A Gapless Code-Based Hash Proof System based on RQC and its Applications

Slim Bettaieb[1], Loïc Bidoux[1], Olivier Blazy[2], Yann Connan[1,2,], and ✉ Philippe Gaborit[2]

[1] Worldline, 23 rue de la Pointe, 59139 Noyelles-les-Seclin, France
slim.bettaieb@worldline.com
loic.bidoux@worldline.com
yann.connan@worldline.com
[2] University of Limoges, 123 Avenue Albert Thomas, 87000 Limoges, France
olivier.blazy@unilim.fr
philippe.gaborit@unilim.fr

Abstract. Cramer and Shoup introduced at Eurocrypt'02 the concept of hash proof system, also designated as smooth projective hash functions. Since then, they have found several applications, from building CCA-2 encryption as they were initially created for, to being at the core of several authenticated key exchange or even allowing witness encryption. In the post-quantum setting, the very few candidates use a language based on ciphertexts to build their hash proof system. This choice seems to inherently introduce a gap, as some elements outside the language could not be distinguish from those in the language. This creates a lawless zone, where an adversary can possibly mount an undetectable attack, particularly problematic when trying to prove security in the UC framework [19]. We show that this gap could be completely withdrawn using code-based cryptography. Starting from RQC [4], a candidate selected for the second round of the National Institute of Standards and Technology (NIST) post-quantum cryptography standardization project, we show how to build such a hash proof system from code-based cryptography and present a way, based on a proof of knowledge, to fully negate the gap. We propose two applications of our construction, a witness encryption scheme and a password authenticated key exchange (PAKE).

Keywords: Code-Based Cryptography · Hash Proof System · Rank Quasi-Cyclic Scheme · Witness Encryption · Password Authenticated Key Exchange.

1 Introduction

Post-quantum cryptography is starting to really bloom, as we see the emergence of various cryptographic primitives, the standard ones such as signatures and encryptions, but also advanced ones like oblivious transfer or authenticated key exchange based on all sorts of assumptions (lattices, error correcting codes, and in some cases super singular isogenies).

There nevertheless remains one primitive for which post-quantum cryptography is still a long way behind: Hash Proof Systems (HPS) also designated as smooth projective hash functions [21]. Those functions were introduced originally to provide a CCA-2 secure encryption scheme. Since then, they have been used in many interactive protocols as the key building block, in particular for authenticated key exchange [29, 2, 33, 34], and oblivious transfers [32, 30, 18]. In addition, they can also be used to produce witness encryption schemes or zero-knowledge arguments. While the initial construction in group settings works pretty well, constructions in the post-quantum settings seem painful. Indeed, in this later setting, languages used for the construction of the HPS are languages of ciphertexts of a given plaintext μ. Those allow to implicitly prove that a specific value is encrypted but, on the other hand, seems to inherently introduce a gap between honestly generated ciphertexts of a given plaintext μ considered for the correctness property and those, who can not be distinguished from honest ciphertexts in some cases, who simply decipher into μ without having the good shape of a ciphertext of μ.

To the best of our knowledge, there are only a few constructions of HPS in the post-quantum realm. First, a construction over a lattice-based encryption scheme in the standard model, proposed by Katz and Vaikuntanathan [33]. However, the language used for their HPS was not simply defined as the set of valid standard LWE ciphertexts, leading to a decryption procedure very costly, as pointed in [15]. To solve this issue, Benhamouda and al proposed a new HPS over a lattice-based encryption scheme, still in the standard model, using the standard language of ciphertexts and based on the learning with errors assumption.

There is also a subsequent work by Zhang and Yu who proposed an interesting new lattice-based HPS in [43]. While it uses zero-knowledge proof to supplement the HPS like our technique, it still suffers from a gap between the language $\mathcal{L}$ of valid encryptions of a message μ used for the (approximate) correctness, and the language $\mathcal{L}^*$ of elements that decrypts to μ for smoothness.

In code-based cryptography, a candidate was evoked in [39], however some weaknesses in the technique does not make it a suitable candidate. The author does not show smoothness but universality. In this paper, universality is properly defined but the proof does not follow. Instead of proving it for every possible keys, it is done for random ones, which means adversarially chosen keys can, and in this case will, make the proof fail. In addition, while there exists generic transformations from universal projective hash functions to word-dependent (also known as GL/CS) hash proof systems, they do not allow to achieve word-independent (KV) hash proof systems. Hence, building a word-independent HPS in code-based cryptography still remained an open-problem.

To the best of our knowledge, in the post-quantum setting the conception of a gapless hash proof system has never been achieved.

1.1 On the Necessity of a Gapless Construction

When proving the security of a HPS, one has to distinguish between distinct languages. The correctness considers correctly generated ciphertexts of a mes-

sage μ. On the contrary, the smoothness property consider all the elements that are not valid ciphertexts. The problem hidden behind this segmentation of the ambiant space is that it contains some problematic elements that are not valid ciphertexts of a given message μ but still decrypt into μ. This gap leaves a huge grey area, where an adversary can maliciously generate such malformed ciphertexts elements, inconspicuous for an honest verifier, and potentially open the door for practical attacks in some context.

Moreover, such a gap is not possible when trying to prove security in the UC framework. As a toy example, imagine an HPS-based PAKE protocol, where users derive a shared key from a hash proof system over their respective commitments to a password. In the proof, one needs to build a simulator abstracting the ideal functionality. As of now, post-quantum schemes based on HPS need to weaken the functionality. The simulator cannot detect whether the adversary sent a valid encryption of the password or just something that (for the particular secret key) can be decrypted back to the password. In the first case, the protocol should always succeed (assuming perfect correctness) while in the latter it is not clear. Either this is an admissible behavior, and so the protocol should always succeed, however this does not happen with existing post-quantum HPS, or this is an inadmissible behavior and the protocol should fail, however the simulator being unable to detect this situation has to flag the authentication as successful and so does not fulfill the functionality. As such, for proper UC instantiations, obtaining a gapless HPS is a major issue that needed to be solved.

1.2 Contribution

Our main contribution consists in proposing a word-independent KV-HPS from an existing code-based encryption scheme and proving its security in a quantum-resistant setting. We propose a first application in the standard model relying on this HPS, namely a witness encryption scheme. We then switch back to the random oracle model, and propose a PAKE secure in the BPR model [11]. To do so, we design a zero-knowledge proof of knowledge asserting if two different ciphertexts of the same message μ are valid. It should be noted that, in all these constructions, we manage to avoid the main caveat of post-quantum constructions, as we propose gapless protocols. To handle the gap, we define our HPS only on the set of valid ciphertexts and check if necessary whether the word is a valid ciphertext by using a proof of knowledge.

Our contribution relies on the RQC candidate of the NIST post-quantum competition. This is one of the rank-metric based encryption schemes that has advanced to the round two of the competition. We view as an important challenge the capacity to build post-quantum HPS without a gap, as this would allow to avoid the duality in languages and would close the door to some possible practical attacks that remain undetectable by a simulator.

1.3 Road Map

In section 2, we give some preliminaries on code-based cryptography, present the RQC encryption scheme and give an overview of hash proof systems. Next, in section 3, we present our code-based HPS in the rank metric setting. Its security is presented in section 4. We describe the zero-knowledge proof of ciphertext validity for RQC in section 5. Then, we detail the witness encryption and PAKE constructed from our HPS in sections 6.1 and 6.2 respectively. Some concrete parameters for these constructions are given in section 7.

2 Preliminaries

In this section, we give some definitions regarding code-based cryptography and the rank metric, present the RQC cryptosystem and give an overview of HPS.

2.1 Code-based Cryptography

Throughout this paper, let q be a power of a prime p. The finite field with q elements is denoted by $\mathbb{F}_q$ and more generally, for any natural number m, the finite field with q^m elements will be denoted by $\mathbb{F}_{q^m}$. Let $\mathcal{B} = (\beta_1, \ldots, \beta_m)$ denotes a basis of $\mathbb{F}_{q^m}$ viewed as a m-dimensional vector space over $\mathbb{F}_q$.

Let $\mathcal{V}$ be a n-dimensional vector space over $\mathbb{F}_{q^m}$. Let $P \in \mathbb{F}_q[X]$ a polynomial of degree n. The vector space $\mathcal{V}$ can be identified with the ring $\mathbb{F}_{q^m}[X]/\langle P \rangle$ where $\langle P \rangle$ denotes the ideal of $\mathbb{F}_{q^m}[X]$ generated by P.

$$\begin{array}{rccc} \Psi: & \mathbb{F}_{q^m}^n & \simeq & \mathbb{F}_{q^m}[X]/\langle P \rangle \\ & \mathbf{x} = (x_0, \ldots, x_{n-1}) & \mapsto & \Psi(\mathbf{x}) = \displaystyle\sum_{i=0}^{n-1} x_i X^i \end{array}$$

By isomorphism, elements of $\mathcal{V}$ can be viewed as polynomials in $\mathbb{F}_{q^m}[X]/\langle P \rangle$. For all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, the product $\mathbf{x} \cdot \mathbf{y}$ is defined as the polynomial multiplication of $\mathbf{x}$ and $\mathbf{y}$ performed modulo P. Formally, $\mathbf{x} \cdot \mathbf{y} \stackrel{\text{def}}{=} \Psi^{-1}[\Psi(\mathbf{x}) \cdot \Psi(\mathbf{y})]$. For a better readibility, we will sometimes omit the symbol Ψ and refer either to the vector or the polynomial form depending on the context.

Rank Metric To any vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{V}$, one can associate the matrix $\mathbf{M_x} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ by expressing its coordinates in the basis $\mathcal{B}$ where $\mathbf{M_x} = (x_{ij})$ such that for all $j \in [\![1, n]\!]$, $x_j = \sum_{i=1}^{m} x_{i,j}\beta_i$.

$$\begin{array}{rccc} \mathbf{M}: & \mathbb{F}_{q^m}^n & \simeq & \mathcal{M}_{m,n}(\mathbb{F}_q) \\ & \mathbf{x} = (x_0, \ldots, x_{n-1}) & \mapsto & \mathbf{M_x} = \begin{pmatrix} x_{1,0} & \cdots & x_{1,n-1} \\ x_{2,0} & \cdots & x_{2,n-1} \\ \vdots & & \vdots \\ x_{m,0} & \cdots & x_{m,n-1} \end{pmatrix} \begin{matrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{matrix} \end{array}$$

Definition 1 (Support). *Let* $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{V}$*. The support of* $\mathbf{x}$*, denoted* $\mathsf{Supp}(\mathbf{x})$*, is the* $\mathbb{F}_q$*-subspace of* $\mathbb{F}_{q^m}$ *generated by the coordinates of* $\mathbf{x}$*, namely* $\mathsf{Supp}(\mathbf{x}) = \langle x_1, \ldots, x_n \rangle_{\mathbb{F}_q}$.

Definition 2 (Rank weight). *Let* $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{V}$*. The rank weight of* $\mathbf{x}$*, denoted* $\omega(\mathbf{x})$*, is equal to the dimension of its support.*

Corollary 1. *The rank of a vector is equal to the rank of its associated matrix.*

Definition 3 (Rank distance). *Let* $\mathbf{x}, \mathbf{y} \in \mathcal{V}$*. Let* d *denotes the application defined as* $d(\mathbf{x}, \mathbf{y}) = \omega(\mathbf{x} - \mathbf{y})$.

Proposition 1. *The application* d *previously defined is a distance over* $\mathcal{V}$.

Notation 1 *As we will often manipulate vectors sharing a common support, we introduce the following notations:*

- $\mathcal{S}_w(\mathcal{V}) = \{\mathbf{x} \in \mathcal{V} \mid \omega(\mathbf{x}) = w\}$ *(Spheres of radius* w*)*
- $\mathfrak{S}_w^n(\mathcal{V}) = \left\{ (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathcal{V}^n \;\middle|\; \begin{array}{l} \exists E \subset \mathcal{V}^n \text{ with } \dim(E) = w \\ \text{such that } \forall i \in [\![1, n]\!],\ \mathsf{Supp}(\mathbf{x}_i) = E \end{array} \right\}$

Coding Theory In this part, we recall some definitions and properties regarding coding theory and introduce ideal codes, Gabidulin codes and LRPC codes.

Definition 4 ($\mathbb{F}_{q^m}$-linear code). *An* $\mathbb{F}_{q^m}$*-linear code* $\mathcal{C}$ *of dimension* k *and length* n *is a subspace of dimension* k *of* $\mathbb{F}_{q^m}^n$ *and is denoted by* $[n, k]_{q^m}$*.* $\mathcal{C}$ *can be represented by two equivalent ways*

- *Using a generator matrix* $\mathbf{G} \in \mathcal{M}_{k,n}(\mathbb{F}_{q^m})$*. Rows of* $\mathbf{G}$ *form a basis of* $\mathcal{C}$*.*

$$\mathcal{C} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{F}_{q^m}^k\}$$

- *Using a parity-check matrix* $\mathbf{H} \in \mathcal{M}_{n-k,n}(\mathbb{F}_{q^m})$*. Rows of* $\mathbf{H}$ *determine a system of equations verified by the elements of* $\mathcal{C}$*.*

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_{q^m}^n \mid \mathbf{H}\mathbf{x}^\top = 0\}$$

A generator matrix $\mathbf{G}$ *(respectively parity-check matrix* $\mathbf{H}$*) is said to be under systematic form if and only if it is of the form* $(\mathbf{I}_k \mid \mathbf{A})$ *(respectively* $(\mathbf{I}_{n-k} \mid \mathbf{B})$*).*

Definition 5 (Syndrome). *Let* $\mathcal{C}$ *be a code with parity-check matrix* $\mathbf{H}$ *and* $\mathbf{x}$ *a vector in* $\mathcal{V}$*. The vector* $\mathbf{H}\mathbf{x}^\top$ *is called the syndrome of* $\mathbf{x}$.

To any vector $\mathbf{v} \in \mathcal{V}$, one can associate an $n \times n$ square matrix with entries in $\mathcal{V}$ corresponding to the product by $\mathbf{v}$. Indeed:

$$
\begin{aligned}
\mathbf{u}\cdot\mathbf{v} &= \mathbf{u}(X)\cdot\mathbf{v}(X) \qquad \bmod P\\
&= \sum_{i=0}^{n-1} u_i X^i\cdot\mathbf{v}(X) \quad \bmod P\\
&= \sum_{i=0}^{n-1} u_i [X^i\mathbf{v}(X)] \quad \bmod P\\
&= \mathbf{u}\cdot\begin{pmatrix} \mathbf{v}(X) & \bmod P\\ X\mathbf{v}(X) & \bmod P\\ \vdots & \\ X^{n-1}\mathbf{v}(X) & \bmod P\end{pmatrix}
\end{aligned}
$$

Definition 6 (Ideal matrix). *Let $\mathbf{a}\in\mathcal{V}$ and P a polynomial of degree n over $\mathbb{F}_q$. The ideal matrix induced by $\mathbf{a}$, denoted $\vec{\mathbf{a}}$, is defined as follow:*

$$
\vec{\mathbf{a}} = \begin{pmatrix} \mathbf{a} & \bmod\ P\\ X\cdot\mathbf{a} & \bmod\ P\\ \vdots & \\ X^{n-1}\cdot\mathbf{a} & \bmod\ P\end{pmatrix}
$$

Definition 7 (Ideal codes). *Let $\mathbf{a}_1,\mathbf{a}_2,\dots\mathbf{a}_s\in\mathcal{V}$. An ideal $[sn,n]_{q^m}$ code of index s is an $\mathbb{F}_{q^m}$-linear code with a generator matrix of the form: $\mathbf{G} = \begin{pmatrix}\vec{\mathbf{a}}_1 & \vec{\mathbf{a}}_2 & \dots & \vec{\mathbf{a}}_s\end{pmatrix}$.*

Definition 8 (Systematic Ideal Codes). *A systematic ideal $[sn,n]_{q^m}$ code of index s is an ideal code with an $(s-1)n\times sn$ parity-check matrix of the form:*

$$
\mathbf{H} = \begin{pmatrix} \mathbf{I}_n & 0 & \dots & 0 & \vec{\mathbf{a}_1}\\ 0 & \mathbf{I}_n & \ddots & \vdots & \vec{\mathbf{a}_2}\\ \vdots & \ddots & \ddots & 0 & \vdots\\ 0 & \dots & 0 & \mathbf{I}_n & \vec{\mathbf{a}_{s-1}}\end{pmatrix}
$$

where for all $i\in[\![1,s-1]\!]$, $\mathbf{a}_i\in\mathcal{V}$.

We now describe Gabidulin codes which can decode deterministically up to $\lfloor\frac{n-k}{2}\rfloor$ errors. These codes were introduced in [23] while the notion of q-polynomial was introduced in [38].

Definition 9 (q-polynomials). *A q-polynomial of q-degree r over $\mathbb{F}_{q^m}$ is a polynomial defined as $P(X)=\sum_{i=0}^{r} p_i X^{[i]}$ with $[i]\stackrel{def}{=}q^i$ and for all $i\in[\![0,r]\!]$, $p_i\in\mathbb{F}_{q^m}$ and $p_r\neq 0$.*

Definition 10 (Gabidulin codes). *Let $k, n, m \in \mathbb{N}$ such that $k \leqslant n \leqslant m$. Let $\mathbf{g} = (g_1, \ldots, g_n) \in \mathcal{V}$ be a $\mathbb{F}_q$-linearly independent family of vector of $\mathbb{F}_q^m$. A Gabidulin code is a $[n,k]_{q^m}$ code whose generator matrix is:*

$$Gab_{\mathbf{g}} = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^{[1]} & g_2^{[1]} & \cdots & g_n^{[1]} \\ \vdots & \vdots & & \vdots \\ g_1^{[k-1]} & g_2^{[k-1]} & \cdots & g_n^{[k-1]} \end{pmatrix}$$

LRPC codes on the other hand are probabilistic codes introduced later on in [24].

