LEDAkem and LEDApkc: key encapsulation and public-key cryptography based on QC-LDPC codes

Paolo Santini

Università Politecnica delle Marche
p.santini@pm.univpm.it

Code-Based Cryptography Workshop 2018

Fort Lauderdale, Florida, USA
April 5-6, 2018

Introduction

- Code-based public-key cryptosystems were introduced by McEliece in 1978.
- In 1986 Niederreiter introduced another code-based public-key cryptosystem in the syndrome domain, while McEliece works in the codeword domain.
- The main drawback of these systems is represented by the dimension of the public key.

- R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.
- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, vol. 15, pp. 159–166, 1986.
- Y. X. Li, R. H. Deng and X. M. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," IEEE Trans. Inf. Theory, vol. 40, no. 1, pp. 271–273, Jan 1994.

Alternatives to Goppa codes

Goppa codes
[McEliece78]

GRS codes
[Niederreiter86]

QD codes
[MisBar09]

Conv. codes
[LönJoh12]

LDPC codes
[MonRosSho00]

QC codes
[Gaborit05]

GRS subcodes
[BerLoi05]

QC-LDPC codes
[BalBodChi08]

Transformed GRS codes
[BalBiaChiRosSch16]

QC-MDPC codes
[MisTilSenBar12]

Alternatives to Goppa codes

Goppa codes
[McEliece78]

GRS codes
[Niederreiter86]

QD codes
[MisBar09]

Conv. codes
[LönJoh12]

LDPC codes
[MonRosSho00]

QC codes
[Gaborit05]

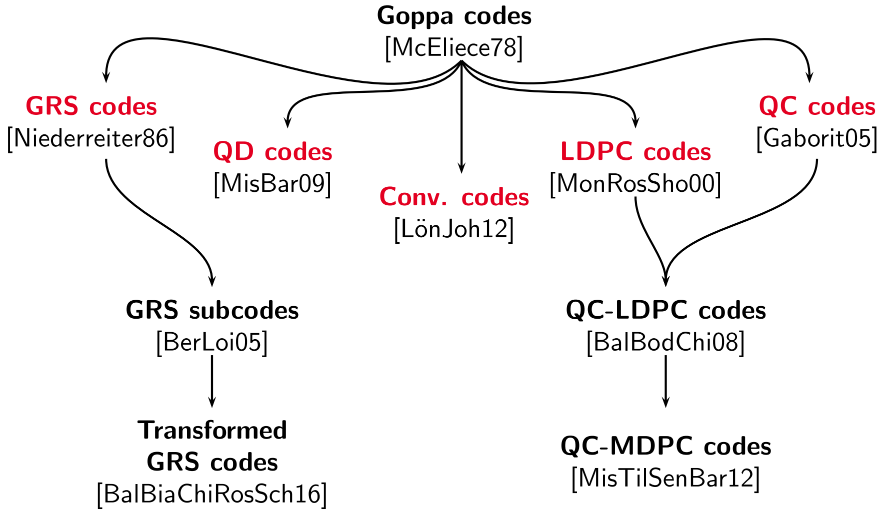
GRS subcodes
[BerLoi05]

Transformed GRS codes
[BalBiaChiRosSch16]

QC-LDPC codes
[BalBodChi08]

QC-MDPC codes
[MisTilSenBar12]

Alternatives to Goppa codes



LEDAkem and LEDApkc

- LEDAkem and LEDApkc are two proposals for the NIST competition, based on QC-LDPC codes.
- LEDAkem (Low dEnsity parity-check coDe-bAsed key encapsulation mechanism):
  - Key Encapsulation Mechanism (KEM) built upon the Niederreiter framework.

- E. Persichetti, "Secure and anonymous hybrid encryption from coding theory," in Post-Quantum Cryptography, P. Gaborit, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 174 - 187.
- M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini: "LEDAkem: first round submission to the NIST post-quantum cryptography call," November 2017.

LEDAkem and LEDApkc

- LEDAkem and LEDApkc are two proposals for the NIST competition, based on QC-LDPC codes.
- LEDAkem (Low dEnsity parity-check coDe-bAsed key encapsulation mechanism):
  - Key Encapsulation Mechanism (KEM) built upon the Niederreiter framework.
- LEDApkc (Low-dEnsity parity-check coDe-bAsed public-key cryptosystem):
  - Public Key Cryptosystem (PKC) built upon the McEliece framework.

