A Key Recovery Attack on TRIKE Based on Decoding Failures

Alessandro Annechini[0009−0009−4452−802X], Alessandro Barenghi[0000−0003−0840−6358], and Gerardo Pelosi[0000−0002−3812−5429]

Department of Electronics, Information and Bioengineering - DEIB, Politecnico di Milano, Milano, Italy
{alessandro.annechini, alessandro.barenghi, gerardo.pelosi}@polimi.it

Abstract. TRIKE is a code-based Key Encapsulation Mechanism submitted to the Next-Generation Commercial Cryptographic algorithms (NGCC) standardization program currently run in People's Republic of China. TRIKE, similarly to other Quasi Cyclic Moderate Density Parity Check (QC-MDPC) code-based KEMs such as BIKE, is built with a decryption procedure which recovers a sparse error vector (i.e., the message) given its syndrome (i.e., the ciphertext) through a secret sparse parity-check matrix, employing an iterative bit-flipping decoder. Differently from BIKE, TRIKE employs a QC-MDPC matrix with a different structure, allowing them to improve the decoding, and hence the decryption, time. In order to guarantee IND-CCA security, the code length is chosen in order to reduce the Decoding Failure Rate (DFR) below $2^{-\lambda}$, where λ is the bit security level; such a DFR is estimated with extrapolation techniques.
In this work, we show that the QC-MDPC codes employed in TRIKE are affected by the so-called *near codewords* phenomenon. We provide a lower bound on the DFR given by such near codewords, showing that the DFR of TRIKE is significantly higher than the required value for all parameter sets. This alone constitutes a breach of IND-CCA security requirements. We propose a key recovery attack based on near codeword-induced decoding failures. For NGCC security level 1, $\lambda = 128$, the attack requires less than 2^{80} (random) ciphertexts and lowers the hardness of the static key recovery step down to $2^{65.52}$ bit operations. We validate our attack strategy and complexity estimation on a toy parameter set with extrapolated DFR below 2^{-80}, recovering the secret key with $2^{29.50}$ decryption queries.

Keywords: Code-based cryptography · Decoding Failure Rate · TRIKE

1 Introduction

The impending realization of large-scale quantum computers threatens the security of classical public-key cryptography. Among the primary paradigms evaluated for standardizing quantum-resistant primitives, code-based cryptography stands out for its extensive theoretical foundation and robust historical resistance

to both classical and quantum cryptanalysis. Code-based Key Encapsulation Mechanisms (KEMs) follow one of two templates, one introduced by Niederreiter in 1986 [20] and one by Alekhnovich in 2003 [1]. The usage of Quasi-Cyclic (QC) codes to reduce public key sizes made the instantiation of both templates practical, and led to the design of the Alekhnovich-based cryptosystem HQC [18] and the Niederreiter-based cryptosystem BIKE [5]. BIKE employs Quasi-Cyclic Moderate Density Parity-Check (QC-MDPC) codes, that guarantee compact public keys and ciphertext, and good performance figures. The most computationally demanding step in BIKE is the decapsulation procedure, where the runtime of the underlying decoder is proportional to the Hamming weight of the secret parity-check matrix columns.

The TRIKE cryptosystem is a post-quantum code-based KEM submitted to the Next-Generation Commercial Cryptographic algorithms (NGCC) program [26]. TRIKE follows the Niederreiter template and is built on a triple quasi-cyclic framework: unlike BIKE, TRIKE combines *three* sparse secret polynomials into a single public ratio. This design choice allows the usage of sparser QC-MDPC codes with respect to BIKE, improving the speed of the decoding procedure, at the cost of a larger ciphertext, while keeping comparable bandwidth requirements overall.

A defining characteristic of QC-MDPC-based cryptosystems is their reliance on iterative, probabilistic decoders, such as the parallel bit-flipping decoder, that exhibit a non-null Decoding Failure Rate (DFR). Since the convergence of such a decoder is sensitive to the structure of the secret parity-check matrix and the specific error vector introduced during encapsulation, decryption failures can provide information to an active attacker [14]. When an IND-CCA secure KEM such as TRIKE is deployed in a static-key scenario, this information leakage becomes a critical vulnerability. For this reason, IND-CCA secure KEMs are required to have a negligible DFR, i.e., below $2^{-\lambda}$ where λ is the bit-security level. To ensure a sufficiently low DFR, in TRIKE, the decoding failure rate is estimated via a Finite-Length Scaling (FLS)-based interpolation, based on the asymptotic behaviour of (non QC) Low Density Parity-Check (LDPC) codes [2].

1.1 Contributions

In this work, we analyse the DFR of TRIKE, showing that that the QC-MDPC codes employed in it are affected by the so called *near codewords* [25], a family of error vectors that lead to decoding failures with a higher probability than uniformly randomly sampled error vectors of the same weight, due to their correlation with the secret, quasi-cyclic parity-check matrix. Modelling the effect of near codewords allows us to derive a lower bound on the DFR of TRIKE, which turns out to be much higher than the required bound for all the proposed parameter sets.

The higher decoding failure rate, and the specific structure of vectors leading to decoding failures, enable a reaction-based key recovery attack. We present a variant of the attack first proposed in [14] specialized for near codewords, based on the observed behaviour of the decapsulation procedure over random (i.e.,

not chosen) inputs, and we analyse its success probability as a function of the number of allowed decryption queries. We validate the attack effectiveness on a toy parameter set with a standard KRA complexity of $2^{65.8}$ an extrapolated DFR of 2^{-87} according to the TRIKE criteria, successfully retrieving the secret key with an average of $2^{29.50}$ decryption queries over random inputs followed by a simple ISD-based step. For NGCC security level 1, $\lambda = 128$, the attack requires approximately $2^{66.9}$ decryption queries, which is below the 2^{80} threshold set for NGCC candidates [15], followed by an ISD-based computational workload of $2^{65.52}$ bit operations.

2 Background

Notation. Scalar values are denoted with lowercase Latin letters (e.g., p, v, t), (column) vectors are denoted with lowercase boldface Latin letters (e.g., $\mathbf{v}$), matrices are denoted with uppercase boldface Latin letters (e.g., $\mathbf{M}$). Elements of the polynomial ring $\mathcal{R} = \mathbb{F}_2[x]/\langle x^p - 1\rangle$ are denoted with lowercase typewriter-font letters (e.g. $\mathtt{e}(x)$). We employ greek letters for probability functions (i.e. $\zeta(\ell)$, $\epsilon(\ell)$). Using the isomorphism between the ring of circulant matrices (i.e., $p \times p$ matrices where the rows from second to last row defined as circulant shifts of the first row) and the polynomial ring $\mathcal{R}$, we identify circulant matrices $\mathbf{A} \in \mathbb{F}_2^{p\times p}$ with their corresponding polynomial $\mathtt{a}(x) \in \mathcal{R}$, obtained by setting as coefficients of the polynomials the entries of the first column of $\mathbf{A}$. Likewise, p-sized binary vectors can be interpreted as elements of $\mathcal{R}$, e.g., $\mathbf{s} \in \mathbb{F}_2^p$ becomes $\mathtt{s}(x) \in \mathcal{R}$ and $\mathbf{e} \in \mathbb{F}_2^{3p}$ becomes $(\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)) \in \mathcal{R}^3$. The number of non-zero entries in a binary vector $\mathbf{v}$ (or a polynomial $\mathtt{v}(x)$ in $\mathcal{R}$) is denoted as its (Hamming) weight $\mathtt{wt}(\mathbf{v})$ (resp. $\mathtt{wt}(\mathtt{v}(x))$).

2.1 QC-MDPC Codes

A binary linear code $\mathcal{C}[n, k]$ is a subspace of $\mathbb{F}_2^n$ with 2^k elements (called *codewords*), where n is the code *length*, k is the code *dimension*, the quantity $\frac{k}{n}$ is called *code rate*, and $r = n-k$ is the code *redundancy*. A binary linear code can be described by means of a *generator* matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ s.t. $\mathcal{C} = \{\mathbf{G}^\mathsf{T}\mathbf{u}$ for each $\mathbf{u} \in \mathbb{F}_2^k\}$, or by means of a *parity-check* matrix $\mathbf{H} \in \mathbb{F}_2^{r\times n}$ s.t. $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_2^n | \mathbf{Hc} = \mathbf{0}\}$. Given any vector $\mathbf{e} \in \mathbb{F}_2^n$, the vector $\mathbf{s} = \mathbf{He}$ is known as the *syndrome* of $\mathbf{e}$ through $\mathbf{H}$.

