Short Accountable Ring Signatures Based on DDH[⋆]

Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos[⋆⋆], Essam Ghadafi, Jens Groth, and Christophe Petit

{jonathan.bootle.14, andrea.cerulli.13, pyrros.chaidos.10, e.ghadafi, j.groth, christophe.petit}@ucl.ac.uk
University College London, UK

Abstract. Ring signatures and group signatures are prominent cryptographic primitives offering a combination of privacy and authentication. They enable individual users to anonymously sign messages on behalf of a group of users. In ring signatures, the group, i.e. the ring, is chosen in an ad hoc manner by the signer. In group signatures, group membership is controlled by a group manager. Group signatures additionally enforce accountability by providing the group manager with a secret tracing key that can be used to identify the otherwise anonymous signer when needed. Accountable ring signatures, introduced by Xu and Yung (CARDIS 2004), bridge the gap between the two notions. They provide maximal flexibility in choosing the ring, and at the same time maintain accountability by supporting a designated opener that can identify signers when needed.

We revisit accountable ring signatures and offer a formal security model for the primitive. Our model offers strong security definitions incorporating protection against maliciously chosen keys and at the same time flexibility both in the choice of the ring and the opener. We give a generic construction using standard tools.

We give a highly efficient instantiation of our generic construction in the random oracle model by meticulously combining Camenisch's group signature scheme (CRYPTO 1997) with a generalization of the one-out-of-many proofs of knowledge by Groth and Kohlweiss (EUROCRYPT 2015). Our instantiation yields signatures of logarithmic size (in the size of the ring) while relying solely on the well-studied decisional Diffie-Hellman assumption. In the process, we offer a number of optimizations for the recent Groth and Kohlweiss one-out-of-many proofs, which may be useful for other applications.

Accountable ring signatures imply traditional ring and group signatures. We therefore also obtain highly efficient instantiations of those primitives with signatures shorter than all existing ring signatures as well as existing group signatures relying on standard assumptions.

Keywords: Accountable ring signatures, group signatures, one-out-of-many zero-knowledge proofs.

1 Introduction

Significant effort has been devoted to the study of signature schemes with privacy properties that allow a signer to remain anonymous within a set of users. Two prominent examples of anonymous signature schemes are ring signatures [RST01] and group signatures [CvH91]. Ring signatures allow a signer to choose any *ad hoc* set of users, i.e. a ring, and sign anonymously on behalf the ring. Group signatures also allow a signer to sign anonymously on behalf of a group of users but here group membership is controlled by a designated group manager. The advantage of group signatures is accountability; in case of abuse, the group manager can revoke anonymity and identify the signer.

Accountable ring signatures [XY04] bridge the gap between ring signatures and group signatures. They offer the flexibility of freely choosing the ring of users when creating a signature and at the same time enforce accountability by including an opener who can open a signature and reveal who signed it. The combination of flexibility and accountability allows applications where ring signatures or group signatures are less suitable. Consider, for instance, an online forum that wants to offer anonymity to users but also wants to be able to trace people who violate the code of conduct. A forum can achieve this by allowing user posts with accountable ring signatures where the owner is the specified opener. This system is decentralized and flexible since different fora can have their own opener keys and users do not have to register with each individual forum they post to. Another potential application is an auction system where

[⋆] The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 307937 and EPSRC grant EP/J009520/1.

[⋆⋆] Was supported by an EPSRC scholarship (EP/G037264/1 – Security Science DTC).

bids are public but unsuccessful bidders want anonymity. Bidders sign bids with the seller as opener and at the end of the auctions the seller can disclose the winner in a verifiable way.

OUR CONTRIBUTION. We introduce a new security model for *accountable* ring signatures. The signer specifies, in addition to a set of users that could have produced the signature, the public key of an opening entity, which will be able to remove anonymity. This opening mechanism offers protection against misbehaving signers while at the same time not relying on a single, centralized group manager. Our security definitions are stringent and when possible incorporate protection against maliciously chosen keys.

We provide a generic construction of accountable ring signatures from standard cryptographic tools. We also give a concrete instantiation, combining ideas from Camenisch's group signature [Cam97] with a generalization of the one-out-of-many proof of knowledge of Groth and Kohlweiss [GK15]. The most efficient ring and group signatures [ACJT00, CL02, CKS09, BBS04, DKNS04, CG05, Ngu05, GK15] in the literature are in the random oracle model [FS87] and so is ours. However, the only other assumption we make is the hardness of the well-established decisional Diffie-Hellman problem.[1]

From a technical viewpoint, we offer two optimisations of Groth-Kohlweiss one-out-of-many proofs. One perspective on their proof system is that they form a binary tree and prove that one of the leaves is selected. We generalise their approach to n-ary trees, allowing us to fine-tune the parameters for better performance. For $N = n^m$, our optimisations reduce the number of group elements in the 1-out-of-N proof from $4m$ to $2m$ with little impact on the number of field elements or computational cost. Also, while their proofs can be used for ElGamal encryption, which is what we need for our scheme, this imposes an overhead in all parts of their protocol. We deploy more efficient Pedersen commitments in some parts of the proof, thus limiting the overhead of ElGamal.

The end result is an accountable ring signature scheme with efficient computation and very small signatures. Namely, for a ring with $N = \mathrm{poly}(\lambda)$ users, we obtain signatures of size approximately $\frac{5}{2}\lambda \log_2 N$ bits, which is smaller than all existing group and ring signatures based on standard assumptions.

RELATED WORK. Accountable ring signatures were informally defined by Xu and Yung [XY04]. Their security model mitigates the trust on the opener by using several openers and a threshold decryption mechanism, whereas we reduce the trust in the opener by allowing users to choose arbitrary openers (and leaving it to the verifier to decide whether they trust the opener). It would be easy to generalize our definitions to accommodate threshold decryption as well. Xu and Yung rely on the tamper-resistance of smart cards to ensure that the signatures contain some footprint of the signer. In our model, we require the signer to provide a proof that his signature is well-formed. Finally, Xu and Yung require the existence of trapdoor permutations whereas we rely on the hardness of the Decision Diffie-Hellman (DDH) problem.

Our security model for accountable ring signatures is also very similar to the identity escrow extension by Dodis et al. [DKNS04], except that we allow for an arbitrary choice of opener and we require openers to prove correctness of their decisions. The construction in [DKNS04] relies on the strong RSA assumption whereas we rely on, in our opinion, the more established DDH assumption.

Traceable ring signatures [FS08] and linkable ring signatures [LWW04] also offer some restricted form of accountability. In traceable ring signatures, any couple of signatures produced by the same user will reveal her identity. In linkable group signatures, it is possible to efficiently decide whether two signatures were produced by the same user but without revealing his identity. Unique ring signatures [FZ13] encompass both traceable and linkable ring signatures.

Formal security models for group signatures were introduced by Bellare et al. [BMW03] in the static case and by Kiayias and Yung [KY05] and Bellare et al. [BSZ05] in the partially dynamic case where users can join the group at any time. A formal security model for ring signatures was provided by Bender et al. [BKM09].

[1] An important advantage of working over a discrete logarithm group is that so do many standard signature schemes, e.g., DSS. We therefore already have many users with suitable public verification keys in a standard cyclic group, e.g., NIST's 256-bit elliptic curve group P-256.

The first practical and provably secure group signature was due to Ateniese et al. [ACJT00]. Their scheme was later improved by Camenisch and Lysyanskaya to allow efficient revocation of group member using dynamic accumulators [CL02]. Both schemes yield signatures of constant size and are based on the DDH and the strong RSA assumptions, in the random oracle model. Boneh et al. [BBS04] also constructed constant size group signatures under the strong Diffie-Hellman and the Decision Linear assumption in pairing groups. Other pairing-based schemes include [ACHdM05, NSN04, CG05, BW07, Gro07, CKS09, LPY12]. Recently, Langlois et al. [LLNW14] gave an efficient lattice-based group signature scheme supporting revocation, based on the hardness of approximating the shortest independent vectors problem in lattice of dimension n within a factor $\tilde{O}(n^{1.5})$. Our scheme achieves roughly the same signature sizes as theirs under an arguably more standard and better understood assumption.

Constant-size ring signatures can also be based on the strong RSA assumption [DKNS04] and on pairing assumptions [Ngu05]. Very recently, Groth and Kohlweiss provided a ring signature scheme based on the discrete logarithm assumption in the random oracle model, which is asymptotically more efficient than previous ones. Our accountable ring signature scheme extends Groth and Kohlweiss' scheme to enforce accountability and due to our optimisations, we get a performance improvement as well.

2 Defining Accountable Ring Signatures

We write $y = A(x; r)$ when the algorithm A on input x and randomness r outputs y. We write $y \leftarrow A(x)$ for the process of setting $y = A(x; r)$ where r is sampled at random. We also write $y \leftarrow S$ for sampling y uniformly at random from a set S. Given two functions $f, g : \mathbb{N} \to [0, 1]$ we write $f(\lambda) \approx g(\lambda)$ if $|f(\lambda) - g(\lambda)| = \lambda^{-\omega(1)}$. We say f is negligible if $f(\lambda) \approx 0$ and that f is overwhelming if $f(\lambda) \approx 1$. By PPT we mean running in probabilistic polynomial time in the relevant security parameter λ.

An accountable ring signature scheme over a PPT setup Setup is a tuple of polynomial-time algorithms $(\mathrm{OKGen}, \mathrm{UKGen}, \mathrm{Sign}, \mathrm{Vfy}, \mathrm{Open}, \mathrm{Judge})$.

$\mathrm{Setup}(1^\lambda)$**:** Given the security parameter, produces public parameters pp used (sometimes implicitly) by the rest of the scheme. The public parameters define key spaces $\mathrm{PK}, \mathrm{DK}, \mathrm{VK}, \mathrm{SK}$ with efficient algorithms for sampling and deciding membership.

$\mathrm{OKGen}(pp)$**:** Given the public parameters pp, produces a public key $pk \in \mathrm{PK}$ and secret key $dk \in \mathrm{DK}$ for an opener. Without loss of generality, we assume dk defines pk deterministically and write $pk = \mathrm{OKGen}(pp, dk)$ when computing pk from dk.

