Efficient Encryption from Random Quasi-Cyclic Codes

Carlos Aguilar-Melchor*, Olivier Blazy†, Jean-Christophe Deneuville†‡,
Philippe Gaborit† and Gilles Zémor§

*ISAE SUPAERO, University of Toulouse, France,
carlos.aguilar@isae-supaero.fr
†XLIM, University of Limoges, France,
{olivier.blazy,jean-christophe.deneuville,philippe.gaborit}@xlim.fr
‡INSA CVL Bourges, University of Orléans, France,
jean-christophe.deneuville@insa-cvl.fr
§IMB, University of Bordeaux, France,
gilles.zemor@math.u-bordeaux.fr

***Abstract*—We propose a framework for constructing efficient code-based encryption schemes that do not hide any structure in their public matrix. The framework is in the spirit of the schemes first proposed by Alekhnovich in 2003 and based on the difficulty of decoding random linear codes from random errors of low weight. We depart somewhat from Alekhnovich's approach and propose an encryption scheme based on the difficulty of decoding random quasi-cyclic codes. We propose two new cryptosystems instantiated within our framework: the Hamming Quasi-Cyclic cryptosystem (HQC), based on the Hamming metric, and the Rank Quasi-Cyclic cryptosystem (RQC), based on the rank metric. We give a security proof, which reduces the IND-CPA security of our systems to a decision version of the well known problem of decoding random families of quasi-cyclic codes for the Hamming and rank metrics (the respective QCSD and RQCSD problems). We also provide an analysis of the decryption failure probability of our scheme in the Hamming metric case: for the rank metric there is no decryption failure. Our schemes benefit from a very fast decryption algorithm together with small key sizes of only a few thousand bits. The cryptosystems are very efficient for low encryption rates and are very well suited to key exchange and authentication. Asymptotically, for λ the security parameter, the public key sizes are respectively in $\mathcal{O}(\lambda^2)$ for HQC and in $\mathcal{O}(\lambda^{\frac{4}{3}})$ for RQC. Practical parameter compares well to systems based on ring-LPN or the recent MDPC system.**

***Index Terms*—Code-based cryptography, public-key encryption, post-quantum cryptography, provable security**

I. Introduction

A. Background and motivation

The first code-based cryptosystem was proposed by McEliece in 1978. This system, which can be seen as a general encryption setting for coding theory, is based on a hidden trapdoor associated to a decodable family of codes, hence a strongly structured family of codes. The inherent construction of the system makes it difficult to formally reduce security to the generic difficulty of decoding random codes. Even if the original McEliece cryptosystem, based on the family of Goppa codes, is still considered secure today, many variants based on alternative families of codes (Reed-Solomon codes, Reed-Muller codes or some alternant codes [MB09, BCGO09]) were broken by recovering in polynomial time the hidden structure [FOPT10]. Moreover, high rate Goppa codes have been proved not to behave like random codes [FGUO+13], invalidating the security proof of the signature scheme [CFS01]. The fact that the hidden code structure may be uncovered (even possibly for Goppa codes, see [COT14, FPDP14, COT17]) lies like a sword of Damocles over the system, and finding a practical alternative cryptosystem based on the difficulty of decoding unstructured or random codes has always been a major issue in code-based cryptography. The recently proposed MDPC cryptosystem [MTSB13] (somewhat in the spirit of the NTRU cryptosystem [HPS98]) addresses the problem by using a hidden code structure which is significantly weaker than that of previously used algebraic codes like Goppa codes. The cryptosystem [GMRZ13] followed this trend with a similar approach. Beside this weak hidden structure, the MDPC system has very nice features and in particular relatively small key sizes, because of the cyclic structure of the public matrix. However, even if this system is a strong step forward for code-based cryptography, the hidden structure issue has not altogether disappeared, and some other attacks

regarding the decryption failure rate have been recently put in light [GJS16].

In 2003, Alekhnovich proposed an innovative approach based on the difficulty of decoding purely random codes [Ale03]. In this system the trapdoor (or secret key) is a random error vector that has been added to a random codeword of a random code. Recovering the secret key is therefore equivalent to solving the problem of decoding a random code – with no hidden structure. Alekhnovich also proved that breaking the system in any way, not necessarily by recovering the secret key, involves decoding a random linear code.

Even if the system was not totally practical, the approach in itself was a breakthrough for code-based cryptography. Its inspiration was provided in part by the Ajtai-Dwork cryptosystem [AD97] which is based on solving hard lattice problems. The Ajtai-Dwork cryptosystem also inspired the Learning With Errors (LWE) lattice-based cryptosystem by Regev [Reg03] which generated a huge amount of work in lattice-based cryptography. Attempts to emulate this approach in code-based cryptography were also made and systems based on the Learning Parity with Noise (LPN) have been proposed by exploiting the analogy with LWE [DV13, KMP14]: the LPN problem is essentially the problem of decoding random linear codes of fixed dimension and unspecified length over a binary symmetric channel. The first version of the LWE cryptosystem was not very efficient, but introducing more structure in the public key (as for NTRU) lead to the very efficient Ring-LWE cryptosystem [LPR10]. One strong feature of this last paper is that it gives a reduction from the decision version of the ring-LWE problem to a search version of the problem. Such a reduction is not known for the case of the ring-LPN problem. A ring version (ring-LPN) was nevertheless introduced in [HKL+12] for authentication and for encryption in [DP12, DMQN12].

In this paper, we propose an efficient cryptosystem based on the difficulty of decoding random quasi-cyclic codes. It is inspired by Ring-LWE encryption but is significantly adapted to the coding theory setting. Our construction benefits from some nice features: a reduction to a decision version of the general problem of decoding random quasi-cyclic codes, hence with no hidden structure, and also quite good parameters and efficiency. Since our approach is relatively general, it can also be used with other metrics such as the rank metric. Finally, another strong feature of our approach is that inherently it leads to a precise analysis of the decryption failure probability, which is also a hard point for the MDPC cryptosystem and is not done in detail for other approaches based on the LPN problem. A relative weakness of our system is its relatively low encryption rate, but this is not a major issue for classical applications of public-key encryption schemes such as authentication or key exchange.

B. Our contributions

We propose an efficient code-based cryptosystem whose security relies on decoding small weight vectors of random quasi-cyclic codes. We provide a reduction of our cryptosystem to this problem together with a detailed analysis of the decryption failure probability. Our analysis allows us to give small parameters for code-based encryption in Hamming and Rank metrics. When compared to the MDPC [MTSB13] or LRPC [GMRZ13] cryptosystems, our proposal offers a higher degree of confidence thanks to the security reduction to a well-known problem and better decryption guarantees for similar parameters (*i.e.* key and communication size), but with a lower encryption rate. Overall we propose concrete parameters for different levels of security, in both the classical and quantum settings. These parameters show the great potential of rank metric for cryptography especially for higher security settings. When compared to the ring-LPN based cryptosystem [DP12] our system has better parameters with factors 10 and 100 respectively for the size of the ciphertext and the size of the public key. We also give a general table comparing the different asymptotic sizes for different code-based cryptosystems.

C. Overview of our techniques

Our cryptosystem is based on two codes. A first code $\mathcal{C}[n,k]$, for which an efficient decoding algorithm $\mathcal{C}.\mathsf{Decode}(\cdot)$ is known. The code $\mathcal{C}$ together with its generator matrix $\mathbf{G}$ are publicly known[1]. The second code is a $[2n,n]$ random double-circulant code in systematic form, with generator matrix $\mathbf{H} = (\mathbf{I}_n \mid \mathbf{rot}(\mathbf{h}))$ (see Eq. (2) for the definition of $\mathbf{rot}(\cdot)$). The general idea of the system is that the double-circulant code is used to generate some noise, which can be handled and decoded by the code $\mathcal{C}$. The system can be seen as a noisy adaptation of the ElGamal cryptosystem.

The secret key for our cryptosystem is a *short* vector $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$ (for some metric), whose syndrome $\mathbf{s}^\top = \mathbf{H}(\mathbf{x}, \mathbf{y})^\top$ is appended to the public key $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$. To encrypt a message $\mathbf{m}$ belonging to some plaintext space, it is first encoded through the generator matrix $\mathbf{G}$, then hidden using the syndrome $\mathbf{s}$ and an additional short

[1]Therefore $\mathbf{G}$ is just a parameter of the cryptosystem and there is no need to include it in the public key.

vector $\mathbf{e}$ to prevent information leakage. In other words, encrypting a message simply consists in providing a noisy encoding of it with a particular shape. Formally, the ciphertext is $(\mathbf{u} = \mathbf{r}\mathbf{H}^\top, \mathbf{v})$, for a short random vector $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2)$ and $\mathbf{v} = \mathbf{m}\mathbf{G} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e}$ for some natural operator $\cdot$ defined in Sec. II. Using his secret key $\mathsf{sk} = (\mathbf{x}, \mathbf{y})$, the legitimate recipient can obtain a noisy version of the plaintext by computing $\mathbf{v} - \mathbf{u} \cdot \mathbf{y}$ and then recover the (noiseless) plaintext using the efficient decoding algorithm $\mathcal{C}.\mathsf{Decode}$.

For correctness, all previous constructions based on a McEliece approach rely on the fact that the error term added to the encoding of the message is less than or equal to the decoding capability of the code being used. In our construction, this assumption is no longer required and the correctness of our cryptosystem is guaranteed assuming the legitimate recipient can remove sufficiently many errors from the noisy encoding $\mathbf{v}$ of the message using sk.

The above discussion leads to the study of the probability that a decoding error occurs, which would yield a decryption failure. We study the typical weight of the error vector $\mathbf{e}'$ that one needs to decode in order to decrypt (see Sec. V for details). With the reasonable assumption, backed up by simulations, that the weight of $\mathbf{e}'$ behaves in a way that is close to a binomial distribution, we manage a precise estimation of a decoding failure and hence calibrate coding parameters accordingly.

Comparison with the McEliece framework. In the McEliece encryption framework, a hidden code is considered. This leads to two important consequences: first, the security depends on hiding the structure of the code, and second, the decryption algorithm consists of decoding the hidden code which cannot be changed. This yields different instantiations depending on the choice of the hidden code, many of which succumb to attacks and few of which resist.

In our framework there is not one unique hidden code, but two independent codes: the random double-circulant structure guarantees the security of the scheme, and the public code $\mathcal{C}$ guarantees correct decryption. It makes it possible to consider public families of codes which are difficult to hide but very efficient for decoding: also it requires finding a tradeoff for the code $\mathcal{C}$, between decoding efficiency and practical decoding complexity. But unlike the McEliece scheme, where the decryption code is fixed, it can be changed depending on the application.

The global decryption failure for our scheme depends on the articulation between the error-vector distribution induced by the double-circulant code and the decoding algorithm $\mathcal{C}.\mathsf{Decode}(\cdot)$. After having studied the error-vector distribution for the Hamming metric we associate it with a particular code adapted to low rates and bit error probability of index $1/3$. Notice that the system could possibly be used for greater encryption rate at the cost of higher parameters. This led us to choose tensor product codes, the composition of two linear codes. Tensor product codes are defined (Def. 14) in Sec. VI, and a detailed analysis of the decryption failure probability for such codes is provided there. For the rank metric case, we consider Gabidulin codes and the case when the error-vector is always decodable, with zero decryption failure probability.

Comparison with Ring-LWE. Our scheme may be considered as in the spirit of the Ring-LWE Encryption scheme but with proofs that work in the coding theory context (for both the Hamming and Rank metrics). It may be considered as a special instance of the general LWE/LPN methodology, as described, for example, in the recent paper [BS+16]. As is mentioned there, even though full LWE-based schemes may, given current knowledge, be asymptotically more efficient than their LPN counterparts, there is still significant appeal in providing a workable variation over the more simple binary field (as it was done with Ring-LWE for the LWE setting). This was previously attempted in [DP12] by relying on the Ring-LPN problem. One of the drawbacks of this last work is to be limited to rings of the form $\mathbb{F}_2[X]/(P(X))$ that are extension fields of $\mathbb{F}_2$. In contrast, we suggest using $\mathbb{F}_q[X]/(X^n - 1)$, which reduces security to a decoding problem for quasi-cyclic codes and draws upon coding theory experience of using this family of codes. Quasi-cyclic codes have indeed been studied for a long time by coding-theorists, and many of the records for minimum distance are held by quasi-cyclic codes. However, no efficient generic decoding algorithm for quasi-cyclic codes has been found, lending faith to the assumption that decoding random quasi-cyclic codes is a hard algorithmic problem. Also, this particular setting also allows us to obtain very good parameters compared to the approach of [DP12] with at least a factor 10 for the size of the keys and messages. Departing from the strict LWE/LPN paradigm also enabled us to derive a security reduction to decoding quasi-cyclic codes and arguably gives us more flexibility for the error model. Notably the rank-metric variation that we introduce has not been investigated before in the LWE/LPN setting, and looks very promising. As mentioned before, one of its features is that it enables a zero error probability of incorrect decryption.

D. Organization of the paper

The rest of the paper is organized as follows: Sec. II gives necessary background on coding theory for Hamming and Rank metrics. Sec. III describes the cryptosystem we propose and its security is discussed in Sec. IV. Sec. V and VI study the decryption failure probability and the family of tensor product codes we consider to perform the decoding for small rate codes. Finally, Sec. VII give parameters.