A linear code is said to be Quasi-Cyclic (QC) if its parity-check matrix $\mathbf{H}$ is composed by tiling $p \times p$ circulant matrices over a $r_0 \times n_0$ grid. In the post-quantum KEM BIKE [5], QC codes are chosen to have $r_0 = 1$ and $n_0 = 2$ (i.e., $\mathbf{H} = [\mathtt{h}_0(x), \mathtt{h}_1(x)]$), while in TRIKE [26] QC codes are chosen to have $r_0 = 1$ and $n_0 = 3$ (i.e., $\mathbf{H} = [\mathtt{h}_0(x), \mathtt{h}_1(x), \mathtt{h}_2(x)]$). If the columns of a QC parity-check matrix $\mathbf{H}$ (with $r_0 = 1$) have a fixed weight v, and the rows of $\mathbf{H}$ have a fixed weight $w = n_0 v$, then the code is said to be a QC Low Density Parity-Check (QC-LDPC) code if $v \in \mathcal{O}(\log(n))$, or QC Moderate Density Parity-Check (QC-MDPC) code if $v \in \mathcal{O}(\sqrt{n})$.

Many code-based cryptosystems rely on the hardness of the following two problems [10,9,3].

Definition 1 (Quasi-Cyclic Syndrome Decoding Problem (QC-SDP)). *Given a QC parity-check matrix* $\mathbf{H}$*, and the syndrome* $\mathbf{s}$ *of a sparse error vector* $\mathbf{e}$ *with* $\mathtt{wt}(\mathbf{e}) = t$ *through* $\mathbf{H}$*, find* $\mathbf{e}$*.*

Definition 2 (Quasi-Cyclic Codeword Finding Problem (QC-CFP)). *Given a QC parity-check matrix* $\mathbf{H}$ *and an integer* w*, find a non-zero vector* $\mathbf{c}$ *with* $\mathtt{wt}(\mathbf{c}) \leq w$ *such that* $\mathbf{Hc} = \mathbf{0}$*.*

The current optimal approach for solving instances of such problems, in the regime where t and w are sub-linear in the code length n, relies on Information Set Decoding techniques [21,17,23,12,8,11], that exhibit an exponential complexity in the problem parameters. Moreover, the complexity of ISD algorithms is larger for higher code rates (if the error density $\frac{t}{n}$ is kept constant, and the system is in a regime where the solution is unique), meaning that solving QC-SDP/QC-CFP instances over a code with rate 2/3 is computationally harder than solving instances over a code with rate 1/2 under similar problem parameters.

2.2 TRIKE

TRIKE is a post-quantum Key Encapsulation Mechanism (KEM) based on QC-MDPC codes, submitted to the Next-Generation Commercial Cryptographic algorithms (NGCC) program [26]. The TRIKE cryptosystem is defined by three primitives, KeyGen (Algorithm 1), Encaps (Algorithm 2), and Decaps (Algorithm 3). The different instances of TRIKE are identified by the parameters (p, w, t), representing respectively the size of the circulant blocks, the combined Hamming weight of the secret key polynomials ($\mathtt{wt}(\mathtt{h}_0(x)) = \mathtt{wt}(\mathtt{h}_1(x)) = \mathtt{wt}(\mathtt{h}_2(x)) = v = w/3$), and the total weight of the error vector $\mathtt{wt}(\mathbf{e})$ (where $\mathbf{e} = [\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)]$). The parameter sets of TRIKE are shown in Table 1, and target different bit-security levels λ, with $\lambda \in \{128, 192, 256, 384, 512\}$. Such security levels are denoted as NGCC security level 1/3/5/7/9 respectively. In the pseudocode, $\leftarrow$ denotes assignment, while $\xleftarrow{\$}$ denotes random sampling from a set. The term $\mathcal{R}_{\mathtt{wt}=v}$ indicates the subset of $\mathcal{R}$ composed by elements having Hamming weight equal to v, $\mathcal{R}_{odd}$ indicates the elements with odd weight, $\mathcal{R}_{even}$ indicates the elements with even weight, $\mathcal{R}^3_{\mathtt{wt}=t}$ indicates triplets of elements in $\mathcal{R}$ with total Hamming weight t. The terms $\mathcal{L}$, $\mathcal{K}$, and $\mathtt{Hash}_i$ $(1 \leq i \leq 4)$ indicate abstract, domain-separated hash functions, with their respective domain and co-domain being:

$$\begin{aligned}
&\mathtt{Hash}_1 : \mathbb{F}_2^{2\lambda} \to \mathcal{R}_{even} && \mathtt{Hash}_2 : \mathbb{F}_2^{2\lambda} \to \mathcal{R}_{even} \\
&\mathtt{Hash}_3 : \mathbb{F}_2^{2\lambda} \to \mathcal{R}_{odd} && \mathtt{Hash}_4 : \mathbb{F}_2^{2\lambda} \times \mathcal{R}_{even} \to \mathcal{R}^3_{\mathtt{wt}=t} \\
&\mathcal{L} : \mathcal{R}^3_{\mathtt{wt}=t} \to \mathbb{F}_2^{2\lambda} && \mathcal{K} : \mathbb{F}_2^{2\lambda} \times \mathcal{R} \times \mathcal{R} \times \mathbb{F}_2^{2\lambda} \to \mathbb{F}_2^{2\lambda}
\end{aligned}$$

Algorithm 1: KeyGen

Input: Security parameter 1^λ
Output: Public key pk, secret key sk
// Sample sparse secret polynomials and seeds
$\mathtt{h}_0(x) \xleftarrow{\$} \mathcal{R}_{\mathtt{wt}=v}$; $\mathtt{h}_1(x) \xleftarrow{\$} \mathcal{R}_{\mathtt{wt}=v}$; $\mathtt{h}_2(x) \xleftarrow{\$} \mathcal{R}_{\mathtt{wt}=v}$;
$\boldsymbol{\sigma} \xleftarrow{\$} \mathbb{F}_2^{2\lambda}, \boldsymbol{\sigma'} \xleftarrow{\$} \mathbb{F}_2^{2\lambda}$;
// Expand seed to public random polynomials
$\mathtt{t}_1(x) \leftarrow \mathtt{Hash}_1(\boldsymbol{\sigma}), \mathtt{t}_2(x) \leftarrow \mathtt{Hash}_2(\boldsymbol{\sigma}), \mathtt{r}_1(x) \leftarrow \mathtt{Hash}_3(\boldsymbol{\sigma})$;
// Compute public ratio
$\mathtt{t}_0(x) \leftarrow \frac{\mathtt{r}_1(x)\mathtt{h}_0(x)+\mathtt{h}_1(x)}{\mathtt{r}_1(x)+\mathtt{t}_1(x)}$; $\mathtt{r}_2(x) \leftarrow \frac{\mathtt{h}_2(x)+\mathtt{t}_0(x)\mathtt{t}_2(x)}{\mathtt{t}_0(x)+\mathtt{h}_0(x)}$;
$\mathsf{pk} \leftarrow (\boldsymbol{\sigma}, \mathtt{r}_2(x))$; $\mathsf{sk} \leftarrow (\boldsymbol{\sigma}, \boldsymbol{\sigma'}, \mathtt{h}_0(x), \mathtt{h}_1(x), \mathtt{h}_2(x))$;
return $(\mathsf{pk}, \mathsf{sk})$;

Algorithm 2: Encaps

Input: Public key $\mathsf{pk} = (\boldsymbol{\sigma}, \mathtt{r}_2)$
Output: Shared secret k, ciphertext ctx
$\mathbf{m} \xleftarrow{\$} \mathbb{F}_2^{2\lambda}$;
// Derive error vector and expand public polynomials
$\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x) \leftarrow \mathtt{Hash}_4(\mathbf{m}, \mathtt{r}_2)$;
$\mathtt{t}_1(x) \leftarrow \mathtt{Hash}_1(\boldsymbol{\sigma}), \mathtt{t}_2(x) \leftarrow \mathtt{Hash}_2(\boldsymbol{\sigma}), \mathtt{r}_1(x) \leftarrow \mathtt{Hash}_3(\boldsymbol{\sigma})$;
// Construct syndrome (ciphertext)
$\mathtt{u}(x) \leftarrow \mathtt{e}_0(x) + \mathtt{e}_1(x)\mathtt{r}_1(x) + \mathtt{e}_2(x)\mathtt{r}_2(x)$;
$\mathtt{v}(x) \leftarrow \mathtt{e}_0(x) + \mathtt{e}_1(x)\mathtt{t}_1(x) + \mathtt{e}_2(x)\mathtt{t}_2(x)$;
$\mathsf{ctx} \leftarrow (\mathtt{u}(x), \mathtt{v}(x), \mathbf{m} \oplus \mathcal{L}(\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)))$;
$\mathsf{k} \leftarrow \mathcal{K}(\mathbf{m}, \mathsf{ctx})$;
return k, ctx;

