A Post-Quantum UC-Commitment Scheme in the Global Random Oracle Model from Code-Based Assumptions

Pedro Branco*

Abstract

In this work, we propose the first post-quantum UC-commitment scheme in the Global Random Oracle Model, where only one non-programmable random oracle is available. The security of our proposal is based on two well-established post-quantum hardness assumptions from coding theory: The Syndrome Decoding and the Goppa Distinguisher. We prove that our proposal is perfectly hiding and computationally binding. The scheme is secure against static malicious adversaries.

1 Introduction

One of the most important cryptographic primitive in today's world are commitment schemes, both in theory and in practice. A commitment scheme involves two parties, usually called the committer C and the receiver R. C commits to a message and sends the commitment to R. Later, C may open the commitment and reveal the message to R, which checks if the opening is a valid one. Two security properties are required for this primitive: the hiding property, which means that R is not able to extract information about the message from the commitment before the opening phase, and the binding property, which means that C cannot open a different message from the one it has committed before.

This simple, yet powerful, primitive has found numerous applications such as zero-knowledge proofs, signature schemes, secure multi-party computation, e-voting or, even, key-exchange. Due to its versatility, a commitment scheme should be secure, not just per se, but also when composed with the same or other protocols. Hence, security of commitment should be analyzed in the UC-framework [Can01], where security under arbitrary composition can be proven.

Commitment schemes can be obtained from oblivious transfer (OT) via a generic compiler [Kil88]. However, the resulting commitment scheme is too inefficient for practical purposes since it requires a large number of OT executions. From a practical perspective, explicit constructions for commitment schemes

*SQIG - IT, IST - University of Lisbon. Email: pmbranco@math.tecnico.ulisboa.pt.

are more interesting since they are designed to be more efficient than generic transformations.

As far as we know, all the explicit constructions for UC-commitment schemes have their security based on number-theoretic hardness assumptions (e.g., [CF01, Lin11, CJS14]) and, thus, they can be broken using Shor's algorithm [Sho97]. With the growing interest in quantum technologies and as we enter the post-quantum era, it becomes crucial to find post-quantum alternatives to this type of protocols.

In this work, we address this problem by presenting the first explicit construction for a post-quantum UC-commitment scheme. Our proposal is proven to be secure in the Global Random Oracle (gRO) model [CJS14]. We prove that our scheme is perfectly hiding (meaning that even a receiver with unlimited computational power cannot break the hiding property) and computationally binding given that the Syndrome Decoding (SD) and the Goppa Distinguisher (GD) assumptions hold.

1.1 Previous work

Commitment schemes. There is a large amount of work done concerning UC-commitment schemes [CF01, DN02, HMQ04, Lin11, FLM11, CJS14, Fuj16, BPRS17]. However, all of these protocols are based on number-theoretic assumptions. Thus, their security is threatened, since Shor's quantum algorithm breaks all of these protocols [Sho97].

A UC-commitment scheme must allow extraction and equivocation by the simulator. These properties are essential to perform the simulation and to prove security in the UC-framework [CF01]. All of the schemes mentioned above take advantage of the nice properties inherent to number-theoretic assumptions to be able to extract and equivocate.[1] However, it seems that most of the techniques used in these schemes cannot be straightforwardly adapted using post-quantum assumptions. Thus, the task of explicitly constructing a post-quantum commitment scheme that is both extractable and equivocable is highly non-trivial.

Post-quantum commitment schemes exist, e.g., [JKPT12], based on the LPN assumption, and [XXW13], based on the RLWE assumption. However, we are not aware of any commitment scheme based on post-quantum assumptions and proven to be secure in the UC-framework.

We remark that commitment schemes can also be obtained from OT [Kil88, GIKW14], and for which there are UC-secure post-quantum proposals (e.g., [PVW08, BDD+17, BDGM18]). However, the resulting commitment scheme is too inefficient and it just has theoretical value.

UC-commitment schemes are used as building block in the design of a wide range of applications, and they are known to imply key-exchange [DG03] and general forms of secure two and multi-party computation [CLOS02].

[1] For example, the scheme in [Lin11] uses efficient zero-knowledge proofs based on the discrete-logarithm assumption, and [CJS14] uses a trapdoor version of the famous Pedersen commitment scheme.

Global random oracle model. As it is well-known, it is impossible to design UC-commitment schemes in the plain model [CF01], thus we work on the Global Random Oracle (gRO) Model. The gRO Model is a more realistic model where only one *global* random oracle is available for every party in every execution of the same, or other, protocol. The notion of a gRO was firstly introduced in [CJS14] to overcome the fact that proofs in the Random Oracle Model (ROM) [BR93] assume a different random oracle for each execution of the protocol. Hence, security under composition may not be guaranteed when we replace the random oracles by a practical cryptographic hash function (which is usually the same for every execution of every protocol).

The main difference between the gRO Model and the ROM is that, in the gRO Model, we restrict the observability and the programmability power of the simulator. More precisely, the simulator is restricted to observe only adversarial queries (queries made by an adversary and not by a honest party) and it is not allowed to program the random oracle. Observe that, if the simulator can program the random oracle, then we can not use the same random oracle in other executions of the same or other protocols. If we do so, an environment could distinguish the ideal and the real world executions by just asking the same queries in two different executions of the protocol, for example. Also, note that we cannot restrict completely the power of the simulator to observe queries. If we do so, the simulator has the same power of the parties involved in the protocol and cannot perform the simulation necessary for security proofs [CJS14] (at least, in the case of commitment schemes [CDG$^+$18]). Hence, the simulator is given only the power to observe queries made by an adversary (but not by a honest party).

By using this security model, the discrepancy between the abstract model and the practical usage of a protocol is softened, since we can just replace calls to the gRO by calls to a *global* cryptographic hash function. This model was also studied in [CDG$^+$18], where it was called the restricted Observable Global Random Oracle Model.

Protocols proven to be UC-secure in the gRO Model include number-theoretic OT and commitment schemes [CJS14, BPRS17], and a two-party computation scheme [CJS14]. Recall that an UC-commitment scheme should allow extractability and equivocability by a simulator. The scheme of [CJS14] uses a trapdoor version of the Pedersen commitment scheme to achieve equivocation. However, as far as we know, there is no post-quantum analogue of this trapdoor version of Pedersen commitment. Hence, we have to follow a different strategy.

1.2 Outline of the protocol

The main contribution of this paper is the construction of a post-quantum UC-commitment scheme. As far as we know, this is the first post-quantum commitment scheme proven to be secure in the UC-framework, in the gRO Model.

As mentioned before, to achieve UC-security for the commitment scheme, we have to construct a commitment scheme such that the simulator in the ideal-

world execution is able to extract the committed message from a corrupted committer and it is able to equivocate a commitment. In our scheme, the committer C queries the gRO on an input that contains the message and this allows for extractability. On the other hand, equivocation is possible due to the use of a zero-knowledge proof-of-knowledge (ZKPoK) with the honest-verifier zero-knowledge (HVZK) property.

More precisely, let $\mathbf{B}$ be a public random matrix. The receiver R starts by sending a matrix $\mathbf{A}$ together with the hash of a trapdoor for it. C commits to its message M by computing $\mathbf{c}_1 = \mathbf{r}_1\mathbf{A}+\mathsf{gRO}(M,t)+\mathbf{e}_1$ and $\mathbf{c}_2 = \mathsf{gRO}(\mathbf{r}_1, M, t)\mathbf{A}+\mathbf{r}_2\mathbf{B}+\mathbf{e}_2$ where $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors and $\mathbf{e}_1$ and $\mathbf{e}_2$ are error vectors. We can already see that the simulator is able to extract the message by looking at the queries to gRO and finding the ones that fulfill the conditions.

