QC-LDPC codes, QC-MDPC codes and their use in post-quantum cryptography

Marco Baldi

Università Politecnica delle Marche
Ancona, Italy
m.baldi@univpm.it

Carleton Finite Fields eSeminar

July 29, 2020

LDPC codes

- Low-Density Parity-Check (LDPC) codes are state-of-art forward error correcting (FEC) codes.
- Introduced by Gallager in 1962 and more recently rediscovered.
- Able to approach the channel capacity under belief propagation decoding.
- Nowadays included in many applications and standards.

- R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.
- D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in Cryptography and Coding. 5th IMA Conference, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Berlin: Springer, 1995, no. 1025, pp. 100–111.
- C. Sae-Young, G. Forney, T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," IEEE Commun. Lett., vol. 5, no. 2, pp. 58–60, Feb. 2001.

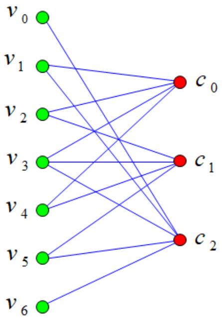
Code representation

- Binary linear block code with
  - n: code length
  - k: code dimension
  - $r = n - k$: code redundancy

Parity-check matrix

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Tanner graph



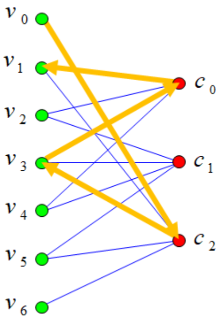
Encoding and decoding of LDPC codes

- Encoding through classical methods (e.g. generator matrix **G**).
- Efficient decoding through iterative algorithms working on the code parity-check matrix/Tanner graph.
- Soft-decision decoders:
  - Sum-product algorithm with log-likelihood ratios (LLR-SPA)
  - Min-sum algorithm and its variants (offset, weighted...)
- Hard-decision decoders:
  - Gallager's A/B algorithm
  - Bit-flipping algorithm
  - Their variants

Encoding and decoding of LDPC codes

- Encoding through classical methods (e.g. generator matrix **G**).
- Efficient decoding through iterative algorithms working on the code parity-check matrix/Tanner graph.

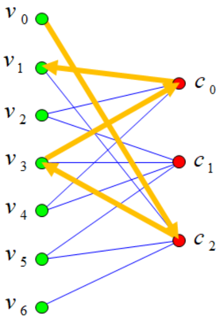
- Soft-decision decoders:
  - Sum-product algorithm with log-likelihood ratios (LLR-SPA)
  - Min-sum algorithm and its variants (offset, weighted...)
- Hard-decision decoders:
  - Gallager's A/B algorithm
  - Bit-flipping algorithm
  - Their variants



Encoding and decoding of LDPC codes

- Encoding through classical methods (e.g. generator matrix **G**).
- Efficient decoding through iterative algorithms working on the code parity-check matrix/Tanner graph.

- Soft-decision decoders:
  - Sum-product algorithm with log-likelihood ratios (LLR-SPA)
  - Min-sum algorithm and its variants (offset, weighted...)
- Hard-decision decoders:
  - Gallager's A/B algorithm
  - Bit-flipping algorithm
  - Their variants

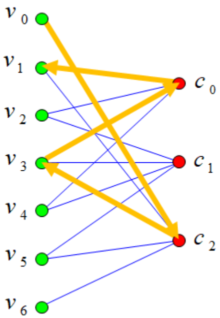


Encoding and decoding of LDPC codes

- Encoding through classical methods (e.g. generator matrix **G**).
- Efficient decoding through iterative algorithms working on the code parity-check matrix/Tanner graph.

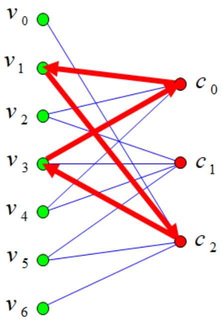
- Soft-decision decoders:
  - Sum-product algorithm with log-likelihood ratios (LLR-SPA)
  - Min-sum algorithm and its variants (offset, weighted...)