Algorithm 3: Decaps

Input: Secret key sk, ciphertext $\mathsf{ctx} = (\mathtt{u}(x), \mathtt{v}(x), \mathbf{tag})$
Output: Shared secret k
// Expand public and private polynomials
$\mathtt{t}_1(x) \leftarrow \mathtt{Hash}_1(\boldsymbol{\sigma}), \mathtt{t}_2(x) \leftarrow \mathtt{Hash}_2(\boldsymbol{\sigma}), \mathtt{r}_1(x) \leftarrow \mathtt{Hash}_3(\boldsymbol{\sigma})$;
$\mathtt{t}_0(x) \leftarrow \frac{\mathtt{r}_1(x)\mathtt{h}_0(x)+\mathtt{h}_1(x)}{\mathtt{r}_1(x)+\mathtt{t}_1(x)}$; $\mathtt{r}_2(x) \leftarrow \frac{\mathtt{h}_2(x)+\mathtt{t}_0(x)\mathtt{t}_2(x)}{\mathtt{t}_0(x)+\mathtt{h}_0(x)}$;
// Map to secret syndrome and decode
$\mathtt{s}(x) \leftarrow (\mathtt{h}_0(x) + \mathtt{t}_0(x))\,\mathtt{u}(x) - \mathtt{t}_0(x)\mathtt{v}(x)$;
$\mathtt{e}'_0(x), \mathtt{e}'_1(x), \mathtt{e}'_2(x) \leftarrow \textsc{BitFlippingDecoder}(\mathtt{h}_0, \mathtt{h}_1, \mathtt{h}_2, \mathtt{s})$;
$\mathbf{m}' \leftarrow \mathbf{tag} \oplus \mathcal{L}(\mathtt{e}'_0(x), \mathtt{e}'_1(x), \mathtt{e}'_2(x))$;
// Re-encryption check
if $(\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)) = \mathtt{Hash}_4(\mathbf{m}', \mathtt{r}_2(x))$ **then** $\mathsf{k} \leftarrow \mathcal{K}(\mathbf{m}', \mathsf{ctx})$;
// Decryption failure case, detectable by an attacker
else $\mathsf{k} \leftarrow \mathcal{K}(\boldsymbol{\sigma'}, \mathsf{ctx})$;
return k;

Instance	λ	p	$w = 3v$	t
TRIKE-1	128	9437	81	201
TRIKE-3	192	19181	123	319
TRIKE-5	256	32987	165	429
TRIKE-7	384	62981	249	659
TRIKE-9	512	103043	333	877

Table 1: Parameter sets of TRIKE [26]. λ is the bit-security level, p is the size of the circulant blocks, v (resp. w) is the Hamming weight of the columns (resp. rows) of the private parity-check matrix, t is the total Hamming weight of the error vector.

In a standard, static key recovery scenario, an attacker seeks to retrieve the secret polynomials $(\mathtt{h}_0(x), \mathtt{h}_1(x), \mathtt{h}_2(x))$ from the public key $pk = (\boldsymbol{\sigma}, \mathtt{r}_2(x))$. For TRIKE, this requires solving an instance of the QC-CFP problem. Indeed, the public element $\mathtt{r}_2(x)$ is linked to the secret blocks via the relation:

$$\mathtt{r}_2(x) = \mathtt{t}_2(x) + (\mathtt{r}_1(x) + \mathtt{t}_1(x))\frac{\mathtt{h}_2(x) + \mathtt{t}_2(x)\mathtt{h}_0(x)}{\mathtt{h}_1(x) + \mathtt{t}_1(x)\mathtt{h}_0(x)}$$

This yields a linear relation between the public and secret polynomials:

$$(\mathtt{y}(x)\mathtt{t}_1(x) + \mathtt{t}_2(x))\,\mathtt{h}_0(x) + \mathtt{y}(x)\mathtt{h}_1(x) + \mathtt{h}_2(x) = 0$$

Where:

$$\mathtt{y}(x) = \frac{\mathtt{r}_2(x) + \mathtt{t}_2(x)}{\mathtt{r}_1(x) + \mathtt{t}_1(x)} = \frac{\mathtt{h}_2(x) + \mathtt{t}_2(x)\mathtt{h}_0(x)}{\mathtt{h}_1(x) + \mathtt{t}_1(x)\mathtt{h}_0(x)}$$

This equality allows an attacker to model key recovery as a QC-CFP instance, where the target codeword has a Hamming weight equal to $w = 3v$, over the rate-2/3 parity-check matrix $\mathbf{H}_{pub}$:

$$\mathbf{H}_{pub} = \left[\mathtt{y}(x)\mathtt{t}_1(x) + \mathtt{t}_2(x) \mid \mathtt{y}(x) \mid I\right]$$

2.3 Bit-flipping Decoder

To recover the error vector $\mathbf{e} = [\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)]$ from the syndrome $\mathtt{s}(x) = \mathtt{e}_0(x)\mathtt{h}_0(x) + \mathtt{e}_1(x)\mathtt{h}_1(x) + \mathtt{e}_2(x)\mathtt{h}_2(x)$, TRIKE relies on an iterative procedure called parallel bit-flipping decoder [13,5,26]. Since the private parity-check matrix $\mathbf{H}$ defined by $(\mathtt{h}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x))$ is moderately sparse, the decoder can efficiently identify the erroneous bits (i.e., set bits in the error vector) by finding the corresponding columns of $\mathbf{H}$ that have been added to compute the syndrome (since $\mathbf{s} = \mathbf{He}$). To do so, the decoder initializes a guess vector $\hat{\mathbf{e}}$ to all zeros, and counts the number of unsatisfied parity checks (**upc**) associated with each bit

position in the error vector (i.e., $\mathbf{upc}_j = \sum_i \mathbf{H}_{i,j}\mathbf{s}_i$, where the sum is computed over the integers). If the **upc** counter for a specific bit exceeds a threshold, the decoder assumes that the bit is erroneous and flips it in $\hat{\mathbf{e}}$. After performing all the necessary flips, the decoder updates the syndrome $\mathbf{s}$, maintaining the invariant $\mathbf{s} = \mathbf{H}(\mathbf{e} + \hat{\mathbf{e}})$. The process is repeated until $\mathbf{s} = \mathbf{0}$, or until a maximum number of iterations is reached, resulting in a decoding failure: the Decoding Failure Rate (DFR) of such decoder is therefore greater than zero.

In 2016, the authors of [14] showed that it is possible to retrieve the private parity-check matrix of QC-MDPC based cryptosystems by observing the set of error vectors that are not decoded correctly by such private key under parallel bit-flipping decoding. Therefore, it is mandatory for cryptosystems targeting INDistinguishability against Chosen Ciphertext Attacks (IND-CCA) security, such as TRIKE, that the DFR of the underlying decoder is lower than $2^{-\lambda}$, where λ is the bit-security level. To estimate the DFR of TRIKE, the authors employ an extrapolated model inspired by the finite-length scaling (FLS) behavior of (non quasi-cyclic) LDPC codes [2].

2.4 Near Codewords

Near codewords are a family of error vectors that induce decoding failure with high probability in QC-LDPC and QC-MDPC codes.

Definition 3 (Near codewords [25]). *Given a QC parity-check matrix* $\mathbf{H} = [\mathtt{h}_0(x), \mathtt{h}_1(x), \mathtt{h}_2(x)]$*, a near codeword is a vector* $\mathbf{e} = [\mathtt{e}_0(x), \mathtt{e}_1(x), \mathtt{e}_2(x)]$*, such that* $\mathtt{e}_\mathtt{i}(x) = 0$ *for all blocks* i*, save for one block* b *where* $\mathtt{e}_\mathtt{b}(x) = x^a\mathtt{h}_\mathtt{b}(x)$ *(for any* $0 \leq a < p$*).*

To understand why a near codeword is hard to correct by a parallel bit-flipping decoder, it is sufficient to compute its syndrome: $\mathtt{s}(x) = \mathtt{e}_\mathtt{b}(x)\mathtt{h}_\mathtt{b}(x) = x^a{\mathtt{h}_\mathtt{b}}^2(x)$. Since squaring in $\mathcal{R}$ is equivalent to a permutation of the polynomial coefficients, and multiplying by a monomial x^a is equivalent to a circulant shift, we have that $\mathtt{wt}(\mathbf{e}) = v$ and $\mathtt{wt}(\mathtt{s}(x)) = v$. Iterative decoders work under the assumption that $\mathtt{wt}(\mathtt{s}(x)) = \Theta(vt)$, therefore an error vector with weight v and syndrome weight v (instead of the expected $\Theta(v^2)$) is basically impossible to decode, since the counters in the **upc** vector are all close to zero. Even when the error vector is not equal to a near codeword, a high overlap is sufficient to trigger a decoding failure [6,4].

