



HAL
open science

Attack on the Edon-K Key Encapsulation Mechanism

Matthieu Lequesne, Jean-Pierre Tillich

► **To cite this version:**

Matthieu Lequesne, Jean-Pierre Tillich. Attack on the Edon-K Key Encapsulation Mechanism. ISIT 2018 - IEEE International Symposium on Information Theory, Jun 2018, Vail, United States. pp.981-985, 10.1109/ISIT.2018.8437498 . hal-01949569

HAL Id: hal-01949569

<https://inria.hal.science/hal-01949569>

Submitted on 10 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Attack on the EDON- \mathcal{K} Key Encapsulation Mechanism

Matthieu Lequesne
Sorbonne Université, UPMC Univ Paris 06
Inria, Team SECRET,
2 rue Simone Iff, CS 42112,
75589 Paris Cedex 12, France
Email: matthieu.lequesne@inria.fr

Jean-Pierre Tillich
Inria, Team SECRET,
2 rue Simone Iff, CS 42112,
75589 Paris Cedex 12, France
Email: jean-pierre.tillich@inria.fr

Abstract—The key encapsulation mechanism EDON- \mathcal{K} was proposed in response to the call for post-quantum cryptography standardization issued by the National Institute of Standards and Technologies (NIST). This scheme is inspired by the McEliece scheme but uses another family of codes defined over $\mathbb{F}_{2^{128}}$ instead of \mathbb{F}_2 and is not based on the Hamming metric. It allows significantly shorter public keys than the McEliece scheme.

In this paper, we give a polynomial time algorithm that recovers the encapsulated secret. This attack makes the scheme insecure for the intended use. We obtain this result by observing that recovering the error in the McEliece scheme corresponding to EDON- \mathcal{K} can be viewed as a decoding problem for the rank-metric. We show that the code used in EDON- \mathcal{K} is in fact a super-code of a Low Rank Parity Check (LRPC) code of very small rank (1 or 2). A suitable parity-check matrix for the super-code of such low rank can be easily derived from for the public key. We then use this parity-check matrix in a decoding algorithm that was devised for LRPC codes to recover the error. Finally we explain how we decapsulate the secret once we have found the error.

I. INTRODUCTION

The syndrome decoding problem is a fundamental problem in complexity theory, since the original paper of Berlekamp, McEliece and van Tilborg [BMvT78] proving its NP-completeness for the Hamming distance. The same year, McEliece proposed a public-key cryptosystem based on this problem [McE78] and instantiated it with binary Goppa codes. This scheme was for a long time considered inferior to RSA due to its large key size. However, this situation has changed drastically when it became apparent in [Sho94] that RSA and actually all the other public-key cryptosystems used in practice could be attacked in polynomial time by a quantum computer. There are now small prototypes of such computers that lead to think that they will become a reality in the future and in 2016, the National Institute of Standards and Technology (NIST) announced a call for standardization of cryptosystems that would be safe against an adversary equipped with a quantum computer. Four families of cryptosystems are often mentioned as potential candidates: cryptosystems based on error correcting codes, lattices, hash functions and multivariate quadratic equations [BBD09]. All of these are based on mathematical problems that are expected to remain hard even in the presence of a quantum computer.

The key encapsulation mechanism EDON- \mathcal{K} [GG17] was proposed by Gligoroski and Gjøsteen in response to the call issued by the NIST. This scheme is inspired by the McEliece scheme but uses another family of codes defined over $\mathbb{F}_{2^{128}}$ instead of \mathbb{F}_2 . This choice leads to very short keys for a code-based scheme. The metric used for the decoding is not properly defined and the security relies on an ad-hoc problem named *finite field vector subset ratio problem* supposedly hard on average.

In this paper, we show that the metric used for EDON- \mathcal{K} is in fact equivalent to the well-known rank metric. This metric was first introduced in 1951 as “arithmetic distance” between matrices over a field \mathbb{F}_q [Hua51]. The notion of rank distance and rank codes over matrices was defined in 1978 by Delsarte [Del78]. He introduced a code family, named maximum rank distance (MRD) codes, that attains the analogue of the MDS (maximum distance separable) bound for the rank metric. Gabidulin suggests in [Gab85] to consider a subfamily of such codes that are linear over an extension field \mathbb{F}_{q^m} . It provides a vectorial representation of these codes and allows to represent them in a much more compact way. This is the main reason why the rank metric based McEliece schemes achieve significantly smaller key sizes. Moreover this vectorial representation allows to view the known families of MRD codes as rank metric analogues of Reed-Solomon codes and to obtain an efficient decoding algorithm for them [Gab85]. There are also rank metric analogues for other families of codes. For instance, the Low Rank Parity-Check (LRPC) codes introduced in [GMRZ13] can be considered as analogues of Low Density Parity-Check (LDPC) codes. Just like their binary cousins, they enjoy an efficient decoding algorithm that is based on a low rank parity-check matrix of such a code.