$\mathrm{UKGen}(pp)$**:** Given the public parameters pp, produces a verification key $vk \in \mathrm{VK}$ and a secret signing key $sk \in \mathrm{SK}$ for a user. We can assume sk deterministically determines vk and write $vk = \mathrm{UKGen}(pp, sk)$ when computing vk from sk.

$\mathrm{Sign}(pk, m, R, sk)$**:** Given an opener's public key, a message, a ring (i.e. a set of verification keys) and a secret key, produces a ring signature σ. The algorithm returns the error symbol $\perp$ if the inputs are malformed, i.e., if $pk \notin \mathrm{PK}$, $R \not\subset \mathrm{VK}$, $sk \notin \mathrm{SK}$ or $vk = \mathrm{UKGen}(pp, sk) \notin R$.

$\mathrm{Vfy}(pk, m, R, \sigma)$**:** Given an opener's public key, a message, a ring and a signature, returns 1 if accepting the signature and 0 otherwise. We assume the algorithm always returns 0 if the inputs are malformed, i.e., if $pk \notin \mathrm{PK}$ or $R \not\subset \mathrm{VK}$.

$\mathrm{Open}(m, R, \sigma, dk)$**:** Given a message, a ring, a ring signature and an opener's secret key, returns a verification key vk and a proof ψ that the owner of vk produced the signature. If any of the inputs are invalid, i.e., $dk \notin \mathrm{DK}$ or σ is not a valid signature using $pk = \mathrm{OKGen}(pp, dk)$, the algorithm returns $\perp$.

$\mathrm{Judge}(pk, m, R, \sigma, vk, \psi)$**:** Given an opener's public key, a message, a ring, a signature, a verification key and a proof, returns 1 if accepting the proof and 0 otherwise. We assume the algorithm returns 0 if σ is invalid or $vk \notin R$.

An accountable ring signature scheme should be correct, fully unforgeable, anonymous and traceable as defined below.

Definition 1 (Perfect correctness). *An accountable ring signature scheme is perfectly correct if for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda); (vk, sk) \leftarrow \mathrm{UKGen}(pp); \\ (pk, m, R) \leftarrow \mathcal{A}(pp, sk); \sigma \leftarrow \mathrm{Sign}(pk, m, R, sk)\ : \\ \text{If } pk \in \mathrm{PK}, R \subset \mathrm{VK}, vk \in R \text{ then } \mathrm{Vfy}(pk, m, R, \sigma) = 1 \end{array}\right] = 1.$$

We remark that correctness of the opening algorithm (w.r.t. an honestly generated opener key) is implied by the other requirements.

Full unforgeability ensures that an adversary, who may control the opener, can neither falsely accuse an honest user of producing a ring signature nor forge ring signatures on behalf of an honest ring. The former should hold even when all other users in the ring are corrupt. This requirement combines the non-frameability of group signatures [BSZ05] and the unforgeability of ring signatures [BKM09] requirements.

Definition 2 (Full Unforgeability). *An accountable ring signature scheme is fully unforgeable if for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda); (pk, vk, m, R, \sigma, \psi) \leftarrow \mathcal{A}^{\mathrm{UKGen},\mathrm{Sign},\mathrm{Reveal}}(pp) : \\ \Big(vk \in Q_{\mathrm{UKGen}} \setminus Q_{\mathrm{Reveal}}\ \wedge\ (pk, vk, m, R, \sigma) \notin Q_{\mathrm{Sign}} \\ \wedge\ \mathrm{Judge}(pk, m, R, \sigma, vk, \psi) = 1\Big) \\ \vee\ \Big(R \subset Q_{\mathrm{UKGen}} \setminus Q_{\mathrm{Reveal}}\ \wedge\ (pk, \cdot, m, R, \sigma) \notin Q_{\mathrm{Sign}} \\ \wedge\ \mathrm{Vfy}(pk, m, R, \sigma) = 1\Big) \end{array}\right] \approx 0.$$

- UKGen *runs* $(vk, sk) \leftarrow \mathrm{UKGen}(pp)$ *and returns* vk. Q_{UKGen} *is the set of verification keys* vk *that have been generated by this oracle.*
- Sign *is an oracle that on query* (pk, vk, m, R) *returns* $\sigma \leftarrow \mathrm{Sign}(pk, m, R, sk)$ *if* $vk \in R \cap Q_{\mathrm{UKGen}}$. Q_{Sign} *contains the queries and responses* (pk, vk, m, R, σ).
- Reveal *is an oracle that when queried on* $vk \in Q_{\mathrm{UKGen}}$ *returns the corresponding signing key* sk. Q_{Reveal} *is the list of verification keys* vk *for which the corresponding signing key has been revealed.*

Anonymity ensures that a signature does not reveal the identity of the ring member who produced it without the opener explicitly wanting to open the particular signature. We allow the adversary to choose the secret signing keys of the users which implies anonymity against full key exposure attacks [BKM09] where the users' secret signing keys have been revealed. Our definition also captures unlinkability as used in [XY04]: if an adversary can link signatures by the same signer, it can break anonymity.

Definition 3 (Anonymity). *An accountable ring signature scheme is anonymous if for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda); b \leftarrow \{0,1\}; (pk, dk) \leftarrow \mathrm{OKGen}(pp) : \\ \mathcal{A}^{\mathrm{Chal}_b,\mathrm{Open}}(pp, pk) = b \end{array}\right] \approx \frac{1}{2}.$$

- Chal_b *is an oracle that the adversary can only call once. On query* (m, R, sk_0, sk_1) *it runs* $\sigma_0 \leftarrow \mathrm{Sign}(pk, m, R, sk_0)$; $\sigma_1 \leftarrow \mathrm{Sign}(pk, m, R, sk_1)$. *If* $\sigma_0 \neq \bot$ *and* $\sigma_1 \neq \bot$ *it returns* σ_b, *otherwise it returns* $\bot$.
- Open *is an oracle that on a query* (m, R, σ) *returns* $\mathrm{Open}(m, R, \sigma, dk)$. *If* σ *was obtained by calling* Chal_b *on* (m, R), *the oracle returns* $\bot$.

Traceability ensures that the specified opener can always identify the ring member who produced a signature and that she is able to produce a valid proof for her decision.

Definition 4 (Traceability). *An accountable ring signature scheme is traceable if for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda); (dk, m, R, \sigma) \leftarrow \mathcal{A}(pp); \\ pk \leftarrow \mathrm{OKGen}(pp, dk); (vk, \psi) \leftarrow \mathrm{Open}(m, R, \sigma, dk) : \\ \mathrm{Vfy}(pk, m, R, \sigma) = 1 \ \wedge \ \mathrm{Judge}(pk, m, R, \sigma, vk, \psi) = 0 \end{array}\right] \approx 0.$$

Tracing soundness ensures that a signature cannot trace to two different users; only one person can be identified as the signer even when all users as well as the opener are fully corrupt. Similarly to the setting of group signatures [SSE$^+$12], this requirement is vital for some applications, e.g., where users might be rewarded for signatures they produced, or to avoid shifting blame when signatures are used as evidence of abuse.

Definition 5 (Tracing Soundness). *An accountable ring signature scheme satisfies tracing soundness if for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathrm{Setup}(1^\lambda); (m, \sigma, R, pk, vk_1, vk_2, \psi_1, \psi_2) \leftarrow \mathcal{A}(pp) : \\ \forall i \in \{1, 2\}, \ \mathrm{Judge}(pk, m, R, \sigma, vk_i, \psi_i) = 1 \wedge \ vk_1 \neq vk_2 \end{array}\right] \approx 0.$$

2.1 Ring and Group Signatures from Accountable Ring Signatures

We will now relate accountable ring signatures to ring signatures and group signatures by showing that the latter are implied by accountable ring signatures.

RING SIGNATURES. Traditional ring signatures [RST01] do not have an opener and their security requires anonymity of the signer and unforgeability [RST01, BKM09]. By simply regarding the opener's public key as part of the signature and ignoring the opening and judge algorithms, we obtain a traditional ring signature scheme from an accountable ring signature. Correctness and anonymity follow from those of the accountable ring signature, whereas unforgeability is implied by full unforgeability and traceability.

GROUP SIGNATURES. Bellare et al. [BMW03] defined group signatures for static groups, where the population of the group is fixed once and for all at the setup time, and where the group manager additionally acts as the designated opener. Besides, correctness, their model requires full anonymity and full traceability. The latter requires that an adversary in possession of the group master secret key who can corrupt members of the group, cannot produce a new signature that does not trace to a user in set of corrupt users. An accountable ring signature satisfying our security definitions gives rise to a group signature scheme as follows: We fix the group manager as the designated opener and set the corresponding decryption key as the group master secret key $gmsk$ used as the tracing key. In the setup, the group members generate their personal key pairs and we publish the ring containing the public keys of the members as part of the group signature public key. Group signatures are then just accountable ring signatures w.r.t. this ring. Full anonymity follows from the anonymity of the accountable ring signature scheme, whereas full traceability follows from full unforgeability and traceability.

The group public key in our scheme is quite large since it grows linearly in the number of members. However, this is a cost that can be amortized over many signatures. An advantage of the group signature scheme on the other hand is that it can easily be made dynamic. The group manager can enrol or remove users by adding or deleting their verification keys from the group public key [DKNS04]. In the dynamic group signature scheme, the group public key is changing and group signatures must be verified against the group as it was at the time of signing, but for scenarios where the group is not changing too often or where great flexibility is desired this is a price worth paying.

3 Preliminaries

We define here the tools and assumptions we use.

3.1 Cyclic Groups and Assumptions

A group generator $\mathcal{G}$ is a PPT algorithm that on input 1^λ (for a security paremeter λ) returns a description $gk = (\mathbb{G}, q, g)$ of a group $\mathbb{G}$ of prime order q and a generator g. We assume the group has associated polynomial time algorithms for computing group operations and deciding membership.

The Discrete Logarithm (DL) assumption holds relative to $\mathcal{G}$ if for all PPT adversaries $\mathcal{A}$

$$\Pr\left[gk = (\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\lambda); x \leftarrow \mathbb{Z}_q; h := g^x : \mathcal{A}(gk, h) = x\right] \approx 0.$$

The Decisional Diffie-Hellman (DDH) assumption holds relative to $\mathcal{G}$ if for all PPT adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{c} gk = (\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\lambda); x, y, z \leftarrow \mathbb{Z}_q; b \leftarrow \{0,1\}; \\ h := g^x; u := g^y; v := g^{(1-b)xy+bz} : \mathcal{A}(gk, h, u, v) = b \end{array}\right] \approx \frac{1}{2}.$$

The DDH assumption relative to $\mathcal{G}$ implies the DL assumption relative to $\mathcal{G}$. The DDH assumption is believed to hold when $\mathbb{G}$ is an appropriately chosen subgroup of elliptic curve groups or multiplicative groups of large characteristic finite fields.