- E. Persichetti, "Secure and anonymous hybrid encryption from coding theory," in Post-Quantum Cryptography, P. Gaborit, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 174 - 187.
- M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini: "LEDAkem: first round submission to the NIST post-quantum cryptography call," November 2017.
- M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini: "LEDApkc: first round submission to the NIST post-quantum cryptography call," November 2017.

Secret and public codes building blocks

- The secret code is an $[n, k]$ QC-LDPC code, with $n = n_0 p$ and $k = (n_0 - 1)p$, with parity check matrix in the form

$$\mathbf{H} = [\mathbf{H}_0, \mathbf{H}_1, \cdots, \mathbf{H}_{n_0-1}]$$

with each $\mathbf{H}_i$ being a circulant matrix of size p and weight $d_v \ll p$.

Secret and public codes building blocks

- The secret code is an $[n, k]$ QC-LDPC code, with $n = n_0 p$ and $k = (n_0 - 1)p$, with parity check matrix in the form

$$\mathbf{H} = [\mathbf{H}_0, \mathbf{H}_1, \cdots, \mathbf{H}_{n_0-1}]$$

  with each $\mathbf{H}_i$ being a circulant matrix of size p and weight $d_v \ll p$.

- The public code is constructed upon $\mathbf{H}$ and a $n \times n$ matrix $\mathbf{Q}$, in QC-form, with row and column weight equal to $m \ll n$.

LEDApkc - Key generation

Secret key

1. Generate a random $p \times n$ binary block circulant matrix $\mathbf{H} = [\mathbf{H}_0, \ldots, \mathbf{H}_{n_0-1}]$ with column weight $d_v \ll p$.
2. Generate a random, non-singular, $n \times n$ binary block circulant matrix $\mathbf{Q}$ with row weight $m \ll n$.
3. $SK = \{\mathbf{H}, \mathbf{Q}\}$

LEDApkc - Key generation

Secret key

1. Generate a random $p \times n$ binary block circulant matrix $\mathbf{H} = [\mathbf{H}_0, \ldots, \mathbf{H}_{n_0-1}]$ with column weight $d_v \ll p$.
2. Generate a random, non-singular, $n \times n$ binary block circulant matrix $\mathbf{Q}$ with row weight $m \ll n$.
3. $SK = \{\mathbf{H}, \mathbf{Q}\}$

Public key

1. Compute $\mathbf{L} = \mathbf{H} \cdot \mathbf{Q} = [\mathbf{L}_0, \ldots, \mathbf{L}_{n_0-1}]$.
2. Compute $\mathbf{M} = (\mathbf{L}_{n_0-1})^{-1} \cdot \mathbf{L} = [\mathbf{M}_l, \mathbf{I}_p]$.
3. $PK = \{\mathbf{M}_l\}$

LEDApkc - Encryption

1. Alice gets Bob's public key $\mathbf{M}_l$.
2. She generates a random length-n error vector $\mathbf{e}$ with weight t.
3. She encrypts any length-k block $\mathbf{u}$ as

$$\begin{aligned}\mathbf{x} &= \mathbf{u} \cdot \left[\mathbf{I}_{(n_0-1)p}, \mathbf{M}_l^T\right] + \mathbf{e} = \\ &= \mathbf{u} \cdot \mathbf{G}' + \mathbf{e}\end{aligned}$$

LEDApkc - Encryption

1. Alice gets Bob's public key $\mathbf{M}_l$.
2. She generates a random length-n error vector $\mathbf{e}$ with weight t.
3. She encrypts any length-k block $\mathbf{u}$ as

$$\begin{aligned}\mathbf{x} &= \mathbf{u} \cdot \left[\mathbf{I}_{(n_0-1)p}, \mathbf{M}_l^T\right] + \mathbf{e} = \\ &= \mathbf{u} \cdot \mathbf{G}' + \mathbf{e}\end{aligned}$$

CCA2 conversion

The use of a proper conversion is necessary to achieve indistinguishability under adaptive chosen cyphertext attack (IND-CCA2) security.

