Ouroboros-E: An efficient Lattice-based Key-Exchange Protocol

Jean-Christophe Deneuville*, Philippe Gaborit†, Qian Guo‡§ and Thomas Johansson‡
*LIFO – INSA-CVL Bourges, 88 boulevard Lahitolle, 18 000 Bourges, France
jean-christophe.deneuville@insa-cvl.fr
†XLIM – University of Limoges, 123 avenue Albert Thomas, 87 000 Limoges, France
philippe.gaborit@unilim.fr
‡Lund University, Box 117, 221 00 Lund, Sweden
{qian.guo, thomas.johansson}@eit.lth.se
§University of Bergen, Box 7803, N-5020 Bergen, Norway
qian.guo@uib.no

***Abstract*—The Bit Flipping algorithm is a hard decision decoding algorithm originally designed by Gallager in 1962 to decode Low Density Parity Check Codes (LDPC). It has recently proved to be much more versatile, for Moderate Parity Check Codes (MDPC) or Euclidean metric. We further demonstrate its power by proposing a noisy Euclidean version of it. This tweak allows to construct a lattice based key exchange analogous to the Ouroboros protocol for Hamming metric but with a reduction to the Short Integer Solution (SIS) problem. The very efficient decoding algorithm permits to consider smaller alphabets than for NTRU or Ring-LWE decryption algorithms. Overall we obtain a new protocol which competes with the recent NEWHOPE and KYBER proposals, and also with NTRU. The resulting scheme exploits the cyclicity of the error, and benefits from the security of the renowned SIS problem.**

I. INTRODUCTION

The NTRU lattice-based protocol was introduced in 1996. This protocol was a major breakthrough since it made practical lattice-based cryptography by the use of cyclicity, which made the size of the public key very small. Meanwhile the security of the system did not rely on a classical problem like SIS but rather relied on finding small (Euclidean) weight vectors in a special matrix generated by small weight vectors. Even if this type of problem is still hard to attack, it remains a specific and not a general problem (although for extreme parameters, not used in practice, such a reduction could be obtained [16]). This means that there may exist specific better attacks for this problem. A few years later, Regev introduced the LWE algorithm, based on a general problem, and in 2010 an ideal-ring version was proposed in [13]. The latter has the advantage to be both practical (like NTRU) and to benefit from a reduction to a general problem based on ideal-rings. A similar idea of using a dual small weight vector generator matrix as trapdoor was also used for other metrics: MDPC codes [14] (based on Gallager's LDPC codes) for Hamming metric, and LRPC codes [8] for rank metric. These coding approaches see the decryption algorithm purely as a decoding algorithm, when the NTRU decryption algorithm does not decode but recover the plaintext by using modular properties and the fact that there is a controlled noise of quadratic size. An equivalent approach to RLWE was also proposed in 2016 for Hamming and rank metric, the HQC and RQC protocols [1].

In 2016, Guo and Johansson [9], [10] proposed a decoding algorithm (and an encryption scheme) for the Euclidean metric in the spirit of MDPC and LRPC codes but as for NTRU, the associated encryption algorithm relies on finding small weight vectors in a specific matrix. The advantage of their approach is that as for MDPC and LRPC codes, the purely decoding approach is more efficient than the decryption approach of NTRU.

In 2017 Deneuville *et al.* [7] introduced Ouroboros for Hamming metric which is based on a modification of the MDPC algorithm: a noisy-MDPC decoder. This permits to benefit from the nice feature of the MDPC decoder for decryption, together with a reduction to a generic problem (decoding random quasi-cyclic codes) rather than on a specific problem (finding very small vectors in a code generated by very small weight vectors). In this present paper we adapt the Ouroboros approach to Euclidean metric context.

