One-out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin

Jens Groth[1]* and Markulf Kohlweiss[2]

[1] University College London
[2] Microsoft Research

Abstract. We construct a 3-move public coin special honest verifier zero-knowledge proof, a so-called Sigma-protocol, for a list of commitments having at least one commitment that opens to 0. It is not required for the prover to know openings of the other commitments. The proof system is efficient, in particular in terms of communication requiring only the transmission of a logarithmic number of commitments.
We use our proof system to instantiate both ring signatures and zerocoin, a novel mechanism for bitcoin privacy. We use our Sigma-protocol as a (linkable) ad-hoc group identification scheme where the users have public keys that are commitments and demonstrate knowledge of an opening for one of the commitments to unlinkably identify themselves (once) as belonging to the group. Applying the Fiat-Shamir transform on the group identification scheme gives rise to ring signatures, applying it to the linkable group identification scheme gives rise to zerocoin.
Our ring signatures are very small compared to other ring signature schemes and we only assume the users' secret keys to be the discrete logarithms of single group elements so the setup is quite realistic. Similarly, compared with the original zerocoin protocol we only rely on a weak cryptographic assumption and do not require a trusted setup.
A third application of our Sigma protocol is an efficient proof of membership of a secret committed value belonging to a public list of values.

Keywords: Sigma-protocol, zero-knowledge, disjunctive proof, ring signature, zerocoin, membership proof.

1 Introduction

A large fraction of deployed cryptographic schemes rely either on cryptographic hash-functions or the discrete logarithm assumption for their security. As a consequence their underlying mathematical structures, compression functions and cyclic prime-order groups respectively, has undergone a lot of cryptanalytic scrutiny. This makes them attractive building-blocks for peer-to-peer applications that operate in a world in which no-one is trusted and everyone is potentially malicious. We revisit two such applications, ring signatures and zerocoin,

* 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 show how to construct both of them using a Σ-protocol that relies only on the security of a homomorphic commitment scheme. When instantiated with Pedersen commitments it is computationally sound, relying only on the discrete logarithm assumption. This results in very efficient instantiations under a weak cryptographic assumption for both ring signatures and zerocoin and reveals a striking connection between the two schemes.

Σ-protocols are 3-move interactive protocols that allow a prover to convince a verifier that a statement is true. The prover sends an initial message, the verifier responds with a random challenge, and the prover sends a response. At the end of the interaction, the verifier looks at the transcript and decides whether to accept or reject the proof that the statement is true. A Σ-protocol should be complete, sound and zero-knowledge in the following sense:

Complete: If the prover knows a witness w for the statement u then she should be able to convince the verifier.

Special sound: If the prover does not know a witness w for the statement, she should not be able to convince the verifier. This is formalized as saying that if the prover can answer several different challenges satisfactorily, then it is possible to extract a witness from the accepting transcripts.

Special honest verifier zero-knowledge: The Σ-protocol should not reveal anything about the prover's witness. This is formalized as saying that given any verifier challenge it is possible to simulate a protocol transcript.

Σ-protocols are widely used. When working in cyclic prime-order groups or RSA-type groups there are very efficient Σ-protocols such as the identification schemes of Schnorr [Sch91] and Guillou-Quisquater [GQ88]. An advantage of Σ-protocols is that they are easy to make non-interactive by using the Fiat-Shamir heuristic [FS86] where a cryptographic hash-function is used to compute the challenge instead of having an online verifier. It can be argued in the random oracle model [BR93] where the hash-function is modeled as a truly random function that this gives us secure non-interactive zero-knowledge proofs. This makes Σ-protocols very useful in the construction of digital signature schemes and encryption schemes, which are non-interactive in nature.

1.1 Our contribution

It is well-known that there are efficient Σ-protocols with linear complexity for NP-complete languages such as circuit satisfiability. We consider statements consisting of N commitments $c_0, \ldots, c_{N-1}$. The prover's claim is that she knows an opening of one of the commitments c_ℓ to the value 0. Our main contribution is a new Σ-protocol for this type of statement that has logarithmic communication complexity.

Our construction works for any additively homomorphic non-interactive commitment scheme (see Sect. 2.1) over $\mathbb{Z}_q$, where q is a large prime. Examples of such commitment schemes include Pedersen commitments [Ped91] and variants of ElGamal encryption [ElG85] where the message is encoded as an exponent.

These commitment schemes specify a commitment key ck, which in the case of Pedersen commitments specifies a prime-order group $\mathbb{G}$ and two group elements g, h. Given a value $m \in \mathbb{Z}_q$ and perhaps some randomness $r \in \mathbb{Z}_q$ it is then possible to compute a commitment, which in the case of Pedersen commitments is computed as $c = g^m h^r$.

Given a commitment key ck and a statement of the form $(c_0, \ldots, c_{N-1})$ the prover who knows an opening $(0, r)$ of one of the commitments $c_\ell = \mathrm{Com}_{ck}(m; r)$ with $m = 0$ can use our Σ-protocol to convince the verifier of having this knowledge. Our Σ-protocol has perfect completeness, i.e., the verifier can always convince the verifier when she has a witness $(0, r)$. It has $(\log N + 1)$-special soundness, which means given $\log N + 1$ accepting transcripts for the statement with distinct challenges $x_0, \ldots, x_{\log N}$ from the verifier, it is possible to compute an opening $(0, r)$ of one of the commitments. Finally, it has special honest verifier zero-knowledge such that for any given challenge x from the verifier it is possible to simulate a transcript without knowing an opening of any of the commitments. When instantiated with the Pedersen commitment scheme our Σ-protocol has perfect special honest verifier zero-knowledge, since the Pedersen commitment scheme is perfectly hiding.

Our Σ-protocol requires the prover to send $4 \log N$ commitments and $3 \log N + 1$ elements in $\mathbb{Z}_q$. When instantiated with Pedersen commitments the prover has to compute roughly $N \log N$ exponentiations and the verifier has to compute roughly N exponentiations. Multi-exponentiation techniques and batching techniques can be used to reduce the computational cost.

If the prover knows the openings of all the commitments its computation can be faster and is determined by the cost of approximately $3N \log N$ multiplications in $\mathbb{Z}_q$ and making $4 \log N$ commitments. This is a huge improvement over existing protocols in the literature like those employed by [DMV13] for rate-limited function evaluation.

Another example where the prover knows the openings is in a membership proof. Here the prover has a commitment c and wants to prove knowledge of an opening to a value u that belongs to a list $\mathcal{L} = \{\lambda_0, \ldots, \lambda_{N-1}\}$. This can be done by forming commitments $c_0 = c \cdot \mathrm{Com}_{ck}(-\lambda_0), \ldots, c_{N-1} = c \cdot \mathrm{Com}_{ck}(-\lambda_{N-1})$ and proving knowledge of an opening of one of the commitments to 0. Due to the special structure of the commitments $c_0, \ldots, c_{N-1}$ this only costs $2N \log N$ multiplications in $\mathbb{Z}_q$ for the prover and $2N$ multiplications in $\mathbb{Z}_q$ for the verifier. This is an improvement over the membership proofs of Bayer and Groth [BG13] that use $O(N \log^2 N)$ multiplications for both the prover and verifier.[3]

1.2 Applications to ring signatures and zerocoin

Ring signatures enable a signer to include herself in an ad-hoc group, a *ring*, and sign a message as a user in the ring without disclosing which one of them is the signer. A ring signature scheme can for instance be used by a whistle

[3] Bayer and Groth's technique also yields a non-membership proof with the same complexity. Our techniques do not provide non-membership proofs.

blower that wants to assure the recipient that the message has been signed by a knowledgeable source, e.g., an employee of a company laundering money, yet at the same time wishes to remain anonymous, such that the company does not fire her when she tells the world about their misdeeds.

Our Σ-protocol gives rise to a natural ad-hoc group identification scheme. All users have a commitment that they know how to open to 0. When a user wants to identify herself as a member of an ad-hoc group, she forms the statement consisting of the commitments $c_0, \ldots, c_{N-1}$ of the users in the group and uses the Σ-protocol to prove she knows an opening of one of the commitments.

By applying the Fiat-Shamir heuristic, i.e., by computing the challenge as a hash of the initial message and the message to be signed, we can convert the group identification scheme into a ring signature scheme. The ring signature scheme inherits the properties of the Σ-protocol. Completeness implies that it is possible for users in the ring to sign messages since they know an opening of one of the commitments to 0. Special soundness implies that ring signatures can only be generated by somebody in the ring, since they imply knowledge of an opening of at least one of the commitments to 0. Special honest verifier zero-knowledge implies that one cannot tell which commitment the signer can open, so the signer remains anonymous within the ring.

Specifying the ring in a ring signature may in the worst case require linear communication but can be amortized over many ring signatures when the same ring is used repeatedly or the ring can be specified indirectly, e.g., by saying the ring is all employees of a particular company. Decreasing the cost of ring signatures has therefore received attention in the cryptographic literature (see related work in Sect. 1.3). Our construction gives rise to a communication-efficient ring signature scheme, where the signature size grows logarithmically in the number of users in the ring. If we use the Pedersen commitment scheme, the ring signature only relies on the discrete logarithm assumption in the random oracle model. Furthermore, the users' keys are just single group elements for which the users know the discrete logarithms. This makes it easy to make ring signatures on top of a pre-existing setup in an organization that has a PKI where users have been assigned public keys consisting of group elements of which they know the discrete logarithms.

Zerocoin, also known as decentralized e-cash, enables users to generate their own coins. Coins become valuable once they are accepted on a public *bulletin board*. These coins can then be anonymously spent by their respective owners without disclosing which coin they are spending. To prevent double spending a secret serial number is revealed during the spending protocol. Zerocoin was proposed as an add on, or decentralized mix, to provide strong anonymity guarantees for bitcoin.

Our Σ-protocol gives rise to a natural one-time ad-hoc group identification scheme. Each user has a commitment c_i to a secret random serial number S that only she knows the opening of. When a user wants to identify herself as a member of an ad-hoc group, she reveals her serial-number S and forms a statement for the Σ-protocol consisting of the commitments

$c_0 \cdot \mathrm{Com}_{ck}(S)^{-1}, \ldots, c_{N-1} \cdot \mathrm{Com}_{ck}(S)^{-1}$ and proves that she knows an opening to zero for one of these commitments. To enforce the one-time property, the verifier accepts the proof only if S has not previously been recorded. By applying the Fiat-Shamir heuristic to this adapted identification scheme one obtains a zerocoin protocol. An important benefit of our construction is that in contrast to existing zerocoin instantiations it does not rely on a trusted setup process assuming the commitment parameters ck have been generated in a way that is publicly verifiable and excludes trapdoors, e.g., using hash functions.