II. Preliminaries

A. General definitions

Notation. Throughout this paper, $\mathbb{Z}$ denotes the ring of integers and $\mathbb{F}$ denotes a finite (hence commutative) field, typically $\mathbb{F}_q$ for a prime $q \in \mathbb{Z}$ for Hamming codes or $\mathbb{F}_{q^m}$ for Rank Metric codes. Let $\mathcal{R} = \mathbb{F}[X]/(X^n - 1)$ denote the quotient ring of polynomials modulo $X^n - 1$ whose coefficients lie in $\mathbb{F}$. Elements of $\mathcal{R}$ will be interchangeably considered as row vectors or polynomials. Additionally for Hamming, we will note $\mathcal{R}_b$ elements of $\mathcal{R}$ whose binary parity is equal to b. Vectors/Polynomials (resp. matrices) will be represented by lower-case (resp. upper-case) bold letters.

For any two elements $\mathbf{x}, \mathbf{y} \in \mathcal{R}$, their product is defined as follows: $\mathbf{x} \cdot \mathbf{y} = \mathbf{z} \in \mathcal{R}$ with

$$z_k = \sum_{i+j \equiv k+1 \mod n} x_i y_j, \text{ for } k \in \{1, \dots, n-1\}.^{2} \tag{1}$$

Notice that as the product of two elements over the *commutative* ring $\mathcal{R}$, we have $\mathbf{x} \cdot \mathbf{y} = \mathbf{y} \cdot \mathbf{x}$.

For any finite set $\mathcal{S}$, $x \xleftarrow{\$} \mathcal{S}$ denotes a uniformly random element sampled from $\mathcal{S}$. For any $x \in \mathbb{R}$, let $\lfloor x \rfloor$ denotes the biggest integer smaller than or equal to x. Finally, all logarithms $\log(\cdot)$ will be base-2 unless explicitly mentioned. For a probability distribution $\mathcal{D}$, we denote by $X \sim \mathcal{D}$ the fact that X is a random variable following $\mathcal{D}$.

Definition 1 (Circulant Matrix)**.** *Let $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}^n$. The* circulant matrix *induced by $\mathbf{x}$ is defined and denoted as follows:*

$$\mathbf{rot}(\mathbf{x}) = \begin{pmatrix} x_1 & x_n & \dots & x_2 \\ x_2 & x_1 & \dots & x_3 \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n-1} & \dots & x_1 \end{pmatrix} \in \mathbb{F}^{n \times n} \tag{2}$$

As a consequence, it is easy to see that the product of any two elements $\mathbf{x}, \mathbf{y} \in \mathcal{R}$ can be expressed as a usual vector-matrix (or matrix-vector) product using the $\mathbf{rot}(\cdot)$ operator as

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x} \times \mathbf{rot}(\mathbf{y})^\top = \left(\mathbf{rot}(\mathbf{x}) \times \mathbf{y}^\top\right)^\top = \mathbf{y} \times \mathbf{rot}(\mathbf{x})^\top = \mathbf{y} \cdot \mathbf{x}. \tag{3}$$

Coding Theory. We now recall some basic definitions and properties about coding theory that will be useful to our construction. We mainly focus on generic definitions, and refer the reader to Sec. II-B for instantiations with a specific metric, and also, to [HP10] for a complete survey on code-based cryptography due to space restrictions.

Definition 2 (Linear code)**.** *A* linear code *$\mathcal{C}$ of length n and dimension k (denoted $[n, k]$) is a subspace of $\mathcal{R}$ of dimension k. Elements of $\mathcal{C}$ are referred to as codewords.*

Definition 3 (Generator Matrix)**.** *We say that $\mathbf{G} \in \mathbb{F}^{k \times n}$ is a* Generator Matrix *for the $[n, k]$ code $\mathcal{C}$ if*

$$\mathcal{C} = \left\{ \mathbf{mG}, \text{ for } \mathbf{m} \in \mathbb{F}^k \right\}. \tag{4}$$

Definition 4 (Parity-Check Matrix)**.** *Given an $[n, k]$ code $\mathcal{C}$, we say that $\mathbf{H} \in \mathbb{F}^{(n-k) \times n}$ is a* Parity-Check Matrix *for $\mathcal{C}$ if $\mathbf{H}$ is a generator matrix of the dual code $\mathcal{C}^\perp$, or more formally, if*

$$\mathcal{C}^\perp = \left\{ \mathbf{x} \in \mathbb{F}^n \text{ such that } \mathbf{H}\mathbf{x}^\top = \mathbf{0} \right\}, \tag{5}$$

where $\mathbf{H}\mathbf{x}^\top$ is the syndrome *of $\mathbf{x}$.*

Definition 5 (Minimum distance)**.** *Let $\mathcal{C}$ be an $[n, k]$ linear code over $\mathcal{R}$ and let ω be a norm on $\mathcal{R}$. The* minimum distance *of $\mathcal{C}$ is*

$$d = \min_{\mathbf{x}, \mathbf{y} \in \mathcal{C}, \mathbf{x} \neq \mathbf{y}} \omega(\mathbf{x} - \mathbf{y}). \tag{6}$$

A code with minimum distance d is capable of decoding arbitrary patterns of up to $\delta = \lfloor \frac{d-1}{2} \rfloor$ errors. Code parameters are denoted $[n, k, d]$.

Code-based cryptography usually suffers from huge keys. In order to keep our cryptosystem efficient, we will use the strategy of Gaborit [Gab05] for shortening keys. This results in Quasi-Cyclic Codes, as defined below.

Definition 6 (Quasi-Cyclic Codes [MS77, Chap. 16, §7])**.** *View a vector $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_s)$ of $\mathbb{F}_2^{sn}$ as s successive blocks (n-tuples). An $[sn, k, d]$ linear code $\mathcal{C}$ is* Quasi-Cyclic (QC) *of index s if, for any $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_s) \in \mathcal{C}$, the vector obtained after applying a simultaneous circular shift to every block $\mathbf{c}_1, \dots, \mathbf{c}_s$ is also a codeword.*

More formally, by considering each block $\mathbf{c}_i$ as a polynomial in $\mathcal{R} = \mathbb{F}[X]/(X^n - 1)$, the code $\mathcal{C}$ is QC of index s if for any $\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_s) \in \mathcal{C}$ it holds that $(X \cdot \mathbf{c}_1, \dots, X \cdot \mathbf{c}_s) \in \mathcal{C}$.

[2] The term $k + 1$ in the sum subscript compensates the fact that indices start at 1 instead of 0.

Definition 7 (Systematic Quasi-Cyclic Codes)**.** *A systematic Quasi-Cyclic $[sn, n]$ code of index s and rate $1/s$ is a quasi-cyclic code with an $(s-1)n \times sn$ parity-check matrix of the form:*

$$\mathbf{H} = \begin{bmatrix} \mathbf{I}_n & 0 & \cdots & 0 & \mathbf{A}_1 \\ 0 & \mathbf{I}_n & & & \mathbf{A}_2 \\ & & \ddots & & \vdots \\ 0 & & \cdots & \mathbf{I}_n & \mathbf{A}_{s-1} \end{bmatrix} \tag{7}$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_{s-1}$ are circulant $n \times n$ matrices.

Remark 1. *The definition of systematic quasi-cyclic codes of index s can of course be generalized to all rates ℓ/s, $\ell = 1 \ldots s-1$, but we shall only use systematic QC-codes of rates $1/2$ and $1/3$ and wish to lighten notation with the above definition. In the sequel, referring to a systematic QC-code will imply by default that it is of rate $1/s$. Note that arbitrary QC-codes are not necessarily equivalent to a systematic QC-code.*

B. Different types of metric

The previous definitions are generic and can be adapted to any type of metric.

Besides the well known Hamming metric, we also consider, in this paper, the rank metric which has interesting properties for cryptography.

We recall some definitions and properties of rank-metric Codes, and refer the reader to [Loi06] for more details. Consider the case where $\mathbb{F}$ is an extension of a finite field, *i.e.* $\mathbb{F} = \mathbb{F}_{q^m}$, and let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ be an element of some vector space of dimension n over $\mathbb{F}_{q^m}$. A basic property of field extensions is that they can be seen as vector spaces over the base field they extend. Hence, by considering $\mathbb{F}_{q^m}$ as a vector space of dimension m over $\mathbb{F}_q$, and given a basis $(\mathbf{e_1}, \ldots, \mathbf{e_m}) \in \mathbb{F}_q^m$, one can express each x_i as

$$x_i = \sum_{j=1}^{m} x_{j,i}\mathbf{e_j} \text{ (or equivalently } x_i = (x_{1,i}, \ldots, x_{m,i})\text{).} \tag{8}$$

Using such an expression, we can expand $\mathbf{x} \in \mathbb{F}_{q^m}^n$ to a matrix $\mathbf{E}(\mathbf{x})$ such that:

$$\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \ldots & x_n \end{pmatrix} \in \mathbb{F}_{q^m}^n \tag{9}$$

$$\mathbf{E}(\mathbf{x}) = \begin{pmatrix} x_{1,1} & x_{1,2} & \ldots & x_{1,n} \\ x_{2,1} & x_{2,2} & \ldots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \ldots & x_{m,n} \end{pmatrix} \in \mathbb{F}_q^{m \times n} \tag{10}$$

The definitions usually associated to Hamming metric codes such as norm (Hamming weight), support (non-zero coordinates), and isometries ($n \times n$ permutation matrices) can be adapted to the Rank metric setting based on the representation of elements as matrices in $\mathbb{F}_q^{m \times n}$.

For an element $\mathbf{x}$ of $\mathbb{F}_{q^m}^n$ we define its rank norm $\omega(\mathbf{x})$ as the rank of the matrix $\mathbf{E}(\mathbf{x})$. A rank metric code $\mathcal{C}$ of length n and dimension k over the field $\mathbb{F}_{q^m}$ is a subspace of dimension k of $\mathbb{F}_{q^m}^n$ embedded with the rank norm. In the following, $\mathcal{C}$ is a rank metric code of length n and dimension k over $\mathbb{F}_{q^m}$, where $q = p^\eta$ for some prime p and positive $\eta \geq 1$. The matrix $\mathbf{G}$ denotes a $k \times n$ generator matrix of $\mathcal{C}$. The minimum rank distance of the code $\mathcal{C}$ is the minimum rank of non-zero vectors of the code. We also considers the usual inner product which allows to define the notion of dual code.

Let $\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{F}_{q^m}^n$ be a vector of rank r. We denote by $E = \langle x_1, \ldots, x_n \rangle$ the $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ generated by the coordinates of $\mathbf{x}$ *i.e.* $E = \text{Vect}\,(x_1, \ldots, x_n)$. The vector space E is called the *support* of $\mathbf{x}$ and denoted $\text{Supp}(\mathbf{x})$. Finally, the notion of *isometry* which in Hamming metric corresponds to the action of the code on $n \times n$ permutation matrices, is replaced for the Rank metric by the action of $n \times n$ invertible matrices over the base field $\mathbb{F}_q$.

Bounds for Rank Metric Codes. The classical bounds for Hamming metric have straightforward rank metric analogues.

Singleton Bound. The classical Singleton bound for linear $[n, k]$ codes of minimum rank r over $\mathbb{F}_{q^m}$ applies naturally in the Rank metric setting. It works in the same way as for linear codes (by finding an information set) and reads $r \leq 1 + n - k$. When $n > m$ this bound can be rewritten [Loi06] as

$$r \leq 1 + \left\lfloor \frac{(n-k)m}{n} \right\rfloor. \tag{11}$$

Codes achieving this bound are called Maximum Rank Distance codes (MRD).

Deterministic Decoding. Unlike the situation for the Hamming metric, there do not exist many families of codes for the rank metric which are able to decode rank errors efficiently up to a given norm. When we are dealing with deterministic decoding, there is essentially only one known family of rank codes which can decode efficiently: the family of Gabidulin codes [Gab85]. These codes are an analogue of Reed-Solomon codes [RS60] where polynomials are replaced by q-polynomials. These codes are defined over $\mathbb{F}_{q^m}$ and for $k \leq n \leq m$, Gabidulin codes of length n and dimension k are optimal and satisfy the Singleton bound for $m = n$ with minimum distance $d = n - k + 1$. They can decode up to $\lfloor \frac{n-k}{2} \rfloor$ rank errors in a deterministic way.

Probabilistic Decoding. There also exists a simple family of codes which has been described for the subspace metric in [SKK10] and can be straightforwardly adapted to the rank metric. These codes reach asymptotically the equivalent of the Gilbert-Varshamov bound for the rank metric, however their non-zero probability of decoding failure makes them less interesting for the cases we consider in this paper.

C. Difficult problems for cryptography

In this section we describe difficult problems which can be used for cryptography. We give generic definitions for these problems which are usually instantiated with the Hamming metric but can also be instantiated with the rank metric. After defining the problems we discuss their complexity.

All problems are variants of the *decoding problem,* which consists of looking for the closest codeword to a given vector: when dealing with linear codes, it is readily seen that the decoding problem stays the same when one is given the *syndrome* of the received vector rather than the received vector. We therefore speak of *Syndrome Decoding* (SD).

Definition 8 (SD Distribution)**.** *For positive integers, n, k, and w, the* $\mathsf{SD}(n,k,w)$ Distribution *chooses* $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ *and* $\mathbf{x} \xleftarrow{\$} \mathbb{F}^n$ *such that* $\omega(\mathbf{x}) = w$*, and outputs* $(\mathbf{H}, \sigma(\mathbf{x}) = \mathbf{H}\mathbf{x}^\top)$.