The different previous approaches for ideal-lattices are as follows:

- NTRU has a security reduction to a specific problem (NTRU lattices) and deals with quadratic size error (of the form $\mathbf{x}\cdot\mathbf{y}$ for $\mathbf{x}$ and $\mathbf{y}$ small weight vectors embedded with polynomial multiplication). NTRU has an efficient decryption algorithm but has to deal with quadratic size errors,
- RLWE has a reduction to a general problem and deals with quadratic size errors of the form $\mathbf{x}_1 \cdot \mathbf{y}_2 + \mathbf{x}_2 \cdot \mathbf{y}_1$ with an efficient decryption algorithm,
- Guo-Johansson decryption algorithm is based on a decoding approach (a p-ary Bit Flipping algorithm in the spirit of MDPC or LRPC codes), for NTRU-like matrices. It is not necessarily efficient for decoding large weight errors, but is very efficient to decode small norm errors with small coordinates (typically $\{-1, 0, 1\}$), so that in practice it is more efficient to deal with smaller errors for a less efficient algorithm than dealing with larger errors

for a more efficient decoding algorithm (like NTRU or RLWE). In practice it means that it is possible to consider smaller alphabets than for NTRU or RLWE for the same type of decryption failure rate (DFR). The security reduction of the Guo-Johansson encryption algorithm is similar to NTRU with a specific problem.

In this paper we develop the Ouroboros approach for the Guo-Johansson encryption algorithm which permits to obtain as for RLWE a new encryption scheme with a security reduction to a general problem, but also benefits from an efficient decryption algorithm: the noisy p-ary Bit Flipping algorithm for Euclidean metric, applied on smaller errors to decode. The general idea of Ouroboros is to use the structure of the quadratic size error to decode: starting from a quadratic size errors of the form $\mathbf{x}_1 \cdot \mathbf{y}_2 + \mathbf{x}_2 \cdot \mathbf{y}_1 + \mathbf{e}$. The Ouroboros approach sees the error as a syndrome associated to an error vector $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{e})$ for a matrix built from y_1 and y_2. So that the decryption consists in recovering $\mathbf{x}_1$ and $\mathbf{x}_2$ from $\mathbf{x}_1, \mathbf{x}_2, \mathbf{e}$ knowing $\mathbf{y}_1$ and $\mathbf{y}_2$: the noisy-q-ary-BitFlipping algorithm introduced in this paper.

Overall our new protocol benefits from the good features of the Guo-Johansson decoding algorithm but has also a reduction to a general problem. In practice it permits to obtain much smaller alphabets than for NTRU, RLWE or the recent Kyber [5] and NewHope [2]. We consider alphabets of size 512 when other protocols consider sizes between 2048 and 12289. Our approach may permit to obtain a 20% gain in the size of the public key for similar security levels and DFR.

Organization. The rest of this paper is organized as follows: Sec. II provides some background on notations and the SIS problem, then a lattice version of the Bit Flipping algorithm is proposed in Sec. III, along with a key exchange protocol. A security proof is presented in Sec. IV.

II. Background

A. Notations

Throughout this paper, $\mathbb{Z}$ denotes the ring of integers and $\mathbb{F}_q$ (for $q \in \mathbb{Z}$ a prime or a power of a prime) a finite field. Vectors and polynomials (resp. matrices) will be denoted in bold lower (resp. upper) case. While the Coding Theory community uses row vectors, fellows from Lattices prefer column notation. Since our work considers both fields, we do not specify a particular notation and leave it implicit in the context. Additionally, we denote by $\omega(\cdot)$ and $\|\cdot\|$ respectively the Hamming weight of a vector and its ℓ_2 euclidean norm. More formally, for $\mathbf{v} \in \mathbb{F}_q^n$, we have: $\omega(\mathbf{v}) = \mathrm{card}(\{i \in [\![0, n-1]\!] \text{ s.t. } v_i \neq 0\})$, and$\|\mathbf{v}\| = \sqrt{\sum_{i=0}^{n-1} v_i^2}$. Additionally, we denote $\mathcal{S}_w^n(\mathbb{F})$ the set of words in $\mathbb{F}^n$ whose metric is equal to w. Formally: $\mathcal{S}_w^n(\mathbb{F}_2) = \{\mathbf{x} \in \mathbb{F}_2^n, \text{ such that } \omega(\mathbf{x}) = w\}$ for Hamming metric, and $\mathcal{S}_w^n(\mathbb{F}_q) = \{\mathbf{x} \in \mathbb{F}_q^n, \text{ such that } \|\mathbf{x}\| = w\}$ for Euclidean metric.

Let $\mathcal{V}$ denotes a vector space of dimension n over $\mathbb{F}_q$ for some positive $n \in \mathbb{Z}$. Elements of $\mathcal{V}$ can be interchangeably considered as vectors or polynomials in $\mathcal{R} = \mathbb{F}_q[x]/(x^n - 1)$.

