A Traceable Ring Signature Scheme based on Coding Theory

Pedro Branco and Paulo Mateus

SQIG-IT, Department of Mathematics, IST-Universidade de Lisboa

Abstract

Traceable ring signatures are a variant of ring signatures which allows the identity of a user to be revealed, when it signs two different messages with respect to the same group of users. It has applications in e-voting and in cryptocurrencies, such as the well-known Monero. We propose the first traceable ring signature scheme whose security is based on the hardness of the Syndrome Decoding problem, a problem in coding theory which is conjectured to be unsolvable by both classical and quantum algorithms. To construct the scheme, we use a variant of Stern's protocol and, by applying the Fiat-Shamir transform to it in an ingenious way, we obtain a ring signature that allows traceability. We prove that the resulting protocol has the standard security properties for traceable ring signatures in the random oracle model: tag-linkability, anonymity and exculpability. As far as we know, this is the first proposal for a traceable ring signature scheme in the post-quantum setting.

1 Introduction

With the National Institute of Standards and Technology (NIST) decision to standardize quantum-resilient protocols, post-quantum cryptography has become a hot topic in the cryptographic community. However post-quantum signature schemes, particularly signatures based on coding theory, are still underdeveloped. Although most of the operations are relatively efficient and easy to implement (even in hardware), code-based signature schemes consume too much memory for practical purposes. If we consider signature schemes with additional properties, the scenario is even worse since most of these schemes do not even have an equivalent version based on hard problems from coding theory. In this paper, we focus on the latter problem by developing a traceable ring signature scheme whose security is based on the Syndrome Decoding (SD) problem, a problem in coding theory which is believed to be hard for both classical and quantum computers. As far as we know, this is the first code-based traceable ring signature scheme to be proposed and the first one in the post-quantum setting.

Traceable ring signature schemes. Ring signatures [RST01] allow for a user from a group to sign messages on behalf of the group such that a verifier is not able to trace the identity of the actual signer. Although in most cases anonymity is of great importance and should be preserved, in some applications

it may become a problem, in the sense that a dishonest user can take advantage of the anonymity to its own interest. Consider, for example, an election where someone votes once and then tries to create a second vote, claiming to be someone else. From this example we can see that, in some cases, we may want to reveal the identity of abusive users. A trivial solution is to use a group signature scheme [CvH91] (and for which there are code-based versions [ABCG16, ABCG17]), where a group manager has much more power than the rest of the users and can open signatures issued by the users of the group. However, in this case, the group manager would have to open all signatures in order to identify those issued by an abusive user, jeopardizing anonymity of honest users.

Traceable ring signatures [FS07] are ring signatures where the identity of a user may be revealed, in the case it signs two messages with respect to the same group of users and the same issue. In this context, an issue may be an election or a transaction, for example. Traceable ring signature schemes solve the problem presented in the previous paragraph: an abusive user in an election gets caught without compromising the anonymity of the other users. Traceable ring signature schemes have also found a lot of applications in e-cash and cryptocurrencies in the last years. In fact, one of the most famous cryptocurrencies nowadays, Monero [VS13], uses a variant of the scheme by Fujisaki and Suzuki [FS07].

Traceable ring signature schemes are closely related to linkable ring signature schemes [LWW04]. Linkable ring signature schemes also allow a verifier to know if two signatures were issued by the same user in a group of users, but its anonymity is kept preserved no matter the number of signatures issued by this user, unlike traceable ring signature schemes where its identity is revealed.

Previous traceable ring signature schemes were all based on the hardness of the discrete logarithm problem [FS07, Fuj11, ALSY13] which can be solved by Shor's algorithm [Sho97] using a quantum computer. Hence, the advent of a practical quantum computer would turn Monero (with a market value of billions of dollars) and other cryptocurrencies obsolete.

To overcome this problem, we base the security of our traceable ring signature scheme on the syndrome decoding problem. This is a classical problem in coding theory that is conjectured to be hard, even for quantum computers. By basing the security of cryptographic primitives on this problem, we can design new protocols that are conjectured to be robust against quantum adversaries. Therefore, as far as we are aware, the traceable ring signature scheme presented in this work is the first that is conjectured to be suitable for the post-quantum era.

Our contribution and techniques. The major contribution of this paper is the construction of a traceable ring signature scheme based on the SD problem. To develop the new traceable ring signature scheme, we build on top of a recently proposed code-based linkable ring signature scheme [BM18]. More precisely, we consider the GStern's protocol, a variant of the famous Stern's protocol [Ste94], that decides the General Syndrome Decoding (GSD). This protocol allows a prover to prove the knowledge of a error vector $\mathbf{e}$ for two instances of the Syndrome Decoding $(\mathbf{H}, \mathbf{s})$ and $(\mathbf{G}, \mathbf{r})$ for an appropriate choice of parameters. After applying the construction by Cramer, Damgård and Shoemakers [CDS94] for the OR relation, we obtain a proof of knowledge protocol ($\binom{N}{1}$-GStern's

protocol) where the prover proves that it knows a witness for one of several instances of the GSD problem.

Let $(\mathbf{H}, \mathbf{s}_i)$ be the public key of a party $\mathcal{P}_i$ and $\mathbf{e}_i$ its secret key, such that $\mathbf{H}\mathbf{e}_i^T = \mathbf{s}_i^T$ and $\mathbf{e}_i$ has small weight. To sign a message using the scheme, a user collects the public keys of the elements in the ring. Let $(\mathbf{H}, \mathbf{s}_1, \ldots, \mathbf{s}_N)$ (the matrix $\mathbf{H}$ is common to every party's public key) be the public keys of the users in the ring. The signer computes $\widetilde{\mathbf{H}}\mathbf{e}_i^T = \mathbf{r}_i^T$, where $\widetilde{\mathbf{H}}$ is a matrix computed using a random oracle and that depends on the ring of users. It creates random vectors $\mathbf{r}_1, \ldots, \mathbf{r}_{i-1}, \mathbf{r}_{i+i}, \ldots, \mathbf{r}_N$ for each user of the ring. Since these vectors must be random, the user computes them using a hash function and depending on the message. Now, the user creates a signature by applying the Fiat-Shamir [FS87] to the $\binom{N}{1}$-GStern's protocol on input $(\mathbf{H}, \mathbf{s}_1, \ldots, \mathbf{s}_N, \widetilde{\mathbf{H}}, \mathbf{r}_1, \ldots, \mathbf{r}_N)$. Suppose that some user $\mathcal{P}_i$ signs creates two signatures for two different messages. Traceability will be possible by checking for which i, $\mathbf{r}_i = \mathbf{r}'_i$ where $\mathbf{r}_i$ is part of one signature and $\mathbf{r}'_i$ is part of the other.

We prove the usual security properties for traceable ring signature schemes in the Random Oracle Model: tag-linkability, anonymity and exculpability.

2 Notation and preliminaries

We begin by presenting some notation. We will use bold lower cases to denote vectors (like $\mathbf{x}$) and bold capital letters to denote matrices (like $\mathbf{H}$). We denote the usual Hamming weight of a vector $\mathbf{x}$ by $w(\mathbf{x})$. If $\mathcal{A}$ is an algorithm, we denote $y \leftarrow \mathcal{A}(x)$ the output y when running $\mathcal{A}$ with input x. If S is a finite set, $|S|$ denotes its cardinality and $y \leftarrow\!\!\$\, S$ means that y was chosen uniformly at random from S. By $\mathsf{negl}(n)$ we denote a function F that is negligible on the parameter n, i.e., $F < 1/\mathsf{poly}(n)$ where $\mathsf{poly}(n)$ represents any polynomial in n. The acronym PPT means probabilistic polynomial-time.

Due to the lack of space, we refer the reader to Appendix A for a brief introduction on sigma protocols[1], the Fiat-Shamir transform [FS87], the Cramer-Damgård-Shoenmakers (CDS) construction for the OR relation [CDS94] and the original Stern's protocol [Ste94].

2.1 Hard problems in coding theory

We present the search version of the Syndrome Decoding (SD) problem, a hard problem in coding theory, proven to be NP-complete [BMvT78] in the worst-case. The problem states that it is hard to decode a random linear code. Recall that a k-dimensional code $\mathcal{C}$ of length n can be represented by its parity-check matrix $\mathbf{H} \in \mathbb{Z}_2^{(n-k)\times n}$.

Problem 1 (Syndrome Decoding)**.** *Given* $\mathbf{H} \in \mathbb{Z}_2^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{Z}_2^{n-k}$ *and* $t \in \mathbb{N}$, *find* $\mathbf{e} \in \mathbb{Z}_2^n$ *such that* $w(\mathbf{e}) \leq t$ *and* $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$.