Moreover, given a specific parameter set (i.e., block size p and column weight v), the higher failure probability caused by near codewords affects *all* QC-MDPC codes with such parameters, meaning that performing weak-keys filtering (as in TRIKE) does not alter the presence of such phenomenon.

3 Decoding Failure Rate Lower Bound

The near codeword phenomenon has been thoroughly studied for QC-MDPC codes having rate 1/2, but it can also affect codes with higher rates. Indeed, a

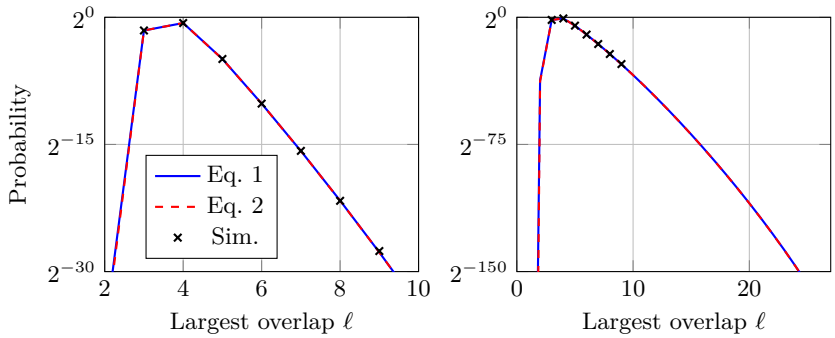


Fig. 1: Estimated values of $\zeta(\ell)$ for TRIKE-1, according to Equation 1, Equation 2, and numerical simulations with 10^9 samples.

high overlap between one error block and its corresponding parity-check block can still lead to a decoding failure, even if the parity-check matrix is composed by more than two blocks. In the following, we will denote as $\mathtt{e}_\mathtt{b}(x)$ $(0 \leq b < 3)$ the b-th p-sized block of the error vector, and with $\mathcal{H}_b$ the set of columns in the b-th block of $\mathbf{H}$ (i.e., $\mathcal{H}_b = \{x^a \mathtt{h}_\mathtt{b}(x) | 0 \leq a < p\}$). We compute a *lower bound* on the DFR of TRIKE by analysing the probability of converging towards a near codeword during decoding. Such probability is composed by the following two terms: 1) the probability $\zeta(\ell)$ that the highest overlap between any error block $\mathtt{e}_\mathtt{b}(x)$ and the corresponding near codeword set $\mathcal{H}_b$ equals ℓ; 2) the probability $\epsilon(\ell)$ that a decoding failure happens, given that the largest overlap between $\mathbf{e}$ and any near codeword is equal to ℓ.

$\zeta(\ell)$ can be easily computed as follows. Let $\psi(\ell)$ be the probability distribution function of the overlap between $\mathtt{e}_\mathtt{b}(x)$ and a random near codeword in $\mathcal{H}_b$. Then:

$$\psi(\ell) = \frac{\binom{v}{\ell}\binom{3p-v}{t-\ell}}{\binom{3p}{t}}$$

In the following, we assume that the overlaps between near codewords can be thought as mutually independent, as it is the case in the other QC-MDPC analyses in literature [6,19,4]. We therefore obtain that the p.d.f. of the maximum overlap between $\mathbf{e}$ and any near codeword is:

$$\left(\sum_{i=0}^{\ell} \psi(i)\right)^{3p} - \left(\sum_{i=0}^{\ell-1} \psi(i)\right)^{3p} \tag{1}$$

This formulation, however, does not take into account the fact that different overlaps are negatively correlated, since the sum of all the $3p$ overlaps between the error vector and each near codeword is bound to equal vt [**?**]. To take this fact into account, we compute $\zeta(\ell)$ as follows. Let $\mathcal{X}_i$ $(0 \leq i < 3p)$ be the random variable modelling the overlap between the i-th near codeword (for any arbitrary ordering) and the error vector, let $\mathcal{S}_i = \sum_{j=0}^{i-1} \mathcal{X}_j$, and let $\mathcal{M}_i = \max_{0 \leq j < i}\{\mathcal{X}_j\}$.

TRIKE Instance	Target DFR	Extrapolated DFR [26]	Samples	Lower Bound
TRIKE-1	2^{-128}	$2^{-174.40}$	10^8	$\mathbf{2^{-53.72}}$
TRIKE-3	2^{-192}	$2^{-193.41}$	10^7	$\mathbf{2^{-96.31}}$
TRIKE-5	2^{-256}	$2^{-271.58}$	10^7	$\mathbf{2^{-148.12}}$
TRIKE-7	2^{-384}	$2^{-393.30}$	10^7	$\mathbf{2^{-221.10}}$
TRIKE-9	2^{-512}	$2^{-560.01}$	10^7	$\mathbf{2^{-319.48}}$

Table 2: Lower bound on the decoding failure rate of TRIKE for all proposed parameter sets. *Extrapolated DFR* refers to the DFR estimate reported in the scheme specification [26]. *Samples* indicates the number of samples employed for each near-codeword overlap value ℓ to estimate its DFR contribution $\epsilon(\ell)$.

Starting from the known distribution of $(\mathcal{S}_0, \mathcal{M}_0)$, i.e., $\Pr\left((\mathcal{S}_0, \mathcal{M}_0) = (0,0)\right) = 1$, we derive the joint distribution of $(\mathcal{S}_{i+1}, \mathcal{M}_{i+1})$ given the one of $(\mathcal{S}_i, \mathcal{M}_i)$ as follows:

$$\Pr\left((\mathcal{S}_{i+1}, \mathcal{M}_{i+1}) = (s,m)\right) = \left(\sum_{a=0}^{m} \Pr\left((\mathcal{S}_i, \mathcal{M}_i) = (s-a, m)\right) \cdot \psi(a)\right) + \\ + \left(\sum_{b=0}^{m-1} \Pr\left((\mathcal{S}_i, \mathcal{M}_i) = (s-m, b)\right) \cdot \psi(m)\right) \tag{2}$$

We can accurately compute $\zeta(\ell)$, accounting for the negative correlation of different overlaps, as $\zeta(\ell) = \Pr\left(\mathcal{M}_{3p} = \ell | \mathcal{S}_{3p} = vt\right)$. Figure 1 shows the values of $\zeta(\ell)$ according to Equation 1 and Equation 2 for TRIKE-1. The two formulas yield essentially the same results, and they both agree with numerical simulations.

An analytical estimate for $\epsilon(\ell)$ is not as tractable. Since we are interested in a lower bound, we estimate $\epsilon(\ell)$ by means of numerical simulations for values of ℓ where decoding failures can be observed, and we set $\epsilon(\ell) = 0$ elsewhere. We thus obtain the following lower bound on the DFR:

$$DFR \geq \sum_{i=0}^{v} \zeta(i)\epsilon(i) \tag{3}$$

Table 2 reports the DFR lower bound computed with our presented methodology. The analysis has been performed for all parameter sets of TRIKE [26]. As it can be seen, the DFR of TRIKE is much higher than the DFR level required to claim IND-CCA security, for all parameter sets. To better understand how tight the reported lower bounds are, we show in Figure 2 and Figure 3 the measured values of $\epsilon(\ell)$, along with $\zeta(\ell)$ and the series of terms $\zeta(\ell)\epsilon(\ell)$ whose sum composes the DFR estimate, for TRIKE-1 and TRIKE-3. As it can be seen, for

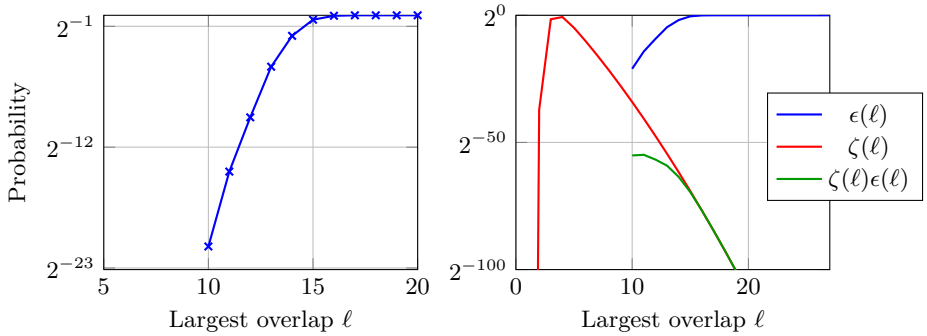


Fig. 2: Estimated values of $\epsilon(\ell)$ for TRIKE-1, along with $\zeta(\ell)$ and their product, for different largest overlap values.