For $\mathbf{u}, \mathbf{v} \in \mathcal{V}$, we define their product similarly as in $\mathcal{R}$, *i.e.* $\mathbf{uv} = \mathbf{w} \in \mathcal{V}$ with $w_k = \sum_{i+j \equiv k \mod n} u_i v_j$, for $k \in [\![0, n-1]\!]$. Additionally, $\mathcal{I}_d = \{-d, \ldots, 0, \ldots, d\}$, $\mathbf{x}_0^{(d)} = \lfloor \frac{q}{2d+1} \rceil (1, 0, 0, \ldots, 0)$, and $\mathbf{y}_0^{(d)} = \lfloor \frac{q}{(2d+1)^2} \rceil (1, 0, 0, \ldots, 0)$.

B. Lattice problems

In this Section, we recall one of the most famous average-case lattice problems that has a connection to worst-case lattice problems, namely the *Short Integer Solution* (SIS). The security of the Ouroboros-E protocol described in Sec. III-A will rely on the hardness of this problem.

Definition 1 (SIS$_{q,n,m,\beta}$ problem). *Given* $\mathbf{A} \xleftarrow{\$} \mathbb{F}_q^{n \times m}$*, the* SIS$_{q,n,m,\beta}$ problem *asks to find a non-zero vector* $\mathbf{z} \in \mathbb{Z}^m$ *such that* $\mathbf{Az} = \mathbf{0}$ *and* $\|\mathbf{z}\| \leq \beta$.

Definition 2 (SIS$_{q,n,m,d}$ distribution). *Sample* $\mathbf{A} \xleftarrow{\$} \mathbb{F}_q^{n \times m}$ *and* $\mathbf{z} \xleftarrow{\$} \{-d, \ldots, 0, \ldots, d\}^m$ *uniformly at random, and output* $(\mathbf{A}, \mathbf{Az})$.

Definition 3 (Search SIS$_{q,n,m,d}$). *Let* χ *denote the* SIS$_{q,n,m,d}$ *distribution. Given* $(\mathbf{A}, \mathbf{t}) \leftarrow \chi$*, find* $\mathbf{z} \in \{-d, \ldots, 0, \ldots, d\}^m$ *such that* $\mathbf{Az} = \mathbf{t}$.

Definition 4 (Decisional SIS$_{q,n,m,d}$). *Given* $(\mathbf{A}, \mathbf{t}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^n$*, the* SIS$_{q,n,m,d}$ decisional problem *asks to decide with non-negligible advantage whether it comes from the* SIS$_{q,n,m,d}$ *distribution or the uniform distribution over* $\mathbb{F}_q^{n \times m} \times \mathbb{F}_q^n$.

A simple pigeonhole argument allows to see that in order for a solution to the problem to exist, both the norm β and the number of columns m must be large enough, namely $\beta \geq \sqrt{\lceil n \log_2 q \rceil}$ and $m \geq \lceil n \log_2 q \rceil$. For the last two problems where an upper bound d on the absolute value of the coefficients is provided, the relation between the parameters defines the hardness of the problem. For $d \ll q^{n/m}$, the SIS$_{q,n,m,d}$ problem has only one solution with high probability, and the problem is said of *low-density*. On the contrary for $d \gg q^{n/m}$, the SIS$_{q,n,m,d}$ distribution gets closer to the uniform distribution $\mathcal{U}\left(\mathbb{F}_q^{n \times m} \times \mathbb{F}_q^n\right)$. Therefore, it is reasonable to argue that the hardest instances of the SIS problem are those for which $d \approx q^{n/m}$ [15].

III. Proposal

A. Euclidean Ouroboros: Ouroboros-E

To construct a lattice key exchange analogue of the Ouroboros protocol, we need to introduce some modifications. We adopt in the general case that $\mathbf{x} = \mathbf{x}_0^{(d)} + \hat{\mathbf{x}}$ and $\mathbf{y} = \mathbf{y}_0^{(d)} + \hat{\mathbf{y}}$, where d is a suitable constant and $\hat{\mathbf{x}}, \hat{\mathbf{y}} \xleftarrow{\$} \mathcal{I}_d^n$.