The problem is also widely believed to be hard on the average-case since the best known generic decoding classical and quantum attacks still take exponential time [CC98, Ber10, MMT11, BJMM12, CTS16] and, when $\mathbf{e}$ is chosen uniformly at random from the set of vectors with weight t and the matrix $\mathbf{H}$ is chosen

[1] We refer the reader to [Dam02] for a more detailed introduction on sigma protocols.

uniformly at random from $\mathbb{Z}_2^{(n-k)\times n}$, the statistical distance between $(\mathbf{H}, \mathbf{H}\mathbf{e}^T)$ and the uniform distribution over $\mathbb{Z}_2^{(n-k)\times n} \times \mathbb{Z}_2^{n-k}$ is negligible [ELL+15].

Next, we present a lemma which will be useful to prove the completeness of the proposed protocols. It states that the equation $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ will most likely have a solution (not necessarily with $w(\mathbf{x}) \leq t$) with $\mathbf{H}$ and $\mathbf{s}$ chosen at random.

Lemma 2. *Let $n, k' \in \mathbb{N}$ such that $k' \leq n/2$. Given $\mathbf{H} \leftarrow_{\$} \mathbb{Z}_2^{k' \times n}$ and $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_2^{k'}$, the probability of existing a vector $\mathbf{x} \in \mathbb{Z}_2^n$ such that $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ is, at least, $1 - \mathsf{negl}(n)$.*

The proof is presented in Appendix B.1

Corollary 3. *Let $n, k' \in \mathbb{N}$ such that $k' \leq n/4$. Given $\mathbf{H}, \mathbf{G} \leftarrow_{\$} \mathbb{Z}_2^{k' \times n}$ and $\mathbf{s}, \mathbf{r} \leftarrow_{\$} \mathbb{Z}_2^{k'}$, the probability that there is a vector $\mathbf{x} \in \mathbb{Z}_2^n$ such that $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ and $\mathbf{G}\mathbf{x}^T = \mathbf{r}^T$ is $1 - \mathsf{negl}(n)$.*

The Corollary can be easily proved by observing that

$$\begin{pmatrix} \mathbf{H} \\ \mathbf{G} \end{pmatrix} \mathbf{x}^T = \begin{pmatrix} \mathbf{s}^T \\ \mathbf{r}^T \end{pmatrix}$$

is a special case of the previous lemma.

For our purpose, we want the equation $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ to have solutions, where $\mathbf{H} \in \mathbb{Z}_2^{(n-k)\times n}$. Hence, we just need to consider $n - k = k' \leq n/4$, that is, $k \geq 3n/4$. To this end, we take $k = 3n/4$.

We now present the Generalized Syndrome Decoding (GSD) problem.

Problem 4. *Given $\mathbf{H}, \mathbf{G} \in \mathbb{Z}_2^{(n-k)\times n}$, $\mathbf{s}, \mathbf{r} \in \mathbb{Z}_2^{n-k}$ and $t \in \mathbb{N}$, find $\mathbf{e} \in \mathbb{Z}_2^n$ such that $w(\mathbf{e}) \leq t$, $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$ and $\mathbf{G}\mathbf{e}^T = \mathbf{r}^T$.*

Note that the SD problem can be trivially reduced to GSD, by choosing as inputs of the reduction $\mathbf{H} = \mathbf{G}$ and $\mathbf{s} = \mathbf{r}$, and so GSD is a NP-complete language.

The next protocol is a proof of knowledge protocol for the GSD problem. We will call GStern's protocol to the protocol presented in Algorithm 1.[2]

In the protocol, presented in Algorithm 1, observe that, when $b = 1$, $\mathcal{V}$ can check that c_1 was honestly computed by verifying whether $\mathbf{H}(\mathbf{y} + \mathbf{e})^T + \mathbf{s}^T = \mathbf{H}\mathbf{y}^T$ and $\mathbf{G}(\mathbf{y} + \mathbf{e})^T + \mathbf{r}^T = \mathbf{G}\mathbf{y}^T$. Also, the verifier can check that it is the same error vector $\mathbf{e}$ that was used to compute the syndrome vectors $\mathbf{s}$ and $\mathbf{r}$.

The protocol is proven to be complete, special sound and honest-verifier zero-knowledge (HVZK) [BM18]. Nevertheless, we sketch the proof here. It is easy to see that, from two valid transcripts $(com, ch, resp)$ and $(com, ch', resp')$ of GStern's protocol, with $ch \neq ch'$, there is a simulator that can extract a valid witness. For instance, when $ch = 0$ and $ch' = 1$, the simulator can extract the secret $\mathbf{e}$ from $\mathbf{y}$ and $\mathbf{y} + \mathbf{e}$. In a similar way, it can always extract $\mathbf{e}$ in the other two cases. To prove HVZK, note that: i) when $b = 0$, the simulator just has to reveal a random vector $\mathbf{y}$ and a random permutation δ; ii) when $b = 1$, the simulator has to reveal a vector $\mathbf{x}$ such that $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ (but not necessarily with $w(\mathbf{x}) = t$). Note that this is possible due to Corollary 3; finally, iii) when $b = 2$, the simulator just has to reveal a vector with weight t.

[2] The name GStern's protocol comes from Generalized Stern's protocol.

Algorithm 1 GStern's protocol

1. **Public information:** $\mathbf{H}, \mathbf{G} \in \mathbb{Z}_2^{(n-k)\times n}$ and $\mathbf{s}, \mathbf{r} \in \mathbb{Z}_2^{n-k}$.
2. **Secret information:** $\mathbf{e} \in \mathbb{Z}_2^n$ such that $\mathbf{He}^T = \mathbf{s}^T$, $\mathbf{Ge}^T = \mathbf{r}^T$ and $w(\mathbf{e}) = t$.
3. **Prover's commitment:**
   - $\mathcal{P}$ chooses $\mathbf{y} \leftarrow_{\$} \mathbb{Z}_2^n$ and a permutation δ;
   - $\mathcal{P}$ computes $c_1 = h(\delta, \mathbf{Hy}^T, \mathbf{Gy}^T)$, $c_2 = h(\delta(\mathbf{y}))$ and $c_3 = h(\delta(\mathbf{y} + \mathbf{e}))$;
   - $\mathcal{P}$ sends c_1, c_2 and c_3.
4. **Verifier's Challenge:** $\mathcal{V}$ sends $b \leftarrow_{\$} \{0, 1, 2\}$.
5. **Prover's answer:**
   - If $b = 0$, $\mathcal{P}$ reveals $\mathbf{y}$ and δ;
   - If $b = 1$, $\mathcal{P}$ reveals $\mathbf{y} + \mathbf{e}$ and δ;
   - If $b = 2$, $\mathcal{P}$ reveals $\delta(\mathbf{y})$ and $\delta(\mathbf{e})$.
6. **Verifier's verification:**
   - If $b = 0$, $\mathcal{V}$ checks if $h(\delta, \mathbf{Hy}^T, \mathbf{Gy}^T) = c_1$ and $h(\delta(\mathbf{y})) = c_2$;
   - If $b = 1$, $\mathcal{V}$ checks if $h(\delta, \mathbf{H}(\mathbf{y} + \mathbf{e})^T + \mathbf{s}^T, \mathbf{G}(\mathbf{y} + \mathbf{e})^T + \mathbf{r}^T) = c_1$ and $h(\delta(\mathbf{y} + \mathbf{e})) = c_3$;
   - If $b = 2$, $\mathcal{V}$ checks if $h(\delta(\mathbf{y})) = c_2$, $h(\delta(\mathbf{y}) + \delta(\mathbf{e})) = c_3$ and $w(\delta(\mathbf{e})) = t$.

To build our signature scheme, we apply the CDS construction [CDS94] to GStern's protocol. We will call the resulting protocol $\binom{N}{1}$-GStern's protocol. We assume that the matrices $\mathbf{H}$ and $\mathbf{G}$, and t are the same for every instance of the GSD problem. In the following, *com*, *ch* and *resp* are commitments, challenges and responses, respectively, of GStern's protocol repeated $\mathcal{O}(1/\epsilon)$ times. Moreover, the challenges are expressed as bit strings. The protocol is presented in Algorithm 2.

The $\binom{N}{1}$-GStern's protocol is a PoK that is complete, special sound and HVZK. This fact is a direct consequence of the results in [CDS94]. We briefly give the sketch of the proof.

Suppose that the prover has a secret for instance j. To prove completeness, note that a honest prover can always create valid transcript for instance j. This follows from the completeness of GStern's protocol. It can also create valid transcripts for the other instances from the HVZK of GStern's protocol. Thus, a prover holding a secret for instance j can always create valid transcripts for $\binom{N}{1}$-GStern's protocol.