Definition 11 (LRPC codes). *A low rank parity check codes of rank d, length n and dimension k over $\mathbb{F}_{q^m}$ is a code with parity check matrix $\mathbf{H} = (h_{i,j}) \in \mathcal{M}_{n-k,n}(\mathbb{F}_{q^m})$ such that the sub-vector space generated by the coeficients of $\mathbf{H}$ has dimension at most d.*

Hard Problems In the rank metric setting, security generally rely on the rank syndrome decoding (RSD) problem, the rank metric variant of the Syndrome Decoding (SD) problem which has been proven NP-complete in [16]. It has been proven in [27] that there exists a probabilistic reduction of the RSD problem to the SD one.

Definition 12 (RSD distribution). *For positive integers n, k and w, the $\mathsf{RSD}(n,k,w)$ distribution is the distribution obtained by choosing $\mathbf{H} \xleftarrow{\$} \mathcal{M}_{n-k,n}(\mathbb{F}_{q^m})$ and $\mathbf{x} \xleftarrow{\$} \mathcal{S}_w(\mathcal{V})$, and outputing the pair $(\mathbf{H}, (\mathbf{H}\mathbf{x}^\top)^\top)$.*

Definition 13 (Search RSD problem). *On input $(\mathbf{H}, \mathbf{y}) \in \mathcal{M}_{n-k,n}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{n-k}$ from the RSD distribution, the rank syndrome decoding problem $\mathsf{RSD}(n,k,w)$ asks to find $\mathbf{x} \in \mathcal{S}_w(\mathcal{V})$ such that $\mathbf{H}\mathbf{x}^\top = \mathbf{y}^\top$.*

Definition 14 (Decision RSD problem). *On input $(\mathbf{H}, \mathbf{y}) \in \mathcal{M}_{n-k,n}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{n-k}$, the decision RSD problem asks to decide with non negligible advantage whether $(\mathbf{H}, \mathbf{y})$ came from the $\mathsf{RSD}(n,k,w)$ distribution or the uniform distribution over $\mathcal{M}_{n-k,n}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{n-k}$.*

We now introduce the Ideal Rank Syndrome Decoding (IRSD) problem, a structured version obtained by instantiating the RSD problem with ideal codes. Let $\mathsf{ideal}_s^{syst}(\mathbb{F}_{q^m})$ denotes the set of parity check matrices of systematic ideal codes of index s over $\mathbb{F}_{q^m}$ modulo P, as defined in definition 8.

Definition 15 (s-IRSD distribution). *For positive integers n, w, s and $P \in \mathbb{F}_q[X]$ a polynomial of degree n, the ideal rank syndrome decoding distribution $s\text{-}\mathsf{IRSD}(n,w)$ is the distribution obtained by choosing $\mathbf{H} \xleftarrow{\$} \mathsf{ideal}_s^{syst}(\mathbb{F}_{q^m})$ and $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \xleftarrow{\$} \mathfrak{S}_w^s(\mathcal{V})$ and outputing the pair $(\mathbf{H}, (\mathbf{H}\mathbf{x}^\top)^\top)$.*

Definition 16 (Search s-IRSD problem). *On input* $(\mathbf{H}, \mathbf{y}) \in \mathsf{ideal}_s^{syst}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{sn}$ *from the* s-IRSD *distribution, the ideal rank syndrome decoding problem* s-IRSD(n, w) *asks to find* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \in \mathfrak{S}_w^s(\mathcal{V})$ *such that* $\mathbf{H}\mathbf{x}^\top = \mathbf{y}^\top$.

Definition 17 (Decision s-IRSD problem). *On input* $(\mathbf{H}, \mathbf{y}) \in \mathsf{ideal}_s^{syst}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{sn}$*, the decision* s-IRSD *problem asks to decide with non negligible advantage whether* $(\mathbf{H}, \mathbf{y})$ *came from the* s-IRSD(n, w) *distribution or the uniform distribution over* $\mathsf{ideal}_s^{syst}(\mathbb{F}_{q^m}) \times \mathbb{F}_{q^m}^{sn-n}$.

Although no general complexity result is known for the ideal variant of the RSD problem, no known attack taking advantage of the ideal structure has been discovered either as long as the polynomial P is chosen irreducible.

2.2 The RQC Cryptosystem

Rank Quasi-Cyclic (RQC) is a code-based IND-CCA2 encryption scheme whose security relies on the rank syndrome decoding problem. The scheme is based on an approach introduced by Alekhnovitch in [7] in which the security is reduced to the problem of decoding random codes. As such, the security do not rely on any additional assumption regarding the indistinguishablity of the family of codes being used [4, 5]. The scheme is based on an IND-CPA construction denoted RQC.PKE (see figure 1) on top of which the HHK transformation [31] is applied in order to obtain an IND-CCA2 cryptosystem.

RQC uses a Gabidulin code of generator matrix $\mathrm{Gab}_{\mathbf{g}}$ denoted $\mathcal{C}$ and a random $[2n, n]_{q^m}$ ideal code of parity-check matrix $(\mathbf{I_n}\ \vec{\mathbf{h}})$ with $\mathbf{h}$ a random element over $\mathcal{V}$.

- $\diamond$ $\mathsf{Setup}(1^{\mathfrak{K}})$: Given the security parameter $\mathfrak{K}$, generates and outputs the global parameters $\mathsf{param} = (n, k, \delta, w, w_r, P)$ where $P \in \mathbb{F}_q[X]$ is an irreducible polynomial of degree n.
- $\diamond$ $\mathsf{KeyGen}(\mathsf{param})$: Samples $\mathbf{h} \xleftarrow{\$} \mathcal{V}$, $\mathbf{g} \in \mathcal{S}_n(\mathcal{V})$, $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathfrak{S}_w^2(\mathcal{V})$. Computes $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}$ and returns $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ and $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s})$.
- $\diamond$ $\mathsf{Encrypt}(\mathsf{pk}, \mu, \theta)$: Uses randomness θ to generate $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \xleftarrow{\$} \mathfrak{S}_{w_r}^3(\mathcal{V})$. Sets $\mathbf{c}_1 = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2$ and $\mathbf{c}_2 = \mu\mathrm{Gab}_{\mathbf{g}} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{r}_3$ and returns $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$.
- $\diamond$ $\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$: Returns $\mathcal{C}.\mathsf{Decode}(\mathbf{c}_2 - \mathbf{y} \cdot \mathbf{c}_1)$.

Fig. 1. Description of RQC.PKE [4]

The correctness of RQC relies on the decoding capability of the Gabidulin code $\mathcal{C}$. Indeed, the decryption consists of decoding $\mathbf{c}_2 - \mathbf{y} \cdot \mathbf{c}_1$ which requires:

$$\omega\left(\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{y} \cdot \mathbf{r}_1 + \mathbf{r}_3\right) \leq \left\lfloor \frac{n-k}{2} \right\rfloor$$

2.3 Hash Proof Systems

Hash proof systems (HPS) were initially introduced by Cramer and Shoup in [21] in order to construct CCA-2 encryption schemes. Since then, they have been used as a building block for several applications (see for instance [29, 32, 2]). HPS are defined over an $\mathcal{NP}$ language $\mathcal{L} \subset \mathcal{X}$ and relies on two keys denoted respectively hk and hp. The *hashing key* hk can be used to associate to any word $W \in \mathcal{X}$ a *hash value* $\mathcal{H}_{\mathsf{hk}}$ while the *projection key* hp, derived from hk, can be used to associate to any word $W \in \mathcal{L}$ a *projected hash value* $\mathcal{H}_{\mathsf{hp}}$ using a witness w for the membership of W in $\mathcal{L}$. Whenever $W \in \mathcal{L}$, the hash values $\mathcal{H}_{\mathsf{hk}}$ and $\mathcal{H}_{\mathsf{hp}}$ are expected to be equal. This property is called the *correctness* of the HPS. However, whenever $W \in \mathcal{X} \setminus \mathcal{L}$, one should not be able to guess the value of $\mathcal{H}_{\mathsf{hk}}$ and thus this value should be indistinguishable from a random one. This property is the *smoothness* of the HPS. Hence, being able to compute the projective hash value of a word given only the projection key can be seen as an implicit proof of knowledge for the membership of the word in the language.

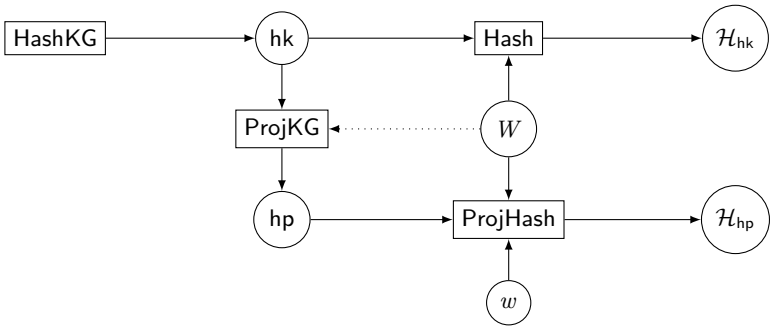

Fig. 2. Hash proof system's scheme [13]

Interesting applications arise when one uses specific languages called hard-subset-membership languages for which it is hard to decide whether an element W is inside the language or not.

Definition 18 (Hard-subset-membership). *A hard-subset-membership language is a language with the following properties:*

- ◇ *$\mathcal{L}$-samplability: There exist a polynomial-time algorithm which takes as input a parameter $\mathfrak{K}$ and randomly sample words W from $\mathcal{L}$ together with a valid witness w according to some distribution (not necessarily the uniform one).*
- ◇ *$\mathcal{X}$-samplability: There exists a polynomial-time algorithm which takes as input a parameter $\mathfrak{K}$ and randomly sample words W from $\mathcal{X}$ according to some distribution (not necessarily the uniform one).*
- ◇ *Hard-subset-membership: Let $\Delta_{\mathcal{X}}$ denotes the random uniform distribution over $\mathcal{X}$ and $\Delta_{\mathcal{L}_\mu}$ the random uniform distribution over $\mathcal{L}_\mu$. The hard-subset-*

membership property states that those two distributions are computationaly indistinguishable.

In the remaining of this section, we introduce hash proof systems formally and discuss their two main properties: *correctness* and *smoothness*.

Definition 19 (Hash Proof System). *A hash proof system over a language $\mathcal{L} \subset \mathcal{X}$, onto a set $\mathcal{V}$, is defined by five algorithms* $(\mathsf{Setup}, \mathsf{HashKG}, \mathsf{ProjKG}, \mathsf{Hash}, \mathsf{ProjHash})$*:*

- $\diamond$ $\mathsf{Setup}(1^{\mathfrak{K}})$*, give the security parameter $\mathfrak{K}$, generates the global parameters* param *of the scheme, and the description of an $\mathcal{NP}$ language $\mathcal{L}$;*
- $\diamond$ $\mathsf{HashKG}(\mathcal{L}, \mathsf{param})$*, outputs a hashing key* hk *for the language $\mathcal{L}$;*
- $\diamond$ $\mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}, \mathsf{param}, W)$*, derives the projection key* hp*, possibly depending on the word W using the hashing key* hk*;*
- $\diamond$ $\mathsf{Hash}(\mathsf{hk}, \mathcal{L}, W)$*, outputs a hash value $\mathcal{H}_{\mathsf{hk}}$, using the hashing key* hk*, and W;*
- $\diamond$ $\mathsf{ProjHash}(\mathsf{hp}, \mathcal{L}, W, w)$*, outputs the hash value $\mathcal{H}_{\mathsf{hp}}$, thanks to the projection key* hp *and the witness w of the membership of W to $\mathcal{L}$.*

In order to construct HPS in the post-quantum setting, the notion of approximate correctness was introduced in [33]. In such a HPS, the correctness property is relaxed so that the two hash values are no longer required to be equal but instead may only be close with respect to a given distance.

Definition 20 (ϵ-Correctness). *Let $W \in \mathcal{L}$ and w a witness of its membership. Let d denotes a distance. A hash proof system satisfies the ϵ-correctness property if, for all hashing keys* hk *and associated projection keys* hp*, $d(\mathcal{H}_{\mathsf{hk}}, \mathcal{H}_{\mathsf{hp}}) \leqslant \epsilon$.*

Several definitions have been proposed for the smoothness property leading to different families of HPS called $\mathsf{CS\text{-}HPS}$, $\mathsf{GL\text{-}HPS}$ and $\mathsf{KV\text{-}HPS}$ from the name of their respective authors. The smoothness ensures that given only hp, the hash value $\mathcal{H}_{\mathsf{hk}}$ of a word $W \in \mathcal{X} \backslash \mathcal{L}$ is indistinguishable from a uniformly chosen value in $\mathcal{V}$. The last notion, namely the KV-smoothness, has been introduced in [34] in order to handle cases where an adversary may generate the word W maliciously after seeing hp. In this paper, we will consider computational KV-smoothness (see figure 3) as initially introduced in [14].

If we denote by $|\mathcal{A}|$ the running time of an adversary $\mathcal{A}$, the global advantage for polynomial time adversaries running in time less than t is:

$$\mathsf{Adv}^{\mathsf{smooth}}(\mathfrak{K}, t) = \max_{|\mathcal{A}| \leqslant t} \mathsf{Adv}^{\mathsf{smooth}}_{\mathcal{A}}(\mathfrak{K})$$

where $\mathsf{Adv}^{\mathsf{smooth}}_{\mathcal{A}}(\mathfrak{K})$ is the advantage of an adversary $\mathcal{A}$ has in winning game $\mathsf{Exp}^{\mathsf{smooth}-\mathsf{b}}_{\mathcal{A}}(\mathfrak{K})$:

$$\mathsf{Adv}^{\mathsf{smooth}}_{\mathcal{A}}(\mathfrak{K}) = \left| \, \mathbb{P}[\mathsf{Exp}^{\mathsf{smooth}-1}_{\mathfrak{K}}(\mathcal{A}) = 1] - \mathbb{P}[\mathsf{Exp}^{\mathsf{smooth}-0}_{\mathfrak{K}}(\mathcal{A}) = 1] \, \right|$$

$\mathsf{Exp}^{\mathsf{smooth}-\mathsf{b}}_{\mathcal{A}}(\mathfrak{K})$:

1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $\mathsf{hk} \leftarrow \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3. $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5. $b \xleftarrow{\$} \{0, 1\}$
6. If $b = 0$, $\mathcal{H}_{\mathsf{hk}} \leftarrow \mathsf{Hash}(\mathsf{hk}, \mathcal{L}_\mu, W)$
7. If $b = 1$, $\mathcal{H}_{\mathsf{hk}} \xleftarrow{\$} \mathcal{V}$
8. $b' \leftarrow \mathcal{A}.\mathsf{guess}(\mathcal{L}_\mu, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$

Fig. 3. Game $\mathsf{Exp}^{\mathsf{smooth}-\mathsf{b}}_{\mathcal{A}}(\mathfrak{K})$ for computational KV-smoothness

3 Code-based Hash Proof System

We now describe our approximate code-based hash proof system. The language $\mathcal{L}_\mu$ used in our construction is the set of valid ciphertexts of a given message μ produced by the RQC encryption scheme. It has been shown in [33] that languages based on ciphertexts are useful to design HPS in the quantum-resistant setting, although they inherently introduce a gap in the smoothness proof of the underlying HPS. Indeed, let $\mathcal{L}^*_\mu$ denotes the set of ciphertexts that decrypt to μ, then one can see that $\mathcal{L}_\mu \subseteq \mathcal{L}^*_\mu$, as a valid RQC ciphertexts of a message μ always decrypt into μ. However, $\mathcal{L}^*_\mu \subseteq \mathcal{L}_\mu$ is generally not true for languages based on ciphertexts, as an element generated improperly can still be decrypted into μ. Elements in this gap are legion. As a toy example, consider the couple $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ with $\mathbf{c}_1 = \mathbf{h}\cdot\mathbf{r}_2$, $\mathbf{c}_2 = \mu\mathrm{Gab}_{\mathbf{g}} + \mathbf{s}\cdot\mathbf{r}_2$ and $\mathbf{r}_2$ of rank w_r (a RQC like ciphertext but with $\mathbf{r}_1$ and $\mathbf{r}_3$ set as zero). Then, one can see that $\mathbf{c}_2 - \mathbf{y}\cdot\mathbf{c}_1 = \mu\mathrm{Gab}_{\mathbf{g}} + \mathbf{x}\cdot\mathbf{r}_2$. This expression decrypts into μ as the rank of $\mathbf{x}\cdot\mathbf{r}_2$ is bellow the correction capacity of the Gabidulin code, meaning that $\mathbf{c} \in \mathcal{L}^*_\mu$, while $\mathbf{c} \notin \mathcal{L}_\mu$ as $\mathbf{c}$ is obviously not a valid RQC ciphertext of μ (remind that $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ should belong to set of the form $\mathfrak{S}^3_{w_r}(\mathcal{V})$).

Such a gap leaves a grey area regarding the security of the scheme as an adversary can produce elements that can still decrypt to μ without belonging to the language, which may open the door for practical attacks in some context such as UC-based applications as previously explained.