Definition 9 (Search SD Problem)**.** *Let ω be a norm over $\mathcal{R}$. On input* $(\mathbf{H}, \mathbf{y}^\top) \in \mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{(n-k)}$ *from the* SD *distribution, the* Syndrome Decoding Problem $\mathsf{SD}(n,k,w)$ *asks to find* $\mathbf{x} \in \mathbb{F}^n$ *such that* $\mathbf{H}\mathbf{x}^\top = \mathbf{y}^\top$ *and* $\omega(\mathbf{x}) = w$.

Depending on the metric the above problem is instantiated with, we denote it either by SD for the Hamming metric or by Rank-SD (RSD) for the Rank metric.

For the Hamming distance the SD problem has been proven to be NP-complete in [BMvT78]. This problem can also be seen as the Learning Parity with Noise (LPN) problem with a fixed number of samples [AIK07]. The RSD problem has recently been proven difficult with a probabilistic reduction to the Hamming setting in [GZ16]. For cryptography we also need a decision version of the problem, which is given in the following Definition:

Definition 10 (Decision SD Problem)**.** *On input* $(\mathbf{H}, \mathbf{y}^\top) \xleftarrow{\$} \mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{(n-k)}$*, the* Decision SD Problem $\mathsf{DSD}(n,k,w)$ *asks to decide with non-negligible advantage whether* $(\mathbf{H}, \mathbf{y}^\top)$ *came from the* $\mathsf{SD}(n,k,w)$ *distribution or the uniform distribution over* $\mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{(n-k)}$.

As mentioned above, this problem is the problem of decoding random linear codes from random errors. The random errors are often taken as independent Bernoulli variables acting independently on vector coordinates, rather than uniformly chosen from the set of errors of a given weight, but this hardly makes any difference and one model rather than the other is a question of convenience. The DSD problem has been shown to be polynomially equivalent to its search version in [AIK07]. The rank metric version of the problem is denoted by DRSD, by applying the transformation described in [GZ16] it can be shown that the problem can be reduced to a search problem for the Hamming metric. Hence even if the reduction is not optimal, it nevertheless shows the hardness of the problem.

Finally, as for both metrics our cryptosystem will use QC-codes, we explicitly define the problem on which our cryptosystem will rely. The following Definitions describe the DSD problem in the QC configuration, and are just a combination of Def. 6 and 10. Quasi-Cyclic codes are very useful in cryptography since their compact description allows to decrease considerably the size of the keys. In particular the case $s = 2$ corresponds to double circulant codes with generator matrices of the form $(\mathbf{I}_n \mid \mathbf{A})$ for $\mathbf{A}$ a circulant matrix. Such double circulant codes have been used for almost 10 years in cryptography (cf [GG07]) and more recently in [MTSB13]. Quasi-cyclic codes of index 3 are also considered in [MTSB13].

Definition 11 (s-QCSD Distribution)**.** *For positive integers n, w and s, the* s-$\mathsf{QCSD}(n,w)$ Distribution *chooses uniformly at random a parity matrix* $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(sn-n)\times sn}$ *of a systematic QC code $\mathcal{C}$ of index s and rate $1/s$ (see Definition 7) together with a vector* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \xleftarrow{\$} \mathbb{F}^{sn}$ *such that* $\omega(\mathbf{x}_i) = w$*, $i = 1..s$, and outputs* $(\mathbf{H}, \mathbf{H}\mathbf{x}^\top)$.

Definition 12 ((Search) s-QCSD Problem)**.** *For positive integers n, w, s, a random parity check matrix* $\mathbf{H}$ *of a systematic QC code $\mathcal{C}$ of index s and* $\mathbf{y} \xleftarrow{\$} \mathbb{F}^{sn-n}$*, the* Search s-Quasi-Cyclic SD Problem s-$\mathsf{QCSD}(n,w)$ *asks to find* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \in \mathbb{F}^{sn}$ *such that* $\omega(\mathbf{x}_i) = w$*, $i = 1..s$, and* $\mathbf{y} = \mathbf{x}\mathbf{H}^\top$.

It would be somewhat more natural to choose the parity-check matrix $\mathbf{H}$ to be made up of independent uniformly random circulant submatrices, rather than with the special form required by (7). We choose this distribution so as to make the security reduction to follow less technical. It is readily seen that, for fixed s,

when choosing quasi-cyclic codes with this more general distribution, one obtains with non-negligible probability, a quasi-cyclic code that admits a parity-check matrix of the form (7). Therefore requiring quasi-cyclic codes to be systematic does not hurt the generality of the decoding problem for quasi-cyclic codes. A similar remark holds for the slightly special form of weight distribution of the vector $\mathbf{x}$.

Assumption 1. *Although there is no general complexity result for quasi-cyclic codes, decoding these codes is considered hard by the community. There exist general attacks which uses the cyclic structure of the code [Sen11, HT15] but these attacks have only a very limited impact on the practical complexity of the problem. The conclusion is that in practice, the best attacks are the same as those for non-circulant codes up to a small factor.*

The problem has a decisional form:

Definition 13 (Decision s-QCSD Problem)**.** *For positive integers n, w, s, a bit b, a random parity check matrix $\mathbf{H}$ of a systematic QC code $\mathcal{C}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{F}^{sn}$, the* Decision s-Quasi-Cyclic SD Problem s-DQCSD(n, w, b) *asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^\top)$ came from the s-QCSD(n, w) distribution or the uniform distribution over vectors of parity b in $\mathbb{F}^{(sn-n)\times sn} \times \mathbb{F}^{(sn-n)}$.*

As for the ring-LPN problem, there is no known reduction from the search version of s-QCSD problem to its decision version. The proof of [AIK07] cannot be directly adapted in the quasi-cyclic case. However the best known attacks on the decision version of the problem s-QCSD remain the direct attacks on the search version. It should be noted that when $\mathbf{H}$ is odd, b has to have the parity of sw for the problem to be hard, otherwise b has to be even. Except for the parity, the situation is similar for the rank version of these problems (respectively denoted by s-RQCSD and s-DRQCSD): the best attacks over the rank decision problem consists in attacking the rank search version of the problem.

D. Practical attacks

The practical complexity of the SD problem for the Hamming metric has been widely studied for more than 50 years. For small weights the best known attacks are exponential in the weight of the researched codeword. The best attacks can be found in [MO15].

The RSD problem is less known in cryptography but has also been studied for a long time, ever since a rank metric version of the McEliece cryptosystem was introduced in 1991 [GPT91]. We recall the main types of attack on the RSD problem below.

The complexity of practical attacks grows very quickly with the size of parameters: there is a structural reason to this. For the Hamming distance, attacks typically rely on enumerating the number of words of length n and support size (weight) t, which amounts to the Newton binomial coefficient $\binom{n}{t}$, whose value is bounded from above by 2^n. In the rank metric case, counting the number of possible supports of size r for a rank code of length n over $\mathbb{F}_{q^m}$ corresponds to counting the number of subspaces of dimension r in $\mathbb{F}_{q^m}$: this involves the *Gaussian binomial coefficient* of size roughly $q^{(m-r)m}$, whose value is also exponential in the blocklength but with a quadratic term in the exponent.

There exist two types of generic attacks on the problem:

- **Combinatorial attacks**: these attacks are usually the best ones for small values of q (typically $q = 2$) and when n and k are not too small: when q increases, the combinatorial aspect makes them less efficient. The best combinatorial attack has recently been updated to $(n-k)^3 m^3 q^{r\lfloor \frac{(k+1)m}{n} \rfloor - m}$ to take into account the value of n and m [GRS16, AGHT17].
- **Algebraic attacks**: the particular nature of the rank metric makes it a natural field for algebraic attacks using Gröbner bases, since these attacks are largely independent of the value of q and in some cases may also be largely independent of m. These attacks are usually the most efficient when q increases. For the cases considered in this paper where q is taken to be small, the complexity is greater than the cost of combinatorial attacks (see [LdVP06, FdVP08, GRS16]).

Note that the recent improvements on decoding random codes for the Hamming distance correspond to birthday paradox attacks. An open question is whether these improvements apply to rank metric codes. Given that the support of the error on codewords in rank metric is not related to the error coordinates, the birthday paradox strategy has failed for the rank metric, which for the moment seems to keep these codes protected from the aforementioned advances.

III. A NEW ENCRYPTION SCHEME

A. Encryption and security

Encryption Scheme. An encryption scheme is a tuple of four polynomial time algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$:

- $\mathsf{Setup}(1^\lambda)$, where λ is the security parameter, generates the global parameters param of the scheme;

- $\mathsf{KeyGen}(\mathsf{param})$ outputs a pair of keys, a (public) encryption key pk and a (private) decryption key sk;
- $\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$ outputs a ciphertext $\mathbf{c}$, on the message $\mathbf{m}$, under the encryption key pk;
- $\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$ outputs the plaintext $\mathbf{m}$, encrypted in the ciphertext $\mathbf{c}$ or $\perp$.

Such an encryption scheme has to satisfy both *Correctness* and *Indistinguishability under Chosen Plaintext Attack* (IND-CPA) security properties.

Correctness: For every λ, every $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$, every pair of keys $(\mathsf{pk}, \mathsf{sk})$ generated by KeyGen, every message $\mathbf{m}$, we should have $P[\mathsf{Decrypt}(\mathsf{sk}, \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})) = \mathbf{m}] = 1 - \mathsf{negl}(\lambda)$ for $\mathsf{negl}(\cdot)$ a negligible function, and where the probability is taken over the randomness $\mathbf{r}_1, \mathbf{r}_2$, and $\mathbf{e}$.

IND-CPA [GM84]: This notion formalized by the game depicted in Fig. 1, states that an adversary should not be able to efficiently guess which plaintext has been encrypted even if he knows it is one among two plaintexts of his choice.

In the following, 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:

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E}}(\lambda, t) = \max_{|\mathcal{A}| \leq t} \mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda), \tag{12}$$

where $\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda)$ is the advantage the adversary $\mathcal{A}$ has in winning game $\mathbf{Exp}^{\mathsf{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$:

$\mathbf{Exp}^{\mathsf{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m_b})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. $\texttt{RETURN}\ b'$

Figure 1. Experiment for the IND-CPA security of the scheme.

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) = \Big|\Pr[\mathbf{Exp}^{\mathsf{ind}-1}_{\mathcal{E},\mathcal{A}}(\lambda) = 1] - \Pr[\mathbf{Exp}^{\mathsf{ind}-0}_{\mathcal{E},\mathcal{A}}(\lambda) = 1]\Big|. \tag{13}$$

IND-CPA and IND-CCA2: Note that the standard security requirement for a public key cryptosystem is IND-CCA2, *indistinguishability against adaptive chosen-ciphertext attacks*, and not just IND-CPA. The main difference is that for IND-CCA2 indistinguishability must hold even if the attacker is given a *decryption oracle* first when running the FIND algorithm and also when running the GUESS algorithm (but cannot query the oracle on the challenge ciphertext $\mathbf{c}^*$). We do not present the associated formal game and definition as an existing (and inexpensive) transformation can be used [HHK17] for our scheme to pass from IND-CPA to IND-CCA2. Various generic techniques transforming a IND-CPA scheme into an IND-CCA2 scheme are known [FO99, FO13, OP01, CHJ+02] but cannot be applied to our scheme due to potential decryption errors.

In [HHK17] Hofheinz et al. present a generic transformation that takes into account decryption errors and can be applied directly to our scheme. The security reduction is tight in the random oracle model and does not require any supplemental property from our scheme (such as being γ-spread [HHK17]) as we have the IND-CPA property (instead of just a weaker property called *One-Wayness*), see App. A for details.

The idea is to de-randomize the encryption function $\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}, \theta)$ by using a hash function $\mathcal{G}$ and do a deterministic encryption of $\mathbf{m}$ by calling $c = \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}, \mathcal{G}(\mathbf{m}))$. The ciphertext is sent together with a hash $K = \mathcal{H}(\mathbf{c}, \mathbf{m})$ that ties the ciphertext to the plaintext. The receiver then decrypts $\mathbf{c}$ into $\mathbf{m}$, checks the hash value, and uses again the deterministic encryption to check that $\mathbf{c}$ is indeed *the* ciphertext associated to $\mathbf{m}$.

As the reduction is tight we do not need to change our parameters when we pass from IND-CPA to IND-CCA2. From a computational point of view, the overhead for the sender is two hash calls and for the receiver it is two hash calls and an encrypt call. From a communication point of view the overhead is the bitsize of a hash (or two if the reduction must hold in the Quantum Random Oracle Model, see [HHK17] for more details).

B. Presentation of the scheme

We begin this Section by describing a generic version of the proposed encryption scheme. This description does not depend on the particular metric used. The particular case of the Hamming metric is denoted by HQC (for Hamming Quasi-Cyclic) and RQC (for Rank Quasi-Cyclic) in the case of the rank metric. Parameter sets for binary Hamming Codes and Rank Metric Codes can be respectively found in Sec. VII-A and VII-B.

Presentation of the scheme. Recall from the introduction that the scheme uses two types of codes, a decodable $[n, k]$ code which can correct δ errors and a random double-circulant $[2n, n]$ code. Now consider a linear code $\mathcal{C}$ over $\mathbb{F}$ of dimension k and length n (generated by $\mathbf{G} \in \mathbb{F}^{k \times n}$), that can correct up to δ errors via an

efficient algorithm $\mathcal{C}$.Decode$(\cdot)$. The four polynomial-time algorithms constituting our scheme are depicted in Fig. 2.

