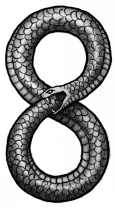
Ouroboros: a simple, secure and efficient key exchange protocol based on coding theory

Jean-Christophe Deneuville[1], Philippe Gaborit[1], and Gilles Zémor[2]

[1] University of Limoges, France
[2] University of Bordeaux, France



Abstract. We introduce Ouroboros[1], a new Key Exchange protocol based on coding theory. The protocol gathers the best properties of the recent MDPC-McEliece and HQC protocols for the Hamming metric: simplicity of decoding and security reduction, based on a double cyclic structure. This yields a simple, secure and efficient approach for key exchange. We obtain the same type of parameters (and almost the same simple decoding) as for MDPC-McEliece, but with a security reduction to decoding random quasi-cyclic codes in the Random Oracle Model.

Keywords: Post-Quantum Cryptography · Coding Theory · Key Exchange

1 Introduction

Code-based cryptography was introduced with the well-known McEliece cryptosystem in 1978: it is in the spirit of the Merkle-Hellman cryptosystem, where the main idea consists in masking an easy instance of a hard problem, hoping that the masking is hard to recover. The McEliece system based on its original family of codes – namely the binary Goppa codes – is still considered unbroken today, but many variants based on alternative families of codes have been proposed over the years and have turned out to be flawed, notably the variants based on the overly structured Reed-Solomon codes. The McEliece system has two main drawbacks: a very large key size and a security reduction to an ad-hoc problem, the difficulty of recovering the hidden structure of a decodable code from the public matrix.

Over the years, researchers have tried to propose alternative schemes to overcome these issues. The first line of improvements consists in adding structure to the public matrix (like cyclicity for instance) in order to decrease the size of the

[1] The Ouroboros symbol is an ancient symbol which represents the notion of cyclicity in many civilizations.

public key. Several approaches were proposed from 2005 [12], and resulted in the McEliece variant based on the MDPC family of error-correcting codes [15], a very efficient family with a very weak structure, compared to classical decodable families. MDPC-McEliece is in the spirit of the NTRU cryptosystem but relies on the Hamming distance rather than on the Euclidean distance. In practice the system has a rather reasonable key-size, but a rather long message-size (comparable to the key-length), it also benefits from a very simple decoding algorithm (the BitFlip algorithm inherited from LDPC codes). Overall, its two main drawbacks are the lack of a security reduction to a classical decoding problem and the fact that the decoding algorithm is only probabilistic, making it hard to obtain precise probabilities of decryption failure for very low probabilities.

A new approach to code-based public-key encryption that broke completely with the McEliece paradigm was proposed by Alekhnovich in 2003 [2]. The focus of this approach is to derive a system with a security reduction to the problem of decoding random linear codes. This approach was very innovative but lead to large parameters, exceeding those of McEliece. An Alekhnovich-inspired approach that features cyclicity was recently proposed in [1]. The new scheme combines the advantages of a security reduction with small public-key sizes resulting from cyclicity and are based on the HQC and RQC (Hamming metric and rank metric quasi-cyclic) families. In practice for the Hamming metric and HQC codes, the obtained parameters are a little larger than for MDPC-McEliece, but the decryption failure is easier to evaluate for very low decryption failure probabilities, and decoding is less simple but still more efficient than for MDPC (decoding a small BCH code against using the BitFlip algorithm for large lengths).

High level overview of our contribution. The previous discussion was mainly focused on encryption algorithms. It is also possible to consider a Key Exchange protocol derived from an encryption algorithm, simply by considering that the public key is ephemeral and changed for each use of the protocol. (This is generally achieved through a Key Encapsulation Mechanism (KEM for short), this point is discussed in more details in Sec. 4.) In that case it is possible to accept low but fixed decryption failures (say) 10^{-5} rather than require proven decryption failures of $2^{-\lambda}$ for a security parameter λ. In that context the very simple BitFlip algorithm for MDPC decoding has renewed appeal since the difficulty of estimating the decoding failure probability is not a serious issue anymore.

Our approach borrows from both MDPC-McEliece and the Alekhnovich approach. In the McEliece paradigm, errors are purposefully added to a codeword, which the receiver can correct because he has a secret version of the code which comes with a decoding algorithm. In contrast, the Alekhnovich strategy consists of creating from a random public code a secret vector that is common to sender and receiver, except that the sender and the receiver's versions of this vector differ by some noise. The natural follow-up is then to resort to an auxiliary code in order to remove this noise. In the present work we use the Alekhnovich approach, except that there is no auxiliary code: the public-key is a random quasi-cyclic code with no extra structure (contrary to McEliece variants) but the noise that needs to be removed is decoded through the secret key that happens to generate an MDPC code.

A structured error for HQC codes. The approach developed in [1] requires recovering a codeword of the form $\mathbf{mG}$, where $\mathbf{G}$ generates some public code of length n, from a quantity of the form $\mathbf{mG} + \mathbf{xr}_2 - \mathbf{yr}_1 + \boldsymbol{\epsilon}$ where $\mathbf{xr}_2 - \mathbf{yr}_1 + \boldsymbol{\epsilon}$ is of weight $\mathcal{O}(n)$, $\mathbf{xr}_2$ and $\mathbf{yr}_1$ are the cyclic products of small weight vectors, and $\boldsymbol{\epsilon}$ is an independent small weight vector. The code generated by $\mathbf{G}$ is therefore chosen to be highly decodable, and in the context of HQC is only required to decode very large errors without taking into account the particular structure of the error. In fact, the errors induced by the HQC approach are very special, indeed looking closely at $\mathbf{xr}_2 - \mathbf{yr}_1 + \boldsymbol{\epsilon}$, and considering the fact that the decoder knows $\mathbf{x}$ and $\mathbf{y}$, it is easy to see that the error has essentially a cyclic structure induced by $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{r}_1, \mathbf{r}_2$ and $\boldsymbol{\epsilon}$ are the unknowns. Seeing this and taking into account the particular error structure, it is easy to reformulate the decoding problem for HQC code into a decoding problem of a quasi-cyclic MDPC code generated by $\mathbf{x}$ and $\mathbf{y}$ (known by the decoder). The only difference being the additional decoding of $\boldsymbol{\epsilon}$, but our experiments show that the BitFlip algorithm can be slightly modified in order to keep handling the case where the syndrome has a small additional error $\boldsymbol{\epsilon}$.