In order to deal with this gap, we restrict $\mathcal{X}$ as the set of all the valid ciphertexts under the RQC encryption scheme whereas the language $\mathcal{L}_\mu$ is defined as the set of all the valid ciphertexts of a given message μ. Interestingly, even if these requirements might seem quite restrictive, we show that an HPS constructed from these definitions can have many applications thanks to the versatility of coding theory. Indeed, we need to verify that we are manipulating a valid ciphertext during the considered protocols which can be performed *a fortiori* if the word W is honestly generated or *a priori* using a proof of ciphertext validity. These strategies are respectively used to construct the applications proposed in sections 6.1 and 6.2.

3.1 Language

Let n, m, k, q, w_r be some positives integers depending on the security parameter $\mathfrak{K}$. The language $\mathcal{L}_\mu$ is the set of valid ciphertexts $\mathbf{c}$ in the ciphertext space $\mathcal{CT}$ of a given message μ in the plaintext space $\mathcal{PT}$ produced using RQC as depicted in figure 1. Thus, given an RQC public key $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s})$, one have:

$$
\begin{aligned}
\mathcal{X} &= \left\{ \mathbf{c} \in \mathcal{CT} \;\middle|\; \exists \mu \in \mathcal{PT}, \exists \theta,\ \mathbf{c} = \mathsf{RQC.Encrypt}(\mathsf{pk}, \mu, \theta) \right\} \\
\mathcal{L}_\mu &= \left\{ \mathbf{c} \in \mathcal{CT} \;\middle|\; \exists \theta,\ \mathbf{c} = \mathsf{RQC.Encrypt}(\mathsf{pk}, \mu, \theta) \right\} \\
&= \left\{ \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2 \\ \mu \cdot \mathrm{Gab}_{\mathbf{g}} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{r}_3 \end{pmatrix} \;\middle|\; (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \xleftarrow{\$} \mathfrak{S}^3_{w_r}(\mathcal{V}) \right\}
\end{aligned}
$$

The witness of the membership of $W \in \mathcal{L}_\mu$ is $w = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$.

Proposition 2. *$\mathcal{L}_\mu$ is a hard-subset-membership language under the* 3-IRSD *assumption.*

Proof. $\mathcal{L}_\mu$ respect the three properties of a hard-subset membership language:

$\diamond$ $\mathcal{L}$-samplability: Sampling a random word $W \in \mathcal{L}_\mu$ with its associated witness w consists in computing a ciphertext of $\mu \in \mathcal{PT}$ using RQC and therefore can be performed in polynomial time.

$\diamond$ $\mathcal{X}$-samplability: Sampling a random element in $\mathcal{X}$ consists in computing a ciphertext of any message $\mu \in \mathcal{PT}$ using RQC and therefore can be performed in polynomial time.

$\diamond$ Hard-subset-membership: Let $\mathcal{U}$, $\Delta_\mathcal{X}$ and $\Delta_{\mathcal{L}_\mu}$ denote the random uniform distributions over $\mathcal{V}^2$, $\mathcal{X}$ and $\mathcal{L}_\mu$ respectively. Whenever $(\mathbf{c}_1, \mathbf{c}_2)$ are sampled from $\Delta_\mathcal{X}$ or $\Delta_{\mathcal{L}_\mu}$, elements of $\Delta = (\mathbf{c}_1, \mathbf{c}_2 - \mu \mathrm{Gab}_{\mathbf{g}})$ can be seen as instances of the 3-IRSD problem therefore distributions Δ and $\mathcal{U}$ are computationally indistinguishable under the 3-IRSD assumption. As the distribution Δ only differs from $\Delta_\mathcal{X}$ (respectively $\Delta_{\mathcal{L}_\mu}$) by a constant term, $\Delta_\mathcal{X}$ (respectively $\Delta_{\mathcal{L}_\mu}$) and $\mathcal{U}$ are computationally indistinguishable. Hence $\Delta_\mathcal{X}$ and $\Delta_{\mathcal{L}_\mu}$ are computationally indistinguishable under the 3-IRSD assumption.

□

3.2 Construction

- $\diamond$ $\mathsf{Setup}(1^{\mathfrak{K}})$: Given the security parameter $\mathfrak{K}$, generates the parameters param of the scheme namely $(n, m, k, q, w_1, w_2, w_\alpha, w, \theta)$. Samples $\mathbf{h} \xleftarrow{\$} \mathcal{V}$, $\mathbf{g} \xleftarrow{\$} \mathcal{S}_n(\mathcal{V})$, $(\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathfrak{S}^2_w(\mathcal{V})$, use randomness θ to generate $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \xleftarrow{\$} \mathfrak{S}^3_{w_r}(\mathcal{V})$. Computes $(\mathbf{c}_1, \mathbf{c}_2) = \mathsf{RQC.Encrypt}(\mathsf{pk}, \mu, \theta)$ and $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}$.
- $\diamond$ $\mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$: Samples $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3) \xleftarrow{\$} \mathfrak{S}^3_{w_\alpha}(\mathcal{V})$.
  Returns $\mathsf{hk} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3)$.
- $\diamond$ $\mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$: Returns $\mathsf{hp} = \mathbf{s} \cdot \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$.
- $\diamond$ $\mathsf{Hash}(\mathsf{hk}, \mathcal{L}_\mu, W)$: Returns $\mathcal{H}_{\mathsf{hk}} = \boldsymbol{\alpha}_1 \cdot (\mathbf{c}_2 - \mu \mathrm{Gab}_{\mathbf{g}}) + \boldsymbol{\alpha}_3$.
- $\diamond$ $\mathsf{ProjHash}(\mathsf{hp}, \mathcal{L}_\mu, W, w)$: Returns $\mathcal{H}_{\mathsf{hp}} = \mathsf{hp} \cdot \mathbf{r}_2$

Fig. 4. A code-based HPS

As an approximate HPS, the construction described in figure 4 computes two values $\mathcal{H}_{\mathsf{hk}}$ and $\mathcal{H}_{\mathsf{hp}}$ such that $\omega(\mathcal{H}_{\mathsf{hk}} - \mathcal{H}_{\mathsf{hp}})$ is relatively small. To achieve this, $\mathcal{H}_{\mathsf{hk}}$ and $\mathcal{H}_{\mathsf{hp}}$ are defined such that they both contain the value $\mathbf{s}\boldsymbol{\alpha}_1\mathbf{r}_2$ but differ due to additional noise from a product space $\langle E_\alpha, E_r \rangle$ where E_r and E_α are two subspaces of small dimension, denoting respectively the shared support of $\mathbf{r}_i$ and $\boldsymbol{\alpha}_i$ values. In addition, the Gabidulin code allows to check if $W \in \mathcal{X}$ is in $\mathcal{L}_\mu$ or not using the secret key $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ as a trapdoor.

4 Security of the HPS

4.1 Correctness Property

Theorem 1. *The* HPS *depicted in figure 4 satisfies the $w_\alpha(w_r + 1)$-correctness property.*

Proof. Let $W \in \mathcal{L}_\mu$. We have $\mathcal{H}_{\mathsf{hk}} - \mathcal{H}_{\mathsf{hp}} = \boldsymbol{\alpha}_1 \cdot \mathbf{r}_3 + \boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_2 \cdot \mathbf{r}_2$. As $\mathsf{Supp}(\boldsymbol{\alpha}_1) = \mathsf{Supp}(\boldsymbol{\alpha}_2)$ and $\mathsf{Supp}(\mathbf{r}_3) = \mathsf{Supp}(\mathbf{r}_2)$, the product space $\langle \boldsymbol{\alpha}_1\mathbf{r}_3 \rangle_{\mathbb{F}_q} = \langle \boldsymbol{\alpha}_2\mathbf{r}_2 \rangle_{\mathbb{F}_q}$ and has dimension $w_\alpha w_r$. Furthermore, the space $\langle \boldsymbol{\alpha}_3 \rangle_{\mathbb{F}_q}$ is not included in this product space and has dimension w_α. Finally, the overall expression has a maximal rank of $w_\alpha(w_r + 1)$.

□

4.2 A Problem Underlying the Smoothness Property

Intuition of the Problem The *smoothness* property of our HPS depend of the decisional version of a special kind of 3-IRSD problem where an adversary

can partially manipulate the parity check matrix being used. Given $\mathbf{s}, \mathbf{t} \xleftarrow{\$} \mathcal{V}$ and $\mathbf{y}_1, \mathbf{y}_2 \in \mathcal{V}$, consider the following syndrome equation:

$$\begin{pmatrix} \vec{\mathbf{s}} & \mathbf{I}_n & \vec{0} \\ \vec{\mathbf{t}} & \vec{0} & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \boldsymbol{\alpha}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$$

Finding $(\boldsymbol{\alpha_1}, \boldsymbol{\alpha_2}, \boldsymbol{\alpha_3}) \in \mathfrak{S}^3_{w_\alpha}(\mathcal{V})$ satisfying the above equation correspond to an instance of a 3-IRSD problem. Now, instead of a random $\mathbf{t}$ value, consider that an adversary can chooses $\mu \in \mathbb{F}_{q^m}^k$ and $(\mathbf{r}_1, \mathbf{r}_2) \in \mathfrak{S}^2_{w_r}(\mathcal{V})$. Let $\mathbf{t} = \mu \mathrm{Gab}_{\mathbf{g}} + \mathbf{s}\mathbf{r}_1 + \mathbf{r}_2$. The question we have to deal with is to determine if, under this particular shape, the problem remains hard, and more specifically its decisional version.

Description of the Problem In this part, we formally describe the problem that we denote flexible ideal rank syndrome decoding problem (FIRSD).

Definition 21 (FIRSD distribution). *Given $\mathbf{g} \in \mathcal{S}_n(\mathcal{V})$, $\mathbf{s} \xleftarrow{\$} \mathcal{V}$ and $w_\alpha, w_r \in \mathbb{N}$, consider an oracle denoted by $\mathcal{O}_{\mathbf{s},\mathbf{g}}(\mathbf{t})$ that generates $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3) \xleftarrow{\$} \mathfrak{S}^3_{w_\alpha}(\mathcal{V})$, calculates and outputs $\mathbf{y}_1 = \mathbf{s}\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$ as a public value, then verify with a proof of validity $\boldsymbol{\Pi}$ that its input $\mathbf{t}$ is of the form $\mathbf{t} = \mu Gab_{\mathbf{g}} + \mathbf{s}\mathbf{r}_1 + \mathbf{r}_2$ with $\mu \in \mathbb{F}_{q^m}^k$ and $(\mathbf{r}_1, \mathbf{r}_2) \in \mathfrak{S}^2_{w_r}(\mathcal{V})$, calculates $\mathbf{y}_2 = \mathbf{t}\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_3$ and outputs $(\mathbf{y}_1, \mathbf{y}_2)$ if $\mathbf{t}$ is valid and $\perp$ otherwise.*

Definition 22 (Decision FIRSD problem). *Given $\mathbf{g} \in \mathcal{S}_n(\mathcal{V})$, $\mathbf{s} \xleftarrow{\$} \mathcal{V}$ and $w_\alpha, w_r \in \mathbb{N}$, consider an oracle denoted by $\tilde{\mathcal{O}}_{\mathbf{s},\mathbf{g}}(\mathbf{t})$ that generates $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3) \xleftarrow{\$} \mathfrak{S}^3_{w_\alpha}(\mathcal{V})$ and coin $\xleftarrow{\$} \{0,1\}$, calculates and outputs $\mathbf{y}_1 = \mathbf{s}\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2$ as a public value, then verify with a proof of validity $\boldsymbol{\Pi}$ that its input $\mathbf{t}$ is of the form $\mathbf{t} = \mu Gab_{\mathbf{g}} + \mathbf{s}\mathbf{r}_1 + \mathbf{r}_2$ with $\mu \in \mathbb{F}_{q^m}^k$ and $(\mathbf{r}_1, \mathbf{r}_2) \in \mathfrak{S}^2_{w_r}(\mathcal{V})$, calculates $\mathbf{y}_2 = \mathbf{t}\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_3$, and outputs, if $\mathbf{t}$ is valid, the couple $(\mathbf{y}_1, \mathbf{y}_2)$ if coin $= 0$, a couple $(\mathbf{y}_1, \mathbf{y}_2)$ with $\mathbf{y}_2$ a random value over $\mathcal{V}$ if coin $= 1$, and $\perp$ if $\mathbf{t}$ is not valid. The decision* FIRSD *problem ask to decide, with non negligible advantage, whether $\mathbf{y}_2$ came from the* FIRSD *distribution or the uniform distribution over $\mathcal{V}$.*

Discussion Upon the Problem's Hardness

Claim. The decisional FIRSD problem is hard.

As previously highlighted, this problem is close to the decisional 3-IRSD problem. However, in this particular case, the adversary has some additional control over a part of the parity check matrix $\mathbf{H}$ being used, namely on the ideal matrix generated from $\mathbf{t}$. The question is to determine how it can impact the hardness of the problem.

General overview. Let $\mathbf{g} \in \mathcal{S}_n(\mathcal{V})$, $\mathbf{s} \xleftarrow{\$} \mathcal{V}$ and $w_\alpha, w_r \in \mathbb{N}$. Consider an adversary choosing $(\mathbf{r}_1, \mathbf{r}_2) \in \mathfrak{S}^2_{w_r}(\mathcal{V})$ and $\mu \in \mathbb{F}_{q^m}^k$, and generating $\mathbf{t} = \mu \mathrm{Gab}_{\mathbf{g}} +$

$\mathbf{sr}_1 + \mathbf{r}_2$. The decision FIRSD problem ask to distinguish a value $\mathbf{y}_2$ having a syndrome form as described bellow from a random generated one.

$$\begin{pmatrix} \vec{\mathbf{s}} & \mathbf{I}_n & 0 \\ \vec{\mathbf{t}} & 0 & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha_1} \\ \boldsymbol{\alpha_2} \\ \boldsymbol{\alpha_3} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$$

Let us consider two extreme cases. The 0-entropy case, where an adversary would have no choice upon the value of the vector $\mathbf{t}$. In this situation, $\mathbf{t}$ would be a random vector over $\mathcal{V}$ and the problem would be a decisional 3-IRSD problem under a systematic form. The full-entropy case, on the other hand, would be the extreme opposite situation where an adversary could manage the value of $\mathbf{t}$ as desired. In that case, the problem would no longer be hard, as an adversary could choose $\mathbf{t} = (1, 0, \ldots, 0)$ in order to obtain the identity matrix, leaking the shared support of $(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3)$ in the case where $\mathbf{y}_2$ is a syndrome.

As we can see, the hardness of the problem is deeply related to the capability of an adversary to handle the value of the vector $\mathbf{t}$. We will now give some insight upon the difficulty of the problem. Firstly, we will consider the problem under an entropy approach, and show that, under a suitable choice of parameters, the part an adversary can manage over the $\mathbf{t}$ value is restricted. Secondly, we will describe the case where an adversary try to forge a $\mathbf{t}$ value with a specific shape and show that such a strategy in unlikely to leak any information. We will consider two cases. The first consisting of forging a vector $\mathbf{t}$ with as many zeros as possible and then a second strategy consisting of lowering the rank of $\mathbf{t}$ as much as possible.

Entropy considerations. As previously mentioned, the difficulty of the problem depends on how much an adversary can control the value of $\mathbf{t}$ which depends on the parameters being used.

In an attempt to manage the value of the $\mathbf{t}$ vector, the adversary can first choose the message $\mu \in \mathbb{F}_{q^m}^k$ of its choice, which represent q^{km} possibilities. Then, he must chooses a subspace E of $\mathbb{F}_{q^m}$ of dimension w_r.

There are up to $q^{w_r n}$ possibilities for the choice of $\mathbf{r}_1$. and up to $q^{w_r n}$ possibilities for $\mathbf{r}_2$ as well. Overall, the number of bits that an adversary can manipulates is upper bounded by $km + 2nw_r$.

On the other hand, choosing a vector over the entire space $\mathcal{V}$ represents q^{mn} possibilities, which can be viewed as an mn bits surface. Therefore, the proportion P of bits that an adversary could handle is equal to $\frac{km+2nw_r}{mn}$. With our set of parameters, as presented in the following table, this value never exceed 13%.

The ideal structure constraint. As we have seen, an adversary is restricted upon the choice of the vector $\mathbf{t}$, and this vector is then used to generate an ideal matrix. As we will see, the inherent cyclicity due to the ideal structure is also a huge obstacle for the elaboration of an attack. Let consider the following sub

Table 1. Proportion of bits possibly manipulated.

Instance	q	n	m	k	w_r	P	Security
I	2	137	139	4	7	0.123	128
II	2	211	223	4	8	0.104	192
III	2	283	293	3	13	0.099	256

2-IRSD problem extracted from the previous 3-IRSD one:

$$\begin{pmatrix} \vec{\mathbf{t}} & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_3 \end{pmatrix} = \mathbf{y}_2$$

First approach: An adversary could put as many zeros as possible on the $\mathbf{t}$ value in order to retrieve information about $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_3$.

This way, by solving a system of linear equations, an adversary could manage to set a bloc of approximatively w_r zeros on the vector $\mathbf{t}$.

The ideal matrix generated from $\mathbf{t}$ will then shift those zeros over the matrix, and some of them will be absorb due to the modulus polynomial P used for the ideal structure.

As the number of zeros are limited to w_r coordinates over a total of n coordinates, the proportion P_0 of zero coordinates can not exceed 6% of the coordinates of $\mathbf{t}$. Therefore, such a strategy is unlikely to leak any information about the syndrome nor modifying the distribution in order to distinguish it from a random uniform one.

Table 2. Proportion of zeros.

