{Q}_1$. As we argued above F_{z_i} is a polynomial of degree at most 1 in x. Therefore if $F_{z_i} = F_{y_i} \cdot F_{v_i}$, then it is clear that either F_{y_i} or F_{v_i} must be a constant. Note that we require that $F_{z_i}(x) = F_{y_i}(x) \cdot F_{v_i}(x)$, for a randomly chosen x. By Lemma 1 we can bound the probability of the case $(F_{z_i} \neq F_{y_i} \cdot F_{v_i}) \wedge (F_{z_i}(x) = F_{y_i}(x) \cdot F_{v_i}(x))$ to happen to $1/p$, which is a negligible function. It follows that if $F_{z_i}(x) = F_{y_i}(x) \cdot F_{v_i}(x)$ then $F_{z_i} = F_{y_i} \cdot F_{v_i}$, with overwhelming probability and thus either F_{y_i} or F_{v_i} is a polynomial of degree 0 with the same probability. □

We proved that for all i at least one between F_{y_i} and F_{v_i} is a polynomial of degree x in 0. We now want to show that in at least one case F_{y_i} is a constant. More formally:

Lemma 3. *Let F_{v_i} be the polynomial corresponding to v_i in $\mathcal{Q}_2$. Then*

$$\mathsf{Pr}[\forall i : \textit{degree of } F_{v_i} \textit{ in } x \textit{ is } 0] \leq \mathsf{negl}(\lambda).$$

Proof (Lemma 3). Assume towards contradiction that for all i it holds that F_{v_i} is a polynomial of degree 0 in x with probability $\epsilon(\lambda)$, for some non-negligible function ϵ. Note that we require that $\sum_i F_{v_i}(x) = x$, or equivalently $a - x = 0$ for some constant $a = F_{v_i}(x)$. By Lemma 1 this happens only with negligible probability over the random choice of x. Therefore we have that ϵ is a negligible function, which is a contradiction. □

Combining Lemma 2 and Lemma 3 we have that with all but negligible probability there exists an i such that F_{v_i} is a polynomial of degree 1 in x and that the corresponding F_{y_i} is a constant. It follows that the extractor $\mathcal{E}$ does not abort with the same probability. This concludes our proof. □

6 A Ring Signature Scheme in the Standard Model

We now have all the tools to upgrade to the standard model our construction presented in Section 4.2. Our first observation is that our ring signature scheme uses the common reference string only for the computation of the zero-knowledge argument. In particular, the signature scheme is completely independent from the crs. Therefore we can potentially use different strings for different rings without compromising the correctness of the scheme. Consider our instantiation of a NIZK scheme as presented in Section 5: If we include a different $\mathsf{crs}_i = g_2^{\alpha_i}$ in each verification key vk_i, the linear combination of the crs_i for a given ring $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$ is still a correctly distributed common reference string $\mathsf{crs} = \prod_{i \in n} g_2^{\alpha_i} = g_2^{\sum_{i \in n} \alpha_i}$. The crux of this transformation is that the common reference string has no underlying hidden structure, other than being composed by a uniformly chosen group element, and that the corresponding group is closed under composition. We can modify our construction to obtain a scheme without setup as follows: Each verification key has extra element $C_i = g_2^{\alpha_i}$. On input a ring $R = (\mathsf{vk}_1, \ldots, \mathsf{vk}_n)$, the signing algorithm sets $\mathsf{crs} = \prod_{i \in n} C_i$. The verification algorithm is modified analogously and the rest of the scheme is unchanged. A complete description of the scheme can be found in Figure 7. Note that the unconditional zero-knowledge of the proof system is a fundamental component for the security of our scheme, as it guarantees that the construction stays anonymous even in presence of adversarially chosen keys. The formal analysis is elaborated below.