In practice this new approach based on the cyclic structure of the error, enables one to keep the security reduction present in HQC-based encryption and to include the simplicity of the BitFlip decoding algorithm used for MDPC codes (mildly tweaked). In some sense this new approach enables one to avoid the use of an external code as in HQC encryption. (The decoding problem is formally stated in Def. 9.) It comes with a price since it makes the evaluation of decryption failure probabilities more difficult, but the algorithm is especially well suited to Key Exchange for which failures are tolerated. In this paper we show that in practice our parameters are almost the same as those of MDPC-McEliece but with a security reduction to decoding quasi-cyclic random binary codes.

We prove that our protocol satisfies the passively secure requirement for KEMs – namely INDistinguishability under Chosen Plaintext Attacks (IND-CPA) – in the Random Oracle Model, with a reduction to a decisional form of the decoding problem for random QC-codes.

Our contributions. To sum up: by considering the special structure of the error vector in the HQC approach our contributions show the following:

- it is possible to obtain a scheme based on the simple BitFlip decoder, with the IND-CPA property and with a security reduction to a decisional version of the decoding problem for random quasi-cyclic codes, whereas MDPC-McEliece has similar parameters but no such reduction,
- our approach improves on HQC-based encryption since in our new construction, the weight of the error vector that needs to be decoded has weight $\mathcal{O}(\sqrt{n})$ whereas the error weight is structurally in $\mathcal{O}(n)$ for HQC,
- the BitFlip decoder is still usable and decodes efficiently when there is an additional small error on the given syndrome, and
- by considering the use of ephemeral keys, an efficient key exchange protocol is obtained with a reasonable probability of failure.

Organization of the paper. Section 2 gives background, Section 3 describes the new decoding problem, the modified BitFlip algorithm as well as the proposed

Ouroboros protocol, Section 4 presents a security proof of this protocol with respect to the standard model for KEM, and finally Section 5 gives examples of parameters.

2 Background

2.1 Coding theory and syndrome decoding problems

Notation. Throughout this paper, $\mathbb{Z}$ denotes the ring of integers and $\mathbb{F}_q$ (for a prime $q \in \mathbb{Z}$) a finite field, typically $\mathbb{F}_2$ for Hamming codes. Additionally, we denote by $\omega(\cdot)$ the Hamming weight of a vector *i.e.* the number of its non-zero coordinates, and by $\mathcal{S}_w^n(\mathbb{F}_2)$ the set of words in $\mathbb{F}_2^n$ of weight w. Formally:

$$\mathcal{S}_w^n(\mathbb{F}_2) = \{\mathbf{x} \in \mathbb{F}_2^n, \text{ such that } \omega(\mathbf{x}) = w\}.$$

$\mathcal{V}$ denotes a vector space of dimension n over $\mathbb{F}_2$ for some positive $n \in \mathbb{Z}$. Elements of $\mathcal{V}$ can be interchangeably considered as row vectors or polynomials in $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$. Vectors/Polynomials (resp. matrices) will be represented by lower-case (resp. upper-case) bold letters. A prime integer n is said to be primitive if the polynomial $(X^n - 1)/(X - 1)$ is irreducible in $\mathcal{R}$.

For $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, we define their product similarly as in $\mathcal{R}$, *i.e.* $\mathbf{xy} = \mathbf{c} \in \mathcal{V}$ with

$$c_k = \sum_{i+j \equiv k \mod n} x_i y_j, \text{ for } k \in \{0, 1, \ldots, n-1\}.$$

Our new protocol uses cyclic (or circulant) matrices. In the same fashion as in [1], $\mathbf{rot}(\mathbf{h})$ for $\mathbf{h} \in \mathcal{V}$ denotes the circulant matrix whose i^{th} column is the vector corresponding to $\mathbf{h}X^i \mod X^n - 1$.

Background on coding theory. We now provide some reminders on coding theory, the SD problem and its quasi-cyclic versions as defined in [1].

Definition 1 (Quasi-Cyclic Codes [15]). *For positive integers s, n and k, a linear code $[sn, k]$ code is said to be* Quasi-Cyclic (QC) of order s *if* $\forall \mathbf{c} = (\mathbf{c}_1, \ldots, \mathbf{c}_s) \in \mathcal{C}$ *it holds that that* $(\mathbf{c}_1 X, \ldots, \mathbf{c}_s X) \in \mathcal{C}$ *(*i.e. *the code is stable by a block circular shift of length n).*

In our case, we will only consider rate $1/s$ *systematic* quasi-cyclic codes. The parity-check matrix of such codes have the convenient shape below.

Definition 2 (Systematic Quasi-Cyclic Codes of rate $1/s$). *A QC $[sn, n]$ code of order s is said to be* systematic *if it admits a parity-check matrix of the form*

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

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

Problems in coding theory. Most code-based primitives rely on the Syndrome Decoding (SD) problem, which has been proved NP-hard [5]. Even if there is no such complexity result for Quasi-Cyclic (QC) codes, the general belief is that the SD remains hard for such matrices. We use the same notations and definitions as [1] for this problem, namely Quasi-Cyclic Syndrome Decoding (QCSD). The following problems are defined for binary codes in the Hamming metric, but easily extend to codes over $\mathbb{F}_q$ and even to other metrics such as the rank metric.

Definition 3 (SD Distribution). *Let $n, k, w \in \mathbb{N}^*$, the* $\mathsf{SD}(n, k, w)$ Distribution *chooses* $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ *and* $\mathbf{x} \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$*, and outputs* $(\mathbf{H}, \sigma(\mathbf{x}) = \mathbf{H}\mathbf{x}^\top)$.

The SD distribution having been defined, we can now define the fundamental problem for code-based cryptography.

Definition 4 (Search SD Problem). *On input* $(\mathbf{H}, \mathbf{y}^\top) \in \mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{(n-k)}$ *from the SD distribution, the* Syndrome Decoding Problem $\mathsf{SD}(n, k, w)$ *asks to find* $\mathbf{x} \in \mathcal{S}_w^n(\mathbb{F}_2)$ *such that* $\mathbf{H}\mathbf{x}^\top = \mathbf{y}^\top$.

The SD problem has a decisional form, which asks to decide whether the given sample came from the SD distribution or the uniform distribution:

Definition 5 (Decisional SD Problem). *Given* $(\mathbf{H}, \mathbf{y}^\top) \xleftarrow{\$} \mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{(n-k)}$, *the* Decisional 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* $\mathbb{F}_2^{(n-k)\times n} \times \mathbb{F}_2^{(n-k)}$.

In order to propose reasonable key sizes, we base our proposition on QC codes. We adapt the previous problems to this configuration.