Instance	q	n	m	k	w_r	P_0	Security
I	2	137	139	4	7	0.051	128
II	2	211	223	4	10	0.047	192
III	2	283	293	3	13	0.046	256

Second approach: An adversary could lower the rank of $\mathbf{t}$ as much as possible. This would lead to the construction of a parity check matrix under systematic form of a known decodable $[2n, n]_{q^m}$ code from $\mathbf{t}$. An adversary could reduce the rank of the vector $\mathbf{t}$ of approximatively w_r, however this would not be enough to enable the possibility of using a decoding algorithm of a LRPC code.

Moreover, reducing the rank of $\mathbf{t}$ to approximatively w_r will not affect the rank of the overall associated syndrome $\mathbf{y}_2 = \mathbf{t} \cdot \boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_3$. Indeed, we would have $\omega(\mathbf{t}) = n - w_r$ and $\omega(\boldsymbol{\alpha}_1) = \omega(\boldsymbol{\alpha}_3) = w_\alpha$, and as $(n - w_r + 1) \cdot w_\alpha \gg m$, the overall rank would not be affected.

4.3 Computational KV-Smoothness

Theorem 2. *The* HPS *depicted in figure 4 satisfies the computational* KV*-smoothness property under the 2-*IRSD *and the decisionnal* FIRSD *assumptions.*

Proof. We now prove the smoothness of our HPS under the 2-IRSD and the decisionnal FIRSD assumptions, by building a sequence of games transitioning from the real game $\mathbf{G}_1$ with an adversary receiving an honest value of $\mathcal{H}_{\mathsf{hk}}$ to an adversary receiving in $\mathbf{G}_4$ a random value over $\mathcal{V}$.

Game $\mathbf{G}_{1,\mathcal{A}}(\mathfrak{K})$:
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3. $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5. $\mathcal{H}_{\mathsf{hk}} \leftarrow \mathsf{Hash}(\mathcal{L}_\mu, \mathsf{hk}, W)$
6. $b' \leftarrow \mathcal{A}.\mathsf{guess}(\mathcal{L}_\mu, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$

Game $\mathbf{G}_{2,\mathcal{A}}(\mathfrak{K})$:
1.a. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
1.b. $\mathsf{param}.\mathbf{s} \xleftarrow{\$} \mathcal{V}$
2. $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3. $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5. $\mathcal{H}_{\mathsf{hk}} \leftarrow \mathsf{Hash}(\mathcal{L}_\mu, \mathsf{hk}, W)$
6. $b' \leftarrow \mathcal{A}.\mathsf{guess}(\mathcal{L}_\mu, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$

$\mathcal{D}^*_{\mathfrak{K}}((\mathbf{I}_n, \vec{\mathbf{h}}), \mathbf{s})$:
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $\mathsf{param}.\mathbf{s} \leftarrow \mathbf{s}$
3. $\mathsf{param}.\mathbf{h} \leftarrow \mathbf{h}$
4. $b' \leftarrow \mathcal{D}_{\mathfrak{K}}(\mathcal{L}_\mu, \mathsf{param})$
5. If $b' == 1$, output $[2,1]$-IRSD
6. If $b' == 2$, output UNIFORM

In game $\mathbf{G}_2$, we forget the secret values associated with $\mathbf{s}$ by taking it randomly over the set $\mathcal{V}$ and then proceed honestly. Let $\mathcal{D}_{\mathfrak{K}}(\mathcal{L}_\mu, \mathsf{param})$ denotes an algorithm that can distinguish between Game $\mathbf{G}_1$ and Game $\mathbf{G_2}$ with advantage ϵ. Then, one can build an algorithm $\mathcal{D}^*_{\mathfrak{K}}$ breaking the 2-IRSD instance with the same advantage ϵ.

Game $\mathbf{G}_{3,\mathcal{A}}(\mathfrak{K})$:
1.a. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
1.b. $\mathsf{param}.\mathbf{s} \xleftarrow{\$} \mathcal{V}$
2. $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3. $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5.a. $\mathcal{H}_{\mathsf{hk}} \leftarrow \mathsf{Hash}(\mathcal{L}_\mu, \mathsf{hk}, W)$
5.b. $\mathcal{H}_{\mathsf{hk}} \xleftarrow{\$} \mathcal{V}$
6. $b' \leftarrow \mathcal{A}.\mathsf{guess}(\mathcal{L}_\mu, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$

$\mathcal{D}^*_{\mathfrak{K}}(\mathbf{s}, \mathbf{g})$:
1.a $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
1.b $\mathsf{param}.\mathbf{s} \leftarrow \mathbf{s}$
2. $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3 $\mathsf{hp} \leftarrow \tilde{\mathcal{O}}_{\mathbf{s},\mathbf{g}}$ (public value)
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5. $\mathcal{H}_{\mathsf{hk}} \leftarrow \tilde{\mathcal{O}}_{\mathbf{s},\mathbf{g}}(\mathbf{c}_2 - \mu\mathrm{Gab}_{\mathbf{g}})$
6. $b' \leftarrow \mathcal{D}_{\mathfrak{K}}(\mathcal{L}_\mu, \mathsf{param}, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$
7. If $b' == 3$ output FIRSD
8. If $b' == 4$ output UNIFORM

In game $\mathbf{G}_3$, we forget the hash value $\mathcal{H}_{\mathsf{hk}}$ by taking it randomly over the set $\mathcal{V}$, and then proceed honestly. Let $\mathcal{D}_{\mathfrak{K}}(\mathcal{L}_\mu, \mathsf{param}, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$ denotes an algorithm that can distinguish between Game $\mathbf{G}_2$ and Game $\mathbf{G_3}$ with advantage ϵ. Then, one can build an algorithm $\mathcal{D}^*_{\mathfrak{K}}$ breaking the decision FIRSD-problem

assumption with the same advantage ϵ.

Game $\mathbf{G}_{4,\mathcal{A}}(\mathfrak{K})$:
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $\mathsf{hk} \xleftarrow{\$} \mathsf{HashKG}(\mathcal{L}_\mu, \mathsf{param})$
3. $\mathsf{hp} \leftarrow \mathsf{ProjKG}(\mathsf{hk}, \mathcal{L}_\mu, \mathsf{param})$
4. $W \in \mathcal{X} \setminus \mathcal{L}_\mu \leftarrow \mathcal{A}.\mathsf{choose}(\mathcal{L}_\mu, \mathsf{hp})$
5. $\mathcal{H}_{\mathsf{hk}} \xleftarrow{\$} \mathcal{V}$
6. $b' \leftarrow \mathcal{A}.\mathsf{guess}(\mathcal{L}_\mu, \mathcal{H}_{\mathsf{hk}}, \mathsf{hp}, W)$

In game $\mathbf{G}_4$, we reconstruct $\mathbf{s}$ as a syndrome and then proceed honestly. Arguments are the same that the ones used for the transition between games $\mathbf{G}_1$ and $\mathbf{G}_2$.

We have built a sequence of games allowing to transition from $\mathsf{Exp}^{\mathsf{smooth}-0}_{\mathfrak{K}}(\mathcal{A})$ to $\mathsf{Exp}^{\mathsf{smooth}-1}_{\mathfrak{K}}(\mathcal{A})$, therefore our HPS satisfies the computational KV-smoothness under the 2-IRSD and the decision FIRSD assumptions.

□

Therefore, the advantage of an adversary against the $\mathsf{Exp}^{\mathsf{smooth}-\mathsf{b}}_{\mathcal{A}}(\mathfrak{K})$ experiment is bounded as:

$$\mathsf{Adv}^{\mathsf{smooth}}_{\mathcal{A}}(\mathfrak{K}) \leqslant 2 \times \mathsf{Adv}^{\text{2-IRSD}}(\mathfrak{K}) + \mathsf{Adv}^{\mathsf{FIRSD}}(\mathfrak{K})$$

□

5 Proof of Ciphertext Validity

In this section, we design a zero-knowledge proof of ciphertext validity for the RQC encryption scheme. The latter allows a prover to convince a verifier that a given ciphertext is well-formed and therefore is a valid RQC encryption. Next, this proof is extended in order to prove that two ciphertexts of the same message μ have been generated correctly. We need this primitive to construct the PAKE described in section 6.2.

We start by briefly reviewing some of the results related to proofs of knowledge in coding theory. Stern's proof system [41] is a 3-rounds zero-knowledge interactive protocol based on the hardness of the syndrome decoding problem. Let $\mathbf{H}$ be a public matrix, $\mathbf{x}$ a vector of small Hamming weight w_x and $\mathbf{s} = \mathbf{H}\mathbf{x}^\top$ the associated syndrome. The protocol allows a prover P to prove to a verifier V that he knows a vector $\mathbf{x}$ of weight w_x such that $\mathbf{s} = \mathbf{H}\mathbf{x}^\top$. The scheme has perfect completeness, however a cheating prover can trick the verifier with probability up to $\frac{2}{3}$. Thus, the scheme has to be repeated many times to obtain a negligible soundness error. In 1995, Chen [20] adapted Stern's scheme to the rank metric setting where the security is based on the rank syndrome decoding problem. In 2011, Gaborit et al. [26] break this protocol in two different ways and then propose a reparation fixing the protocol. In 2016, a even more flexible variant of this Stern like proof of knowledge called *Rank Concatenated Stern's*

Protocol ($\mathcal{RCSP}$) was proposed in [6]. In order to achieve a proof of knowledge for a RQC ciphertext, we need to go deeper and propose an even more flexible version of this Stern-like proof.

5.1 Proof of Ciphertext Validity for RQC

The *Rank Concatenated Stern's Protocol* ($\mathcal{RCSP}$) variant proposed in [6] allows a prover to convince a verifier that he knows two secrets vectors of small rank weights. Suppose $\mathbf{Q} \xleftarrow{\$} \mathcal{M}_{k,n}(\mathbb{F}_{q^m})$, $\mathbf{R} \xleftarrow{\$} \mathcal{M}_{k,n'}(\mathbb{F}_{q^m})$ and $\mathbf{x}$, $\mathbf{y}$ are two vectors of small weight w_x and w_y such that the following equation holds:

$$(\mathbf{Q} \;\; \mathbf{R}) \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{s}$$

The protocol depicted in [6] then allows to prove the knowledge of the couple of vectors satisfying this syndrome equation and such that $\omega(\mathbf{x}) = w_x$ and $\omega(\mathbf{y}) = w_y$.

For our purpose, we need a more flexible Stern-like protocol that constrains the blocks independently which is not provided by the $\mathcal{RCSP}$ protocol. More precisely, we consider the following expression:

$$\begin{pmatrix} \mathbf{I}_n & \vec{\mathbf{h}} & 0 & 0 \\ 0 & \vec{\mathbf{s}} & \mathbf{I}_n & \mathrm{Gab}_{\mathbf{g}} \end{pmatrix} \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mu \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}$$

The prover should be able to prove that values $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ verify this syndrome equation, that $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$ and we want no condition on μ as the support of μ has nothing to do with the supports of $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{r}_3$. We now describe some supporting definitions and results in order to achieve this goal.

Definition 23. *For $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, we say that $\mathbf{x}$ and $\mathbf{y}$ are equivalent, denoted $\mathbf{x} \sim \mathbf{y}$, if $\mathsf{Supp}(\mathbf{x}) = \mathsf{Supp}(\mathbf{y})$.*

As explained in the preliminary section, to any vector $\mathbf{x} \in \mathcal{V}$, one can associate a matrix $\mathbf{M}_{\mathbf{x}} \in \mathcal{M}_{m,n}(\mathbb{F}_q)$ for a basis $\mathcal{B}$ of $\mathbb{F}_{q^m}$. Let $\phi_{\mathcal{B}}$ denotes this application.

$$\begin{array}{ccccc} \phi_{\mathcal{B}} : & \mathbb{F}_{q^m}^n & \simeq & \mathcal{M}_{m,n}(\mathbb{F}_q) & \\ & \mathbf{v} = (v_0, \ldots, v_{n-1}) & \mapsto & \mathbf{M}_{\mathbf{v}} = \begin{pmatrix} v_{1,0} & \cdots & v_{1,n-1} \\ v_{2,0} & \cdots & v_{2,n-1} \\ \vdots & & \vdots \\ v_{m,0} & \cdots & v_{m,n-1} \end{pmatrix} & \begin{matrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{matrix} \end{array}$$

Definition 24. *Let $\mathbf{Q} \in \mathsf{GL}_m(q)$, $\mathbf{v} \in \mathcal{V}$ and $\mathcal{B}$ a basis of $\mathbb{F}_{q^m}$. We define the product $\mathbf{Q} * \mathbf{v}$ such that $\mathbf{Q} * \mathbf{v} = \phi_{\mathcal{B}}^{-1}(\mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{v}))$.*

Lemma 1. *For all $\mathbf{v} \in \mathcal{V}$, $P \in \mathsf{GL}_n(q)$ and $Q \in \mathsf{GL}_m(q)$, we have :*

$$\phi_{\mathcal{B}}(\mathbf{v} \cdot P) = \phi_{\mathcal{B}}(\mathbf{v}) \cdot P$$

Proof. Let $\mathbf{v} \in \mathcal{V}$, $P = (p_{ij}) \in \mathsf{GL}_n(q)$ and $Q = (q_{ij}) \in \mathsf{GL}_m(q)$. Let denotes $\mathbf{v} = (v_1, \ldots, v_n)$ and, for all $i \in [\![1, n]\!]$, $v_i = \sum_{i=0}^{m} v_{ik}\beta_k$ (decomposition of $\mathbf{v}$ in the basis $\mathcal{B}$).

$$\begin{aligned}
\phi_{\mathcal{B}}(\mathbf{v} \cdot P) &= \phi_{\mathcal{B}}(\sum_{i=1}^{n} v_i p_{i1}, \ldots, \sum_{i=1}^{n} v_i p_{in}) \\
&= \phi_{\mathcal{B}}(\sum_{i=1}^{n}\sum_{k=1}^{m} v_{ik}\beta_k p_{i1}, \ldots, \sum_{i=1}^{n}\sum_{k=1}^{m} v_{ik}\beta_k p_{in}) \\
&= \phi_{\mathcal{B}}(\sum_{k=1}^{m} \beta_k \sum_{i=1}^{n} v_{ik} p_{i1}, \ldots, \sum_{k=1}^{m} \beta_k \sum_{i=1}^{n} v_{ik}\ p_{in}) \\
&= \begin{pmatrix} \sum_{i=1}^{n} v_{i1}p_{i1} & \cdots & \sum_{i=1}^{n} v_{i1}p_{in} \\ \vdots & & \vdots \\ \sum_{i=1}^{n} v_{im}p_{i1} & \cdots & \sum_{i=1}^{n} v_{im}p_{in} \end{pmatrix} \\
&= \phi_{\mathcal{B}}(\mathbf{v}) \cdot P
\end{aligned}$$

□

Proposition 3. *For all $\mathbf{v} \in \mathcal{V}$, $P \in \mathsf{GL}_n(q)$ and $Q \in \mathsf{GL}_m(q)$, we have :*

$$(Q * \mathbf{v}) \cdot P = Q * (\mathbf{v} \cdot P)$$

Proof. Let $\mathbf{v} \in \mathcal{V}, P \in \mathsf{GL}_n(q)$ and $Q \in \mathsf{GL}_m(q)$.

$$\begin{aligned}
(Q * \mathbf{v}) \cdot P &= (Q \cdot \phi_{\mathcal{B}}(\mathbf{v})) \cdot P \\
&= Q \cdot (\phi_{\mathcal{B}}(\mathbf{v}) \cdot P) \\
&= Q \cdot \phi_{\mathcal{B}}(\mathbf{v} \cdot P) \\
&= Q * (\mathbf{v} \cdot P)
\end{aligned}$$

□

Lemma 2. *For all $\mathbf{x} \in \mathcal{V}$ and $\mathbf{Q} \in \mathsf{GL}_m(q)$, $\omega(\mathbf{Q} * \mathbf{x}) = \omega(\mathbf{x})$.*

Proof. Let $\mathbf{x} \in \mathcal{V}$ and $\mathbf{Q} \in \mathsf{GL}_m(q)$. Let us first show that $\mathbf{Q} * \mathbf{x}$ can not have a rank greater than $\mathbf{x}$.

Multiplying $\mathbf{x}$ on the left by $\mathbf{Q}$ is equivalent to do linear combinations over the lines of $\phi_{\mathbf{x}}$. Increasing the rank of $\mathbf{x}$ means that $\mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{x})$ contains lines that are not linear combinations of the lines of $\phi_{\mathcal{B}}(\mathbf{x})$ which is impossible by doing just linear combinations over the lines of $\phi_{\mathbf{x}}$.

Now, let us show that its rank can not be decreased neither. Let w_x denotes the rank of $\mathbf{x}$ ($w_x < \min(m, n)$). Suppose, by the absurd, that the rank of $\omega(\mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{x}))$ is lower than w_x. As $\mathbf{Q} \in \mathsf{GL}_m(q)$, there exists $\mathbf{Q}^{-1} \in \mathsf{GL}_m(q)$ such that $\mathbf{Q}^{-1} \cdot \mathbf{Q} = \mathbf{I}_m$. So $\omega(\mathbf{Q}^{-1} \cdot \mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{x})) = \omega(\phi_{\mathcal{B}}(\mathbf{x}))$. As $\omega(\mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{x})) < w_x$, it means that multiplying on the left by $\mathbf{Q}^{-1}$ has increased the rank of $\mathbf{Q} \cdot \phi_{\mathcal{B}}(\mathbf{x})$. Contradiction. Hence, $\omega(\mathbf{Q} * \mathbf{x}) = \omega(\mathbf{x})$.

□

Lemma 3. *For all* $\mathbf{x} \in \mathcal{V}$ *and* $\mathbf{P} \in \mathsf{GL}_n(q)$, $\omega(\mathbf{x} \cdot \mathbf{P}) = \omega(\mathbf{x})$.