- K. Kobara, H. Imai, "Semantically secure McEliece public-key cryptosystems — conversions for McEliece PKC," PKC 2001, vol. 1992 of Springer LNCS, pp. 19–35, 2001.

LEDApkc - Decryption

1. Bob computes

$$\begin{aligned}\mathbf{s} = \mathbf{x} \cdot \mathbf{L}^T &= \\ &= \left(\mathbf{u} \cdot \mathbf{G}' + \mathbf{e}\right) \cdot \mathbf{L}^T = \\ &= \mathbf{e} \cdot \mathbf{L}^T\end{aligned}$$

2. Bob applies Q-decoding on $\mathbf{s}$ and obtains $\mathbf{e}$.
3. Bob computes $\mathbf{x} + \mathbf{e} = \mathbf{u} \cdot [\mathbf{I}_k, \mathbf{M}_l]$, and looks at the first k bits to recover the plaintext.

LEDAkem

- LEDAkem shares the same secret/public code structure of LEDApkc, but is built upon the Niederreiter framework.
- Ephemeral keys are used.
- The system achieves Indistinguishability under Chosen Plaintext Attack (IND-CPA).
- Since LEDApkc and LEDAkem are built upon the same code, they are equivalent from the security point of view.

- M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini: "LEDAkem: first round submission to the NIST post-quantum cryptography call," November 2017.
- M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDAkem: a post-quantum key encapsulation mechanism based on QC-LDPC codes," CoRR, vol. abs/1801.08867, 2018.

Information set decoding attacks

- Recovering the error vector used by Alice results in solving a syndrome decoding problem (SDP) instance.
- The best SDP solvers are information set decoding (ISD) algorithms: given a code with length n and dimension k, searching for an error weight with weight t requires a complexity $\mathtt{C}_{\mathtt{ISD}}(n, k, t)$.
- Modern ISD algorithms are based on the fact that the general decoding problem can be related to the one of finding low weight-codewords in a code.

- E. Prange, "The use of information sets in decoding cyclic codes," Information Theory, IRE Transactions on, vol. 8, no. 5, pp. 5–9, 1962.
- J. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," IEEE Trans. Inform. Theory, vol. 34, no. 5, pp. 1354–1359, 1988.
- A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," Advances in Cryptology - EUROCRYPT 2012, vol. 7237 of Springer LNCS, pp. 520–536, 2012.

Information set decoding attacks

- The public code admits $\mathbf{L}$, whose rows have weight $\leq n_0 m d_v$, as parity check matrix.
- An ISD algorithm might be used to search for rows of $\mathbf{L}$ in the dual of the public code.

Information set decoding attacks

- The public code admits **L**, whose rows have weight $\leq n_0 m d_v$, as parity check matrix.
- An ISD algorithm might be used to search for rows of **L** in the dual of the public code.

Work Factor of ISD attacks

$$\mathtt{WF}_{\mathtt{DA}} = \frac{\mathtt{C}_{\mathtt{ISD}}(n,k,t)}{\sqrt{p}}, \quad \mathtt{WF}_{\mathtt{KRA}} = \frac{\mathtt{C}_{\mathtt{ISD}}(n,n-k,n_0 m d_v)}{p}$$

- N. Sendrier, "Decoding one out of many," in Proc. PQCrypto 2011, vol. 7071 of Springer LNCS, pp. 51–67, 2011.
- D. J. Bernstein, "Grover vs. McEliece," in Proc. PQCrypto 2010, vol. 6061 of Springer LNCS, pp. 73–80, 2010.
- S.H.S. de Vries, "Achieving 128-bit Security against Quantum Attacks in OpenVPN," Master Thesis, University of Twente, 2016.

Reaction attacks

- The DFR depends on the number of overlapping ones between the error vector and the rows of $\mathbf{H}$ and $\mathbf{Q}$.
- The opponent can produce cyphertexts and send them to Bob; the analysis of Bob's decoding failures reveal information about the distances among the ones in the secret key.

- Q. Guo, T. Johansson, P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors," Advances in Cryptology ASIACRYPT 2016, vol. 10031 of Springer LNCS, pp. 789–815.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem," PQCrypto 2017, vol. 10346 of Springer LNCS, pp. 51–68.