Sampling at random from $\mathcal{I}_d^n$ instead of $\mathcal{S}_w^n(\mathbb{F}_q)$ with $d \approx O(w/\sqrt{n}) \approx O(1)$ yields a more efficient variant. In this case, we can let $\mathbf{x} = \mathbf{x}_0^{(d)} + \hat{\mathbf{x}}$ and $\mathbf{y} = \mathbf{y}_0^{(d)} + \hat{\mathbf{y}}$, which is preferred in decoding. We have other proposals in-between, by using a different noise distribution.

We also introduce the noisy-q-ary-BitFlipping, described in Alg. 1. It is a fast decoding algorithm for the efficient variant with $d = 1$ which is the case we propose. The generalization to $d > 1$ is direct, but additional optimization is necessary.

With these modifications we now give the new Ouroboros-E protocol (Ouroboros using Euclidean metric) using the proposed noisy-q-ary-BitFlipping algorithm, depicted in Fig. 1. It requires a Hash function from $\{0,1\}^*$ which produce small norm vectors in $\mathcal{I}_d^n$. Such functions can be found in NewHope [2] for instance.

B. A noisy q-ary bit flipping algorithm

We modify the q-ary iterative decoding algorithm [10] in Alg. 1 with tweaks for managing noisy syndromes and introducing efficient performance. The algorithm takes as input the two known vectors $\mathbf{x}, \mathbf{y}$ of the form as described above, a known (syndrome) vector $\mathbf{e_c}$, and algorithm parameter t for the number of iterations. The decoding algorithm will deliver $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{I}_d^n$ such that $\mathbf{e_c} - \mathbf{x}\mathbf{r}_2 + \mathbf{y}\mathbf{r}_1$ is very small.

This new algorithm is a one-step hard-decision version aiming for the decoding efficiency, which also adopts the special structure of the Ouroboros family of Key Exchange protocols [7]. In the description, the **rot** operation transforms a polynomial (vector) over the ring $\mathcal{R}$ to a matrix over $\mathbb{F}_q$.

As we only consider $d = 1$ in the proposal, we have 9 signal points to place (two significant entries are chosen in a parity-check equation and each of them is sampled from $\{-1, 0, 1\}$). We equally divide the cycle $[\![0, q]\!] \pmod q$ into 9 intervals and make decisions accordingly. The latter is trivial and adopted in almost all lattice-based cryptosystems.

The non-trivial part is that, other than some small noise added (e.g., the noise variable $\mathbf{e}$ in Alg. 1), the noise entries in one parity-check equation are also signal points in several other parity-check equations, which can be corrected iteratively. Thus, if the majority of the secret entries are correctly decided in the first iteration, then the noise in the second iteration will be much smaller, which leads to better decisions, and so on and so forth.

Therefore, the field size can be highly reduced as we only need a good success probability in the first iteration – in contrast to other schemes requiring the one-iteration[1] error probability close to 0.

IV. Security

A. Security model

Key Encapsulation Mechanism (KEM). The proposed Key Exchange protocol follows the KEM paradigm. It is meant to exchange (ephemeral) session keys using a public key encryption scheme. Therefore, it is sufficient to prove that the scheme satisfies INDistinguishability under Chosen Plaintext Attacks (IND-CPA for short).

Beyond passive adversaries. Notice that it is pretty straightforward to obtain a CCA-secure KEM from a CPA onesimply by involving the message (or key being exchanged) in the hash (the value committed by Bob in the second step). On recovering Bob's randomness $\mathbf{r}_1$ and $\mathbf{r}_2$, Alice can check whether Bob followed the protocol or tried to fool her.