1.3 Related work

There has been a significant amount of research on efficient zero-knowledge proofs. An important early work in this direction was by Kilian [Kil92] that used probabilistically checkable proofs and hash-trees to create an interactive argument for circuit satisfiability with polylogarithmic communication complexity. Kilian's argument has computational soundness; if we require unconditional soundness the communication complexity grows linearly in the witness size as is for instance the case in Ishai et al. [IKOS09]. Using fully homomorphic encryption it is possible to get unconditionally sound proofs of size $|w| + \mathrm{poly}(\lambda)$ [GGI$^+$14], where w is the witness and λ is the security parameter, although this comes at a huge computational cost. There has also been works targeting specifically the discrete logarithm setting such as Cramer and Damgård [CD98] getting linear communication complexity and Groth [Gro09] that gives computationally sound arguments for circuit satisfiability with communication that is proportional to the square root of the circuit size. Our Σ-protocol is much more efficient than these works since it is fine-tuned for a specific language.

Our Σ-protocol can be used to prove that one out of many commitments can be opened to 0, which can be seen as a large disjunctive statement. Cramer, Damgård and Schoenmakers [CDS94] gave a general method to construct Σ-protocols for disjunctive statements. Their technique leads to a Σ-protocol with linear communication complexity. There has been works on related types of statements to the one we consider, i.e., proving something about one out of N elements [Gro09,CGS07,BDD07] that could be potentially be used to get a square root complexity although we are not aware of this actually having been done.

Bayer and Groth [BG13] give logarithmic size arguments for proving membership in a list, i.e., having values $\lambda_0, \ldots, \lambda_{N-1}$ and commitment c to a value λ_ℓ in the list. This can be seen as a dual to our type of disjunctive statement, we in contrast have many commitments $c_0, \ldots, c_{N-1}$ but just a single value $\lambda_\ell = 0$ and want to prove one of the commitments c_ℓ contains this value. The membership proofs of Bayer and Groth rely on an efficient proof of correct polynomial evaluation in a a secret committed value.

The strategy both here and in [BG13] is to construct polynomials of degree $\log N$ in a random challenge chosen by the verifier. In both cases, we can see the constructed polynomials as arising from a weighted sum or product (with weights depending on the statement) over the vertices of a hypercube of dimension $\log N$

but the papers differ in the weights at the vertices of the hypercube and how they are used. In [BG13] the weights are the coefficients of the polynomial P and the vertices in the hypercube contain N powers u^i of a point u where the polynomial is evaluated. In our paper the weights are the commitments and the hypercube has a single non-zero vertex corresponding to the commitment (out of N) that we are interested in. The correct evaluation of the hypercube is built and verified using polynomials of degree $\log N$ in a challenge x.

Ring signatures were introduced by Rivest, Shamir and Tauman [RST01] and Bender, Katz and Morselli [BKM09] provide rigorous security definitions for ring signatures and generic constructions based on trapdoor permutations. The idea of using Σ-protocol for anonymous identification within a group has been proposed before, see e.g. [CDS94,Cam97], and has found use in the constructions of ring signatures based on non-interactive zero-knowledge proofs in the random oracle model or using pairings. Courtois [Cou01] constructs a ring signature scheme based on a Σ-protocol for the MinRank problem. Abe et al. [AOS04] use disjunctive proofs to demonstrate possession of one out of N secret keys to construct ring signatures. The instantiation of their scheme based on the discrete logarithm assumption and using the same group for all users is similar to our ring signature except their Σ-protocol based on techniques from [CDS94] give signatures that grow linearly in the size of the ring. Herranz and Sáez [HS03] also give a linear size ring signature based on the discrete logarithm problem in the random oracle model. There are also several pairing-based constructions of ring signatures including [BGLS03,CWLY06,SW07,Boy07,CGS07]. The most efficient without random oracles is by Chandran, Groth and Sahai [CGS07] who exhibit square root size ring signatures using pairing based non-interactive zero-knowledge proofs.

The smallest ring signatures are by Dodis et al. [DKNS04] who use accumulators based on the strong RSA assumption [CL02] to get ring signatures consisting of a constant number of group elements in the random oracle model. Their construction, however, requires a setup that includes an RSA modulus, which may not be readily available. Furthermore, since RSA moduli have to be of size $\frac{\lambda^3}{\text{polylog}\lambda}$ bits to resist factorization attacks they end up with ring signatures where the size has cubic growth in the security parameter. Nguyen [Ngu05] also give constant size ring signatures in the random oracle model, but requires a linear size public key and relies on pairing-based cryptography, which also leads to a ring signature size of $\frac{\lambda^3}{\text{polylog}\lambda}$ bits. In contrast, our construction is based on the discrete logarithm assumption and if we use elliptic curve groups with group elements of size $O(\lambda)$ bits, we end up with an asymptotic quasilinear complexity of $O(\lambda \log N) = O(\lambda \log \lambda)$ bits for our ring signatures when the ring size N is polynomial in the security parameter.

Zerocoin was introduced by Miers et al. [MGGR13]. Their construction is in the random oracle model and uses an accumulator based on the strong RSA assumption together with cut-and-choose techniques to prove group representations in the exponent. The cut-and-choose technique results in their proofs of spending having quintic growth in the security parameter. Danezis et al. [DFKP13] show how to efficiently construct zerocoin using succinct arguments of knowledge (SNARKs). Ben-Sasson et al. [BSCG$^+$14] extend zerocoin with secret balances to build a SNARK-based alternative currency. All known zerocoin constructions rely on a common reference string with a specific probability distribution, except for the original zerocoin protocol when used together with the techniques of Sander [San99] to construct theoretically efficient RSA UFOs[4]

While existing constructions are constant in the number of coins on the bulletin board, RSA accumulator based zerocoin proofs consist of $\sim 50,000$ bytes, compared with $32(7\log N + 1)$ bytes in our construction using 256-bit elliptic curve groups. This means that for all practical purposes the logarithmic size will be preferable. The constant size of SNARK based constructions, usually below a dozen group elements, is hard to beat, and indeed these constructions pay for this by having to rely on knowledge of exponent assumptions.

2 Preliminaries

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 picking randomness r at random and setting $y = A(x; r)$. We also write $y \leftarrow S$ for sampling y uniformly at random from a set S.

All algorithms in our schemes get a security parameter $\lambda \in \mathbb{N}$ as input written in unary 1^λ. The intuition is that the higher the security parameter, the greater security we get.

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 $f(\lambda) \approx 1$.

2.1 Homomorphic commitment schemes

A non-interactive commitment scheme allows a sender to construct a commitment to a value. The sender may later open the commitment and reveal the value. The receiver of the commitment can then verify the opening and check that indeed it was this particular value that was committed in the first place. A commitment scheme must be hiding and binding. Hiding means that the commitment does not reveal the committed value. Binding means that the sender cannot open the commitment to two different values.

Formally, a non-interactive commitment scheme is a pair of probabilistic polynomial time algorithms $(\mathcal{G}, \mathrm{Com})$. The setup algorithm $ck \leftarrow \mathcal{G}(1^\lambda)$ generates a commitment key ck. The commitment key specifies a message space $\mathcal{M}_{ck}$, a randomness space $\mathcal{R}_{ck}$ and a commitment space $\mathcal{C}_{ck}$. The commitment algorithm

[4] An RSA UFO is a large integer generated in a specific way from a source of uniformly random bits such that there is overwhelming probability that there are two large random primes that cannot be split from each other in a factorization of the integer. Known constructions of RSA UFOs yield integers much larger than standard RSA moduli, so in practice protocols built on RSA UFOs are inefficient.

combined with the commitment key specifies a function $\mathrm{Com}_{ck} : \mathcal{M}_{ck} \times \mathcal{R}_{ck} \to \mathcal{C}_{ck}$. Given a message $m \in \mathcal{M}_{ck}$ the sender picks uniformly at random $r \leftarrow \mathcal{R}_{ck}$ and computes the commitment $c = \mathrm{Com}_{ck}(m; r)$.

Definition 1 (Hiding). *A non-interactive commitment scheme* $(\mathcal{G}, \mathrm{Com})$ *is hiding if a commitment does not reveal the value. For all probabilistic polynomial time stateful adversaries* $\mathcal{A}$

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

where $\mathcal{A}$ *outputs* $m_0, m_1 \in \mathcal{M}_{ck}$*. If the probability is exactly* $\frac{1}{2}$ *we say the commitment scheme is* perfectly *hiding.*

Definition 2 (Binding). *A non-interactive commitment scheme* $(\mathcal{G}, \mathrm{Com})$ *is binding if a commitment can only be opened to one value. For all probabilistic polynomial time adversaries* $\mathcal{A}$

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

where $\mathcal{A}$ *outputs* $m_0, m_1 \in \mathcal{M}_{ck}$ *and* $r_0, r_1 \in \mathcal{R}_{ck}$*. If the probability is exactly 0 we say the commitment scheme is* perfectly *binding.*

Definition 3 (Strongly binding). *A non-interactive commitment scheme* $(\mathcal{G}, \mathrm{Com})$ *is strongly binding if a commitment can only be opened in one way, i.e., not even the randomness can change. For all probabilistic polynomial time adversaries* $\mathcal{A}$

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

where $\mathcal{A}$ *outputs* $m_0, m_1 \in \mathcal{M}_{ck}$ *and* $r_0, r_1 \in \mathcal{R}_{ck}$*.*

We will focus on the case where the message and randomness spaces are $\mathbb{Z}_q$ for a prime $q > 2^\lambda$ specified in the commitment key ck. Furthermore, we require the commitment scheme to be homomorphic, which means that the commitment space is also a group (written multiplicatively) and we have for all well-formed commitment keys ck and $m_0, m_1 \in \mathcal{M}_{ck}$ and $r_0, r_1 \in \mathcal{R}_{ck}$ that

$$\mathrm{Com}_{ck}(m_0; r_0) \cdot \mathrm{Com}_{ck}(m_1; r_1) = \mathrm{Com}_{ck}(m_0 + m_1; r_0 + r_1).$$

Pedersen commitments. The Pedersen commitment scheme [Ped91] is a natural example of a homomorphic commitment scheme with the desired properties. The key generation algorithm $\mathcal{G}$ outputs a description of a cyclic group $\mathbb{G}$ of prime order q and random generators g, h. The commitment key is $ck = (\mathbb{G}, q, g, h)$. To commit to $m \in \mathbb{Z}_q$ the committer picks randomness $r \in \mathbb{Z}_q$ and computes $\mathrm{Com}_{ck}(m; r) = g^m h^r$. The commitment scheme is perfectly hiding and computationally strongly binding under the discrete logarithm assumption.

2.2 Σ-protocols

A Σ-protocol is a special type of 3-move interactive proof system that allows a prover to convince a verifier that a statement is true. The prover sends an initial message to the verifier, the verifier picks a random *public coin* challenge $x \leftarrow \{0,1\}^\lambda$, and the prover responds to the challenge. Finally the verifier checks the transcript of the interaction and decides whether the proof should be accepted or rejected.

We assume the existence of a probabilistic polynomial time setup algorithm $\mathcal{G}$ that generates a common reference string ck that is available to all parties. In this paper the common reference string will be a public key ck for a homomorphic non-interactive commitment scheme. It is worth noting that such keys may be set up using prime order groups based on the discrete logarithm problem, which makes it possible to sample them from uniformly random bits. So at the cost of a small overhead stemming from the use of uniformly random bits, we could set our schemes up in the common *random* string model.