- Hard-decision decoders:
  - Gallager's A/B algorithm
  - Bit-flipping algorithm
  - Their variants



Properties affecting iterative decoding

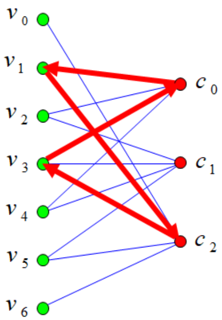
Closed loops in the Tanner graph contaminate iterative decoders' information with correlation.



- Thus LDPC codes usually have:
  - Low density of ones in the parity check matrix
  - Few edges in the Tanner graph
  - No more than one overlapping one between any two rows/columns
  - Local cycles in the Tanner graph as long as possible
- These requirements result in LDPC codes with:
  - very small Hamming weight of the rows of $\mathbf{H}$ ($d_c \approx \log n$)
  - very long length ($n \gg 1000$)

Properties affecting iterative decoding

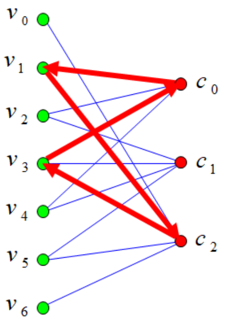
Closed loops in the Tanner graph contaminate iterative decoders' information with correlation.



- Thus LDPC codes usually have:
  - Low density of ones in the parity check matrix
  - Few edges in the Tanner graph
  - No more than one overlapping one between any two rows/columns
  - Local cycles in the Tanner graph as long as possible
- These requirements result in LDPC codes with:
  - very small Hamming weight of the rows of $\mathbf{H}$ ($d_c \approx \log n$)
  - very long length ($n \gg 1000$)

Properties affecting iterative decoding

Closed loops in the Tanner graph contaminate iterative decoders' information with correlation.



- Thus LDPC codes usually have:
  - Low density of ones in the parity check matrix
  - Few edges in the Tanner graph
  - No more than one overlapping one between any two rows/columns
  - Local cycles in the Tanner graph as long as possible
- These requirements result in LDPC codes with:
  - very small Hamming weight of the rows of $\mathbf{H}$ ($d_c \approx \log n$)
  - very long length ($n \gg 1000$)

Hard-decision decoding: Gallager's A/B

Gallager's A decoding:

1. Variable nodes send their initial value (0/1) to their neighboring check nodes.
2. Each check node c sends back to each variable node v the binary sum of all values received from its neighbours except v (marginalization).
3. Each variable node v counts the number of values received from its neighboring check nodes that disagree with its own value.
4. For each check node c, if all neighboring check nodes other than c (marginalization) disagree with the value of v, then v sends its flipped value to c, otherwise it sends its original value to c.
5. Decoding iterates from step 2, unless all parity checks are satisfied or a maximum number of iterations is reached.

Gallager's B decoding:

- Steps 1, 2, 3 and 5 like Gallager's A.
- In step 4, the value of v is flipped if the number of disagreeing check nodes except c (marginalization) exceeds a given threshold b.

Hard-decision decoding: Gallager's A/B

Gallager's A decoding:

1. Variable nodes send their initial value (0/1) to their neighboring check nodes.
2. Each check node c sends back to each variable node v the binary sum of all values received from its neighbours except v (marginalization).
3. Each variable node v counts the number of values received from its neighboring check nodes that disagree with its own value.
4. For each check node c, if all neighboring check nodes other than c (marginalization) disagree with the value of v, then v sends its flipped value to c, otherwise it sends its original value to c.
5. Decoding iterates from step 2, unless all parity checks are satisfied or a maximum number of iterations is reached.

Gallager's B decoding:

- Steps 1, 2, 3 and 5 like Gallager's A.
- In step 4, the value of v is flipped if the number of disagreeing check nodes except c (marginalization) exceeds a given threshold b.

Bit flipping decoding

- Similar to Gallager's B algorithm, but without marginalization.
- From [Gallager1962]:

`The decoder computes all the parity checks and then changes any digit that is contained in more than some fixed number of unsatisfied parity-check equations. Using these new values, the parity checks are recomputed, and the process is repeated until the parity checks are all satisfied.`

▶ R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. 8, pp. 21–28, 1962.