To open a message, C opens $\mathbf{r}_1, M, t$. Then both C and R engage in a ZKPoK [JKPT12] where C proves that it has a witness for public information $\mathbf{c}_2 - \mathsf{gRO}(\mathbf{r}_1, M, t)\mathbf{A}$. In order for the simulator to equivocate in this phase of the protocol, it extracts the trapdoor for $\mathbf{A}$ and finds another $\mathbf{r}'_1$ such that $\mathbf{c}_1-(\mathbf{r}'_1\mathbf{A}+\mathsf{gRO}(M',t'))$ has low Hamming weight. This can be done on polynomial time for certain classes of codes and for an appropriate choice of parameters (e.g. [CFS01]). Now, the simulator must prove that it has a secret for public information $\mathbf{c}_2-\mathsf{gRO}(\mathbf{r}'_1, M', t')\mathbf{A}$ (which it does not have). We make R to commit to the challenge of the ZKPoK using gRO before C sends the first message of the ZKPoK. In this way, the simulator knows which is the challenge that R is going to ask and can cheat in the protocol, using the HVZK property of the ZKPoK. This technique is similar to the one used in [Lin11]. However, since the scheme of [Lin11] is proven in the CRS model, a dual-mode encryption scheme [PVW08] is used as a commitment scheme (for technical reasons), while in our scheme, it is enough to use the gRO as a commitment scheme.

We prove that the scheme is perfectly hiding and computationally binding. We also prove security against static malicious adversaries in the UC-framework, in the gRO Model.

Since we base the security of our scheme in well-established code-based assumptions, namely the SD and the GD assumptions, our protocol is considered to be post-quantum and it is the first explicit construction for UC-commitment that has this property.

2 Preliminaries

We denote matrices by capital bold letters (e.g., $\mathbf{A}$) and vectors by bold lowercase letters (e.g., $\mathbf{v}$). If S is a finite set, $|S|$ denotes its cardinality and $x \leftarrow\!\!\$\, S$ denotes the experiment of choosing an element x uniformly at random from S. If $\mathcal{A}$ is an algorithm, $y \leftarrow \mathcal{A}(x)$ denotes the experiment of running $\mathcal{A}$ on input x and setting the output to be y. If $\mathbf{x}$ is a vector, $w(\mathbf{x})$ denotes its Hamming weight, that is, the number of coordinates of $\mathbf{x}$ which are different from zero. If $\mathbf{A}$ is a matrix, we denote its transpose by $\mathbf{A}^T$.

A k-dimensional binary (linear) code $\mathcal{C}$ of length n is defined by its generating

matrix $\mathbf{A} \in \{0,1\}^{k\times n}$, that is,

$$\mathcal{C} = \{\mathbf{c} \in \{0,1\}^n : \exists \mathbf{s} \in \{0,1\}^k \text{ s.t. } \mathbf{c} = \mathbf{sA}\},$$

or by its parity-check matrix $\mathbf{H} \in \{0,1\}^{(n-k)\times n}$, that is,

$$\mathcal{C} = \{\mathbf{c} \in \{0,1\}^n : \mathbf{Hc}^T = 0\}.$$

The generating and parity-check matrices fulfill the condition $\mathbf{HA}^T = 0$.

Throughout this work, let $\mathfrak{B}^n_{=\omega} = \{\mathbf{e} \in \{0,1\}^n : w(\mathbf{e}) = \omega\}$ and $\mathfrak{B}^n_{\leq\omega} = \{\mathbf{e} \in \{0,1\}^n : w(\mathbf{e}) \leq \omega\}$.

In the description of the protocols, we denote by $x =_? y$ the experiment of testing if x and y are equal. If they are not, then the party executing the experiment aborts the protocol.

A PPT algorithm means a probabilistic polynomial-time algorithm.

2.1 Hardness assumptions in coding theory

We present the Syndrome Decoding (SD) problem, which states that it is hard to decode a linear code chosen uniformly at random. The problem is proven to be NP-complete in the worst-case [BMvT78].

Definition 1 (Syndrome Decoding)**.** Let $n, k, \omega \in \mathbb{N}$, $\mathbf{H} \leftarrow_\$ \{0,1\}^{(n-k)\times n}$ and $e \leftarrow_\$ \mathfrak{B}^n_{\leq\omega}$. The SD_ω problem is ε-hard if for every PPT algorithm D we have

$$\Pr\left[\mathbf{e} \leftarrow \mathsf{D}(\mathbf{H}, \mathbf{He}^T)\right] \leq \varepsilon.$$

The SD problem is a classical problem in coding theory and, by now, it is a well-established problem in code-based cryptography. Although no worst-case to average-case reduction is known, it is widely believed that, for an appropriate choice of parameters, the problem is hard for any random instance since the best known classical and quantum attacks still run in exponential time (e.g. [BJMM12, EKM17, Kir18]).

The following problem is a particular case of the SD problem, which was presented in [JKPT12].

Definition 2 (Exact Learning Parity with Noise)**.** Let $\tau \in]0, 1/2[$, $n, k, \omega \in \mathbb{N}$, $\omega = \lfloor \tau n \rceil$, $\mathbf{s} \leftarrow_\$ \{0,1\}^k$, $\mathbf{A} \leftarrow_\$ \{0,1\}^{k\times n}$ and $e \leftarrow_\$ \mathfrak{B}^n_{=\omega}$. The decisional version of the xLPN_τ problem is (n, ε)-hard if for every PPT algorithm D we have

$$\left|\Pr[1 \leftarrow \mathsf{D}(\mathbf{A}, \mathbf{sA} + \mathbf{e})] - \Pr[1 \leftarrow \mathsf{D}(\mathbf{A}, \mathbf{r})]\right| \leq \varepsilon$$

where $\mathbf{r} \leftarrow_\$ \{0,1\}^n$. The search version of the $\mathsf{xLPN}_{\tau,k}$ problem is (n, ε)-hard if for every PPT algorithm D we have

$$\Pr[\mathbf{s} \leftarrow \mathsf{D}(\mathbf{A}, \mathbf{sA} + \mathbf{e})] \leq \varepsilon.$$

It is easy to see that the xLPN is a particular case of the SD problem, that is, if we are able to solve the $\mathsf{SD}_{\omega,n}$ problem then we are able to solve the $\mathsf{xLPN}_{\tau,n}$, where $\omega = \lfloor \tau n \rceil$.

The proof of knowledge of [JKPT12]. Recall that a sigma protocol is a three-round protocol (commitment *com*, challenge *ch* and response *resp*) between two parties, a prover P and a verifier V. P tries to convince V that some statement is true. A zero-knowledge proof-of-knowledge (ZKPoK) is a sigma protocol where P, not only proves that the statement is true, but also proves that it has a witness for it. Figure 1 presents a scheme of any sigma protocol.

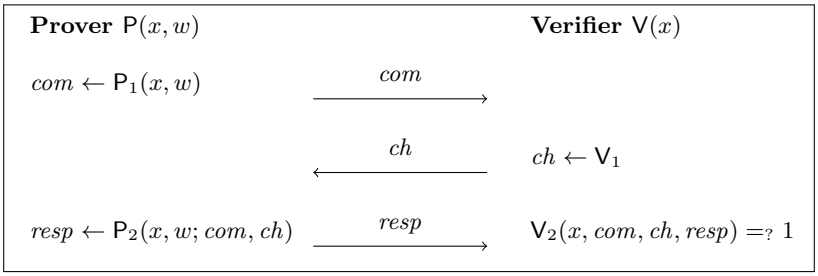

Figure 1: *Sigma protocol structure. The value x is public information and w is usually called the witness. Let $\sim$ be a relation and $\mathcal{R} = \{(x, w) : x \sim w\}$. A transcript $T = (com, ch, resp)$ is the tuple of messages exchange and we say that it is valid when $\mathsf{V}_2(x, T) = 1$.*

Besides being correct and special sound, a ZKPoK should also be honest-verifier zero-knowledge (HVZK). This property ensures that no information is gained by V by looking at a valid transcript. This is usually proved by showing the existence of a simulator that can generate transcripts that are computationally indistinguishable from transcripts generated by the interaction between P and V.

We present a ZKPoK where, given a matrix $\mathbf{A}$ and a vector $\mathbf{y}$, the prover is able to prove knowledge of vector $\mathbf{s}$ and $\mathbf{e}$ such that $\mathbf{sA} + \mathbf{e} = \mathbf{y}$ and $\mathbf{e} \in \mathfrak{B}^n_{=\omega}$. This protocol was proposed in [JKPT12]. The protocol is presented in Figure 2.