Let R be a polynomial time decidable ternary relation, we call w a witness for a statement u if $(ck, u, w) \in R$. We define the CRS-dependent language

$$L_{ck} = \{u \mid \exists w : (ck, u, w) \in R\}$$

as the set of statements u that have a witness w in the relation R.

A Σ-protocol for R is a triple of probabilistic polynomial time stateful interactive algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$. The following run of a Σ-protocol describes the interaction of the algorithms

$ck \leftarrow \mathcal{G}(1^\lambda)$: Generates the common reference string.
$a \leftarrow \mathcal{P}(ck, u, w)$: Given $(ck, u, w) \in R$ the prover generates an initial message a.
$x \leftarrow \{0,1\}^\lambda$: The verifier's challenge x is chosen uniformly at random.
$z \leftarrow \mathcal{P}(x)$: The prover responds to the challenge x.
$b \leftarrow \mathcal{V}(ck, u, a, x, z)$: The verifier algorithm, which will always be deterministic in this paper, returns 1 if accepting the proof and 0 if rejecting the proof.

The triple $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is called a Σ-protocol for R if it is complete, special sound and special honest verifier zero-knowledge as defined below.

Definition 4 (Perfect completeness). *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is perfectly complete if for all $\lambda \in \mathbb{N}, ck \leftarrow \mathcal{G}(1^\lambda)$ and (u, w) such that $(ck, u, w) \in R$*

$$\Pr\Big[a \leftarrow \mathcal{P}(ck, u, w); x \leftarrow \{0,1\}^\lambda; z \leftarrow \mathcal{P}(x) : \mathcal{V}(ck, u, a, x, z) = 1\Big] = 1.$$

A Σ-protocol should be a proof of knowledge; a prover should only be able to respond to a random challenge if the prover "knows" a witness for the statement u. We define this in the form of special soundness, which says that given responses to a number of different challenges it is possible to compute a witness for the statement.

Definition 5 (n-special soundness). *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is n-special sound if there is an efficient extraction algorithm $\mathcal{X}$ that can compute the witness given n accepting transcripts with the same initial message. Formally, for all probabilistic polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[\begin{matrix} ck \leftarrow \mathcal{G}(1^\lambda); (u, a, x_1, z_1, \ldots, x_n, z_n) \leftarrow \mathcal{A}(ck) \\ w \leftarrow \mathcal{X}(ck, u, a, x_1, z_1, \ldots, x_n, z_n) \end{matrix} : (ck, u, w) \in R\right] \approx 1,$$

where $\mathcal{A}$ outputs distinct $x_1, \ldots, x_n \in \{0,1\}^\lambda$ and for all $i \in \{1, \ldots, n\}$ the transcript is accepting, i.e., $\mathcal{V}(ck, u, a, x_i, z_i) = 1$.

We say the proof is perfect *n-special sound if the probability is exactly 1.*

A non-standard requirement that many Σ-protocols satisfy is that responses are unique, or at least quasi unique, i.e. given an accepting proof an adversary cannot find a new valid response for the challenge in the proof. This non-malleability property is important to achieve simulation soundness [Fis05,FKMV12].

Definition 6 (Quasi unique response). *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ has quasi unique responses if for all probabilistic polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[\begin{matrix} ck \leftarrow \mathcal{G}(1^\lambda) \\ (u, a, x, z, z') \leftarrow \mathcal{A}(ck) \end{matrix} : \begin{matrix} z \neq z' \\ \mathcal{V}(ck, u, a, x, z) = \mathcal{V}(ck, u, a, x, z') = 1 \end{matrix}\right] \approx 1.$$

A Σ-protocol is zero-knowledge if it does not leak information about the witness beyond what can be inferred from the truth of the statement. We will present Σ-protocols that are special honest verifier zero-knowledge in the sense that if the verifier's challenge is known in advance, then it is possible to simulate the entire argument without knowing the witness.

Definition 7 (Special honest verifier zero-knowledge (SHVZK)). *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is special honest verifier zero knowledge if there exists a probabilistic polynomial time simulator $\mathcal{S}$ such that for all interactive probabilistic polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[ck \leftarrow \mathcal{G}(1^\lambda); (u, w, x) \leftarrow \mathcal{A}(ck); a \leftarrow \mathcal{P}(ck, u, w); z \leftarrow \mathcal{P}(x) : \mathcal{A}(a, z) = 1\right]$$
$$\approx \Pr\left[ck \leftarrow \mathcal{G}(1^\lambda); (u, w, x) \leftarrow \mathcal{A}(ck); (a, z) \leftarrow \mathcal{S}(ck, u, x) : \mathcal{A}(a, z) = 1\right],$$

where $\mathcal{A}$ outputs (u, w, x) such that $(ck, u, w) \in R$ and $x \in \{0,1\}^\lambda$.

The Σ-protocol is said to be perfect *special honest verifier zero-knowledge if the two probabilities are exactly equal to each other.*

In real life applications, special honest verifier zero-knowledge may not suffice since a malicious verifier may give non-random challenges. However, it is easy to convert an SHVZK argument into a full zero-knowledge argument secure against *arbitrary* verifiers in the common reference string model using standard techniques (see e.g. [Dam00]). The conversion can be very efficient and only incur a small additive overhead, so we will in the paper without loss of generality just focus on building efficient SHVZK arguments.

For our application to ring signatures and zerocoin we do not need full zero-knowledge. It suffices to have have witness-indistinguishability, which is implied by perfect special honest verifier zero-knowledge. A Σ-protocol is witness indistinguishable if it is infeasible to distinguish which of several possible witnesses the prover uses.

Definition 8 (Witness-indistinguishability). *$(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is witness indistinguishable if for all interactive polynomial time adversaries $\mathcal{A}$*

$$\Pr\left[\begin{array}{l} ck \leftarrow \mathcal{G}(1^\lambda); (u, w_0, w_1) \leftarrow \mathcal{A}(ck); b \leftarrow \{0,1\} \\ a \leftarrow \mathcal{P}(ck, u, w_b); x \leftarrow \mathcal{A}(a); z \leftarrow \mathcal{P}(x) \end{array} : \mathcal{A}(z) = b\right] \approx \frac{1}{2},$$

where $\mathcal{A}$ outputs (u, w_0, w_1) such that $(ck, u, w_0) \in R$ and $(ck, u, w_1) \in R$ and $x \in \{0,1\}^\lambda$.

The Σ-protocol is perfectly *witness-indistinguishable if the probability is exactly half.*

Theorem 1 ([CDS94]). *A perfect SHVZK Σ-protocol is perfectly witness-indistinguishable.*

Proof. Perfect special honest verifier zero-knowledge implies the existence of a simulator that for any $x \in \{0,1\}^\lambda$ simulates (a, z) that is perfectly indistinguishable from a real proof. This means that conditioned on any particular $x \in \{0,1\}^\lambda$, two different witnesses w_0 and w_1 both lead to proofs with the same probability distribution as the simulation. This implies that conditioned on a, x we get the same probability distribution of the response z regardless of which witness was used. Moreover, the perfect special honest verifier zero-knowledge property also guarantees that the initial messages a are distributed identically regardless of the witness used. $\square$

2.3 Σ-protocol for commitment to 0 or 1

We will now give a well-known example of a Σ-protocol for knowledge of a committed value being 0 or 1, which will be useful later. Let ck be a commitment key for a homomorphic commitment scheme as described in Sect. 2.1 and let R be the relation consisting of commitments to 0 or 1, with the witnesses being openings of the commitment, i.e.,

$$R = \left\{ \big(ck, c, (m, r)\big) \mid c = \mathrm{Com}_{ck}(m; r) \text{ and } m \in \{0,1\} \text{ and } r \in \mathbb{Z}_q \right\}.$$

Fig. 1 gives a Σ-protocol $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ for R, where $\mathcal{G}$ is the key generation algorithm for the commitment scheme, and where $\mathcal{P}, \mathcal{V}$ are running on $ck \leftarrow \mathcal{G}(1^\lambda)$, $m \in \{0,1\}$ and $r \in \mathbb{Z}_q$.

Theorem 2. *The Σ-protocol in Fig. 1 for commitment to $m \in \{0,1\}$ is perfectly complete, perfect 2-special sound and perfect SHVZK.*

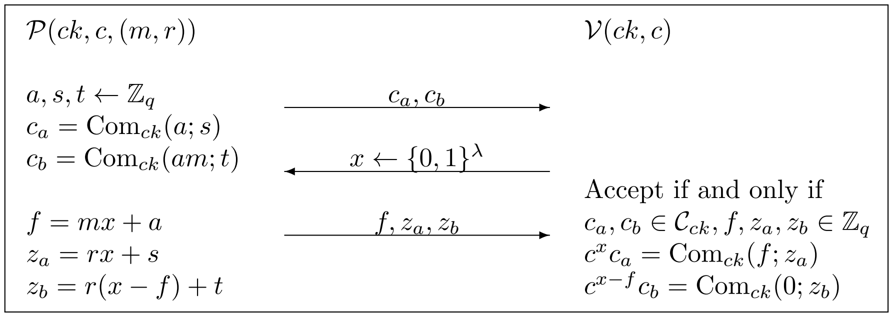
$\mathcal{P}(ck, c, (m, r))$
$\mathcal{V}(ck, c)$
$a, s, t \leftarrow \mathbb{Z}_q$
$c_a = \mathrm{Com}_{ck}(a; s)$
$c_b = \mathrm{Com}_{ck}(am; t)$
c_a, c_b
$x \leftarrow \{0,1\}^\lambda$
$f = mx + a$
$z_a = rx + s$
$z_b = r(x - f) + t$
f, z_a, z_b
Accept if and only if
$c_a, c_b \in \mathcal{C}_{ck}, f, z_a, z_b \in \mathbb{Z}_q$
$c^x c_a = \mathrm{Com}_{ck}(f; z_a)$
$c^{x-f} c_b = \mathrm{Com}_{ck}(0; z_b)$

Fig. 1. Σ-protocol for commitment to $m \in \{0, 1\}$.

Proof. By the homomorphic property of the commitment scheme $c^{x-f}c_b$ is a commitment to $m(x - f) + am = m(1 - m)x - ma + am = x(1 - m)m$, which is 0 if $m \in \{0, 1\}$. With this in mind, it is straightforward to verify that the Σ-protocol is perfectly complete.