Reaction attacks

- The DFR depends on the number of overlapping ones between the error vector and the rows of $\mathbf{H}$ and $\mathbf{Q}$.
- The opponent can produce cyphertexts and send them to Bob; the analysis of Bob's decoding failures reveal information about the distances among the ones in the secret key.
- The minimum number of observed decoding failures, in order to make the attack successful, is in the order of 10^5 or more.

- Q. Guo, T. Johansson, P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors," Advances in Cryptology ASIACRYPT 2016, vol. 10031 of Springer LNCS, pp. 789–815.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem," PQCrypto 2017, vol. 10346 of Springer LNCS, pp. 51–68.

Avoiding reaction attacks

- Reaction attacks can be avoided by bounding the lifetime M of a key-pair (M corresponds to the number of cyphertexts encrypted/decrypted with the same key-pair):
  - $M = 1$ for LEDAkem (ephemeral keys);
  - $M = 10^4 \cdot DFR^{-1}$ for LEDApkc.

Avoiding reaction attacks

- Reaction attacks can be avoided by bounding the lifetime M of a key-pair (M corresponds to the number of cyphertexts encrypted/decrypted with the same key-pair):
  - $M = 1$ for LEDAkem (ephemeral keys);
  - $M = 10^4 \cdot DFR^{-1}$ for LEDApkc.
- A new reaction attack on LEDApkc has been recently proposed:
  1. the opponent builds candidates for $\mathbf{Q}^T$;
  2. a set of candidates for $\mathbf{G} = \mathbf{G}' \cdot \mathbf{Q}^T$ is efficiently computed;
  3. an ISD algorithm is applied on each candidate to search for rows of $\mathbf{H}$.

T. Fabsic, V. Hromada, and P. Zajac, "A reaction attack on LEDApkc," Cryptology ePrint Archive, Report 2018/140, 2018, https://eprint.iacr.org/2018/140.

Avoiding reaction attacks

- The Work Factor of this attack can be estimated as

$$\mathtt{WF}_{\mathtt{FHZ}} \geq 2^{n_0\left[n_0 - n^{(1)}\right]} \cdot p^{n_0^2 - n_0} \cdot C_{\mathtt{ISD}}\left(n, n-k, n_0 d_v\right)$$

  with $n^{(1)}$ being the number of weight-1 blocks in a row of $\mathbf{Q}$.
- A proper parameters choice guarantees that $\mathtt{WF}_{\mathtt{FHZ}}$ is above the target security level.

Avoiding reaction attacks

- The Work Factor of this attack can be estimated as

$$\mathtt{WF}_{\mathtt{FHZ}} \geq 2^{n_0\left[n_0 - n^{(1)}\right]} \cdot p^{n_0^2 - n_0} \cdot C_{\mathtt{ISD}}\left(n, n-k, n_0 d_v\right)$$

  with $n^{(1)}$ being the number of weight-1 blocks in a row of $\mathbf{Q}$.
- A proper parameters choice guarantees that $\mathtt{WF}_{\mathtt{FHZ}}$ is above the target security level.

Conservative lifetime of a key-pair

All reaction attacks can be avoided by choosing $M = DFR^{-1}$.

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector to be found.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector to be found.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

- The rows of $\mathbf{Q}^T$ are sparse ($\mathrm{wt}(\mathbf{q}_i) = m \ll n$), hence their supports are (almost) disjoint.

Rationale of the Q-decoder

- Decoding is performed on the syndrome

$$\mathbf{s} = \mathbf{e} \cdot \mathbf{L}^T = \mathbf{e} \cdot \mathbf{Q}^T \cdot \mathbf{H}^T = \mathbf{e}' \cdot \mathbf{H}^T$$

  where $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}^T$ is the expanded error vector to be found.
- Let $\phi(\mathbf{e})$ denote the support of $\mathbf{e}$ and $\mathbf{q}_j$ be the j-th row of $\mathbf{Q}^T$, then

$$\mathbf{e}' = \sum_{j \in \phi(\mathbf{e})} \mathbf{q}_j$$

- The rows of $\mathbf{Q}^T$ are sparse ($\mathrm{wt}(\mathbf{q}_i) = m \ll n$), hence their supports are (almost) disjoint.
- Also $\mathbf{e}$ is sparse ($\mathrm{wt}(\mathbf{e}) = t \ll n$), hence