Theorem 3 ([JKPT12])**.** $(\mathsf{P}, \mathsf{V})_{\mathsf{xLPN}}$ *is a complete, special sound and HVZK ZKPoK for the relation*

$$\mathcal{R}_{\mathsf{xLPN}} = \{((\mathbf{A}, \mathbf{y}), (\mathbf{s}, \mathbf{e})) : \mathbf{sA} + \mathbf{e} = \mathbf{y} \wedge w(\mathbf{e}) = t\}.$$

The cheating probability of $(\mathsf{P}, \mathsf{V})_{\mathsf{xLPN}}$ is 2/3. This means that a cheating prover can still answer correctly for two possible values of the challenge *ch*. More precisely, as a consequence of being HVZK, if the prover knows the challenge beforehand, then it can cheat in the protocol. In this work, we use this property of $(\mathsf{P}, \mathsf{V})_{\mathsf{xLPN}}$ in the security proof of the commitment scheme of Section 3.

2.2 On the existence of trapdoors for codes

We present the Goppa Distinguisher (GD) problem, one of the assumptions that guarantee the security of the McEliece public-key encryption scheme [McE78]

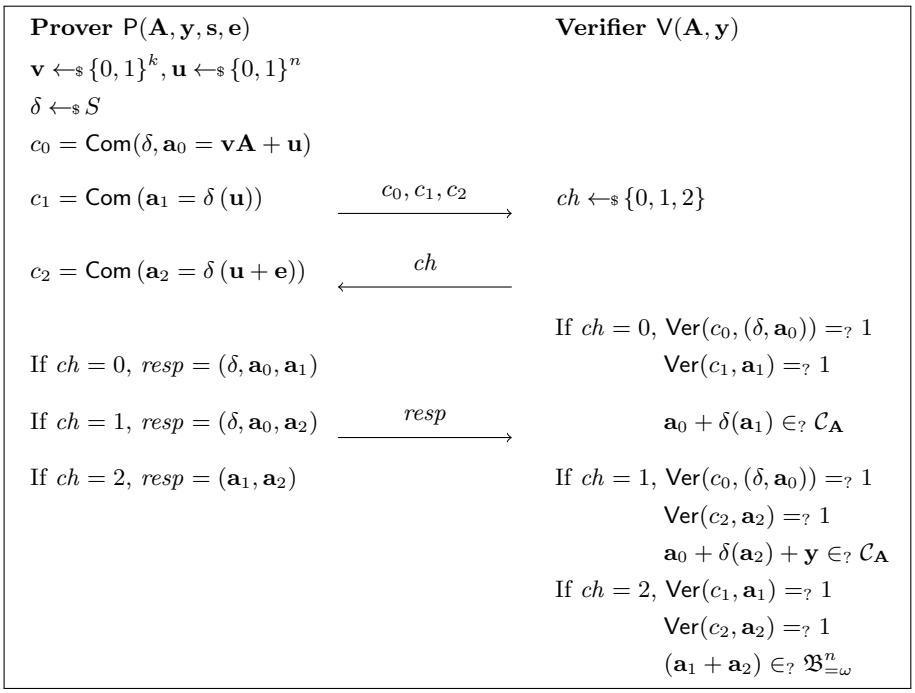

Figure 2: $(\mathsf{P}, \mathsf{V})_{\mathsf{xLPN}}$ scheme. Let $\mathbf{A} \in \{0,1\}^{k \times n}$, $\mathbf{s} \in \{0,1\}^k$, $\mathbf{e} \in \{0,1\}^n$ such that $\mathbf{e} \in \mathfrak{B}^n_{=\omega}$ and let $\mathbf{y} \in \{0,1\}^n$ such that $\mathbf{sA} + \mathbf{e} = \mathbf{y}$. By $\mathcal{C}_\mathbf{A}$ we denote the code defined by $\mathbf{A}$. Let S be the set of permutations of size n and let $(\mathsf{Com}, \mathsf{Ver})$ be a commitment scheme where Com is the commitment algorithm and Ver is the opening algorithm.

and several code-based cryptosystems. This problem was firstly assumed to be hard in [CFS01] in the context of signature schemes.

Definition 4 (Goppa Distinguisher)**.** Let $n, k \in \mathbb{N}$ such that $k < n$. Let $\mathsf{Gop}(n, k)$ be an algorithm that outputs a matrix defining a binary Goppa code of size $n \times k$. The GD is ε-hard, if for every PPT algorithms D, we have

$$|\mathsf{Pr}[1 \leftarrow \mathsf{D}(\mathbf{A}) : \mathbf{A} \leftarrow \mathsf{Gop}(n, k)] - \mathsf{Pr}\left[1 \leftarrow \mathsf{D}(\mathbf{A}) : \mathbf{A} \leftarrow_\$ \{0,1\}^{k \times n}\right]| \leq \varepsilon.$$

Although the problem can be solved for codes with high rate (that is, when $k \approx n$) [FGUO+11], the GD problem is assume to be computationally hard in the general case, since the best known attack still has exponential runtime [LS01].

The following lemma guarantees the existence of code-based trapdoors (also called decoding algorithms).

Lemma 5 ([CFS01])**.** *Let $n, k, \omega \in \mathbb{N}$ such that $k < n$. There exists an algorithm $\mathsf{GenTd}_\mathbf{H}$ that receives as input n, k and ω and outputs a pair $(\mathbf{H}, \mathsf{td})$ where*

$\mathbf{H} \in \{0,1\}^{(n-k)\times n}$ *is a matrix and* td *is a trapdoor for* $\mathbf{H}$ *with the following properties:*

- *There is an algorithm* $\mathsf{Decode}_{\mathbf{H}}$ *that takes as input* td *and a word* $\mathbf{z} \in \{0,1\}^{n-k}$*. It outputs either* $\mathbf{e}$*, if* $\mathbf{z}^T = \mathbf{H}\mathbf{e}^T$ *and* $w(\mathbf{e}) \leq \omega$*, or a message error* $\perp$*, otherwise;*
- *There is a non-negligible number of words* $\mathbf{z}$ *in* $\{0,1\}^{n-k}$ *for which* $\mathsf{Decode}_{\mathbf{H}}(\mathbf{z}, \mathsf{td})$ *does not output a message error* $\perp$*;*
- $\mathbf{H}$ *is indistinguishable from a uniformly chosen matrix* $\mathbf{U}$ *from* $\{0,1\}^{(n-k)\times n}$ *given that the* GD *problem is hard.*

From this lemma, we can derive the following corollary. It states that, if we have a trapdoor for parity-check matrices, then we also have a trapdoor for generating matrices.

Corollary 6. *Let* $n, k, \omega \in \mathbb{N}$ *such that* $k < n$*. There exists an algorithm* GenTd *that receives as input* n*,* k *and* ω *and outputs a pair* $(\mathbf{A}, \mathsf{td})$ *where* $\mathbf{A} \in \{0,1\}^{k\times n}$ *is a matrix and* td *is a trapdoor for* $\mathbf{A}$ *with the following properties:*

- *There is an algorithm* $\mathsf{Decode}_{\mathbf{A}}$ *that takes as input* td *and a word* $\mathbf{c} \in \{0,1\}^{n}$*. It outputs either* $\mathbf{s}$*, if* $\mathbf{c} = \mathbf{s}\mathbf{A} + \mathbf{e}$ *for some* $\mathbf{s} \in \{0,1\}^{k}$ *and* $w(\mathbf{e}) \leq \omega$*, or a message error* $\perp$*, otherwise;*
- *There is a non-negligible number of words* $\mathbf{c}$ *in* $\{0,1\}^{n}$ *for which* $\mathsf{Decode}_{\mathbf{A}}(\mathbf{c}, \mathsf{td})$ *does not output a message error* $\perp$*;*
- $\mathbf{A}$ *is indistinguishable from a uniformly chosen matrix* $\mathbf{U}$ *from* $\{0,1\}^{k\times n}$ *given that the* GD *problem is hard.*

Proof. Let $(\mathbf{H}, \mathsf{td}) \leftarrow \mathsf{GenTd}_{\mathbf{H}}$. Consider $\mathbf{A}$ to be the generating matrix associated with $\mathbf{H}$. That is, consider $\mathbf{A}$ such that $\mathbf{H}\mathbf{A}^T = 0$ and $\mathsf{Decode}_{\mathbf{A}}$ to be the algorithm that takes td and $\mathbf{c}$ as input and does the following:

1. It computes $\mathbf{H}\mathbf{c}^T = \mathbf{y}^T$;
2. It computes $\mathbf{e}' = \mathsf{Decode}_{\mathbf{H}}(\mathsf{td}, \mathbf{y})$;
3. If $\mathbf{e}' = \perp$, it outputs $\perp$;
4. Else, it outputs $\mathbf{c}' = \mathbf{c} - \mathbf{e}'$.