We will now show that the Σ-protocol is perfect 2-special sound. Given responses f, z_a, z_b and f', z'_a, z'_b to two different challenges x and x' on the same initial commitments c_a, c_b we get by combining the verification equations that $c^{x-x'} = \mathrm{Com}_{ck}(f - f'; z_a - z'_a)$ and $c^{x-f-x'+f'} = \mathrm{Com}_{ck}(0; z_b - z'_b)$. Defining $m = \frac{f-f'}{x-x'}$ and $r = \frac{z_a - z'_a}{x-x'}$ we extract an opening of $c = \mathrm{Com}_{ck}(m; r)$. Furthermore, since $c^{x-x'+f'-f} = c^{(1-m)(x-x')} = \mathrm{Com}_{ck}(m(1 - m)(x - x'); r(1 - m)(x - x')) = \mathrm{Com}_{ck}(0; z_b - z'_b)$ we either get a breach of the binding property of the commitment scheme (in which case the opening of c can be modified into an opening to $m \in \{0, 1\}$) or we have $m(1 - m) = 0$, which implies $m \in \{0, 1\}$.

Finally, let us prove that the protocol is perfect special honest verifier zero-knowledge. The simulator given ck, c and x first chooses $f, z_a, z_b \leftarrow \mathbb{Z}_q$. It then computes $c_a = c^{-x}\mathrm{Com}_{ck}(f; z_a)$ and $c_b = c^{f-x}\mathrm{Com}_{ck}(0; z_b)$. Both in a real proof and in the simulation this gives independent and uniformly random $f, z_a, z_b \in \mathbb{Z}_q$. Conditioned on these values and x the verification equations uniquely determine c_a, c_b in both real proofs and simulated proofs. This shows that real proofs and simulated proofs have identical probability distributions. $\square$

3 Σ-protocol for one out of N commitments containing 0

We will now give a Σ-protocol for knowledge of one out of N commitments $c_0, \ldots, c_{N-1}$ being a commitment to 0. More precisely, we will give a Σ-protocol for the relation

$$R = \left\{ (ck, (c_0, \ldots, c_{N-1}), (\ell, r)) \;\middle|\; \begin{array}{l} c_0, \ldots, c_{N-1} \in \mathcal{C}_{ck} \text{ and } \ell \in \{0, \ldots, N-1\} \\ \text{and } r \in \mathbb{Z}_q \text{ and } c_\ell = \mathrm{Com}_{ck}(0; r) \end{array} \right\}.$$

To explain the idea behind the Σ-protocol let us for simplicity assume the commitment scheme is perfectly binding such that each commitment has a unique committed value. Saying that one of the commitments contains 0 is equivalent to saying there exists an index ℓ such that $\prod_{i=0}^{N-1} c_i^{\delta_{i\ell}}$ is a commitment to 0, where $\delta_{i\ell}$ is Kronecker's delta, i.e., $\delta_{\ell\ell} = 1$ and $\delta_{i\ell} = 0$ for $i \neq \ell$. We can always copy some commitments in the statement, so let us without loss of generality assume $N = 2^n$. Writing $i = i_1 \ldots i_n$ and $\ell = \ell_1 \ldots \ell_n$ in binary, we have $\delta_{i\ell} = \prod_{j=1}^{n} \delta_{i_j \ell_j}$ so we can reformulate what we want to prove as $\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{n} \delta_{i_j \ell_j}}$ being a commitment to 0.

The prover will start by making commitments $c_{\ell_1}, \ldots, c_{\ell_n}$ to the bits $\ell_1, \ldots, \ell_n$. She then engages in n parallel Σ-protocols as described in Sect. 2.3 to demonstrate knowledge of openings of these commitments to values $\ell_j \in \{0, 1\}$. In the Σ-protocols for $\ell_j \in \{0, 1\}$ the prover reveals $f_1, \ldots, f_n$ of the form $f_j = \ell_j x + a_j$. Let $f_{j,1} = f_j = \ell_j x + a_j = \delta_{1\ell_j} x + a_j$ and $f_{j,0} = x - f_j = (1 - \ell_j)x - a_j = \delta_{0\ell_j} x - a_j$. Then we have for each i that the product $\prod_{j=1}^{n} f_{j,i_j}$ is a polynomial of the form

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

The idea is now that the prover in the initial message will send commitments $c_{d_0}, \ldots, c_{d_{n-1}}$ that will be used to cancel out the low order coefficients corresponding to $x^0, \ldots, x^{n-1}$. Meanwhile the high order coefficient for x^n will guarantee the commitment c_ℓ can be opened to 0. More precisely, the verifier will at the end check that

$$\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{n} f_{j,i_j}} \cdot \prod_{k=0}^{n-1} c_{d_k}^{-x^k}$$

is a commitment to 0, which by the Schwartz-Zippel lemma has negligible probability of being true unless indeed c_ℓ is a commitment to 0.

Fig. 2 gives the full Σ-protocol $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ for R with $\mathcal{G}$ being the key generation algorithm for the commitment scheme and $\mathcal{P}, \mathcal{V}$ running on $ck \leftarrow \mathcal{G}(1^\lambda)$, $c_0, \ldots, c_{N-1} \in \mathcal{C}_{ck}$, $\ell \in \{0, \ldots, N-1\}$ and $r \in \mathbb{Z}_q$ such that $c_\ell = \mathrm{Com}_{ck}(0; r)$. Without loss of generality we assume $N = 2^n$.

Theorem 3. *The Σ-protocol in Fig. 2 for knowledge of one out of N commitments opening to 0 is perfectly complete. It is (perfect) $(n+1)$-special sound if the commitment scheme is (perfectly) binding. It is (perfect) special honest verifier zero-knowledge if the commitment scheme is (perfectly) hiding.*

Proof. To see that the Σ-protocol is complete observe that $\prod_{j=1}^{n} f_{j,i_j}$ is a polynomial in the challenge x of the form $p_i(x) = \delta_{i\ell} x^n + \sum_{k=0}^{n-1} p_{i,k} x^k$. When c_ℓ is a commitment to 0 we therefore get that $c_\ell^{\prod_{j=1}^{n} f_{j,\ell_j}}$ in the verification equation is a commitment to 0, while the other commitments c_i get raised to polynomials of degree $n-1$ in x as $c_i^{\prod_{j=1}^{n} f_{j,i_j}}$ in the verification equation. With this in mind straightforward verification shows that the Σ-protocol is perfectly complete.

$\mathcal{P}(ck, (c_0, \ldots, c_{N-1}), (\ell, r))$		$\mathcal{V}(ck, (c_0, \ldots, c_{N-1})))$
For $j = 1, \ldots, n$		
$r_j, a_j, s_j, t_j, \rho_k \leftarrow \mathbb{Z}_q$		
$c_{\ell_j} = \mathrm{Com}_{ck}(\ell_j; r_j)$		
$c_{a_j} = \mathrm{Com}_{ck}(a_j; s_j)$	$c_{\ell_1}, c_{a_1}, c_{b_1}, c_{d_0}, \ldots,$	
$c_{b_j} = \mathrm{Com}_{ck}(\ell_j a_j; t_j)$	$\xrightarrow{c_{\ell_n}, c_{a_n}, c_{b_n}, c_{d_{n-1}}}$	Accept if and only if
$c_{d_k} = \prod_i c_i^{p_{i,k}} \mathrm{Com}_{ck}(0; \rho_k)$		$c_{\ell_1}, \ldots, c_{d_{n-1}} \in \mathcal{C}_{ck}$
using $k = j - 1$		$f_1, \ldots, z_d \in \mathbb{Z}_q$
and $p_{i,k}$ from (1)	$\xleftarrow{x \leftarrow \{0,1\}^\lambda}$	For all $j \in \{1, \ldots, n\}$
		$c_{\ell_j}^x c_{a_j} = \mathrm{Com}_{ck}(f_j; z_{a_j})$
For $j = 1, \ldots, n$	$f_1, z_{a_1}, z_{b_1}, \ldots,$	$c_{\ell_j}^{x-f_j} c_{b_j} = \mathrm{Com}_{ck}(0; z_{b_j})$
$f_j = \ell_j x + a_j$	$\xrightarrow{f_n, z_{a_n}, z_{b_n}, z_d}$	$\prod_i c_i^{\prod_{j=1}^n f_{j,i_j}} \cdot \prod_{k=0}^{n-1} c_{d_k}^{-x^k}$
$z_{a_j} = r_j x + s_j$		$= \mathrm{Com}_{ck}(0; z_d)$
$z_{b_j} = r_j(x - f_j) + t_j$		using $f_{j,1} = f_j$
$z_d = rx^n - \sum_{k=0}^{n-1} \rho_k x^k$		and $f_{j,0} = x - f_j$

Fig. 2. Σ-protocol for commitment to $m = 0$ in list $c_0, \ldots, c_{N-1}$.

We will now show how to convert an adversary with probability ε of breaking $(n+1)$-soundness, into an adversary with approximately the same runtime that has probability ε of breaking the binding property of the commitment scheme.

Suppose the adversary creates $n + 1$ accepting responses $f_1^{(0)}, \ldots, z_d^{(0)}, \ldots, f_1^{(n)}, \ldots, z_d^{(n)}$ to $n + 1$ different challenges $x^{(0)}, \ldots, x^{(n)}$ on the same initial message $c_{\ell_1}, \ldots, c_{d_{n-1}}$.

The 2-special soundness of the Σ-protocol from Sect. 2.3 gives us openings of $c_{\ell_1}, \ldots, c_{\ell_n}$ of the form $c_{\ell_j} = \mathrm{Com}_{ck}(\ell_j; r_j)$ with $\ell_j \in \{0, 1\}$. From the verification equations it is then easy to get openings of $c_{a_j} = \mathrm{Com}_{ck}(a_j; s_j)$. Unless the adversary breaks the binding property of the commitment scheme, it must hold for all challenges that $f_j^{(0)} = \ell_j x^{(0)} + a_j, \ldots, f_j^{(n)} = \ell_j x^{(n)} + a_j$ for all $j = 1, \ldots, n$.

The form of the f_j's gives us that $f_{j,1} = \ell_j x + a_j$ and $f_{j,0} = (1 - \ell_j)x - a_j$. For $i \neq \ell$ we therefore get that $\prod_{j=1}^n f_{j,i_j}$ is a degree $n-1$ polynomial $p_i(x)$ and for $i = \ell$ it is a polynomial of the form $p_\ell(x) = x^n + \ldots$. This means we can rewrite the last verification as

$$c_\ell^{x^n} \cdot \prod_{k=0}^{n-1} c_{*_k}^{x^k} = \mathrm{Com}_{ck}(0; z_d)$$

for some fixed $c_{*_0}, \ldots, c_{*_{n-1}}$ that can be computed from commitments in the statement and the initial message.

Observe that the vectors $(1, x^{(e)}, \ldots, (x^{(e)})^n)$ can be viewed as rows in a Vandermonde matrix and since $x^{(0)}, \ldots, x^{(n)}$ are all different the matrix is invertible and we can therefore find a linear combination $(\alpha_0, \ldots, \alpha_n)$ of the rows that gives us the vector $(0, \dots, 0, 1)$. Combining the $n+1$ accepting verification equations we therefore get

$$c_\ell = \prod_{e=0}^{n} \left(c_\ell^{(x^{(e)})^n} \cdot \prod_{k=0}^{n-1} c_{*_k}^{(x^{(e)})^k} \right)^{\alpha_e} = \mathrm{Com}_{ck}(0; \sum_{e=0}^{n} \alpha_e z_d^{(e)}).$$

This gives us an extracted opening of c_ℓ to 0 with randomness $r = \sum_{e=0}^{n} \alpha_e z_d^{(e)}$.