Proof. The same reasonning as the previous proof can be applied with the columns of $\phi_{\mathcal{B}}(\mathbf{x})$ instead of its lines.

□

Corollary 2. *For all* $\mathbf{x} \in \mathcal{V}$, $\mathbf{Q} \in \mathsf{GL}_m(q)$ *and* $\mathbf{P} \in \mathsf{GL}_n(q)$, $\omega(\mathbf{Q} * \mathbf{x} \cdot \mathbf{P}) = \omega(\mathbf{x})$.

Proof. It is a direct consequency of the two previous lemmas.

□

Proposition 4. *For all* $\mathbf{x}, \mathbf{y} \in \mathcal{V}$ *with* $\omega(\mathbf{x}) = \omega(\mathbf{y})$, *there exists* $\mathbf{Q} \in \mathsf{GL}_m(q)$ *and* $\mathbf{P} \in \mathsf{GL}_n(q)$ *such that* $\mathbf{y} = \mathbf{Q} * \mathbf{x}\mathbf{P}$.

Proof. Let $\mathbf{x}, \mathbf{y} \in \mathcal{V}$ with $\omega(\mathbf{x}) = \omega(\mathbf{y}) = r$ (necessarly $r < \min(m, n)$). There exists $\mathbf{P}_1 \in \mathsf{GL}_n(q)$ such that $\phi_{\mathcal{B}}(\mathbf{x}) \cdot \mathbf{P}_1$ is in systematic form with the first r columns non equal to zero. There exists $\mathbf{P}_2 \in \mathsf{GL}_n(q)$ such that $\phi_{\mathcal{B}}(\mathbf{y}) \cdot \mathbf{P}_2$ is in systematic form with the first r columns non equal to zero. The first r columns of $\phi_{\mathcal{B}}(\mathbf{x}) \cdot \mathbf{P}_1$ form a basis $\mathcal{B}_1$ of $\mathbb{F}_{q^m}^r$. The first r columns of $\phi_{\mathcal{B}}(\mathbf{y}) \cdot \mathbf{P}_2$ form another basis $\mathcal{B}_2$ of $\mathbb{F}_{q^m}^r$. Hence, if we denote by $\mathbf{Q}$ the matrix which transforms the basis $\mathcal{B}_1$ into $\mathcal{B}_2$, we have $\mathbf{Q} \cdot \mathbf{x} \cdot \mathbf{P}_1 = \mathbf{y} \cdot \mathbf{P}_2$. Finally, $\mathbf{y} = \mathbf{Q} \cdot \mathbf{x} \cdot (\mathbf{P}_1 \cdot \mathbf{P}_2^{-1})$.
□

In the $\mathcal{RCSP}$ protocol, an element $\mathbf{v} \in \mathcal{V}$ is transformed to any element of the same rank by calculating $\mathbf{Q} * \mathbf{v}\mathbf{P}$ with $\mathbf{Q} \in \mathsf{GL}_m(q)$ and $\mathbf{P} \in \mathsf{GL}_n(q)$. For each block, different matricies $\mathbf{Q}$ and $\mathbf{P}$ are used. For our purpose, we need to keep a support relationship between vectors, not restricted to their rank, as in the RQC cryptosystem. This property is ensured by the following propositions.

Lemma 4. *For all* $\mathbf{x} \in \mathcal{V}$, *for all* $\mathbf{P} \in \mathsf{GL}_n(q)$, $\mathbf{x} \sim \mathbf{x} \cdot \mathbf{P}$.

Proof. Let $\mathbf{x} \in \mathcal{V}$ and $\mathbf{P} \in \mathsf{GL}_n(q)$. Let r denotes the rank of $\mathbf{x}$, and $\mathcal{B}_\gamma = \{\gamma_1, \ldots, \gamma_r\}$ a basis of the support of $\mathbf{x}$. As multiplying to the right by $\mathbf{P}$ is equivalent to do linear combinations over the columns of $\phi_{\mathcal{B}}(\mathbf{x})$, $\mathcal{B}_\gamma$ generates all the coordinates of $\mathbf{x} \cdot \mathbf{P}$. As we have already proved that $\omega(\mathbf{x}) = \omega(\mathbf{x} \cdot \mathbf{P})$, $\phi_{\mathcal{B}}(\mathbf{x})$ is a basis of the support of $\mathbf{x} \cdot \mathbf{P}$. Hence, $\mathbf{x} \sim \mathbf{x} \cdot \mathbf{P}$

□

Proposition 5. *For all* $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, *for all* $\mathbf{Q} \in \mathsf{GL}_m(q)$, *for all* $\mathbf{P}_1, \mathbf{P}_2 \in \mathsf{GL}_n(q)$, *if* $\mathbf{x} \sim \mathbf{y}$ *then* $(\mathbf{Q} * \mathbf{x}\mathbf{P}_1) \sim (\mathbf{Q} * \mathbf{y}\mathbf{P}_2)$.

Proof. Let $\mathbf{x}, \mathbf{y} \in \mathcal{V}$ such that $\mathbf{x} \sim \mathbf{y}$. Let $\mathbf{Q} \in \mathsf{GL}_m(q)$ and $\mathbf{P}_1, \mathbf{P}_2 \in \mathsf{GL}_n(q)$.

Let us first prove that $\mathbf{Q} * \mathbf{x} \sim \mathbf{Q} * \mathbf{y}$. Let denotes $\mathcal{B}_\gamma = \{\gamma_1, \ldots, \gamma_r\}$ a basis of the support of $\mathbf{x}$ and $\mathbf{y}$. Each column of $\phi_{\mathcal{B}}(\mathbf{x})$ and $\phi_{\mathcal{B}}(\mathbf{y})$ is then a combination of the vectors in $\mathcal{B}_\gamma$.

$$\phi_{\mathcal{B}}(\mathbf{x}) = \left(\sum_{k=0}^{r} x_{1k}\gamma_k \mid \ldots \mid \sum_{k=0}^{r} x_{nk}\gamma_k\right)$$

$$\phi_{\mathcal{B}}(\mathbf{y}) = \left(\sum_{k=0}^{r} y_{1k}\gamma_k \mid \ldots \mid \sum_{k=0}^{r} y_{nk}\gamma_k\right)$$
$$\mathbf{Q}\cdot\phi_{\mathcal{B}}(\mathbf{x}) = \left(\sum_{k=0}^{r} x_{1k}\cdot\mathbf{Q}\cdot\gamma_k \mid \ldots \mid \sum_{k=0}^{r} x_{nk}\cdot\mathbf{Q}\cdot\gamma_k\right)$$
$$\mathbf{Q}\cdot\phi_{\mathcal{B}}(\mathbf{y}) = \left(\sum_{k=0}^{r} y_{1k}\cdot\mathbf{Q}\cdot\gamma_k \mid \ldots \mid \sum_{k=0}^{r} y_{nk}\cdot\mathbf{Q}\cdot\gamma_k\right)$$

Therefore, $\{\mathbf{Q}\cdot\gamma_1,\ldots,\mathbf{Q}\cdot\gamma_k\}$ is a common support of $\mathbf{Q}*\mathbf{x}$ and $\mathbf{Q}*\mathbf{y}$, which means that $\mathbf{Q}*\mathbf{x}\sim\mathbf{Q}*\mathbf{y}$.

Now, as multiplying a vector to the right by a matrix in $\mathsf{GL}_n(q)$ does not change the support as proved in the previous lemma, we have $\mathbf{Q}*\mathbf{x}\mathbf{P}_1\sim\mathbf{Q}*\mathbf{y}\mathbf{P}_2$. □

Proposition 6. *For all $\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2\in\mathcal{V}$ of rank $r\in\mathbb{N}$ such that $\mathbf{x}_1\sim\mathbf{x}_2$ and $\mathbf{y}_1\sim\mathbf{y}_2$, there exists $\mathbf{Q}\in\mathsf{GL}_m(q)$ and $\mathbf{P}_1,\mathbf{P}_2\in\mathsf{GL}_n(q)$ such that $\mathbf{y}_1=\mathbf{Q}*\mathbf{x}_1\mathbf{P}_1$ and $\mathbf{y}_2=\mathbf{Q}*\mathbf{x}_2\mathbf{P}_2$.*

Proof. Let $\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2\in\mathcal{V}$ or rank $r\in\mathbb{N}$ such that $\mathbf{x}_1\sim\mathbf{x}_2$ and $\mathbf{y}_1\sim\mathbf{y}_2$.

There exists $\mathbf{P}_1\in\mathsf{GL}_n(q)$ such that $\mathbf{x}_1\mathbf{P}_1$ is under systematic form with only the first r columns are non equal to zero. There exists $\mathbf{P}_2\in\mathsf{GL}_n(q)$ such that $\mathbf{x}_2\mathbf{P}_2$ is under systematic form, and its first r columns are then equal to the first r columns of $\mathbf{x}_1\mathbf{P}_1$ and the rest are zero columns.

Similarly, there exists $\mathbf{P}_3\in\mathsf{GL}_n(q)$ such that $\mathbf{y}_1\mathbf{P}_3$ is under systematic form with only the first r columns are non equal to zero. There exists $\mathbf{P}_4\in\mathsf{GL}_n(q)$ such that $\mathbf{y}_2\mathbf{P}_4$ is under systematic form, and its first r columns are then equal to the first r columns of $\mathbf{y}_1\mathbf{P}_3$ and the rest are zero columns.

The first r columns of $\mathbf{x}_1\mathbf{P}_1$ form a basis $\mathcal{B}_1$ of $\mathbb{F}_{q^m}^r$ and the first r columns of $\mathbf{y}_1\mathbf{P}_3$ form a basis $\mathcal{B}_2$ of $\mathbb{F}_{q^m}^r$. Let $\mathbf{Q}$ denotes the matrix which transforms the basis $\mathbf{B}_1$ into $\mathbf{B}_2$.

Then $\mathbf{Q}*\mathbf{x}_1\mathbf{P}_1=\mathbf{y}_1\mathbf{P}_3=\mathbf{y}_2\mathbf{P}_4=\mathbf{Q}*\mathbf{x}_2\mathbf{P}_2$. Finally, $\mathbf{y}_1=\mathbf{Q}*\mathbf{x}_1(\mathbf{P}_1\mathbf{P}_3^{-1})$ and $\mathbf{y}_2=\mathbf{Q}*\mathbf{x}_2(\mathbf{P}_2\mathbf{P}_4^{-1})$

□

Based on those results, we can now achieve a proof of knowledge for a RQC ciphertext. The syndrome equation for a RQC ciphertext has the following shape:

$$\begin{pmatrix}\mathbf{I}_n & \vec{\mathbf{h}} & 0 & 0\\ 0 & \vec{\mathbf{s}} & \mathbf{I}_n & \mathrm{Gab}_{\mathbf{g}}\end{pmatrix}\begin{pmatrix}\mathbf{r}_1\\ \mathbf{r}_2\\ \mathbf{r}_3\\ \mu\end{pmatrix}=\begin{pmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{pmatrix} \tag{1}$$

where $(r_1,r_2,r_3)\in\mathfrak{S}_{w_r}^3(\mathcal{V})$ and $\mu\in\mathbb{F}_{q^m}^k$.

We can now describe the protocol more formally. The common input is a pair $(\mathbf{H},\mathbf{c})$, and the prover's auxiliary input is a vector $\mathbf{r}$ such that $\mathbf{r}=(\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3,\mu)^\top$ and $\mathbf{c}=(\mathbf{c}_1,\mathbf{c}_2)$. $\mathbf{H}$ is splitted into several submatricies $\mathbf{H}=\begin{pmatrix}\mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4\end{pmatrix}$ where:

$$\mathbf{H}_1=\begin{pmatrix}\mathbf{I}_n\\ 0\end{pmatrix}\quad \mathbf{H}_2=\begin{pmatrix}\vec{\mathbf{h}}\\ \vec{\mathbf{s}}\end{pmatrix}\quad \mathbf{H}_3=\begin{pmatrix}0\\ \mathbf{I}_n\end{pmatrix}\quad \mathbf{H}_4=\begin{pmatrix}0\\ \mathrm{Gab}_{\mathbf{g}}\end{pmatrix}$$

Using those notations, the equation (1) is equivalent to $\mathbf{H}\mathbf{r}^\top = \mathbf{c}^\top$. The prover P and the verifier V interact as described in figure 5.

Notations: We denote $a \mid b$ the concatenation of two elements and by $\|_{i=1}^{n} a_i = a_1 \mid \ldots \mid a_n$ the concatenation of a sequence of elements.

1. [Commitment-Step] P samples $\mathbf{Q} \xleftarrow{\$} \mathsf{GL}_m(q)$; $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\mathbf{P}_4 \xleftarrow{\$} \mathsf{GL}_k(q)$; $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \xleftarrow{\$} \mathcal{V}$; $\mathbf{v}_4 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \xleftarrow{\$} 1^{\mathfrak{K}}$ and sends the commitment CMT:=(cm_1, cm_2, cm_3), where
   $cm_1 = \text{Hash}(\mathbf{Q} \mid \|_{i=1}^{4}\mathbf{P}_i \mid \sum_{i=1}^{4} \mathbf{H}_i\mathbf{v}_i^\top \mid \mathbf{z}_1)$
   $cm_2 = \text{Hash}(\|_{i=1}^{4}\mathbf{Q} * \mathbf{v}_i\mathbf{P}_i \mid \mathbf{z}_2)$
   $cm_3 = \text{Hash}(\|_{i=1}^{3}\ \mathbf{Q} * (\mathbf{v}_i + \mathbf{r}_i)\mathbf{P}_i \mid \mathbf{Q} * (\mathbf{v}_4 + \mu)\mathbf{P}_4 \mid \mathbf{z}_3)$
2. [Challenge-Step] V sends a random challenge $ch \in \{0, 1, 2\}$ to P.
3. [Answer-Step] P replies as follows:
   - ◇ If $ch = 0$, let $\mathbf{E} = \mathbf{Q}$, $\forall i \in [\![1, 4]\!]$, let $\mathbf{a}_i = \mathbf{v}_i$, $\mathbf{F}_i = \mathbf{P}_i$.
     Send RSP:=$(\mathbf{E} \mid \|_{i=1}^{4}\mathbf{F}_i \mid \|_{i=1}^{4}\mathbf{a}_i \mid \mathbf{z}_1 \mid \mathbf{z}_2)$
   - ◇ If $ch = 1$, let $\mathbf{I} = \mathbf{Q}$, $\mathbf{b}_4 = \mathbf{v}_4 + \mu$, $\forall i \in [\![1, 3]\!]$, let $\mathbf{b}_i = \mathbf{v}_i + \mathbf{r}_i$, $\mathbf{J}_i = \mathbf{P}_i$.
     Send RSP:=$(\mathbf{I} \mid \|_{i=1}^{4}\mathbf{J}_i \mid \|_{i=1}^{4}\mathbf{b}_i \mid \mathbf{z}_1 \mid \mathbf{z}_3)$
   - ◇ If $ch = 2$, $\forall i \in [\![1, 4]\!]$, let $\mathbf{c}_i = \mathbf{Q} * \mathbf{v}_i\mathbf{P}_i$, $\forall i \in [\![1, 3]\!]$, let $\mathbf{d}_i = \mathbf{Q} * \mathbf{r}_i\mathbf{P}_i$ and let $\mathbf{d}_4 = \mathbf{Q} * \mu\mathbf{P}_4$.
     Send RSP:=$(\|_{i=1}^{4}\mathbf{c}_i \mid \|_{i=1}^{4}\mathbf{d}_i \mid \mathbf{z}_2 \mid \mathbf{z}_3)$
4. [Verification-Step], V performs the following checks:
   - ◇ If $ch = 0$, check that
     $cm_1 = \text{Hash}(\mathbf{E} \mid \|_{i=1}^{4}\mathbf{F}_i \mid \sum_{i=1}^{4} \mathbf{H}_i\mathbf{a}_i^\top \mid \mathbf{z}_1)$
     $cm_2 = \text{Hash}(\|_{i=1}^{4}\mathbf{E} * \mathbf{a}_i\mathbf{F}_i \mid \mathbf{z}_2)$
   - ◇ If $ch = 1$, check that
     $cm_1 = \text{Hash}(\mathbf{I} \mid \|_{i=1}^{4}\mathbf{J}_i \mid \sum_{i=1}^{4} \mathbf{H}_i\mathbf{b}_i^\top - \mathbf{c}^\top \mid \mathbf{z}_1)$
     $cm_3 = \text{Hash}(\|_{i=1}^{4}\mathbf{I} * \mathbf{b}_i\mathbf{J}_i \mid \mathbf{z}_3)$
   - ◇ If $ch = 2$, check that
     $cm_2 = \text{Hash}(\|_{i=1}^{4}\mathbf{c}_i \mid \mathbf{z}_2)$
     $cm_3 = \text{Hash}(\|_{i=1}^{4}\mathbf{c}_i + \mathbf{d}_i \mid \mathbf{z}_3)$
     $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3) \in \mathfrak{S}_{w_r}^3(\mathcal{V})$

V outputs the decision $d = 1$ (Accept) if all checks are passed.
Otherwise he outputs $d = 0$ (Reject).

Fig. 5. Proof of validity for a single RQC ciphertext

Definition 25. *An interactive protocol between two PPT machines P and V is statistical zero-knowledge if, for every PPT machine $\tilde{V}$, there exists a machine*

S which generates, in expected polynomial time, an output having a distribution statistically indistinguishable from the content of the communication tape produced during the interaction of P and $\tilde{V}$.