Note that, if $\mathbf{c}$ is of the form $\mathbf{s}\mathbf{A} + \mathbf{e}$, where $\mathbf{e}$ is a error vector, then $\mathbf{H}\mathbf{c}^T = \mathbf{H}(\mathbf{s}\mathbf{A})^T + \mathbf{H}\mathbf{e}^T = \mathbf{H}\mathbf{e}^T$, the $\mathsf{Decode}_{\mathbf{H}}$ algorithm outputs $\mathbf{e}$ and the procedure $\mathsf{Decode}_{\mathbf{A}}$ outputs $\mathbf{c}' = \mathbf{s}\mathbf{A}$. Else, if $\mathbf{c}$ is not of the form $\mathbf{s}\mathbf{A} + \mathbf{e}$, then the procedure $\mathsf{Decode}_{\mathbf{A}}$ outputs an error message $\perp$.

If there is a non-negligible number of words $\mathbf{y} \in \{0,1\}^{n-k}$ for which $\mathsf{Decode}_{\mathbf{H}}$ does not output a message error $\perp$, then there is also a non-negligible number of words $\mathbf{c} \in \{0,1\}^{n}$ such that $\mathsf{Decode}_{\mathbf{A}}$ does not output a message error $\perp$.

If $\mathbf{H}$ is indistinguishable from a uniform matrix in $\{0,1\}^{(n-k)\times n}$, then $\mathbf{A}$ is also indistinguishable from a uniform matrix in $\{0,1\}^{k\times n}$. To see this, assume that we can distinguish $\mathbf{A}$ from a uniformly chosen matrix. Then we can construct a distinguisher algorithm for $\mathbf{H}$, which contradicts the assumption. □

Recently, a new code-based trapdoor function was presented in [DAST18]. It has the same properties as the one presented in Lemma 6. The only difference is that this trapdoor is not based on the GD assumption but rather on the hardness of distinguish *generalized admissible* codes $(\mathbf{U}, \mathbf{U}+\mathbf{V})$ from uniformly chosen codes.

Observe that the construction of Section 3 works with any code-based trapdoor function fulfilling the conditions of Lemma 6. However, for simplicity, we explicitly use the trapdoor function of [CFS01], based on the GD assumption. The reason for using trapdoor functions based on the GD assumption in our construction is that its security well-study (or, at least, it is more studied than other code-based distinguisher assumptions). We do not care about its efficiency, since the trapdoor is only used in the security proof, as long as it runs in polynomial time.

Let ω be the error decoding capability of the code defined by $\mathbf{H}$. In order to be able to use the trapdoor in polynomial time, ω must be chosen well below the Gilbert-Varshamov bound [CFS01, DAST18]. Hence, 2ω is still much smaller than the Gilbert-Varshamov bound. In this work, we assume the hardness of the $\mathsf{SD}_{2\omega}$, where ω is the decoding capability of the code $\mathbf{H}$, and which is widely assumed to be hard in the average-case [DAST18].

2.3 UC-security and the Global random oracle model

The Universal Composability (UC) framework of Canneti [Can01] guarantees security of a protocol under arbitrary composition. Let π be a protocol and let $\mathcal{F}$ be an ideal functionality that implement the same cryptographic primitive. Let $\mathcal{E}$ be an environment that oversees both executions of the real-world protocol π and of the ideal-world functionality $\mathcal{F}$. We say that π is secure if no environment can distinguish the real-world execution from the ideal-world execution.

The Global Random Oracle. The notion of a Global Random Oracle gRO was firstly introduced in [CJS14]. We present its definition as in [CJS14]. As mentioned in the introduction, the gRO is available to every party. However, contrarily to the ROM, the simulator will only be able to observe to adversarial queries, that is, queries made by an adversary.

gRO functionality

Parameters: a range $\mathcal{D}$ of size $\ell(\kappa)$ and a list $\bar{\mathcal{F}}$ of ideal functionalities.
Upon receiving a query x from a party $\mathsf{P} = (\mathsf{pid}, \mathsf{sid})$ or from an adversary $\mathcal{A}$ do:

- If $(x, v) \in \mathbb{Q}$, return v to P;
- Else, $v \leftarrow\!\!\$\ \{0,1\}^{\ell(\kappa)}$, store (x, v) in $\mathbb{Q}$ and return v to P.
- Parse x as (s, x'). If $\mathsf{sid} \neq s$, add (s, x', v) to $\mathbb{Q}_{|\mathsf{sid}}$.

Upon receiving a request from an ideal functionality in the list $\bar{\mathcal{F}}$, with SID sid, return the list $\mathbb{Q}_{|\mathsf{sid}}$.

Let $\mathsf{IDEAL}_{\mathcal{F},\mathsf{S},\mathcal{E}}$ denote the distribution of the output of the environment $\mathcal{E}$ after the ideal-world execution of $\mathcal{F}$ with adversary S and $\mathsf{EXEC}^{\mathcal{G}_{\mathsf{gRO}}}_{\pi,\mathcal{A},\mathcal{E}}$ denote the output of $\mathcal{E}$ after the real-world execution of π with adversary $\mathcal{A}$ where every party has access to the ideal functionality gRO. Security in the gRO-hybrid model is defined as follows.

Definition 7 ([CJS14])**.** Let π be a protocol with n parties involved and an adversary $\mathcal{A}$. We say that π UC-realizes $\mathcal{F}$ in the $\mathcal{G}_{\mathsf{gRO}}$-hybrid model if for every PPT adversary $\mathcal{A}$ there is a PPT simulator S such that for all PPT environments $\mathcal{E}$,

$$\mathsf{IDEAL}_{\mathcal{F},\mathsf{S},\mathcal{E}} \approx \mathsf{EXEC}^{\mathcal{G}_{\mathsf{gRO}}}_{\pi,\mathcal{A},\mathcal{E}}$$

where $\mathcal{F}$ is an ideal functionality.

In this work, we consider only static malicious adversaries. That is, adversaries that may deviate in any arbitrary way from the protocol. However, parties involved in the protocol are corrupted before the beginning of the protocol and remain like that until the end.

Some remarks regarding the gRO. The gRO Model assumes the existence of a single global random oracle, available to every party. However, we remark that, from this single global random oracle (which outputs strings of a given size $\ell(\kappa)$), we can create other global random oracles with the output size that we want. For example, if we want a global random oracle gRO' with output size $2\ell(\kappa)$, we can define $\mathsf{gRO}'(x) = (\mathsf{gRO}(x)||\mathsf{gRO}(x+1))$ where $||$ is the concatenation of binary strings. Note that, since the output of gRO is completely random, then so it is the output of gRO'. Throughout this work, the size of the output of gRO is specified in each case. When it is not explicit, then we consider the size of the output to be $\ell(\kappa)$.

The commitment ideal functionality. We present the ideal commitment functionality $\mathcal{F}_{\mathsf{tcom}}$, as presented in [CJS14].

$\mathcal{F}_{\text{tcom}}$ **functionality**

Commitment phase: Upon receiving $(\mathsf{sid}, \mathsf{commit}, \mathsf{C}, \mathsf{R}, M)$ from C, $\mathcal{F}_{\text{tcom}}$ stores (sid, M) and sends $(\mathsf{sid}, \mathsf{receipt}, \mathsf{C}, \mathsf{R})$ to R. It ignores future commit messages from C with the same sid.

Opening phase: Upon receiving $(\mathsf{sid}, \mathsf{reveal}, \mathsf{C}, \mathsf{R})$ from C, $\mathcal{F}_{\text{tcom}}$ checks whether it has recorded $(\mathsf{sid}, \mathsf{commit}, \mathsf{C}, \mathsf{R}, M)$. If so, returns $(\mathsf{sid}, \mathsf{reveal}, \mathsf{C}, Rs, M)$ to R and halts.

When the adversary S asks for the list $\mathbb{Q}_{|\mathsf{sid}}$, with session ID sid, $\mathcal{F}_{\text{tcom}}$ obtains it from $\mathcal{G}_{\mathsf{gRO}}$ and sends it to S.