Finally, let us describe a special honest verifier zero-knowledge simulator that is given a challenge $x \in \{0,1\}^\lambda$. It starts by picking the elements of the response uniformly at random as $f_1, \dots, z_d \leftarrow \mathbb{Z}_q$. It then chooses $c_{\ell_1}, \dots, c_{\ell_n}, c_{d_1}, \dots, c_{d_{n-1}} \leftarrow \mathrm{Com}_{ck}(0)$ as random commitments to 0. Finally, it computes $c_{a_j} = c_{\ell_j}^{-x} \mathrm{Com}_{ck}(f_j; z_{a_j})$ and $c_{b_j} = c_{\ell_j}^{x-f} \mathrm{Com}_{ck}(0; z_{b_j})$ to finish the simulation of the proofs that $c_{\ell_1}, \dots, c_{\ell_n}$ contain 0 and $c_{d_0} = \prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{n} f_{j,i_j}} \cdot \prod_{k=1}^{n-1} c_{d_k}^{-x^k} \cdot \mathrm{Com}_{ck}(0; -z_d)$ to satisfy the last verification equation. It returns the simulated initial message and response $(c_{\ell_1}, \dots, c_{d_{n-1}}, f_1, \dots, z_d)$.

We will now argue that an adversary that distinguishes the simulation from a real argument with ε advantage can be turned into an adversary that breaks the hiding property of the commitment scheme with $\frac{\varepsilon}{2n-1}$ advantage. First, we observe that in both real proofs and simulated proofs $f_1, \dots, z_d$ are uniformly random in $\mathbb{Z}_q$. Furthermore, the verification equations uniquely determine $c_{a_1}, c_{b_1}, \dots, c_{a_n}, c_{b_n}$ and c_{d_0} conditioned on $f_1, \dots, z_d$ and $c_{\ell_1}, \dots, c_{d_{n-1}}$ both in real and in simulated proofs. The adversary's advantage of ε must therefore come from being able to distinguish real and simulated commitments $c_{\ell_1}, \dots, c_{\ell_n}, c_{d_1}, \dots, c_{d_{n-1}}$. A standard hybrid argument gives us a $\frac{\varepsilon}{2n-1}$ advantage in breaking the hiding property of the commitment scheme. □

We state in the following two lemmas a couple of additional properties that will be useful later.

Lemma 1. *If the commitment scheme is strongly binding, the Σ-protocol in Fig. 2 has quasi unique responses.*

Lemma 2. *For each possible initial message in the Σ-protocol in Fig. 2 there is negligible probability that it will be chosen by the SHVZK simulator.*

Proof. The simulator picks c_{ℓ_1} as a random commitment to 0. We will now argue that c_{ℓ_1} has negligible probability of matching a fixed value c. We have by the hiding and binding properties

$$\Pr\left[ck \leftarrow \mathcal{G}(1^\lambda); c \leftarrow \mathrm{Com}_{ck}(0); c_{\ell_1} \leftarrow \mathrm{Com}_{ck}(0) : c_{\ell_1} = c\right]$$
$$\approx \Pr\left[ck \leftarrow \mathcal{G}(1^\lambda); c \leftarrow \mathrm{Com}_{ck}(0); c_{\ell_1} \leftarrow \mathrm{Com}_{ck}(1) : c_{\ell_1} = c\right] \approx 0.$$

□

Efficiency. The prover sends $4 \log N$ commitments and $3 \log N + 1$ field elements. With N being polynomial in the security parameter the prover therefore only sends $O(\log \lambda)$ commitments and field elements. If we use the Pedersen commitment scheme in an elliptic curve based group where the group elements are of size $O(\lambda)$ bits the total communication cost is just $O(\lambda \log \lambda)$ bits.

If we are using the Pedersen commitment scheme the prover's cost is dominated by n multi-exponentiations of N group elements when computing $c_{d_0}, \ldots, c_{d_{n-1}}$. Using multi-exponentiation techniques [Lim00] we can reduce the cost of computing $c_{d_0}, \ldots, c_{d_{n-1}}$ to roughly N single exponentiations. Computing the commitments is more efficient than this once pre-computation techniques are factored in; and the polynomial coefficients $p_{i,k}$ can be computed by fast polynomial multiplication techniques, which will have significantly smaller cost than the exponentiations because they are done over $\mathbb{Z}_q$.

The verifier's computation is dominated by the multi-exponentiation $\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{n} f_{j,i_j}}$. If we are using Pedersen commitments this can be done at a cost that is not much higher than $\frac{N}{\log N}$ single exponentiations.

When prover knows openings of all commitments. If the prover knows openings of all commitments $c_0, \ldots, c_{N-1}$ she can reduce her computation significantly. Observe that if $c_i = \mathrm{Com}_{ck}(m_i; \gamma_i)$ then $c_{d_k} = \prod_{i=0}^{N-1} c_i^{p_{i,k}} \mathrm{Com}_{ck}(0; \rho_k) = \mathrm{Com}_{ck}(d_k, \phi_k + \rho_k)$ where $d_0, \phi_0, \ldots, d_{n-1}, \phi_{n-1}$ are coefficients in the two polynomials

$$d(x) = \sum_{k=0}^{n-1} d_k x^k = \sum_{i=0}^{N-1} m_i p_i(x) \qquad \phi(x) = \sum_{k=0}^{n-1} \phi_k x^k = \sum_{i=0}^{N-1} \gamma_i p_i(x) - \gamma_\ell x^n,$$

where the latter holds because $p_i(x) = \delta_{i\ell} x^n + \sum_{k=0}^{n-1} p_{i,k} x^k$, so $p_\ell(x)$ is the only polynomial with a non-zero coefficient for x^n.

The two polynomials $d(x)$ and $\phi(x)$ can be efficiently computed using Lagrange interpolation. Picks n distinct elements $\omega_1, \ldots, \omega_n \in \mathbb{Z}_q$ and evaluate $d(\omega_1), \phi(\omega_1), \ldots, d(\omega_n), \phi(\omega_n)$ from which the coefficients $d_0, \phi_0, \ldots, d_{n-1}, \phi_{n-1}$ can be computed in time depending only on $n = \log N$.

We will now show that given $\omega \in \mathbb{Z}_q$ it is possible to compute both $d(\omega)$ and $\phi(\omega)$ using $3N$ multiplications in $\mathbb{Z}_q$. Each $f_{j,0}$ and $f_{j,1}$ is a degree 1 polynomial in x and we can compute all $f_{j,0}(\omega), f_{j,1}(\omega)$ for $j = 1, \ldots, n$ using a few modular additions for each of them. Now, $p_i(\omega) = \prod_{j=1}^{n} f_{j,i_j}(\omega)$, so we can view $p_0(\omega), \ldots, p_{N-1}(\omega)$ as leaves on a binary tree, where the root is $\prod_{j=1}^{n} f_{j,0}(\omega)$ and for each parent at level $j-1$ we let the left child be the same as the parent and the right child be the parent multiplied by $\frac{f_{j,1}(\omega)}{f_{j,0}(\omega)}$. The leaves can be computed using roughly $N = 2^n$ multiplications, which gives us $p_0(\omega), \ldots, p_{N-1}(\omega)$. Computing the sums $d(\omega) = \sum_{i=0}^{N-1} m_i p_i(\omega)$ and $\phi(\omega) = \sum_{i=0}^{N-1} \gamma_i p_i(\omega) - \gamma_\ell \omega^n$ costs an additional $2N$ multiplications, for a total of $3N + o(N)$ multiplications to compute $d(\omega), \phi(\omega)$. Doing this for n distinct elements $\omega_1, \ldots, \omega_n$ costs roughly $3N \log N$ multiplications. Once we have the evaluations in the n elements, we can at moderate cost compute $d_0, \phi_0, \ldots, d_{n-1}, \phi_{n-1}$ using Lagrange interpolation.

The prover's computation when she knows the openings of all the commitments is therefore determined by the cost of approximately $3N \log N$ multiplications in $\mathbb{Z}_q$ and making $4 \log N$ commitments.

Membership proof. The Σ-protocol in Fig. 2 can be used to construct a membership proof. We are given a commitment c and a set of values $\lambda_0, \ldots, \lambda_{N-1}$ and want to prove that we know an opening of the commitment c to one of the values λ_ℓ. This can be done using our 1-out-of-N Σ-protocol by defining $c_0 = c \cdot \mathrm{Com}_{ck}(-\lambda_0; 0), \ldots, c_{N-1} = c \cdot \mathrm{Com}_{ck}(-\lambda_{N-1}; 0)$ and proving there is a c_ℓ with an opening to 0.

From the prover's perspective this is a case where all the commitments have known openings $(\lambda_\ell - \lambda_i, \gamma_\ell)$ of commitment c_i. Observe that all the commitments have the same randomness, which implies $\phi(x) = 0$ and reduces the computation to $2N \log N$ multiplications for the prover. To see that $\phi(x) = 0$ recall that $\phi(x) = \gamma_\ell \sum_{i=0}^{N-1} p_i(x) - \gamma_\ell x^n$ and

$$\sum_{i=0}^{N-1} p_i(x) = \sum_{i=0}^{N-1} \prod_{j=1}^{n} f_{j,i_j}(x) = \prod_{j=1}^{n} (f_{j,0}(x) + f_{j,1}(x)) = \prod_{j=1}^{n} x = x^n.$$

The verifier is also very efficient, he can compute the product

$$\begin{aligned}
\prod_{i=0}^{N-1} c_i^{\prod_{j=1}^{n} f_{j,i_j}} &= \prod_{i=0}^{N-1} (c \cdot \mathrm{Com}_{ck}(-\lambda_i; 0))^{p_i(x)} \\
&= c^{\sum_{i=0}^{N-1} p_i(x)} \cdot \mathrm{Com}_{ck}\big(-\sum_{i=0}^{N-1} \lambda_i p_i(x); 0\big) \\
&= c^{x^n} \cdot \mathrm{Com}_{ck}\big(-\sum_{i=0}^{N-1} \lambda_i p_i(x); 0\big)
\end{aligned}$$

using $2N$ multiplications, which dominates the computation for large N.

This efficiency compares favorably with the membership proof in Bayer and Groth [BG13]. They prove membership by demonstrating the committed value is a root in the polynomial $P(u) = \prod_{i=0}^{N-1}(u - \lambda_i)$, but the initial step of computing the coefficients of the polynomial requires $O(N \log^2 N)$ multiplications (and only if the modulus q is of a form suitable for using the Fast Fourier Transform).

Bayer and Groth's method also gives rise to a non-membership proof: prove that the polynomial $P(u)$ does not evaluate to 0 to show the committed value u does not belong to the list. Our Σ-protocol does not appear to yield a non-membership proof.

4 Ring Signature

Ring signatures allow users to sign messages on behalf of ad-hoc groups that include themselves. The ad-hoc groups are called rings and contain public keys for the signer and the other users that the signer has chosen to include to include. We formally define ring-signatures in the following section.