3.2 One-way Function

A function $f : X \rightarrow Y$ (over setup gk, which defines the function f, the domain X and range Y) is *one-way* if f is polynomial-time computable and is hard to invert, i.e. for all PPT adversaries $\mathcal{A}$

$$\Pr\left[gk \leftarrow \mathcal{G}(1^\lambda); x \leftarrow \mathrm{X}; y := f(x) : \mathcal{A}(gk, y) = x\right] \approx 0.$$

We will instantiate f via group exponentiation, i.e. $x \mapsto g^x$ with domain $\mathbb{Z}_q$ and range $\mathbb{G}$. The one-wayness of f is then implied by the DL assumption.

3.3 Non-Interactive Zero-Knowledge (NIZK) Proofs

A NIZK proof system (over a setup gk) for an NP-relation $\mathcal{R}$ defining the language $\mathcal{L}_\mathcal{R} := \{s \mid \exists w : (s, w) \in \mathcal{R}\}$, where s is a statement and w is a witness, is a tuple of polynomial-time algorithms $(\mathrm{CRSGen}, \mathrm{Prove}, \mathrm{PVfy})$ which are defined as follows:

$\mathrm{CRSGen}(gk)$ on input the setup gk, outputs a common reference string crs.

$\mathrm{Prove}(crs, s, w)$ on input the reference string crs, a statement s and a witness w, outputs a proof π that $(s, w) \in \mathcal{R}$.

$\mathrm{PVfy}(crs, s, \pi)$ on input the reference string crs, a statement s and a proof π, verifies that π is a valid proof for $s \in \mathcal{L}_\mathcal{R}$ outputting a bit accordingly.

Perfect completeness of the proof system requires that for any crs generated by CRSGen and any pair $(s, w) \in \mathcal{R}$ we have $\Pr[\mathrm{PVfy}(crs, s, \mathrm{Prove}(crs, s, w))] = 1$. Additionally, we require *soundness* and *zero-knowledge*, which are as follows:

- Soundness: For all PPT adversaries $\mathcal{A}$, we have

$$\Pr\left[\begin{array}{c} gk \leftarrow \mathcal{G}(1^\lambda); crs \leftarrow \mathrm{CRSGen}(gk); (s, \pi) \leftarrow \mathcal{A}(gk, crs) : \\ \mathrm{PVfy}(crs, s, \pi) = 1 \ \wedge \ s \notin \mathcal{L}_\mathcal{R} \end{array}\right] \approx 0.$$

- Zero-Knowledge: There exist PPT algorithms $(\mathrm{SimCRSGen}, \mathrm{SimProve})$, where $\mathrm{SimCRSGen}(gk)$ outputs a simulated reference string crs and possibly a simulation trapdoor τ, and $\mathrm{SimProve}(crs, s, \tau)$ produces a simulated proof (without knowing a witness). We require that

$$\begin{aligned} &\Pr\left[gk \leftarrow \mathcal{G}(1^\lambda); crs \leftarrow \mathrm{CRSGen}(gk) : \mathcal{A}^{\mathrm{Prove}}(gk, crs) = 1\right] \\ \approx &\Pr\left[gk \leftarrow \mathcal{G}(1^\lambda); (crs, \tau) \leftarrow \mathrm{SimCRSGen}(gk) : \mathcal{A}^{\mathrm{Sim}}(gk, crs) = 1\right], \end{aligned}$$

where on query $(s, w) \in \mathcal{R}$, Sim returns $\pi \leftarrow \mathrm{SimProve}(crs, s, \tau)$.

Sigma-protocols. We will in our instantiation use NIZK proofs in the random oracle model obtained by applying the Fiat-Shamir transformation [FS87] to interactive Σ-protocols, which are 3-move protocols that allow a prover to convince a verifier that a certain statement is true.

A Σ-protocol for a relation $\mathcal{R}$ w.r.t. a setup gk is a tuple $(\mathcal{G}_{crs}, \mathcal{P}, \mathcal{V})$ which are defined as follows:

$\mathcal{G}_{crs}(gk)$ on input the setup gk, outputs a common reference string crs.
$\mathcal{P}(crs, s, w)$ on input the reference string crs, a statement s and a witness w, generates an initial message a.
$\mathcal{P}(x)$ on input a random challenge x, computes a response z.
$\mathcal{V}(crs, s, a, x, z)$ on input the reference string crs, a statement s, an initial message a, a random challenge x, and a response z, verifies the proof and outputs either 1 for acceptance or 0 for rejection.

Besides completeness, we require Σ-protocols to have *Special Honest Verifier Zero-Knowledge (SHVZK)* and *n-Special Soundness* [GK15]:

- SHVZK: Given any statement $s \in \mathcal{L}_{\mathcal{R}}$ and any verifier challenge x, it is possible to simulate a transcript of the protocol.
- n-Special Soundness: For any statement s, from n accepting transcripts $\{(a, x_i, z_i)\}_{i=1}^{n}$ for $s \in \mathcal{L}_{\mathcal{R}}$ where the challenges x_i are distinct, we can extract w s.t. $(s, w) \in \mathcal{R}$.

3.4 Signature of Knowledge

A *Signature of Knowledge* (SoK) for an NP-relation $\mathcal{R}$ w.r.t. a setup gk is a tuple $(\mathrm{SoKSetup}, \mathrm{SoKSign}, \mathrm{SoKVerify})$ which are defined as follows:

$\mathrm{SoKSetup}(gk)$ on input the setup gk, outputs public parameters pp.
$\mathrm{SoKSign}(\mathsf{pp}, s, w, m)$ on input the public parameters pp, a statement s, a witness w, and a message m, outputs a signature σ_{SoK} on m if $(s, w) \in \mathcal{R}$.
$\mathrm{SoKVerify}(\mathsf{pp}, s, m, \sigma_{\mathrm{SoK}})$ on input the public parameters pp, a statement s, a message m, and a signature σ_{SoK}, outputs 1 if σ_{SoK} is a valid signature on m or 0 otherwise.

The (game-based) security definition for signatures of knowledge (*SimExt*) [CL06], besides correctness, requires *Simulatability* and *Extractability*. We consider a *stronger* generalisation of the latter called f-extractability [BCKL08]:

- Simulatability: There are PPT algorithms $(\mathrm{SoKSimSetup}, \mathrm{SoKSimSign})$, where $\mathrm{SoKSimSetup}(gk)$ outputs public parameters pp and some trapdoor τ, whereas $\mathrm{SoKSimSign}(\mathsf{pp}, \tau, s, m)$ outputs a signature σ_{SoK}, such that

$$\begin{aligned} &\Pr\left[gk \leftarrow \mathcal{G}(1^\lambda); (\mathsf{pp}, \tau) \leftarrow \mathrm{SoKSimSetup}(gk) : \mathcal{A}^{\mathrm{SoKSim}}(gk, \mathsf{pp}) = 1\right] \\ \approx &\Pr\left[gk \leftarrow \mathcal{G}(1^\lambda); \mathsf{pp} \leftarrow \mathrm{SoKSetup}(gk) : \mathcal{A}^{\mathrm{SoKSign}}(gk, \mathsf{pp}) = 1\right], \end{aligned}$$

for all PPT adversaries $\mathcal{A}$, where $\mathrm{SoKSim}(s, w, m)$ returns $\mathrm{SoKSimSign}(\mathsf{pp}, \tau, s, m)$ if $(s, w) \in \mathcal{R}$ and $\perp$ otherwise.

- f-Extractability: For all PPT adversaries $\mathcal{A}$, there exists a polynomial time algorithm $\mathrm{SoKExtract}$ such that:

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathcal{G}(1^\lambda);\, (\mathsf{pp}, \tau) \leftarrow \mathrm{SoKSimSetup}(gk); \\ (s, m, \sigma_{\mathrm{SoK}}) \leftarrow \mathcal{A}^{\mathrm{SoKSim}}(gk, \mathsf{pp}); \\ y \leftarrow \mathrm{SoKExtract}(\mathsf{pp}, \tau, s, m, \sigma_{\mathrm{SoK}}) : \\ \quad (s, m, \sigma_{\mathrm{SoK}}) \in Q_{\mathrm{SoKSim}} \vee \ \mathrm{SoKVerify}(\mathsf{pp}, s, m, \sigma_{\mathrm{SoK}}) = 0 \\ \quad \vee \ \big(\exists w \text{ s.t. } (s, w) \in \mathcal{R} \wedge y = f(w)\big) \end{array}\right] \approx 1\cdot$$

In the above, Q_{SoKSim} is a list of queries to the SoKSimSign oracle. Note that our extractability definition is stronger than that of [CL06], as we allow the adversary to ask for signatures w.r.t. statements for which it does know the witness. In the definition, if f is the identity function, we get the standard notion of extractability.

Signatures of knowledge in the random oracle model can be efficiently realized by applying the Fiat-Shamir transformation to Σ-protocols. Applying the transformation to Σ-protocols having quasi-unique responses (i.e. given an accepting transcript, it is infeasible to find a different accepting response w.r.t. the same initial message and challenge) provides weak simulation-extractability [FKMV12], where the extractor needs to rewind the prover. To get straightline f-extractability, i.e. without rewinding [Fis05], we additionally encrypt a function f of the witness with a public key in the reference string and prove that the encrypted value is consistent with the witness. This way we get both full weak extractability and straightline f-extractability simultaneously.