Algorithm 1: Noisy-q-ary-BitFlipping$(\mathbf{x}, \mathbf{y}, \mathbf{e_c}, t)$

```
Input: $\mathbf{x}, \mathbf{y}$, $\mathbf{e_c} = \mathbf{x}\mathbf{r}_1 + \mathbf{y}\mathbf{r}_2 + \mathbf{e}$, and number of
       iterations $t$, where $\mathbf{x} = \mathbf{x}_0^{(1)} + \hat{\mathbf{x}}$, $\mathbf{y} = \mathbf{y}_0^{(1)} + \hat{\mathbf{y}}$,
       and $\hat{\mathbf{x}}, \hat{\mathbf{y}}, \mathbf{r_1}, \mathbf{r_2}, \mathbf{e} \overset{\$}{\leftarrow} \mathcal{I}_1^n$
Output: $(\mathbf{r}_1, \mathbf{r}_2)$ if the algorithm succeeds, $\perp$ otherwise.
$(\mathbf{u}, \mathbf{v}) \leftarrow (\mathbf{0}, \mathbf{0}) \in \left(\mathbb{F}_q^n\right)^2$,
  $\mathbf{H} \leftarrow \left(\mathbf{rot}(\mathbf{x})^\top, \mathbf{rot}(\mathbf{y})^\top\right) \in \mathbb{F}_q^{n \times 2n}$, psyndrome $\leftarrow \mathbf{e_c}$;
for $i = 0$; $i \leq t$; $i++$ do
    $(\mathbf{u}_{\mathsf{temp}}, \mathbf{v}_{\mathsf{temp}}) \leftarrow (\mathbf{0}, \mathbf{0}) \in \left(\mathbb{F}_q^n\right)^2$;
    for $j \in [\![0, n-1]\!]$ do
        if psyndrome$[j] \in [\![\lceil \frac{q}{6} \rceil, \lfloor \frac{q}{2} \rfloor]\!]$ then
            $\mathbf{u}_{\mathsf{temp}}[j] = 1$;
        else if psyndrome$[j] \in [\![\lceil \frac{q}{2} \rceil, \lfloor \frac{5q}{6} \rfloor]\!]$ then
            $\mathbf{u}_{\mathsf{temp}}[j] = -1$;
        else
            $\mathbf{u}_{\mathsf{temp}}[j] = 0$;
        if (psyndrome$[j] \mod \lfloor \frac{q}{3} \rceil) \in [\![\lceil \frac{q}{18} \rceil, \lfloor \frac{q}{6} \rfloor]\!]$ then
            $\mathbf{v}_{\mathsf{temp}}[j] = 1$;
        else if (psyndrome$[j] \mod \lfloor \frac{q}{3} \rceil) \in [\![\lceil \frac{q}{6} \rceil, \lfloor \frac{5q}{18} \rfloor]\!]$
         then
            $\mathbf{v}_{\mathsf{temp}}[j] = -1$;
        else
            $\mathbf{v}_{\mathsf{temp}}[j] = 0$;
    $\mathbf{u} \leftarrow \mathbf{u} + \mathbf{u}_{\mathsf{temp}}$;
    $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{v}_{\mathsf{temp}}$;
    for $j \in [\![0, n-1]\!]$ do
        if $\mathbf{u}[j] = 2$ then
            $\mathbf{u}[j] = -1$;
        if $\mathbf{u}[j] = -2$ then
            $\mathbf{u}[j] = 1$;
        if $\mathbf{v}[j] = 2$ then
            $\mathbf{v}[j] = -1$;
        if $\mathbf{v}[j] = -2$ then
            $\mathbf{v}[j] = 1$;
    psyndrome $\leftarrow \mathbf{e_c} - \mathbf{H} \times (\mathbf{u}, \mathbf{v})^\top$;
    if Entries in psyndrome are all $\in \{-1, 0, 1\}$ then
        return $(\mathbf{u}, \mathbf{v})$;
return $\perp$;
```

B. Main theorem

The semantic security of Ouroboros-E relies on the following problem: finding the exchanged secret given pk and the transcripts. (Lattice attacks targeting key-recovery are covered Sec. IV-C.) The following (main) theorem reduces these two problems to the hardness of the SIS problem.

Theorem 1. *Ouroboros-E is IND-CPA secure under the* SIS *assumption.*

[1] These decoding procedures only run one iteration.