Our Σ-protocol for one out of N commitments containing 0 can be used as an ad-hoc group identification scheme. Each user has a commitment to 0 with the private key being the randomness used. To identify yourself as a member of a group you prove that you know the opening of one of the commitments to 0. We can use the Fiat-Shamir heuristic to transform the ad-hoc group identification scheme into a ring signature scheme.

4.1 Definitions

A ring signature scheme consists of a quadruple of PPT algorithms $(\mathrm{Setup}, \mathrm{KGen}, \mathrm{Sign}, \mathrm{Vfy})$ for generating a common key available to all users, generating keys for users, signing messages and verifying ring signatures.

$pp \leftarrow \mathrm{Setup}(1^\lambda)$: Generates and outputs public parameters pp available to all users.

$(vk, sk) \leftarrow \mathrm{KGen}(pp)$: Generates a public verification key vk and a private signing key sk.

$\sigma \leftarrow \mathrm{Sign}_{pp,sk}(M, R)$: Outputs a signature σ on the message $M \in \{0,1\}^*$ with respect to the ring $R = (vk_1, \dots, vk_N)$. We require that there is a $vk \in R$ such that (vk, sk) is a valid key pair output by $\mathrm{KGen}(pp)$.

$b \leftarrow \mathrm{Vfy}_{pp}(M, R, \sigma)$: Verifies a purported ring signature σ on a message M with respect to the ring of public keys R. It outputs 1 if accepting and 0 if rejecting the ring signature.

The quadruple $(\mathrm{Setup}, \mathrm{KGen}, \mathrm{Sign}, \mathrm{Vfy})$ is a ring signature scheme with perfect anonymity if it is correct, unforgeable and anonymous as defined below.

Definition 9 (Perfect correctness). *We require that a user can sign any message on behalf of a ring where she is a member. A ring signature scheme* $(\mathrm{Setup}, \mathrm{KGen}, \mathrm{Sign}, \mathrm{Vfy})$ *has perfect correctness if for all adversaries* $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathrm{Setup}(1^\lambda); (vk, sk) \leftarrow \mathrm{KGen}(pp) \\ (M, R) \leftarrow \mathcal{A}(pp, vk, sk); \sigma \leftarrow \mathrm{Sign}_{pp,sk}(M, R) \end{array} : \begin{array}{c} \mathrm{Vfy}_{pp}(M, R, \sigma) = 1 \\ \text{or } vk \notin R \end{array}\right] = 1.$$

Definition 10 (Unforgeability). *A ring signature scheme* $(\mathrm{Setup}, \mathrm{KGen}, \mathrm{Sign}, \mathrm{Vfy})$ *is unforgeable (with respect to insider corruption) if it is infeasible to forge a ring signature on a message without controlling one of the members in the ring. Formally, it is unforgeable when for all probabilistic polynomial time adversaries* $\mathcal{A}$

$$\Pr\left[pp \leftarrow \mathrm{Setup}(1^\lambda); (M, R, \sigma) \leftarrow \mathcal{A}^{\mathrm{VKGen},\mathrm{Sign},\mathrm{Corrupt}}(pp) : \mathrm{Vfy}_{pp}(M, R, \sigma) = 1\right] \approx 0,$$

- VKGen *on the ith query picks randomness* r_i*, runs* $(vk_i, sk_i) \leftarrow \mathrm{KGen}(pp; r_i)$ *and returns* vk_i*.*
- $\mathrm{Sign}(i, M, R)$ *returns* $\sigma \leftarrow \mathrm{Sign}_{pp,sk_i}(M, R)$*, provided* (vk_i, sk_i) *has been generated by* VKGen *and* $vk_i \in R$*.*

– Corrupt(i) *returns* r_i *(from which* sk_i *can be computed) provided* (vk_i, sk_i) *has been generated by* VKGen.
– $\mathcal{A}$ *outputs* (M, R, σ) *such that* Sign *has not been queried with* $(*, M, R)$ *and* R *only contains keys* vk_i *generated by* VKGen *where* i *has not been corrupted.*

Definition 11 (Perfect anonymity). *A ring signature scheme* (Setup, KGen, Sign, Vfy) *has perfect anonymity, if a signature on a message* M *under a ring* R *and key* vk_{i_0} *looks exactly the same as a signature on the message* M *under the ring* R *and key* vk_{i_1}. *This means that the signer's key is hidden among all the honestly generated keys in the ring. Formally, we require that for any adversary* $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathrm{Setup}(1^\lambda); (M, i_0, i_1, R) \leftarrow \mathcal{A}^{\mathrm{KGen}(pp)}(pp) \\ b \leftarrow \{0,1\}; \sigma \leftarrow \mathrm{Sign}_{pp, sk_{i_b}}(M, R) \end{array} : \mathcal{A}(\sigma) = b\right] = \frac{1}{2},$$

where $\mathcal{A}$ *chooses* i_0, i_1 *such that* $(vk_{i_0}, sk_{i_0}), (vk_{i_1}, sk_{i_1})$ *have been generated by the key generation oracle* KGen(pp) *and* $vk_{i_0}, vk_{i_1} \in R$.

We remark that perfect anonymity implies anonymity against full key exposure, which is the strongest definition of anonymity of ring signatures in [BKM09].

4.2 Construction

An additively homomorphic commitment perfectly hiding scheme $(\mathcal{G}, \mathrm{Com})$ as defined in Sect. 2.1 and the Σ-protocol $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ in Fig. 2 for one out of N commitments being a commitment to 0 can be combined to build an ad-hoc group identification scheme. We generate a commitment key as setup and let the users' verification keys be commitments to 0. In order to identify herself as a member of an ad-hoc group with N members, the user uses the Σ-protocol to prove that she knows an opening of one of the commitments. If her commitment is among the commitments in the ad-hoc group the correctness of the Σ-protocol guarantees that she manages to identify herself as a member. If on the other hand her commitment is not among the commitments in the group, then the $(\lceil \log N \rceil + 1)$-special soundness of the Σ-protocol guarantees that she has negligible chance of answering a challenge and being accepted. Finally, the special honest verifier zero-knowledge property of the Σ-protocol implies that it is witness-indistinguishable, i.e., even a malicious verifier cannot tell which commitment opening it is that she knows how to open.

We will use the Fiat-Shamir heuristic to make the ad-hoc group identification scheme non-interactive. Let $\mathcal{H}$ be a hash-function generator such that $H \leftarrow \mathcal{H}(1^\lambda)$ returns a hash-function $H : \{0,1\}^* \to \{0,1\}^\lambda$. By computing the challenge x in the Σ-protocol using the hash function on the initial message in the Σ-protocol and the message to be signed, we get a transformation of the ad-hoc group identification protocol to a ring signature scheme. Modeling the hash-function H as a random oracle allows us to give a heuristic proof that the ring signature scheme is unforgeable. The ring signature scheme is described in Fig. 3

$\mathrm{Setup}(1^\lambda)$	$\mathrm{Sign}_{pp,sk}(M,R)$	$\mathrm{Vfy}_{pp}(M,R,\sigma)$
$ck \leftarrow \mathcal{G}(1^\lambda)$	Parse $R=(c_0,\ldots,c_{N-1})$	Parse $R=(c_0,\ldots,c_{N-1})$
$H \leftarrow \mathcal{H}(1^\lambda)$	with $c_\ell = \mathrm{Com}_{ck}(0;sk)$	Parse $\sigma=(a,z)$
Return $pp=(ck,H)$		
	$a \leftarrow \mathcal{P}(ck,R,(\ell,sk))$	
$\mathrm{KGen}(pp)$	$x = H(ck,M,R,a)$	$x = H(ck,M,R,a)$
$r \leftarrow \mathbb{Z}_q$	$z \leftarrow \mathcal{P}(x)$	
$c = \mathrm{Com}_{ck}(0;r)$		
Return $(vk,sk)=(c,r)$	Return $\sigma=(a,z)$	Return $\mathcal{V}(ck,R,a,x,z)$

Fig. 3. Ring signature based on Σ-protocol $(\mathcal{G},\mathcal{P},\mathcal{V})$ for 1-out-of-N commitments containing 0.

Theorem 4. *The scheme* (Setup, KGen, Sign, Vfy) *is a ring signature scheme with perfect correctness. It has perfect anonymity if the commitment scheme is perfectly hiding. It is unforgeable in the random oracle model if the commitment scheme is perfectly hiding and computationally binding.*

Proof. Perfect correctness follows from the perfect completeness of the Σ-protocol. Perfect anonymity follows from the perfect witness indistinguishability of the Σ-protocol, which guarantees that it is impossible to distinguish which secret key has been used to generate the ring signature.

To see that the ring signature scheme is unforgeable we will rely on the $(n+1)$-special soundness of the Σ-protocol in Fig. 2 and model the hash-function H as a random oracle. Consider a polynomial time adversary $\mathcal{A}$ that makes at most $q_V(\lambda), q_S(\lambda)$ and $q_H(\lambda)$ queries to VKGen, Sign and the random oracle, respectively, and for infinitely many $\lambda \in \mathbb{N}$ has at least $\frac{1}{p(\lambda)}$ probability of breaking the unforgeability property for a positive polynomial p. We will show that it can be used to construct a polynomial time attack that breaks the binding property of the commitment scheme with approximately $\frac{1}{2q_V(\lambda)p(\lambda)}$ chance on infinitely many $\lambda \in \mathbb{N}$. We will without loss of generality assume the adversary checks that it has made a successful forgery, which simplifies the proof since it guarantees the adversary does at some point call the random oracle on a query (ck, M, R, a) corresponding to the forged ring signature.

Given the public parameters we first pick at random $j \in \{1,\ldots,q_V\}$ and set $vk_j = \mathrm{Com}_{ck}(1;r_j)$ for $r_j \leftarrow \mathbb{Z}_q$. Our goal is to run $\mathcal{A}$ using this key for user j and hoping to use rewinding to get $n+1$ forgeries with a ring R that includes vk_j. The $(n+1)$-soundness of the SHVZK argument may permit extraction of an opening of some vk_i to $(0, r_i)$. By the perfect hiding property of the commitment scheme, with probability $\frac{1}{q_V}$ we have $i = j$ giving us a breach of the commitment scheme's binding property.

Let us now give more details of how the attack works. Whenever $\mathcal{A}$ queries VKGen we run as in a real ring signature scheme, except on the jth query where we return vk_j. If $\mathcal{A}$ ever queries Corrupt(j) we abort (type I). If $\mathcal{A}$ queries Sign(j, M, R) we pick $x \leftarrow \{0,1\}^\lambda$ at random and use the special honest verifier zero-knowledge simulator to simulate the proof (a, z). We then program the random oracle $H(\cdot)$ to have $H(ck, M, R, a) = x$, except if (ck, M, R, a) has already been queried before in which case we abort (type II).