Here, we prove that the code used in EDON- \mathcal{K} is actually a super-code of an LRPC code of rank 2. What is more, this LRPC code is itself a subspace of codimension 1 of another LRPC code of rank 1. It turns out that parity-check matrices of rank 2 for the first super-code and rank 1 for the second one can easily be derived from the public key. In both cases, this allows us to decode the ciphertext without the secret key. This gives a way to recover the encapsulated secret and breaks completely the EDON- \mathcal{K} system.

The paper is organized as follows. First, we recall some basic definitions and properties of rank metric and LRPC codes in Section II. In Section III we present the scheme of EDON- \mathcal{K} . Then we explain the general idea of our attack in section IV. In Section V, we detail how we reconstruct a parity-check matrix of the code and in Section VI how we decode the ciphertext. In Section VII, we explain how we derive the encapsulated secret from the error. Finally in Section VIII we discuss the cost of this attack and its consequences.

II. RANK METRIC CODES

A. Notation

In the following document, q denotes a power of a prime number. In the case of EDON- \mathcal{K} , we will have $q = 2$. \mathbb{F}_q denotes the finite field with q elements and, for any positive integer m , \mathbb{F}_{q^m} denotes the finite field with q^m elements. We will sometimes view \mathbb{F}_{q^m} as an m -dimensional vector space over \mathbb{F}_q .

We use bold lowercase and capital letters to denote vectors and matrices respectively.

We denote $\langle x_1, \dots, x_k \rangle_{\mathbb{K}}$ the \mathbb{K} -vector space generated by the elements $\{x_1, \dots, x_k\}$.

B. Definitions

Definition 1 (Rank metric over $\mathbb{F}_{q^m}^n$). Let $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_{q^m}^n$ and $(\beta_1, \dots, \beta_m)$ be a basis of \mathbb{F}_{q^m} viewed as an m -dimensional vector space over \mathbb{F}_q . Each coordinate $x_j \in \mathbb{F}_{q^m}$ is associated to a vector of \mathbb{F}_q^m in this basis: $x_j = \sum_{i=1}^m m_{i,j} \beta_i$. The $m \times n$ matrix associated to \mathbf{x} is given by $\mathbf{M}(\mathbf{x}) := (m_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$.

The rank weight $\text{wt}(\mathbf{x})$ of \mathbf{x} is defined as :

$$\text{wt}(\mathbf{x}) := \text{Rank } \mathbf{M}(\mathbf{x}).$$

The associated distance $d(\mathbf{x}, \mathbf{y})$ between elements \mathbf{x} and \mathbf{y} of $\mathbb{F}_{q^m}^n$ is defined by $d(\mathbf{x}, \mathbf{y}) := \text{wt}(\mathbf{x} - \mathbf{y})$.

Definition 2 (Support of a word). Let $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{F}_{q^m}^n$. The support of \mathbf{x} , denoted $\text{Supp}(\mathbf{x})$, is the \mathbb{F}_q -subspace of $\mathbb{F}_{q^m}^n$ generated by the coordinates of \mathbf{x} :

$$\text{Supp}(\mathbf{x}) := \langle x_1, \dots, x_n \rangle_{\mathbb{F}_q}.$$

We have $\dim(\text{Supp}(\mathbf{x})) = \text{wt}(\mathbf{x})$.

Definition 3 (\mathbb{F}_{q^m} -linear code). An \mathbb{F}_{q^m} -linear code \mathcal{C} of dimension k and length n is a subspace of dimension k of $\mathbb{F}_{q^m}^n$. \mathcal{C} can be represented in two equivalent ways: by a generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$ such that $\mathcal{C} = \{\mathbf{x}\mathbf{G} \mid \mathbf{x} \in \mathbb{F}_{q^m}^k\}$ and by a parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ such that $\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_{q^m}^n \mid \mathbf{H}\mathbf{x}^\top = \mathbf{0}_{n-k}\}$.