As usual, to prove special soundness of $\binom{N}{1}$-GStern's protocol, the simulator runs the prover and gets two valid transcripts $(Com, Ch, Resp)$ and $(Com, Ch', Resp')$, where $Com = \{com_i\}_i$, $Ch = \{ch_i\}_i$, $Ch' = \{ch'_i\}_i$, $Resp = \{resp_i\}_i$, $Resp' = \{resp'_i\}_i$, $\sum_i ch_i = b$ and $\sum_i ch'_i = b'$. Suppose that the prover has the secret for the instance j. Then $ch_i = ch'_i$ and $resp_i = resp'_i$ for every $i \neq j$. Also, $ch_j \neq ch'_j$ and $resp_j \neq resp'_j$, except with negligible probability. Thus, by the special soundness of the GStern's protocol, the simulator can extract a valid witness for instance j from these transcripts.

To prove HVZK, we have to show that there is a simulator capable of creating

Algorithm 2 $\binom{N}{1}$-GStern's protocol

1. Public information: N instances of the GSD problem $(\mathbf{H}, \mathbf{s}_1, \ldots, \mathbf{s}_N, \mathbf{G}, \mathbf{r}_1, \ldots, \mathbf{r}_N, t)$

2. Secret information: $\mathbf{e} \in \{0,1\}^n$ such that $w(\mathbf{e}) = t$, $\mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T$ and $\mathbf{G}\mathbf{e}^T = \mathbf{r}_i^T$ for some $i \in \{1, \ldots, N\}$.

3. Prover's commitment:

- $\mathcal{P}^*$ simulates transcripts $(com_j, ch_j, resp_j)$ using the simulator $\mathcal{S}$ for $j \neq i$ according to GStern's protocol;
- $\mathcal{P}^*$ computes com_i according to GStern's protocol;
- $\mathcal{P}^*$ sends $com_1, \ldots, com_N$.

4. Verifier's challenge: $\mathcal{V}$ sends $b \leftarrow_{\$} C$.

5. Prover's answer:

- $\mathcal{P}$ computes $ch_i = b + \sum_{j \neq i} ch_j$;
- $\mathcal{P}$ computes $resp_i$ according to com_i and ch_i;
- Sends $(com_j, ch_j, resp_j)$ for every j.

6. Verifier's verification:

- $\mathcal{V}$ checks that $(com_j, ch_j, resp_j)$ is valid according to GStern's protocol, for every j;
- $\mathcal{V}$ checks that $b = \sum_j ch_j$;
- $\mathcal{V}$ accepts if it passes all the verification tests.

valid transcripts for $\binom{N}{1}$-GStern's protocol, even when not holding a witness for any of the instances. But observe that, by the HVZK property of the GStern's protocol, the simulator can create valid transcripts for each of the instances. Hence, a valid transcript for $\binom{N}{1}$-GStern's protocol follows from these transcripts of GStern's protocol.

Therefore, we can use the Fiat-Shamir transform to create a secure signature scheme [AABN02].

2.2 Traceable ring signature schemes

We present the definition of traceable ring signature scheme along with the security model we consider, originally presented in [FS07]. In the following, let $\overline{\mathbf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$, *issue* be a string denoting the goal of the signature (for example, an election or a transaction) and $L = (issue, \overline{\mathbf{pk}})$. We will call L the tag of the signature.

Definition 5. A *traceable ring signature scheme* is defined by a tuple of algorithms $(KeyGen, Sign, Ver, Trace)$ where:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow KeyGen(1^\kappa)$ is a PPT algorithm that takes as input a security parameter κ and outputs a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$;
- $\sigma \leftarrow Sign(\mathsf{sk}_i, L, M)$ is a PPT algorithm that takes as input a secret key sk_i, a tag $L = (issue, \overline{\mathbf{pk}})$ and a message to be signed M and outputs a signature σ.

- $b \leftarrow \mathit{Ver}(L, M, \sigma)$ is a deterministic algorithm that takes as input a tag $L = (\mathit{issue}, \overline{\mathbf{pk}})$, a signature σ and a message M and outputs a bit b such that $b = 1$ if the signature is valid and $b = 0$ otherwise.
- $s \leftarrow \mathit{Trace}(L, M_1, \sigma_1, M_2, \sigma_2)$ is a deterministic algorithm that takes as input a tag $L = (\mathit{issue}, \overline{\mathbf{pk}})$ and two pairs of messages and corresponding signatures (M_1, σ_1) and (M_2, σ_2) and outputs a string s that is either equal to *indep*, *linked* or to an element $\mathsf{pk} \in \overline{\mathbf{pk}}$ such that, if $\sigma_1 \leftarrow \mathit{Sign}(\mathsf{sk}_i, L, M_1)$ and $\sigma_2 \leftarrow \mathit{Sign}(\mathsf{sk}_j, L, M_2)$, then

$$\mathit{Trace}(L, M_1, \sigma_1, M_2, \sigma_2) \coloneqq \begin{cases} \mathit{indep} \text{ if } i \neq j, \\ \mathit{linked} \text{ else if } M_1 = M_2, \\ \mathsf{pk}_i \text{ otherwise.} \end{cases}$$

The security requirements for a traceable ring signature scheme are three: tag-linkability, anonymity and exculpability. Unforgeability comes from tag-linkability and exculpability. In the following, let κ be a security parameter, N be the number of users in the ring, $L = (\mathit{issue}, \overline{\mathbf{pk}})$ where $\overline{\mathbf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$ are the public keys of each user and $\mathit{Sign}(\mathsf{sk}, \cdot)$ is a signing oracle that receives queries of the form (L, M) and outputs $\sigma \leftarrow \mathit{Sign}(\mathsf{sk}, L, M)$.

Tag-linkability. Informally, it must be infeasible for an adversary to create $N + 1$ signatures having access to N pairs of public and secret keys. Let $\mathcal{A}$ be a PPT adversary. Consider the following game:

$\mathtt{Game}_{\mathcal{A}}^{tagLink}(\kappa, N)$:

1 : $(L, (M_1, \sigma_1), \ldots, (M_{n+1}, \sigma_{n+1})) \leftarrow \mathcal{A}(1^\kappa)$
2 : $b_i \leftarrow \mathit{Ver}(L, M_i, \sigma_i) \quad \forall i \in \{1, \ldots, N+1\}$
3 : $s_{i,j} \leftarrow \mathit{Trace}(L, M_i, \sigma_i, M_j, \sigma_j) \quad \forall i, j \in \{1, \ldots, N+1\} \wedge i \neq j$
4 : **return** $b_1, \ldots, b_{N+1}, s_{1,1}, s_{1,2}, \ldots, s_{N+1,N+1}$

where $L = (\mathit{issue}, \mathbf{pk})$ and $\mathbf{pk} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$.

We define

$$\mathtt{Adv}_{\mathcal{A}}^{tagLink}(\kappa, N) \coloneqq \mathtt{Pr}\left[\bigwedge_{i=1}^{N+1} b_i = 1 \wedge \bigwedge_{\substack{i,j=1 \\ i \neq j}}^{N+1} s_{i,j} = \mathit{indep}\right].$$

If, for all PPT adversaries $\mathcal{A}$ we have that $\mathtt{Adv}_{\mathcal{A}}^{tagLink}(\kappa, N) \leq \mathsf{negl}(\kappa, N)$ then we say that the traceable ring signature scheme is tag-linkable.

Anonymity. Informally, it must be infeasible for an adversary to know who signed the message. Let $\mathcal{A}$ be a PPT adversary. Consider the following game:

$\mathtt{Game}_{\mathcal{A}}^{anon}(\kappa, N)$:

1 : $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathit{KeyGen}(1^\kappa), \quad i = 0, 1$
2 : $b \leftarrow_{\$} \{0, 1\}$
3 : $b' \leftarrow \mathcal{A}^{\mathit{Sign}(\mathsf{sk}_b, \cdot), \mathit{Sign}(\mathsf{sk}_0, \cdot), \mathit{Sign}(\mathsf{sk}_1, \cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
4 : **return** b'

where the adversary is not allowed to ask queries with different tags to $Sign(\mathsf{sk}_b, \cdot)$ nor to ask queries with the same tag to both $Sign(\mathsf{sk}_b, \cdot)$ and $Sign(\mathsf{sk}_0, \cdot)$ or to both $Sign(\mathsf{sk}_b, \cdot)$ and $Sign(\mathsf{sk}_1, \cdot)$. We do not allow this to happen to avoid the trivial attacks.

We define

$$\mathtt{Adv}^{anon}_{\mathcal{A}}(\kappa, N) \coloneqq \mathtt{Pr}\left[b = b'\right] - \frac{1}{2}.$$

If for all PPT adversaries $\mathcal{A}$ we have that $\mathtt{Adv}^{anon}_{\mathcal{A}}(\kappa, N) \leq \mathsf{negl}(\kappa, N)$ then we say that the traceable ring signature scheme is anonymous.