In the end, $\mathcal{A}$ tries to create a forged ring signature with uncorrupted users in the ring and where the signature does not come from the signing oracle. If $\mathcal{A}$ fails to create a forgery we halt. Otherwise, we get a successful forged ring signature $\sigma = (a, z)$ on M using ring R coming from a random oracle query $H(ck, M, R, a)$ used to get a challenge $x^{(0)}$. We now rewind the adversary to the point where it made the query $H(ck, M, R, a)$ used in the forged signature and give it random answers to the oracle query until it has produced n additional forged ring signatures with challenges $x^{(1)}, \ldots, x^{(n)}$ using the same query. As above, in each rewinding, if the simulation of a signature leads to reuse of a query to the oracle we abort (type II). Furthermore, if the number of rewindings exceed $2p(\lambda)n$ we halt.

If the adversary after rewinding gave us answers to a total of $n + 1$ distinct challenges, we can now use the $(n+1)$-special soundness property to either break the binding property of the commitment scheme or to get an opening $(0, r_i)$ of some $vk_i = \text{Com}_{ck}(0; r_i)$. With probability $\frac{1}{q_V}$ we have $vk_i = vk_j$, giving us a breach of the binding property of the commitment scheme.

Let us analyze the attack described above. A useful starting point is running the real unforgeability experiment, i.e., instead of picking $vk_j = \text{Com}_{ck}(1; r_j)$ we pick $vk_j = \text{Com}_{ck}(0; r_j)$ as a correctly generated key and answer all queries honestly (so we do not have type I or II aborts). Let us consider some $\lambda \in \mathbb{N}$ where $\mathcal{A}$ has at least $\frac{1}{p(\lambda)}$ chance of creating a successful forgery. Observe that an adversary that has probability γ of using a specific random oracle query in a successful forgery will be rewound $n = \gamma \cdot \frac{n}{\gamma}$ times on average on this query to sample n additional forgeries. The probability of the attack entering the rewinding stage and exceeding $2p(\lambda)n$ rewindings will therefore be at most $\frac{1}{2p(\lambda)}$, since otherwise we would exceed the expected number of rewindings. This means we have at least $\frac{1}{p(\lambda)} - \frac{1}{2p(\lambda)} = \frac{1}{2p(\lambda)}$ chance of getting $n + 1$ successful forgeries using a specific oracle query (ck, M, R, a).

Switching to simulation of ring signatures instead of giving real ring signatures may result in type II aborts when the simulation accidentally results in an oracle query $H(ck, M, R, a)$ that has been used before, but with a different challenge. However, Lemma 2 tells us that the simulator has negligible probability of colliding with another oracle query: the probability of a single simulation hitting a specific oracle query is a negligible function $\nu(\lambda)$ and with a maximum of $q_S(\lambda)$ signing queries in each run of the adversary, and a total of $q_H(\lambda) + q_S(\lambda)$ random oracle queries in each run of the adversary we get an upper bound of $(2p(\lambda)n + 1)q_S(\lambda)(q_H(\lambda) + q_S(\lambda)\nu(\lambda)$ for the probability of running into a type II abort.

Another problem that can arise is a collision in the $n+1$ challenges we get after rewinding. With a maximum of $q_S(\lambda) + q_H(\lambda)$ queries to the random oracle in each run of $\mathcal{A}$ we get a total risk of $\frac{2((1+2p(\lambda)n)(q_S(\lambda)+q_H(\lambda)))^2}{2^\lambda}$ of having a collision in any random oracle outputs. Avoiding type II aborts and collisions leaves us with $\frac{1}{2p(\lambda)} - (2p(\lambda)n+1)q_S(\lambda)(q_H(\lambda)+q_S(\lambda)\nu(\lambda) - \frac{2((1+2p(\lambda)n)(q_S(\lambda)+q_H(\lambda)))^2}{2^\lambda} \approx \frac{1}{2p(\lambda)}$ chance of being able to use $(n+1)$-special soundness to break the commitment scheme or extract an opening $(0, r_i)$ of some vk_i in the ring of a ring signature forgery.

If we extract an opening $(0, r_i)$ of some vk_i in the ring of a ring signature forgery there is $\frac{1}{q_V(\lambda)}$ chance that $i = j$. If $i = j$ we observe as a part of this being a successful forgery, the adversary never queried $\mathrm{Corrupt}(j)$, so we do not have any type I aborts. Since the commitment scheme is perfectly hiding, the switch to using $vk_j = \mathrm{Com}_{ck}(1; r'_j)$ does not change the success probability of the attack. But now an opening of $vk_i = vk_j$ to $vk_i = \mathrm{Com}_{ck}(0; r_i)$ corresponds to a breach of the binding property of the commitment scheme. So for infinitely many $\lambda \in \mathbb{N}$ our attack has close to or higher than $\frac{1}{2q_V(\lambda)p(\lambda)}$ chance of breaking the binding property of the commitment scheme. The attack runs in polynomial time since it will make at most $1+2p(\lambda)n$ runs of the polynomial time adversary $\mathcal{A}$. $\square$

Instantiation with Pedersen commitments. The Pedersen commitment scheme is a natural candidate for the commitment scheme. When our ring signature scheme is instantiated with the Pedersen commitment scheme, the public keys are of the form $c = h^r$, i.e., they are single group elements and the corresponding secret keys are the discrete logarithms.

The instantiation with Pedersen commitments requires a simple setup that is realistic in many settings. Consider any organization where a standard group $\mathbb{G}$ is used for all users and their secret keys are discrete logarithms of public group elements. The ring signature easily fits on top of this setup.

The ring signature scheme yields small signatures. The signature size is logarithmic in the number of ring members and instantiated over a compact group where elements have size $O(\lambda)$ it is $O(\lambda \log N) = O(\lambda \log \lambda)$ bits. This compares favorably with all previous ring signature schemes.

The signer computes $\log N$ multi-exponentiations of N elements to generate a ring signature and the verifier uses a multi-exponentiation of N elements to verify a ring signature. However, when the same ring is used many times or there is significant overlap between different rings, the cost of verification can be reduced to $O(N)$ multiplications in $\mathbb{Z}_q$ by batching the verification of many signatures.

5 Zerocoin

Zerocoin enables users to generate their own coins which become valuable by public consensus by being included on a bulletin board. These coins can then be spent anonymously with double spending being prevented by a secret serial number encoded in each coin which is revealed during the spend protocol.

5.1 Definition

A zerocoin scheme consists of a quadruple of PPT algorithms $(\text{Setup}, \text{Mint}, \text{Spend}, \text{Vfy})$ for generating a common setup available to all users, generating coins, generating proofs that a coin was spend to pay for a transaction and verifying proofs of spending.

- $pp \leftarrow \text{Setup}(1^\lambda)$. Generates public parameters available to all users.
- $(c, skc) \leftarrow \text{Mint}(pp)$. Mints a coin c together with a key skc used to authorize its spending.
- $(\pi, S) \leftarrow \text{Spend}_{pp,skc}(M, c, C)$. On input some transaction string $M \in \{0,1\}^*$ and an arbitrary set of coins C containing c, the algorithm outputs a proof π and a serial number S. We require that skc is a valid key for coin c as produced by $\text{Mint}(pp)$ and that $c \in C$.
- $b \leftarrow \text{Vfy}_{pp}(M, S, C, \pi)$. Verifies a purported proof π of a spend transaction with string M of a coin with serial number S from the set of coins C.

The transaction string M in the call to Spend is intended, e.g., for the identity of the transaction recipient, or the terms of a contract.

The quadruple $(\text{Setup}, \text{Mint}, \text{Spend}, \text{Vfy})$ is a zerocoin scheme with perfect anonymity if it is correct, balanced and anonymous as defined next.

Definition 12 (Perfect correctness). *We require that a user can spend any coin with respect to any set of coins. A zerocoin scheme* $(\text{Setup}, \text{Mint}, \text{Spend}, \text{Vfy})$ *has perfect correctness if for all adversaries* $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \text{Setup}(1^\lambda); (c, skc) \leftarrow \text{Mint}(pp) \\ (M, C) \leftarrow \mathcal{A}(pp, c, skc) \\ (\pi, S) \leftarrow \text{Spend}_{pp,skc}(M, c, C \cup \{c\}) \end{array} : \ \text{Vfy}_{pp}(M, S, C \cup \{c\}, \pi) = 1\right] = 1.$$

Our balance definition is a strengthening of the original zero-coin definition. As for ring signature unforgeability, we allow for Corrupt queries that give the adversary access to the randomness of coins.

Definition 13 (Balance). *A zerocoin scheme* $(\text{Setup}, \text{Mint}, \text{Spend}, \text{Vfy})$ *is balanced (with respect to insider corruption) if an adversary cannot spend more coins than he controls. Formally, it is balanced when for all probabilistic polynomial time adversaries* $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \text{Setup}(1^\lambda) \\ (\tilde{c}_1, \dots, \tilde{c}_m, \mathcal{S}_1, \dots, \mathcal{S}_m, \mathcal{S}_{m+1}) \leftarrow \mathcal{A}^{\text{CoinGen,Spend,Corrupt}}(pp) \end{array} : \forall_i . \text{Vfy}_{pp}(\mathcal{S}_i) = 1\right] \approx 0,$$

- CoinGen *on query number i selects randomness r_i, runs $(c_i, skc_i) \leftarrow$* $\text{Mint}(pp; r_i)$ *and returns c_i after adding c_i to a set $\mathcal{C}$.*
- $\text{Spend}(i, M, C)$ *returns* $(\pi, S) \leftarrow \text{Spend}_{pp,skc_i}(M, c_i, C)$*, provided (c_i, skc_i) has been generated by* CoinGen *and was not leaked using* $\text{Corrupt}(i)$*. The oracle records (M, S, C, π) in a set $\mathcal{T}$.*

- Corrupt(i) *provided* (c_i, skc_i) *has been generated by* CoinGen *runs* $(\pi, S) \leftarrow \text{Spend}_{pp,skc_i}(\text{`` ''}, c_i, \{c_i\})$ *to determine the serial number of the coin and then returns* r_i *(from which* skc_i *can be computed). The oracle removes any tuple matching the pattern* $(*, S, *, *)$ *from* $\mathcal{T}$ *and records* $(*, S, *, *)$ *in* $\mathcal{T}$*.*
- $\mathcal{A}$ *outputs* $\tilde{c}_1, \ldots, \tilde{c}_m, \mathcal{S}_1, \ldots, \mathcal{S}_m, \mathcal{S}_{m+1}$ *such that* $\mathcal{S}_i = (M_i, S_i, C_i, \pi_i)$*,* $C_i \subset \mathcal{C} \cup \{\tilde{c}_1, \ldots, \tilde{c}_m\}$*, no* $\mathcal{S}_i$ *matches a pattern in* $\mathcal{T}$*, and all* S_i *are distinct.*

Definition 14 (Perfect anonymity). (Setup, Mint, Spend, Vfy) *has perfect anonymity if a proof of spending with transaction string* M *for a set of coins* C *and coin* c_{i_0} *looks exactly the same as a proof of spending with transaction string* M *for the set* C *and coin* c_{i_1}*. This means that the spender's coin is hidden among all the honestly generated coins in the set. Formally, we require that for any adversary* $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \text{Setup}(1^\lambda); (M, i_0, i_1, C) \leftarrow \mathcal{A}^{\text{Mint}(pp)}(pp) \\ b \leftarrow \{0,1\}; (\pi, S) \leftarrow \text{Spend}_{pp,skc_{i_b}}(M, c, C) \end{array} : \mathcal{A}(\pi, S) = b\right] = \frac{1}{2},$$

where $\mathcal{A}$ *chooses* i_0, i_1 *such that* $(c_{i_0}, skc_{i_0}), (c_{i_1}, skc_{i_1})$ *have been generated by the minting oracle* Mint(pp) *and* $c_{i_0}, c_{i_1} \in C$*.*

5.2 Construction

While a ring-signature scheme can be constructed from an ad-hoc group identification scheme using the Fiat-Shamir heuristic, a zerocoin scheme can be obtained from a *linkable* ad-hoc group identification scheme. In particular, almost the same construction can be used to construct zerocoin schemes from a Σ-protocol for 1-out-of-N commitments containing 0. Instead of public keys that are commitments to 0 we now employ coins that are commitments to serial numbers. We homomorphically subtract a serial number S from all coins used in a statement by multiplying them with $\text{Com}_{ck}(S; 0)^{-1}$ before computing the proof, such that the commitment with this serial number turns into a commitment to 0. The zerocoin scheme is described in Fig. 4.