The decoding problem in the rank metric can be described as follows.

Problem 1 (Decoding problem for the rank metric). Let \mathcal{C} be an \mathbb{F}_{q^m} -linear code of dimension k and length n . Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$ and $\mathbf{e} \in \mathbb{F}_{q^m}^n$ is of rank weight $\leq r$ find \mathbf{c} and \mathbf{e} .

C. LRPC codes

Definition 4 (LRPC code). A Low Rank Parity Check (LRPC) code of rank d , length n and dimension k over \mathbb{F}_{q^m} is a code that admits a parity-check matrix $\mathbf{H} = (h_{i,j}) \in \mathbb{F}_{q^m}^{(n-k) \times n}$ such that the vector space of $\mathbb{F}_{q^m}^n$ generated by its coefficients $h_{i,j}$ has dimension at most d .

LRPC codes can be viewed as analogues of LDPC codes for the rank metric. In particular, they enjoy an efficient decoding algorithm based on their low rank parity-check matrix. Roughly speaking, Algorithm 1 of [GMRZ13] decodes up to d errors when $rd \leq n - k$ in polynomial time (see [GMRZ13, Theorem 1]). It uses in a crucial way the notion of the linear span of a product of subspaces of \mathbb{F}_{q^m} .

Definition 5. Let U and V be two \mathbb{F}_q subspaces of \mathbb{F}_{q^m} . We denote by $U \cdot V$ the linear span of the product of U and V :

$$U \cdot V := \langle uv : u \in U, v \in V \rangle_{\mathbb{F}_q}.$$

III. THE EDON- \mathcal{K} KEM

EDON- \mathcal{K} [GG17] is a key encapsulation mechanism proposed by Gligoroski and Gjøsteen for the NIST post-quantum cryptography call. Here we describe the key generation, encapsulation and decapsulation, omitting some details that are not relevant for the attack. We refer to [GG17] for the full description.

A. Parameters and notations

The parameters for EDON- \mathcal{K} are given in Table I. In this paper we often refer to the parameters of edonk128ref, the reference version proposed for 128 security-bits.

TABLE I
PARAMETERS PROPOSED FOR EDON- \mathcal{K}

Name	m	N	K	R	ν	L
edonk128ref	128	144	16	40	8	6
edonk128K16N80nu8L6	128	80	16	40	8	6
edonk128K08N72nu8L8	128	72	8	40	8	8
edonk128K32N96nu4L4	128	96	32	40	4	4
edonk128K16N80nu4L6	128	80	16	40	4	6
edonk192ref	192	112	16	40	8	8
edonk192K48N144nu4L4	192	144	48	40	4	4
edonk192K32N128nu4L6	192	128	32	40	4	6
edonk192K16N112nu4L8	192	112	16	40	4	8

The scheme makes use of a hash function $\mathcal{H}(\cdot)$ corresponding to standard SHA2 functions (SHA-256 or SHA-384 depending on the parameters). We will denote $\mathcal{H}^i(\cdot) := \underbrace{\mathcal{H}(\dots \mathcal{H}(\cdot))}_{i \text{ times}}$

Given a binary matrix $\mathbf{P} = (p_{i,j})$ and two non-zero elements $a \neq b$ of \mathbb{F}_{2^m} , $\mathbf{P}_{a,b} = (\tilde{p}_{i,j})$ denotes the matrix of the same size with coefficients in \mathbb{F}_{2^m} where $\tilde{p}_{i,j} = a$ if $p_{i,j} = 0$ and $\tilde{p}_{i,j} = b$ if $p_{i,j} = 1$.

In particular, if \mathbf{P} is orthogonal:

$$\mathbf{P}_{c,d}^\top = (\mathbf{P}_{a,b})^{-1} \quad (1)$$

where $c := \frac{a}{a^2+b^2}$ and $d := \frac{b}{a^2+b^2}$.

For two vectors (or matrices) \mathbf{x} and \mathbf{y} , we will denote $\mathbf{x}||\mathbf{y}$ their concatenation.

B. Key generation

Given the security level and the appropriate parameters.