Notice that the generator matrix $\mathbf{G}$ of the code $\mathcal{C}$ is publicly known, so the security of the scheme and the ability to decrypt do not rely on the knowledge of the error correcting code $\mathcal{C}$ being used.

Correctness. The correctness of our new encryption scheme clearly relies on the decoding capability of the code $\mathcal{C}$. Specifically, assuming $\mathcal{C}$.Decode correctly decodes $\mathbf{v} - \mathbf{u} \cdot \mathbf{y}$, we have:

$$\mathsf{Decrypt}\left(\mathsf{sk}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathbf{m}\right)\right) = \mathbf{m}. \tag{14}$$

And $\mathcal{C}$.Decode correctly decodes $\mathbf{v} - \mathbf{u} \cdot \mathbf{y}$ whenever

$$\omega\left(\mathbf{s}\mathbf{r}_2 - \mathbf{u}\mathbf{y} + \mathbf{e}\right) \leq \delta \tag{15}$$

$$\omega\left(\left(\mathbf{x} + \mathbf{h}\mathbf{y}\right)\mathbf{r}_2 - \left(\mathbf{r}_1 + \mathbf{h}\mathbf{r}_2\right)\mathbf{y} + \mathbf{e}\right) \leq \delta \tag{16}$$

$$\omega\left(\mathbf{x}\mathbf{r}_2 - \mathbf{r}_1\mathbf{y} + \mathbf{e}\right) \leq \delta \tag{17}$$

Remark 2. *In order to provide an upper bound on the decryption failure probability, an analysis of the distribution of the error vector* $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ *is provided in Sec. V.*

Remark 3. *The scheme can be directly adapted to rank metric, the only difference relies in the way the weight are defined, in that case* $(\mathbf{x}, \mathbf{y})$ *is a random codeword of rank weight* w*,* $(\mathbf{r}_1, \mathbf{r}_2)$ *is a random codeword of rank weight* $w_{\mathbf{r}}$*, and* $\mathbf{e}$ *is also chosen of rank weight* $w_{\mathbf{r}}$ *with same support than the word* $(\mathbf{r}_1, \mathbf{r}_2)$*. In the general case the error to decode for decryption,* $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ *has rank weight upper bounded by* $(w+1)w_{\mathbf{r}}$ *since* $\mathbf{x}$ *and* $\mathbf{y}$ *have the same support of dimension* w*, so as* $\mathbf{r}_1$ *and* $\mathbf{r}_2$ *of dimension* $w_{\mathbf{r}}$*.*

IV. Security of the Scheme

In this section we prove the security of our scheme, the proof is generic for any metric, and the security is reduced to the respective quasi-cyclic problems defined for Hamming and rank metric in Section 2.

Theorem 4. *The scheme presented above is* IND-CPA *under the* 2-DQCSD *and* 3-DQCSD *assumptions.*

Proof. To prove the security of the scheme, we are going to build a sequence of games transitioning from an adversary receiving an encryption of message $\mathbf{m}_0$ to an adversary receiving an encryption of a message $\mathbf{m}_1$ and show that if the adversary manages to distinguish one from the other, then we can build a simulator breaking the DQCSD assumption, for QC codes of index 2 or 3 (codes with parameters $[2n, n]$ or $[3n, n]$), and running in approximately the same time.

Game G_0: This is the real game, which we can state algorithmically as follows:

$\mathbf{Game}^0_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = \left(\mathbf{h}, \mathbf{s} = \mathsf{sk} \cdot (\mathbf{1}, \mathbf{h})^\top\right)$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m_0})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN b'

Game G_1: In this game we start by forgetting the decryption key sk, and taking $\mathbf{s}$ at random, and then proceed honestly:

$\mathbf{Game}^1_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = \left(\mathbf{h}, \mathbf{s} = \mathsf{sk} \cdot (\mathbf{1}, \mathbf{h})^\top\right)$
2b. $\mathbf{s} \xleftarrow{\$} \mathcal{R}_b$
2c. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{h}, \mathbf{s}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathbf{m_0})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN b'

The adversary has access to pk and $\mathbf{c}^*$. As he has access to pk and the Encrypt function, anything that is computed from pk and $\mathbf{c}^*$ can also be computed from just pk. Moreover, the distribution of $\mathbf{c}^*$ is independent of the game we are in, and therefore we can suppose the only input of the adversary is pk. Suppose he has an algorithm $\mathcal{D}_\lambda$, taking pk as input, that distinguishes with advantage ϵ Game G_0 and Game G_1, for some security parameter λ. Then he can also build an algorithm $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ which solves the 2-DQCSD(n, w, b_1) problem for parameters (n, w) resulting from $\mathsf{Setup}(\lambda)$, with the same advantage ϵ, when given as input a challenge $(\mathbf{H}, \mathbf{y}^\top) \in \mathbb{F}^{n \times 2n} \times \mathbb{F}^n$.

$\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{H}, \mathbf{y}^\top))$
1. Set $\mathsf{param} \leftarrow \mathsf{Setup}(\lambda)$ and get $\mathbf{G}$ from $\mathsf{KeyGen}(\mathsf{param})$
2. $\mathsf{pk} \leftarrow (\mathbf{h}, \mathbf{y})$
2. $b' \leftarrow \mathcal{D}_\lambda(\mathsf{pk})$
4. If $b' == 0$ output QCSD
5. If $b' == 1$ output UNIFORM

Note that if we define pk as $(\mathbf{h}, \mathbf{y})$ with $\mathbf{G}$ generated by $\mathsf{KeyGen}(n, k, \delta, w)$ and $(\mathbf{H}, \mathbf{y}^\top)$ from a 2-QCSD(n, w, b_1) distribution pk follows exactly the same distribution as in Game G_0. On the other

- $\mathsf{Setup}(1^\lambda)$: generates and outputs the global parameters $\mathsf{param} = (n, k, \delta, w, w_{\mathbf{r}}, w_{\mathbf{e}})$.
- $\mathsf{KeyGen}(\mathsf{param})$: samples $\mathbf{h} \xleftarrow{\$} \mathcal{R}$, the generator matrix $\mathbf{G} \in \mathbb{F}^{k\times n}$ of $\mathcal{C}$, $\mathsf{sk} = (\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{x}) = \omega(\mathbf{y}) = w$, sets $\mathsf{pk} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h}\cdot\mathbf{y})$, and returns $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m})$: generates $\mathbf{e} \xleftarrow{\$} \mathcal{R}$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{e}) = w_{\mathbf{e}}$ and $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w_{\mathbf{r}}$, sets $\mathbf{u} = \mathbf{r}_1 + \mathbf{h}\cdot\mathbf{r}_2$ and $\mathbf{v} = \mathbf{mG} + \mathbf{s}\cdot\mathbf{r}_2 + \mathbf{e}$, returns $\mathbf{c} = (\mathbf{u}, \mathbf{v})$.
- $\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$: returns $\mathcal{C}.\mathsf{Decode}(\mathbf{v} - \mathbf{u}\cdot\mathbf{y})$.

Figure 2. Description of our proposal.

hand if $(\mathbf{H}, \mathbf{y}^\top)$ comes from a uniform distribution, pk follows exactly the same distribution as in Game $\boldsymbol{G}_1$. Thus we have

$$\Pr\Big[\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{h}, \mathbf{y}^\top)) = \texttt{QCSD} \mid (\mathbf{h}, \mathbf{y}^\top) \leftarrow \text{2-QCSD}(n, w, b)\Big] = \Pr\left[\mathcal{D}_\lambda(\mathsf{pk}) = 0 \mid \mathsf{pk} \text{ from } \mathbf{Game}^0_{\mathcal{E},\mathcal{A}}(\lambda)\right] \tag{18}$$

and

$$\Pr\Big[\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{h}, \mathbf{y}^\top)) = \texttt{UNIFORM} \mid (\mathbf{h}, \mathbf{y}^\top) \leftarrow \text{2-QCSD}(n, w, b_1)\Big] = \Pr\left[\mathcal{D}_\lambda(\mathsf{pk}) = 1 \mid \mathsf{pk} \text{ from } \mathbf{Game}^0_{\mathcal{E},\mathcal{A}}(\lambda)\right]. \tag{19}$$

And similarly when $(\mathbf{h}, \mathbf{y}^\top)$ is uniform the probabilities of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ outputs match those of $\mathcal{D}_\lambda$ when pk is from $\mathbf{Game}^1_{\mathcal{E},\mathcal{A}}(\lambda)$. The advantage of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ is therefore equal to the advantage of $\mathcal{D}_\lambda$.

Game $\boldsymbol{G}_2$: Now that we no longer know the decryption key, we can start generating random ciphertexts. So instead of picking correctly weighted $\mathbf{r}_1, \mathbf{r}_2, \mathbf{e}$, the simulator now picks random vectors in the full space.

$\mathbf{Game}^2_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = \left(\mathbf{h}, \mathbf{s} = \mathsf{sk}\cdot(\mathbf{1}, \mathbf{h})^\top\right)$
2b. $\mathbf{s} \xleftarrow{\$} \mathcal{R}_{b_1}$
2c. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{h}, \mathbf{s}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4a. Generate $\mathbf{e} \xleftarrow{\$} \mathcal{R}_{b_2}$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2_{b_2}$ uniformly at random
4b. $\mathbf{u}^\top \leftarrow \mathbf{Hr}^\top$ and $\mathbf{v} \leftarrow \mathbf{m}_0\mathbf{G} + \mathbf{s}\cdot\mathbf{r}_2 + \mathbf{e}$
4c. $\mathbf{c}^* \leftarrow (\mathbf{u}, \mathbf{v})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN b'

As we have

$$(\mathbf{u}, \mathbf{v} - \mathbf{m}_0\mathbf{G})^\top = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{h}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{s}) \end{pmatrix} \cdot (\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)^\top,$$

the difference between Game $\boldsymbol{G}_1$ and Game $\boldsymbol{G}_2$ is that in the former

$$\left(\begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{h}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{s}) \end{pmatrix}, (\mathbf{u}, \mathbf{v} - \mathbf{m}_0\mathbf{G})^\top\right)$$

follows the 3-QCSD distribution (for a $2n \times 3n$ QC matrix of index 3), and in the latter it follows a uniform distribution (as $\mathbf{r}_1$ and $\mathbf{e}$ are uniformly distributed and independently chosen One-Time Pads). Note that an adversary is not able to obtain $\mathbf{c}^*$ from pk any more, as depending on which game we are $\mathbf{c}^*$ is generated differently. The input of a game distinguisher will therefore be $(\mathsf{pk}, \mathbf{c}^*)$. As it must interact with the challenger as usually we suppose it has two access modes FIND and GUESS to process first pk and later $\mathbf{c}^*$.

Suppose the adversary is able to distinguish Game $\boldsymbol{G}_1$ and Game $\boldsymbol{G}_2$, with a distinguisher $\mathcal{D}_\lambda$, which takes as input $(\mathsf{pk}, \mathbf{c}^*)$ and outputs a guess $b' \in \{1, 2\}$ of the game we are in.

Again, we can build a distinguisher $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ that will break the 3-DQCSD(n, w) assumption for parameters (n, w) from $\mathsf{Setup}(1^\lambda)$ with the same advantage as the game distinguisher, when given an input $(\mathbf{H}, \mathbf{y}^\top) \in \mathbb{F}^{2n\times 3n} \times \mathbb{F}^{2n}$. In the 3-DQCSD$(n, w, b_2)$ problem, matrix $\mathbf{H}$ is assumed to be of the form

$$\begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{a}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{b}) \end{pmatrix}.$$

In order to use explicitly $\mathbf{a}$ and $\mathbf{b}$ we note the matrix $\mathbf{H}_{\mathbf{a},\mathbf{b}}$ instead of just $\mathbf{H}$. We will also note $\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2})$.

$\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}((\mathbf{H}_{\mathbf{a},\mathbf{b}}, (\mathbf{y_1}, \mathbf{y_2})^\top))$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$
2b. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{G}, (\mathbf{I_n}\ \mathbf{rot}(\mathbf{a})), \mathbf{b}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{D}_\lambda(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{u} \leftarrow \mathbf{y_1}$, $\mathbf{v} \leftarrow \mathbf{m}_0\mathbf{G} + \mathbf{y_2}$ and $\mathbf{c}^* \leftarrow (\mathbf{u}, \mathbf{v})$
5. $b' \leftarrow \mathcal{D}_\lambda(\texttt{GUESS} : \mathbf{c}^*)$
4. If $b' == 1$ output QCSD
5. If $b' == 2$ output UNIFORM

The distribution of pk is unchanged with respect to the games as the first matrix is from KeyGen, the second matrix follows the same distribution as in KeyGen, and the vectors $\mathbf{b}$ and $\mathbf{s}$ are both uniformly chosen. If $(\mathbf{H_{a,b}}, (\mathbf{y_1}, \mathbf{y_2})^\top)$ follows the 3-QCSD(n, w, b_2) distribution, then

$$(\mathbf{y_1}, \mathbf{y_2})^\top = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot(a)} \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot(b)} \end{pmatrix} \cdot (\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})^\top$$

with $\omega(\mathbf{x}_1) = \omega(\mathbf{x}_2) = \omega(\mathbf{x}_3) = w$. Thus, $\mathbf{c}^*$ follows the same distribution as in Game $\boldsymbol{G}_1$. If $(\mathbf{H_{a,b}}, (\mathbf{y_1}, \mathbf{y_2})^\top)$ follows an uniform distribution, then $\mathbf{c}^*$ follows the same distribution as in Game $\boldsymbol{G}_2$. We obtain therefore the same equalities for the output probabilities of $\mathcal{D}'_{\mathcal{E},\mathcal{D}_\lambda}$ and $\mathcal{D}_\lambda$ as with the previous games and therefore the advantages of both distinguishers are equal.