Theorem 5. *The scheme* (Setup, Mint, Spend, Vfy) *is a zerocoin scheme with perfect correctness. It has perfect anonymity if the commitment scheme is perfectly hiding. It is balanced in the random oracle model if the commitment scheme is perfectly hiding and strongly binding.*

Proof. Perfect correctness follows from the perfect completeness of the Σ-protocol. Perfect anonymity follows from the perfect witness indistinguishability of the Σ-protocol, which guarantees that it is impossible to distinguish which coin has been used to generate a proof of spending.

To see that the zerocoin scheme is balanced we will rely on the $(n+1)$-special soundness of the Σ-protocol and model the hash-function H as a random oracle. We will show that a zerocoin adversary $\mathcal{A}$, which for a positive polynomial p and an infinite number of $\lambda \in \mathbb{N}$ has more than $\frac{1}{p(\lambda)}$ chance of forging more spending

$\mathrm{Setup}(1^\lambda)$	$\mathrm{Spend}_{pp,skc}(M,c,C)$	$\mathrm{Vfy}_{pp}(M,S,C,\pi)$
$ck \leftarrow \mathcal{G}(1^\lambda)$	Parse $C=(c_0,\dots,c_{N-1})$	Parse $C=(c_0,\dots,c_{N-1})$
$H \leftarrow \mathcal{H}(1^\lambda)$	and $skc=(r,S)$	Parse $\pi=(a,z)$
Return $pp=(ck,H)$	with $c_\ell = c = \mathrm{Com}_{ck}(S;r)$	
$\mathrm{Mint}(pp)$	$c'_i \leftarrow c_i \cdot \mathrm{Com}_{ck}(S;0)^{-1}$	$c'_i \leftarrow c_i \cdot \mathrm{Com}_{ck}(S;0)^{-1}$
$r \leftarrow \mathbb{Z}_q$	$a \leftarrow \mathcal{P}(ck,(c'_0,\dots,c'_{N-1}),(\ell,r))$	
$S \leftarrow \mathbb{Z}_q$	$x = H(ck,M,S,C,a)$	$x = H(ck,M,S,C,a)$
$c \leftarrow \mathrm{Com}_{ck}(S;r)$	$z \leftarrow \mathcal{P}(x)$	
$skc \leftarrow (r,S)$		Return
Return (c, skc)	Return $\pi=(a,z)$ and S	$\mathcal{V}(ck,(c'_0,\dots,c'_{N-1}),a,x,z)$

Fig. 4. Zerocoin protocol based on Σ-protocol $(\mathcal{G},\mathcal{P},\mathcal{V})$ for 1-out-of-N commitments containing 0.

proofs than controlled coins can be used to construct an attack on the strong binding property of the commitment scheme.

Given the public parameters pp we start by forwarding them to $\mathcal{A}$. We simulate the random oracle and the CoinGen, and Corrupt oracles honestly. We use the SHVZK simulator and the random oracle programmability to answer Spend queries with fresh random serial numbers. If this fails because the pre-image is already in the oracle list we abort with "Error 1". Finally $\mathcal{A}$ outputs m coins $(\tilde{c}_1,\dots,\tilde{c}_m)$ and $m+1$ valid spending proofs $(\mathcal{S}_1,\dots,\mathcal{S}_m,\mathcal{S}_{m+1})$. We will for simplicity assume in the proof that $\mathcal{A}$ checks that all its spendings are valid such that for each spent coin the random oracle has been queried on $H(ck,M,S,C,a)$.

For each $1 \le i \le m+1$ we do the following. We find the first entry on the oracle list where $\mathcal{A}$ asked (ck,M_i,S_i,C_i,a_i) to the random oracle; if we created the entry ourselves during the simulation of a Spend query we abort with "Error 2". We then simulate a fresh copy of $\mathcal{A}$ identically up to the point where the above query was asked and answer with a different uniformly random value from the oracle. We repeat this process until we obtain $n_i = \lceil \log |C_i| \rceil + 1$ proofs with the same a_i. If the total number of rewindings exceeds $2p(\lambda)\sum_{i=1}^{m+1} n_i$ we halt. Since the expected number of rewindings for each query is n_i, we have at least $\frac{1}{2p(\lambda)}$ chance of getting the desired number of proofs for each i before running out of time.

If we end up with a collision in the oracle answers such that for any query i there are two rewindings that yield the same uniformly random challenge x we abort. However, since we run in polynomial time such collisions happen with negligible probability, so let us analyze the case where we have n_i+1 distinct challenges for each proof. We can now use the (n_i+1)-special soundness property to break the binding property of the commitment scheme or to get an opening $\mathrm{Com}_{ck}(0;r)$ for one of the commitments in the statement, which translates into an opening $\mathrm{Com}_{ck}(S_i;r)$ for one of the commitments in C_i.

Let us now consider the probability of "Error 1" and "Error 2". "Error 1" occurs if $\mathcal{A}$ already queried (ck, M, S, C, a) before the simulation of a spend query but this happens with negligible probability. "Error 2" occurs if the adversary finds a different answer to a challenge than we used in the simulation, since a successful attack on the balance property implies one of the spending proofs $\mathcal{S}_i$ does not match with a pattern in $\mathcal{T}$. By Lemma 1 this happens with negligible probability. We will now proceed under the assumption that such errors did not happen.

Consider the serial numbers of coins in $\mathcal{C}$. As commitments are perfectly hiding and as we revealed freshly sampled random serial numbers in the simulation of Spend an attacker that uses an honest coin to win the game will with high probability also use a different serial number. In case of corrupted coins he is forced by the rules of the security game to always use a different serial number. This yields a break of the binding property of the commitment scheme.

From now on we assume that $\mathcal{A}$ did not use an honest coin. Then there are m different adversary controlled coins and thus commitments but $m+1$ verifying proofs with distinct serial numbers. By extracting from all $m+1$ proofs we are guaranteed that one commitment is opened twice to different serial numbers, which yields a break of the binding property of the commitment scheme. $\square$

Further applications. One-out-of-many proofs are compatible with extended Pedersen commitments, where there is one commitment for a vector of values. They can thus also be employed in the construction of decentralized anonymous credentials [GGM14] and zero-cash protocols [BSCG$^+$14].

References

[AOS04] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. 1-out-of-n signatures from a variety of keys. *IEICE Transactions*, 87-A(1):131–140, 2004.

[BDD07] Stefan Brands, Lisa Demuynck, and Bart De Decker. A practical system for globally revoking the unlinkable pseudonyms of unknown users. In *ACISP*, volume 4586 of *Lecture Notes in Computer Science*, pages 400–415, 2007.

[BG13] Stephanie Bayer and Jens Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 646–663, 2013.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 416–432, 2003.

[BKM09] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology*, 22(1):114–138, 2009.

[Boy07] Xavier Boyen. Mesh signatures. In *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 210–227, 2007.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pages 62–73, 1993.

[BSCG$^+$14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, , and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *IEEE Symposium on Security and Privacy*, 2014.

[Cam97] Jan Camenisch. Efficient and generalized group signatures. In *EUROCRYPT*, volume 1233 of *Lecture Notes in Computer Science*, pages 465–479, 1997.

[CD98] Ronald Cramer and Ivan Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 424–441, 1998.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, volume 893 of *Lecture Notes in Computer Science*, pages 174–187, 1994.

[CGS07] Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434, 2007.

[CL02] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76, 2002.

[Cou01] Nicolas Courtois. Efficient zero-knowledge authentication based on a linear algebra problem minrank. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 402–421, 2001.

[CWLY06] Sherman S. M. Chow, Victor K.-W. Wei, Joseph K. Liu, and Tsz Hon Yuen. Ring signatures without random oracles. In *ASIACCS*, pages 297–302, 2006.

[Dam00] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430, 2000.

[DFKP13] George Danezis, Cédric Fournet, Markulf Kohlweiss, and Bryan Parno. Pinocchio coin: building zerocoin from a succinct pairing-based proof system. In *PETShop at CCS*, 2013.

[DKNS04] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous identification in ad hoc groups. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[DMV13] Özgür Dagdelen, Payman Mohassel, and Daniele Venturi. Rate-limited secure function evaluation: Definitions and constructions. In *PKC*, volume 7778 of *Lecture Notes in Computer Science*, pages 461–478, 2013.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168, 2005.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *Lecture Notes in Computer Science*, pages 60–79, 2012.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, 1986.

[GGI$^+$14] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, pages 1–24, 2014.

[GGM14] Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2013*, 2014.

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In *EUROCRYPT*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, 1988.

[Gro09] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 192–208, 2009.

[HS03] Javier Herranz and Germán Sáez. Forking lemmas for ring signature schemes. In *INDOCRYPT*, volume 2904 of *Lecture Notes in Computer Science*, pages 266–279, 2003.

[IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM Journal on Computing*, 39(3):1121–1152, 2009.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *STOC*, pages 723–732, 1992.

[Lim00] Chae Hoon Lim. Efficient multi-exponentiation and application to batch verification of digital signatures, 2000. Manuscript available at `http://dasan.sejong.ac.kr/~chlim/pub/multi_exp.ps`.

[MGGR13] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE Symposium on Security and Privacy*, 2013.

[Ngu05] Lan Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292, 2005.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140, 1991.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 552–565, 2001.

[San99] Tomas Sander. Efficient accumulators without trapdoor extended abstracts. In *ICICS*, pages 252–262, 1999.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[SW07] Hovav Shacham and Brent Waters. Efficient ring signatures without random oracles. In *PKC*, volume 4450 of *Lecture Notes in Computer Science*, pages 166–180, 2007.