Theorem 3. *The protocol depicted in figure 5 is a statistical zero-knowledge proof in the random oracle model.*

Theorem 4. *If there exists a PPT cheating prover $\widetilde{\mathsf{P}}$ who convinces the verifier with probability $\frac{2}{3}+\varepsilon$, where ε is non-negligible, then there exists a PPT knowledge extractor who outputs with overwhelming probability a tuple $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4)$ such that $\sum_{i=1}^{4} \mathbf{H}_i \mathbf{y}_i^\top = \mathbf{c}^\top$ with $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$.*

Proof. The proofs are deferred to Appendices 1 and 2.

As a corollary of Theorem 4, the protocol has soundness equal to $\frac{2}{3}$, based on the hardness of the search IRSD problem for a $[4n, 2n]_{q^m}$ code.

5.2 Proof of RQC Ciphertexts Validity for Identical Plaintexts

For the PAKE construction in section 6.2, we need to derive from the protocol in figure 5 a proof that two ciphertexts under two different keys of the $\mathsf{RQC.PKE}$ cryptosystem encrypt the same message μ. For a given message μ, let $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ and $\mathbf{c}' = (\mathbf{c}_3, \mathbf{c}_4)$ be two valid ciphertexts of μ under the key pairs denoted by $(\mathsf{pk}, \mathsf{sk})$ and $(\mathsf{pk}', \mathsf{sk}')$ where $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$, $\mathsf{pk} = (\mathbf{g}, \mathbf{h}, \mathbf{s})$ with $\mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}$, and $\mathsf{sk}' = (\mathbf{x}', \mathbf{y}')$, $\mathsf{pk}' = (\mathbf{g}, \mathbf{h}', \mathbf{s}')$ with $\mathbf{s}' = \mathbf{x}' + \mathbf{h}' \cdot \mathbf{y}'$. One can see that a proof that the two ciphertexts $\mathbf{c}$ and $\mathbf{c}'$ encrypt the same message μ is a proof of knowledge of the vectors $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$, $\mathbf{r}' = (\mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6)$ and μ such that the following relation is satisfied:

$$\mathcal{R} = \begin{cases} \mathbf{c_1} = \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2 \\ \mathbf{c_2} = \mu \mathrm{Gab}_{\mathbf{g}} + \mathbf{s}\mathbf{r}_2 + \mathbf{r}_3 \end{cases} \Bigg| \; (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V}) \\ \begin{cases} \mathbf{c_3} = \mathbf{r}_4 + \mathbf{h}'\mathbf{r}_5 \\ \mathbf{c_4} = \mu \mathrm{Gab}_{\mathbf{g}} + \mathbf{s}'\mathbf{r}_5 + \mathbf{r}_6 \end{cases} \Bigg| \; (\mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$$

Let $\tilde{\mathbf{H}}$ denotes the following matrix:

$$\tilde{\mathbf{H}} = \begin{pmatrix} \mathbf{I}_n & \vec{\mathbf{h}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \vec{\mathbf{s}} & \mathbf{I}_n & 0 & 0 & 0 & \mathrm{Gab}_{\mathbf{g}} \\ 0 & 0 & 0 & \mathbf{I}_n & \vec{\mathbf{h}'} & 0 & 0 \\ 0 & 0 & 0 & 0 & \vec{\mathbf{s}'} & \mathbf{I}_n & \mathrm{Gab}_{\mathbf{g}} \end{pmatrix},$$

and let $\mathbf{d} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4)$. Let $\tilde{\mathbf{r}} = (\mathbf{r}_1, \mathbf{r}_2 \dots \mathbf{r}_6, \mu)$ be a witness for the relation $\mathcal{R}$ such that $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$, $(\mathbf{r}_4, \mathbf{r}_5, \mathbf{r}_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$, and $\mu \in \mathbb{F}_{q^m}^k$. Then, we have that $\tilde{\mathbf{H}}\tilde{\mathbf{r}}^\top = \mathbf{d}^\top$.

Therefore, $\tilde{\mathbf{r}}$ is a witness for the protocol given in figure 5 when it is instantiated using $(\tilde{\mathbf{H}}, \mathbf{d})$. In figure 8 (see Appendix 3) we present our interactive proof

that two ciphertexts under two different keys encrypt the same message μ. The protocol takes as input the couple $(\tilde{\mathbf{H}}, \mathbf{d})$ where $\tilde{\mathbf{H}} = (\|_{i=1}^{7} \tilde{\mathbf{H}}_i)$.

Theorem 5. *The protocol depicted in figure 8 is a statistical zero-knowledge proof in the random oracle model.*

Theorem 6. *If there exists a PPT cheating prover $\widetilde{\mathsf{P}}$ who convinces the verifier with probability $\frac{2}{3} + \varepsilon$, where ε is non-negligible, then there exists a PPT knowledge extractor who outputs with overwhelming probability a tuple $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_7)$ such that $\sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{y}_i^\top = \mathbf{c}^\top$, $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \in \mathfrak{S}_{w_r}^3(\mathcal{V})$ and $(\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6) \in \mathfrak{S}_{w_r}^3(\mathcal{V})$.*

Proof. The proofs are deferred to Appendices 4 and 5.

Non-interactive protocol. We apply the Fiat-Shamir transformation [22, 40] in order to make the protocol non-interactive in the random oracle model. Let us consider a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1,2\}^\kappa$ (where $\kappa = \omega(\log \mathfrak{K})$) that is modeled as a random oracle. Therefore, we obtain the following non-interactive proof $\mathbf{\Pi} = (\|_{i=1}^{\kappa} \mathrm{CMT}^{(i)} \mid \|_{i=1}^{\kappa} ch^{(i)} \mid \|_{i=1}^{\kappa} \mathrm{RSP}^{(i)})$, where $\|_{i=1}^{\kappa} ch^{(i)} = \mathcal{H}(M \mid \|_{i=1}^{\kappa} \mathrm{CMT}^{(i)})$ and M is a random message.

6 Applications

Hash Proof Systems can be used as a primitive to build many cryptographic protocols. Hereafter, we describe a Witness Encryption (WE) scheme and a Password Authenticated Key Exchange (PAKE) based on the HPS described in figure 4. We stress that the above constructions can be adapted in the UC setting. It would require the use of an extractable/equivocable commitment scheme which can be achieved, with no innovating details, using a code-based Haralambiev like commitment as in [17]. We choose to not introduce the UC framework in order to focus on the main results of this paper, namely the construction of a code-based HPS and the crucial role of the gap upon the security of this primitive.

6.1 Witness Encryption

The concept of *witness encryption* (WE) was introduced by Garg et al. [28] and allows to encrypt a message μ using a word $W \in \mathcal{L}$ so that the knowledge of the witness w for the membership of W in $\mathcal{L}$ is required to decrypt the ciphertext. A common strategy [1, 15] for constructing WE schemes consists to use an exact HPS in the following way: given a word W and a message μ, a sender generates a hashing key hk, a projection key hp, a hash value $\mathcal{H}_{\mathsf{hk}}$ and masks the message μ using $\mathcal{H}_{\mathsf{hk}}$. In order to decrypt the ciphertext, the recipient uses the witness w associated with the word W along with the projection key hp to compute the projected hash value $\mathcal{H}_{\mathsf{hp}}$ and retrieve μ.

The figure 6 describes a construction that achieve a similar result using our approximate HPS. The main idea is to mask $\mu \mathrm{Gab}_{\mathbf{g}}$ rather than μ in order to be able to remove the noise introduced by our HPS using a decoding algorithm.

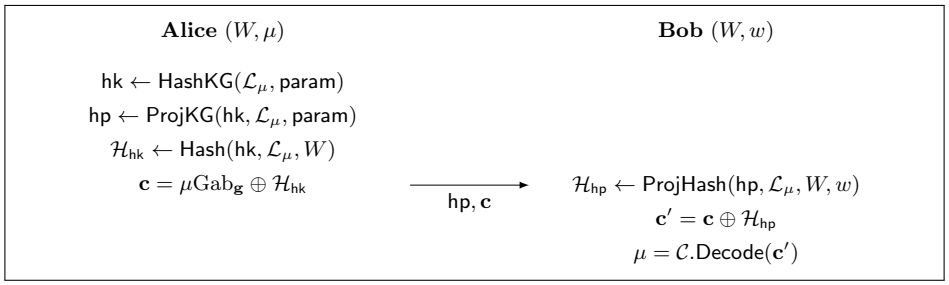

Fig. 6. Witness Encryption from a code-based approximate HPS

6.2 Password Authenticated Key Exchange

Password Authenticated Key Exchange (PAKE) was first introduced by Bellovin and Merritt [12]. The aim of such protocols is to allow users to generate a strong cryptographic key based on a shared "human memorable" password pw without requiring a public-key infrastructure. In this setting, an adversary controlling all the communication in the network should not be able to mount an off-line dictionary attack. HPS offers an interesting edge to construct such schemes. Several papers present PAKE in the lattice-based field [33, 43, 15] however we are the first, to the best of our knowledge, to present a PAKE based on coding theory.

[14] proposed a construction allowing to build BPR [11] secure one-round PAKE. Their construction requires each user to send a CCA-2 encryption of their password together with the projection key for a KV-HPS for the language of valid encryption of the expected password.

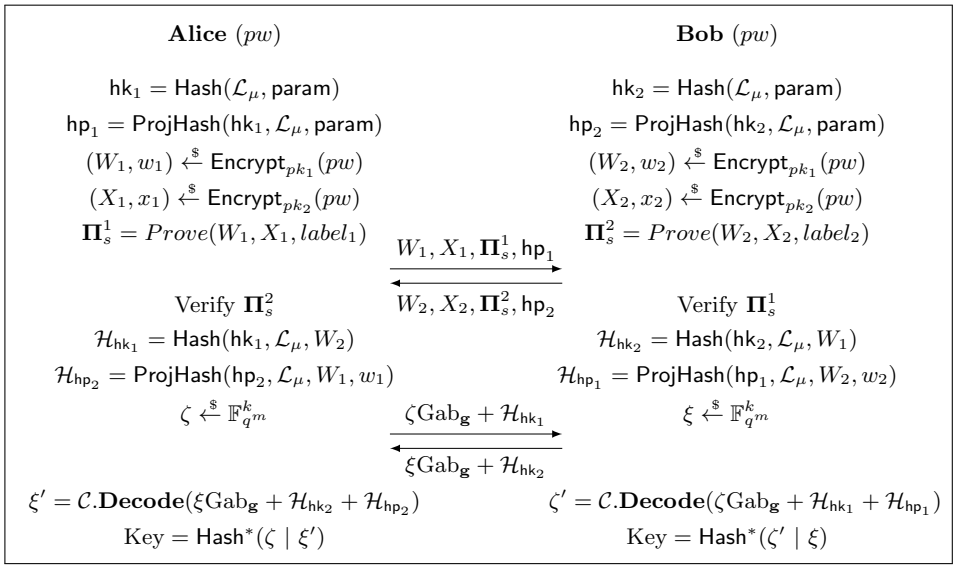

Fig. 7. Two rounds Password Authenticated Key Exchange

We have shown in section 3 how to build such HPS. Using the Naor Yung transform [37], one can build a CCA-2 encryption of pw with two CPA encryption of pw under different keys along with a simulation-sound zero knowledge proof that the ciphertexts are well-formed and decrypt to the same plaintext. We have shown how one can use a Stern-like protocol to build such proof in section 5.

Using those building blocks, one directly obtains the protocol described in figure 7. The first flow corresponds to the proper PAKE protocol while the last one is the reconciliation phase. If one wants to drop the asynchronous design, one can obtain a three flow protocol by merging the two flows from Bob.

7 Parameters and Performances

We give some parameters for the protocols described in section 6. Those parameters are valid for both protocols and hence are not optimized for any particular application. The security of the constructions depends on the following instances:

Attack on the hashing key hk: in order to retrieve hk from hp one has to consider the following 2-IRSD problem

$$\begin{pmatrix} \vec{\mathbf{s}} & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \end{pmatrix} = \mathsf{hp}$$

Attack on the witness values: in order to guarantee the secrecy of the $(\mathbf{r_i})_{i\in[\![1,3]\!]}$ values, one has to consider the following instance of the 3-IRSD problem:

$$\begin{pmatrix} \mathbf{I}_n & 0 & \vec{\mathbf{h}} \\ 0 & \mathbf{I}_n & \vec{\mathbf{s}} \end{pmatrix} \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_3 \\ \mathbf{r}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 - \mu\mathrm{Gab}_{\mathbf{g}} \end{pmatrix}$$

Smoothness related: the proof of the KV-smoothness involve two different problems. The first one is an instance of a 2-IRSD problem:

$$\begin{pmatrix} \mathbf{I}_n & \vec{\mathbf{h}} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \mathbf{s}$$

The second one is the following instance of the FIRSD problem:

$$\begin{pmatrix} \mathbf{s} & \mathbf{I}_n & 0 \\ \mathbf{c}_2 - \mu\mathrm{Gab}_{\mathbf{g}} & 0 & \mathbf{I}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_2 \\ \boldsymbol{\alpha}_3 \end{pmatrix} = \begin{pmatrix} \mathsf{hp} \\ \mathcal{H}_{\mathsf{hk}} \end{pmatrix}$$

Decoding capability of the Gabidulin code: Finally, in order to decode Gabidulin codewords, we need to have $m \geqslant n$ and $\omega\left(\mathbf{x}\cdot\mathbf{r}_2 - \mathbf{y}\cdot\mathbf{r}_1 + \mathbf{r}_3\right) \leq \left\lfloor \frac{n-k}{2} \right\rfloor$, which is equivalent to $(\omega_x + 1)\cdot\omega_r \leq \left\lfloor \frac{n-k}{2} \right\rfloor$.

The two main approaches used to solve the rank syndrome decoding problem are combinatorial and algebraïc attacks. While combinatorial attacks were

always the best approach [25, 8], recent improvements have been made in algebraïc ones [9, 10]. We will rely on [10] for the calculation of the parameters. We have only described complexities for $[3n, n]$ codes as the relative complexities for $[2n, n]$ codes are always greater in this case.

- hyb3n(a): hybrid attack for a code of length $3n$, a is defined as the smallest value to reach the overdetermined case, $a = 0$ meaning that parameters are already in the overdetermined case.
- hyb2n(a): homogeneous hybrid attack for a code of length $2n$, a defined the same way as above.
- und2n: homogeneous underminated attack for a code of length $2n$, b defined as the smallest positive integer such that the number of unknowns is lower than the number of equations.
- und3n: underminated attack for a code of length $3n$. b defined as the smallest positive integer such that the number of unknowns is lower than the number of equations.
- comb3n: combinatorial attack for a code of length $3n$.

Table 3. Parameters and associated security, homogeneous case.

Instance	q	n	m	k	w_r	w_x	w_α	Vector length	Security
I	2	137	139	4	7	8	11	19 043	128
II	2	211	223	4	10	9	12	47 053	192
III	2	283	293	3	13	10	15	82 919	256

Table 4. Security levels and associated complexities, homogeneous case.

Instance	hybr3n(a)	und3n(b)	comb3n	Security
I	138(0)	162(1)	220	128
II	201(0)	229(1)	560	192
III	264(0)	293(1)	1 028	256

The previous table show that the vectors length are quite practical. For the PAKE protocol nevertheless, one has to consider lengths approximately hundred times larger for the use of the Stern's proofs of knowledge from section 5 along with a Fiat Shamir transformation.

Notice that it is possible to considerably decrease those parameters by considering non-homogeneous version of the rank syndrome decoding problem, as it is done in the last version of RQC [4] (specification of April, 2020). In that scenario, r_1, r_2, r_3 would no longer be all of the same support, but instead we would have $\mathsf{Supp}(r_1) = \mathsf{Supp}(r_2) \subset \mathsf{Supp}(r_3)$ with $\omega(r_1) = \omega(r_2) = w_1$ and

$\omega(r_3) = w_1 + w_2$. Similarly, $\alpha_1, \alpha_2, \alpha_3$ would not be all of the same support, $\mathsf{Supp}(\alpha_1) = \mathsf{Supp}(\alpha_2) \subset \mathsf{Supp}(\alpha_3)$ with $\omega(\alpha_1) = \omega(\alpha_2) = w_1$ and $\omega(\alpha_3) = w_1 + w_2$. We have described the homogenous version for the sake of clarity and simplicity. For the calculation of the parameters in the non-homogeneous cases, we refer to [4].

Table 5. Parameters and associated security, non-homogeneous case.

Instance	q	n	m	k	w_1	w_2	w_x	Vector length	Security
I	2	127	131	3	7	6	7	16 637	128
II	2	163	167	5	8	8	8	27 221	192
III	2	181	191	3	9	7	9	34 571	256

Table 6. Security levels and associated complexities, non-homogeneous case.

Instance	hybr3n(a)	und3n(b)	hybr2n(a)	und2n(b)	Security
I	216(0)	240(1)	156(5)	151(1)	128
II	268(0)	293(1)	325(23)	228(3)	192
III	295(2)	304(1)	545(43)	300(5)	256

8 Conclusion

Hash proof systems are an attractive and powerful tool to build many cryptographic primitives including CCA-2 secure encryption schemes, authenticated key exchange, oblivious transfer, zero-knowledge arguments or witness encryption.

In this paper, we have focused on the construction of a quantum-resistant HPS and we have answered positively two open questions: whether it is possible to design a code-based HPS in the rank metric, and whether it is possible to design a quantum-resistant gapless HPS. In order to provide this gapless construction, we have chosen to define our HPS over a set of valid ciphertexts and to check whether a word is a valid ciphertext using a Stern-like proof of ciphertext validity. However, the conception of a gapless post-quantum HPS without the use of a proof of knowledge to supplement it is still an open question. As an application of this HPS, we have presented witness encryption in the standard model. In addition, we have also presented, in the random oracle model, a PAKE that is secure in the BPR model.