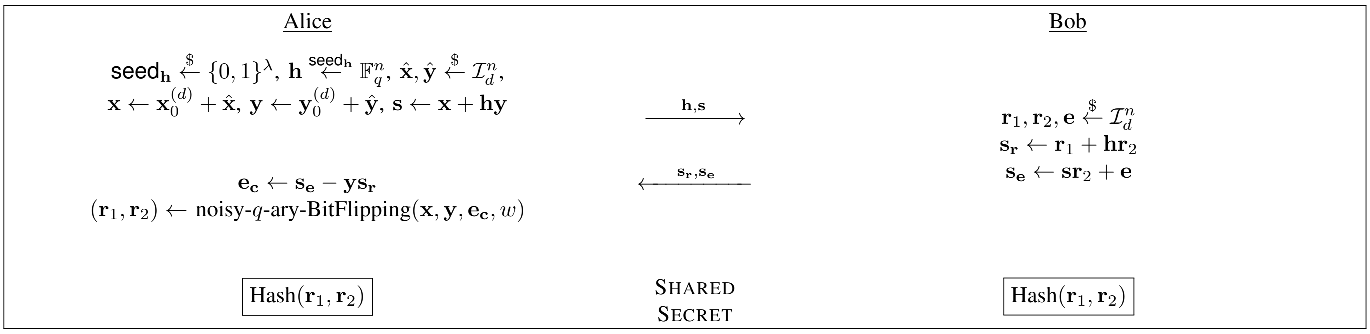
Alice
Bob
$\mathsf{seed}_{\mathbf{h}} \overset{\$}{\leftarrow} \{0,1\}^{\lambda}$, $\mathbf{h} \overset{\mathsf{seed}_{\mathbf{h}}}{\leftarrow} \mathbb{F}_q^n$, $\hat{\mathbf{x}}, \hat{\mathbf{y}} \overset{\$}{\leftarrow} \mathcal{I}_d^n$,
$\mathbf{x} \leftarrow \mathbf{x}_0^{(d)} + \hat{\mathbf{x}}$, $\mathbf{y} \leftarrow \mathbf{y}_0^{(d)} + \hat{\mathbf{y}}$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{h}\mathbf{y}$
$\xrightarrow{\mathbf{h},\mathbf{s}}$
$\mathbf{r}_1, \mathbf{r}_2, \mathbf{e} \overset{\$}{\leftarrow} \mathcal{I}_d^n$
$\mathbf{s_r} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$
$\mathbf{s_e} \leftarrow \mathbf{s}\mathbf{r}_2 + \mathbf{e}$
$\xleftarrow{\mathbf{s_r},\mathbf{s_e}}$
$\mathbf{e_c} \leftarrow \mathbf{s_e} - \mathbf{y}\mathbf{s_r}$
$(\mathbf{r}_1, \mathbf{r}_2) \leftarrow$ noisy-q-ary-BitFlipping$(\mathbf{x}, \mathbf{y}, \mathbf{e_c}, w)$
$\boxed{\mathrm{Hash}(\mathbf{r}_1, \mathbf{r}_2)}$
SHARED SECRET
$\boxed{\mathrm{Hash}(\mathbf{r}_1, \mathbf{r}_2)}$

Fig. 1. General description of our Ouroboros-E 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 $\mathbf{h}$).

The proof follows the same ideas as the original one from [1, Proof of Theorem 1].

Proof. To prove the IND-CPA property, we need to prove that an adversary $\mathcal{A}$ is unable to distinguish between honest and fake runs of the protocol. We are going to proceed in a sequence of games moving from the real world with a valid execution of the protocol, to an idealistic version where both the ciphertext and the key are random. Let $\mathcal{A}$ be a probabilistic polynomial time adversary against the IND-CPA of our scheme and consider the following games where we consider that $\mathcal{A}$ receives the encapsulation at the end of each game.

Game G_1: This game corresponds to an honest run of the protocol. In particular, the simulator has access to all keys / randomness.

Game G_2: Now the simulator picks uniformly at random $\mathbf{x}, \mathbf{y}$ (except for the first coordinate), resulting in a random $\mathbf{s}$. He then proceeds honestly.

An adversary distinguishing between those two games, can distinguish between a well-formed pk and a random one. The public key in the first game correspond to a valid SIS instance, while it is a random one in the second. Hence $\mathsf{Adv}_{\mathcal{A}}^{G_1-G_2} \leq \mathsf{Adv}_{\mathrm{SIS}}(\lambda)$

Game G_3: Now the simulator also picks uniformly at random $\mathbf{e}$, $\mathbf{r}_1$ and $\mathbf{r}_2$ and uses them to generate $\mathbf{s_r}, \mathbf{s_e}$.