- $a, b \xleftarrow{\$} \mathbb{F}_{2^m}$ non-zero elements such that $a \neq b$.
- $\mathbf{P} \xleftarrow{\$} \mathbb{F}_2^{N \times N}$ an orthogonal matrix.
- $\mathbf{H} \xleftarrow{\$} \mathbb{F}_2^{R \times N}$ such that $\mathbf{H} = [\mathbf{H}_T || \mathbf{H}_B]^\top$ where \mathbf{H}_B is an $R \times R$ orthogonal matrix and \mathbf{H}_T has columns of even Hamming weight.
- $c := \frac{a}{a^2+b^2}$, $d := \frac{b}{a^2+b^2}$.
- $\tilde{\mathbf{g}} \xleftarrow{\$} \mathbb{F}_{2^m}^\nu$.
- $\mathcal{V}_g := \text{Support}(\tilde{\mathbf{g}})$.
- $\mathbf{G} \xleftarrow{\$} \mathcal{V}_g^{K \times N}$ such that

$$\mathbf{G}\mathbf{H}^\top = \mathbf{0}_{K \times R}. \quad (2)$$

- $\mathbf{G}_{\text{pub}} := \mathbf{G}\mathbf{P}_{c,d}^\top$.
- Return (PublicKey := \mathbf{G}_{pub} , SecretKey := $(a, b, \mathbf{P}, \mathbf{H})$).

C. Encapsulation

Given the PublicKey and the public parameters.

- $\mathbf{m} \xleftarrow{\$} \mathbb{F}_{2^m}^K$.
- $\tilde{\mathbf{e}} \in \mathbb{F}_{2^m}^L$ generated as follows:
 - $(\tilde{e}_0, \tilde{e}_1) \xleftarrow{\$} \mathbb{F}_{2^m}^2$;
 - for $1 \leq i \leq \frac{L}{2} - 1$, $(\tilde{e}_{2i}, \tilde{e}_{2i+1}) = \mathcal{H}(\tilde{e}_{2i-2} || \tilde{e}_{2i-1})$.
- $\mathcal{V}_e := \text{Support}(\tilde{\mathbf{e}})$.
- $\mathbf{e} \xleftarrow{\$} \mathcal{V}_e^N$.
- $\mathbf{c} := \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e}$.
- $(s_0, s_1) := \mathcal{H}(\tilde{e}_{L-2} || \tilde{e}_{L-1})$.
- SharedSecret := $\mathcal{H}(s_0 || s_1 || \mathcal{H}(\mathbf{c}))$.
- $h := \mathcal{H}(s_1 || s_0 || \mathcal{H}(\mathbf{c}))$.
- Ciphertext := (\mathbf{c}, h) .
- Return (Ciphertext, SharedSecret).

D. Decapsulation

Given Ciphertext, SecretKey and the public parameters.

- Recover \mathbf{e} by decoding the \mathbf{c} using the private matrix $\mathbf{H}' := \mathbf{H}\mathbf{P}_{a,b}^\top$.
 - Deduce \mathcal{V}_e the vector space spanned by the coefficients of the vector \mathbf{e} .
 - For all $(\lambda, \nu) \in \mathcal{V}_e \times \mathcal{V}_e$, for $1 \leq i \leq \frac{L}{2} - 1$:
 - $(s'_0, s'_1) := \mathcal{H}^i(\lambda || \mu || \mathcal{H}(\mathbf{c}))$;
 - if $\mathcal{H}(s'_1 || s'_0 || \mathbf{c}) = h$:
- Return SharedSecret := $\mathcal{H}(s'_0 || s'_1 || \mathcal{H}(\mathbf{c}))$.

IV. OUTLINE OF THE ATTACK ON EDON-K

Our attack is based on three observations

- The ciphertext is a vector \mathbf{c} such that

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e}. \quad (3)$$

This error \mathbf{e} is of low rank, since its rank is at most L .