Exculpability. Informally, it must be infeasible for an adversary $\mathcal{A}$ to produce two pairs of messages and respective signatures that seem to be issued by some user i, without knowledge of the secret key. In this case, we say that $\mathcal{A}$ frames user i. Let $\mathcal{A}$ be a PPT adversary. Consider the following game:

$\mathtt{Game}^{exc}_{\mathcal{A}}(\kappa, N)$:

1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow KeyGen(1^{\kappa})$

2 : $(L, M_1, \sigma_1), (L, M_2, \sigma_2) \leftarrow \mathcal{A}^{Sign(\mathsf{sk}, \cdot)}(\mathsf{pk})$

3 : $s \leftarrow Trace(L, M_1, \sigma_1, M_2, \sigma_2)$

4 : **return** s

where $Ver(L, M_1, \sigma_1) = 1$, $Ver(L, M_2, \sigma_2) = 1$, $\mathsf{pk} \in \overline{\mathbf{pk}}$ and at least one of the signatures must not be linked[3] to any query to $Sign(\mathsf{sk}, \cdot)$ made by $\mathcal{A}$ (to avoid the trivial attacks).

We define

$$\mathtt{Adv}^{exc}_{\mathcal{A}}(\kappa, N) \coloneqq \mathtt{Pr}\left[s = \mathsf{pk}\right].$$

If for all PPT adversaries $\mathcal{A}$ we have that $\mathtt{Adv}^{exc}_{\mathcal{A}}(\kappa, N) \leq \mathsf{negl}(\kappa, N)$ then we say that the traceable ring signature scheme is exculpable.

Unforgeability comes directly from the properties of tag-linkability and exculpability, as the next theorem states.

Theorem 6 ([FS07])**.** *Assume that a traceable ring signature scheme is tag-linkable and exculpable, then it is unforgeable.*

3 A code-based traceable ring signature scheme

In this section we propose a new traceable ring signature scheme based on the SD problem.

The scheme is presented in Algorithm 3. In a nutshell, the traceable ring signature scheme is obtained by applying the Fiat-Shamir transform to the $\binom{N}{1}$-GStern's protocol. To achieve traceability, we construct a set of random syndromes $\mathbf{r}_1 \ldots, \mathbf{r}_N$ of a random matrix $\widetilde{\mathbf{H}}$ (generated via a cryptographic hash function g and depending on the tag L), where one of the $\mathbf{r}_i$ is the syndrome of the secret vector known by the actual signer. When signing two different

[3]That is, at least one of the messages (M_1 or M_2) was not asked in a query to the oracle $Sign(\mathsf{sk}, \cdot)$.

messages with respect to the same tag, this syndrome will be the same in both signatures and, thus, we can identify the signer of the message. To prevent the signer from cheating when signing, we force it to generate the other syndromes with another cryptographic hash function f in such a way that the verifier will be able to check that these syndromes were honestly and randomly generated.

The new traceable ring signature scheme is presented in Algorithm 3. In the following, let $\overline{\mathbf{pk}} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_N)$ be the set of public keys of the users $\mathcal{P}_1, \ldots, \mathcal{P}_N$ in the ring and $L = (issue, \overline{\mathbf{pk}})$ be a tag. Let $\bar{\mathbf{s}} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)$, $\bar{\mathbf{r}} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$ and $\mathbf{H}$ be a parity-check matrix of a random code.

Let f, $\bar{f}$, g and h be four different cryptographic hash functions (modeled as random oracles). The function h is the one used in the $\binom{N}{1}$-GStern's protocol, $\bar{f}$ is the one used in the Fiat-Shamir transform, $g : \mathbb{Z}_2^* \to \mathbb{Z}_2^{(n-k)\times n}$ is used to compute a matrix from the issue L and $f : \mathbb{Z}_2^* \to \mathbb{Z}_2^{n-k}$ is used to compute random syndromes to allow traceability (as mentioned before). By $f^i(x)$ we denote the function f applied i times on input x.

Note that, by Corollary 3, the probability that the prover cannot simulate transcripts for the keys that it does not know is negligible since it can easily find a solution $\mathbf{x} \in \mathbb{Z}_2^n$ for an equation of the type $\mathbf{H}\mathbf{x}^T = \mathbf{s}^T$ where $\mathbf{H} \in \mathbb{Z}_2^{(n-k)\times n}$ and $\mathbf{s} \in \mathbb{Z}_2^{n-k}$ (when $k = 3n/4$). Thus, Corollary 3 guarantees the correctness of the protocol.

4 Security analysis

In this section we give the security proofs for the proposed traceable ring signature scheme. Recall that unforgeability for the scheme follows from the tag-linkability and exculpability properties. We begin by proving tag-linkability for our scheme, but first we present two lemmas. Detailed proofs are in the full version of this paper.

Lemma 7. *Given a valid signature (L, M, σ), the probability that*

$$\left|\{i \in \mathbb{N} : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) = t \wedge \mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T \wedge \widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T\}\right| = 1$$

is $1 - \mathsf{negl}(n)$.

Lemma 8. *Given two valid signatures (L, M, σ) and (L, M', σ') such that they are independent (that is, $Trace(L, M, \sigma, M', \sigma') = indep$) the probability that $|traceList| > 1$ is $\mathsf{negl}(n)$.*

Theorem 9 (Tag-linkability)**.** *The traceable ring signature scheme proposed is tag-linkable in the ROM.*

Before proving anonymity, note that, given an instance of the SD problem where we know the position of $t/2$ non-null coordinates of the error vector, this is still an instance of the SD problem, for an appropriate choice of parameters. So, it is still a hard problem to find the rest of the $t/2$ non-null positions of the error vector. We briefly sketch the reduction here: suppose that we have an algorithm $\mathcal{A}$ that solves the SD problem knowing $t/2$ positions of the error vector. The algorithm that breaks the SD problem receives as input $(\mathbf{H}, \mathbf{s}^T = \mathbf{H}\mathbf{e}^T, t/2)$. Then it computes a new matrix $\mathbf{H}' = (\mathbf{H}|\mathbf{R})$ where $\mathbf{R} \leftarrow_{\$} \mathbb{Z}_2^{(n-k)\times t/2}$ and it

Algorithm 3 A new traceable ring signature scheme

1. **Parameters:** $n, k, t \in \mathbb{N}$ such that $k = 3n/4$, $\mathbf{H} \leftarrow_{\$} \{0,1\}^{(n-k)\times n}$
2. **Key Generation:** Each user $\mathcal{P}_i$:
   - Chooses $\mathbf{e}_i \leftarrow_{\$} \{0,1\}^n$ such that $w(\mathbf{e}_i) = t$
   - Computes $\mathbf{s}_i^T = \mathbf{H}\mathbf{e}_i^T$

   Public key of user $\mathcal{P}_i$***:*** $\mathbf{H}, \mathbf{s}_i$
   Secret key of user $\mathcal{P}_i$***:*** $\mathbf{e}_i$ such that $w(\mathbf{e}_i) = t$ and $\mathbf{H}\mathbf{e}_i^T = \mathbf{s}_i^T$
3. **Sign:** To sign message M, user $\mathcal{P}_i$:
   - Computes matrix $g(L) = \widetilde{\mathbf{H}}$ and $\widetilde{\mathbf{H}}\mathbf{e}_i^T = \mathbf{r}_i^T$;
   - Sets $A_0 = \mathbf{r}_i + f(M) + \cdots + f^i(M)$;
   - Compute $\mathbf{r}_j = A_0 + f(M) + f^2(M) + \cdots + f^j(M)$, for $j \neq i$;
   - Applies the Fiat-Shamir transform to $\binom{N}{1}$-GStern's protocol on input $(\mathbf{H}, \overline{\mathbf{s}}, \widetilde{\mathbf{H}}, \overline{\mathbf{r}})$ where $\overline{\mathbf{s}} = (\mathbf{s}_1, \ldots, \mathbf{s}_N)$ and $\overline{\mathbf{r}} = (\mathbf{r}_1, \ldots, \mathbf{r}_N)$:
     - Computes the commitments *Com* according to $\binom{N}{1}$-GStern's protocol;
     - Simulates the verifier's challenge as $Ch = \bar{f}(Com, M)$;
     - Computes the corresponding responses *Resp* according to $\binom{N}{1}$-GStern's protocol;
     - Outputs the transcript $T = (Com, Ch, Resp)$.
   - Outputs the signature (L, M, σ) where $\sigma = (A_0, Com, Resp)$.
4. **Verify:** To verify, the verifier:
   - Computes $\mathbf{r}_j = A_0 + f(M) + f^2(M) + \cdots + f^j(M)$ for all $j \in \{1, \ldots, N\}$;
   - Computes $Ch = \bar{f}(Com, M)$;
   - Verifies that $T = (Com, Ch, Resp)$ is a valid transcript, according to $\binom{N}{1}$-GStern's protocol.