Finally, as a further work, even if the two metrics have real differences, our construction could probably be adapted to the Hamming setting using the HQC

cryptosystem [3], the Hamming version of the RQC scheme, a candidate that have been selected for the third and last round of the NIST post-quantum cryptography standardization project in the alternate candidates category.

A Appendices

Appendix 1 - Proofs of theorem 3: The protocol depicted in figure 5 is a statistical zero-knowledge proof in the random oracle model.

Proof. The proof uses techniques in the same spirit of those in [41, 35, 36]. Therefore, we construct a simulator S which, given the public inputs of the protocol and interacting with a cheating verifier $\widetilde{\mathsf{V}}$, outputs a simulated transcript with probability $\frac{2}{3}$ that is statistically close to the distribution of the real transcript. The public inputs are $(\mathbf{H}, \mathbf{c})$, the simulator S starts by choosing a random $\bar{ch} \in \{0, 1, 2\}$, which is a prediction of the challenge that $\widetilde{\mathsf{V}}$ will not choose.

$\diamondsuit$ **Case** $\bar{ch} = 0$**:**

S samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$ $\quad$ $\mathbf{P}'_1, \mathbf{P}'_2, \mathbf{P}'_3 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\quad$ $\mathbf{P}'_4 \xleftarrow{\$} \mathsf{GL}_k(q)$;

$\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3 \xleftarrow{\$} \mathcal{V}$; $\quad$ $\mathbf{v}'_4 \xleftarrow{\$} \mathbb{F}_{q^m}^k$;

$(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \xleftarrow{\$} \mathfrak{S}_w^3(\mathcal{V})$; $\quad$ $\mathbf{r}'_4 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\quad$ $\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^4 \mathbf{P}'_i \mid \sum_{i=1}^4 \mathbf{H}_i({\mathbf{v}'_i}^\top + {\mathbf{r}'_i}^\top) \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^4 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^4 \mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Output $\perp$ and abort.
- $\diamond$ If $ch = 1$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^4 \mathbf{P}'_i \mid \|_{i=1}^4 \mathbf{v}'_i + \mathbf{r}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_3)$
- $\diamond$ If $ch = 2$, Send RSP:=$(\|_{i=1}^4 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \|_{i=1}^4 \mathbf{Q}' * \mathbf{r}'_i \mathbf{P}'_i \mid \mathbf{z}'_2 \mid \mathbf{z}'_3)$

$\diamondsuit$ **Case** $\bar{ch} = 1$**:**

S samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$; $\quad$ $\mathbf{P}'_1, \mathbf{P}'_2, \mathbf{P}'_3 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\quad$ $\mathbf{P}'_4 \xleftarrow{\$} \mathsf{GL}_k(q)$;

$\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3 \xleftarrow{\$} \mathcal{V}$; $\quad$ $\mathbf{v}'_4 \xleftarrow{\$} \mathbb{F}_{q^m}^k$;

$(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \xleftarrow{\$} \mathfrak{S}_w^3(\mathcal{V})$; $\quad$ $\mathbf{r}'_4 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\quad$ $\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^4 \mathbf{P}'_i \mid \sum_{i=1}^4 \mathbf{H}_i {\mathbf{v}'_i}^\top \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^4 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^4 \mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{4}\mathbf{P}'_i \mid \|_{i=1}^{4}\mathbf{v}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_2)$
- $\diamond$ If $ch = 1$, Output $\perp$ and abort.
- $\diamond$ If $ch = 2$, Send RSP:=$(\|_{i=1}^{4}\mathbf{Q}' * \mathbf{v}'_i\mathbf{P}'_i \mid \|_{i=1}^{4}\mathbf{Q}' * \mathbf{r}'_i\mathbf{P}'_i \mid \mathbf{z}'_2 \mid \mathbf{z}'_3)$

$\diamondsuit$ **Case** $\bar{ch} = 2$**:**

Using linear algebra, S computes a vector $\mathbf{r}' = (\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3, \mathbf{r}'_4)$, such that $\mathbf{H}{\mathbf{r}'}^{\top} = \mathbf{c}^{\top}$.
Next, it samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$; $\quad \mathbf{P}'_1, \mathbf{P}'_2, \mathbf{P}'_3 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\quad \mathbf{P}'_4 \xleftarrow{\$} \mathsf{GL}_k(q)$;
$\mathbf{v}'_1, \mathbf{v}'_2, \mathbf{v}'_3 \xleftarrow{\$} \mathcal{V}$; $\quad \mathbf{v}'_4 \xleftarrow{\$} \mathbb{F}_{q^m}^{k}$; $\quad \mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^{4}\mathbf{P}'_i \mid \textstyle\sum_{i=1}^{4}\mathbf{H}_i{\mathbf{v}'_i}^{\top} \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^{4}\mathbf{Q}' * \mathbf{v}'_i\mathbf{P}'_i \mid \mathbf{P}'_4 \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^{4}\mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{4}\mathbf{P}'_i \mid \|_{i=1}^{4}\mathbf{v}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_2)$
- $\diamond$ If $ch = 1$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{4}\mathbf{P}'_i \mid \|_{i=1}^{4}\mathbf{v}'_i + \mathbf{r}'_i \mid \mathbf{z}_1 \mid \mathbf{z}_3)$
- $\diamond$ If $ch = 2$, Output $\perp$ and abort.

It can be seen that the probability that the simulator outputs $\perp$ is close to $\frac{1}{3}$. Additionally, when the simulator does not halt, the distribution of the generated transcripts is statistically close to the distribution of the real transcript when the hash function is modeled as a random oracle. Therefore, we have build a simulator that succeeds the protocol with probability $\frac{2}{3}$ without having any information about the secret values.

□

Appendix 2 - Proof of theorem 4: If there exists a PPT cheating prover $\widetilde{\mathsf{P}}$ who convinces the verifier with probability $\frac{2}{3}+\varepsilon$, where ε is non-negligible, then there exists a PPT knowledge extractor who outputs with overwhelming probability a tuple $(\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3,\mathbf{y}_4)$ such that $\sum_{i=1}^{4}\mathbf{H}_i\mathbf{y}_i^{\top}=\mathbf{c}^{\top}$ with $(\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3)\in\mathfrak{S}^3_{w_r}(\mathcal{V})$.

Proof. We show how to construct a knowledge extractor $\mathcal{K}$. Let $\widetilde{\mathsf{P}}$ be the cheating prover who convinces the verifier with probability $\frac{2}{3}+\varepsilon$. Applying the technique of Véron [42], that rewinds $\widetilde{\mathsf{P}}$ a number of times polynomial in $\frac{1}{\varepsilon}$, the knowledge extractor can obtain with overwhelming probability a commitment, for which $\widetilde{\mathsf{P}}$ can correctly answer all three challenges. Therefore, $\mathcal{K}$ obtains the following equations:

$$cm_1=\mathrm{Hash}(\mathbf{E}\mid\|_{i=1}^{4}\mathbf{F}_i\mid\sum_{i=1}^{4}\mathbf{H}_i\mathbf{a}_i^{\top})=\mathrm{Hash}(\mathbf{I}\mid\|_{i=1}^{4}\mathbf{J}_i\mid\sum_{i=1}^{4}\mathbf{H}_i\mathbf{b}_i^{\top}-\mathbf{c}^{\top})$$
$$cm_2=\mathrm{Hash}(\|_{i=1}^{4}\mathbf{E}*\mathbf{a}_i\mathbf{F}_i)=\mathrm{Hash}(\|_{i=1}^{4}\mathbf{c}_i)$$
$$cm_3=\mathrm{Hash}(\|_{i=1}^{4}\mathbf{I}*\mathbf{b}_i\mathbf{J}_i)=\mathrm{Hash}(\|_{i=1}^{4}\mathbf{c}_i+\mathbf{d}_i)$$
$$(\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3)\in\mathfrak{S}^3_{w_r}(\mathcal{V}).$$

Since Hash is modeled as a random oracle (an adversary cannot find a collision on it), it follows that:

- $\mathbf{E}=\mathbf{I}$ and $\forall i\in[\![1,4]\!],\ \mathbf{F}_i=\mathbf{J}_i$ and $\sum_{i=1}^{4}\mathbf{H}_i\mathbf{a}_i^{\top}=\sum_{i=1}^{4}\mathbf{H}_i\mathbf{b}_i^{\top}-\mathbf{c}^{\top}$
- $\forall i\in[\![1,4]\!],\ \mathbf{E}*\mathbf{a}_i\mathbf{F}_i=\mathbf{c}_i,\ \mathbf{I}*\mathbf{b}_i\mathbf{J}_i=\mathbf{c}_i+\mathbf{d}_i$

Let $i\in[\![1,4]\!]$. We have $\mathbf{I}*\mathbf{b}_i\mathbf{J}_i=\mathbf{E}*\mathbf{b}_i\mathbf{F}_i=\mathbf{c}_i+\mathbf{d}_i$. It follows that $\mathbf{E}*(\mathbf{b}_i-\mathbf{a}_i)\mathbf{F}_i=\mathbf{d}_i$, which implies that $(\mathbf{b}_i-\mathbf{a}_i)=\mathbf{E}^{-1}*\mathbf{d}_i\mathbf{F}_i^{-1}$.

$$\forall i\in[\![1,4]\!],\ (\mathbf{b}_i-\mathbf{a}_i)=\mathbf{E}^{-1}*\mathbf{d}_i\mathbf{F}_i^{-1}$$

Since $\mathbf{E}^{-1}\in\mathsf{GL}_m(q)$ and $\mathbf{F}_i^{-1}\in\mathsf{GL}_m(q)$, we have $(\mathbf{b}_1-\mathbf{a}_1,\mathbf{b}_2-\mathbf{a}_2,\mathbf{b}_3-\mathbf{a}_3)\in\mathfrak{S}^3_{w_r}(\mathcal{V})$. Therefore, the knowledge extractor $\mathcal{K}$ obtains vectors $\mathbf{y}_i=\mathbf{b}_i-\mathbf{a}_i$, with $i\in[\![1,4]\!]$, such that: $\sum_{i=1}^{4}\mathbf{H}_i\mathbf{y}_i^{\top}=\mathbf{c}^{\top}$ and $(\mathbf{y}_1,\mathbf{y}_2,\mathbf{y}_3)\in\mathfrak{S}^3_{w_r}(\mathcal{V})$.

□

Appendix 3 - Proof of RQC ciphertexts validity for identical plaintexts:

1. [Commitment-Step] P samples $\mathbf{Q} \xleftarrow{\$} \mathsf{GL}_m(q)$; $\{\mathbf{P}_i\}_{i\in[\![1,6]\!]} \xleftarrow{\$} \mathsf{GL}_n(q)$; $\mathbf{P}_7 \xleftarrow{\$} \mathsf{GL}_k(q)$; $\{\mathbf{v}_i\}_{i\in[\![1,6]\!]} \xleftarrow{\$} \mathcal{V}$; $\mathbf{v}_7 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3 \xleftarrow{\$} 1^{\lambda}$ and sends the commitment CMT:=$(cm_1 \mid cm_2 \mid cm_3)$, where:
   $cm_1 = \text{Hash}(\mathbf{Q} \mid \|_{i=1}^{7}\mathbf{P}_i \mid \sum_{i=1}^{7} \tilde{\mathbf{H}}_i\mathbf{v}_i^\top \mid \mathbf{z}_1)$
   $cm_2 = \text{Hash}(\|_{i=1}^{7}\mathbf{Q} * \mathbf{v}_i\mathbf{P}_i \mid \mathbf{z}_2)$
   $cm_3 = \text{Hash}(\|_{i=1}^{6}\mathbf{Q} * (\mathbf{v}_i + \mathbf{r}_i)\mathbf{P}_i \mid \mathbf{Q} * (\mathbf{v}_7 + \mu)\mathbf{P}_7 \mid \mathbf{z}_3)$

2. [Challenge-Step] V sends a random challenge $ch \in \{0, 1, 2\}$ to P.

3. [Answer-Step] P replies as follows:
   - If $ch = 0$, let $\mathbf{E} = \mathbf{Q}$, $\forall i \in [\![1, 7]\!]$, let $\mathbf{a}_i = \mathbf{v}_i$, $\mathbf{F}_i = \mathbf{P}_i$.
     Send RSP:=$(\mathbf{E} \mid \|_{i=1}^{7}\mathbf{F}_i \mid \|_{i=1}^{7}\mathbf{a}_i \mid \mathbf{z}_1 \mid \mathbf{z}_2)$
   - If $ch = 1$, let $\mathbf{I} = \mathbf{Q}$, $\forall i \in [\![1, 6]\!]$ let $\mathbf{b}_i = \mathbf{v}_i + \mathbf{r}_i$, $\mathbf{b}_7 = \mathbf{v}_7 + \mu$ and $\forall i \in [\![1, 7]\!]$ let $\mathbf{J}_i = \mathbf{P}_i$.
     Send RSP:=$(\mathbf{I} \mid \|_{i=1}^{7}\mathbf{J}_i \mid \|_{i=1}^{7}\mathbf{b}_i \mid \mathbf{z}_1 \mid \mathbf{z}_3)$
   - If $ch = 2$, $\forall i \in [\![1, 7]\!]$ let $\mathbf{c}_i = \mathbf{Q} * \mathbf{v}_i\mathbf{P}_i$, $\forall i \in [\![1, 6]\!]$ let $\mathbf{d}_i = \mathbf{Q} * \mathbf{r}_i\mathbf{P}_i$ and $\mathbf{d}_7 = \mathbf{Q} * \mu\mathbf{P}_7$.
     Send RSP:=$(\|_{i=1}^{7}\mathbf{c}_i \mid \|_{i=1}^{7}\mathbf{d}_i \mid \mathbf{z}_2 \mid \mathbf{z}_3)$

4. [Verification-Step], V performs the following checks:
   - If $ch = 0$, check that:
     $cm_1 = \text{Hash}(\mathbf{E} \mid \|_{i=1}^{7}\mathbf{F}_i \mid \sum_{i=1}^{7} \tilde{\mathbf{H}}_i\mathbf{a}_i^\top \mid \mathbf{z}_1)$,
     $cm_2 = \text{Hash}(\|_{i=1}^{7}\mathbf{E} * \mathbf{a}_i\mathbf{F}_i \mid \mathbf{z}_2)$.
   - If $ch = 1$, check that:
     $cm_1 = \text{Hash}(\mathbf{I} \mid \|_{i=1}^{7}\mathbf{J}_i \mid \sum_{i=1}^{7} \tilde{\mathbf{H}}_i\mathbf{b}_i^\top - \mathbf{d} \mid \mathbf{z}_1)$,
     $cm_3 = \text{Hash}(\|_{i=1}^{7}\mathbf{I} * \mathbf{b}_i\mathbf{J}_i \mid \mathbf{z}_3)$.
   - If $ch = 2$, check that:
     $cm_2 = \text{Hash}(\|_{i=1}^{7}\mathbf{c}_i \mid \mathbf{z}_2)$,
     $cm_3 = \text{Hash}(\|_{i=1}^{7}\mathbf{c}_i + \mathbf{d}_i \mid \mathbf{z}_3)$,
     $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3) \in \mathfrak{S}_{w_r}^3(\mathcal{V})$, $(\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) \in \mathfrak{S}_{w_r}^3(\mathcal{V})$.

V outputs $d = 1$ (Accept) if all checks are passed. Otherwise, he outputs $d = 0$ (Reject).

Fig. 8. Proof of RQC ciphertexts validity for identical plaintexts

Appendix 4 - Proof of theorem 5: The protocol depicted in figure 8 is a statistical zero-knowledge proof in the random oracle model.

Proof. The proof uses techniques in the same spirit of those in [41, 35, 36]. Therefore, we construct a simulator S which is given the public inputs of the protocol and interacting with a cheating verifier $\widetilde{\mathsf{V}}$, outputs a simulated transcript with probability $\frac{2}{3}$ that is statistically close to the distribution of the real transcript. The public inputs are $(\mathbf{H}, \mathbf{c})$, the simulator S starts by choosing a random $\bar{ch} \in \{0, 1, 2\}$, which is a prediction of the challenge that $\widetilde{\mathsf{V}}$ will not choose.