$$\mathrm{wt}(\mathbf{e}') \approx mt$$

Rationale of the Q-decoder

- Let us consider the (integer) inner product $\rho = \mathbf{e}' * \mathbf{q}_v$:
  - if $v \notin \phi(\mathbf{e})$, then the supports of $\mathbf{e}'$ and $\mathbf{q}_v$ have a small intersection and ρ is small;
  - if $v \in \phi(\mathbf{e})$, then $\mathbf{q}_v$ is one of the rows forming $\mathbf{e}'$, hence ρ is large.
- As in BF decoding, an estimate of $\mathbf{e}'$ is obtained by computing the (integer) inner product between the syndrome and each column of $\mathbf{H}$

$$\mathbf{\Sigma} = \mathbf{s} * \mathbf{H}$$

  and thresholding the vector $\mathbf{\Sigma}$.
- So we can estimate $\phi(\mathbf{e})$ by replacing $\mathbf{e}'$ with $\mathbf{\Sigma}$ to compute

$$\mathbf{R} = [\rho_0, \rho_1, \cdots, \rho_{n-1}] = \mathbf{\Sigma} * \mathbf{Q}$$

  and thresholding the vector $\mathbf{R}$.

Algorithmic procedure

Initialization

$\mathbf{s}^{(0)} = \mathbf{x} \cdot \mathbf{L}^T, \mathbf{e}^{(0)} = \mathbf{0}$

Description of the j-th iteration

Input: $\mathbf{e}^{(j-1)}, \mathbf{s}^{(j-1)}$

1. Compute $\mathbf{\Sigma} = [\sigma_0, \sigma_1, \cdots, \sigma_{n-1}] = \mathbf{s}^{(j-1)} * \mathbf{H}$.
2. Compute $\mathbf{R} = [\rho_0, \rho_1, \cdots, \rho_{n-1}] = \mathbf{\Sigma} * \mathbf{Q}$.
3. Compute $\Psi = \left\{ i \,\middle|\, \rho_i \geq b^{(j)} \right\}$.
4. Update the error vector as $\mathbf{e}^{(j)} = \mathbf{e}^{(j-1)} + \mathbf{1}_\Psi$.
5. Update the syndrome as $\mathbf{s}^{(j)} = \mathbf{s}^{(j-1)} + \sum_{i \in \Psi} \mathbf{q}_i \cdot \mathbf{H}^T$.

Output: $\mathbf{e}^{(j)}, \mathbf{s}^{(j)}$

Statistical determination of the flipping thresholds

We define the following probabilities:

$$p_{ci}(t) = \sum_{j=0,\ j \text{ odd}}^{\min[n_0 d_v - 1, mt]} \frac{\binom{n_0 d_v - 1}{j}\binom{n - n_0 d_v}{mt - j}}{\binom{n-1}{mt}}$$

$$p_{ic}(t) = \sum_{j=0,\ j \text{ even}}^{\min[n_0 d_v - 1, mt-1]} \frac{\binom{n_0 d_v - 1}{j}\binom{n - n_0 d_v}{mt - j - 1}}{\binom{n-1}{mt-1}}$$

where:

- $p_{ci}(t)$ is the probability that a codeword bit is error-free and a parity-check equation evaluates it wrongly;
- $p_{ic}(t)$ is the probability that a codeword bit is in error and a parity-check equation evaluates it correctly.

Statistical determination of the flipping thresholds

- We consider the i-th bit and define the following probability

$$P\left\{e_i = 1|\rho_i\right\} = \left(1 + \frac{P\left\{e_i = 0, \rho_i\right\}}{P\left\{e_i = 1, \rho_i\right\}}\right)^{-1} = $$
$$= \frac{1}{1 + \frac{n-t}{t}\left[\frac{p_{ci}(t)}{p_{ic}(t)}\right]^{\rho_i}\left[\frac{1-p_{ci}(t)}{1-p_{ic}(t)}\right]^{md_v - \rho_i}}$$

- We define a margin $\Delta \geq 0$, such that

$$P\left\{e_i = 1|\rho_i\right\} > (1+\Delta)P\left\{e_i = 0|\rho_i\right\}$$

- Increasing Δ increases the average number of iterations as well, but lowers the DFR.