Definition 6 (s-QCSD Distribution). *Let $n, k, w, s \in \mathbb{N}^*$, the* s-$\mathsf{QCSD}(n, k, w, s)$ Distribution *samples* $\mathbf{H} \xleftarrow{\$} \mathbb{F}_2^{(sn-k)\times sn}$*, the parity-check matrix of a QC-code of order s and* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \xleftarrow{\$} \mathbb{F}_2^{sn}$ *such that* $\omega(\mathbf{x}_i) = w$*, and outputs* $(\mathbf{H}, \mathbf{H}\mathbf{x}^\top)$.

Definition 7 ((Search) s-QCSD Problem). *For positive integers n, k, w, s, a random parity check matrix* $\mathbf{H}$ *of a systematic QC code* $\mathcal{C}$ *and* $\mathbf{y} \xleftarrow{\$} \mathbb{F}_2^{sn-k}$, *the* Search s-Quasi-Cyclic SD Problem s-$\mathsf{QCSD}(n, k, w)$ *asks to find* $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s) \in \mathbb{F}_2^{sn}$ *such that* $\omega(\mathbf{x}_i) = w$, $i = 1..s$, *and* $\mathbf{y} = \mathbf{x}\mathbf{H}^\top$.

Assumption 1 *The Search s-QCSD problem is hard on average.*

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 use the cyclic structure of the code [19, 13] 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.

Remark. Since systematic quasi-cyclic codes make up a large proportion of the whole ensemble of quasi-cyclic codes, restricting the s-QCSD Problem to systematic codes is not a significant specialisation.

Definition 8 (Decisional s-QCSD Problem). *For positive integers n, k, w, s, a random parity check matrix $\mathbf{H}$ of a systematic QC code $\mathcal{C}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{F}_2^{sn}$, the* Decisional s-Quasi-Cyclic SD Problem s-$\mathsf{DQCSD}(n,k,w)$ *asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^\top)$ came from the s-$\mathsf{QCSD}(n,k,w)$ distribution or the uniform distribution over $\mathbb{F}_2^{(sn-k)\times sn} \times \mathbb{F}_2^{sn-k}$.*

As for the ring Learning Parity from Noise problem, there is no known reduction from the search version of s-QCSD problem to its decisional version. The proof of [4] cannot be directly adapted in the quasi-cyclic case, however the best known attacks on the decisional version of the problem s-QCSD remain the direct attacks on the search version of the problem s-QCSD.

2.2 HQC Scheme

We now recall the Hamming Quasi-Cyclic (HQC) Scheme from [1], which shares some similarities with the proposed protocol. This scheme in turn is inspired by Alekhnovich's proposal based on random matrices [2], but is much more efficient due to the use of the cyclic structure. The main differences between HQC, Alekhnovich's scheme, and our proposal Ouroboros will be discussed in Sec. 3.3.

HQC uses two types of codes, a decodable $[n,k]$ code which can correct δ errors and a random double-circulant $[2n,n]$ code. Using the same notation as before, consider a linear code $\mathcal{C}$ over $\mathbb{F}_2$ of dimension k and length n (generated by $\mathbf{G} \in \mathbb{F}_2^{k\times n}$), that can correct up to δ errors via an efficient algorithm $\mathcal{C}.\mathsf{Decode}(\cdot)$. The scheme consists of the following four polynomial-time algorithms:

- $\mathsf{Setup}(1^\lambda)$: generates the global parameters $n = n(1^\lambda), k = k(1^\lambda)$, $\delta = \delta(1^\lambda)$, and $w = w(1^\lambda)$. The plaintext space is $\mathbb{F}_2^k$. Outputs $\mathsf{param} = (n,k,\delta,w)$.
- $\mathsf{KeyGen}(\mathsf{param})$: generates $\mathbf{q}_r \xleftarrow{\$} \mathcal{V}$, matrix $\mathbf{Q} = (\mathbf{I}_n \mid \mathbf{rot}(\mathbf{q}_r))$, the generator matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ of $\mathcal{C}$, $\mathsf{sk} = (\mathbf{x},\mathbf{y}) \xleftarrow{\$} \mathcal{V}^2$ such that $\omega(\mathbf{x}) = \omega(\mathbf{y}) = w$, sets $\mathsf{pk} = \left(\mathbf{G}, \mathbf{Q}, \mathbf{s} = \mathsf{sk}\cdot\mathbf{Q}^\top\right)$, and returns $(\mathsf{pk},\mathsf{sk})$.
- $\mathsf{Encrypt}(\mathsf{pk} = (\mathbf{G},\mathbf{Q},\mathbf{s}), \boldsymbol{\mu}, \theta)$: uses randomness θ to generate $\boldsymbol{\epsilon} \xleftarrow{\$} \mathcal{V}$, $\mathbf{r} = (\mathbf{r}_1,\mathbf{r}_2) \xleftarrow{\$} \mathcal{V}^2$ such that $\omega(\boldsymbol{\epsilon}), \omega(\mathbf{r}_1), \omega(\mathbf{r}_2) \leq w$, sets $\mathbf{v}^\top = \mathbf{Q}\mathbf{r}^\top$ and $\boldsymbol{\rho} = \boldsymbol{\mu}\mathbf{G} + \mathbf{s}\cdot\mathbf{r}_2 + \boldsymbol{\epsilon}$. It finally returns $\mathbf{c} = (\mathbf{v},\boldsymbol{\rho})$, an encryption of $\boldsymbol{\mu}$ under pk.
- $\mathsf{Decrypt}(\mathsf{sk} = (\mathbf{x},\mathbf{y}), \mathbf{c} = (\mathbf{v},\boldsymbol{\rho}))$: returns $\mathcal{C}.\mathsf{Decode}(\boldsymbol{\rho} - \mathbf{v}\cdot\mathbf{y})$.

A key feature of HQC is that the generator matrix $\mathbf{G}$ of the code $\mathcal{C}$ is publicly known. In this way, the security of the scheme and the ability to decrypt only rely on the knowledge of the secret key to remove sufficiently many errors, so that the code $\mathcal{C}$ being used can decode correctly.

3 The Ouroboros protocol

We begin this Section by restating formally the decoding problem obtained by providing a noisy input to the classical BitFlip Algorithm. We then describe an efficient modified BitFlip algorithm which actually solves the stated problem. Finally we describe our new key exchange protocol: Ouroboros.