5. Trace: Given two signatures (L, M, σ) and (L, M', σ') where $\sigma = (A_0, Com, Resp)$ and $\sigma' = (A_0', Com', Resp')$ such that $Ver(L, M, \sigma) = 1$ and $Ver(L, M', \sigma') = 1$, the verifier:

- Computes $\mathbf{r}_j = A_0 + f(M) + f^2(M) + \cdots + f^j(M)$ and $\mathbf{r}_j' = A_0' + f(M') + f^2(M') + \cdots + f^j(M')$ for all j;
- Checks if $\mathbf{r}_j = \mathbf{r}_j'$. If this happens, it stores pk_j in a list *traceList*, which is initially empty, for all j;
- Outputs the only $\mathsf{pk}_i \in traceList$ if $|traceList| = 1$; else if $traceList = \overline{\mathbf{pk}} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$ it outputs *linked*; else it outputs *indep*.

computes the vector $\mathbf{s}' = \mathbf{s} + \mathbf{R}(1, \dots 1)^T$ where $(1, \dots, 1)$ has size $t/2$. Now we call the algorithm $\mathcal{A}$ on input $\mathbf{H}', \mathbf{s}', t/2$ and the last positions of the error vector. The reduction is obviously not tight. We take in account this fact when proposing parameters for the scheme.

We now turn our attention to the anonymity of the scheme. In order to prove anonymity for the proposed traceable ring signature scheme, we reduce a variant of the decision version of the GSD problem to the problem of breaking the anonymity of the scheme. This variant is the GSD problem when $t/2$ positions of the error vector are known. Note that this does not threat the security since, even when knowing half of the positions of the error vector, the GSD problem is still computationally hard.

We need to know $t/2$ positions of the error vector because of following technical reason: we know how the algorithm that breaks the anonymity behaves when it is given two valid public keys or when it is given two random values as public keys. However, we do not know how it behaves when it is given one valid public key and one random value as public key. More precisely, given a tuple $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t)$, we do not know if this represents a valid public key of the signature scheme or if it is a random tuple. However, if we know part of the secret, we are able to construct another tuple $(\mathbf{H}, \mathbf{s}', \mathbf{G}, \mathbf{r}', t)$ that is a GSD tuple, if $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t)$ is a GSD tuple, or that is a random tuple, otherwise. We elaborate more on this in the proof of anonymity in Appendix C.4.

Theorem 10 (Anonymity)**.** *The traceable ring signature scheme proposed is anonymous in the ROM, given that the language*

$$GSD = \left\{(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) \leq t \wedge \mathbf{H}\mathbf{e}^T = \mathbf{s}^T \wedge \mathbf{G}\mathbf{e}^T = \mathbf{r}^T\right\}$$

is hard to decide knowing $t/2$ positions of the error.

Finally, we prove that our scheme is exculpable.

Theorem 11 (Exculpability)**.** *The traceable ring signature scheme proposed is exculpable in the ROM and given that the GSD problem is hard.*

5 Parameters and key size

To conclude, we propose parameters for the scheme and analyze its signature and key size. For the cheating probability of GStern's protocol to be approximately 2^{-128}, it has to be iterated 220 times. Recall that anonymity for our traceable ring signature scheme is proven when knowing $t/2$ positions of the error vector. Hence, to yield the standard security of approximately 128 bits for signature schemes according to the generic decoding attack in [BJMM12], we consider a code with $n = 4150$, $k = 3n/4$ and $t = 132$ (similar to [CTS16]). Note that a code with these parameters has a security of approximately 128 bits even when knowing $t/2$ positions of the error vector. This is necessary to maintain the anonymity of the scheme. Let N be the number of users in the ring.

Size of the sigma protocol. The $\binom{N}{1}$-GStern's protocol has approximately $8700N$ bits of exchange information in each round.

Signature size. The signature size is approximately $240N$ kBytes. For example, for a ring with $N = 100$ users, the signature size is approximately 24 MBytes.

Public key size. The public key is approximately $12918950+1037$ bits, which is linear in the number of users in the ring. For example, for a ring with $N = 100$ users, the public key has size approximately 1.6 MBytes.

6 Conclusion

Traceable ring signature schemes have a wide range of applications. Currently they are used in the implementation of Monero, one of the most famous cryptocurrencies, but they also have other applications, such as, in e-voting. However, the constructions for traceable ring signatures that exist in the literature are all based on the discrete logarithm problem and, thus, they can be broken using Shor's algorithm.

We proposed the first traceable ring signature whose security does not rely on the discrete logarithm problem, but rather on the SD problem, a problem that is conjectured to be unsolvable in polynomial time by any classical or quantum computer. Our construction is conjectured to be robust to quantum attacks. We proved the usual security properties for traceable ring signature schemes in the ROM.

However, the key and signature size of the protocol are too large for some applications. This is a common problem to all code-based cryptosystems. Finding new techniques to reduce the key and the signature size of code-based signature schemes is an obvious direction for future work.

We also leave as an open question to prove the security of the protocol in the Quantum Random Oracle Model (QROM) [BDF+11], where the parties can query random oracles in superposition. Note that our proofs do not apply to the quantum setup. For example, observe that the proof of exculpability uses a rewind technique and the problem of quantum rewind is more subtle than in the classical setup [ARU14]. Also, Unruh [Unr17] proved that the Fiat-Shamir transform can be applied to obtain secure signature schemes in the QROM, under certain conditions. However these results are not known to hold for the case of ring signatures constructed using the Fiat-Shamir transform.

Acknowledgments

The first author would like to thank the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017.

This work is funded by FCT/MEC through national funds and when applicable co-funded by FEDER – PT2020 partnership agreement under the project UID/EEA/50008/2013, and IT internal project QBigData, FCT through national funds, by FEDER, through COMPETE 2020, and by Regional Operational Program of Lisbon, under projects Confident PTDC/EEI-CTP/4503/2014, QuantumMining POCI-01-0145-FEDER-031826 and Predict PTDC/CCI-CIF/ 29877/2017. It was funded by European project H2020-SU-ICT-2018-2020.

References

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 418–433, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[ABCG16] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A practical group signature scheme based on rank metric. In Sylvain Duquesne and Svetla Petkova-Nikova, editors, *Arithmetic of Finite Fields*, pages 258–275, Cham, 2016. Springer International Publishing.

[ABCG17] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A code-based group signature scheme. *Designs, Codes and Cryptography*, 82(1):469–493, Jan 2017.

[ALSY13] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. Secure ID-based linkable and revocable-iff-linked ring signature with constant-size construction. *Theoretical Computer Science*, 469:1 – 14, 2013.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 474–483, Washington, DC, USA, 2014. IEEE Computer Society.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Ber10] Daniel J. Bernstein. Grover vs. mceliece. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, pages 73–80, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 520–536, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[BM18] Pedro Branco and Paulo Mateus. A code-based linkable ring signature scheme. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security*, pages 203–219, Cham, 2018. Springer International Publishing.

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[CC98] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, Jan 1998.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 174–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[CTS16] Rodolfo Canto Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography*, pages 144–161, Cham, 2016. Springer International Publishing.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In Donald W. Davies, editor, *Advances in Cryptology — EUROCRYPT '91*, pages 257–265, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[Dam02] Ivan Damgård. On σ-protocols. *Lecture Notes, University of Aarhus, Department for Computer Science*, 2002.

[ELL+15] Martianus Frederic Ezerman, Hyung Tae Lee, San Ling, Khoa Nguyen, and Huaxiong Wang. A provably secure group signature scheme from code-based assumptions. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, pages 260–285, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[FS07] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 181–200, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[Fuj11] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, pages 393–415, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[LWW04] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan,

editors, *Information Security and Privacy*, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 107–124, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, Jun 2000.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[Ste94] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 13–21, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[Unr17] Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 65–95, Cham, 2017. Springer International Publishing.

[VS13] Nicolas Van Saberhagen. CryptoNote v 2.0, 2013.

Appendix A Sigma protocols

A.1 Fiat-Shamir transform

A sigma protocol $(\mathcal{P}, \mathcal{V})$ is a three-round protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ where the prover tries to convince the verifier about the validity of some statement. In this work, we are only interested in a particular case of sigma protocols which are proof of knowledge (PoK) protocols. Here, the prover $\mathcal{P}$ convinces the verifier $\mathcal{V}$, not only about the veracity of the statement, but also that $\mathcal{P}$ has a witness for it. The three rounds of any sigma protocol are the commitment (com) by the prover, the challenge (ch) by the verifier and the response ($resp$) by the prover. A transcript ($com, ch, resp$) is said to be valid if the verifier accepts it as a valid proof.