3.5 Commitment Scheme

A non-interactive commitment scheme (over a setup gk) consists of two polynomial-time algorithms $(\mathrm{CGen}, \mathrm{Com}_{ck})$, where $\mathrm{CGen}(gk)$ outputs a commitment key ck, and Com_{ck} is a randomized algorithm that on input a message m and a randomness r outputs a commitment c. To open a commitment, one reveals m and r allowing anyone to verify that c is indeed a commitment to m. We require that the scheme is *hiding* and *binding*. Hiding requires that for all PPT stateful adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{l} gk \leftarrow \mathcal{G}(1^\lambda);\, ck \leftarrow \mathrm{CGen}(gk);\, (m_0, m_1) \leftarrow \mathcal{A}(gk, ck); \\ b \leftarrow \{0, 1\};\, c \leftarrow \mathrm{Com}_{ck}(m_b) : \mathcal{A}(c) = b \end{array}\right] \approx \frac{1}{2}.$$

Binding requires that for all polynomial-time stateful adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{c} gk \leftarrow \mathcal{G}(1^\lambda);\, ck \leftarrow \mathrm{CGen}(gk);\, (m_0, r_0, m_1, r_1) \leftarrow \mathcal{A}(gk, ck) : \\ m_0 \neq m_1 \ \wedge \ \mathrm{Com}_{ck}(m_0, r_0) = \mathrm{Com}_{ck}(m_1, r_1) \end{array}\right] \approx 0.$$

Pedersen commitments [Ped91] are of the form $c = g^r h^m$ where $r \leftarrow \mathbb{Z}_q^*$, $h \leftarrow \mathbb{G}$ and $m \in \mathbb{Z}_q$. They are perfectly hiding and computationally binding assuming the DL assumption holds.

$\mathrm{CGen}(gk)$	$\mathrm{Com}_{ck}(m_1, \ldots, m_n)$
$h_1, \ldots, h_n \leftarrow \mathbb{G}$. Return $ck := (h_1, \ldots, h_n)$.	If $(m_1, \ldots, m_n) \notin \mathbb{Z}_q^n$ return $\perp$. $r \leftarrow \mathbb{Z}_q$; Return $c := g^r \prod_{i=1}^n h_i^{m_i}$.

Fig. 1. Pedersen commitment to multiple elements.

We exploit the fact that the Pedersen commitment scheme is homomorphic, i.e., for all correctly generated gk, ck and all $m, m', r, r' \in \mathbb{Z}_q$

$$\mathrm{Com}_{ck}(m; r) \cdot \mathrm{Com}_{ck}(m'; r') = \mathrm{Com}_{ck}(m + m'; r + r').$$

We will use a variant of the Pedersen commitment scheme to commit to multiple messages at once as shown in Fig. 1.

3.6 IND-CPA Public-Key Encryption

A public-key encryption (PKE) scheme (over setup gk) consists of three algorithms $(\mathrm{PKEGen}, \mathrm{Enc}, \mathrm{Dec})$, which are defined as follows:

$\mathrm{PKEGen}(gk)$ is a probabilistic algorithm that on input the setup gk, generates a public key and decryption key pair (pk, dk). Without loss of generality, we assume pk can be efficiently computed given dk and write $pk = \mathrm{PKEGen}(gk, dk)$ for this computation which returns $\perp$ if dk is not valid.

$\mathrm{Enc}(pk, m)$ is a probabilistic algorithm that on input the public key pk and a message m, returns a ciphertext c if all its inputs are valid and $\perp$ otherwise.

$\mathrm{Dec}(dk, c)$ is a deterministic algorithm that on input the decryption key dk and a ciphertext c, returns either the message m or the failure symbol $\perp$.

We assume that gk, which is an implicit input to Enc and Dec, defines the public key, decryption key, message, randomness and ciphertext spaces PK, DK, M, Rnd and C.

Correctness of PKE requires that for all $m \in \mathrm{M}$, all $gk \leftarrow \mathcal{G}(1^\lambda)$ and all $(pk, dk) \leftarrow \mathrm{PKEGen}(gk)$, it holds that $m = \mathrm{Dec}(dk, \mathrm{Enc}(pk, m))$. We also require that the scheme is *indistinguishable under chosen plaintext attacks (IND-CPA)*, i.e., for all PPT stateful adversaries $\mathcal{A}$

$$\Pr\left[\begin{matrix} gk \leftarrow \mathcal{G}(1^\lambda); (pk, dk) \leftarrow \mathrm{PKEGen}(gk) \\ (m_0, m_1) \leftarrow \mathcal{A}(gk, pk); b \leftarrow \{0,1\}; c \leftarrow \mathrm{Enc}(pk, m_b) \end{matrix} : \mathcal{A}(c) = b\right] \approx \frac{1}{2},$$

where we require $\mathcal{A}$ outputs $m_0, m_1 \in \mathrm{M}$.

We will in our instantiation use ElGamal encryption described in Fig. 2, which is IND-CPA secure if the DDH assumption holds relative to $\mathcal{G}$ where $gk = (\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\lambda)$. We also note that ElGamal ciphertexts are homomorphic, similarly to Pedersen commitments. We have $\mathrm{PK} := \mathbb{G}^*$, $\mathrm{DK} := \mathbb{Z}_q^*$, $\mathrm{M} := \mathbb{G}$, $\mathrm{Rnd} := \mathbb{Z}_q$, and $\mathrm{C} := \mathbb{G}^2$.

$\mathrm{PKEGen}(gk)$	$\mathrm{Enc}(pk, m)$	$\mathrm{Dec}(dk, c = (u, v))$
$dk \leftarrow \mathbb{Z}_q^*; pk := g^{dk}$.	If $pk \notin \mathbb{G}^*$ or $m \notin \mathbb{G}$ return $\perp$.	If $dk \notin \mathbb{Z}_q^*$ or $c \notin \mathbb{G}^2$ return $\perp$.
Return (pk, dk).	$r \leftarrow \mathbb{Z}_q$; Return $c := (pk^r, g^r m)$.	Return $m := vu^{-\frac{1}{dk}}$.

Fig. 2. ElGamal encryption.

4 Constructing Accountable Ring Signatures

Our generic construction (shown in Fig. 3) uses a one-way function f, an IND-CPA public-key encryption scheme, a signature of knowledge, and a zero-knowledge proof of membership, all of which share the same setup gk. The setup gk defines domain SK and range VK for f, and key, message, randomness and ciphertext spaces $\mathrm{PK}, \mathrm{DK}, \mathrm{M}, \mathrm{Rnd}, \mathrm{C}$ for the encryption scheme. The range of f and the message space of the encryption scheme need to be compatible such that $\mathrm{VK} \subseteq \mathrm{M}$.

The idea is that an opener will have a key pair for the encryption scheme and the user will have a secret key sk and corresponding verification key $vk = f(sk)$. To sign a message m w.r.t. a ring R, the signer first encrypts her verification key under the opener's public key and provides a signature of knowledge on m proving the ciphertext encrypts a verification key in the ring and that she knows the secret key associated with the encrypted verification key. To open a signature, the opener decrypts the ciphertext to obtain the user's verification key and provides an NIZK proof of correct decryption.

The relations $\mathcal{R}_{\mathrm{sig}}$ and $\mathcal{R}_{\mathrm{open}}$ associated with the signature of knowledge and the NIZK system, respectively, are as follows:

```
$\mathrm{Setup}(1^\lambda)$
    $gk \leftarrow \mathcal{G}(1^\lambda); crs \leftarrow \mathrm{CRSGen}(gk)$
    $\mathsf{pp}_{\mathrm{SoK}} \leftarrow \mathrm{SoKSetup}(gk)$
    Return $pp := (gk, \mathsf{pp}_{\mathrm{SoK}}, crs)$

$\mathrm{OKGen}(pp)$
    $(pk, dk) \leftarrow \mathrm{PKEGen}(gk)$
    Return $(pk, dk)$

$\mathrm{Open}(m, R, \sigma, dk)$
    $pk \leftarrow \mathrm{OKGen}(pp; dk)$
    If $\mathrm{Vfy}(pk, m, R, \sigma) = 0$ return $\bot$
    Parse $\sigma$ as $(c, \sigma_{\mathrm{SoK}})$
    $vk := \mathrm{Dec}(dk, c)$
    $\psi \leftarrow \mathrm{Prove}(crs, (pk, c, vk), dk)$
    Return $(vk, \psi)$

$\mathrm{UKGen}(pp)$
    $sk \leftarrow \mathrm{SK};\ vk := f(sk);$ Return $(vk, sk)$

$\mathrm{Sign}(pk, m, R, sk)$
    $vk \leftarrow f(sk); r \leftarrow \mathrm{Rnd}; c \leftarrow \mathrm{Enc}(pk, vk; r)$
    $\sigma_{\mathrm{SoK}} \leftarrow \mathrm{SoKSign}(\mathsf{pp}_{\mathrm{SoK}}, (pk, R, c), (sk, r), m)$
    Return $\sigma := (c, \sigma_{\mathrm{SoK}})$

$\mathrm{Vfy}(pk, m, R, \sigma)$
    Parse $\sigma$ as $(c, \sigma_{\mathrm{SoK}})$
    Return $\mathrm{SoKVerify}(\mathsf{pp}_{\mathrm{SoK}}, (pk, R, c), m, \sigma_{\mathrm{SoK}})$

$\mathrm{Judge}(pk, m, R, \sigma, vk, \psi)$
    If $\mathrm{Vfy}(pk, m, R, \sigma) = 0$ return 0
    Parse $\sigma$ as $(c, \sigma_{\mathrm{SoK}})$
    Return $\mathrm{PVfy}(crs, (pk, c, vk), \psi)$
```

Fig. 3. A generic construction for Accountable Ring Signatures.

$$\mathcal{R}_{\mathrm{sig}} := \left\{ \begin{array}{c} \big((pk, R, c), (sk, r)\big) : \\ R \subset \mathrm{VK} \ \wedge \ vk := f(sk) \in R \ \wedge \ c = \mathrm{Enc}(pk, vk; r) \end{array} \right\}.$$

$$\mathcal{R}_{\mathrm{open}} := \left\{ \begin{array}{c} \big((pk, c, vk), dk\big) : \\ pk = \mathrm{PKEGen}(gk; dk) \in \mathrm{PK} \ \wedge \ vk = \mathrm{Dec}(dk, c) \ \wedge \ vk \in \mathrm{VK} \end{array} \right\}.$$

We prove the following theorem in Appendix A.

Theorem 1. *The accountable ring signature construction in Fig. 3 is perfectly correct, anonymous, fully unforgeable, traceable, and satisfies tracing soundness if the building blocks satisfy the security definitions in Sec. 3.*

Since all the building blocks can be constructed from (doubly enhanced) trapdoor permutations, we get as a corollary that trapdoor permutations imply the existence of accountable ring signatures.