3.1 Decoding cyclic errors

Our new key exchange protocol requires to decode cyclic errors. We therefore introduce a new problem that we call the Cyclic Error Decoding (CED) problem. Essentially, this problem asks to recover information hidden with some noise, where the noise has a cyclic structure. The problem is defined as follows:

Definition 9 (Cyclic Error Decoding (CED) Problem). *Let* $\mathbf{x}, \mathbf{y}, \mathbf{r}_1$ *and* $\mathbf{r}_2$ *be random vectors of length* n *and weight* $w = \mathcal{O}(\sqrt{n})$*, and let* $\mathbf{e}$ *be a random error vector of weight* $w_{\mathbf{e}} = cw$ *for some non-negative constant* c*. Considering the cyclic products of vectors modulo* $X^n - 1$*, the problem is defined as follows: given* $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ *and* $\mathbf{e_c} \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \boldsymbol{e}$ *such that* $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$*, the* Cyclic Error Decoding *problem asks to recover* $(\mathbf{r}_1, \mathbf{r}_2)$*.*

One can immediately notice that this problem essentially corresponds to an instance of the SD problem on matrix $\mathbf{H} = \left(\mathbf{rot}(\mathbf{x})^\top, \mathbf{rot}(\mathbf{y}^\top)\right)$, with the particularity that the syndrome itself is faulty. Alternatively, it can also be thought of as a correct instance of the same problem, but on the longer matrix $\mathbf{H} = \left(\mathbf{rot}(\mathbf{x})^\top, \mathbf{rot}(\mathbf{y})^\top, \mathbf{I}_n\right)$.

A modified BitFlip algorithm. In the case when $w_e = 0$, the problem is exactly the MDPC problem [15]: now when $w_e \neq 0$ but remains small, the BitFlip decoder used for MDPC codes can be directly adapted to this case. The only difference is that the STOP condition is not that the weight of the recurring syndrome obtained at each step becomes 0 at some point but rather that its weight is lower than w_e (for $w_e \neq 0$).

We present in Algo. 1 a slightly modified BitFlip algorithm following [15, 9]. Our experiments showed that this Hamming-QC-Decoder algorithm can correctly perform decoding even when the input of the traditional BitFlip algorithm is a moderately noisy syndrome.

There exist different ways to tune the BitFlip algorithm, the reader is referred to [9] to see more details. In our version we consider the simple case where a threshold t is used at each step to make a decision on the bit to flip or not. We run many experiments for different sizes of parameters, in practice the results obtained show that for the parameters considered the w_e impacts decoding only marginally. The main impact is a slightly lower decoding probability.

3.2 Description of the Ouroboros protocol

Our protocol requires a function f which constructs fixed weight vectors of given weight w from an entry r. In general for code-based protocols one requires an invertible function f (see [18]), but in our case since we only consider key exchange, f is not required to be invertible and a simple repetition of a hash function from the entry r, giving the positions of the '1' is enough to obtain random vectors of fixed weight. We denote such a function by f_w.

Description of the protocol. Our protocol is described in a generic fashion in Fig. 1. It uses a hash function $\mathsf{Hash} : \{0,1\}^* \longrightarrow \mathcal{S}_w^n(\mathbb{F}_2)$. For $\mathbf{h}$ a random vector, Alice constructs a random syndrome $\mathbf{s}$ from its secret $\mathbf{x}, \mathbf{y}$. Upon reception of the syndrome $\mathbf{s} = \mathbf{x} + \mathbf{h}\mathbf{y}$ from Alice, Bob constructs its own random syndrome

Algorithm 1: Hamming-QC-Decoder$(\mathbf{x}, \mathbf{y}, \mathbf{e_c}, t, w, w_{\boldsymbol{e}})$

Input: $\mathbf{x}, \mathbf{y}$, and $\mathbf{e_c} = \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \boldsymbol{e}$, threshold value t required to flip a bit, weight w (resp. w_e) of $\mathbf{r}_1$ and $\mathbf{r}_2$ (resp. $\boldsymbol{e}$).

Output: $(\mathbf{r}_1, \mathbf{r}_2)$ if the algorithm succeeds, $\perp$ otherwise.

$(\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{0}, \mathbf{0}) \in (\mathbb{F}_2^n)^2$, $\mathbf{H} \leftarrow \left(\mathbf{rot}(-\mathbf{y})^\top, \mathbf{rot}(\mathbf{x})^\top\right) \in \mathbb{F}_2^{n \times 2n}$, syndrome $\leftarrow \mathbf{e_c}$;
while $[\omega(\mathbf{u}) \neq w$ **or** $\omega(\mathbf{v}) \neq w]$ **and** $\omega(\text{syndrome}) > w_e$ **do**
  sum $\leftarrow$ syndrome$\times\mathbf{H}$; `/* No modular reduction */`
  flipped_positions $\leftarrow \mathbf{0} \in \mathbb{F}_2^{2n}$;
  for $i \in [\![0, 2n-1]\!]$ **do**
    if sum$[i] \geq t$ **then**
      flipped_positions$[i]$ = flipped_positions$[i] \oplus 1$;
  $(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}) \oplus$ flipped_positions;
  syndrome = syndrome $-\mathbf{H} \times$ flipped_positions$^\top$;
if $\omega\left(\mathbf{e_c} - \mathbf{H} \times (\mathbf{u}, \mathbf{v})^\top\right) > w_{\boldsymbol{e}}$ **then**
  return $\perp$;
else
  return $(\mathbf{u}, \mathbf{v})$;

$\mathbf{s} = \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$ from random $\mathbf{r}_1$ and $\mathbf{r}_2$ of weight w, and also constructs a second syndrome $\mathbf{s}_{\boldsymbol{\epsilon}}$ associated with $\mathbf{r}_2$ on one side and on the other side to a small weight vector $\mathbf{e}$ composed of two error vectors: the vector $\boldsymbol{\epsilon}$ which will be the shared secret and the error $\mathbf{e}_r$ obtained from the secret $\mathbf{r}_1, \mathbf{r}_2$. Upon receiving $\mathbf{s_r}$ and $\mathbf{s}_{\boldsymbol{\epsilon}}$, Alice computes $\mathbf{e_c} = \mathbf{s}_{\boldsymbol{\epsilon}} - \mathbf{y}\mathbf{s_r} = \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \mathbf{e_r} + \boldsymbol{\epsilon}$, which corresponds to the cyclic-error decoding problem with $\mathbf{e} = \mathbf{e_r} + \boldsymbol{\epsilon}$. The value $w_{\mathbf{e}}$ is taken as $\omega(\boldsymbol{\epsilon}) + \omega(\mathbf{e})$, in practice it can be a little smaller, but it does not change the decoding.