- This code \mathcal{C}_{pub} generated by \mathbf{G}_{pub} is a subcode of an LRPC code, namely the code \mathcal{C}' with parity-check matrix $\mathbf{H}' := \mathbf{H}\mathbf{P}_{a,b}^\top$. This code is indeed an LRPC code of rank 2 since all the entries of \mathbf{H}' belong to $\langle a, b \rangle_{\mathbb{F}_2}$. We have

$$\mathcal{C}_{\text{pub}} \subset \mathcal{C}' \quad (4)$$

since

$$\begin{aligned} \mathbf{G}_{\text{pub}}\mathbf{H}'^\top &= \mathbf{G}\mathbf{P}_{c,d}^\top(\mathbf{H}\mathbf{P}_{a,b}^\top)^\top \\ &= \mathbf{G}\mathbf{P}_{c,d}^\top\mathbf{P}_{a,b}\mathbf{H}^\top \\ &= \mathbf{G}\mathbf{H}^\top \text{ (from (1))} \\ &= \mathbf{0}_{K \times R} \text{ (from (2))}. \end{aligned}$$

This equation also appears as Corollary 1 of [GG17, p.19]. We have given its proof here for the convenience of the reader. Let $K' = N - R$ be the dimension of \mathcal{C}' .

- If we recover a parity-check matrix of rank 2 for \mathcal{C}' we will be able to recover $\mathbf{m}\mathbf{G}_{\text{pub}}$ and \mathbf{e} from \mathbf{c} . Indeed, $\mathbf{m}\mathbf{G}_{\text{pub}} \in \mathcal{C}'$ and we can decode \mathcal{C}' using a variation of Algorithm 1 of [GMRZ13] and the knowledge of the parity-check matrix, provided $\text{wt}(\mathbf{e}) \leq L < (N - K')/2 = R/2$ is verified, which is the case for the parameters of EDON-K.

Hence we will proceed in three steps:

- 1) constructing and solving a linear system of equations to find a parity-check matrix for the code \mathcal{C}' (detailed in Section V);
- 2) decoding the ciphertext using a slight variation of Algorithm 1 of [GMRZ13] (see Section VI);
- 3) recovering the secret from the error vector (explained in Section VII).

V. RECONSTRUCTING THE PARITY-CHECK MATRIX

A. Compressed public key

In order to reduce the public key size, the designers of EDON-K chose to represent the public key in a compressed form. They took advantage of the fact that all the coefficients of \mathbf{G}_{pub} live in the vector space $\mathcal{V}_{g,c,d} := \langle c\tilde{g}_1, \dots, c\tilde{g}_\nu, d\tilde{g}_1, \dots, d\tilde{g}_\nu \rangle_{\mathbb{F}_2}$ of dimension 2ν . Hence, the compressed public key consists in two parts: first the basis $\tilde{\mathbf{g}}_{c,d} := (c\tilde{g}_1, \dots, c\tilde{g}_\nu, d\tilde{g}_1, \dots, d\tilde{g}_\nu) \in \mathbb{F}_2^{2\nu}$ of the vector-space $\mathcal{V}_{g,c,d}$, then the entries of the matrix \mathbf{G}_{pub} such that each entry is represented by its coefficients in the basis $\tilde{\mathbf{g}}_{c,d}$. For example, if an entry x of \mathbf{G}_{pub} is equal to $c \sum_{i=1}^\nu \gamma_i \tilde{g}_i + d \sum_{i=1}^\nu \delta_i \tilde{g}_i$ with $\gamma_i, \delta_i \in \mathbb{F}_2$, x will be represented by $(\gamma_1, \dots, \gamma_\nu, \delta_1, \dots, \delta_\nu) \in \mathbb{F}_2^{2\nu}$. There is another subtlety in the compression that we will not mention here.

B. Finding a basis

The attacker does not have access to the value of a and b but can deduce the value of $ab^{-1} = cd^{-1} = (c\tilde{g}_1)(d\tilde{g}_1)^{-1}$ from $\tilde{g}_{c,d}$ as mentioned in paragraph 7.2.2 of the documentation of EDON- \mathcal{K} [GG17].

Let us bring in

$$\alpha := ab^{-1}.$$

We notice that $\mathbf{H}'' := b^{-1}\mathbf{H}'$ is also a parity-check matrix of the LRPC code \mathcal{C}' . This matrix has all its coefficients in $\langle 1, \alpha \rangle_{\mathbb{F}_2}$. We use this information to reconstruct such a parity-check matrix of the code \mathcal{C}' by solving a linear system, similarly to what is done in [GRS16, Section IV B]. This system is derived from the following facts:

- (i) $\mathbf{G}_{\text{pub}} \mathbf{H}''^T = \mathbf{0}_{K \times R}$;
- (ii) the entries of \mathbf{H}'' belong to $\langle 1, \alpha \rangle_{\mathbb{F}_2}$.