We define the security properties for a commitment scheme: binding and hiding properties. A binding commitment scheme is a scheme such that a commitment cannot be opened to two different messages. A hiding commitment scheme is a scheme that hides the message from the receiver. We say that a scheme is perfectly hiding if the former holds for any adversary (not necessarily running in polynomial time).

3 UC-commitment scheme

Consider two parties, a committer C and a receiver R. Suppose that C wants to commit to a message M of size λ.

Let $(\mathsf{P}^{\varepsilon}, \mathsf{V}^{\varepsilon})_{\mathsf{xLPN}}$, where $\mathsf{P}^{\varepsilon} = (\mathsf{P}_1^{\varepsilon}, \mathsf{P}_2^{\varepsilon})$ and $\mathsf{V}^{\varepsilon} = (\mathsf{V}_1^{\varepsilon}, \mathsf{V}_2^{\varepsilon})$ be the sigma-protocol $(\mathsf{P}, \mathsf{V})_{\mathsf{xLPN}}$ repeated $\mathcal{O}(1/\varepsilon)$ in order to obtain a negligible soundness error of ε.

Public parameters. Let $n, k, k', \omega \in \mathbb{N}$ such that $n > k$ and $n > k'$, $\mathbf{B} \in \{0,1\}^{n \times k'}$ be a public matrix uniformly chosen at random and let gRO be the ideal Global Random Oracle. Here, λ is the size of the message M.

Commitment phase. Both parties, the committer C and the receiver R, are activated by their inputs. They proceed as follows:

1. R generates $(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{GenTd}(n, k, \omega)$. It chooses a random string u_1 and queries gRO on $(\mathsf{sid}, \mathsf{R}, \mathsf{td}, u_1)$ setting the output to y_1. It sends $(\mathsf{sid}, \mathbf{A}, y_1)$ to C.

2. Upon receiving $(\mathsf{sid}, \mathbf{A}, y)$ from R, C it chooses $\mathbf{r}_1 \leftarrow_{\$} \{0,1\}^k$ and $\mathbf{r}_2 \leftarrow_{\$} \{0,1\}^{k'}$ and two error vectors $\mathbf{e}_1, \mathbf{e}_2 \leftarrow_{\$} \mathfrak{B}^n_{=\omega}$. It chooses a random string t_1 and queries gRO on $(\mathsf{sid}, \mathsf{C}, M, t_1)$ setting the output to $\mathbf{x}_1 \in \{0,1\}^n$. It computes
$$\mathbf{c}_1 = \mathbf{r}_1\mathbf{A} + \mathbf{x}_1 + \mathbf{e}_1.$$

Next, it queries gRO on $(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, M, t_1)$, using the same random string t_1, setting the output to $\mathbf{x}_2 \in \{0,1\}^{k'}$. It computes

$$\mathbf{c}_2 = \mathbf{x}_2\mathbf{A} + \mathbf{r}_2\mathbf{B} + \mathbf{e}_2.$$

It sends $(\mathsf{sid}, \mathbf{c}_1, \mathbf{c}_2)$ to R.

A scheme can be found in Figure 3.

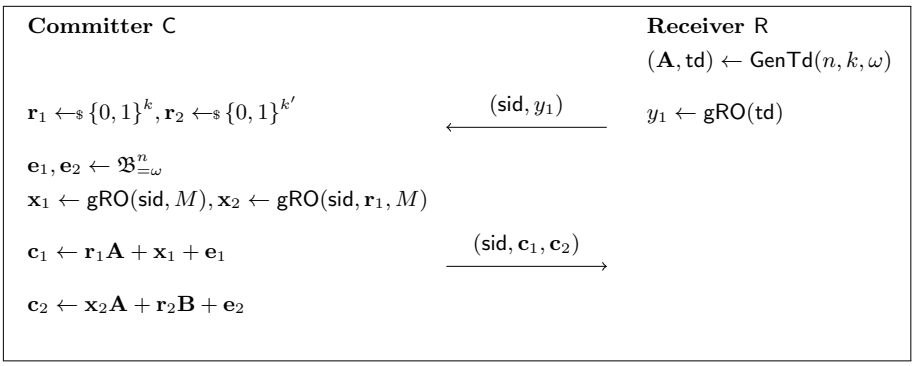

Figure 3: Commitment phase. The nonce t_1 and u_1 are omitted

Opening phase. To open a message M, the parties proceed as follows.

1. R first chooses $ch \leftarrow \mathsf{V}_1^{\varepsilon}$, consistent with the challenges from the $(\mathsf{P}^{\varepsilon}, \mathsf{V}^{\varepsilon})_{\mathsf{xLPN}}$ protocol. It chooses a random string u_2 and queries gRO on $(\mathsf{sid}, \mathsf{R}, ch, u_2)$ setting the output to y_2. It sends (sid, y_2) to C.

2. Upon receiving (sid, y_2) from R, C computes $com \leftarrow \mathsf{P}_1^{\varepsilon}\left((\mathbf{B}, \mathbf{r}_2\mathbf{B} + \mathbf{e}_2), \mathbf{r}_2\right)$, a commitment of $(\mathsf{P}^{\varepsilon}, \mathsf{V}^{\varepsilon})_{\mathsf{xLPN}}$ for public information $(\mathbf{B}, \mathbf{r}_2\mathbf{B} + \mathbf{e}_2)$ and secret information $\mathbf{r}_2$. It chooses a random string t_3 and queries $(\mathsf{sid}, \mathsf{C}, com, \mathbf{r}_2\mathbf{B} + \mathbf{e}_2, t_3)$ to gRO setting the output to x_3. It sends (sid, x_3) to R.

3. Upon receiving (sid, x_3) from C, R reveals td and ch by sending $(\mathsf{sid}, \mathsf{td}, u_1, ch, u_2)$ to C.

4. Upon receiving $(\mathsf{sid}, \mathsf{td}, u_1, ch, u_2)$ from R, C checks that the opening td, t_1 is consistent with y_1 and that ch, t_2 is consistent with y_2. It does this by querying gRO on $(\mathsf{sid}, \mathsf{R}, \mathsf{td}, u_1)$ and on $(\mathsf{sid}, \mathsf{R}, ch, u_2)$, setting the output to y_1' and to y_2' (respectively) and by checking if $y_1 = y_1'$ and if $y_2 = y_2'$. If any of these tests fails, C aborts the protocol. Otherwise, it computes $resp \leftarrow \mathsf{P}_2^{\varepsilon}\left((\mathbf{B}, \mathbf{r}_2\mathbf{B} + \mathbf{e}_2), \mathbf{r}_2; com, ch\right)$ and sends $(\mathsf{sid}, \mathbf{r}_1, M, t_1, t_3, com, resp)$ to R.

   Upon receiving $(\mathsf{sid}, \mathbf{r}_1, M, t_1, t_3, com, resp)$ from C, R sets $T = (com, ch, resp)$. It queries gRO on $(\mathsf{sid}, \mathsf{C}, M, t_1)$ and on $(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, M, t_1)$ and sets the output to $\mathbf{x}_1'$ and to $\mathbf{x}_2'$ respectively. It also queries gRO on $(\mathsf{sid}, com, \mathbf{c}_2 -$

$\mathbf{x}'_2\mathbf{A}, t_3)$ and sets the output to x'_3. It checks if $x_3 = x'_3$, if $w(\mathbf{c}_1 - (\mathbf{r}_1\mathbf{A} + \mathbf{x}'_1)) \leq \omega$ and if $\mathsf{V}_2^{\varepsilon}(\mathbf{c}_2 - \mathbf{x}'_2\mathbf{A}, T) = 1$. If any of these tests fail, R outputs 0. Else, it accepts the opening and outputs 1.

A scheme of the opening phase is presented in Figure 4.