Game $\boldsymbol{G}_3$: We now encrypt the other plaintext. We chose $\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}'$ uniformly and set $\mathbf{u}^\top = \mathbf{h}\mathbf{r}'^\top$ and $\mathbf{v} = \mathbf{m}_1\mathbf{G} + \mathbf{s} \cdot \mathbf{r}'_2 + \mathbf{e}'$. This is the last game we describe explicitly, since, even if it is a mirror of Game $\boldsymbol{G}_2$, it involves a new proof.

$\mathbf{Game}^3_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2a. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$ with $\mathsf{pk} = (\mathbf{h}, \mathbf{s})$
2b. $\mathbf{s} \xleftarrow{\$} \mathcal{R}_{b_1}$
2c. $(\mathsf{pk}, \mathsf{sk}) \leftarrow ((\mathbf{h}, \mathbf{s}), \mathbf{0})$
3. $(\mathbf{m}_0, \mathbf{m}_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4a. Generate $\mathbf{e}' \xleftarrow{\$} \mathcal{R}_{b_2}$, $\mathbf{r}' = (\mathbf{r}'_1, \mathbf{r}'_2) \xleftarrow{\$} \mathcal{R}^2_{b_2}$ uniformly at random
4b. $\mathbf{u}^\top \leftarrow \mathbf{Q}\mathbf{r}'^\top$ and $\mathbf{v} \leftarrow \mathbf{m}_1\mathbf{G} + \mathbf{s} \cdot \mathbf{r}'_2 + \mathbf{e}'$
4c. $\mathbf{c}^* \leftarrow (\mathbf{u}, \mathbf{v})$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. $\texttt{RETURN}\ b'$

The outputs from Game $\boldsymbol{G}_2$ and Game $\boldsymbol{G}_3$ follow the exact same distribution, and therefore the two games are indistinguishable from an information-theoretic point of view. Indeed, for each tuple $(\mathbf{r}, \mathbf{e})$ of Game $\boldsymbol{G}_2$, resulting in a given $(\mathbf{u}, \mathbf{v})$, there is a one to one mapping to a couple $(\mathbf{r}', \mathbf{e}')$ resulting in Game $\boldsymbol{G}_3$ in the *same* $(\mathbf{u}, \mathbf{v})$, namely $\mathbf{r}' = \mathbf{r}$ and $\mathbf{e}' - \mathbf{m}_0\mathbf{G} + \mathbf{m}_1\mathbf{G}$. This implies that choosing uniformly $(\mathbf{r}, \mathbf{e})$ in Game $\boldsymbol{G}_2$ and choosing uniformly $(\mathbf{r}', \mathbf{e}')$ in Game $\boldsymbol{G}_3$ leads to the same output distribution for $(\mathbf{u}, \mathbf{v})$.

Game $\boldsymbol{G}_4$: In this game, we now pick $\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{e}'$ with the correct weight.

Game $\boldsymbol{G}_5$: We now conclude by switching the public key to an honestly generated one.

We do not explicit these last two games as Game $\boldsymbol{G}_3$ and Game $\boldsymbol{G}_4$ are the equivalents of Game $\boldsymbol{G}_2$ and Game $\boldsymbol{G}_1$ except that $\mathbf{m}_1$ is used instead of $\mathbf{m}_0$. A distinguisher between these two games breaks therefore the 3-DQCSD assumption too. Similarly Game $\boldsymbol{G}_4$ and Game $\boldsymbol{G}_5$ are the equivalents of Game $\boldsymbol{G}_1$ and Game $\boldsymbol{G}_0$ and a distinguisher between these two games breaks the 2-DQCSD assumption.

We managed to build a sequence of games allowing a simulator to transform a ciphertext of a message $\mathbf{m}_0$ to a ciphertext of a message $\mathbf{m}_1$. Hence, the advantage of an adversary against the IND-CPA experiment is bounded as:

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2\left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right). \tag{20}$$

□

V. Analysis of the error vector distribution for Hamming distance

The aim of this Section is to determine the probability that the condition in Eq. (17) holds. In order to do so, we study the error distribution of the error vector $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$.

The vectors $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r_2}, \mathbf{e}$ have been taken to be uniformly and independently chosen among vectors of weight w, $w_\mathbf{r}$ or $w_\mathbf{e}$. A very close probabilistic model is when all these independent vectors are chosen to follow the distribution of random vectors whose coordinates are independent Bernoulli variables of parameter $p = w/n$ (or $p_\mathbf{r} = w_\mathbf{r}/n$ and $p_\mathbf{e} = w_\mathbf{e}/n$ respectively). To simplify analysis we shall assume this model rather than the constant weight uniform model. Both models are very close, and our cryptographic protocols work just as well in both settings.

We first evaluate the distributions of the products $\mathbf{x} \cdot \mathbf{r}_2$ and $\mathbf{r}_1 \cdot \mathbf{y}$.

Proposition 5. *Let $\mathbf{x} = (X_1, \ldots, X_n)$ (resp. $\mathbf{r} = (R_1, \ldots, R_n)$) be a random vector where the X_i (resp. R_i) are independent Bernoulli variables of parameter p (resp. $p_\mathbf{r}$), $P(X_i = 1) = p$ and $P(R_i = 1) = p_\mathbf{r}$. Assuming $\mathbf{x}$ and $\mathbf{r}$ are independent, and denoting $\mathbf{z} = \mathbf{x} \cdot \mathbf{r} = (Z_1, \ldots, Z_n)$ as defined in Eq. (1), we have:*

$$\begin{cases} \Pr[Z_k = 1] = \frac{1}{2} - \frac{1}{2}(1 - 2pp_\mathbf{r})^n, \\ \Pr[Z_k = 0] = \frac{1}{2} + \frac{1}{2}(1 - 2pp_\mathbf{r})^n. \end{cases} \tag{21}$$

Proof. We have

$$Z_k = \sum_{i+j=k+1 \bmod n} X_i R_j \mod 2. \tag{22}$$

Every term $X_i R_j$ is the product of two independent Bernoulli variables of parameter respectively p and $p_{\mathbf{r}}$, and is therefore a Bernoulli variable of parameter $p \times p_{\mathbf{r}}$. The variable Z_k is the sum modulo 2 of n such products, which are all independent since every variable X_i is involved exactly once in (22), for $1 \leq i \leq n$, and similarly every variable R_j is involved once in (22). Therefore Z_k is the sum modulo 2 of n independent Bernoulli variables of parameter $p \times p_{\mathbf{r}}$, and we have

$$\Pr[Z_k = 1] = \sum_{0 \leq i \leq n, i \text{ odd}} \binom{n}{i} (pp_{\mathbf{r}})^i (1 - pp_{\mathbf{r}})^{n-i}$$

which, using the equations:

$$\sum_{\substack{0 \leq i \leq n, \\ i \text{ odd}}} \binom{n}{i} a^i b^{n-i} = \frac{(a+b)^n - (a-b)^n}{2}, \text{ and} \quad (23)$$

$$\sum_{\substack{0 \leq i \leq n, \\ i \text{ even}}} \binom{n}{i} a^i b^{n-i} = \frac{(a+b)^n + (a-b)^n}{2} \quad (24)$$

with $a = pp_{\mathbf{r}}$ and $b = 1 - pp_{\mathbf{r}}$, simplifies into the claimed result. □

Let us denote by $\tilde{p} = \tilde{p}(n, w) = \Pr[Z_k = 1]$ from Eq. ((21)). Let $\mathbf{x}, \mathbf{y}$ (resp. $\mathbf{r}_1, \mathbf{r}_2$) be independent random vectors whose coordinates are independently Bernoulli distributed with parameter p (resp. $p_{\mathbf{r}}$). Then the k-th coordinates of $\mathbf{x} \cdot \mathbf{r}_2$ and of $\mathbf{r}_1 \cdot \mathbf{y}$ are independent and Bernoulli distributed with parameter $\tilde{p}$. Therefore their modulo 2 sum $\mathbf{t} = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y}$ is Bernoulli distributed with

$$\begin{cases} \Pr[t_k = 1] = 2\tilde{p}(1 - \tilde{p}), \\ \Pr[t_k = 0] = (1 - \tilde{p})^2 + \tilde{p}^2. \end{cases} \quad (25)$$

Finally, by adding the term $\mathbf{e}$ to $\mathbf{t}$, we obtain the distribution of the coordinates of the error vector $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$. Since the coordinates of $\mathbf{e}$ are Bernoulli of parameter $p_{\mathbf{e}}$ and those of $\mathbf{t}$ are Bernoulli distributed as (25) and independent from $\mathbf{e}$, we obtain :

Proposition 6. *Let* $\mathbf{x}, \mathbf{y} \sim \mathcal{B}\left(n, \frac{w}{n}\right)$, $\mathbf{r}_1, \mathbf{r}_2 \sim \mathcal{B}\left(n, \frac{w_{\mathbf{r}}}{n}\right)$ *and* $\mathbf{e} \sim \mathcal{B}\left(n, \frac{w_{\mathbf{e}}}{n}\right)$, *and let* $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$. *Then*

$$\begin{cases} \Pr[e'_k = 1] = 2\tilde{p}(1 - \tilde{p})(1 - \frac{w_{\mathbf{e}}}{n}) + \left((1 - \tilde{p})^2 + \tilde{p}^2\right) \frac{w_{\mathbf{e}}}{n}, \\ \Pr[e'_k = 0] = \left((1 - \tilde{p})^2 + \tilde{p}^2\right)(1 - \frac{w_{\mathbf{e}}}{n}) + 2\tilde{p}(1 - \tilde{p}) \frac{w_{\mathbf{e}}}{n}. \end{cases} \quad (26)$$

Proposition 6 gives us the probability that a coordinate of the error vector $\mathbf{e}'$ is 1. In our simulations to follow, which occur in the regime $p = \alpha\sqrt{n}$ with constant α, we make the simplifying assumption that the coordinates of $\mathbf{e}'$ are independent, meaning that the weight of $\mathbf{e}'$ follows a binomial distribution of parameter $p^\star$, where $p^\star$ is defined as in Eq. (26): $p^\star = 2\tilde{p}(1 - \tilde{p})(1 - \frac{w_{\mathbf{e}}}{n}) + \left((1 - \tilde{p})^2 + \tilde{p}^2\right) \frac{w_{\mathbf{e}}}{n}$. This approximation will give us, for $0 \leq d \leq \min(2 \times w \times w_{\mathbf{r}} + w_{\mathbf{e}}, n)$,

$$\Pr[\omega(\mathbf{e}') = d] = \binom{n}{d} (p^\star)^d (1 - p^\star)^{(n-d)}. \quad (27)$$

In practice, the results obtained by simulation on the decryption failure are very coherent with this assumption.

VI. Decoding codes with low rates and good decoding properties

The previous Section allowed us to determine the distribution of the error vector $\mathbf{e}$ in the configuration where a simple linear code is used. Now the decryption part corresponds to decoding the error described in the previous section. Any decodable code can be used at this point, depending on the considered application: clearly small dimension codes will allow better decoding, but at the cost of a lower encryption rate. The particular case that we consider corresponds typically to the case of key exchange or authentication, where only a small amount of data needs to be encrypted (typically 80, 128 or 256 bits, a symmetric secret key size). We therefore need codes with low rates which are able to correct many errors. Again, a tradeoff is necessary between efficiently decodable codes but with a high decoding cost and less efficiently decodable codes but with a smaller decoding cost.

An example of such a family of codes with good decoding properties, meaning a simple decoding algorithm which can be analyzed, is given by Tensor Product Codes, which are used for biometry [BCC+07], where the same type of issue appears. More specifically, we will consider a special simple case of Tensor Product Codes (BCH codes and repetition codes), for which a precise analysis of the decryption failure can be obtained in the Hamming distance case.

A. Tensor product codes

Definition 14 (Tensor Product Code)**.** *Let* $\mathcal{C}_1$ *(resp.* $\mathcal{C}_2$*) be a* $[n_1, k_1, d_1]$ *(resp.* $[n_2, k_2, d_2]$*) linear code over* $\mathbb{F}$. *The* Tensor Product Code *of* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *denoted* $\mathcal{C}_1 \otimes \mathcal{C}_2$ *is defined as the set of all* $n_2 \times n_1$ *matrices whose rows are codewords of* $\mathcal{C}_1$ *and whose columns are codewords of* $\mathcal{C}_2$.

More formally, if $\mathcal{C}_1$ *(resp.* $\mathcal{C}_2$*) is generated by* $\mathbf{G}_1$ *(resp.* $\mathbf{G}_2$*), then*

$$\mathcal{C}_1 \otimes \mathcal{C}_2 = \left\{ \mathbf{G}_2^\top \mathbf{X} \mathbf{G}_1 \text{ for } \mathbf{X} \in \mathbb{F}^{k_2 \times k_1} \right\} \quad (28)$$

Remark 7. *Using the notation of the above Definition, the tensor product of two linear codes is a* $[n_1 n_2, k_1 k_2, d_1 d_2]$ *linear code.*

B. Specifying the tensor product code

Even if tensor product codes seem well-suited for our purpose, an analysis similar to the one in Sec. V becomes much more complicated. Therefore, in order to provide strong guarantees on the decryption failure probability for our cryptosystem, we chose to restrict ourselves to a tensor product code $\mathcal{C} = \mathcal{C}_1 \otimes \mathcal{C}_2$, where $\mathcal{C}_1$ is a BCH(n_1, k_1, δ_1) code of length n_1, dimension k_1, and correcting capability δ_1 (*i.e.* it can correct up to δ_1 errors), and $\mathcal{C}_2$ is the repetition code of length n_2 and dimension 1, denoted $\mathbb{1}_{n_2}$. (Notice that $\mathbb{1}_{n_2}$ can decode up to $\delta_2 = \lfloor \frac{n_2-1}{2} \rfloor$.) Subsequently, the analysis becomes possible and remains accurate but the negative counterpart is that there probably are some other tensor product codes achieving better efficiency (or smaller key sizes).