In other words, the possible rows $\mathbf{x} = (x_1, \dots, x_N)$ of \mathbf{H}'' are solutions of the following system

$$\begin{cases} \mathbf{G}_{\text{pub}} \mathbf{x}^T &= \mathbf{0}_K \\ x_i &\in \langle 1, \alpha \rangle_{\mathbb{F}_2} \text{ for all } i \in \{1, \dots, N\}. \end{cases} \quad (5)$$

This system is linear over \mathbb{F}_2 and the solution set is an \mathbb{F}_2 -linear subspace which can be computed in $O(KmN^2)$ operations. A basis of this subspace can then be used as rows for \mathbf{H}'' . The details on how to transform this system into a linear system over \mathbb{F}_2 can be found in the long version of the article [LT18].

VI. DECODING STEP

The previous step recovers an $R \times N$ matrix $\mathbf{H}^{(3)}$ whose entries all belong to $\langle 1, \alpha \rangle_{\mathbb{F}_2}$. The matrices $\mathbf{H}^{(3)}$ and \mathbf{H}'' share the property that their rows form a basis of solutions of (5). Therefore, there exists an $R \times R$ binary invertible matrix \mathbf{Q} such that

$$\mathbf{H}^{(3)} = \mathbf{Q}\mathbf{H}'' . \quad (6)$$

We use $\mathbf{H}^{(3)}$ to decode and recover \mathbf{e} from the ciphertext \mathbf{c} . The vectors are linked by the equation

$$\mathbf{c} = \mathbf{m}\mathbf{G}_{\text{pub}} + \mathbf{e} . \quad (7)$$

We use here a slight variation of Algorithm 1 of [GMRZ13] to decode. Algorithm 1 would consist in performing the following steps:

- 1) Compute $\mathbf{s}^T := \mathbf{H}^{(3)}\mathbf{c}^T$ and then $V := \text{Support}(\mathbf{s})$. Here we typically have $V = \text{Support}(\mathbf{e}) \cdot \langle 1, \alpha \rangle_{\mathbb{F}_q}$ when $\mathbf{H}^{(3)}$ is a random matrix.
- 2) Compute $V' := V \cap \alpha^{-1}V$. This step typically recovers $\text{Support}(\mathbf{e})$ when $V = \text{Support}(\mathbf{e}) \cdot \langle 1, \alpha \rangle_{\mathbb{F}_q}$.
- 3) Once we have $\text{Support}(\mathbf{e})$ we recover $\mathbf{e} = (e_1, \dots, e_N)$ by solving the linear equation $\mathbf{H}^{(3)}\mathbf{e}^T = \mathbf{s}^T$ with the additional constraints $e_i \in \text{Support}(\mathbf{e})$ for $i \in \{1, \dots, N\}$.

In our case, due to the special structure of \mathbf{H} which contains only a 's and b 's V is not equal to $\text{Support}(\mathbf{e}) \cdot \langle 1, \alpha \rangle_{\mathbb{F}_q}$. This is due to the following result.

Proposition 1. We have for every $\mathbf{e} \in \mathbb{F}_2^N$:

$$\text{Support}(\mathbf{H}^{(3)}\mathbf{e}^T) \subset (1 + \alpha)\text{Support}(\mathbf{e}) + \left\langle \sum_{i=1}^N e_i \right\rangle_{\mathbb{F}_2} .$$

Proof. From (6), we deduce that

$$\text{Support}(\mathbf{H}^{(3)}\mathbf{e}^T) = \text{Support}(\mathbf{Q}\mathbf{H}''\mathbf{e}^T) = \text{Support}(\mathbf{H}''\mathbf{e}^T).$$

Let $\mathbf{s}^T := \mathbf{H}''\mathbf{e}^T$. Denote the i -entry of \mathbf{s} by s_i and the entry of \mathbf{H}'' in row i and column j by h''_{ij} . We have:

$$\begin{aligned} s_i &= \sum_{j=1}^N h''_{ij} e_j \\ &= \sum_{j \text{ s.t. } h''_{ij}=1} e_j + \sum_{j \text{ s.t. } h''_{ij}=\alpha} \alpha e_j \\ &= \sum_{j=1}^N e_j + (1 + \alpha) \sum_{j \text{ s.t. } h''_{ij}=\alpha} e_j . \end{aligned}$$