5 Efficient Instantiation

We give here an efficient instantiation of the generic construction from Fig. 3. The instantiation is secure in the random oracle model under the well-established DDH assumption. As specified in Sec. 3, we instantiate f with group exponentiation and the IND-CPA encryption scheme with ElGamal. We will get the Signature of Knowledge and NIZK proof for the relations $\mathcal{R}_{\mathrm{sig}}$ and $\mathcal{R}_{\mathrm{open}}$ by applying the Fiat-Shamir transform to suitable Σ-protocols for these relations. Thanks to the straightline f-Extractability of our instantiation of the signature of knowledge, we can answer the adversary's Open queries in the anonymity game by extracting $vk = f(sk)$ from σ_{SoK} without rewinding.

5.1 Details of the Σ-protocols

We give here the details of the different Σ-protocols used in the instantiation.

For all Σ-protocols, the setup includes the group description gk and the common reference string $crs := (ck, ek)$, where $ck \leftarrow \mathrm{CGen}(gk)$, $(ek, \tau) \leftarrow \mathrm{PKEGen}(gk)$ and $ek = g^\tau$ for $\tau \leftarrow \mathbb{Z}_q^*$, for the

Pedersen commitment scheme and the ElGamal encryption scheme, respectively. The proofs of the lemmata can be found in Appendix B.

Committed bits. We first give a Σ-protocol for a commitment B having an opening consisting of sequences of bits, where in each sequence there is exactly one 1. More precisely, we give in Fig. 4 a Σ-protocol $(\mathcal{G}_{crs}, \mathcal{P}_1, \mathcal{V}_1)$ for the relation

$$\mathcal{R}_1 = \left\{ \begin{array}{c} (B, (b_{0,0}, \ldots, b_{m-1,n-1}, r)) : \\ (\forall i, j : b_{j,i} \in \{0,1\}) \wedge (\forall j : \sum_{i=0}^{n-1} b_{j,i} = 1) \wedge B = \mathrm{Com}_{ck}(b_{0,0}, \ldots, b_{m-1,n-1}; r) \end{array} \right\}$$

The main idea is to prove that $b_{j,i}(1 - b_{j,i}) = 0$ for all i, j, and also that $\sum_{i=1}^{n} b_{j,i} = 1$.

$$\begin{array}{lcl}
\underline{\mathcal{P}_1(gk, crs, B, (b_{0,0}, \ldots, b_{m-1,n-1}, r))} & & \underline{\mathcal{V}_1(gk, crs, B)} \\
r_A, r_C, r_D, a_{j,1}, \ldots, a_{j,n-1} \leftarrow \mathbb{Z}_q & & \\
\forall j : a_{j,0} := -\sum_{i=1}^{n-1} a_{j,i} & A, C, D & \\
A := \mathrm{Com}_{ck}(a_{0,0}, \ldots, a_{m-1,n-1}; r_A) & \xrightarrow{\hspace{8em}} & \\
C := \mathrm{Com}_{ck}(\{a_{j,i}(1 - 2b_{j,i})\}_{j,i=0}^{m-1,n-1}; r_C) & & \text{Accept if and only if} \\
D := \mathrm{Com}_{ck}(-a_{0,0}^2, \ldots, -a_{m-1,n-1}^2; r_D) & x \leftarrow \{0,1\}^\lambda & \quad A, B, C, D \in \mathbb{G} \\
 & \xleftarrow{\hspace{8em}} & \quad f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C \in \mathbb{Z}_q \\
\forall j, i : f_{j,i} := b_{j,i}x + a_{j,i} & & \quad \forall j : f_{j,0} := x - \sum_{i=1}^{n-1} f_{j,i} \\
z_A := rx + r_A & f_{0,1}, f_{1,1}, \ldots, f_{m-1,n-1}, z_A, z_C & \quad B^x A = \mathrm{Com}_{ck}(f_{0,0}, \ldots, f_{m-1,n-1}; z_A) \\
z_C := r_C x + r_D & \xrightarrow{\hspace{8em}} & \quad C^x D = \mathrm{Com}_{ck}(\{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{m-1,n-1}; z_C)
\end{array}$$

Fig. 4. Σ-protocol for relation $\mathcal{R}_1$.

Lemma 1. *The* Σ-protocol *in Fig. 4 is perfectly complete, perfect SHVZK, computational 3-special sound and has quasi-unique responses.*

List Containing Encryption of 1. We now describe a Σ-protocol that a list of N ElGamal ciphertexts $(c_0, \ldots, c_{N-1})$ includes an encryption of 1. More precisely, we give a Σ-protocol $(\mathcal{G}_{crs}, \mathcal{P}_2, \mathcal{V}_2)$ (see Fig. 5) for the relation:

$$\mathcal{R}_2 = \left\{ ((\{c_i\}_{i=0}^{N-1}), (\ell, r)) : (\forall i, c_i \in \mathbb{G}^2) \wedge \ell \in \{0, \ldots, N-1\} \wedge c_\ell = \mathrm{Enc}_{ek}(1; r) \right\}$$

This generalizes easily to other homomorphic encryption and commitment schemes.

Since we can pad the list with copies of the last ciphertext (at little extra cost in the protocol), we may assume $N = n^m$. We will later discuss the efficiency implications of different choices of n. The idea behind our Σ-protocol is to prove knowledge of an index ℓ for which the product $\prod_{i=0}^{N-1} c_i^{\delta_{\ell,i}}$ is an encryption of 1, where as usual $\delta_{\ell,i} = 1$ when $i = \ell$ and $\delta_{\ell,i} = 0$ otherwise. We have $\delta_{\ell,i} = \prod_{j=0}^{m-1} \delta_{\ell_j, i_j}$, where $\ell = \sum_{j=0}^{m-1} \ell_j n^j$ and $i = \sum_{j=0}^{m-1} i_j n^j$ are the n-ary representations of ℓ and i respectively.

The prover first commits to m sequences of n bits $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,n-1})$. It runs the Σ-protocol in Fig. 4 to prove that the commitment is well-formed. On receiving a challenge x, the prover discloses elements $f_{j,i} = \delta_{\ell_j,i}x + a_{j,i}$ as in Fig. 4. Observe that for every $i \in \{0, \ldots, N-1\}$, the product $\prod_{j=0}^{m-1} f_{j,i_j}$ is the evaluation at x of the polynomial $p_i(x) = \prod_{j=0}^{m-1}(\delta_{\ell_j,i}x + a_{j,i})$. For $0 \leq i \leq N-1$, we have:

$$p_i(x) = \prod_{j=0}^{m-1} \delta_{\ell_j,i_j} x + \sum_{k=0}^{m-1} p_{i,k} x^k = \delta_{\ell,i} x^m + \sum_{k=0}^{m-1} p_{i,k} x^k, \tag{1}$$

for some coefficients $p_{i,k}$ depending on ℓ and $a_{j,i}$. Note that $p_{i,k}$ can be computed by the prover independently of x, and that $p_\ell(x)$ is the only degree m polynomial amongst $p_0(x), \ldots, p_{N-1}(x)$.

From these coefficients and some random noise values ρ_k, the prover computes ciphertexts $G_k := \prod_{i=0}^{N-1} c_i^{p_{i,k}} \cdot \mathrm{Enc}_{ek}(1; \rho_k)$ and includes them in the initial message. These ciphertexts are then used to cancel out the low degree terms in (1). Namely, if c_ℓ is an encryption of 1, the following product is an encryption of 1 for any x

$$\prod_{i=0}^{N-1} c_i^{\prod_{j=0}^{m-1} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} G_k^{-x^k} = \left(\prod_{i=0}^{N-1} c_i^{\delta_{\ell,i}} \right)^{x^m} .$$

$\mathcal{P}_2(gk, crs, (c_0, \dots, c_{N-1}), (\ell, r))$		$\mathcal{V}_2(gk, crs, (c_0, \dots, c_{N-1}))$
$r_B, \rho_k \leftarrow \mathbb{Z}_q$		
$B := \mathrm{Com}_{ck}(\delta_{\ell_{0,0}}, \dots, \delta_{\ell_{m-1,n-1}}; r_B)$		
$(A, C, D) \leftarrow \mathcal{P}_1(gk, crs, B, (\{\delta_{\ell_j,i}\}_{j,i=0}^{m-1,n-1}, r_B))$		
For $k = 0, \dots, m-1$	$\xrightarrow{A, B, C, D, \{G_k\}_{k=0}^{m-1}}$	Accept if and only if
$G_k = \prod_{i=0}^{N-1} c_i^{p_{i,k}} \cdot \mathrm{Enc}_{ek}(1; \rho_k)$		$A, B, C, D, G_0, \dots, G_{m-1} \in \mathbb{G}$
using $p_{i,k}$ from (1)	$\xleftarrow{x \leftarrow \{0,1\}^\lambda}$	$f_{0,1}, \dots, f_{m-1,n-1}, z_A, z_C, z \in \mathbb{Z}_q$
		$\mathcal{V}_1(gk, crs, B, x, A, B, C, \{f_{j,i}\}_{j=0,i=1}^{m-1,n-1}, z_A, z_C) = 1$
$(f_{0,1}, \dots, f_{m-1,n-1}, z_A, z_C) \leftarrow \mathcal{P}_1(x)$	$f_{0,1}, \dots, f_{m-1,n-1}, z_A, z_C, z$	$\forall j : f_{j,0} := x - \sum_{i=1}^{n-1} f_{j,i}$
$z := rx^m - \sum_{k=0}^{m-1} \rho_k x^k$	$\longrightarrow$	$\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{m} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} G_k^{-x^k} = \mathrm{Enc}_{ek}(1; z)$

Fig. 5. Σ-protocol for a list $c_0, \dots, c_{N-1}$ containing an encryption of 1

Lemma 2. *Let $m \geq 2$. The* Σ-protocol *in Fig. 5 is perfectly complete, SHVZK, $(m+1)$-special sound and has quasi-unique responses.*

Correct Signature. We give in Fig. 6 a Σ-protocol for the relation
$\mathcal{R}_{\text{sig}} = \left\{ ((pk, m, R, c), (sk, r)) : sk \in \mathbb{Z}_q \wedge vk = g^{sk} \in R \subset \mathbb{G}^* \wedge c = \mathrm{Enc}_{pk}(vk; r) \right\}$