In the Hamming metric version of the cryptosystem we propose, a message $\mathbf{m} \in \mathbb{F}^{k_1}$ is first encoded into $\mathbf{m}_1 \in \mathbb{F}^{n_1}$ with a BCH$(n_1, k_1 = k, \delta_1)$ code, then each coordinate $\mathbf{m}_{1,i}$ of $\mathbf{m}_1$ is re-encoded into $\tilde{\mathbf{m}}_{1,i} \in \mathbb{F}^{n_2}$ with a repetition code $\mathbb{1}_{n_2}$. We denote $n = n_1 n_2$ the length of the tensor product code[3] (its dimension is $k = k_1 \times 1$), and by $\tilde{\mathbf{m}}$ the resulting encoded vector, *i.e.* $\tilde{\mathbf{m}} = (\tilde{\mathbf{m}}_{1,1}, \ldots, \tilde{\mathbf{m}}_{1,n_1}) \in \mathbb{F}^{n_1 n_2}$.

The efficient algorithm used for the repetition code is the majority decoding, *i.e.* more formally:

$$\mathbb{1}_{n_2}.\mathsf{Decode}(\tilde{\mathbf{m}}_{1,j}) = \begin{cases} 1 \text{ if } \sum_{i=0}^{n_2-1} \tilde{\mathbf{m}}_{1,j,i} \geq \lceil \frac{n_2+1}{2} \rceil, \\ 0 \text{ otherwise.} \end{cases} \tag{29}$$

Decryption Failure Probability. With a tensor product code $\mathcal{C} = \text{BCH}(n_1, k_1, \delta) \otimes \mathbb{1}_{n_2}$ as defined above, a decryption failure occurs whenever the decoding algorithm of the BCH code does not succeed in correcting errors that would have arisen after wrong decodings by the repetition code. Therefore, the analysis of the decryption failure probability is again split into three steps: evaluating the probability that the repetition code does not decode correctly, the conditional probability of a wrong decoding for the BCH code given an error weight and finally, the decryption failure probability using the law of total probability.

Step 1. We now focus on the probability that an error occurs while decoding the repetition code. As shown in Sec. V, the probability for a coordinate of $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ to be 1 is $p^\star$ (see Eq. (26)). As mentioned above, $\mathbb{1}_{n_2}$ can decode up to $\delta_2 = \lfloor \frac{n_2-1}{2} \rfloor$ errors. Therefore, assuming that the error vector $\mathbf{e}'$ has weight γ (which occurs with the probability given in Eq. (27)), the probability of getting a decoding error on a single block of the repetition code $\mathbb{1}_{n_2}$ is hence given by $\bar{p}_\gamma = \bar{p}_\gamma(n_1, n_2)$ with:

$$\bar{p}_\gamma = \sum_{i=\lfloor \frac{n_2-1}{2} \rfloor + 1}^{n_2} \binom{n_2}{i} \left(\frac{\gamma}{n_1 n_2}\right)^i \left(1 - \frac{\gamma}{n_1 n_2}\right)^{n_2 - i}. \tag{30}$$

Step 2. We now focus on the BCH(n_1, k_1, δ_1) code, and recall that it can correct up to δ_1 errors. Now the probability $\mathcal{P}$ that the BCH(n_1, k_1, δ_1) code fails to decode correctly the encoded message $\mathbf{m}_1$ back to $\mathbf{m}$ is given by the probability that an error occurred on at least $\delta_1 + 1$ blocks of the repetition code. Therefore, we have

$$\mathcal{P} = \mathcal{P}(\delta_1, n_1, n_2, \gamma) = \sum_{i=\delta_1+1}^{n_1} \binom{n_1}{i} (\bar{p}_\gamma)^i (1 - \bar{p}_\gamma)^{n_1 - i}. \tag{31}$$

Step 3. Finally, using the law of total probability, we have that the decryption failure probability is given by the sum *over all the possible weights* of the probability that the error has this specific weight times the probability of a decoding error for this weight. This is captured in the following theorem, whose proof is a straightforward consequence of the formulae of Sec. V and VI-A.

Theorem 8. *Let* $\mathcal{C} = \text{BCH}(n_1, k_1, \delta) \otimes \mathbb{1}_{n_2}$, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}$, *and* $\mathbf{m} \xleftarrow{\$} \mathbb{F}_2^{k_1}$, *then with the notations above, the decryption failure probability is*

$$p_{\text{fail}} = \Pr\left[\mathsf{Decrypt}\left(\mathsf{sk}, \mathsf{Encrypt}\left(\mathsf{pk}, \mathbf{m}\right)\right) \neq \mathbf{m}\right] \tag{32}$$

$$= \sum_{\gamma=0}^{\min(2w w_{\mathbf{r}} + w_{\mathbf{e}}, n_1 n_2)} \Pr[\omega(\mathbf{e}') = \gamma] \times \mathcal{P} \tag{33}$$

Eq. (33) gives a theoretical approximation of the decryption failure rate. The parameters presented in Tab. I were obtained using this formula. Experimental evidences supporting the validity of the assumptions made to obtained this formula are provided in Fig. 3.

VII. Parameters

A. HQC instantiation for Hamming metric

In this Section, we describe our new cryptosystem in the Hamming metric setting. As mentioned in the previous Section, we use a tensor product code (Def. 14) $\mathcal{C} = \text{BCH}(n_1, k_1, \delta) \otimes \mathbb{1}_{n_2}$. A message $\mathbf{m} \in \mathbb{F}^{k_1}$ is encoded into $\mathbf{m}_1 \in \mathbb{F}^{n_1}$ with the BCH code, then each

[3] In practice, the length is the smallest primitive prime greater than n to avoid algebraic attacks.

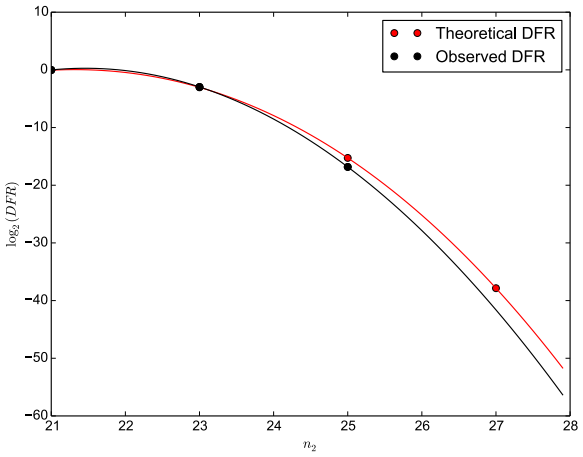

Figure 3. Logarithm of theoretical and observed decryption failure rates (DFR). The red curve corresponding to theoretical DFR was obtained using Eq. (33) while the black curve corresponding to experimental DFR was obtained by running 10^5 encryption/decryption over 10^3 codes with $n_1 = 766$, $k_1 = 256$, $\delta_1 = 57$, $w = 67$, $w_{\mathbf{r}} = 77$. The parameters have been selected to make the theoretical DFR sufficiently high to compare it to experiments. Finally, the curves have been interpolated to the second order on the logarithm of the probability.

coordinate $\mathbf{m}_{1,i}$ of $\mathbf{m}_1$ is encoded into $\tilde{\mathbf{m}}_{1,i} \in \mathbb{F}^{n_2}$ with $\mathbb{1}_{n_2}$. To match the description of our cryptosystem in Sec. III-B, we have $\mathbf{mG} = \tilde{\mathbf{m}} = (\tilde{\mathbf{m}}_{1,1}, \ldots, \tilde{\mathbf{m}}_{1,n_1}) \in \mathbb{F}^{n_1 n_2}$. To obtain the ciphertext, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ and $\mathbf{e} \xleftarrow{\$} \mathcal{R}$ are generated and the encryption of $\mathbf{m}$ is $\mathbf{c} = (\mathbf{u} = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2, \mathbf{v} = \mathbf{mG} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e})$.

Parameters and tightness of the reduction. We propose different sets of parameters: basic, advanced, and paranoiac which respectively provide 128, 192, and 256 bits of classical (*i.e.* pre-quantum) security. The quantum-safe security is obtained by dividing the security bits by two (taking the square root of the complexity) [Ber10]. For each security level, we provide different decryption failure rates to better adapt to the adversary computing power. Notice that even if the adversary has access to a quantum computer, this *does not* change the decryption failure rate.[4] Best known attacks include the works from [CC98, BLP08, FS09, MMT11, BJMM12, MO15] and for quantum attacks, the work of [Ber10]. In the setting $w = \mathcal{O}(\sqrt{n})$, best known attacks have a complexity in $2^{-t\ln(1-R)(1+o(1))}$ where $t = \mathcal{O}(w)$ and R is the rate of the code [CS16]. In our configuration, we have $t = 2w$ and $R = 1/2$ for the reduction to the 2-DQCSD problem, and $t = 3w_{\mathbf{r}}$ and $R = 1/3$ for the 3-DQCSD problem. By taking into account the DOOM attack [Sen11], and also the fact that we consider balanced vectors $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)$ for the attack (which costs only a very small factor, since random words have a good probability to be balanced on each block), we need to divide this complexity by approximately $\sqrt{n}$ (up to polylog factor). The term $o(1)$ is respectively $\log\left(\binom{n}{w}^2 / \binom{2n}{2w}\right)$ and $\log\left(\binom{n}{w_{\mathbf{r}}}^3 / \binom{3n}{3w_{\mathbf{r}}}\right)$ for the 2-DQCSD and 3-DQCSD problems. Overall our security reduction is tight corresponding to generic instances of the classical 2-DQCSD and 3-DQCSD problems according to the best attacks of [CS16].

[4]We do not consider the very strong adversarial model where the adversary is given access to a quantum decryption oracle.

Specific structural attacks. Quasi-cyclic codes have a special structure which may potentially open the door to specific structural attacks. A first generic attack is the DOOM attack [Sen11] which because of cyclicity implies a gain of $\mathcal{O}(\sqrt{n})$ (when the gain is in $\mathcal{O}(n)$ for MDPC codes, since the code is generated by a small weight vector basis). It is also possible to consider attacks on the form of the polynomial generating the cyclic structure. Such attacks have been studied in [GJL15, LJK+16, Sen11], and are especially efficient when the polynomial $x^n - 1$ has many low degree factors. These attacks become inefficient as soon as $x^n - 1$ has only two irreducible factors of the form $(x - 1)$ and $x^{n-1} + x^{n-2} + \ldots + x + 1$, which is the case when n is prime and q generates the multiplicative group $(\mathbb{Z}/n\mathbb{Z})^*$. Such numbers are known up to very large values. We consider such n for our parameters.

In Tab. I, n_1 denotes the length of the BCH code, n_2 the length of the repetition code $\mathbb{1}$ so that the length of the tensor product code $\mathcal{C}$ is $n \approx n_1 n_2$ (actually the smallest primitive prime greater than $n_1 n_2$). k_1 is the dimension of the BCH code and hence also the dimension of $\mathcal{C}$. δ_1 is the decoding capability of the BCH code, *i.e.* the maximum number of errors that the BCH can decode. (If we denote d_1 the minimum distance of the BCH code, we have that $\delta_1 = \left\lfloor \frac{d_1 - 1}{2} \right\rfloor$.) w is the weight of the n-dimensional vectors $\mathbf{x}$, $\mathbf{y}$, $w_{\mathbf{r}}$ the weight of $\mathbf{r}_1$, and $\mathbf{r}_2$ and similarly $w_{\mathbf{e}} = \omega(\mathbf{e})$ for our cryptosystem.