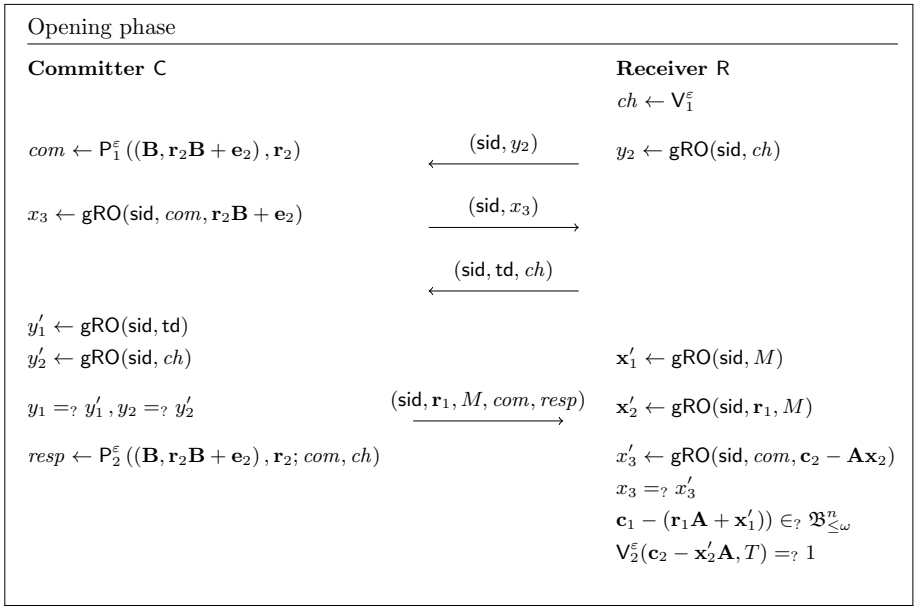

Figure 4: Opening phase. The nonce t_1, t_3, u_1 and u_2 are omitted.

4 Security

In this section, we prove the security of the scheme. We also present the simulation required to guarantee universal composability.

4.1 Committer's and receiver's privacy

Let α be the size of the random string t_1.

Theorem 8. *Suppose that $\alpha = \lambda + n$. The commitment scheme is perfectly hiding and computationally binding in the* gRO*-model, given that the* $\mathsf{SD}_{2\omega}$ *and* GD *assumptions hold.*

Proof. We begin by proving the hiding property. Then we prove the binding property for our scheme.

Perfectly hiding property. The commitment corresponds to $\mathbf{c}_1 = \mathbf{r}_1\mathbf{A} + \mathbf{x}_1 + \mathbf{e}_1$ and $\mathbf{c}_2 = \mathbf{x}_2\mathbf{A} + \mathbf{r}_2\mathbf{B} + \mathbf{e}_2$, where $\mathbf{x}_1 \leftarrow \mathsf{gRO}(\mathsf{sid}, M, t_1)$ and $\mathbf{x}_2 \leftarrow \mathsf{gRO}(\mathsf{sid}, \mathbf{r}_1, M, t_2)$. This means that the message is hidden by the random oracle.

We prove that the probability of an adversary to find M is negligible, even when it has unlimited computational power. To this end, we prove that, given an output y of gRO, the probability that the output of a query of the form $(\mathsf{sid}, \mathsf{C}, M, t_1)$ is y is negligible for every message M and random t_1, if the size of t_1 is equal to the size of M plus the size of y.

Let $\mathcal{S}_y$ be the random variable that corresponds to the number of fixed size x such that $y \leftarrow \mathsf{gRO}(x)$, where x is of the form (sid, M, t) where M is a fixed message of size λ, t has size α and y has size n. We have that $\mathcal{S}_y = \sum_t \mathrm{Ber}_t(2^n)$ where Ber denotes a Bernoulli distribution for each t. Hence, the expected value of $\mathcal{S}_y$ is $\mathbb{E}(\mathcal{S}_y) = 2^\alpha 2^{-n}$.

By the Chernoff-Hoeffding inequality (and considering the relative distance), we have

$$\mathsf{Pr}\left[|\mathcal{S}_y/\mathbb{E}(\mathcal{S}_y) - 1| \geq \delta\right] \leq \gamma e^{-2^\alpha.\delta^2}$$

where γ is some constant. Consider δ to be $2^{-\alpha/4}$, so that the distance between $\mathcal{S}_y$ and $\mathbb{E}(\mathcal{S}_y)$ is negligible.

Taking the union bound for every y and for every M, we get

$$\mathsf{Pr}\left[\exists M \exists y : |\mathcal{S}_y/\mathbb{E}(\mathcal{S}_y) - 1| \geq 2^{-\alpha/4}\right] \leq \gamma 2^{\lambda+n} e^{-2^\alpha.\delta^2}.$$

By hypothesis, consider $\alpha = \lambda + n$. We conclude that the relative distance between $\mathcal{S}_y$ and $\mathbb{E}(\mathcal{S}_y)$ is at most $1/2^{(\lambda+\beta)/4}$, except with negligible probability.

Binding property. Suppose that C is able to find $\mathbf{c}_1$ and $\mathbf{c}_2$, given $\mathbf{A}$ and y_1, such that it is able to open different messages. That is, C is able to interact with R following the opening phase of the protocol such that R accepts the opening for two different messages. This situation is sketched below:

1. R sends y_2;	1. R sends y_2;
2. C sends x_3;	2. C sends x'_3;
3. R sends $\mathsf{td}, ch, u_1, u_2$;	3. R sends $\mathsf{td}, ch, u_1, u_2$;
4. C sends com, $resp$,$\mathbf{r}_1$, M, s, t_1, t_3.	4. C sends com', $resp'$, $\mathbf{r}'_1$, M', s', t'_1, t'_3.

Since R accepts both openings as valid ones, they must fulfill the following conditions:

1. $w(\mathbf{c}_1 + \mathbf{r}_1\mathbf{A} + \mathbf{x}_1) \leq \omega$, $T = (com, ch, resp)$ is a valid transcript for public information $(\mathbf{B}, \mathbf{c}_2 + \mathbf{x}_2)$ and $x_3 = \mathsf{gRO}(\mathsf{sid}, \mathbf{c}_2 + \mathbf{x}_2, com)$;
2. $w(\mathbf{c}_1 + \mathbf{r}'_1\mathbf{A} + \mathbf{x}'_1) \leq \omega$, $T' = (com', ch, resp')$ is a valid transcript for public information $(\mathbf{B}, \mathbf{c}_2 + \mathbf{x}'_2)$ and $x'_3 = \mathsf{gRO}(\mathsf{sid}, \mathbf{c}_2 + \mathbf{x}'_2, com')$

where $\mathbf{x}_1 = \mathsf{gRO}(\mathsf{sid}, M, t_1)$, $\mathbf{x}_2 = \mathsf{gRO}(\mathsf{sid}, \mathbf{r}_1, M, t_1)$, $\mathbf{x}'_1 = \mathsf{gRO}(\mathsf{sid}, M', t'_1)$ and $\mathbf{x}_2 = \mathsf{gRO}(\mathsf{sid}, \mathbf{r}'_1, M', t'_1)$.

Therefore, we have that

$$(\mathbf{r}_1 + \mathbf{r}'_1)\mathbf{A} + (\mathbf{e}_1 + \mathbf{e}'_1) = \mathbf{x}_1 + \mathbf{x}'_1.$$

Let $\mathbf{r}_1 + \mathbf{r}'_1 = \tilde{\mathbf{r}}$, $\mathbf{e}_1 + \mathbf{e}'_1 = \tilde{\mathbf{e}}$ and $\mathbf{x}_1 + \mathbf{x}'_1 = \tilde{\mathbf{x}}$. Rewriting the equation, we have

$$\tilde{\mathbf{r}}\mathbf{A} + \tilde{\mathbf{e}} = \tilde{\mathbf{x}},$$

where $\tilde{\mathbf{x}}$ is chosen uniformly at random from $\{0,1\}^n$ and $w(\tilde{\mathbf{e}}) \leq 2\omega$. By the $\mathsf{SD}_{2\omega}$, the probability that the receiver finds $\tilde{\mathbf{r}}$ and $\tilde{\mathbf{e}}$ for a random $\tilde{\mathbf{x}}$ is negligible. Else, the receiver would be able to solve the $\mathsf{SD}_{2\omega}$ for random instances.