$\diamondsuit$ **Case** $\bar{ch} = 0$**:**

S samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$; ; $\mathbf{P}'_1, \mathbf{P}'_2, \ldots, \mathbf{P}'_6 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\mathbf{P}'_7 \xleftarrow{\$} \mathsf{GL}_k(q)$;
$\mathbf{v}'_1, \mathbf{v}'_2, \ldots, \mathbf{v}'_6 \xleftarrow{\$} \mathcal{V}$; $\mathbf{v}'_7 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;
$(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$; $(\mathbf{r}'_4, \mathbf{r}'_5, \mathbf{r}'_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$; $\mathbf{r}'_7 \xleftarrow{\$} \mathbb{F}_{q^m}^k$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^7 \mathbf{P}'_i \mid \sum_{i=1}^7 \mathbf{H}_i({\mathbf{v}'_i}^\top + {\mathbf{r}'_i}^\top) \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^7 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^7 \mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Output $\perp$ and abort.
- $\diamond$ If $ch = 1$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^7 \mathbf{P}'_i \mid \|_{i=1}^7 \mathbf{v}'_i + \mathbf{r}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_3)$
- $\diamond$ If $ch = 2$, Send RSP:=$(\|_{i=1}^7 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \|_{i=1}^7 \mathbf{Q}' * \mathbf{r}'_i \mathbf{P}'_i \mid \mathbf{z}'_2 \mid \mathbf{z}'_3)$

$\diamondsuit$ **Case** $\bar{ch} = 1$

S samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$; $\mathbf{P}'_1, \mathbf{P}'_2, \ldots, \mathbf{P}'_6 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\mathbf{P}'_7 \xleftarrow{\$} \mathsf{GL}_k(q)$;
$\mathbf{v}'_1, \mathbf{v}'_2, \ldots, \mathbf{v}'_6 \xleftarrow{\$} \mathcal{V}$; $\mathbf{v}'_7 \xleftarrow{\$} \mathbb{F}_{q^m}^k$; $\mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;
$(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$; $(\mathbf{r}'_4, \mathbf{r}'_5, \mathbf{r}'_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$; $\mathbf{r}'_7 \xleftarrow{\$} \mathbb{F}_{q^m}^k$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^7 \mathbf{P}'_i \mid \sum_{i=1}^7 \mathbf{H}_i {\mathbf{v}'_i}^\top \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^7 \mathbf{Q}' * \mathbf{v}'_i \mathbf{P}'_i \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^7 \mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{7}\mathbf{P}'_i \mid \|_{i=1}^{7}\mathbf{v}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_2)$
- $\diamond$ If $ch = 1$, Output $\perp$ and abort.
- $\diamond$ If $ch = 2$, Send RSP:=$(\|_{i=1}^{7}\mathbf{Q}' * \mathbf{v}'_i\mathbf{P}'_i \mid \|_{i=1}^{7}\mathbf{Q}' * \mathbf{r}'_i\mathbf{P}'_i \mid \mathbf{z}'_2 \mid \mathbf{z}'_3)$

$\diamondsuit$ **Case** $\bar{ch} = 2$

Using linear algebra, S computes a vector $\mathbf{r}' = (\mathbf{r}'_1, \mathbf{r}'_2 \dots \mathbf{r}'_7)$, such that $\mathbf{H}\mathbf{r}'^{\top} = \mathbf{c}^{\top}$.
Next, it samples:

$\mathbf{Q}' \xleftarrow{\$} \mathsf{GL}_m(q)$; $\quad \mathbf{P}'_1, \mathbf{P}'_2, \dots, \mathbf{P}'_6 \xleftarrow{\$} \mathsf{GL}_n(q)$; $\quad \mathbf{P}'_7 \xleftarrow{\$} \mathsf{GL}_k(q)$;
$\mathbf{v}'_1, \mathbf{v}'_2, \dots, \mathbf{v}'_6 \xleftarrow{\$} \mathcal{V}$; $\quad \mathbf{v}'_7 \xleftarrow{\$} \mathbb{F}_{q^m}^{k}$; $\quad \mathbf{z}'_1, \mathbf{z}'_2, \mathbf{z}'_3 \xleftarrow{\$} 1^{\mathfrak{K}}$;

and sends the commitment CMT:=$(cm'_1 \mid cm'_2 \mid cm'_3)$ to $\widetilde{\mathsf{V}}$, where:

$$cm'_1 = \text{Hash}(\mathbf{Q}' \mid \|_{i=1}^{7}\mathbf{P}'_i \mid \textstyle\sum_{i=1}^{7}\mathbf{H}_i\mathbf{v}'^{\top}_i \mid \mathbf{z}'_1)$$
$$cm'_2 = \text{Hash}(\|_{i=1}^{7}\mathbf{Q}' * \mathbf{v}'_i\mathbf{P}'_i \mid \mathbf{P}'_7 \mid \mathbf{z}'_2)$$
$$cm'_3 = \text{Hash}(\|_{i=1}^{7}\mathbf{Q}' * (\mathbf{v}'_i + \mathbf{r}'_i)\mathbf{P}'_i \mid \mathbf{z}'_3)$$

Receiving a challenge ch from $\widetilde{\mathsf{V}}$, the simulator S responds as follows:

- $\diamond$ If $ch = 0$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{7}\mathbf{P}'_i \mid \|_{i=1}^{7}\mathbf{v}'_i \mid \mathbf{z}'_1 \mid \mathbf{z}'_2)$
- $\diamond$ If $ch = 1$, Send RSP:=$(\mathbf{Q}' \mid \|_{i=1}^{7}\mathbf{P}'_i \mid \|_{i=1}^{7}\mathbf{v}'_i + \mathbf{r}'_i \mid \mathbf{z}_1 \mid \mathbf{z}_3)$
- $\diamond$ If $ch = 2$, Output $\perp$ and abort.

It can be seen that the probability that the simulator outputs $\perp$ is close to $\frac{1}{3}$. Additionally, when the simulator does not halt, the distribution of the generated transcripts is statistically close to the distribution of the real transcript when the hash function is modeled as a random oracle. Therefore, we have build a simulator that succeeds the protocol with probability $\frac{2}{3}$ without having any information about the secret values.

□

Appendix 5 - Proof of theorem 6: If there exists a PPT cheating prover $\widetilde{\mathsf{P}}$ who convinces the verifier with probability $\frac{2}{3}+\varepsilon$, where ε is non-negligible, then there exists a PPT knowledge extractor who outputs with overwhelming probability a tuple $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_7)$ such that $\sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{y}_i^\top = \mathbf{c}^\top$, $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$ and $(\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$.

Proof. We show how to construct a knowledge extractor $\mathcal{K}$. Let $\widetilde{\mathsf{P}}$ be the cheating prover who convinces the verifier with probability $\frac{2}{3}+\varepsilon$. Applying the technique of Véron [42], that rewinds $\widetilde{\mathsf{P}}$ a number of times polynomial in $\frac{1}{\varepsilon}$, the knowledge extractor can obtain with overwhelming probability a commitment, for which $\widetilde{\mathsf{P}}$ can correctly answer all three challenges. Therefore, $\mathcal{K}$ obtains the following equations:

$$cm_1 = \text{Hash}(\mathbf{E} \mid \|_{i=1}^{7} \mathbf{F}_i \mid \sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{a}_i^\top) = \text{Hash}(\mathbf{I} \mid \|_{i=1}^{7} \mathbf{J}_i \mid \sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{b}_i^\top - \mathbf{c}^\top)$$
$$cm_2 = \text{Hash}(\|_{i=1}^{7} \mathbf{E} * \mathbf{a}_i \mathbf{F}_i) = \text{Hash}(\|_{i=1}^{7} \mathbf{c}_i)$$
$$cm_3 = \text{Hash}(\|_{i=1}^{7} \mathbf{I} * \mathbf{b}_i \mathbf{J}_i) = \text{Hash}(\|_{i=1}^{7} \mathbf{c}_i + \mathbf{d}_i)$$
$$(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V}),\ (\mathbf{d}_4, \mathbf{d}_5, \mathbf{d}_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V}).$$

Since Hash is modeled as a random oracle (an adversary cannot find a collision on it), it follows that:

- $\mathbf{E} = \mathbf{I}$ and $\forall i \in [\![1, 7]\!],\ \mathbf{F}_i = \mathbf{J}_i$ and $\sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{a}_i^\top = \sum_{i=1}^{4} \tilde{\mathbf{H}}_i \mathbf{b}_i^\top - \mathbf{c}^\top$
- $\forall i \in [\![1, 7]\!],\ \mathbf{E} * \mathbf{a}_i \mathbf{F}_i = \mathbf{c}_i,\ \mathbf{I} * \mathbf{b}_i \mathbf{J}_i = \mathbf{c}_i + \mathbf{d}_i$

Let $i \in [\![1, 7]\!]$. We have $\mathbf{I} * \mathbf{b}_i \mathbf{J}_i = \mathbf{E} * \mathbf{b}_i \mathbf{F}_i = \mathbf{c}_i + \mathbf{d}_i$. It follows that $\mathbf{E} * (\mathbf{b}_i - \mathbf{a}_i)\mathbf{F}_i = \mathbf{d}_i$, which implies that $(\mathbf{b}_i - \mathbf{a}_i) = \mathbf{E}^{-1} * \mathbf{d}_i \mathbf{F}_i^{-1}$.

$$\forall i \in [\![1, 7]\!],\ (\mathbf{b}_i - \mathbf{a}_i) = \mathbf{E}^{-1} * \mathbf{d}_i \mathbf{F}_i^{-1}$$

Since $\mathbf{E}^{-1} \in \mathsf{GL}_m(q)$ and $\mathbf{F}_i^{-1} \in \mathsf{GL}_m(q)$, we have $(\mathbf{b}_1 - \mathbf{a}_1, \mathbf{b}_2 - \mathbf{a}_2, \mathbf{b}_3 - \mathbf{a}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$ and $(\mathbf{b}_4 - \mathbf{a}_4, \mathbf{b}_5 - \mathbf{a}_5, \mathbf{b}_6 - \mathbf{a}_5) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$. Therefore, the knowledge extractor $\mathcal{K}$ obtains vectors $\mathbf{y}_i = \mathbf{b}_i - \mathbf{a}_i$, with $i \in [\![1, 7]\!]$, such that: $\sum_{i=1}^{7} \tilde{\mathbf{H}}_i \mathbf{y}_i^\top = \mathbf{c}^\top$, $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$ and $(\mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6) \in \mathfrak{S}^3_{w_r}(\mathcal{V})$.

□

References

1. Abdalla, M., Benhamouda, F., and Pointcheval, D. Disjunctions for hash proof systems: New constructions and applications. In *EUROCRYPT 2015, Part II* (Apr. 2015), E. Oswald and M. Fischlin, Eds., vol. 9057 of *LNCS*, Springer, Heidelberg, pp. 69–100.
2. Abdalla, M., Chevalier, C., and Pointcheval, D. Smooth projective hashing for conditionally extractable commitments. In *CRYPTO 2009* (Aug. 2009), S. Halevi, Ed., vol. 5677 of *LNCS*, Springer, Heidelberg, pp. 671–689.
3. Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J.-C., Gaborit, P., Persichetti, E., Robert, J.-M., Véron, P., and Zémor, G. Hamming Quasi-Cyclic (HQC).
4. Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Couvreur, A., Deneuville, J.-C., Gaborit, P., Hauteville, A., and Zémor, G. Rank Quasi-Cyclic (RQC).
5. Aguilar-Melchor, C., Blazy, O., Deneuville, J.-C., Gaborit, P., and Zémor, G. Efficient encryption from random quasi-cyclic codes. *IEEE Transactions on Information Theory 64*, 5 (2018), 3927–3943.
6. Alamélou, Q., Blazy, O., Cauchie, S., and Gaborit, P. A practical group signature scheme based on rank metric. In *International Workshop on the Arithmetic of Finite Fields* (2016), Springer, pp. 258–275.
7. Alekhnovich, M. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.* (2003), IEEE, pp. 298–307.
8. Aragon, N., Gaborit, P., Hauteville, A., and Tillich, J.-P. A new algorithm for solving the rank syndrome decoding problem. In *2018 IEEE International Symposium on Information Theory (ISIT)* (2018), IEEE, pp. 2421–2425.
9. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.-P., and Verbel, J. Algebraic attacks for solving the rank decoding and minrank problems without grobner basis. *arXiv preprint arXiv:2002.08322* (2020).
10. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.-P., and Verbel, J. Improvements of algebraic attacks for solving the rank decoding and minrank problems.
11. Bellare, M., Pointcheval, D., and Rogaway, P. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT 2000* (May 2000), B. Preneel, Ed., vol. 1807 of *LNCS*, Springer, Heidelberg, pp. 139–155.
12. Bellovin, S. M., and Merritt, M. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *1992 IEEE Symposium on Security and Privacy* (May 1992), IEEE Computer Society Press, pp. 72–84.
13. Benhamouda, F. *Diverse modules and zero-knowledge*. PhD thesis, PSL Research University - ENS, July 2016.
14. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., and Vergnaud, D. New techniques for SPHFs and efficient one-round PAKE protocols. In *CRYPTO 2013, Part I* (Aug. 2013), R. Canetti and J. A. Garay, Eds., vol. 8042 of *LNCS*, Springer, Heidelberg, pp. 449–475.
15. Benhamouda, F., Blazy, O., Ducas, L., and Quach, W. Hash proof systems over lattices revisited. In *PKC 2018, Part II* (Mar. 2018), M. Abdalla and R. Dahab, Eds., vol. 10770 of *LNCS*, Springer, Heidelberg, pp. 644–674.

16. Berlekamp, E., McEliece, R., and Van Tilborg, H. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory 24*, 3 (1978), 384–386.
17. Blazy, O., and Chevalier, C. Generic construction of UC-secure oblivious transfer. In *ACNS 15* (June 2015), T. Malkin, V. Kolesnikov, A. B. Lewko, and M. Polychronakis, Eds., vol. 9092 of *LNCS*, Springer, Heidelberg, pp. 65–86.
18. Blazy, O., Chevalier, C., and Germouty, P. Adaptive oblivious transfer and generalization. In *ASIACRYPT 2016, Part II* (Dec. 2016), J. H. Cheon and T. Takagi, Eds., vol. 10032 of *LNCS*, Springer, Heidelberg, pp. 217–247.
19. Canetti, R. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS* (Oct. 2001), IEEE Computer Society Press, pp. 136–145.
20. Chen, K. A new identification algorithm. In *Cryptography: Policy and Algorithms* (Berlin, Heidelberg, 1996), E. Dawson and J. Golić, Eds., Springer Berlin Heidelberg, pp. 244–249.
21. Cramer, R., and Shoup, V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002* (Apr. / May 2002), L. R. Knudsen, Ed., vol. 2332 of *LNCS*, Springer, Heidelberg, pp. 45–64.
22. Fiat, A., and Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86* (Aug. 1987), A. M. Odlyzko, Ed., vol. 263 of *LNCS*, Springer, Heidelberg, pp. 186–194.
23. Gabidulin, E. M. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii 21*, 1 (1985), 3–16.
24. Gaborit, P., Murat, G., Ruatta, O., and Zémor, G. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC* (2013), vol. 2013.
25. Gaborit, P., Ruatta, O., and Schrek, J. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory 62*, 2 (2015), 1006–1019.
26. Gaborit, P., Schrek, J., and Zémor, G. Full cryptanalysis of the chen identification protocol. In *International Workshop on Post-Quantum Cryptography* (2011), Springer, pp. 35–50.
27. Gaborit, P., and Zémor, G. On the hardness of the decoding and the minimum distance problems for rank codes. *IEEE Transactions on Information Theory 62*, 12 (2016), 7245–7252.
28. Garg, S., Gentry, C., Sahai, A., and Waters, B. Witness encryption and its applications. In *45th ACM STOC* (June 2013), D. Boneh, T. Roughgarden, and J. Feigenbaum, Eds., ACM Press, pp. 467–476.
29. Gennaro, R., and Lindell, Y. A framework for password-based authenticated key exchange. *ACM Transactions on Information and System Security 9*, 2 (2006), 181–234.
30. Halevi, S., and Kalai, Y. T. Smooth projective hashing and two-message oblivious transfer. *Journal of Cryptology 25*, 1 (Jan. 2012), 158–193.
31. Hofheinz, D., Hövelmanns, K., and Kiltz, E. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017, Part I* (Nov. 2017), Y. Kalai and L. Reyzin, Eds., vol. 10677 of *LNCS*, Springer, Heidelberg, pp. 341–371.
32. Kalai, Y. T. Smooth projective hashing and two-message oblivious transfer. In *EUROCRYPT 2005* (May 2005), R. Cramer, Ed., vol. 3494 of *LNCS*, Springer, Heidelberg, pp. 78–95.

33. Katz, J., and Vaikuntanathan, V. Smooth projective hashing and password-based authenticated key exchange from lattices. In *ASIACRYPT 2009* (Dec. 2009), M. Matsui, Ed., vol. 5912 of *LNCS*, Springer, Heidelberg, pp. 636–652.
34. Katz, J., and Vaikuntanathan, V. Round-optimal password-based authenticated key exchange. In *TCC 2011* (Mar. 2011), Y. Ishai, Ed., vol. 6597 of *LNCS*, Springer, Heidelberg, pp. 293–310.
35. Kawachi, A., Tanaka, K., and Xagawa, K. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *International Conference on the Theory and Application of Cryptology and Information Security* (2008), Springer, pp. 372–389.
36. Ling, S., Nguyen, K., Stehlé, D., and Wang, H. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC 2013* (Feb. / Mar. 2013), K. Kurosawa and G. Hanaoka, Eds., vol. 7778 of *LNCS*, Springer, Heidelberg, pp. 107–124.
37. Naor, M., and Yung, M. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC* (May 1990), ACM Press, pp. 427–437.
38. Ore, O. On a special class of polynomials. *Transactions of the American Mathematical Society 35*, 3 (1933), 559–584.
39. Persichetti, E. Code-based public-key encryption resistant to key leakage. In *International Conference on Availability, Reliability, and Security* (2013), Springer, pp. 44–54.
40. Pointcheval, D., and Stern, J. Security proofs for signature schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques* (1996), Springer, pp. 387–398.
41. Stern, J. A new paradigm for public key identification. *IEEE Transactions on Information Theory 42*, 6 (1996), 1757–1768.
42. Véron, P. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing 8*, 1 (1997), 57–69.
43. Zhang, J., and Yu, Y. Two-round PAKE from approximate SPH and instantiations from lattices. In *ASIACRYPT 2017, Part III* (Dec. 2017), T. Takagi and T. Peyrin, Eds., vol. 10626 of *LNCS*, Springer, Heidelberg, pp. 37–67.