A PoK must have the following properties: i) completeness, which ensures that the verifier will accept the proof with high probability if the prover has the secret; ii) special soundness, which ensures that there is an extractor such that, given two valid transcripts ($com, ch, resp$) and ($com, ch', resp'$) where $ch \neq ch'$, then it can extract the secret; and iii) honest-verifier zero-knowledge (HVZK) which ensures that no information is gained by the verifier just by looking at

the transcript. This is usually proven by showing the existence of a simulator that can generate transcripts that are computationally indistinguishable from transcripts generated by the interaction between the prover and the verifier. A detailed survey on sigma protocols can be found in [Dam02].

The Fiat-Shamir transform [FS87] is a generic method to convert any PoK protocol that is complete, special sound and HVZK into a signature scheme. The security of the Fiat-Shamir transform is proven to be secure both in the random oracle model (ROM) [AABN02] and in the quantum random oracle model (QROM) [Unr17], under certain conditions.

The idea behind the Fiat-Shamir transform is that the prover simulates the challenge that is usually sent by the verifier. Since this challenge should be chosen uniformly at random, the prover sets the challenge according to a cryptographic hash function receiving as input the message to be signed and the commitment chosen previously by the prover. More precisely, given a proof of knowledge $(\mathcal{P}, \mathcal{V})$, the prover computes com, then it sets $ch = \bar{f}(com, M)$ where $\bar{f}$ is a cryptographic hash function and M is the message to be signed. Finally, it computes $resp$ such that $(com, ch, resp)$ is a valid transcript. The signature of M is $(com, resp)$. To verify the validity of the signature, one just has to compute $ch = \bar{f}(com, M)$ and check that $(com, ch, resp)$ is a valid transcript.

A.2 CDS construction

The Cramer-Damgård-Shoenmakers (CDS) construction [CDS94] is a generic way to construct a proof of knowledge $(\mathcal{P}^*, \mathcal{V}^*)$ where the prover proves knowledge of the solution to some subset of instances of a problem, given any PoK protocol $(\mathcal{P}, \mathcal{V})$ and a secret sharing scheme $\mathcal{SS}$.

Given N instances of a problem, let A be the set of indexes for which the prover $\mathcal{P}^*$ knows the solution. The idea behind the CDS construction is that the new prover $\mathcal{P}^*$ simulates transcripts $(com_j, ch_j, resp_j)$ for the instances it does not know the solution, that is, for $j \notin A$. For the instances that it knows the secret, it computes the commitment com_i, for $i \in A$, following the protocol $(\mathcal{P}, \mathcal{V})$. After receiving the commitments for all instances, the verifier sends a random bit string b to the prover. The string b will be interpreted as the secret in $\mathcal{SS}$ and the challenges ch_j, for $j \notin A$, as shares such that they form an unqualified set. Now, this set of shares can be extended to a qualified set by choosing properly the challenges ch_i, for $i \in A$. The prover then computes valid transcripts $(com_i, ch_i, resp_i)$ for $i \in A$. It can do this because it has witnesses for these instances. Finally, the prover $\mathcal{P}^*$ sends the transcripts $(com_i, ch_i, resp_i)$ for all i to the verifier. The verifier can check that these are valid transcripts and that the shares ch_i constitute a qualified set for $\mathcal{SS}$.

A.3 Stern's protocol

Stern's protocol [Ste94] is a protocol in which, given a matrix $\mathbf{H}$ and a syndrome vector $\mathbf{s}$, a prover proves the knowledge of an error vector $\mathbf{e}$ with $w(\mathbf{e}) = t$ and syndrome $\mathbf{s}$. The protocol is presented in Algorithm 4. Here, h denotes a cryptographic hash function.

The security of Stern's protocol is based on the hardness of the SD problem. The protocol has been proven to be complete, special sound and HVZK and, furthermore, has a cheating probability of 2/3 [Ste94].

Algorithm 4 Stern's protocol

1. **Public information:** $\mathbf{H} \in \mathbb{Z}_2^{n\times(n-k)}$ and $\mathbf{s} \in \mathbb{Z}_2^{n-k}$
2. **Secret information:** $\mathbf{e} \in \mathbb{Z}_2^n$ such that $\mathbf{He}^T = \mathbf{s}^T$ and $w(\mathbf{e}) = t$.
3. **Prover's commitment:**
   - $\mathcal{P}$ chooses $\mathbf{y} \leftarrow_{\$} \mathbb{Z}_2^n$ and a permutation δ;
   - $\mathcal{P}$ computes $c_1 = h(\delta, \mathbf{Hy}^T)$, $c_2 = h\left(\delta(\mathbf{y})\right)$ and $c_3 = h\left(\delta(\mathbf{y}+\mathbf{e})\right)$;
   - $\mathcal{P}$ sends c_1, c_2 and c_3.
4. **Verifier's challenge:** $\mathcal{V}$ sends $b \leftarrow_{\$} \{0,1,2\}$.
4. **Prover's answer:**
   - If $b = 0$, $\mathcal{P}$ reveals $\mathbf{y}$ and δ;
   - If $b = 1$, $\mathcal{P}$ reveals $\mathbf{y}+\mathbf{e}$ and δ;
   - If $b = 2$, $\mathcal{P}$ reveals $\delta(\mathbf{y})$ and $\delta(\mathbf{e})$.
6. **Verifier's verification:**
   - If $b = 0$, $\mathcal{V}$ checks if $h(\delta, \mathbf{Hy}^T) = c_1$ and $h\left(\delta(\mathbf{y})\right) = c_2$;
   - If $b = 1$, $\mathcal{V}$ checks if $h(\delta, \mathbf{H}(\mathbf{y}+\mathbf{e})^T + \mathbf{s}^T) = c_1$ and $h\left(\delta(\mathbf{y}+\mathbf{e})\right) = c_3$;
   - If $b = 2$, $\mathcal{V}$ checks if $h\left(\delta(\mathbf{y})\right) = c_2$, $h(\delta(\mathbf{y}) + \delta(\mathbf{e})) = c_3$ and $w\left(\delta(\mathbf{e})\right) = t$.

Appendix B Auxiliary results

B.1 Proof of Lemma 2

The probability of existing a vector $\mathbf{x}$ such that $\mathbf{Hx}^T = \mathbf{s}^T$ is the probability of $\mathbf{H}$ being a matrix representing a surjective application, i.e., it is the probability of $\mathbf{H}$ being a full rank matrix. Hence, we have to compute the probability of choosing k' linearly independent vectors of size n to form the rows of $\mathbf{H}$. We have

$$\Pr\left[\exists \mathbf{x} \in \mathbb{Z}_2^n : \mathbf{Hx}^T = \mathbf{s}^T\right] = \frac{(2^n-1)(2^n-2)\ldots(2^n-2^{k'})}{2^{k'n}}.$$

Since $(2^n-1) \geq (2^n-2^{k'})$, $(2^n-2) \geq (2^n-2^{k'})$ and $(2^n-2^{k'-1}) \geq (2^n-2^{k'})$, we have that

$$\frac{(2^n-1)(2^n-2)(2^n-4)\ldots(2^n-2^{k'})}{2^{k'n}} \geq \frac{\left(2^n-2^{k'}\right)^{k'+1}}{2^{k'n}} \geq \frac{\left(2^n-2^{k'}\right)^{k'}}{2^{k'n}}.$$

Now, note that

$$\frac{\left(2^n-2^{k'}\right)^{k'}}{2^{k'n}} = \frac{\left(2^n(1-2^{k'-n})\right)^{k'}}{2^{k'n}} = \left(1-\frac{1}{2^{n-k'}}\right)^{k'}.$$

So, it remains to show that

$$\left(1-1/2^{n-k'}\right)^{k'} = 1-\mathsf{negl}(n)$$

for $k' \leq n/2$. Note that the expression decreases with k' and so it is enough to show for $k' = n/2$.

Expanding the expression on the left using the Binomial theorem we get

$$\left(1 - \frac{1}{2^{n/2}}\right)^{n/2} = \sum_{i=0}^{n/2} \binom{n/2}{i} \left(-\frac{1}{2^{n/2}}\right)^i .$$

When $i = 0$ we have

$$\binom{n/2}{i} \left(-\frac{1}{2^{n/2}}\right)^i = 1.$$

The expression is maximal when $i = n/4$. Hence, if we show that

$$\binom{n/2}{i} \left(-\frac{1}{2^{n/2}}\right)^i = \mathsf{negl}(n)$$

when $i = n/4$, then

$$\sum_{i=0}^{n/2} \binom{n/2}{i} \left(-\frac{1}{2^{n/2}}\right)^i = 1 + \sum_{i=1}^{n/2} \binom{n/2}{i} \left(-\frac{1}{2^{n/2}}\right)^i = 1 - \mathsf{negl}(n)\,.$$

In fact, it can be proved using Stirling approximation (which is tight) for $n!$ that

$$\lim_{n\to\infty} n^b \binom{n/2}{n/4} \left(-\frac{1}{2^{n/2}}\right)^{n/4} = 0$$

for any $b \in \mathbb{N}$. Hence, we have shown that the expression $\binom{n/2}{n/4}\left(-\frac{1}{2^{n/2}}\right)^{n/4}$ goes to zero faster than any function of the form $1/n^b$, for any $b \in \mathbb{N}$. Thus, the expression is negligible in n and the result follows. □

Appendix C Security proofs

C.1 Proof of Lemma 7

We will prove that

$$\mathtt{Pr}\left[\left|\left\{i \in \mathbb{N} : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) = t \wedge \mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T \wedge \widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T\right\}\right| \neq 1\right] = \mathsf{negl}(n)$$

and the results follows. We divide the proof in two cases.