$\mathcal{P}_{\text{sig}}(gk, crs, (pk, m, R, c), (r, sk))$		$\mathcal{V}_{\text{sig}}(gk, crs, (pk, m, R, c))$
$s, t, r_a, r_b \leftarrow \mathbb{Z}_q$; $d \leftarrow \mathrm{Enc}_{ek}(g^{sk}; t)$		
$A \leftarrow \mathrm{Enc}_{pk}(g^s; r_a)$; $B \leftarrow \mathrm{Enc}_{ek}(g^s; r_b)$	d, A, B, a_2	Accept iff
$c_0 := d \cdot \mathrm{Enc}_{ek}(vk_0^{-1}; 0), \dots, c_{N-1} := d \cdot \mathrm{Enc}_{ek}(vk_{N-1}^{-1}; 0)$	$\longrightarrow$	$R \subset \mathbb{G}$; $pk \in \mathbb{G}^*$; $d, A, B \in \mathbb{G}^2$
$a_2 \leftarrow \mathcal{P}_2(gk, crs, (c_0, \dots, c_{N-1}), (\ell, t))$	$x \leftarrow \{0,1\}^\lambda$	$z_s, z_a, z_b \in \mathbb{Z}_q$
	$\longleftarrow$	$c^x A = \mathrm{Enc}_{pk}(g^{z_s}; z_a)$
$z_s := sk \cdot x + s$; $z_a := rx + r_a$; $z_b := tx + r_b$	z_a, z_b, z_s, z_2	$d^x B = \mathrm{Enc}_{ek}(g^{z_s}; z_b)$
$z_2 \leftarrow \mathcal{P}_2(x)$	$\longrightarrow$	$\mathcal{V}_2(gk, crs, (c_0, \dots, c_{N-1}), a_2, x, z_2) = 1$

Fig. 6. Σ-protocol for $\mathcal{R}_{\text{sig}}$.

Lemma 3. *The* Σ-protocol *in Fig. 6 is perfectly complete, SHVZK, $m+1$-special sound and has quasi-unique responses.*

Lemma 4. *Applying the Fiat-Shamir transformation to the protocol in Fig. 6 with* SoKSetup *as in Sect. 5 produces a signature of knowledge in the random oracle model, that is extractable and straightline f-extractable.*

Proof. For simulatability, SoKSimSetup is identical to SoKSetup and SoKSimSign programs the random oracle to simulate proofs. Simulatability then follows from SHVZK.

For extractability we rely on rewinding, $m+1$ special soundness and quasi-unique responses, using [FKMV12]. For straightline f-extractability, we use the trapdoor τ to decrypt d in the proof transcript and obtain $vk = f(sk)$. □

Correct Opening. Writing out the details of ElGamal encryption we get

$$\mathcal{R}_{\text{open}} = \left\{ \begin{array}{c} ((pk, c, vk), dk) : \\ dk \in \mathbb{Z}_q \wedge pk = g^{dk} \neq 1 \wedge c = (u, v) \in \mathbb{G}^2 \wedge vk \in \mathbb{G} \wedge (v/vk)^{dk} = u \end{array} \right\}$$

$\mathcal{P}_{\text{open}}(gk, crs, (pk, u, v, vk), dk)$		$\mathcal{V}_{\text{open}}(gk, crs, (pk, u, v, vk))$
$a \leftarrow \mathbb{Z}_q$	A, B	Accept if and only if
$A := g^a$	$\longrightarrow$	$pk, u, v, vk \in \mathbb{G}$
$B := (v/vk)^a$	$x \leftarrow \mathbb{Z}_q$	$pk \neq 1$
	$\longleftarrow$	$pk^x A = g^z$
$z = dk \cdot x + a$	z	$u^x B = (v/vk)^z$
	$\longrightarrow$	

Fig. 7. Σ-protocol for correct decryption.

Lemma 5. *The Σ-protocol in Fig. 7 is perfectly complete, perfect SHVZK, perfect 2-special sound and has unique responses. Also, applying the Fiat-Shamir transformation to it produces a NIZK proof.*

5.2 Efficiency of our schemes

The efficiency of our schemes is determined by the signature of knowledge of Fig. 6. For a ring of $N = n^m$ users, this requires the prover to send $m+4$ ElGamal ciphertexts, 4 Pedersen commitments and $m(n-1)+6$ elements of $\mathbb{Z}_q$. A full accountable ring signature includes an additional ElGamal encryption, i.e. $2m+12$ group elements and $m(n-1)+6$ field elements in total.

A signature can be computed using $mN + 3mn + 2m + 12$ group exponentiations as follows. Computing A, C and D in the bit proof requires $2mn+3$ exponentiations since exponentiation by $(1-2b_{i,j})$ amounts to a multiplication. By construction of c_i in Fig. 6, the first components of all c_i are identical in Fig. 5, so computing the first components of all G_k costs $2m$ exponentiations. The second components of all G_k requires $mN+m$ exponentiations. We also need 9 exponentiations to compute B in Fig. 5, d, A and B in Fig. 6, and the ElGamal encryption of the public key.

Signatures can be verified using $N + 2mn + 2m + 15$ group exponentiations as follows: $N + 2m + 3$ exponentiations for the last verification equation in Fig. 5, $2mn+4$ for the equations in Fig. 4 and 8 for the first two verification equations in Fig. 6.

Our schemes can be instantiated over any group $\mathbb{G}$ where the DDH problem is computationally hard. Let us say the security parameter λ determines the bit size of the field elements as $|q| \approx \lambda$ bits and let $N = \text{poly}(\lambda)$. When group elements are much larger than field elements, say more than a factor λ, it is convenient to choose a large n. For instance, setting $n = \lambda + 1$ (in which case $m = O(1)$) the communication complexity amounts to a constant number of group elements and $m\lambda + 6$ field elements. When group and field elements are of roughly the same size, as can be the case for elliptic curve groups, our signatures have total size $m(n+1)+18$ elements. Setting $n=4$ gives communication of roughly $5\log_4 N + 18 = \frac{5}{2}\log_2 N + 18$ elements.

In table 8, we compare our instantiation with prior work. Since our signatures require a logarithmic number of group elements, they enjoy shorter sizes than all previous signatures based on RSA and/or DDH assumptions, for sufficiently large security parameters. Indeed, a constant number of RSA ring elements typically requires $O(\lambda^3)$ bits whereas the elliptic curve instantiation of our protocol achieves $O(\lambda \log N)$ bit size. As long as λ is large enough and $N \leq 2^{\lambda^2}$, our signatures will be shorter. Our signatures are also a factor 2.8 shorter than Groth and Kohlweiss ring signatures.

Scheme	Signature Size	Assumptions	Type
[ACJT00]	$3\mathbb{Z}_n^* + 4\mathbb{Z}$	Strong RSA	Group
[CL02]	$6\mathbb{Z}_n^* + 8\mathbb{Z}$	Strong RSA	Group
[DKNS04]	$12\mathbb{Z}_n^* + 12\mathbb{Z}$	Strong RSA	Ring/Group
[CG05]	$4\mathbb{Z}_n^* + 4\mathbb{Z}$	Strong RSA + DDH	Group
[GK15]	$(4\log_2 N)\mathbb{G} + (3\log_2 N + 1)\mathbb{Z}_q^*$	DDH	Ring
Ours	$(\log_2 N + 12)\mathbb{G} + \frac{1}{2}(3\log_2 N + 12)\mathbb{Z}_q^*$	DDH	Ring/Group

Fig. 8. Efficiency comparison between our instantiation and most efficient group and ring signatures based on RSA and/or DDH assumptions. $\mathbb{Z}_n^*, \mathbb{Z}, \mathbb{G}, \mathbb{Z}_q^*$ represent the size of RSA ring elements, integers, group elements and field elements, respectively.

References

ACHdM05. G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros. *Practical group signatures without random oracles.* Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org/`.

ACJT00. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. *A practical and provably secure coalition-resistant group signature scheme.* In *EUROCRYPT*, vol. 1880 of *LNCS*. 2000.

BBS04. D. Boneh, X. Boyen, and H. Shacham. *Short group signatures.* In *CRYPTO*, vol. 3152 of *LNCS*. Springer, 2004.

BCKL08. M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. *P-signatures and noninteractive anonymous credentials.* In *TCC*, vol. 4948 of *LNCS*. Springer, 2008.

BKM09. A. Bender, J. Katz, and R. Morselli. *Ring signatures: Stronger definitions, and constructions without random oracles.* *Journal of Cryptology*, 22(1):114, 2009.

BMW03. M. Bellare, D. Micciancio, and B. Warinschi. *Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions.* In *EUROCRYPT*, vol. 2656 of *LNCS*. Springer, 2003.

BSZ05. M. Bellare, H. Shi, and C. Zhang. *Foundations of group signatures: The case of dynamic groups.* In *CT-RSA 2005*, vol. 3376 of *LNCS*. Springer, 2005.

BW07. X. Boyen and B. Waters. *Full-domain subgroup hiding and constant-size group signatures.* In *PKC*, vol. 4450. Springer, 2007.

Cam97. J. Camenisch. *Efficient and generalized group signatures.* In *EUROCRYPT*, vol. 1233 of *LNCS*. Springer, 1997.

CG05. J. Camenisch and J. Groth. *Group signatures: Better efficiency and new theoretical aspects.* In *SCN*, vol. 3352 of *LNCS*. Springer, 2005.

CKS09. J. Camenisch, M. Kohlweiss, and C. Soriente. *An accumulator based on bilinear maps and efficient revocation for anonymous credentials.* In *PKC*, vol. 5443 of *LNCS*. Springer, 2009.

CL02. J. Camenisch and A. Lysyanskaya. *Dynamic accumulators and application to efficient revocation of anonymous credentials.* In *CRYPTO*. 2002.

CL06. M. Chase and A. Lysyanskaya. *On signatures of knowledge.* In *CRYPTO 2006*, vol. 4117 of *LNCS*. Springer, 2006.

CvH91. D. Chaum and E. van Heyst. *Group signatures.* In *EUROCRYPT*, vol. 547 of *LNCS*. Springer, 1991.

DKNS04. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. *Anonymous identification in ad hoc groups.* In *EUROCRYPT*, vol. 3027 of *LNCS*. Springer, 2004.