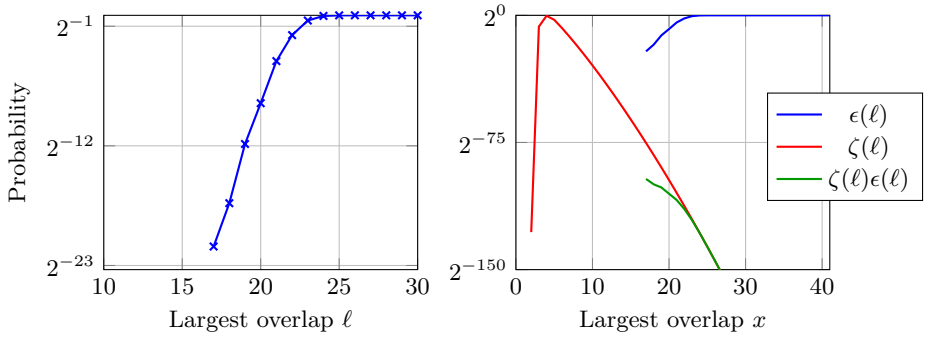


Fig. 3: Estimated values of $\epsilon(\ell)$ for TRIKE-3, along with $\zeta(\ell)$ and their product, for different largest overlap values.

TRIKE-1 the term $\zeta(\ell)\epsilon(\ell)$ reaches its peak at $\ell = 11$, meaning that the missing terms of $\epsilon(\ell)$ for $\ell \leq 9$ only have a marginal contribution to the overall DFR. This implies that the reported lower bound is expected to be near the actual DFR value of TRIKE-1. On the other hand, for TRIKE-3 (the same holds for higher security levels), the peak in the function $\zeta(\ell)\epsilon(\ell)$ is expected to be at $\ell < 17$ (which is the smallest input for which we were able to observe failures using 10^7 samples), meaning that the reported lower bound is not tight, and thus that the actual DFR is likely to be much higher.

4 Key Recovery Attack

The rationale behind our proposed key recovery attack is the following. Since the probability of a decoding failure is positively correlated with the highest overlap between the error $\mathbf{e}$ and one of the columns of $\mathbf{H}$, then wherever a decoding failure happens the pairwise distances in each p-sized block $\mathbf{e}_b(x)$ $(0 \leq b < 3)$

are more likely to appear in the support of one of the secret columns. This concept has been formalized in [14] with the notion of *distance spectrum*:

Definition 4 (Distance Spectrum [14]). *The **distance spectrum** $D(\mathbf{v})$ of a binary vector $\mathbf{v} \in \mathbb{F}_2^p$ (or a polynomial $\mathtt{f}(x) \in \mathcal{R}$) is the set of pairwise distances appearing in the support of $\mathbf{v}$ (equiv. $\mathtt{f}(x)$):*

$$D(\mathbf{v}) = \{d \in \mathbb{Z}_p \mid \exists i, j \in Supp(\mathbf{v}) \text{ and } (i-j) = d \mod p\}$$

*The **multiplicity** $\mu_{\mathbf{v}}(d)$ of a distance $d \in D(\mathbf{v})$ is the number of coordinate pairs i, j such that $i, j \in Supp(\mathbf{v})$ and $(i-j) = d \mod p$.*

Since the distance spectrum is rotation-invariant, we have that all near-codeword in each set $\mathcal{H}_b$ share the same distance spectrum; we indicate such set as $D(\mathcal{H}_b)$. Let $\epsilon(\ell)$ be the probability of a decoding failure given that the largest overlap between $\mathbf{e}$ and any near-codeword is equal to ℓ, and let $\zeta(\ell)$ be the p.m.f. of the largest overlap between $\mathbf{e}$ and any near-codeword.

In the first step of the algorithm, the attacker uses the information provided by decoding failures to recover the distance spectrum of *one* of the blocks in $\mathbf{H}$ (or a noisy version of it, possibly containing false positives). In the second step, the attacker retrieves two candidates for such secret block. The last step exploits the two candidates to reduce the CFP instance with weight $3v$ over a code with rate $2/3$ to two SDP instances with weight $2v$ over a code with rate $1/2$, one of which admits the rest of the private key as solution.

4.1 Retrieving the Distance Spectrum

We now compute, given a uniformly random error vector $\mathbf{e}$ and a specific pairwise distance d appearing within a single block b in $\mathbf{e}$ (i.e., $d \in D(\mathtt{e_b}(x))$, $0 \leq b < 3$), the probability of a decoding failure happening, conditioned over the multiplicity of such distance d in $D(\mathcal{H}_b)$ (assuming $\mu_{\mathcal{H}_b}(d) = 0$ if $d \notin D(\mathcal{H}_b)$). We start by computing the p.m.f. $\widetilde{\zeta}(\ell, m)$ of the highest overlap between $\mathbf{e}$ and any near-codeword, conditioned over the event $\mu_{\mathcal{H}_b}(d) = m$. We have that at least $\mu_{\mathcal{H}_b}(d)$ near-codewords share at least 2 asserted bits with the error. On such near-codewords, the overlap with $\mathbf{e}$ is bound to the following probability distribution function:

$$\widetilde{\psi}(\ell) = \begin{cases} 0 & \text{if } \ell < 2 \\ \frac{\binom{v-2}{\ell-2}\binom{n-v}{t-\ell}}{\binom{n-2}{t-2}} & \text{if } \ell \geq 2 \end{cases}$$

For the other $3p - m$ near codewords, the overlap distribution is the same as before ($\psi(\ell)$). The desired distribution $\widetilde{\zeta}(\ell, m)$ can be easily approximated as follows:

$$\left(\sum_{i=0}^{\ell} \widetilde{\psi}(i)\right)^m \cdot \left(\sum_{i=0}^{\ell} \psi(i)\right)^{3p-m} - \left(\sum_{i=0}^{\ell-1} \widetilde{\psi}(i)\right)^m \cdot \left(\sum_{i=0}^{\ell-1} \psi(i)\right)^{3p-m}$$

	TRIKE-1	**TRIKE-3**	**TRIKE-5**	**TRIKE-7**	**TRIKE-9**
$\mathtt{p}_{\mathtt{inc}\|0}$	0.3763	0.4437	0.4607	0.5342	0.5629
$\mathtt{p}_{\mathtt{inc}\|1}$	0.4164	0.4863	0.5117	0.5775	0.6084
$\mathtt{p}_{\mathtt{inc}\|2}$	0.4565	0.5289	0.5628	0.6208	0.6539
$\mathtt{p}_{\mathtt{inc}\|3}$	0.4966	0.5714	0.6139	0.6641	0.6994

Table 3: Probability that $d \in D(\mathtt{e}_\mathtt{b}(x))$ in error vectors triggering a decoding failure, conditioned over the multiplicity of the corresponding distance $\mu_{\mathcal{H}_b}(d) = m$, for $0 \leq m \leq 3$, computed over different TRIKE parameter sets.

To correctly account for the negative covariance between different overlaps, we employ the same formula for $\zeta(\ell)$ shown in the previous section, substituting $\psi(\ell)$ with $\widetilde{\psi}(\ell)$ in Equation 2 for $1 \leq i \leq m$. The probability that a failure happens, given that $d \in D(\mathtt{e}_\mathtt{b}(x))$ and $\mu_{\mathcal{H}_b}(d) = m$, is therefore $\sum_i \widetilde{\zeta}(i, m)\epsilon(i)$. Finally, given a random error vector $\mathbf{e}$, the probability that $d \in D(\mathtt{e}_\mathtt{b}(x))$ is (by Kaplansky's second lemma [16]):

$$\Pr(d \in D(\mathtt{e}_\mathtt{b}(x))) = 1 - \sum_i \frac{\binom{p}{i}\binom{2p}{t-i}}{\binom{3p}{t}} \cdot \frac{\frac{p}{p-i}\binom{p-i}{i}}{\binom{p}{i}}$$

Now, we can compute the probability $\mathtt{p}_{\mathtt{inc}|m}$ that, whenever a failure happens, a specific distance appears in $D(\mathtt{e}_\mathtt{b}(x))$, given that the multiplicity of the corresponding distance d is $\mu_{\mathcal{H}_b}(d) = m$. Let FAIL be the event of a failure, and DFR be the decoding failure rate computed in the previous section (Equation 3). Then:

$$\mathtt{p}_{\mathtt{inc}|m} := \Pr\left(d \in D(\mathtt{e}_\mathtt{b}(x)) \mid \text{FAIL} \wedge \mu_{\mathcal{H}_b}(d) = m\right) =$$

$$= \frac{\Pr\left(\text{FAIL} \mid d \in D(\mathtt{e}_\mathtt{b}(x)) \wedge \mu_{\mathcal{H}_b}(d) = m\right) \Pr\left(d \in D(\mathtt{e}_\mathtt{b}(x))\right)}{\Pr(\text{FAIL})} =$$

$$= \frac{\Pr(d \in D(\mathtt{e}_\mathtt{b}(x))) \cdot \sum_i \widetilde{\zeta}(i, m)\epsilon(i)}{DFR}$$

Table 3 shows the values of $\mathtt{p}_{\mathtt{inc}|m}$ for $0 \leq m \leq 3$ and different TRIKE parameter sets. As it can be seen, distances having multiplicity $\mu_{\mathcal{H}_d} \geq 1$ are more likely to appear in the error vector spectrum whenever a decoding failure takes place.