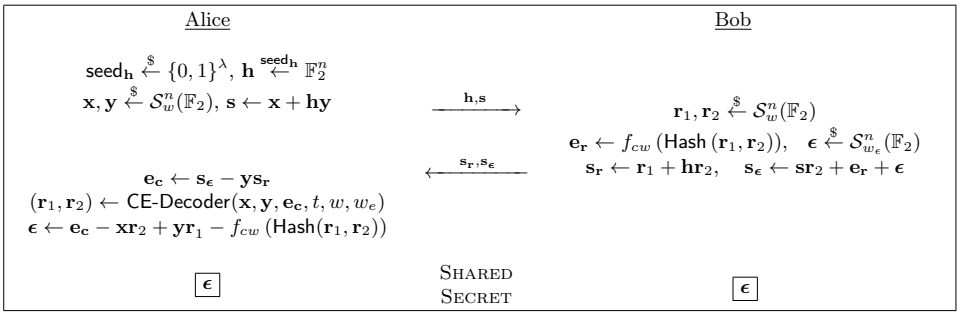


Fig. 1. Description of our new Key Exchange protocol. $\mathbf{h}$ and $\mathbf{s}$ constitute the public key. $\mathbf{h}$ can be recovered by publishing only the λ bits of the seed (instead of the n coordinates of $\mathbf{h}$).

Having this double error is essential for the security proof. Upon reception of the two syndromes $\mathbf{s_r}$ and $\mathbf{s}_{\boldsymbol{\epsilon}}$, Alice constructs an instance of the CED problem. The

result of the CED decoder is then used to recover $\mathbf{e_r} + \boldsymbol{\epsilon}$ and the $\mathbf{e_r}$ part of the error is removed through the knowledge of $(\mathbf{r}_1, \mathbf{r}_2)$.

3.3 Comparison with HQC and Alekhnovich

The Ouroboros approach differs fundamentally from the HQC approach concerning the decoding algorithm used. For HQC [1] (and Aleknovich's approach) the decoding code $\mathcal{C}$ does not depend on the error, the code is fixed and is only required to decode an error of the form $\mathbf{x}\mathbf{r}_2 + \mathbf{y}\mathbf{r}_1 + \boldsymbol{\epsilon}$. Since $\mathbf{x}, \mathbf{y}, \mathbf{r}_1$ and $\mathbf{r}_2$ have weight in $\mathcal{O}(\sqrt{n})$, the code $\mathcal{C}$ has to decode $\mathcal{O}(n)$ errors. For Ouroboros we use the special cyclic structure of the error vector so that the code that is being decoded is necessarily a MDPC type code and the error that one needs to decode has weight $\mathcal{O}(\sqrt{n})$ rather than $\mathcal{O}(n)$. Having to decode a smaller weight error yields better parameters with the Ouroboros approach than with the HQC approach. However there is a price to pay, the BitFlip decoding algorithm leads to a probabilistic decoding where the decoding probability is obtained by simulation and is hard to estimate theoretically, whereas the HQC approach gives the freedom to choose an auxiliary code for decoding with a decoding failure probability easier to estimate.

4 Security of the protocol

In this section we prove the security of our key exchange protocol. Following Alekhnovich's construction, HQC benefits from a security reduction against passive adversaries. This represents a strong advance compared to the MDPC-McEliece scheme. We note that the security proof from [1] carries over to our key exchange protocol.

Security Model. While encryption schemes and long-term key exchange protocols require strong semantic security against active adversaries, protocols meant to exchange purely ephemeral session keys (such as Key Encapsulation Mechanisms *aka* KEMs) are considered secure whenever they provide security against merely passive adversaries (*aka* INDistinguishability under Chosen Plaintext Attacks, or IND-CPA for short). This approach has been followed by several lattice-based key exchange protocols such as [10, 11, 17, 7], or more recently the so-called NewHope protocol [3]. Exchanging ephemeral keys through passively secure KEMS exploits the fact that a (say) 256 bits randomness string chosen by one party can be sent encrypted using the other party's (long term) public key so that both parties end up with shared secret randomness from which they can derive a secret symmetric key. Passively secure KEMs viewed as key exchanged protocols are covered by the IND-CPA security model [14]. (It turns out that this security model has been chosen with a minimal security requirement by Nist in its post-quantum call for proposal [16].) Therefore, we prove our key exchange protocol (viewed as a KEM) to be (passively) secure in this IND-CPA model.

IND-CPA. IND-CPA is generally proved through the following game: the adversary $\mathcal{A}$ chooses two plaintexts μ_0 and μ_1 and sends them to the challenger who flips a coin $b \in \{0, 1\}$, encrypts μ_b into ciphertext c and returns c to $\mathcal{A}$. The encryption

scheme is said to be IND-CPA secure if $\mathcal{A}$ has a negligible advantage in deciding which plaintext c encrypts. This game is formally described on the right.

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

Fig. 2. Experiment against the IND-CPA security

The global advantage for polynomial time adversaries (running in time less than t) is:

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

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

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

Hybrid argument. Alternatively (and equivalently by the hybrid argument), it is possible to construct a sequence of games from a valid encryption of a first message μ_0 to a valid encryption of another message μ_1 and show that these games are two-by-two indistinguishable. We follow this latter approach and prove the security of our protocol (viewed as a KEM) similarly to [1]. Our proof can be thought as similar to [1], without their public code $\mathcal{C}$, and with $\boldsymbol{\epsilon}$ playing the role of the message being encrypted.

Theorem 1. *The protocol presented in Figure 1 is* IND-CPA *under the* 2-*DQCSD and* 3-*DQCSD assumptions.*

The proof is inspired from [1, Proof of Theorem 1], with some slight differences and adjustments. As mentioned at the beginning of this Section, the standard security model for a key exchange protocol such as Ouroboros (or NewHope) is the same as passively secure KEMs [14]. In a KEM spirit, our key exchange protocol can be seen as an ephemeral key encryption protocol where the (long-term) public key is the syndrome $\mathbf{s}$ sent by Alice, and the plaintext (or shared secret randomness is the value $\boldsymbol{\epsilon}$ encrypted in the ciphertext formed by $\mathbf{s_r}$ and $\mathbf{s}_{\boldsymbol{\epsilon}}$.