An adversary has access to:

$$\begin{pmatrix} \mathbf{s_r} \\ \mathbf{s_e} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{h} \\ \mathbf{0} & \mathbf{I}_n & \mathbf{s} \end{pmatrix} (\mathbf{r}_1, \mathbf{e}, \mathbf{r}_2)^{\top}$$

The syndrome $(\mathbf{s_r}, \mathbf{s_e})$ follows the SIS distribution in game G_2 and the uniform distribution over $\left(\mathbb{F}_q^n\right)^2$ in G_3. If an adversary is able to distinguish games G_2 from G_3, then a simulator can break the underlying problem. Hence $\mathsf{Adv}_{\mathcal{A}}^{G_2-G_3} \leq \mathsf{Adv}_{\mathrm{SIS}}(\lambda)$.

$$\mathsf{Adv}_{\mathsf{KEM}}^{indcpa}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}_{\mathrm{SIS}}(\lambda).$$

Therefore, a PPT adversary $\mathcal{A}$ breaking the protocol with non negligible advantage can be used to solve the SIS problem. □

C. Lattice attacks

Due to space restrictions, we only mention the attacks considered to set the parameters. We refer the reader to [3] and its citations for a panorama of such attacks.

The most trivial attack would be an exhaustive search one the secret key $\hat{\mathbf{x}}, \hat{\mathbf{y}}$, where each part is a trinary n-dimensional vector with roughly the same number of coordinates equal to -1, 0, and 1. Finding one part of secret key therefore requires $\binom{n}{n/3}/n$ operations, where the division by n accounts for the ring structure. Such attacks are way unpractical.

Lattice reduction is probably the most famous type of attacks among lattice attacks. The best lattice reduction algorithm is BKZ 2.0 [6]. Its running time is exponential in the lattice dimension n and strongly depends on the root Hermite factor (RHF) δ. A conservative estimate for the logarithm of the running time is given in [12] by: $t_{\mathrm{BKZ}}(\delta) = 1.8/\log(\delta) - 110$. Such attacks have been considered out of range for RHF below 1.005. They tend to be superseded by sieving and/or enumeration approaches.

Hybrid attacks were introduced in [11] and benefit from the best of the two previous approaches. Nevertheless, due to the dimension of the lattices considered, they are also inefficient.

Enumeration algorithms performs better than sieving algorithms for small dimensions, but have a much worse asymptotic complexity than the latter, both in time and space.

Finally, the best (known) sieving algorithm [4] runs in time $(3/2)^{n/2+o(n)} \approx 2^{0.292n}$ and space $(4/3)^{n/2+o(n)} \approx 2^{0.207n}$. We set our parameters to make such attacks out of range.

D. Parameters

In Tab. I, threshold denotes the maximum number of iterations to recover the error, DFR and RHF stands for "decryption failure rate" and "root Hermite factor" respectively, and t_{LS} and s_{LS} denote the time and space required for best known lattice sieving techniques.The corresponding key and transcript sizes are provided in Tab. II.

E. Further works

A first research axis is the failure probability: the DFR reported in Tab. I was verified experimentally. We believe it is much below 2^{-32} and actually closer to $2^{-\lambda}$ for λ bits of security. This belief is supported by other experiments on

TABLE I
PARAMETER SETS FOR OUROBOROS-E.

Instance	n	q	d	threshold	security	DFR	RHF	t_{LS}	s_{LS}
Toy	293	512	1	18	80	$\ll 2^{-32}$	1.0033	85	60
Standard	389	512	1	20	100	$\ll 2^{-32}$	1.0025	113	80
Advanced	491	512	1	23	128	$\ll 2^{-32}$	1.0020	140	99

TABLE II
KEYS AND TRANSCRIPT SIZES IN BYTES FOR OUROBOROS-E.

Instance	pk size		sk size		transcript size	
	full	seed	full	seed	A→B	B→A
Toy	660	340	147	84	340	660
Standard	876	451	195	110	451	876
Advanced	1,105	569	246	139	569	1,105

smaller moduli, plotted in Fig. 2. A more precise analysis will be conducted in a extended version of this work.

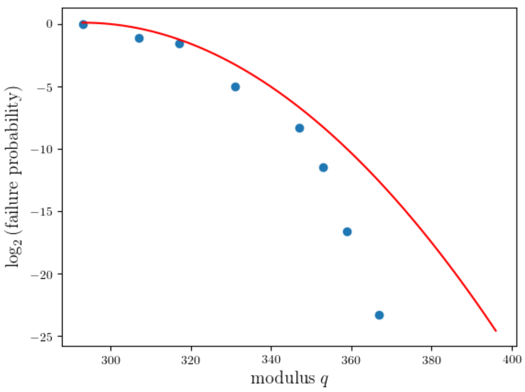

Fig. 2. Logarithm of the observed probability that a decryption failure occurs as function of the modulus q (dots), and a conservative quadratic approximation (curve).