Case 1. $\mathtt{Pr}\left[\left|\left\{i \in \mathbb{N} : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) = t \wedge \mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T \wedge \widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T\right\}\right| < 1\right]$

If the signature is valid, but there is no $\mathbf{e} \in \mathbb{Z}_2^n$ such that $w(\mathbf{e}) = t$, $\mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T$ and $\widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T$, then the signer was able to forge a proof of knowledge for $\binom{N}{1}$-GStern's protocol. But this would break the soundness of the protocol, and recall that the probability of this event is negligible.

Case 2. $\mathtt{Pr}\left[\left|\left\{i \in \mathbb{N} : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) = t \wedge \mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T \wedge \widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T\right\}\right| > 1\right]$

If the signature is valid, then there is at least one index i such that there is $\mathbf{e}_i \in \mathbb{Z}_2^n$ that satisfies $w(\mathbf{e}_i) = t$, $\mathbf{H}\mathbf{e}_i^T = \mathbf{s}_i^T$ and $\widetilde{\mathbf{H}}\mathbf{e}_i^T = \mathbf{r}_i^T$. The probability of existing more than one index that satisfies these conditions is the probability that $\mathbf{r}_i + \mathbf{x}$ (where $\mathbf{x}$ is a random vector) gives the actual syndrome of another

$\mathbf{e}_j$ by the matrix $\widetilde{\mathbf{H}}$. That is, it is the probability of fixing an $\mathbf{s}_j$, such that there is a vector $\mathbf{e}_j$ with $w(\mathbf{e}_j) = t$ and $\mathbf{H}\mathbf{e}_j^T = \mathbf{s}_j^T$, and choosing $\mathbf{x}$ to be such that $\widetilde{\mathbf{H}}\mathbf{e}_j^T = (\mathbf{r}_i + \mathbf{x})^T$. Since f is treated has a random oracle, then the probability that the former happens for some index other than the actual signer is $q_f/2^{n-k}$ where q_f is the number of queries to f. Since there are N users, the total probability is $Nq_f/2^{n-k}$. □

C.2 Proof of Lemma 8

The proof follows from noticing that $|traceList| > 1$ only happens if collisions for one of the cryptographic hash function used in the protocol are found, given that the signatures were traced. Given two valid pairs of message and signature (L, M, σ) and (L, M', σ') such that $Trace(L, M, \sigma, M', \sigma') = indep$, this means that $|traceList| \neq 1$ and $|traceList| \neq \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$. If both signatures are valid, then there are indexes i and j for which there are unique vectors $\mathbf{e}$ and $\mathbf{e}'$ such that both have weight t and $\mathbf{H}\mathbf{e}^T = \mathbf{s}_i^T$, $\widetilde{\mathbf{H}}\mathbf{e}^T = \mathbf{r}_i^T$, $\mathbf{H}\mathbf{e}'^T = \mathbf{s}_j^T$ and $\widetilde{\mathbf{H}}\mathbf{e}'^T = \mathbf{r}_j^T$, with probability very close to one by the previous lemma. So, the probability that both signatures have more than two error vectors is the probability that collisions are found for the hash function f which is equal to $q_f/2^{n-k}$, where q_f is the number of queries to f. □

C.3 Proof of tag-linkability

To prove tag-linkability we have to prove that the advantage of a adversary $\mathcal{A}$ in the tag-linkability game is negligible. Assume that there is an adversary $\mathcal{A}$ with non-negligible probability of breaking tag-linkability of the proposed traceable ring signature scheme. Then, $\mathcal{A}$ can find a tag $L = (issue, \overline{\mathbf{pk}})$, where $\overline{\mathbf{pk}} = \{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}$, and $N + 1$ pairs of message and signature (M_i, σ_i) such that $Ver(L, M_i, \sigma_i) = 1$ and $Trace(L, M_i, \sigma_i, M_j, \sigma_j) = indep$ for $i, j \in \{1, \ldots, N+1\}$ and $i \neq j$.

By Lemma 8, we have that

$$\mathtt{Pr}\left[|traceList_{i,j}| = 0 \mid Trace(L, M_i, \sigma_i, M_j, \sigma_j) = indep\right] = 1 - \mathsf{negl}(n)$$

for any $i \neq j$ and where $traceList_{i,j}$ is the list obtained by applying *Trace* algorithm to $(L, M_i, \sigma_i, M_j, \sigma_j)$. Remark that, if $|traceList| = 1$, then the *Trace* algorithm would not output *indep* but rather the only element in the list.

On the other hand, by Lemma 7, we have that

$$\mathtt{Pr}\left[\left|\{i : \exists \mathbf{e} \in \mathbb{Z}_2^n \quad w(\mathbf{e}) = t \wedge \mathbf{H}\mathbf{e} = \mathbf{s}_i \wedge \widetilde{\mathbf{H}}\mathbf{e} = \mathbf{r}_i\}\right| = 1\right] = 1 - \mathsf{negl}(n)\,.$$

By the pigeonhole principle (there are only N different syndromes $\mathbf{s}_i$ but there are $N + 1$ valid signatures), with overwhelming probability, there are $i, j \in \{1, \ldots, N + 1\}$, where $i \neq j$, and a position k such that $\mathbf{r}_k = \mathbf{r}'_k$ where $\mathbf{r}_k$ is part of the signature σ_i and $\mathbf{r}'_k$ is part of the signature σ_j. This contradicts Lemma 8. So, we conclude that the advantage of $\mathcal{A}$ is negligible. □

C.4 Proof of anonymity

We will prove that, given an adversary $\mathcal{B}$ that breaks anonymity of our protocol with some non-negligible advantage ϵ, then we can build an $\mathcal{A}$ algorithm that

decides language GSD given $t/2$ positions of the error vector as input.

The adversary $\mathcal{A}$ works in the following way. First, $\mathcal{A}$ receives as input a tuple $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t)$ and $t/2$ of non-null positions of a solution of $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$ and $\mathbf{G}\mathbf{e}^T = \mathbf{r}^T$. Recall that the goal of $\mathcal{A}$ is to decide whether $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \text{GSD}$.

The algorithm $\mathcal{A}$ starts by choosing a random bit $b \leftarrow_{\$} \mathbb{Z}_2$ and setting $\mathsf{pk}_b = \mathbf{s}$. Now, consider the set of all vectors that have non-null entries in the $t/2$ positions received as input by $\mathcal{A}$. $\mathcal{A}$ chooses randomly $\mathbf{e}'$ with weight t from this set, and sets $\mathsf{pk}_{1-b}^T = \mathbf{H}\mathbf{e}'^T + \mathbf{s}^T$. Note that, if $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \text{GSD}$, both the witnesses $\mathbf{e}$ and $\mathbf{e}'$ will have weight t and non-null values in these $t/2$ positions, received as input by $\mathcal{A}$. When $(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t)$ is a random tuple, then pk_{1-b} will also be a random tuple. Since pk_{1-b} depends on pk_b, either both keys produce valid signatures or both keys produce non-valid signatures. This is necessary because we know how $\mathcal{B}$ behaves when it is given two valid public keys (it outputs an answer with an advantage of ϵ) or when it is given two random values as public keys (here, it has no advantage at all and its best guess is to choose at random). On the other hand, we do not know what happens when $\mathcal{B}$ is given a random value and a valid public key. For example, it could be the case that $\mathcal{B}$ distinguishes random tuples from valid public keys with probability 1, before distinguishing signatures, and this would give $\mathcal{B}$ an advantage of 1.

Next, $\mathcal{A}$ feeds pk_0 and pk_1 to $\mathcal{B}$. The adversary $\mathcal{B}$ will now make queries to h, f, g and $\bar{f}$ (here, treated as random oracles) and to the oracles $\mathit{Sign}(\mathsf{sk}_0, \cdot)$, $\mathit{Sign}(\mathsf{sk}_1, \cdot)$ and $\mathit{Sign}(\mathsf{sk}_b, \cdot)$. We describe how $\mathcal{A}$ can simulate each one of these oracles.