Fis05. M. Fischlin. *Communication-efficient non-interactive proofs of knowledge with online extractors.* In *CRYPTO*, vol. 3621 of *LNCS*. Springer, 2005.

FKMV12. S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. *On the non-malleability of the Fiat-Shamir transform.* In *INDOCRYPT*, vol. 7668 of *LNCS*. Springer, 2012.

FS87. A. Fiat and A. Shamir. *How to prove yourself: Practical solutions to identification and signature problems.* In *CRYPTO*, vol. 263 of *LNCS*. Springer, 1987.

FS08. E. Fujisaki and K. Suzuki. *Traceable ring signature.* *IEICE Transactions*, 91-A(1):83, 2008.

FZ13. M. K. Franklin and H. Zhang. *Unique ring signatures: A practical construction.* In *FC*, vol. 7859 of *LNCS*. Springer, 2013.

GK15. J. Groth and M. Kohlweiss. *One-out-of-many proofs: Or how to leak a secret and spend a coin.* In *EUROCRYPT*, vol. 9057 of *LNCS*. Springer, 2015.

Gro07. J. Groth. *Fully anonymous group signatures without random oracles.* In *ASIACRYPT*, vol. 4833 of *LNCS*. Springer, 2007.

KY05. A. Kiayias and M. Yung. *Group signatures with efficient concurrent join.* In *EUROCRYPT*, vol. 3494 of *LNCS*. Springer, 2005.

LLNW14. A. Langlois, S. Ling, K. Nguyen, and H. Wang. *Lattice-based group signature scheme with verifier-local revocation.* In *PKC*, vol. 8383 of *LNCS*. Springer, 2014.

LPY12. B. Libert, T. Peters, and M. Yung. *Group signatures with almost-for-free revocation.* In *CRYPTO*, vol. 7417 of *LNCS*. Springer, 2012.

LWW04. J. K. Liu, V. K. Wei, and D. S. Wong. *Linkable spontaneous anonymous group signature for ad hoc groups.* In *ACISP*, vol. 3108 of *LNCS*. 2004.

Ngu05. L. Nguyen. *Accumulators from bilinear pairings and applications.* In *CT-RSA*, vol. 3376 of *LNCS*. Springer, 2005.

NSN04. L. Nguyen and R. Safavi-Naini. *Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings.* In *ASIACRYPT*, vol. 3329 of *LNCS*. Springer, 2004.

Ped91. T. P. Pedersen. *Non-interactive and information-theoretic secure verifiable secret sharing.* In *CRYPTO*, vol. 576 of *LNCS*. Springer, 1991.

RST01. R. L. Rivest, A. Shamir, and Y. Tauman. *How to leak a secret.* In *ASIACRYPT*, vol. 2248 of *LNCS*. Springer, 2001.

SSE$^+$12. Y. Sakai, J. C. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. *On the security of dynamic group signatures: Preventing signature hijacking.* In *PKC*, vol. 7293 of *LNCS*. Springer, 2012.

XY04. S. Xu and M. Yung. *Accountable ring signatures: A smart card approach.* In *CARDIS*, vol. 153 of *IFIP*. Springer, 2004.

A Proof of Theorem 1

Proof. Perfect correctness follows from that of the building blocks and is easy to verify. Anonymity, full unforgeability, traceability and tracing soundness follow from Lemmata 6, 7, 8 and 9, respectively.

Lemma 6. *The accountable ring signature scheme in Fig. 3 is anonymous.*

Proof. We start by replacing the algorithm $\mathrm{SoKSetup}$ of the signature of knowledge with $\mathrm{SoKSimSetup}$, and when answering the challenge query, we use $\mathrm{SoKSimSign}$ instead of $\mathrm{SoKSign}$. By the SimExt security of the SoK, the adversary has a negligible probability in distinguishing between the two settings. This ensures that the signature of knowledge σ_{SoK} reveals no information about the underlying witness.

Next, we replace the algorithm CRSGen of the NIZK system with $\mathrm{SimCRSGen}$ and when answering opening queries, we use $\mathrm{SimProve}$ instead of Prove. By the zero-knowledge property of the NIZK system, the adversary has a negligible probability in distinguishing between the two settings.

Now, we modify the Open oracle into Open' such that instead of decrypting the ciphertext, we run $\mathrm{SoKExtract}$ to extract the verification key vk from the signature of knowledge σ_{SoK}. By the SimExt security of the signature of knowledge, with overwhelming probability in each query, we get the same vk as the plaintext of c.

As we are no longer using the decryption algorithm, by the IND-CPA security of the encryption scheme, the probability of $\mathcal{A}$ winning the anonymity game is close to $\frac{1}{2}$.

Lemma 7. *The accountable ring signature scheme in Fig. 3 is fully unforgeable.*

Proof. We start by running the Setup algorithm as normal with the exception that here we replace $\mathrm{SoKSetup}$ with $\mathrm{SoKSimSetup}$. We forward pp to the adversary. By the simulatability of the signature of knowledge, the adversary has a negligible probability in distinguishing between the two settings. From now on, we use $\mathrm{SoKSimSign}$ instead of $\mathrm{SoKSign}$ when answering Sign queries. The adversary can win in two ways:

- **Case I:** The adversary forges a valid ring signature on a message m w.r.t. an honest ring R where $(pk, \cdot, m, R, \sigma) \notin Q_{\mathrm{Sign}}$. By the SimExt security of the signature of knowledge, we can extract a valid witness for the statement $(pk, R, c) \in \mathcal{L}_{\mathcal{R}_{\mathrm{sig}}}$ from which we obtain (vk, sk) such that $vk := f(sk) \in R$. We use this to break the one-wayness of the function f which contradicts the security of the function f.

- **Case II:** The adversary outputs a valid ring signature $\sigma := (c, \sigma_{\mathrm{SoK}})$ on a message m w.r.t. a ring R and a proof ψ that the honest user with key vk produced the signature while such user never did so. We start by guessing the user the adversary is going to frame. We have a probability $\frac{1}{\eta(\lambda)}$ of guessing correctly, where $\eta(\lambda)$ is a polynomial representing an upper bound on the number of honest users $\mathcal{A}$ uses in the game. By the soundness of the NIZK system, ψ is a proof for a valid statement $(pk, c, vk) \in \mathcal{L}_{\mathcal{R}_{\mathrm{open}}}$. In the game, we abort if $\mathcal{A}$ asks for the secret key of the user we guessed. For all other honest users, we have chosen their key pairs ourselves and thus know their secret keys.
  Again, by the SimExt of the signature of knowledge, with overwhelming probability, we can extract a valid witness for the statement $(pk, R, c) \in \mathcal{L}_{\mathcal{R}_{\mathrm{sig}}}$ from σ_{SoK}, from which we obtain (vk, sk) such that $vk := f(sk) \in R$. We use this to break the one-wayness of f which contradicts the security of the function f.

Lemma 8. *The accountable ring signature scheme in Fig. 3 is traceable.*

Proof. By the security of the signature of knowledge, we are able to extract a valid witness from σ_{SoK} part of the valid signature $\sigma = (c, \sigma_{\mathrm{SoK}})$ the adversary outputs. The witness thus satisfies $vk = f(sk)$ where $vk \in R \subset \mathrm{VK}$, $pk \in \mathrm{PK}$ and $c = \mathrm{Enc}(pk, vk; r)$ for some $r \in \mathrm{Rnd}$ and $sk \in \mathrm{SK}$.

Since $pk = \mathrm{PKEGen}(gk; dk)$, we see from $pk \neq \perp$ that $dk \in \mathrm{DK}$. Correctness of the encryption algorithm implies that $\mathrm{Dec}(dk, c) = vk$, which is the first part of the opening algorithm's output. Now the opening algorithm has a statement (pk, c, vk) and a corresponding witness dk. By the completeness of the NIZK proof system, ψ will verify correctly. This means that the Judge algorithm will output 1 which is a contradiction.

Lemma 9. *The construction satisfies tracing soundness if* SoK *is SimExt secure, the NIZK proof system is sound and the encryption scheme is perfectly correct.*

Proof. The SimExt security of the signature of knowledge ensures that from any signature σ_{SoK} (w.r.t. a statment s) output by the adversary, we can extract a valid witness w such that $(s, w) \in \mathcal{R}_{\mathrm{sig}}$ which eliminates the case that the adversary forges a signature for a statement $s^* \notin \mathcal{L}_{\mathcal{R}_{\mathrm{sig}}}$. If this is not the case, we can use such an adversary to construct another adversary against the SimExt security of the signature of knowledge.

The soundness of the NIZK system for the relation $\mathcal{R}_{\mathrm{open}}$ ensures that ciphertext c contained in the ring signature decrypts to vk, which eliminates the case that the adversary can produce a proof ψ for a statement $s^* \notin \mathcal{L}_{\mathcal{R}_{\mathrm{open}}}$. Finally, the perfect correctness of the public-key encryption scheme (which is regarded as a perfectly-binding commitment scheme) ensures that a ciphertext has a unique decryption.

B Security proofs of our Σ-protocols

B.1 Proof of Lemma 1

Proof. Perfect completeness follows by inspection. The SHVZK simulator, given a challenge x, can simulate the transcript by picking $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C \leftarrow \mathbb{Z}_q$, $C \leftarrow \mathbb{G}$ and computing $f_{j,0} := x - \sum_{i=1}^{n-1} f_{j,i}$, $A := \mathrm{Com}_{ck}(f_{0,0}, \ldots, f_{m-1,n-1}, z_A)B^{-x}$, $D = \mathrm{Com}_{ck}(\{f_{i,j}(x - f_{i,j})\}_{i,j=0}^{m-1,n-1}; z_C)C^{-x}$. In both simulations and real proofs, $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C$ and C are independent, uniformly random and uniquely determine $\{f_{j,0}\}_{j=0}^{m-1}, A, D$, so the simulation is perfect. We also have quasi-unique responses, since two different valid answers $f_{0,1}, \ldots, f_{m-1,n-1}, z_A, z_C$ and $f'_{0,1}, \ldots, f'_{m-1,n-1}, z'_A, z'_C$ to one challenge would break the binding property of $B^x A$ and $C^x D$.