Assuming the binary logarithm of the DFR behaves like a quadratic function of q (which seems reasonable given the shape of the curve induced by the plots), we can derive a conservative extrapolation. The latter gives $\mathrm{DFR}(512) < 2^{-130}$.

Another research axis is the variants of the protocol. It is also possible to instantiate the Ouroboros protocol in a regular, non-ideal way. While this might give more confidence about the underlying problems, this would yield much worse parameters. It is also possible to consider other rings such as $\mathbb{F}_q[x]/(x^n + 1)$, which would yield even better efficiency thanks to fast implementation techniques (FFT).

Additionally, it is also possible to consider the dual version of this protocol, where the public key is no longer an SIS instance of the secret key, but rather an LWE instance of it, *i.e.* a lattice point shifted by some small secret noise. This version was not detailed here due to space restrictions.

CONCLUSION

In this abstract, we adapted the Ouroboros framework [7] to build key exchange protocols to the euclidean metric. We show that it is possible to modify the lattice Bit Flipping algorithm from [10] to handle noisy syndromes. The resulting protocol is competitive with existing KEMs, and features smaller alphabets.

Finally, our non-optimized C++ implementation takes from 2 to 7ms depending on the parameter sets, on an Intel® Core™ i7-6920HQ CPU @ 2.90GHz (turboboost disabled).

Acknowledgement. The third author is supported by the Norwegian Research Council (Grant No. 247742/070). The fourth author is supported by the Swedish Research Council (Grant No. 2015-04528).

REFERENCES

[1] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor, “Efficient encryption from random quasi-cyclic codes,” *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3927–3943, 2018.

[2] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, “Post-quantum key exchange - A new hope,” in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, 2016, pp. 327–343.

[3] S. Bai, T. Laarhoven, and D. Stehlé, “Tuple lattice sieving,” *LMS Journal of Computation and Mathematics*, vol. 19, no. A, pp. 146–162, 2016.

[4] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, “New directions in nearest neighbor searching with applications to lattice sieving,” in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2016, pp. 10–24.

[5] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, and D. Stehlé, “CRYSTALS–Kyber: a CCA-secure module-lattice-based KEM,” *IACR Cryptology ePrint Archive*, vol. 2017, p. 634, 2017.

[6] Y. Chen and P. Q. Nguyen, “BKZ 2.0: Better lattice security estimates,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2011, pp. 1–20.

[7] J.-C. Deneuville, P. Gaborit, and G. Zémor, “Ouroboros: a simple, secure and efficient key exchange protocol based on coding theory,” in *International Workshop on Post-Quantum Cryptography*. Springer, 2017, pp. 18–34.

[8] P. Gaborit, G. Murat, O. Ruatta, and G. Zémor, “Low rank parity check codes and their application to cryptography,” in *Proc. WCC*, 2013, pp. 168–180.

[9] Q. Guo and T. Johansson, “An iterative trapdoor in lattice-based cryptography,” Full version of [10], in preparation.

[10] ——, “A p-ary MDPC scheme,” in *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1356–1360.

[11] N. Howgrave-Graham, “A hybrid lattice-reduction and meet-in-the-middle attack against NTRU,” *Advances in Cryptology-CRYPTO 2007*, pp. 150–169, 2007.

[12] R. Lindner and C. Peikert, “Better key sizes (and attacks) for LWE-based encryption.” in *CT-RSA*, vol. 6558. Springer, 2011, pp. 319–339.

[13] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 1–23.

[14] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. Barreto, “MDPC-McEliece: New McEliece variants from moderate density parity-check codes,” in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2069–2073.

[15] C. Peikert, “A decade of lattice cryptography,” *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 4, pp. 283–424, 2016.

[16] D. Stehlé and R. Steinfeld, “Making NTRU as secure as worst-case problems over ideal lattices.” in *Eurocrypt*, vol. 6632. Springer, 2011, pp. 27–47.