Now, consider the following key recovery attack procedure. First, the attacker initializes three p-sized integer vectors acting as histograms $\mathbf{ctr}^{(b)} = \mathbf{0}$, $0 \leq b < 3$. Then, the attacker observes the error vectors that cause a decoding failure during q decryption queries. Whenever a failure happens, the entries $\mathbf{ctr}_d^{(b)}$ such that $d \in D(\mathtt{e}_\mathtt{b}(x))$ are increased by one. Let $\mathcal{F}$ be the r.v. modelling the number of failures taking place during q decryption queries, and let $\mathcal{B}_m$ be the r.v. modelling the entries of $\mathbf{ctr}$ in positions corresponding to distances having multiplicity $\mu(d) = m$ in their respective blocks. Then, $\mathcal{F} \sim \text{Bin}(q, DFR)$ and $\mathcal{B}_m \sim \text{Bin}(\mathcal{F}, \mathtt{p}_{\mathtt{inc}|m})$.

After q queries, the attacker can identify the distances appearing in the spectrum by selecting the $v(v-1)$ positions with the highest value in **ctr**. Indeed, assuming $\mu_{\mathcal{H}_b}(d) = 1$ for all distances in the spectrum $D(\mathcal{H}_b)$, we have that $|D(\mathcal{H}_b)| = v(v-1)$. Indicating as $\max_N\{\mathcal{B}_m\}$ (resp. $\min_N\{\mathcal{B}_m\}$) the maximum (resp. minimum) among N samples of $\mathcal{B}_m$, an estimate of the probability of successfully isolating the distance spectrum of one block is:

$$\mathtt{p_{succ}} = \sum_{f=0}^{q} \Pr(\mathcal{F} = f) \Pr(\max_{p-v(v-1)}\{\mathcal{B}_0\} < \min_{v(v-1)}\{\mathcal{B}_1\})$$

This is a conservative estimate for the success probability. Indeed, in the case where some distances have multiplicity $\mu(d) > 1$, the corresponding probability $\mathtt{p_{inc|}}_m$ is larger than $\mathtt{p_{inc|}}_1$. If a coordinate pair $i, j \in \mathrm{Supp}(\mathtt{e_b}(x))$ aligns with more than one near codeword, then the distribution of the largest intersect between $\mathtt{e_b}(x)$ and any near codeword in $\mathcal{H}_b$ is skewed towards higher values, making decoding failures even more likely and therefore increasing the probability of successfully identifying the corresponding distance in the spectrum. This fact, however, introduces two problems.

- If a block has a high number of distances with $\mu(d) > 1$, then failures are more likely to be related to such block. This implies that the counters in the other two blocks are mainly updated with incorrect increments. For this reason, we employ the distance spectrum recovery to find just *one* of the three secret blocks. The probability of finding at least one of the three secret blocks in this scenario is lower-bounded by $\mathtt{p_{succ}}$.
- If some distances in a block have multiplicity greater than 1, then $|D(\mathcal{H}_b)| < v(v-1)$, meaning that some of the $v(v-1)$ selected distances are not in the spectrum. We show how to deal with such false positives in the next section.

4.2 Retrieving a Block from its Distance Spectrum

Given a candidate set Φ containing the distance spectrum $D(\mathcal{H}_b)$ of one block, it is easy to find *two* candidates for the corresponding sparse block $\mathtt{h_b}(x)$ composing the private parity-check matrix [14]. We hereby provide a variant of the block recovery procedure of [14].

For any rotation $\mathtt{h_b^{(i)}}(x) = x^i \mathtt{h_b}(x)$ such that $\mathtt{h_b^{(i)}}(0) = 1$ (i.e., such that the first coefficient in the polynomial is 1) the following holds: $\mathrm{Supp}(\mathtt{h_b^{(i)}}(x)) \subset D(\mathcal{H}_b)$. Moreover, by selecting a non-zero distance $j \in D(\mathcal{H}_b)$, if $j \in \mathrm{Supp}(\mathtt{h_b^{(i)}}(x))$ then we also have that:

$$\mathrm{Supp}(\mathtt{h_b^{(i)}}(x)) \subset (D(\mathcal{H}_b) \cap \{d + j \mod p | d \in D(\mathcal{H}_b)\})$$

The procedure employed to retrieve the secret block given its distance spectrum is shown in Algorithm 4. By performing a simple tree-based search, starting from the set $\Phi_0 = \Phi$ and recursively computing $\Phi_{i+1} = \Phi_i \cap \{d + j | d \in \Phi\}$ for every $j \in \Phi_i$ (plus a mechanism to avoid visiting the same nodes multiple times),

Algorithm 4: RetrieveBlockFromDS

Input: Initial distance set Φ, current distance set Φ_i, ordering limit ℓ
Output: Valid block candidates `sol`

1 **if** $|\Phi_i| < v$ **then return** $\emptyset$;
2 **if** $|\Phi_i| = v$ **and** $\ell = \max\{\Phi_i\}$ **then return** $\{\Phi_i\}$;
3 `sol` $\leftarrow \emptyset$;
4 **for** $j \in \Phi_i$ **do**
5 | **if** $j \leq \ell$ **then continue**;
6 | $\Phi_{i+1} \leftarrow \Phi_i \cap \{d + j \mod p \mid d \in \Phi\}$;
7 | `sol` $\leftarrow$ `sol` $\cup$ RetrieveBlockFromDS(Φ, Φ_{i+1}, j);
8 **return** `sol`

after a small amount of steps we obtain $|\Phi_i| = v$, deriving at the end a total of $2v$ candidate supports of size v. Out of these, v correspond to the v vectors $\mathtt{h_b^{(i)}}(x) = x^i \mathtt{h_b}(x)$ such that $\mathtt{h_b^{(i)}}(0) = 1$. The v additional candidates stem from the fact that $d \in D(\mathcal{H}_b)$ implies $(-d \mod p) \in D(\mathcal{H}_b)$, and constitute therefore the transposed copies of $\mathtt{h_b^{(i)}}(x)$. Ignoring rotated versions of the same supports, we obtain two candidates for $\mathtt{h_b}$: the true value (up to a cyclic shift) and its transpose. Regarding the correctness of Algorithm 4 we have that, if $D(\mathcal{H}_b) \not\subseteq \Phi$, then the probability of executing line 2 exactly $2v$ times, obtaining two groups of valid candidates composed by circulant shifts of the same support (or its transposition), is negligible. Therefore, if Algorithm 4 returns a non-empty set of solutions, a simple check on the outputs is sufficient to determine if the found solutions are correct or not. Since the presence of some false positives in Φ does not alter the correctness of the procedure, we can select more than $v(v-1)$ terms in the recovered set with the goal of increasing the probability of including the whole spectrum in Φ.

We now analyse the complexity of Algorithm 4, assuming $|\Phi| = \alpha \cdot v(v-1)$ (i.e., where Φ contains at least $(\alpha - 1) \cdot v(v-1)$ false positives). The size of the input set Φ_0 is $|\Phi|$, and it is reduced approximately by a factor of $\frac{|\Phi|}{p}$ in each recursive call. This means that the depth of the recursion tree δ is such that $((\alpha v(v-1))/p)^\delta \approx v/p$, hence:

$$\delta \approx \frac{\ln(v) - \ln(p)}{\ln(\alpha) + 2\ln(v) - \ln(p)}$$

The complexity of Algorithm 4 is therefore $\mathcal{O}\left((\alpha v^2)^\delta\right)$. By setting $\alpha = 1$ (i.e., without considering intentional false positives), the exponent δ is bounded between 2.2 and 3.3 for all parameter sets of TRIKE, meaning that the procedure has a manageable complexity for all input regimes of interest. The value of α can be tuned in order to increase the success probability and thus reduce the number of required decryption queries; in the attack to the toy parameter set presented in the next section, we employed $\alpha = 5$.