Statistical determination of the flipping thresholds

- The optimal threshold value is chosen as

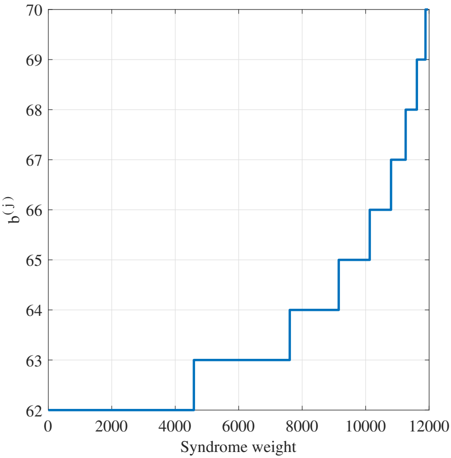
$$b = \min\left\{\rho_i \in [0; md_v],\ \ \text{s.t. } P\left\{e_i = 1|\rho_i\right\} > \frac{1+\Delta}{2+\Delta}\right\}$$

- The average syndrome weight can be related to the weight of the error vector

$$E\left[\mathrm{wt}(\mathbf{s})\right] = \left[p_{ic}(t) + p_{ci}(t)\right] p$$

Statistical determination of the flipping thresholds

Flipping thresholds rule for the instance with $\lambda = 128$, $n_0 = 2$.



Statistical determination of the flipping thresholds

- The approach used for the determination of the thresholds is only based on statistical arguments, and can also be applied to a bit flipping (BF) decoder.
- The thresholds are precomputed and given as input to the decoder, in the form of a look-up table with few entries.
- The threshold values change throughout the iterations, depending on the observed syndrome weights.

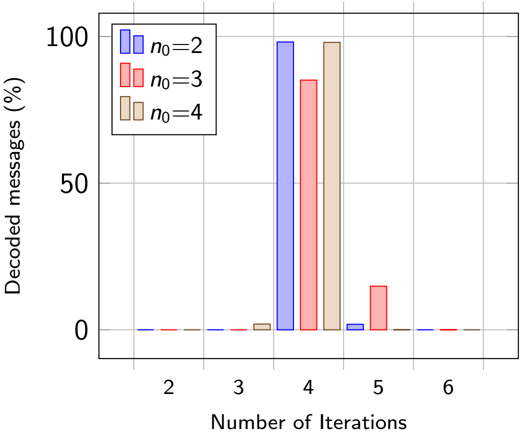
Proposed parameters set

Proposed parameters sets for the NIST competition; the savings in the public key size are computed with respect to the case of a bit flipping (BF) decoder.

λ	$\mathbf{n_0}$	**p**	$\mathbf{d_v}$	**m**	**t**	**DFR**	**PK reduction**
128	2	27,779	17	7	224	$\approx 8.3 \cdot 10^{-9}$	$\approx 47\%$
	3	18,701	19	7	141	$\lesssim 10^{-9}$	$\approx 56\%$
	4	17,027	21	7	112	$\lesssim 10^{-9}$	$\approx 57\%$
192	2	57,557	17	11	349	$8 \cdot \lesssim 10^{-8}$	$\approx 63\%$
	3	41,507	19	11	220	$8 \cdot \lesssim 10^{-8}$	$\approx 64\%$
	4	35,027	17	13	175	$8 \cdot \lesssim 10^{-8}$	$\approx 76\%$
256	2	99,053	19	13	474	$\lesssim 10^{-8}$	$\approx 63\%$
	3	72,019	19	15	301	$\lesssim 10^{-8}$	$\approx 75\%$
	4	60,509	23	13	239	$\lesssim 10^{-8}$	$\approx 70\%$

Number of iterations

Percentage of decoded messages as a function of the number of iterations, for the proposed instances with $\lambda = 128$, in the case of $\Delta = 0.3$.



Room for improvements

- Improve the decoding procedure, in order to achieve smaller public key sizes.
- Exploit the structure of both $\mathbf{H}$ and $\mathbf{Q}$ in order to avoid reaction attacks:
  - proper parameters sets and decoding procedures might prevent known reactions attacks.
- Define an upper bound and/or a closed form expression for the DFR.

Thanks for the attention