Computational costs of the system. For encryption the main cost is a product of a cyclic matrix of size n with a vector of weight $\mathcal{O}(\sqrt{n})$. Using the Fourier transform the asymptotical cost is in $\mathcal{O}(n \log(n))$ but for our range of parameters, taking into account the weight $\mathcal{O}(\sqrt{n})$ allows to obtain a cost in $\mathcal{O}(n^{\frac{3}{2}})$ which is better in practice that what is obtained with Fourier transform. For decryption, there is always the cost of a matrix times a small vector in $\mathcal{O}(n^{\frac{3}{2}})$, plus the cost of decoding. For our proposition the decoding consists in a repetition code

Instance	Cryptosystem Parameters								
	n_1	n_2	$n \approx n_1 n_2$	k_1	δ_1	w	$w_{\mathbf{r}} = w_{\mathbf{e}}$	security	p_{fail}
Basic-I	766	29	22,229	256	57	67	77	128	$< 2^{-64}$
Basic-II	766	31	23,747	256	57	67	77	128	$< 2^{-96}$
Basic-III	796	31	24,677	256	57	67	77	128	$< 2^{-128}$
Advanced-I	796	51	40,597	256	60	101	117	192	$< 2^{-64}$
Advanced-II	766	57	43,669	256	57	101	117	192	$< 2^{-128}$
Advanced-III	766	61	46,747	256	57	101	117	192	$< 2^{-192}$
Paranoiac-I	766	77	59,011	256	57	133	153	256	$< 2^{-64}$
Paranoiac-II	766	83	63,587	256	57	133	153	256	$< 2^{-128}$
Paranoiac-III	796	85	67,699	256	60	133	153	256	$< 2^{-192}$
Paranoiac-IV	796	89	70,853	256	60	133	153	256	$< 2^{-256}$

Table I
PARAMETER SETS FOR OUR CRYPTOSYSTEM IN HAMMING METRIC. THE TENSOR PRODUCT CODE USED IS $\mathcal{C} = \text{BCH}(n_1, k_1, \delta_1) \otimes \mathbb{1}_{n_2}$. THE CONSIDERED BCH CODES ARE INITIALLY OF LENGTH 1023, THEN SHORTENED TO SUPPORT 256 BITS DIMENSION. THE PUBLIC KEY SIZE, CONSISTING OF $(\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y})$, HAS SIZE $2n$ (IN BITS) (ALTHOUGH CONSIDERING A SEED FOR $\mathbf{h}$ THE SIZE CAN BE REDUCED TO n PLUS THE SIZE OF THE SEED), AND THE SECRET KEY (CONSISTING OF $\mathbf{x}$ AND $\mathbf{y}$ BOTH OF WEIGHT w) HAS SIZE $2w\lceil \log_2(n) \rceil$ (BITS) - WHICH AGAIN CAN BE REDUCED TO THE SIZE OF A SEED. FINALLY, THE SIZE OF THE ENCRYPTED MESSAGE IS $2n$.

of length n_2 and the decoding of BCH code of length n_1 ($256 \leq n_1 \leq 768$), the cost of the repetition code decoding is hence linear, when the cost of the BCH is quadratic in the length n_1 of the BCH code. Overall the main cost remains the computation of the matrix-vector product in $\mathcal{O}(n^{\frac{3}{2}})$. In practice the different times for encryption and decryption for a 256 bits security are of order 1 ms on a 4GHz computer.

Notice that it would be possible to consider other types of decodable codes in order to increase the encryption rate to $1/4$ (say), but at the cost of an increase of the length of the code, for instance using LDPC (3,6) codes would increase the rate, but multiply the length by a factor of roughly three.

B. RQC instantiation for rank metric

Error distribution and decoding algorithm: no decryption failure. The case of the rank metric is much simpler than for the Hamming metric. Indeed in that case the decryption algorithm of our cryptosystem asks to decode an error $\mathbf{e}' = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \mathbf{e}$ where the words $\mathbf{x}$ and $\mathbf{y}$ (resp. $\mathbf{r}_1$ and $\mathbf{r}_2$)) have rank weight w (resp. $w_{\mathbf{r}}$). Unlike the Hamming metric weight, the rank weight of the vector $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y}$ is almost always $ww_{\mathbf{r}}$ and is in any case bounded from above by $ww_{\mathbf{r}}$. In particular, with a strong probability, the rank weight of $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y}$ is the same as the rank weight of $\mathbf{x} \cdot \mathbf{r}_2$ since $\mathbf{x}$ and $\mathbf{y}$ share the same rank support, as do $\mathbf{r}_1$ and $\mathbf{r}_2$. We consider the additional error $\mathbf{e}$ of rank $w_{\mathbf{e}} = w_{\mathbf{r}}$ with same error support as $\mathbf{r}_1$ and $\mathbf{r}_2$. So that overall the error e' to decode for decryption has a rank weight upper bounded by $(w+1)w_{\mathbf{r}}$.

Now it is possible to optimize a little bit the weight of $\mathbf{e}'$ by considering that the support of the secret vector $(\mathbf{x}, \mathbf{y})$ is a random subspace of $\mathbb{F}(q^m)$ of dimension w containing 1, indeed in that case the weight of $\mathbf{e}'$ is upper bounded by $ww_{\mathbf{r}}$ since the support of $\mathbf{e}$ is included in the product of the supports of $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{r}_1, \mathbf{r}_2)$. This does not modify the security proof, and impacts only the value of w in the choice of parameters.

For decoding, we consider Gabidulin $[n, k]$ codes over $\mathbb{F}_{q^n}$, which can decode $\frac{n-k}{2}$ rank errors and choose our parameters such that $ww_{\mathbf{r}} \leq \frac{n-k}{2}$, so that, unlike the Hamming metric case, *there is no decryption failure*.

Parameters and tightness of the reduction. As recalled in the Hamming case, the practical security of the scheme relies on the 2-DRQCSD problem for the public key, for a small weight vector of weight $w = \omega(\mathbf{x}) = \omega(\mathbf{y})$ with $w = \mathcal{O}(\sqrt{n})$. The IND-CPA security of the scheme could be reduced to the 3-DRQCSD problem, decoding a random quasi-cyclic $[3n, n]$ code for a small weight vector $(\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)$. In the proof, the error vectors $\mathbf{r}_1$ and $\mathbf{r}_2$ share the same error support E of dimension $w_{\mathbf{r}}$, for the encryption part the error support of $\mathbf{e}$ can also be taken as E, so that the problem is tightly reduced to the 3-DRQCSD problem for rank metric with weight $w_{\mathbf{r}}$, since all three vectors $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{e}$ have the same error support E of dimension $w_{\mathbf{r}}$. In that case the attacker wants to decode a $[3n, n]$ rank metric code, the best known attack is described in [GRS16, AGHT17]. Since on one hand the attacker wants to attack a length $2n$ code and on the other hand to attack a length $3n$ code, which is easier, we consider different weights for the

secret key $\mathbf{x}, \mathbf{y}$ of weight w and for the random chosen values for the encryption $\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2$ of weight $w_{\mathbf{r}} = w_{\mathbf{e}}$, typically we chose $w \approx \frac{2}{3} w_{\mathbf{r}}$. For the secret key, we consider $1 \in Support(\mathbf{x}, \mathbf{y})$, now since finding a small weight codeword of weight w with support containing 1 is harder than finding a small weight vector of weight $w-1$, we consider $w-1$ for the security reduction to the 2-DRQCSD problem, and the weight $w_{\mathbf{r}} = w_{\mathbf{e}}$ is chosen according to the 3-DRQCSD problem and the best known attacks of [GRS16, AGHT17], whose complexity is given in Section II-D. The best quantum attacks on the rank metric problems follow [GHT16], in that case there is square root gain on the probabilistic part of the attack (details are given in [GHT16]).

Overall the parameters proposed in Tab. II and III correspond to tight reduction for generic instances of the 2-DRQCSD and 3-DRQCSD problems in the rank metric.

Remark 9. *The system is based on cyclic codes, which means considering polynomials modulo $x^n - 1$, interestingly enough, and only in the case of the rank metric, the construction remains valid when considering not only polynomials modulo $x^n - 1$ but also modulo a polynomial with coefficient in the base field $\mathbb{F}_q$. Indeed in that case the modulo does not change the rank weight of a codeword. Such a variation on the scheme may be interesting to avoid potential structural attacks which may use the factorization of the quotient polynomial for the considered polynomial ring.*

Computational Cost. The encryption cost corresponds to a matrix-vector product over $\mathbb{F}_{q^m}$, for a multiplication cost of elements of $\mathbb{F}_{q^m}$ in $m \log(m) \log(\log(m))$, we obtain an encryption complexity in $\mathcal{O}\left(n^2 m \log(m) \log(\log(m))\right)$. The decryption cost is also a matrix-vector multiplication plus the decoding cost of the Gabidulin codes, both have the complexities in $\mathcal{O}\left(n^2 m \log(m) \log(\log(m))\right)$. In practice the different times for encryption and decryption for a 128 bits security are of order 2 ms on a 4GHz computer.

C. Possible optimization for the public key

We saw in the previous section that parameters were chosen to have a tight reduction to generic problems. It is possible to optimize public parameters at the cost of losing the tightness of the reduction. For the Hamming case we consider in the IND-CPA proof a small vector $(\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)$, if one considers the case $w_{\mathbf{r}} = w_{\mathbf{e}}$ the problem corresponds to a general instance of 3-DQCSD problem, but in that case the asymptotic complexity of the attack is in $2^{1.75 w_{\mathbf{r}}}$ rather than $2^{2 w_{\mathbf{r}}}$. A way to improve on this is to consider $w_{\mathbf{e}} > w_{\mathbf{r}}$ for instance $w_{\mathbf{e}} = 2 w_{\mathbf{r}}$, increasing the value of $w_{\mathbf{e}}$ does not really change the decoding but it gives an unbalanced distribution for the error $(w_{\mathbf{r}}, 2w_{\mathbf{r}}, w_{\mathbf{r}})$, in that case the best attacks (for our case with $w_{\mathbf{r}} \sim \sqrt{n}$) give a complexity in $2^{2 w_{\mathbf{r}}}$ rather than $2^{1.75 w_{\mathbf{r}}}$ (the complexity of the attack on the 2-DQCSD problem rather than 3-DQCSD problem). This may allow to improve the parameter by 10% but at the cost of losing the tightness of the reduction. Indeed even if finding a codeword with a $(w_{\mathbf{r}}, 2w_{\mathbf{r}}, w_{\mathbf{r}})$ distribution is harder than finding a codeword with distribution $(w_{\mathbf{r}}, w_{\mathbf{r}}, w_{\mathbf{r}})$ (just add a vector with distribution $(0, w_{\mathbf{r}}, 0)$ to a vector with distribution $(w_{\mathbf{r}}, w_{\mathbf{r}}, w_{\mathbf{r}})$), the problem of finding a vector with an unbalanced distribution is not a generic problem.

For the rank metric the same approach is possible, by considering a support of the error $w_{\mathbf{e}}$ greater than $w_{\mathbf{r}}$ such that $Support(\mathbf{e})$ contains $Support(\mathbf{r}_1, \mathbf{r}_2)$, the problem is equivalent to Hamming metric, the attacker has to attack an "unbalanced" small weight vector in term of support. The problem is, as for the Hamming case, harder than attacking a generic instance for rank $w_{\mathbf{r}}$ but in practice for the best attacks it allows to recover the complexity of attacking a $[2n, n]$ code rather than a $[3n, n]$ code at the cost of decoding a slightly larger error. This type of optimization allows a 20% gain for the size of the public key, but again, as in the Hamming case, at the cost of losing a tight reduction to the generic problems.

D. Comparison with other code-based cryptosystems

In the following we consider the different types of code-based cryptosystems and express different parameters of the different systems in terms of the security parameters λ, considering best known attacks of complexity $2^{\mathcal{O}(w)}$ for decoding a word of weight w for Hamming distance and complexity in $2^{\mathcal{O}(wn)}$ for decoding a word of rank weight w for a code of double-circulant code of length $2n$ for rank metric. McEliece-Goppa corresponds to the original scheme proposed by McEliece [McE78] of dimension rate $\frac{1}{2}$. We present in Tab. IV the different comparisons we obtain, the table is obtained by considering the security of the scheme as 2^{λ} and rewriting the other values in terms of λ from the best known attacks. For instance, for the RQC scheme with $m = \mathcal{O}(n), k = n/2, w = \mathcal{O}(\sqrt{n}), w_{\mathbf{r}} = \mathcal{O}(\sqrt{n})$ corresponding to the type of considered parameters, we obtain a public key in $\mathcal{O}(n^2)$, a security in $\mathcal{O}(2^{\mathcal{O}(nw)})$, taking $w = \mathcal{O}(\sqrt{n})$ leads to $\lambda = \mathcal{O}(n^{\frac{3}{2}})$. From what we deduce $n = \mathcal{O}(\lambda^{\frac{2}{3}})$ and replace the values in n with

	Cryptosystem Parameters								
Instance	n	k	m	q	w	$w_{\mathbf{r}}$	plaintext	key size	security
RQC-I	67	7	89	2	5	6	623	5,963	128
RQC-II	97	13	113	2	6	7	1261	10,961	192
RQC-III	101	5	139	2	6	8	695	14,039	256

Table II
PARAMETER SETS FOR RQC: OUR CRYPTOSYSTEM IN RANK METRIC. THE PLAINTEXTS, KEY SIZES, AND SECURITY ARE EXPRESSED IN BITS.

	Cryptosystem Parameters								
Instance	n	k	m	q	w	$w_{\mathbf{r}}$	plaintext	key size	security
RQC-IV	101	5	113	2	6	8	565	11,413	128

Table III
QUANTUM SAFE PARAMETER SET FOR RQC FOR 128 BITS OF SECURITY. THE PLAINTEXT AND KEY SIZES ARE EXPRESSED IN BITS.

Cryptosystem		Code length	Public key size	Ciphertext size	Existence of a hidden structure	Cyclic structure
Goppa-McEliece	[McE78]	$\mathcal{O}(\lambda \log \lambda)$	$\mathcal{O}(\lambda^2 (\log \lambda)^2)$	$\mathcal{O}(\lambda \log \lambda)$	Yes	No
MDPC	[MTSB13]	$\mathcal{O}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	Yes	Yes
LRPC	[GMRZ13]	$\mathcal{O}(\lambda^{\frac{2}{3}})$	$\mathcal{O}(\lambda^{\frac{4}{3}})$	$\mathcal{O}(\lambda^{\frac{4}{3}})$	Yes	Yes
HQC	[Sec. VII-A]	$\mathcal{O}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	$\mathcal{O}(\lambda^2)$	No	Yes
RQC	[Sec. VII-B]	$\mathcal{O}(\lambda^{\frac{2}{3}})$	$\mathcal{O}(\lambda^{\frac{4}{3}})$	$\mathcal{O}(\lambda^{\frac{4}{3}})$	No	Yes

Table IV
PARAMETER COMPARISON FOR DIFFERENT CODE-BASED CRYPTOSYSTEMS WITH RESPECT TO THE SECURITY PARAMETER λ

values in λ. We apply the same approach for the different rows of the table.