4.3 Retrieving the Private Key

Once the two candidates for a block of the secret matrix $\mathtt{h}_\mathtt{b}(x)$ have been found, we can modify the hard problem to be solved to recover the complete secret key. Starting from the QC-CFP instance whose solution is the private key, i.e.,

$$\mathtt{c}_0(x)\mathtt{h}_0(x) + \mathtt{c}_1(x)\mathtt{h}_1(x) + \mathtt{c}_2(x)\mathtt{h}_2(x) = 0,$$

let i, j be the two blocks different from b. We can compute the term $\mathtt{s}(x) = \mathtt{c}_\mathtt{b}(x)\mathtt{h}_\mathtt{b}(x)$, obtaining the following Quasi-Cyclic Syndrome Decoding Problem (QC-SDP) instance:

$$\mathtt{c}_\mathtt{i}(x)\mathtt{h}_\mathtt{i}(x) + \mathtt{c}_\mathtt{j}(x)\mathtt{h}_\mathtt{j}(x) = \mathtt{s}(x)$$

By solving the two equations (corresponding to the two candidates for $\mathtt{h}_\mathtt{b}(x)$) in parallel, the attacker can retrieve the two remaining blocks as soon as one of the two QC-SDP instances yields a valid solution.

5 Attack Validation and Complexity Estimation

In this section, we present the results of our test aimed at validating the effectiveness of our key recovery attack, and the complexity estimates for a KRA against instances of TRIKE targeting NGCC security level 1.

5.1 Complete Key Recovery on Toy Parameters

We validate our attack by performing a complete key recovery, under no assumption, on a toy parameter set. We select $p = 3011$, $v = 13$ (hence $w = 39$), $t = 86$, and the following fixed-threshold policy for the 7-iterations decoder: $\mathtt{th}_\mathtt{iter} = (10, 8, 7, 7, 7, 7, 7)$, noting that the use of a more complex threshold policy (e.g., based on the Hamming weight of the syndrome, as in TRIKE) can improve the DFR for smaller code lengths, but does not alter the effectiveness of our attack. Such parameters provide a KRA security of 65.8 bits (i.e., the standard QC-CFP based KRA requires $2^{65.8}$ bit operations according to [11]), and the FLS-based interpolation over simulations in the waterfall regime employed by TRIKE extrapolates a DFR of 2^{-87}. Due to the near-codeword phenomenon, the true DFR of such codes is approximately $2^{-18.95}$. The DFR of QC-MDPC codes with these parameters (varying r), along with the FLS interpolation in the waterfall region, is shown in Figure 4.

The attack was performed on a server equipped with two AMD EPYC 7551 CPU @ 2.0 GHz and 256 GB of RAM. For solving the reduced QC-SDP instance, we employed the Dumer ISD [12] implementation available at [24]. We carried out 10 attacks on different private keys: after an average of $\approx 2^{29.50}$ total decryption queries the attack script recovered one of the secret blocks of each key, and the resulting QC-SDP instances were solved in around 5 to 30 minutes each, successfully retrieving the full secret keys.

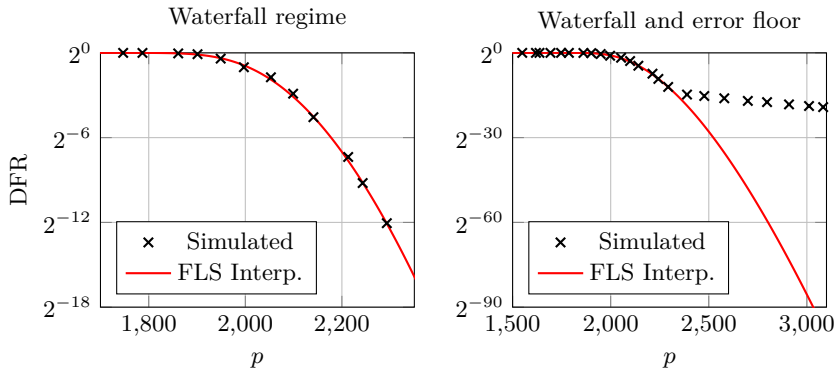


Fig. 4: Decoding failure rate of TRIKE QC-MDPC codes with $v = 13$ and $t = 86$. *Simulated DFR* has been obtained by performing 10^8 decoding trials for each data point. *FLS Interp.* has been computed via FLS-based interpolation over the data points in the range $1901 \leq p \leq 2293$, using the procedure described in TRIKE specification [26].

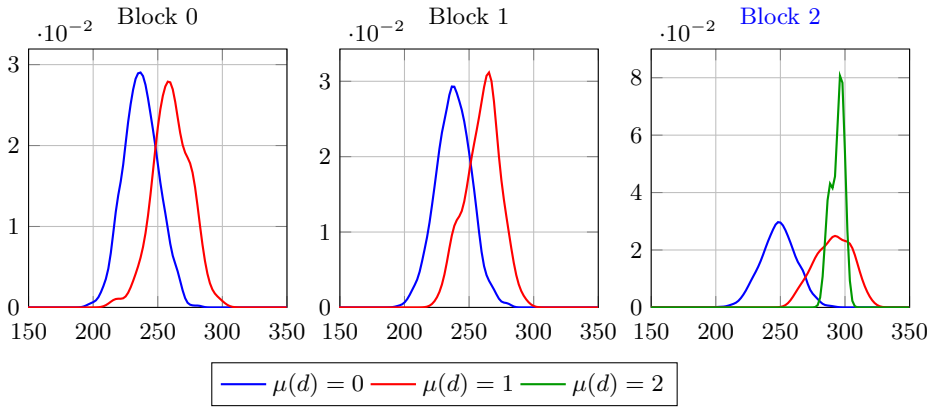


Fig. 5: Normalized distribution of the values of **ctr** for entries not in the distance spectrum ($\mu(d) = 0$, blue line) and entries in the distance spectrum ($\mu(d) = 1$ and $\mu(d) = 2$, red and green lines respectively), for each of the three blocks composing the private key. Block 2 is the one retrieved by our attack script.

Figure 5 shows the distribution of the counters in **ctr** in one attack instance. In this specific case, blocks 0 and 1 did not contain any distance with multiplicity $\mu(d) \geq 2$, while block 2 had several distances with $\mu(d) = 2$. As it can be seen, in the first two blocks the counters distributions are not distinct enough to allow a recovery. For the last block, since distances with $\mu(d) = 2$ lead to a higher failure probability, the distributions for distances inside and outside the

TRIKE Instance	Std. KRA Compl.[26]	DFR Lower Bound	Required Queries	QC-SDP Compl.[11]
TRIKE-1	$\mathbf{2^{133.83}}$	$\mathbf{2^{-53.72}}$	$\mathbf{2^{66.9}}$	$\mathbf{2^{65.52}}$
TRIKE-3	$2^{200.01}$	$2^{-96.31}$	$2^{109.5}$	$2^{92.90}$
TRIKE-5	$2^{266.32}$	$2^{-148.12}$	$2^{160.8}$	$2^{120.63}$
TRIKE-7	$2^{398.25}$	$2^{-221.10}$	$2^{234.4}$	$2^{175.96}$
TRIKE-9	$2^{530.91}$	$2^{-319.48}$	$2^{332.7}$	$2^{231.56}$

Table 4: Complexity estimates for our attack against all TRIKE parameter sets. *Std. KRA Compl.* reports the standard KRA bit-operations-complexity as reported in TRIKE specification [26]. *Required Queries* shows the estimated amount of decryption queries required to obtain $\mathtt{p_{succ}} \geq 0.9$ according to our model.

spectrum are more distinct. Despite the overlap in the distributions for counters inside and outside the spectrum, the resilience of the block recovery procedure to false positives allowed recovering the last block of the private key, by including a number of entries larger than $v(v-1)$ (specifically, $5v(v-1)$) in the block recovery phase.

5.2 Attack Complexity Estimation

Table 4 shows the complexity estimates of each step of our key recovery attack, for our toy parameter set and all TRIKE parameter sets. The complexity of the QC-SDP step has been estimated according to [11], and multiplied by two since the attacker has to solve the two QC-SDP instances simultaneously. The number of required decryption queries is estimated by forcing $\mathtt{p_{succ}} \geq 0.9$. To compute $\mathtt{p_{succ}}$ in a conservative manner, we use the same values of $\epsilon(\ell)$ employed in Section 3. Note that, with such criterion, the estimated number of queries for the toy parameter set is $2^{31.7}$, which is higher than the $2^{29.50}$ queries that allowed the attack to work in practice. Moreover, we highlight the fact that the attack does not require plaintext/ciphertext pairs to be stored, as it is sufficient to elaborate the (few thousands of) error vectors that cause a decoding failure, sequentially. For NGCC security level 1, the number of decryption queries required by our attack is $2^{66.9}$, below the 2^{80} limit imposed by NGCC guidelines [15], and the cost of the resulting QC-SDP instance is $2^{65.52}$ (i.e., feasible with modern computational resources).