This implies the proposition. \square

This proposition directly gives a subspace of dimension $L + 1$ that contains $\text{Support}(\mathbf{e})$ since we deduce from it that

$$\text{Support}(\mathbf{e}) \subset (1 + \alpha)^{-1}\text{Support}(\mathbf{H}''\mathbf{e}) . \quad (8)$$

A slight modification of Algorithm 1 of [GMRZ13] yields therefore e:

- 1) compute the syndrome $\mathbf{s}^T := \mathbf{H}^{(3)}\mathbf{c}^T$ and then $V := (1 + \alpha)^{-1}\text{Support}(\mathbf{s})$;
- 2) The space V contains $\text{Support}(\mathbf{e})$, so we can recover $\mathbf{e} = (e_1, \dots, e_N)$ by solving the linear equation $\mathbf{H}^{(3)}\mathbf{e}^T = \mathbf{s}^T$ with the additional constraints $e_i \in V$ for $i \in \{1, \dots, N\}$.

Note that we can also skip step 2 and directly look for s_0 and s_1 in the space V of dimension $L + 1$ instead of decoding exactly the value of e . In fact, this is what is specified in the decapsulation of EDON- \mathcal{K} .

VII. RECOVERING THE SHARED SECRET

Once we have recovered the error vector $\mathbf{e} \in \mathbb{F}_q^N$, we need to recover s_0 and s_1 to obtain the value of SharedSecret. We know that the elements of \mathbf{e} were picked randomly in $\mathcal{V}_e = \text{Support}(\mathbf{e})$. We proceed just like in the decapsulation algorithm.

We generate $\text{Support}(\mathbf{e})$ which is equal to \mathcal{V}_e with high probability. More exactly, the probability that $\text{Support}(\mathbf{e})$ is of dimension $< L$ is $\left(\frac{L-1}{L}\right)^N$. For the parameters of edonk128ref this probability is 2^{-37} . In such a case, the attack might fail, but the decapsulation would fail too.

Then, among the 2^L elements of \mathcal{V}_e , we need to identify a couple of consecutive elements of $\tilde{\mathbf{e}}$ to deduce the secret. For all pairs of candidates $(\lambda, \mu) \in \mathcal{V}_e \times \mathcal{V}_e$, for $1 \leq i \leq \frac{L}{2} - 1$ we compute $(s'_0, s'_1) := \mathcal{H}^i(\lambda || \mu || \mathcal{H}(\mathbf{c}))$. If $\mathcal{H}(s'_1 || s'_0 || \mathbf{c}) = h$ then we have $(s'_0, s'_1) = (s_0, s_1)$. Finally we recover the secret SharedSecret = $\mathcal{H}(s_0 || s_1 || \mathbf{c})$. In total this operation requires $\mathcal{O}(L2^{2L})$ operations, just like the decapsulation. This is the reason why the value of L needs to remain small, otherwise the decapsulation is not possible.

VIII. CONCLUDING REMARKS

A. Cost of the attack

Let us analyze the cost of the three steps of the attack mentioned in Section IV.

Step 1 and 2 are polynomial in terms of the parameters of the code. Step 1 only uses linear algebra operations and has a complexity at most $\mathcal{O}(KmN^2)$. The complexity of step 2 is given by Theorem 1 of [GMRZ13] (using $n = N, k = N - R, r = L$ and $d = 2$), hence is equal to $L^2(16m + N^2)$. The complexity of step 3 is $\mathcal{O}(L2^{2L})$. This is not polynomial in L but L is a very small parameter ($4 \leq L \leq 8$ in the proposal). Moreover this third step is the same as the decapsulation algorithm, so L needs to stay small, otherwise the decapsulation would become too costly or even impossible. So L can be considered as a constant ≤ 10 to allow a reasonable decapsulation. Hence the most costly operation appears to be step 1.

B. Without compression of the public key

Our attack takes advantage of the compressed form of the public key that allows a direct access to the value $\alpha = ab^{-1}$. One could think that this is the origin of the attack, and decide to express the public key in its uncompressed form to fix the attack. As a consequence, the public key would be of size $K \times N \times m$ bits instead of $K \times N \times \nu$ bits in the compressed form. In practice the public key for edonk128ref would be 16 times longer (around 288 kbits). This inflation of the key size could be avoided by sending out a random basis of $\mathcal{V}_{g,c,d}$.