Proof. Instead of directly proving that an PPT adversary only has a negligible advantage of distinguishing between two encrypted plaintexts, we construct a sequence of game transitioning from a valid encryption of a plaintext to a valid encryption of another plaintext. By showing these games to be two-by-two indistinguishable,

the Hybrid argument allows us to obtain the claimed result. The sequence of games starts with a valid encryption of a message $\boldsymbol{\epsilon}^{(0)}$ and ends with a valid encryption of message $\boldsymbol{\epsilon}^{(1)}$. The aim is to prove that an adversary distinguishing one game from another can be exploited to break either the 2-DQCSD or the 3-DQCSD assumption (respectively on $[2n, n]$ or $[3n, n]$ codes) in polynomial time. Let $\mathcal{A}$ be a probabilistic polynomial time adversary against the IND-CPA of our scheme and consider the following games ($\mathcal{A}$ gets the output ciphertext at the end of each game).

Game $\boldsymbol{G}_1$: This game corresponds to an honest run of the protocol. In particular, the challenger encrypts $\boldsymbol{\epsilon}^{(0)}$ with $\mathbf{x}$, $\mathbf{y}$, $\mathbf{r}_1$ and $\mathbf{r}_2$ of small (*i.e.* correct) weight w.

Game $\boldsymbol{G}_2$: This game is also an honest run of the protocol, still with the same plaintext $\boldsymbol{\epsilon}^{(0)}$ but the challenger uses a random $\mathbf{e_r}' \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ instead of $f_{cw}\left(\mathsf{Hash}(\mathbf{r}_1, \mathbf{r}_2)\right)$.

Game $\boldsymbol{G}_3$: This game differs from $\boldsymbol{G}_2$ in the fact that the challenger uses a random (*i.e.* fake) secret $\mathbf{x}$ and $\mathbf{y}$ random (resulting in a random $\mathbf{s}$). He proceeds to the rest of the protocol honestly to encrypt $\boldsymbol{\epsilon}^{(0)}$.

Game $\boldsymbol{G}_4$: Similar to $\boldsymbol{G}_3$. Additionally, the challenger samples $\mathbf{e_r}'$, $\mathbf{r}_1$ and $\mathbf{r}_2$ at random (resulting in fake $\mathbf{s_r}$ and $\mathbf{s}_{\boldsymbol{\epsilon}}$) to encrypt $\boldsymbol{\epsilon}^{(0)}$.

Game $\boldsymbol{G}_5$: In this game, the challenger creates a fake encryption of another plaintext $\boldsymbol{\epsilon}^{(1)}$ (presumably but not necessarily different from $\boldsymbol{\epsilon}^{(0)}$). He chooses $\mathbf{r}_1', \mathbf{r}_2', \mathbf{e_r}^* \xleftarrow{\$} \mathbb{F}_2^n$ uniformly at random and runs the protocol.

Game $\boldsymbol{G}_6$: Similar to $\boldsymbol{G}_5$, but the challenger encrypts $\boldsymbol{\epsilon}^{(1)}$ using valid, *i.e.* correctly weighted, randomness: $\mathbf{r}_1'$ and $\mathbf{r}_2'$ are sampled with the correct weight w, and $\mathbf{e_r}^* \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$.

Game $\boldsymbol{G}_7$: In this game, the challenger uses a correctly weighted secret key $\mathbf{x}$, $\mathbf{y}$ to encrypt $\boldsymbol{\epsilon}^{(1)}$.

Game $\boldsymbol{G}_8$: In this last game, the challenger uses the hash function to encrypt $\boldsymbol{\epsilon}^{(1)}$, with $\mathbf{e_r}^* \leftarrow f_{cw}\left(\mathsf{Hash}\left(\mathbf{r}_1, \mathbf{r}_2\right)\right)$.

First, games $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are indistinguishable under the Random Oracle assumption.

Secondly, games $\boldsymbol{G}_2$ and $\boldsymbol{G}_3$ are indistinguishable under the 2-DQCSD assumption. Indeed, assume we are given access to an oracle distinguishing these games. Any 2-DQCSD instance $\left(\left(\mathbf{I}_n, \mathbf{rot}(\mathbf{h})\right), \mathbf{s}\right)$ can be viewed as a public key. By providing this public key to the distinguishing oracle, we will be told whether it is valid, which is the configuration of game $\boldsymbol{G}_2$, or not (game $\boldsymbol{G}_3$). But this very key comes from the QCSD distribution in the former case and from the uniform distribution in the latter, which yields a 2-DQCSD oracle.

Then, games $\boldsymbol{G}_3$ and $\boldsymbol{G}_4$ both involve the encryption of the plaintext $\boldsymbol{\epsilon}^{(0)}$, which is known to $\mathcal{A}$, who can hence compute:

$$\begin{pmatrix} \mathbf{s_r} \\ \mathbf{s}_{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{rot}(\mathbf{h}) \\ \mathbf{0} & \mathbf{I}_n & \mathbf{rot}(\mathbf{s}) \end{pmatrix} \left(\mathbf{r}_1, \mathbf{e_r}', \mathbf{r}_2\right)^\top$$

The syndrome $\left(\mathbf{s}_{\mathbf{r}}, \mathbf{s}_{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}^{(0)}\right)$ follows the QCSD distribution in game $\boldsymbol{G}_3$ and the uniform distribution over $\left(\mathbb{F}_2^n\right)^2$ in $\boldsymbol{G}_4$. Assume an adversary is able to distinguish games $\boldsymbol{G}_3$ and $\boldsymbol{G}_4$, then it suffices to provide him with the syndrome and matrix described above to straightforwardly break the 3-DQCSD assumption.

Next, the outputs from games $\boldsymbol{G}_4$ and $\boldsymbol{G}_5$ follow the exact same distribution: they are uniformly random (hence making these games indistinguishable from an information theoretic point of view). Now that the messages being (falsely) encrypted have been permuted, the rest of the proof consists in proving the indistinguishability with a game involving a valid encryption of this second message.

We can start reintroducing correct values in the ciphertext. Games $\boldsymbol{G}_5$ and $\boldsymbol{G}_6$ are indistinguishable using the same argument as between $\boldsymbol{G}_3$ and $\boldsymbol{G}_4$: $\left(\mathbf{s}_{\mathbf{r}}, \mathbf{s}_{\boldsymbol{\epsilon}} - \boldsymbol{\epsilon}^{(1)}\right)$ follows a uniform distribution for $\boldsymbol{G}_5$ versus a QCSD distribution in $\boldsymbol{G}_6$. Therefore an adversary distinguishing these games breaks the 3-DQCSD assumption.