Also, by the soundness of the $(\mathsf{P}^{\varepsilon}, \mathsf{V}^{\varepsilon})_{\mathsf{xLPN}}$ protocol, we have

$$\mathbf{x}_2\mathbf{A} + \mathbf{r}_2\mathbf{B} + \mathbf{e}_2 = \mathbf{x}'_2\mathbf{A} + \mathbf{r}'_2\mathbf{B} + \mathbf{e}'_2$$

where $\mathbf{x}_2 \leftarrow \mathsf{gRO}(\mathbf{r}_1, M, t_1)$, $\mathbf{x}'_2 \leftarrow \mathsf{gRO}(\mathbf{r}'_1, M', t'_1)$ and $\mathbf{e}_2, \mathbf{e}'_2 \in \mathfrak{B}^n_{=\omega}$, except with negligible probability. This means that the word

$$(\mathbf{A}|\mathbf{B}) \begin{pmatrix} \mathbf{x}_2 + \mathbf{x}'_2 \\ \mathbf{r}_2 + \mathbf{r}'_2 \end{pmatrix}$$

has Hamming weight less or equal than 2ω. Let $\mathcal{C}_{\mathbf{A}}$ be the code defined by the generating matrix $\mathbf{A}$ and $\mathcal{C}_{\mathbf{B}}$ the code defined by $\mathbf{B}$. By definition of $\mathcal{C}_{\mathbf{A}}$, every $\mathbf{c} \in \mathcal{C}_{\mathbf{A}}$ has Hamming weight greater than 2ω, otherwise we could not be able to some words of the form $\mathbf{y} = \mathbf{c} + \tilde{\mathbf{e}}$ where $\mathbf{c} \in \mathcal{C}_{\mathbf{A}}$ and $\tilde{\mathbf{e}} \in \mathfrak{B}^n_{\omega}$. Moreover, if $\mathbf{d} \in \mathcal{C}_{\mathbf{B}}$ then $\mathbf{d}$ has Hamming weight greater than 2ω, except with negligible probability, by the Gilbert-Varshamov bound [MS77]. We conclude that $\begin{pmatrix} \mathbf{x}_2 + \mathbf{x}'_2 \\ \mathbf{r}_2 + \mathbf{r}'_2 \end{pmatrix} = 0$, except with negligible probability. Therefore, $\mathbf{x}_2 = \mathbf{x}'_2$ and $\mathbf{r}_2 = \mathbf{r}'_2$, except with negligible probability. So, if it is infeasible to find collisions for gRO, then it is infeasible to open two different messages in the opening phase. □

4.2 UC-security

In this section, we prove that the scheme is secure in the UC-framework.

Theorem 9. *The commitment scheme presented securely UC-realizes $\mathcal{F}_{\text{com}}$ against static malicious adversaries in the $\mathcal{G}_{\mathsf{gRO}}$-hybrid model, given that the $\mathsf{SD}_{2\omega}$ and GD assumptions hold.*

Proof. Let $\mathcal{A}$ be an adversary in the real-world, S be an adversary in the ideal-world, called the simulator, and $\mathcal{E}$ be any environment. As usual, the communication between $\mathcal{A}$ and $\mathcal{E}$ is simulated by S by forwarding messages from one party to the other.

The trivial cases where the adversary corrupts both parties and when it does not corrupt any party can be simulated by S. When $\mathcal{A}$ corrupts both parties,

S just runs $\mathcal{A}$ internally and let it generate all the messages. When $\mathcal{A}$ does not corrupt any party, S generates all the messages from C and R, following the protocol and forwarding every message to $\mathcal{A}$.

We now show how to construct the simulator when $\mathcal{A}$ corrupts only one of the parties.

Simulation when only the committer C is corrupted by $\mathcal{A}$. Here, S controls R and tries to extract C's input.

> **Commitment phase.** The simulator S follows the protocol honestly by generating $(\mathbf{A}, \mathsf{td}) \leftarrow \mathsf{GenTd}(n, m, \omega)$ and sending $(\mathsf{sid}, \mathbf{A}, y_1)$ to C where y_1 is the output of gRO when queried on $(\mathsf{sid}, \mathsf{td}, u_1)$.
>
> Let $\mathbb{Q}_{|\mathsf{sid}}$ be the list of queries to gRO by any party with sid that does not pertain that session ID. Upon receiving $(\mathsf{sid}, \mathbf{c}_1, \mathbf{c}_2)$, S checks if there are two queries of the form $(\mathsf{sid}, \mathsf{C}, M, t_1)$ (with output $\mathbf{x}_1$) and $(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, M, t_1)$ such that $w(\mathbf{c}_1 + \mathbf{r}_1\mathbf{A} + \mathbf{x}_1) \leq \omega$. If there is such a query and it is unique, S sets $M' = M$. Else, it sets $M' = 0^\lambda$.
>
> The simulator S sends the message $(\mathsf{sid}, \mathsf{commit}, \mathsf{C}, \mathsf{R}, M)$ to $\mathcal{F}_{\text{tcom}}$.
>
> **Opening phase.** The simulator S follows the protocol honestly, controlling R. If S accepts the opening of M^* after interacting with C, S checks if $M^* = M$. If so, it sends the message $(\mathsf{sid}, \mathsf{reveal}, \mathsf{C}, \mathsf{R})$ to $\mathcal{F}_{\text{tcom}}$. Else, it aborts the execution.

The ideal-world and real-world execution differ only when S aborts when it should not abort. This happens only if C is able to open a message M^* different from the message M that S sends to $\mathcal{F}_{\text{tcom}}$. This may happen when:

1. There are no queries in $\mathbb{Q}_{|\mathsf{sid}}$ of the form $(\mathsf{sid}, \mathsf{C}, M, t_1)$ and $(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, M, t_1)$ that fulfill $w(\mathbf{c}_1 + \mathbf{r}_1\mathbf{A} + \mathbf{x}_1) \leq \omega$, but S accepts the opening of C;
2. There are several queries in $\mathbb{Q}_{|\mathsf{sid}}$ of the form $(\mathsf{sid}, \mathsf{C}, M, t_1)$ and $(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, M, t_1)$ that fulfill $w(\mathbf{c}_1 + \mathbf{r}_1\mathbf{A} + \mathbf{x}_1) \leq \omega$, but S accepts the opening of C;
3. There are such queries and they are unique, but C opens a different M^* than M.

The probability of the event in 1 is negligible, since C would have to find an opening that fulfill $w(\mathbf{c}_1 + \mathbf{r}_1\mathbf{A} + \mathbf{x}_1) \leq \omega$. As we have seen in the proof of Lemma 8, this happens with negligible probability for PPT algorithms. The probability of the event in 2 is also negligible, by the proof of the previous theorem. The third event happens with negligible probability by the binding property of the scheme.

We conclude that both executions are indistinguishable from the point-of-view of $\mathcal{E}$, except with negligible probability. Hence, the protocol is secure in the UC-framework against a corrupted C.

Simulation when only the receiver R is corrupted by $\mathcal{A}$. Here, S controls C and commits to a message (for example 0^λ). After receiving a message from the ideal functionality, S has to open the received message to R. This process is usually called equivocation.