Bit flipping decoding performance

- The decoding radius of LDPC codes under BF decoding cannot be determined analytically through closed form expressions.
- However, the average BF decoder performance can be estimated through a probabilistic model (under some assumptions).
- It allows computing a threshold for the number of errors for which BF converges to the right codeword in asymptotic conditions.
- It can be adapted to modeling the BF decoder performance in code-based cryptosystems.

▶ M. Baldi, QC-LDPC Code-Based Cryptography, SpringerBriefs in Electrical and Computer Engineering, Springer, 2014.

Bit flipping decoding performance

- The decoding radius of LDPC codes under BF decoding cannot be determined analytically through closed form expressions.
- However, the average BF decoder performance can be estimated through a probabilistic model (under some assumptions).
- It allows computing a threshold for the number of errors for which BF converges to the right codeword in asymptotic conditions.
- It can be adapted to modeling the BF decoder performance in code-based cryptosystems.

▶ M. Baldi, QC-LDPC Code-Based Cryptography, SpringerBriefs in Electrical and Computer Engineering, Springer, 2014.

Bit flipping decoding performance

- The decoding radius of LDPC codes under BF decoding cannot be determined analytically through closed form expressions.
- However, the average BF decoder performance can be estimated through a probabilistic model (under some assumptions).
- It allows computing a threshold for the number of errors for which BF converges to the right codeword in asymptotic conditions.
- It can be adapted to modeling the BF decoder performance in code-based cryptosystems.

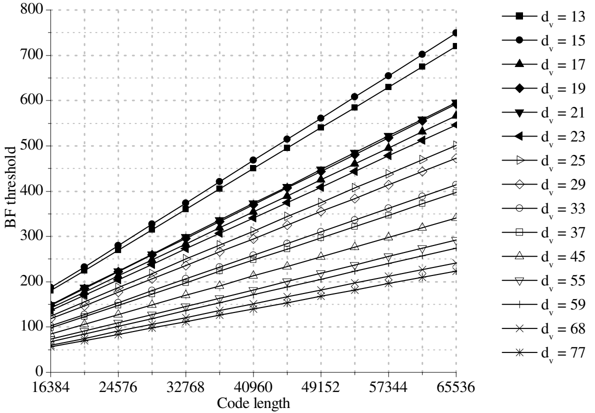
► M. Baldi, QC-LDPC Code-Based Cryptography, SpringerBriefs in Electrical and Computer Engineering, Springer, 2014.

Bit flipping decoding performance

- The decoding radius of LDPC codes under BF decoding cannot be determined analytically through closed form expressions.
- However, the average BF decoder performance can be estimated through a probabilistic model (under some assumptions).
- It allows computing a threshold for the number of errors for which BF converges to the right codeword in asymptotic conditions.
- It can be adapted to modeling the BF decoder performance in code-based cryptosystems.

► M. Baldi, QC-LDPC Code-Based Cryptography, SpringerBriefs in Electrical and Computer Engineering, Springer, 2014.

Bit flipping decoding performance - examples



BF decoding thresholds versus code length (n) for LDPC codes with code rate 3/4 and several parity-check matrix column weights (d_v).

Moderate-density parity-check (MDPC) codes

- Special case of LDPC codes with density larger than usual ($d_c \approx \sqrt{n}$).
- Mostly used in code-based cryptosystems.
- The density of their parity-check matrices/Tanner graphs does not allow avoiding short cycles.
- However, they can still be decoded through iterative decoders.

- S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," arXiv eprint 0911.3262, 2009.
- R. Misoczki, J. P. Tillich, N. Sendrier and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," Proc. IEEE ISIT 2013, Istanbul, Turkey, pp. 2069–2073.

Moderate-density parity-check (MDPC) codes

- Special case of LDPC codes with density larger than usual ($d_c \approx \sqrt{n}$).
- Mostly used in code-based cryptosystems.
- The density of their parity-check matrices/Tanner graphs does not allow avoiding short cycles.
- However, they can still be decoded through iterative decoders.