Then, by reintroducing a $(\mathbf{x}, \mathbf{y})$ with correct weight, the argument from the second step also applies and an adversary distinguishing $\boldsymbol{G}_6$ and $\boldsymbol{G}_7$ can identify valid keys from invalid ones, hence breaking the 2-DQCSD assumption.

Finally, games $\boldsymbol{G}_7$ and $\boldsymbol{G}_8$ are again indistinguishable in the Random Oracle Model.

By the hybrid argument, an adversary against the IND-CPA experiment has an advantage (in the Random Oracle Model) bounded by:

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

□

5 Parameter Sets

In this Section, since our Key Exchange protocol is based on an ephemeral encryption algorithm, we keep the same terminology: the public key corresponds to the data that Alice sends to Bob, and the message corresponds to the data sent by Bob to Alice upon receiving Alice's data. In the following we only give parameters for classical attacks, quantum safe parameters are derived by taking the square root of the complexity since the best attacks for our type of parameters $w \ll n$, it was proven in [8] that all known attacks lead to the exact same asymptotical complexity: the complexity of the classical Information Set Decoding (ISD), for which it is possible to apply directly Grover algorithm [6], and hence to divide the bit security level by 2.

5.1 Parameters

The threshold value t is the most sensitive parameter of both the original BitFlip algorithm and the modified one depicted in Algo. 1. A little bit too big and the algorithm misses correct positions, a little bit too low and it includes wrong positions. Chaulet and Sendrier recently conduct a study on the worst-case behaviour

of QC-MDPC codes, and gave some hints on how to choose this threshold value to maximize the error correcting capacity [9]. Based upon their results, we explored several values for t for our context where there is an additional error to consider and chose the lowest t (in order to optimize efficiency) giving a reasonable Decryption Failure Rate (DFR).[2]

The parameters we obtain are given in Tab. 1. For our parameters we chose the weight $w_{\mathbf{e}}$ in Tab. 1 of the additional error $\boldsymbol{\epsilon}$ and the weight of $\mathbf{e_r}$ to be w, so that $w_{\mathbf{e}} = 2w$ in order to fit with the security reduction.

The security of our system is reduced to either decoding a word of weight $2w$ for a $[2n, n]$ or decoding an error of weight $3w$ for a $[3n, n]$ code. For a $[2n, n]$ code the attacker knows the weight of the error is (w, w) on each block of the matrix, a precise analysis is done in [9] and leads to an asymptotic complexity in 2^{2w}. For the case $[3n, n]$ the asymptotic complexity is better since the attacker chooses $\frac{2n}{3}$ columns among $2n$ columns, since the error distribution is (w, w, w) it leads to a complexity in $(\frac{3}{2})^{3w} = 2^{3\log_2(3/2)w} \simeq 2^{1.75w}$, hence a little better than the attack on the $[2n, n]$ code. Notice also that for the MDPC matrix, the weight w has to be taken greater than what we consider in our case since in the case of the MDPC matrix, the attacker can search for all the cyclic permutations of the small weight vector and profit by a factor n for its attack, when in our case the factor is only $\sqrt{n}$ (see [19]). Finally, for our parameters, in order to avoid potential attacks based on the polynomial decomposition of $X^n - 1$, we chose n a primitive prime for $\mathbb{F}_2$. Overall Tab. 1 presents our results, the DFR is obtained by simulations on random instances with given parameters. The results show that our parameters are very close to parameters proposed by MDPC but profit by an IND-CPA security reduction to decoding random quasi-cyclic matrices.

	Ouroboros Parameters					
Instance	n	w	$w_{\mathbf{e}}$	threshold	security	DFR
Low-I	$5,851$	47	94	30	80	$0.92 \cdot 10^{-5}$
Low-II	$5,923$	47	94	30	80	$2.3 \cdot 10^{-6}$
Medium-I	$13,691$	75	150	45	128	$0.96 \cdot 10^{-5}$
Medium-II	$14,243$	75	150	45	128	$1.09 \cdot 10^{-6}$
Strong-I	$40,013$	147	294	85	256	$4.20 \cdot 10^{-5}$
Strong-II	$40,973$	147	294	85	256	$< 10^{-6}$

Table 1. Parameter sets for Ouroboros

[2] This terminology is borrowed from [15]. DFR is the fraction of decoding failures in a given number of decoding tests.

5.2 Optimized parameters

We saw in the previous subsection that the security reduction lead to attacking a $[3n, n]$ quasi-cyclic code, for a small weight error of weight $3w$ more precisely. We also saw that in that case the decoding complexity was lower than for the $[2n, n]$ case. Modifying the weight of $\mathbf{e_r}$ does not really change drastically the decoding capacity of the modified BitFlip algorithm, but it may permit to obtain a higher complexity attack for the $[3n, n]$ matrix of the security reduction. Hence it seems a natural idea to increase the weight of $\mathbf{e_r}$ so that in that case we can still use the modified BitFlip algorithm but the practical security is reduced to decoding a random $[2n, n]$ code for weight $2w$. This is done on the parameters presented in Tab 2.

Notice that without loss of generality for parameters such that $w = \mathcal{O}(\sqrt{n})$ the decoding of vector of length $3n$ with weights of the form (w, w, w), can be reduced to decoding vectors of the form (w, aw, w) for $a > 1$, simply by adding a random known vector of weight $(a-1)w$ on the second n-length block to a (w, w, w) vector, we omit the obvious details of this proof in this short version of the paper.

Suppose the weight of $\mathbf{e_r}$ is aw (with $a > 1$) rather than w, then according to the security reduction, an attacker has to search for a word of the form (w, aw, w). For this case ($w = \mathcal{O}(\sqrt{n}) \ll n$) the best attacks corresponds to the classical ISD approach. When the the weight is regular of the form (w, w, w) the attacker will consider the same number of columns for each block, now for a weight (w, aw, w) the attacker chooses $2n$ columns but will consider more columns where the weight is aw. Let us denote by αn ($0 \leq \alpha \leq 1$) the number of columns for the first and third block and $(2 - 2\alpha)n$ (with $2 - 2\alpha \geq 0$) the number of columns for the second block. The asymptotic probability P that the attacker finds the error columns is hence:

$$P = (\alpha)^w \cdot (2 - 2\alpha)^{aw} \cdot (\alpha)^w.$$

	Ouroboros Optimized Parameters					
Instance	n	w	$w_{\mathbf{e}}$	threshold	security	DFR
Low-I	$4,813$	41	123	27	80	$2.23 \cdot 10^{-5}$
Low-II	$5,003$	41	123	27	80	$2.60 \cdot 10^{-6}$
Medium-I	$10,301$	67	201	42	128	$1.01 \cdot 10^{-4}$
Medium-II	$10,837$	67	201	42	128	$< 10^{-7}$
Strong-I	$32,771$	131	393	77	256	$< 10^{-4}$
Strong-II	$33,997$	131	393	77	256	$< 10^{-7}$