Theorem 10. *Let NIZK be an unconditional zero-knowledge argument, and let $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ be a programmable hash function, then the construction in Figure 7 is an anonymous ring signature scheme.*

```
$\mathsf{RGen}(1^\lambda)$
$(x, \alpha) \leftarrow \mathbb{Z}_p^2$
$k \leftarrow \mathsf{HGen}(1^\lambda)$
$C \leftarrow g_2^\alpha$
return $(x, (g_2^x, k, C))$
```

```
$\mathsf{RSig}(\mathsf{sk}, m, R)$
parse $R = (\mathsf{vk}_1, \dots, \mathsf{vk}_n)$
if $\nexists i : \mathsf{vk} = \mathsf{vk}_i$
    return $\perp$
parse $\mathsf{vk} = (z, k, C)$
parse $\mathsf{vk}_i = (z_i, k_i, C_i)$
$(s, \rho, \delta) \leftarrow \mathbb{Z}_p^3$
$z' \leftarrow z \cdot g_2^\rho$
$x' \leftarrow \mathsf{sk} + \rho$
$c \leftarrow \mathsf{HEval}(k, m||R)^\delta$
$x := R||z'||c||(m, R)$
$\pi \leftarrow \mathcal{P}\left(\prod_i C_i, (\rho, \delta, i), x\right)$
$y \leftarrow c^{\frac{1}{x'}}$
$\sigma = (s, y, c)$
return $(\sigma, \pi, z')$
```

```
$\mathsf{RVer}(R, \sigma, m)$
parse $R = (\mathsf{vk}_1, \dots, \mathsf{vk}_n)$
parse $\mathsf{vk}_i = (z_i, k_i, C_i)$
parse $\sigma = (\sigma', \pi, z')$
parse $\sigma' = (s, y, c)$
$x := R||z'||c||(m, R)$
$b \leftarrow \mathcal{V}\left(\prod_i C_i, x, \pi\right)$
$b' = 1$ if
    $e(y, \mathsf{vk}' \cdot g_2^s) = e(c, g_2)$
return $(b = b' = 1)$
```

Fig. 7: A ring signature in the standard model

Proof. The proof is essentially equivalent to the one for Theorem 4. The only subtlety that we need to address is that the simulator does not necessarily know the trapdoor for the common reference string corresponding to the challenge ring, since it may contain adversarially generated keys. However, note that any crs has a well defined discrete logarithm that the simulator can compute and use as a trapdoor to output the simulated proof. We stress that, since we are proving statistical anonymity, we do not require the simulator to run in polynomial time. By the unconditional zero-knowledge of the NIZK, the indistinguishability argument follows. □

Theorem 11. *Let* NIZK *be a computationally sound argument of knowledge that is extractable in presence of a signing oracle and let* $\mathsf{H} = (\mathsf{HGen}, \mathsf{HEval})$ *be a programmable hash function, then the construction in Figure 7 is an unforgeable ring signature scheme under the q-strong Diffie-Hellman assumption.*

Proof. The proof of unforgeability follows along the same lines of the one for Theorem 5: For the extraction it is enough to observe that the challenge ring R^* must be composed exclusively by honest verification keys. It follows that the corresponding crs is a random element of $\mathbb{G}_2$ of the form $g_2^{\sum_{i \in n} \alpha_i}$. We can therefore execute the extractor $\mathcal{E}$ on input $\sum_{i \in n} \alpha_i$ and learn a correct witness with overwhelming probability. □

Efficiency. For our standard model ring signature scheme as defined in Figure 7 a signing key sk is composed by a single integer in $\mathbb{Z}_p$ and a verification key is a collection of λ elements of $\mathbb{G}_1$ and two elements of $\mathbb{G}_2$. For a ring of size n, signatures are composed by $(2 \cdot n + 2)$ elements of $\mathbb{G}_1$, $(2 \cdot n + 1)$ elements of $\mathbb{G}_2$ and an integer in $\mathbb{Z}_p$. Signing requires $(4 \cdot n + 3)$ modular exponentiations and n computations of a programmable hash. The verification algorithm is roughly as efficient as $(4 \cdot n + 2)$ pairings, a modular exponentiation, and n computations of a programmable hash function.