6 Conclusions

In this work, we analysed the decoding failure rate of the NGCC candidate TRIKE [26]. We showed that the family of error vectors known as near code-

words affects the average DFR of the QC-MDPC codes employed in TRIKE. We quantified the DFR of TRIKE after taking the existence of near codewords into account, obtaining lower bounds that exceed the DFR levels required by IND-CCA security for all parameter sets. Then, we showed how decoding failures induced by near codewords can be efficiently employed to retrieve a block of the secret key, reducing the complexity of the ISD-based step by a large factor. For TRIKE-1, the complete attack requires $2^{66.9}$ decryption queries and an ISD-based search with a complexity of $2^{65.52}$ bit operations. Our attack technique can be potentially improved by precomputing chosen ciphertexts with the goal of reducing the number of decryption queries, or by cancelling the cross-block noise in the counter computation to retrieve the distance spectrum of multiple blocks simultaneously, thus removing the ISD step altogether.

We note that there are possible avenues to compensate for this attack, although incurring in performance penalties. The easiest one would be to increase the block size p until the DFR estimates fall below $2^{-\lambda}$, but the slow decrease in the DFR curve would cause the required p to be much larger than its current value. More effective countermeasures consist in either increasing the weight v of the parity-check matrix column (thus reducing the probability of a large enough overlap between the errors and near codewords [6,4]), or opting for near-codewords-aware decoding algorithms [7,22]. Both these solutions, however, heavily penalize the speed performance of the decryption procedure, which is the main advantage of TRIKE over more conventional KEM designs such as BIKE.

References

1. 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. pp. 298–307. IEEE Computer Society (2003). https://doi.org/10.1109/SFCS.2003.1238204
2. Amraoui, A., Montanari, A., Richardson, T.J., Urbanke, R.L.: Finite-length scaling for iteratively decoded LDPC ensembles. IEEE Trans. Inf. Theory **55**(2), 473–498 (2009). https://doi.org/10.1109/TIT.2008.2009580
3. Annechini, A., Barenghi, A., Pelosi, G.: On the Hardness of Decoding Quasi-Cyclic Codes and the Security of Code-Based Public-Key Cryptosystems. In: Pietro, R.D., Renaud, K., Mori, P. (eds.) Proceedings of the 12th International Conference on Information Systems Security and Privacy, ICISSP 2026 - Volume 2, Marbella, Spain, March 04-06, 2026. pp. 49–60. SCITEPRESS (2026). https://doi.org/10.5220/0014296400004061, iSBN 978-989-758-800-6, ISSN 2184-4356
4. Annechini, A., Barenghi, A., Pelosi, G., Perriello, S.: Efficient QC-MDPC cryptosystems with bounded decoding failure rate. In: Heninger, N., Rosulek, M. (eds.) Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 16803, pp. 347–379. Springer (2026). https://doi.org/10.1007/978-3-032-35398-6_12
5. Aragon, N., Barreto, P.S.L.M., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Ghosh, S., Gueron, S., Güneysu, T., Melchor, C.A., Misoczki,

R., Persichetti, E., Richter-Brockmann, J., Sendrier, N., Tillich, J.P., Vasseur, V., Zémor, G.: BIKE: Bit Flipping Key Encapsulation. Round 4 Submission - Version 5.2 - 10/10/2024. [Online] Available: https://bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf (2024)
6. Arpin, S., Lau, J.B., Mesnard, A., Perlner, R.A., Robinson, A., Tillich, J., Vasseur, V.: Error Floor Prediction with Markov Models for QC-MDPC Codes. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16000, pp. 221–252. Springer (2025). https://doi.org/10.1007/978-3-032-01855-7_8
7. Baldelli, A., Baldi, M., Zuane, D.D., Santini, P.: Near-codewords aware bit flipping decoding of QC-MDPC codes. In: IEEE International Symposium on Information Theory, ISIT 2026, Guangzhou, China, June 28 - July 3, 2026. pp. 1–6. IEEE (2026). https://doi.org/10.1109/ISIT62367.2026.11653920
8. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How 1+ 1= 0 improves information set decoding. In: Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31. pp. 520–536. Springer (2012)
9. Berger, T.P., Cayrel, P., Gaborit, P., Otmani, A.: Reducing Key Length of the McEliece Cryptosystem. In: Preneel, B. (ed.) AFRICACRYPT 2009, Second International Conference on Cryptology in Africa, Gammarth, Tunisia, June 21-25, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5580, pp. 77–97. Springer (2009). https://doi.org/10.1007/978-3-642-02384-2_6
10. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems (Corresp.). IEEE Trans. Inf. Theory **24**(3), 384–386 (1978). https://doi.org/10.1109/TIT.1978.1055873
11. Carrier, K., Hatey, V., Luzzi, L., Tillich, J.P.: Multilevel amortized gaussian elimination in information-set decoding: Applications to HQC and PCG. Cryptology ePrint Archive, Paper 2026/1498 (2026), https://eprint.iacr.org/2026/1498
12. Dumer, I.: On Minimum Distance Decoding of Linear Codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52. Moscow (1991)
13. Gallager, R.G.: Low-density parity-check codes. IRE Trans. Inf. Theory **8**(1), 21–28 (1962). https://doi.org/10.1109/TIT.1962.1057683
14. Guo, Q., Johansson, T., Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 789–815 (2016). https://doi.org/10.1007/978-3-662-53887-6_29
15. Institute of Commercial Cryptography Standards: Evaluation criteria for public-key cryptographic algorithms. Tech. rep., ICCS (2025), https://www.niccs.org.cn/niccs/Notice/tT7TSQiz.pdf, next-generation Commercial Cryptographic Algorithm Program (NGCC)
16. Kaplansky, I.: Solution of the problème des ménages. Bulletin of the American Mathematical Society **49**, 784–785 (1943)
17. Lee, P.J., Brickell, E.F.: An Observation on the Security of McEliece's Public-Key Cryptosystem. In: Günther, C.G. (ed.) Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings. Lecture Notes in Computer

Science, vol. 330, pp. 275–280. Springer (1988). https://doi.org/10.1007/3-540-45961-8_25
18. Melchor, C.A., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J.C., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J.M., Véron, P., Zémor, G.: Hamming Quasi-Cyclic (HQC) - v5.1. https://pqc-hqc.org/doc/hqc_specifications_v5.1.pdf (2026)
19. Mesnard, A., Tillich, J., Vasseur, V.: The syndrome weight distribution in quasi-cyclic codes, applications to BIKE and HQC. In: Bardet, M., Niederhagen, R. (eds.) Post-Quantum Cryptography - 17th International Workshop, PQCrypto 2026, Saint-Malo, France, April 14-16, 2026, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16491, pp. 156–187. Springer (2026). https://doi.org/10.1007/978-3-032-22695-2_6
20. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory (1986), https://vdoc.pub/download/knapsack-type-cryptosystems-and-algebraic-coding-theory-2tgbc23plpdg
21. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962)
22. Santini, P., Zuane, D.D., Baldelli, A., Baldi, M.: Flip a failure into a success: Improved bit flipping decoding for QC-MDPC codes. IACR Cryptol. ePrint Arch. **2026**, 1616 (2026), https://eprint.iacr.org/2026/1616
23. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings. Lecture Notes in Computer Science, vol. 388, pp. 106–113. Springer (1988). https://doi.org/10.1007/BFB0019850
24. Vasseur, V.: An implementation of dumer's algorithm for information set decoding. https://github.com/vvasseur/isd (2019)
25. Vasseur, V.: Post-quantum cryptography: a study of the decoding of QC-MDPC codes. (Cryptographie post-quantique : étude du décodage des codes QC-MDPC). Ph.D. thesis, University of Paris, France (2021)
26. Wang, T., Li, Y., Tian, J., Zhou, Y., Zhang, Z., Wang, A.: TRIKE: Triple Quasi-Cyclic Code Key Encapsulation Mechanism. Submission to the Next-Generation Commercial Cryptographic algorithms program (NGCC), Round 1 (2026), https://www.niccs.org.cn/niccs/Proposal/Public-Key%20Cryptographic%20Algorithms/Round%201%20candidates/Key%20Encapsulation/TRIKE.zip, Algorithm Specifications and Supporting Documentation (v1.6)