We prove 3-special soundness in three parts. First, we show that any answers to 3 (actually 2) different challenges provide an opening of B. Second, we show that these answers imply that committed values are bits. Finally, we show that they imply that the sum of the committed values is 1. For the first part, suppose that a prover has answered two different challenges x and x' correctly with answers $(f_{0,1},\ldots,f_{m-1,n-1},z_A,z_C)$ and $(f'_{0,1},\ldots,f'_{m-1,n-1},z'_A,z'_C)$. Since we have $B^xA = \mathrm{Com}_{ck}(f_{0,0},\ldots,f_{m-1,n-1};z_A)$ and $B^{x'}A = \mathrm{Com}_{ck}(f'_{0,0},\ldots,f'_{m-1,n-1};z'_A)$, from the first verification equation we have $B^{x-x'} = \mathrm{Com}_{ck}(f_{0,0}-f'_{0,0},\ldots,f_{m-1,n-1}-f'_{m-1,n-1};z_A-z'_A)$. Thus $b_{i,j} = \frac{f_{i,j}-f'_{i,j}}{x-x'}$, with $r = \frac{z_A-z'_A}{x-x'}$, gives us an opening of B. The first verification equation also gives an opening $(a_0,\ldots,a_0;r_A)$ of A using $a_{j,i} = f_{j,i} - xb_{j,i}$ and $r_A = z_A - xr$. Note that by the binding properties of the commitment scheme, the prover cannot know a second opening of A or B, and must respond to any challenge with $f_{j,i} = b_{j,i}x + a_{j,i}$. We can get openings of C and D to values $c_{j,i}$, $d_{j,i}$ from the second equation in a similar way.

By the second verification equation, the values satisfy $c_{j,i}x + d_{j,i} = f_{j,i}(x - f_{j,i}) = b_{j,i}(1-b_{j,i})x^2 + (1-2b_{j,i})a_{j,i}x - a_{j,i}^2$. If this holds for three different x, x' and x'' then the polynomials are identical. So, $b_{j,i}(1-b_{j,i}) = 0$ and $b_{j,i} \in \{0,1\}$ for all i, j.

By construction we have $\sum_{i=0}^{n-1} f_{j,i} = \sum_{i=0}^{n-1} b_{j,i}x + \sum_{i=0}^{n-1} a_{j,i} = x$ for all $j = 0,\ldots,m-1$. This holds for two challenges x and x'. Therefore $\sum_{i=0}^{n-1} b_{j,i} = 1$. □

B.2 Proof of Lemma 2

Proof. First we prove perfect completeness. By the perfect completeness of the Σ-protocol in Fig. 4 we have that $\mathcal{V}_1$ always accepts. Correctness of the last equation follows from the homomorphic property of ElGamal encryption since

$$\begin{aligned}
\prod_{i=0}^{N-1} c_i^{\prod_{j=0}^{m-1} f_{j,i_j}} \cdot \prod_{k=0}^{m-1} G_k^{-x^k} &= \prod_{i=0}^{N-1} c_i^{p_i(x)} \cdot \prod_{k=0}^{m-1} \left(\prod_{i=0}^{N-1} c_i^{p_{i,k}} \cdot \mathrm{Enc}(1;\rho_k) \right)^{-x^k} \\
&= \prod_{i=0}^{N-1} c_i^{p_i(x)} \cdot \prod_{k=0}^{m-1} \left(\prod_{i=0}^{N-1} c_i^{-p_{i,k}x^k} \cdot \mathrm{Enc}(1;-x^k\rho_k) \right) \\
&= \prod_{i=0}^{N-1} c_i^{p_i(x)} \cdot \prod_{i=0}^{N-1} c_i^{-\sum_{k=0}^{m-1} p_{i,k}x^k} \cdot \mathrm{Enc}\left(1; -\sum_{k=0}^{m-1} x^k\rho_k\right) \\
&= \prod_{i=0}^{N-1} c_i^{\delta_{\ell,i}x^m} \cdot \mathrm{Enc}\left(1; -\sum_{k=0}^{m-1} x^k\rho_k\right) \\
&= c_\ell^{x^m} \mathrm{Enc}\left(1; -\sum_{k=0}^{m-1} x^k\rho_k\right) \\
&= \mathrm{Enc}(1; rx^m) \cdot \mathrm{Enc}\left(1; -\sum_{k=0}^{m-1} x^k\rho_k\right) \\
&= \mathrm{Enc}(1; z).
\end{aligned}$$

We now describe a special honest verifier zero-knowledge simulator. It chooses $B \leftarrow \mathbb{G}$ and $G_1,\ldots,G_{m-1} \leftarrow \mathbb{G}^2$. It runs the SHVZK simulator for $\mathcal{P}_1$ to simulate $A, C, D, z_A, z_C, f_{0,1},\ldots,f_{m-1,n-1}$ and computes the $f_{j,0}$'s accordingly. It picks $z \leftarrow \mathbb{Z}_q$ and computes G_0 from the last verification equation.

By the DDH assumption, $G_1,\ldots,G_{m-1}$ in a real proof are indistinguishable from picking random pairs in $\mathbb{G}^2$ as in the simulation. We get independent, uniformly random B and z in both real proofs and simulations. By the perfect SHVZK of the simulator for $\mathcal{P}_1$ we also have the same distribution of $A, B, C, f_{j,i}, z_A, z_C$ as in a real proof. Finally, G_0 is uniquely determined by the last verification equation in both real proofs and in simulations, so the two are indistinguishable. The last verification equation uniquely determines z, thus quasi-unique responses follow from the quasi-unique responses of the underlying Σ-protocol for $\mathcal{R}_1$.

Now we prove the protocol is $(m+1)$-special sound. Suppose an adversary can produce $(m+1)$ different accepting responses $(f_{j,i}^{(0)}, z^{(0)}), \dots, (f_{j,i}^{(m)}, z^{(m)})$ with respect to $m+1$ different challenges $x^{(0)}, \dots, x^{(m)}$ and the same initial message. Assume that $m > 1$. We use 3-special soundness of the Σ-protocol for $\mathcal{R}_1$ to extract openings $\delta_{\ell_j,i}, a_{j,i}$ for B and A with $\delta_{\ell_j,i} \in \{0,1\}$ and $\sum_{i=0}^{n-1} \delta_{\ell_j,i} = 1$. The openings define $\ell := \sum_{j=0}^{m-1} \ell_j n^j$, where ℓ_j is the index of the only 1 in the sequence $(\delta_{\ell_j,0}, \dots, \delta_{\ell_j,n-1})$. Following the proof, all answers satisfy $f_{j,i}^{(e)} = \delta_{\ell_j,i} x^{(e)} + a_{j,i}$ for $0 \le e \le m$, with overwhelming probability due to the binding property of the commitment scheme.

From $\delta_{\ell_j,i}, a_{j,i}$ we can compute the polynomials $p_i(x) = \prod_{j=0}^{m-1} (\delta_{\ell_j,i} x + a_{j,i})$. Note that $p_\ell(x)$ is the only such polynomial with degree m in x. Based on this observation we rewrite the last verification equation as: $c_\ell^{x^m} \cdot \prod_{k=0}^{m-1} \tilde{G}_k^{x^k} = \mathrm{Enc}(1; z)$. Here the $\tilde{G}_k$ values are derived from the initial statement and $\delta_{\ell_j,i}, a_{j,i}$. This equation holds for $x^{(0)}, \dots, x^{(m)}$. Consider the Vandermonde matrix with the eth row given by $(1, x^{(e)}, \dots, (x^{(e)})^m)$. The $x^{(e)}$ values are distinct, so the matrix is invertible. We can thus obtain a linear combination $\theta_0, \dots, \theta_n$ of the rows producing the vector $(0, \dots, 0, 1)$. We deduce $c_\ell = \prod_{e=0}^{m} \left(c_\ell^{(x^{(e)})^m} \cdot \prod_{k=0}^{m-1} \tilde{G}_k^{(x^{(e)})^k} \right)^{\theta_e} = \mathrm{Enc}\left(1; \sum_{e=0}^{m} \theta_e z^{(e)}\right)$, which provides an opening of c_ℓ to the plaintext 1 with randomness $r = \sum_{e=0}^{m} \theta_e z^{(e)}$. □

B.3 Proof of Lemma 3

Proof. Perfect completeness follows by direct verification and the perfect completeness of $(\mathcal{P}_2, \mathcal{V}_2)$. The SHVZK simulator chooses $z_a, z_b, z_s \leftarrow \mathbb{Z}_q$ and $d \leftarrow \mathbb{G}^2$ at random and computes A, B from the verification equations. It runs the perfect SHVZK simulator for $\mathcal{P}_2$ to get a_2 and z_2. By the DDH assumption, d is indistinguishable from the ciphertexts in the real proof. In Both real proofs and simulations, z_a, z_b, z_t are uniformly random and uniquely determine A, B giving us SHVZK. Since the verification equations uniquely determine z_a, z_b and z_s and $(\mathcal{P}_2, \mathcal{V}_2)$ has quasi-unique responses, so must this protocol.

For $(m+1)$-special soundness, consider accepting answers z_a, z_b, z_s and z'_a, z'_b, z'_s to distinct challenges x and x'. From the first verification equation we get $c^{x-x'} = \mathrm{Enc}_{pk}(g^{z_s - z'_s}; z_a - z'_a)$ giving $sk = \frac{z_s - z'_s}{x - x'}$ and $r = \frac{z_a - z'_a}{x - x'}$. The second verification equation gives $d^{x-x'} = \mathrm{Enc}_{pk}(g^{z_s - z'_s}; z_b - z'_b)$ so d also encrypts g^{sk}. Finally, $(m+1)$-special soundness of the Σ-protocol for $\mathcal{R}_2$ then shows that $g^{sk} \in R$. □

B.4 Proof of Lemma 5

Proof. Perfect completeness follows by direct verification. The SHVZK simulator picks $z \leftarrow \mathbb{Z}_q$ and computes A, B from the verification equations. Both in real proofs and simulated proofs z is uniformly random and the verification equations determine the initial message uniquely, so we have perfect simulation. As the first verification equation determines z we have unique responses.

For 2-special soundness, let z and z' be accepting answers to distinct challenges x, x'. The first verification equation gives $pk^{x-x'} = g^{z-z'}$, so $dk = \frac{z-z'}{x-x'}$. The second gives $u^{x-x'} = (v/vk)^{z-z'}$, which shows $u = (v/vk)^{dk}$. Thus, vk was encrypted in (u, v). □