However, this is not enough. There is an even more direct way to proceed, without the value of α . Instead of looking for a matrix $\mathbf{H}^{(3)}$ with entries lying in $\langle 1, \alpha \rangle_{\mathbb{F}_2}$, we can use the following result.

Proposition 2. *There exists a full rank $(R - 1) \times N$ binary matrix $\mathbf{H}^{(4)}$ that satisfies*

$$\mathbf{G}_{\text{pub}} \mathbf{H}^{(4)\top} = \mathbf{0}_{K \times (R-1)}.$$

Proof. Let \mathbf{T} be a binary full-rank matrix $(R - 1) \times R$ matrix that has rows of even Hamming weight. For instance we can choose

$$\mathbf{T} = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \end{pmatrix}.$$

We observe now that \mathbf{TH} has all its entries in $\{0, a + b\}$. This follows directly from the fact that if we sum an even number of elements in $\{a, b\}$ we either get 0 (if the number of a 's is even, and therefore also the number of b 's) or $a + b$ (if the number of a 's is odd). From this, it follows immediately that

$$\mathbf{H}^{(4)} := \frac{1}{a + b} \mathbf{TH}$$

satisfies the property. First, it is clear that this is a binary matrix and we also have

$$\mathbf{G}_{\text{pub}} \mathbf{H}^{(4)\top} = \frac{1}{a + b} \mathbf{G}_{\text{pub}} \mathbf{H}^\top \mathbf{T}^\top = \mathbf{0}_{K \times (R-1)}.$$

□

Obtaining such a matrix $\mathbf{H}^{(4)}$ is straightforward. We just have to use the algorithm given in Section V to recover a basis of dimension $R - 1$ of binary vectors \mathbf{x} satisfying

$$\mathbf{G}_{\text{pub}} \mathbf{x}^\top = \mathbf{0}_K.$$

We then use this matrix $\mathbf{H}^{(4)}$ to compute the syndrome $\mathbf{s} = \mathbf{H}^{(4)} \mathbf{c}^\top$. Since $\mathbf{H}^{(4)} \mathbf{c}^\top = \mathbf{H}^{(4)} \mathbf{e}^\top$ we directly obtain with very high probability that

$$\text{Support}(\mathbf{e}) = \text{Support}(\mathbf{H}^{(4)} \mathbf{c}^\top).$$

This reveals the support of the error and from there we can go directly to the last step of the attack to reconstruct the shared secret.

C. Security of the scheme

Considering the attack that we described, there is a way to recover the secret of the edonk128ref scheme from a public key without the private key in polynomial time. In practice, the attack implemented with Sage on a personal computer recovers the secret in less than a minute, so the scheme is far from achieving the 128-bits security claimed in [GG17]. Hence this scheme is insecure for the intended use. Moreover, the cost of this attack is polynomial in terms of the parameters, so there is no proper way to increase the parameters to achieve the intended security level while keeping a reasonably small key size.

REFERENCES

- [BBD09] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors. *Post-Quantum Cryptography*. Springer-Verlag, 2009.
- [BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, 24(3):384–386, May 1978.
- [Del78] Philippe Delsarte. Bilinear forms over a finite field, with applications to coding theory. *J. Comb. Theory, Ser. A*, 25(3):226–241, 1978.
- [Gab85] Ernest Mukhamedovich Gabidulin. Theory of codes with maximum rank distance. *Problemy Peredachi Informatsii*, 21(1):3–16, 1985.
- [GG17] Danilo Gligoroski and Kristian Gjøsteen. Edon-k. first round submission to the NIST post-quantum cryptography call, November 2017.
- [GMRZ13] Philippe Gaborit, Gaëtan 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 Trans. Information Theory*, 62(2):1006–1019, 2016.
- [Hua51] Loo-Keng Hua. A theorem on matrices over a sfield and its applications. *J. Chinese Math. Soc.*, 1(2):109–163, 1951.
- [LT18] Matthieu Lequesne and Jean-Pierre Tillich. Attack on the Edon-K key encapsulation mechanism. Preprint, 2018. arXiv:1802.06157.
- [McE78] Robert J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44.
- [Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In S. Goldwasser, editor, *FOCS*, pages 124–134, 1994.