Tab. IV shows that even if the recent cryptosystem MDPC has a smaller public key and a structure easier to hide than for the McEliece cryptosystem, the size of the ciphertext remains non negligible. HQC benefits from the same type of parameters as the MDPC systems but with no hidden structure at the cost of a smaller encryption rate. Finally, the table shows the very strong potential of rank metric based cryptosystems, whose parameters remain rather low compared to MDPC and HQC cryptosystems.

VIII. CONCLUSION AND FUTURE WORK

We have presented an efficient general approach for constructing code-based cryptosystems. This approach originates in Alekhnovich's blueprint [Ale03] on random matrices. Our construction is generic enough so that we provide two instantiations of our framework associated with particular decoding: one for the Hamming metric (HQC), and one for the Rank metric (RQC). Both constructions are efficient and compare favourably to previous works, especially in the rank metric setting. Additionally, for the Hamming metric we provide for the proposed decoding algorithm, an analysis of the error term yielding a concrete, precise and easy-to-verify decryption failure.

This analysis was facilitated by the choice of the decoding algorithm based on a tensor product code. More complex-to-analyze tensor product codes might yield slightly shorter keys and better efficiency.

However, for such a tensor product code, the analysis of the decryption failure probability becomes more complex, and finding suitable upper bounds for it, necessitates future work.

REFERENCES

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *FOCS* 1997.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 92–110. Springer, Heidelberg, August 2007.

[AGHT17] Nicolas Aragon, Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. Improvement of generic attacks on the Rank Syndrome Decoding problem. Preprint available at http://unilim.fr/pages_perso/philippe.gaborit/newGRS.pdf

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1 + 1 = 0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, Heidelberg, April 2012.

[BCGO09] Thierry P. Berger, Pierre-Louis Cayrel, Philippe Gaborit, and Ayoub Otmani. Reducing key length of the McEliece cryptosystem. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 77–97. Springer, Heidelberg, June 2009.

[BS+16] Eli Ben-Sasson, Iddo Bentov, Ivan Damgård, Yuval Ishai, and Noga Ron-Zewi. On Public Key Encryption from Noisy Codewords. In *Public Key Cryptography* pages 417-446. 2016.

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

[BLP08] Daniel J Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. In *Post-Quantum Cryptography*, pages 31–46. Springer, 2008.

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

[BCC+07] Julien Bringer, Hervé Chabanne, Gérard Cohen, Bruno Kindarji, and Gilles Zémor. Optimal iris fuzzy sketches. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–6. IEEE, 2007.

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

[CS16] Rodolfo Canto Torres and Nicolas Sendrier. Analysis of information set decoding for a sub-linear error weight. In Takagi [Tak16], pages 144–161.

[CHJ+02] Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. Gem: A generic chosen-ciphertext secure encryption method. In *Cryptographers' Track at the RSA Conference*, pages 263–276. Springer, 2002.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *Asiacrypt*, volume 2248, pages 157–174. Springer, 2001.

[COT14] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild mceliece over quadratic extensions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 17–39. Springer, 2014.

[COT17] Alain Couvreur, Ayoub Otmani, and Jean-Pierre Tillich. Polynomial time attack on wild mceliece over quadratic extensions. *IEEE Transactions on Information Theory*, 63(1):404–427, 2017.

[DP12] Ivan Damgård and Sunoo Park. Is public-key encryption based on lpn practical? *IACR Cryptology ePrint Archive*, 2012:699, 2012.

[DMQN12] Nico Döttling, Jörn Müller-Quade, and Anderson CA Nascimento. Ind-cca secure cryptography based on a variant of the lpn problem. In *ASIACRYPT*, volume 7658, pages 485–503. Springer, 2012.

[DV13] Alexandre Duc and Serge Vaudenay. Helen: a public-key cryptosystem based on the lpn and the decisional minimal distance problems. In *International Conference on Cryptology in Africa*, pages 107–126. Springer, 2013.

[FdVP08] Jean-Charles Faugère, Françoise Levy dit Vehel, and Ludovic Perret. Cryptanalysis of minrank. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 280–296. Springer, Heidelberg, August 2008.

[FGUO+13] Jean-Charles Faugere, Valérie Gauthier-Umana, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high-rate mceliece cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

[FOPT10] Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In Gilbert [Gil10], pages 279–298.

[FPDP14] Jean-Charles Faugere, Ludovic Perret, and Frédéric De Portzamparc. Algebraic attack against variants of mceliece with goppa polynomial of a special form. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 21–41. Springer, 2014.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, Heidelberg, December 2009.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Crypto*, volume 99, pages 537–554. Springer, 1999.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, pages 1–22, 2013.

[Gab85] Ernest Mukhamedovich Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, 1985.

[GPT91] Ernst M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and thier applications in cryptology. In Donald W. Davies, editor, *EUROCRYPT'91*, volume 547 of *LNCS*, pages 482–489. Springer, Heidelberg, April 1991.

[Gab05] Philippe Gaborit. Shorter keys for code based cryptography. In *Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005)*, pages 81–91, 2005.

[GG07] Philippe Gaborit and Marc Girault. Lightweight code-based identification and signature. In *2007 IEEE International Symposium on Information Theory*, pages 191–195. IEEE, 2007.

[GHT16] Philippe Gaborit, Adrien Hauteville, and Jean-Pierre Tillich. Ranksynd a PRNG based on rank metric. In Takagi [Tak16], pages 18–28.

[GMRZ13] Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC'2013*,

Bergen, Norway, 2013. Available on www.selmer.uib.no/WCC2013/pdfs/Gaborit.pdf.

[GRS16] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, 2016.

[GZ16] Philippe Gaborit and Gilles Zémor. On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Trans. Information Theory 62(12): 7245-7252 (2016).

[Gil10] Henri Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Heidelberg, May 2010.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[GJL15] Qian Guo and Thomas Johansson and Carl Löndahl, A New Algorithm for Solving Ring-LPN With a Reducible Polynomial, In IEEE Trans. Information Theory, vol. 61,(11), pp. 6204–6212, (2015)

[GJS16] Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on mdpc with cca security using decoding errors. In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pages 789–815. Springer, 2016.

[HT15] Adrien Hauteville and Jean-Pierre Tillich. New algorithms for decoding in the rank metric and an attack on the lrpc cryptosystem. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 2747–2751. IEEE, 2015.

[HKL+12] Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An efficient authentication protocol based on ring-lpn. In *Fast Software Encryption*, pages 346–365. Springer, 2012.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423, pages 267–288. Springer, 1998.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. Cryptology ePrint Archive, Report 2017/604, 2017. http://eprint.iacr.org/2017/604.

[HP10] W Cary Huffman and Vera Pless. *Fundamentals of error-correcting codes*. Cambridge university press, 2010.

[KMP14] Eike Kiltz, Daniel Masny, and Krzysztof Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 1–18. Springer, Heidelberg, March 2014.

[LdVP06] Françoise Levy-dit Vehel and L Perret. Algebraic decoding of rank metric codes. *Proceedings of YACC*, 2006.

[LJK+16] Carl Löndahl and Thomas Johansson and Masoumeh Koochak Shooshtari and Mahmoud Ahmadian-Attari and Mohammad Reza Aref, Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension. In *Des. Codes Cryptography*, Vol. 80, pp. 359–377,2016.

[Loi06] Pierre Loidreau. Properties of codes in rank metric. *arXiv preprint cs/0610057*, 2006.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Gilbert [Gil10], pages 1–23.

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

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In *Asiacrypt*, volume 7073, pages 107–124. Springer, 2011.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *EUROCRYPT (1)*, pages 203–228, 2015.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[MB09] Rafael Misoczki and Paulo S. L. M. Barreto. Compact McEliece keys from goppa codes. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *SAC 2009*, volume 5867 of *LNCS*, pages 376–392. Springer, Heidelberg, August 2009.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo SLM Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2069–2073. IEEE, 2013.

[OP01] Tatsuaki Okamoto and David Pointcheval. React: Rapid enhanced-security asymmetric cryptosystem transform. *Topics in Cryptology—CT-RSA 2001*, pages 159–174, 2001.

[RS60] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *35th ACM STOC*, pages 407–416. ACM Press, June 2003.

[Sen11] Nicolas Sendrier. Decoding one out of many. In *International Workshop on Post-Quantum Cryptography*, pages 51–67. Springer, 2011.

[SKK10] Danilo Silva, Frank R Kschischang, and Ralf Kotter. Communication over finite-field matrix channels. *IEEE Transactions on Information Theory*, 56(3):1296–1305, 2010.

[Tak16] Tsuyoshi Takagi, editor. *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, volume 9606 of *Lecture Notes in Computer Science*. Springer, 2016.

APPENDIX

A. Description of the [HHK17] KEM/DEM transformation

1) Reminders on HQC cryptosystem:

- $\mathsf{Setup}(1^\lambda)$: generates the global parameters $n = n(1^\lambda), k = k(1^\lambda)$, $\delta_1 = \delta_1(1^\lambda)$, and $w = w(1^\lambda)$, and specify a generator matrix $\mathbf{G} \in \mathbb{F}^{k\times n}$ of $\mathcal{C}$. The plaintext space is $\mathbb{F}^k$. Outputs $\mathsf{param} = (n, k, \delta, w)$.
- $\mathsf{KeyGen}(\mathsf{param})$: generates $\mathbf{h} \xleftarrow{\$} \mathcal{R}$, matrix $\mathbf{H} = (\mathbf{I}_n \mid \mathbf{rot}(\mathbf{h}))$, $\mathsf{sk} = (\mathbf{x}, \mathbf{y}) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{x}) = \omega(\mathbf{y}) = w$, sets $\mathsf{pk} = \left(\mathbf{h}, \mathbf{s} = \mathsf{sk} \cdot \mathbf{H}^\top\right)$, and returns $(\mathsf{pk}, \mathsf{sk})$.

- $\mathsf{Encrypt}(\mathsf{pk} = (\mathbf{h}, \mathbf{s}), \mathbf{m}, \theta)$: uses randomness θ to generate $\mathbf{e} \xleftarrow{\$} \mathcal{R}$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \xleftarrow{\$} \mathcal{R}^2$ such that $\omega(\mathbf{e}), \omega(\mathbf{r}_1), \omega(\mathbf{r}_2) \leq w$, sets $\mathbf{u}^\top = \mathbf{H}\mathbf{r}^\top$ and $\mathbf{v} = \mathbf{m}\mathbf{G} + \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e}$. It finally returns $\mathbf{c} = (\mathbf{u}, \mathbf{v})$, an encryption of $\mathbf{m}$ under pk.
- $\mathsf{Decrypt}(\mathsf{sk} = (\mathbf{x}, \mathbf{y}), \mathbf{c} = (\mathbf{u}, \mathbf{v}))$: returns $\mathcal{C}.\mathsf{Decode}(\mathbf{v} - \mathbf{u} \cdot \mathbf{y})$.

2) KEM-DEM HQC: Let $\mathcal{E}$ be an instance of the HQC cryptosystem as described above. Let $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{K}$ be hash functions, typically SHA512 as advised by NIST. The KEM-DEM version of the HQC cryptosystem is defined as follows:

- $\mathsf{Setup}(1^\lambda)$: as before, except that k will be the length of the symmetric key being exchanged, typically $k = 256$.
- $\mathsf{KeyGen}(\mathsf{param})$: exactly as before.
- $\mathsf{Encapsulate}(\mathsf{pk} = (\mathbf{h}, \mathbf{s}))$: generate $\mathbf{m} \xleftarrow{\$} \mathbb{F}^k$ (this will serve as a seed to derive the shared key). Derive the randomness $\theta \leftarrow \mathcal{G}(\mathbf{m})$. Generate the ciphertext $c \leftarrow (\mathbf{u}, \mathbf{v}) = \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}, \theta)$, and derive the symmetric key $K \leftarrow \mathcal{K}(\mathbf{m}, \mathbf{c})$. Let $\mathbf{d} \leftarrow \mathcal{H}(\mathbf{m})$, and send $(\mathbf{c}, \mathbf{d})$.
- $\mathsf{Decapsulate}(\mathsf{sk} = (\mathbf{x}, \mathbf{y}), \mathbf{c}, \mathbf{d})$: Decrypt $\mathbf{m}' \leftarrow \mathcal{E}.\mathsf{Decrypt}(\mathsf{sk}, \mathbf{c})$, compute $\theta' \leftarrow \mathcal{G}(\mathbf{m}')$, and (re-)encrypt $\mathbf{m}'$ to get $\mathbf{c}' \leftarrow \mathcal{E}.\mathsf{Encrypt}(\mathsf{pk}, \mathbf{m}', \theta')$. If $\mathbf{c} \neq \mathbf{c}'$ or $\mathbf{d} \neq \mathcal{H}(\mathbf{m}')$ then abort. Otherwise, derive the shared key $K \leftarrow \mathcal{K}(\mathbf{m}, \mathbf{c})$.

According to [HHK17], the KEM-DEM version of HQC is IND-CCA2. See discussion in paragraph "IND-CPA and IND-CCA2" on page 8 (Sec. III-A).