6.1 Alternative Instantiations

We observe that our techniques are generically applicable to all NIZK systems whose common reference string has suitable homomorphic properties. Let us consider the NIZK of Groth [29], here the common reference string comes in two forms: The honestly generated string $(g, h = g^x, f = g^y, f^r, h^s, g^t)$ gives perfectly sound proofs, whereas the simulated string $(g, h = g^x, f = g^y, f^r, h^s, g^{r+s})$ gives perfectly zero-knowledge proofs and allows one to simulate proofs with the knowledge of (x, y, r, s). Assume that an independently sampled simulated string $(g, h_i, f_i, u_i, v_i, w_i)$ is included in each key vk_i and that the each argument of knowledge for a ring $(\mathsf{vk}_1, \dots, \mathsf{vk}_n)$ is proven against the reference string $\mathsf{crs} := \left(g, \prod_{i \in n} h_i, \prod_{i \in n} f_i, \prod_{i \in n} u_i, \prod_{i \in n} v_i, \prod_{i \in n} w_i\right)$, similarly as what has been done before. Our observation is that, if all of the strings are distributed according to the simulated variant, then one can simulate and extract proofs with the knowledge of $\left(\sum_{i \in n} x_i, \sum_{i \in n} y_i, \sum_{i \in n} r_i, \sum_{i \in n} s_i\right)$ and, since crs is a simulated string, the resulting proof is correctly distributed. Thus, the resulting ring signature scheme is anonymous as long as all the keys in the challenge ring are honestly generated. This weaker variant of the property is called *basic anonymity* in [4]. For unforgeability we leverage the fact that the challenge ring has to be composed exclusively of honestly generated keys. It follows that the relation $\mathsf{Dlog}_f\left(\prod_{i \in n} u_i\right) + \mathsf{Dlog}_h\left(\prod_{i \in n} v_i\right) = \mathsf{Dlog}_g\left(\prod_{i \in n} w_i\right)$ holds. Since the simulator knows the trapdoor, we can conclude that the extraction succeeds with overwhelming probability. The rest of the argument stays unchanged.

It follows that one could also implement our construction of Section 4.2 with the zero-knowledge argument in [29] to obtain a standard model instantiation provably secure against the Decisional-Linear assumption (DLIN). This yields a ring signature scheme without setup from more "classical" assumptions, although at the cost of a slower running time of the signing and verification algorithms, an increased size of the signatures, and weaker anonymity guarantees.

Acknowledgement This research is based upon work supported by the German research foundation (DFG) through the collaborative research center 1223, by the German Federal Ministry of Education and Research (BMBF) through the project PROMISE (16KIS0763), and by the state of Bavaria at the Nuremberg Campus of Technology (NCT). NCT is a research cooperation between the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Technische

Hochschule Nürnberg Georg Simon Ohm (THN). We thank Sherman S. M. Chow and the reviewers for for valuable comments that helped to improve our paper.

References

1. Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 118–136, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany.
2. Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. In Yuliang Zheng, editor, *Advances in Cryptology – ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 415–432, Queenstown, New Zealand, December 1–5, 2002. Springer, Heidelberg, Germany.
3. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, Berkeley, CA, USA, May 18–21, 2014. IEEE Computer Society Press.
4. Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 60–79, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.
5. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 15–19, 2004. Springer, Heidelberg, Germany.
6. Dan Boneh and David Mandell Freeman. Homomorphic signatures for polynomial functions. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 149–168, Tallinn, Estonia, May 15–19, 2011. Springer, Heidelberg, Germany.
7. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.
8. Priyanka Bose, Dipanjan Das, and Chandrasekaran Pandu Rangan. Constant size ring signature without random oracle. In Ernest Foo and Douglas Stebila, editors, *ACISP 15: 20th Australasian Conference on Information Security and Privacy*, volume 9144 of *Lecture Notes in Computer Science*, pages 230–247, Wollongong, NSW, Australia, June 29 – July 1, 2015. Springer, Heidelberg, Germany.
9. Xavier Boyen. Mesh signatures. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 210–227, Barcelona, Spain, May 20–24, 2007. Springer, Heidelberg, Germany.
10. Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15, Beijing, China, April 16–20, 2007. Springer, Heidelberg, Germany.

11. Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519, Buenos Aires, Argentina, March 26–28, 2014. Springer, Heidelberg, Germany.
12. Christina Brzuska, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schröder, and Florian Volk. Security of sanitizable signatures revisited. In Stanislaw Jarecki and Gene Tsudik, editors, *PKC 2009: 12th International Conference on Theory and Practice of Public Key Cryptography*, volume 5443 of *Lecture Notes in Computer Science*, pages 317–336, Irvine, CA, USA, March 18–20, 2009. Springer, Heidelberg, Germany.
13. Christina Brzuska, Marc Fischlin, Anja Lehmann, and Dominique Schröder. Unlinkability of sanitizable signatures. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 444–461, Paris, France, May 26–28, 2010. Springer, Heidelberg, Germany.
14. Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Heidelberg, Germany.
15. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218, Dallas, TX, USA, May 23–26, 1998. ACM Press.
16. Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007: 34th International Colloquium on Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434, Wroclaw, Poland, July 9–13, 2007. Springer, Heidelberg, Germany.
17. Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Heidelberg, Germany.
18. David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, Brighton, UK, April 8–11, 1991. Springer, Heidelberg, Germany.
19. Sherman S. M. Chow, Victor K. Wei, Joseph K. Liu, and Tsz Hon Yuen. Ring signatures without random oracles. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, pages 297–302. ACM, 2006.
20. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456, Santa Barbara, CA, USA, August 11–15, 1992. Springer, Heidelberg, Germany.
21. David Derler and Daniel Slamanig. Key-homomorphic signatures and applications to multiparty signatures. Cryptology ePrint Archive, Report 2016/792, 2016. `http://eprint.iacr.org/2016/792`.

22. Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.
23. Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st Annual Symposium on Foundations of Computer Science*, pages 283–293, Redondo Beach, CA, USA, November 12–14, 2000. IEEE Computer Society Press.
24. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.
25. Dario Fiore and Anca Nitulescu. On the (in)security of SNARKs in the presence of oracles. In *TCC 2016-B: 14th Theory of Cryptography Conference, Part I*, Lecture Notes in Computer Science, pages 108–138, Beijing, China, November 1–3, 2016. Springer, Heidelberg, Germany.
26. Nils Fleischhacker, Johannes Krupp, Giulio Malavolta, Jonas Schneider, Dominique Schröder, and Mark Simkin. Efficient unlinkable sanitizable signatures from signatures with re-randomizable keys. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 301–330, Taipei, Taiwan, March 6–9, 2016. Springer, Heidelberg, Germany.
27. Rosario Gennaro and Daniel Wichs. Fully homomorphic message authenticators. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 301–320, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany.
28. Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled fully homomorphic signatures from standard lattices. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 469–477, Portland, OR, USA, June 14–17, 2015. ACM Press.
29. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459, Shanghai, China, December 3–7, 2006. Springer, Heidelberg, Germany.
30. Jens Groth. Short non-interactive zero-knowledge proofs. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 341–358, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany.
31. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 305–326, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany.
32. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 253–280, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

33. Javier Herranz and Germán Sáez. Forking lemmas for ring signature schemes. In Thomas Johansson and Subhamoy Maitra, editors, *Progress in Cryptology - INDOCRYPT 2003: 4th International Conference in Cryptology in India*, volume 2904 of *Lecture Notes in Computer Science*, pages 266–279, New Delhi, India, December 8–10, 2003. Springer, Heidelberg, Germany.
34. Dennis Hofheinz and Eike Kiltz. Programmable hash functions and their applications. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 21–38, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany.
35. Russell W. F. Lai, Tao Zhang, Sherman S. M. Chow, and Dominique Schröder. Efficient sanitizable signatures without random oracles. In *ESORICS 2016: 21st European Symposium on Research in Computer Security, Part I*, Lecture Notes in Computer Science, pages 363–380. Springer, Heidelberg, Germany, September 2016.
36. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany.
37. Sven Schäge and Jörg Schwenk. A CDH-based ring signature scheme with short signatures and public keys. In Radu Sion, editor, *FC 2010: 14th International Conference on Financial Cryptography and Data Security*, volume 6052 of *Lecture Notes in Computer Science*, pages 129–142, Tenerife, Canary Islands, Spain, January 25–28, 2010. Springer, Heidelberg, Germany.
38. J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, pages 10–1145, 1980.
39. Hovav Shacham and Brent Waters. Efficient ring signatures without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180, Beijing, China, April 16–20, 2007. Springer, Heidelberg, Germany.
40. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.
41. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg, Germany.