Table 2. Optimized parameter sets for Ouroboros in Hamming metric

For $a = 1$ with the conditions $0 \leq \alpha \leq 1$ and $2 - 2\alpha \geq 0$, we obtain that P is maximal for $\alpha = 2/3$ and we recover the complexity in $2^{1.75w}$, now when a increases

this probability decreases and for $a = 2$ computations show that the maximum P induces a complexity in 2^{2w}, hence considering the case $a = 2.1w$ and $w(\epsilon) = 0.9w$ permits to obtain $w_{\mathbf{e}} \simeq 3w$ for the BitFlip algorithm, and permits to obtain that the best attacks of the system are obtained for decoding $2w$ errors for a $[2n, n]$ quasi-cyclic code. This permits to obtain better parameters (about 20% better in terms of size of public key) and which are presented in Tab. 2. These parameters are very similar to the parameters proposed for MDPC-McEliece.

6 Conclusion

In this paper we introduced Ouroboros: an efficient, secure and conceptually simple key exchange protocol based on coding theory. This new protocol benefits from the security proof of the HQC and RQC family based on the Alekhnovich approach, and have an IND-CPA security reduction to decoding random quasi-cyclic codes, moreover because of its inherent double circulant structure it also benefits from the simple MDPC structure and the simple BitFlip decoding algorithm, for almost the same type of parameters as MDPC codes but with better parameters than for the HQC protocol (about 40% better for the same DFR).

While the approach is presented only for the Hamming metrics, it is possible to implement a rank metric analog: Ouroboros-R. The resulting protocol also yields better parameters (about 20% better) in comparison to the RQC approach and also to benefits from the simple decoding algorithm of LRPC codes. The price to pay is a probabilistic decoding, which makes this approach especially well suited for Key Exchange. Ouroboros-R will be described into more details in an extended version of this work.

The Ouroboros protocol leads to somewhat higher public key parameters than the recent lattice-based key exchange NewHope protocol [3] but Ouroboros-R has the potential to give better parameters than the NewHope protocol.

References

[1] Aguilar Melchor, C., Blazy, O., Deneuville, J.C., Gaborit, P., Zémor, G.: Efficient encryption from random quasi-cyclic codes. CoRR **abs/1612.05572** (2016) http://arxiv.org/abs/1612.05572. *2, 3, 4, 5, 6, 9, 10*

[2] Alekhnovich, M.: More on average case vs approximation complexity. In: 44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings. (2003) 298–307 http://www.cs.toronto.edu/~toni/Courses/PCP/handouts/misha.pdf. *2, 6*

[3] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - A new hope. In Holz, T., Savage, S., eds.: 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016., USENIX Association (2016) 327–343 https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_alkim.pdf. *9, 15*

[4] Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with constant input locality. In Menezes, A., ed.: Advances in Cryptology - CRYPTO 2007, 27th

Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings. Volume 4622 of Lecture Notes in Computer Science., Springer (2007) 92–110 http://www.eng.tau.ac.il/~bennyap/pubs/input-locality-full-revised-1.pdf. 6

[5] Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory **24**(3) (1978) 384–386 http://authors.library.caltech.edu/5607/1/BERieeetit78.pdf. 5

[6] Bernstein, D.J.: Grover vs. mceliece. In Sendrier, N., ed.: Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings. Volume 6061 of Lecture Notes in Computer Science., Springer (2010) 73–80 https://cr.yp.to/codes/grovercode-20091123.pdf. 12

[7] Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy, IEEE Computer Society Press (May 2015) 553–570 http://eprints.qut.edu.au/86651/1/main.pdf. 9

[8] Canto Torres, R., Sendrier, N.: Analysis of information set decoding for a sub-linear error weight. In Takagi, T., ed.: 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) 144–161 http://dx.doi.org/10.1007/978-3-319-29360-8_10. 12

[9] Chaulet, J., Sendrier, N.: Worst case qc-mdpc decoder for mceliece cryptosystem. In: Information Theory (ISIT), 2016 IEEE International Symposium on, IEEE (2016) 1366–1370 https://arxiv.org/pdf/1608.06080.pdf. 7, 13

[10] Ding, J.: New cryptographic constructions using generalized learning with errors problem. Cryptology ePrint Archive, Report 2012/387 (2012) http://eprint.iacr.org/2012/387.pdf. 9

[11] Ding, J., Xie, X., Lin, X.: A simple provably secure key exchange scheme based on the learning with errors problem. Cryptology ePrint Archive, Report 2012/688 (2012) http://eprint.iacr.org/2012/688. 9

[12] Gaborit, P.: Shorter keys for code based cryptography. In: Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005). (2005) 81–91 http://www.unilim.fr/pages_perso/philippe.gaborit/shortIC.ps. 2

[13] Hauteville, A., Tillich, J.P.: New algorithms for decoding in the rank metric and an attack on the lrpc cryptosystem. In: 2015 IEEE International Symposium on Information Theory (ISIT), IEEE (2015) 2747–2751 https://arxiv.org/pdf/1504.05431.pdf. 5

[14] Herranz, J., Hofheinz, D., Kiltz, E.: KEM/DEM: Necessary and sufficient conditions for secure hybrid encryption. Cryptology ePrint Archive, Report 2006/265 (2006) http://eprint.iacr.org/2006/265.pdf. 9, 10

[15] Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.: Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In: Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on, IEEE (2013) 2069–2073 https://eprint.iacr.org/2012/409.pdf. 2, 4, 7, 13

[16] National Institute of Standards and Technology: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (call for proposal) (December 2016) http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf. *9*

[17] Peikert, C.: Lattice cryptography for the internet. In Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Volume 8772 of Lecture Notes in Computer Science., Springer (2014) 197–219 http://web.eecs.umich.edu/~cpeikert/pubs/suite.pdf. *9*

[18] Sendrier, N.: Encoding information into constant weight words. In: Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on, IEEE (2005) 435–438 http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1523371. *7*

[19] Sendrier, N.: Decoding one out of many. In: International Workshop on Post-Quantum Cryptography, Springer (2011) 51–67 https://eprint.iacr.org/2011/367.pdf. *5*, *13*