- S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," arXiv eprint 0911.3262, 2009.
- R. Misoczki, J. P. Tillich, N. Sendrier and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," Proc. IEEE ISIT 2013, Istanbul, Turkey, pp. 2069–2073.

Moderate-density parity-check (MDPC) codes

- Special case of LDPC codes with density larger than usual ($d_c \approx \sqrt{n}$).
- Mostly used in code-based cryptosystems.
- The density of their parity-check matrices/Tanner graphs does not allow avoiding short cycles.
- However, they can still be decoded through iterative decoders.

- S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," arXiv eprint 0911.3262, 2009.
- R. Misoczki, J. P. Tillich, N. Sendrier and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," Proc. IEEE ISIT 2013, Istanbul, Turkey, pp. 2069–2073.

Moderate-density parity-check (MDPC) codes

- Special case of LDPC codes with density larger than usual ($d_c \approx \sqrt{n}$).
- Mostly used in code-based cryptosystems.
- The density of their parity-check matrices/Tanner graphs does not allow avoiding short cycles.
- However, they can still be decoded through iterative decoders.

- S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," arXiv eprint 0911.3262, 2009.
- R. Misoczki, J. P. Tillich, N. Sendrier and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," Proc. IEEE ISIT 2013, Istanbul, Turkey, pp. 2069–2073.

QC-LDPC codes

- A linear block code is a quasi-cyclic (QC) code if:
  - its dimension and length are multiple of an integer p ($k = k_0 p, n = n_0 p$),
  - every cyclic shift of a codeword by n_0 positions yields another codeword.
- The generator and parity-check matrices of a QC code can assume two forms:
  - Circulant of blocks.
  - Block of circulants.

Advantage

The QC structure allows to represent the generator and parity-check matrices in a compact way (each circulant is completely described by its first row).

QC-LDPC codes

- A linear block code is a QC code if:
  - its dimension and length are multiple of an integer p ($k = k_0 p, n = n_0 p$),
  - every cyclic shift of a codeword by n_0 positions yields another codeword.
- The generator and parity-check matrices of a QC code can assume two forms:
  - Circulant of blocks.
  - Block of circulants.

Advantage

The QC structure allows to represent the generator and parity-check matrices in a compact way (each circulant is completely described by its first row).

Example of QC-(almost)LDPC code

$$
\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}\right]
$$

- Number of ciculant blocks: $n_0 = 2$.
- Code rate: $R = \frac{n_0 - 1}{n_0} = 1/2$.
- Parity-check matrix column weight: $d_v = 3$.
- Parity-check matrix row weight: $d_c = n_0 d_v = 6$.

QC-LDPC and QC-MDPC codes

$$\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc}
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & & & & \vdots & & & & & & & & & & & \vdots & & & & &
\end{array}\right]$$

- The parity-check matrix is described by its first row.
- The storage size increases linearly in the code length.
- The code length is usually very large ($10'000 \lessapprox n \lessapprox 100'000$)
- QC-LDPC codes usually have $d_c \approx \log n$.
- QC-MDPC codes usually have $d_c \approx \sqrt{n}$.

QC-LDPC and QC-MDPC codes

$$\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc}
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & & & & \vdots & & & & & & & & & & & \vdots & & & & &
\end{array}\right]$$

- The parity-check matrix is described by its first row.
- The storage size increases linearly in the code length.
- The code length is usually very large ($10'000 \lessapprox n \lessapprox 100'000$)
- QC-LDPC codes usually have $d_c \approx \log n$.
- QC-MDPC codes usually have $d_c \approx \sqrt{n}$.

QC-LDPC and QC-MDPC codes

$$\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc}
\mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & & & & & \vdots & & & & & & & & & & & \vdots & & & & &
\end{array}\right]$$

- The parity-check matrix is described by its first row.
- The storage size increases linearly in the code length.
- The code length is usually very large ($10'000 \lessapprox n \lessapprox 100'000$)
- QC-LDPC codes usually have $d_c \approx \log n$.
- QC-MDPC codes usually have $d_c \approx \sqrt{n}$.