> **Commitment phase.** Upon receiving a message $(\mathsf{sid}, \mathsf{receipt}, \mathsf{C}, \mathsf{R})$ from $\mathcal{F}_{\text{tcom}}$, S follows the protocol honestly committing to the message 0^λ, for example. That is, upon receiving $\mathbf{A}$ and y_1 from R, it computes $\mathbf{c}_1 = \mathbf{r}_1\mathbf{A} + \mathbf{x}_1 + \mathbf{e}_1$ and $\mathbf{c}_2 = \mathbf{x}_2\mathbf{A} + \mathbf{r}_2\mathbf{B} + \mathbf{e}_2$, where $\mathbf{x}_1 \leftarrow \mathsf{gRO}(\mathsf{sid}, \mathsf{C}, 0^\lambda, t_1)$ and $\mathbf{x}_2 \leftarrow \mathsf{gRO}(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1, 0^\lambda, t_1)$, and sends $(\mathsf{sid}, \mathbf{c}_1, \mathbf{c}_2)$ to R.
>
> **Opening phase.** Upon receiving a message $(\mathsf{sid}, \mathsf{reveal}, \mathsf{C}, \mathsf{R}, M)$ from $\mathcal{F}_{\text{tcom}}$, S asks the list $\mathbb{Q}_{|\mathsf{sid}}$ from $\mathcal{F}_{\text{tcom}}$ to gRO. It checks if there is a query whose output is equal to y_1. If there is such a query, let us say $(\mathsf{sid}, \mathsf{R}, \mathsf{td}, u_1)$, and if td is a trapdoor for $\mathbf{A}$, it sets $\mathsf{td}' = \mathsf{td}$. Else, it sets $\mathsf{td}' = \perp$.
>
> If $\mathsf{td}' \neq \perp$, it sets $\mathbf{x}_1 \leftarrow \mathsf{gRO}(\mathsf{sid}, \mathsf{C}, M, t_1')$, where t_1' is a random string, and finds $\mathbf{r}_1'$ such that $\mathbf{r}_1'\mathbf{A} + \mathbf{e}_1' = \mathbf{c}_1 + \mathbf{x}_1'$ using td'. If this fails, it chooses another random string t_1' and tries again. it repeats the process until it is successful. Observe that this task can be done in polynomial time by Corollary 6.
>
> Upon receiving (sid, y_2) from R, S checks if there is a query of the form $(\mathsf{sid}, \mathsf{R}, w, u_2)$ in $\mathbb{Q}_{|\mathsf{sid}}$ such that its output is y_2. If there is, it sets $ch' = w$, else it chooses $ch' \leftarrow \mathsf{V}_1^\varepsilon$. It computes com for public information $(\mathbf{B}, \tilde{\mathbf{c}}_2)$ where $\tilde{\mathbf{c}}_2 = \mathbf{c}_2 + \mathbf{x}_2\mathbf{A}$ and $\mathbf{x}_2 \leftarrow \mathsf{gRO}(\mathsf{sid}, \mathsf{C}, \mathbf{r}_1', M, t_1')$ such that it can create valid transcripts (with challenge ch'), even without knowing the secret. Observe that, since the cheating probability of a dishonest prover in $(\mathsf{P}^\varepsilon, \mathsf{V}^\varepsilon)_{\mathsf{xLPN}}$ is $2/3$, then S can always find such com. It queries gRO on $x_3 \leftarrow (\mathsf{sid}, \mathsf{C}, \tilde{\mathbf{c}}_2, com)$. If S was not able to find $\mathbf{r}_1'$ (because $\mathsf{td}' = \perp$) then, it chooses x_3 at random. It sends (sid, x_3) to R.
>
> Upon receiving $(\mathsf{sid}, \mathsf{td}, u_1, ch, u_2)$ from R, S checks if $ch = ch'$. If not, then it aborts the protocol. If $\mathsf{td}' \neq \perp$, S follows the protocol as the honest C would do from this point on. Else if $\mathsf{td}' = \perp$, it chooses random values for $\mathbf{r}_1, t_1, t_3$ and sends them (together with $M, com, resp$) as the opening.

The real-world and the ideal-world executions differ only when:

1. S is not able to extract a valid trapdoor for matrix $\mathbf{A}$ and R sends td such that it is a valid trapdoor for $\mathbf{A}$ and its output by gRO is y_1. In this case, S has a negligible probability of completing the protocol with a valid opening for M;
2. S aborts when $ch \neq ch'$.

The probability of S accepting the opening of td as a valid one is negligible since it is the probability of R finding collisions for gRO. Hence, S (controlling C) will abort the execution before sending the last message (corresponding to

the opening of $\mathbf{r}_1$, M, t_1, t_3, *com* and *resp*), except with negligible probability. Moreover, the probability of the event 2 happening is also negligible, by the gRO assumption.

We conclude that both executions are indistinguishable from the point-of-view of $\mathcal{E}$, except with negligible probability. Hence, the protocol is secure in the UC-framework against a corrupted R. □

5 Conclusion

We proposed the first ever post-quantum UC-commitment scheme in the global random oracle. The security of our proposal is based on the Syndrome Decoding and on the Goppa Distinguisher assumptions, two well-established assumptions in post-quantum cryptography.

Our proposal is proven to be perfectly hiding, meaning that even an all-powerful receiver is not able to find the message that the committer is committing to, before the opening phase, and (computationally) binding.

The scheme is proven secure in the UC-framework against static malicious adversaries. However, it seems that the scheme cannot be proven secure against adaptive malicious adversaries, that is, adversaries that may corrupt parties after the beginning of the execution of the scheme. To see this, note that the simulator has access only to adversarial queries made to gRO. However, if an adversary is allowed to corrupt after the beginning, the simulator may not have enough information to perform the simulation. More precisely, the necessary queries to perform the simulation can be made by a honest party and, thus, the simulator does not have access to them.

We leave as future work to develop a post-quantum UC-commitment scheme secure against adaptive malicious adversaries.

Acknowledgment

The author thanks João Ribeiro for insightful discussions and comments on a early draft of this work. The author thanks the support from DP-PMI and FCT (Portugal) through the grant PD/BD/135181/2017.

References

[BDD+17] Paulo S. L. M. Barreto, Bernardo David, Rafael Dowsley, Kirill Morozov, and Anderson C. A. Nascimento. A framework for efficient adaptively secure composable oblivious transfer in the rom. Cryptology ePrint Archive, Report 2017/993, 2017. `https://eprint.iacr.org/2017/993`.

[BDGM18] Pedro Branco, Jintai Ding, Manuel Goulão, and Paulo Mateus. Universally composable oblivious transfer protocol based on the

RLWE assumption. Cryptology ePrint Archive, Report 2018/1155, 2018. https://eprint.iacr.org/2018/1155.

[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.

[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.

[BPRS17] Megha Byali, Arpita Patra, Divya Ravi, and Pratik Sarkar. Fast and universally-composable oblivious transfer and commitment scheme with adaptive security. Cryptology ePrint Archive, Report 2017/1165, 2017. https://eprint.iacr.org/2017/1165.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, pages 62–73, New York, NY, USA, 1993. ACM.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 136–, Washington, DC, USA, 2001. IEEE Computer Society.

[CDG+18] Jan Camenisch, Manu Drijvers, Tommaso Gagliardoni, Anja Lehmann, and Gregory Neven. The wonderful world of global random oracles. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 280–312, Cham, 2018. Springer International Publishing.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *Advances in Cryptology — CRYPTO 2001*, pages 19–40, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[CFS01] Nicolas T. Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 157–174, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[CJS14] Ran Canetti, Abhishek Jain, and Alessandra Scafuro. Practical UC security with a global random oracle. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 597–608, New York, NY, USA, 2014. ACM.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 494–503, New York, NY, USA, 2002. ACM.

[DAST18] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new code-based signature scheme. Cryptology ePrint Archive, Report 2018/996, 2018. `https://eprint.iacr.org/2018/996`.

[DG03] Ivan Damgard and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 426–437, New York, NY, USA, 2003. ACM.

[DN02] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 581–596, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[EKM17] Andre Esser, Robert Kübler, and Alexander May. LPN decoded. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 486–514, Cham, 2017. Springer International Publishing.

[FGUO+11] J. Faugre, V. Gauthier-Uman, A. Otmani, L. Perret, and J. Tillich. A distinguisher for high rate McEliece cryptosystems. In *2011 IEEE Information Theory Workshop*, pages 282–286, Oct 2011.

[FLM11] Marc Fischlin, Benoît Libert, and Mark Manulis. Non-interactive and re-usable universally composable string commitments with adaptive security. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 468–485, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Fuj16] Eiichiro Fujisaki. Improving practical UC-secure commitments based on the DDH assumption. In Vassilis Zikas and Roberto De Prisco, editors, *Security and Cryptography for Networks*, pages 257–272, Cham, 2016. Springer International Publishing.

[GIKW14] Juan A. Garay, Yuval Ishai, Ranjit Kumaresan, and Hoeteck Wee. On the complexity of UC commitments. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 677–694, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[HMQ04] Dennis Hofheinz and Jörn Müller-Quade. Universally composable commitments using random oracles. In Moni Naor, editor, *Theory of Cryptography*, pages 58–76, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 663–680, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Kil88] Joe Kilian. Founding cryptography on oblivious transfer. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 20–31, New York, NY, USA, 1988. ACM.

[Kir18] Elena Kirshanova. Improved quantum information set decoding. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography*, pages 507–527, Cham, 2018. Springer International Publishing.

[Lin11] Yehuda Lindell. Highly-efficient universally-composable commitments based on the DDH assumption. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 446–466, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[LS01] P. Loidreau and N. Sendrier. Weak keys in the McEliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, March 2001.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report*, 44, 05 1978.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Elsevier, 1977.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 554–571, Berlin, Heidelberg, 2008. 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.

[XXW13] Xiang Xie, Rui Xue, and Minqian Wang. Zero knowledge proofs from ring-LWE. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *Cryptology and Network Security*, pages 57–73, Cham, 2013. Springer International Publishing.