Simulation of h, f and $\bar{f}$: $\mathcal{B}$ can submit two types of queries: a new one, or one that has already been asked before. When $\mathcal{B}$ submits a new query, $\mathcal{A}$ chooses uniformly at random an output value and sends it to $\mathcal{B}$. If the query was already asked before, $\mathcal{A}$ returns precisely the same value that it had returned before.

Simulation of g: Similarly to previous case, when $\mathcal{B}$ submits a new query, $\mathcal{A}$ chooses a random invertible square matrix $\mathbf{P}$ and returns $\mathbf{PG}$. If the query was already asked, $\mathcal{A}$ returns the same value that it had returned before.

Simulation of the oracle $\mathit{Sign}(\mathsf{sk}_b, \cdot)$: When $\mathcal{B}$ submits a query (L, M) to the oracle $\mathit{Sign}(\mathsf{sk}_b, \cdot)$, $\mathcal{A}$ chooses a random invertible square matrix $\mathbf{P}$ and sets $g(L) = \mathbf{PG}$ and $\mathbf{r}_b = \mathbf{Pr}$. Moreover, $\mathcal{A}$ chooses a random value for A_0 and for the several calls of oracle f. Furthermore, $\mathcal{A}$ simulates a proof for the $\binom{N}{1}$-GStern's protocol setting $\bar{f}(Com, M)$ to something that $\mathcal{A}$ knows how to answer. Of course there is the risk of collisions for g, but the probability of such event is negligible. If the query was already asked before, $\mathcal{A}$ returns the same signature that it had returned before.

Simulation of the oracle $\mathit{Sign}(\mathsf{sk}_{b-1}, \cdot)$: When $\mathcal{B}$ submits a query (L, M) to the oracle $\mathit{Sign}(\mathsf{sk}_{b-1}, \cdot)$, $\mathcal{A}$ chooses a random full rank square matrix $\mathbf{P}$ and sets $g(L) = \mathbf{PG}$ and $\mathbf{r}_{b-1} = \mathbf{PG}\mathbf{e}' + \mathbf{Pr}$. It generates a signature by simulating a proof for the $\binom{N}{1}$-GStern's protocol, in a similar way as above. If the query was already asked before, $\mathcal{A}$ returns the same signature that it had returned before.

The adversary $\mathcal{B}$ can query these oracles a polynomial number of times. Eventually, $\mathcal{B}$ outputs a bit b'. If $b = b'$, then $\mathcal{A}$ outputs 1; else, it outputs a bit $b'' \leftarrow_{\$} \mathbb{Z}_2$.

It remains to analyze the advantage of $\mathcal{A}$, defined as

$$\mathtt{Pr}\left[1 \leftarrow \mathcal{A} \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] - \mathtt{Pr}\left[1 \leftarrow \mathcal{A} \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \notin \mathrm{GSD}\right].$$

We start by the term $\mathtt{Pr}\left[1 \leftarrow \mathcal{A} \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right]$. Note that

$$\begin{aligned}\mathtt{Pr}\left[1 \leftarrow \mathcal{A}|\ (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] = \mathtt{Pr}\left[b = b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] + \\ \mathtt{Pr}\left[b'' = 1 \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD} \wedge b \neq b'\right] \mathtt{Pr}\left[b \neq b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right].\end{aligned}$$

Moreover, by assumption, we have that:

1. $\mathtt{Pr}\left[b = b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] = 1/2 + \epsilon$;
2. $\mathtt{Pr}\left[b \neq b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] = 1 - 1/2 - \epsilon$.

Furthermore, by construction of $\mathcal{A}$, we have

$$\mathtt{Pr}\left[b'' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD} \wedge b \neq b'\right] = 1/2.$$

Hence, $\mathtt{Pr}\left[1 \leftarrow \mathcal{A} \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \in \mathrm{GSD}\right] = 3/4 + \epsilon/2$.

Concerning the second term of the advantage of $\mathcal{A}$, we have that

$$\begin{aligned}\mathtt{Pr}\left[1 \leftarrow \mathcal{A}|\,(\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \notin \mathrm{GSD}\right] = \mathtt{Pr}\left[b = b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \notin \mathrm{GSD}\right] + \\ \mathtt{Pr}\left[b'' = 1 \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \notin \mathrm{GSD} \wedge b \neq b'\right] \mathtt{Pr}\left[b \neq b' \mid (\mathbf{H}, \mathbf{s}, \mathbf{G}, \mathbf{r}, t) \notin \mathrm{GSD}\right]\end{aligned}$$

which is equal to $1/2 + 1/2^2 = 3/4$ since the bit b is perfectly hidden from $\mathcal{D}$.

We conclude that the advantage of $\mathcal{A}$ is at least $\epsilon/2$ minus the probability of $\mathcal{A}$ setting the same random value for different queries to the random oracles. □

C.5 Proof of exculpability

Suppose that there is a polynomial-time adversary $\mathcal{B}$ that breaks exculpability of the proposed scheme with a non-negligible advantage of ϵ. We will show that, in this case, we can conceive a polynomial-time algorithm $\mathcal{A}$ that solves the SD problem with some non-negligible probability.

First, $\mathcal{A}$ receives as input an instance $(\mathbf{H}, \mathbf{s}, t)$ of the SD problem. The goal of $\mathcal{A}$ is to find an error vector $\mathbf{e}$ such that $w(\mathbf{e}) = t$ and $\mathbf{H}\mathbf{e}^T = \mathbf{s}^T$, given access to $\mathcal{B}$. $\mathcal{A}$ sets the public key pk as $(\mathbf{H}, \mathbf{s}, t)$ and feeds it to $\mathcal{B}$.

As in the previous proof, we have to specify how does $\mathcal{A}$ simulate the oracles for $\mathcal{B}$.

Simulation of the oracles h, f, g and $\bar{f}$: When $\mathcal{B}$ submits a new query to one of these oracles, $\mathcal{A}$ chooses a random value an returns it to $\mathcal{B}$. If the query was already asked by $\mathcal{B}$, $\mathcal{A}$ returns the same value that it had previously returned. Note that the probability that $\mathcal{A}$ returns the same value for two different queries to the same oracle is negligible.

Simulation of $Sign(\mathsf{sk}, \cdot)$: When $\mathcal{B}$ submits a new query (L, M) to $Sign(\mathsf{sk}, \cdot)$, $\mathcal{A}$ chooses a random invertible square matrix $\mathbf{P}$ and sets $g(L) = \mathbf{PH}$ and $\mathbf{r}_i = \mathbf{Ps}$. $\mathcal{A}$ generates a signature in a similar way as in the previous proof: it simulates a proof for $\binom{N}{1}$-GStern's protocol setting $Ch = \bar{f}(Com, M)$ to some challenge for which it knows the correct answer. Again, note that it is possible that the query L to g and query (Com, M) to $\bar{f}$ were previously asked and set to a different value by $\mathcal{A}$, but the probability of this event is negligible. $\mathcal{A}$ stores all these queries and respective answers in a list Q_{Sign}.

At some point, $\mathcal{B}$ outputs two signatures (L, M, σ) and (L, M', σ') such that user i is framed by $\mathcal{B}$ with non-negligible advantage ϵ, that is,

$$\mathtt{Pr}\left[Trace(L, M, \sigma, M', \sigma') \geq \mathsf{pk}_i\right] = \epsilon.$$

Note that, at least, one of the signatures was not asked by $\mathcal{B}$ to $Sign(\mathsf{sk}, \cdot)$. Let us assume that the first one (L, M, σ) was not asked to the signing oracle, without loss of generality. We now use a similar technique as the one used in [FS07]. $\mathcal{A}$ reruns adversary $\mathcal{B}$ giving it the same values for the random oracles and signing oracle, except that $\mathcal{A}$ chooses another random value for $Ch'' = \bar{f}(Com, M)$ in the generation of the signature. $\mathcal{B}$ will output a new signature (L, M, σ'') with some non-negligible probability by the forking lemma [PS00]. Now, $\mathcal{A}$ can check if $ch_i \neq ch_i''$ and if this happens, it can recover the secret by the special soundness of the $\binom{N}{1}$-GStern's protocol. Recall that the $\binom{N}{1}$-GStern's protocol is special sound (as the original Stern's protocol [Ste94]) since one needs to open the three commitments in order to extract the witness. Since in each valid transcript, two commitments are opened, we just need two valid transcripts to extract a witness. Note that, the probability that $ch_i = ch_i''$ is negligible.

The advantage of $\mathcal{A}$ is the probability of $\mathcal{B}$ outputting a valid signature (L, M, σ'') given that it succeeded in the first run, which is non-negligible as we have seen. Hence, we conclude that $\mathcal{A}$ has a non-negligible probability of solving the GSD problem and we conclude the proof. □