QC-LDPC and QC-MDPC codes

$$\mathbf{H} = \left[\begin{array}{ccccccccccc|ccccccccccc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & & & & & \vdots & & & & & & & & & & & \vdots & & & & & \end{array}\right]$$

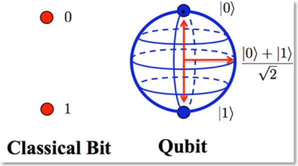
- The parity-check matrix is described by its first row.
- The storage size increases linearly in the code length.
- The code length is usually very large ($10'000 \lessapprox n \lessapprox 100'000$)
- QC-LDPC codes usually have $d_c \approx \log n$.
- QC-MDPC codes usually have $d_c \approx \sqrt{n}$.

Quantum computing

- Computer using quantum mechanics phenomena, such as quantum superposition and quantum correlation for performing calculations.
- Theorized by Richard Feynman and Yuri Manin in the early 1980s.
- Shor's algorithm (1994):
  - factorizes integers on a quantum computer,
  - given an integer N, it factors it in a time polynomial in $\log(N)$,
  - on a classic computer the time is exponential in N.
- Grover's algorithm (1996):
  - performs a search in an unordered list on a quantum computer,
  - it finds an entry in a list of N in a time proportional to $\sqrt{N}$,
  - on a classic computer the time is proportional to N.

Quantum computing

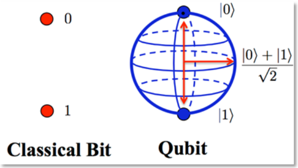
- Computer using quantum mechanics phenomena, such as quantum superposition and quantum correlation for performing calculations.
- Theorized by Richard Feynman and Yuri Manin in the early 1980s.



- Shor's algorithm (1994):
  - factorizes integers on a quantum computer,
  - given an integer N, it factors it in a time polynomial in $\log(N)$,
  - on a classic computer the time is exponential in N.
- Grover's algorithm (1996):
  - performs a search in an unordered list on a quantum computer,
  - it finds an entry in a list of N in a time proportional to $\sqrt{N}$,
  - on a classic computer the time is proportional to N.

Quantum computing

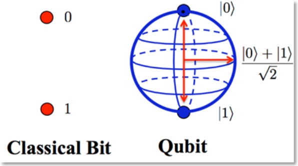
- Computer using quantum mechanics phenomena, such as quantum superposition and quantum correlation for performing calculations.
- Theorized by Richard Feynman and Yuri Manin in the early 1980s.



- Shor's algorithm (1994):
  - factorizes integers on a quantum computer,
  - given an integer N, it factors it in a time polynomial in $\log(N)$,
  - on a classic computer the time is exponential in N.
- Grover's algorithm (1996):
  - performs a search in an unordered list on a quantum computer,
  - it finds an entry in a list of N in a time proportional to $\sqrt{N}$,
  - on a classic computer the time is proportional to N.

Quantum computing

- Computer using quantum mechanics phenomena, such as quantum superposition and quantum correlation for performing calculations.
- Theorized by Richard Feynman and Yuri Manin in the early 1980s.



- Shor's algorithm (1994):
  - factorizes integers on a quantum computer,
  - given an integer N, it factors it in a time polynomial in $\log(N)$,
  - on a classic computer the time is exponential in N.
- Grover's algorithm (1996):
  - performs a search in an unordered list on a quantum computer,
  - it finds an entry in a list of N in a time proportional to $\sqrt{N}$,
  - on a classic computer the time is proportional to N.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).
- It exploits quantum superposition.
- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).

- It exploits quantum superposition.
- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).
- It exploits quantum superposition.

- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).
- It exploits quantum superposition.
- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).
- It exploits quantum superposition.

- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Towards practical quantum computers (3)

- On January 2019 IBM announced Q System One, the first commercial quantum computer.
- It has 20 qubits (50 qubits are deemed necessary to compete with classic computers).
- It exploits quantum superposition.
- It must be kept at a very low temperature and isolated from any form of electromagnetic noise.
- Quantum equivalent of the first computers of the 1950s and 1960s.
- Simulators and software models available for programming.

Quantum supremacy

- Google and IBM are competing towards achieving quantum supremacy.
- The 72-qubit system that Google was developing in 2017 proved too difficult to control.
- Google then started the development of a 53-qubit system called Sycamore.
- In October 2019, Google claimed that the Sycamore processor was able to perform a calculation in 200 seconds that would have taken the world's most powerful supercomputer 10,000 years.
- IBM disclaimed this, stating that Google's system is specialized to solve a single problem, differently from IBM's general-purpose quantum computer.

F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.

Quantum supremacy

- Google and IBM are competing towards achieving quantum supremacy.
- The 72-qubit system that Google was developing in 2017 proved too difficult to control.
- Google then started the development of a 53-qubit system called Sycamore.
- In October 2019, Google claimed that the Sycamore processor was able to perform a calculation in 200 seconds that would have taken the world's most powerful supercomputer 10,000 years.
- IBM disclaimed this, stating that Google's system is specialized to solve a single problem, differently from IBM's general-purpose quantum computer.

F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.

Quantum supremacy

- Google and IBM are competing towards achieving quantum supremacy.
- The 72-qubit system that Google was developing in 2017 proved too difficult to control.
- Google then started the development of a 53-qubit system called Sycamore.
- In October 2019, Google claimed that the Sycamore processor was able to perform a calculation in 200 seconds that would have taken the world's most powerful supercomputer 10,000 years.
- IBM disclaimed this, stating that Google's system is specialized to solve a single problem, differently from IBM's general-purpose quantum computer.

F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.

Quantum supremacy

- Google and IBM are competing towards achieving quantum supremacy.
- The 72-qubit system that Google was developing in 2017 proved too difficult to control.
- Google then started the development of a 53-qubit system called Sycamore.
- In October 2019, Google claimed that the Sycamore processor was able to perform a calculation in 200 seconds that would have taken the world's most powerful supercomputer 10,000 years.
- IBM disclaimed this, stating that Google's system is specialized to solve a single problem, differently from IBM's general-purpose quantum computer.

F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.

Quantum supremacy

- Google and IBM are competing towards achieving quantum supremacy.
- The 72-qubit system that Google was developing in 2017 proved too difficult to control.
- Google then started the development of a 53-qubit system called Sycamore.
- In October 2019, Google claimed that the Sycamore processor was able to perform a calculation in 200 seconds that would have taken the world's most powerful supercomputer 10,000 years.
- IBM disclaimed this, stating that Google's system is specialized to solve a single problem, differently from IBM's general-purpose quantum computer.

▶ F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505–510, 2019.

Quantum-vulnerable cryptography

The most widespread cryptographic systems today are based on mathematical problems that can be solved with Shor's algorithm:

- **RSA**: public key cryptosystem based on integer factorization (used in SSL/TLS, online banking, ATM, ...).
- **ElGamal**: public key cryptosystem based on discrete logarithm (used in SSL/TLS, ...).
- **DSA**: digital signature algorithm based on discrete logarithm (used in SSL/TLS, ...).
- **Diffie-Hellman**: key exchange protocol based on discrete logarithm (used in SSL/TLS, NFC, contactless payments, ...).
- **ECDH**: Elliptic-curve Diffie–Hellman, used for end-to-end encryption (Signal, WhatsApp, Facebook Messenger, Skype, ...).
- **ECDSA**: Elliptic-curve digital signature algorithm (used in Bitcoin (secp256k1), Ethereum, ...).

Post-quantum cryptography

Asymmetric schemes:

- Based on lattices
- Based on codes
- Based on multivariate polynomials
- Based on hash functions
- Others (isogenies ...)

Symmetric schemes:

- Symmetric encryption schemes (AES ...)
- Hash functions (SHA ...)
- Can still be used as long as Grover's algorithm is taken into account

NIST PQcrypto Project

- **NIST** has initiated a process for the development and standardization of one or more public-key cryptographic algorithms to enrich:

  - Recommendation FIPS 186-4 (Digital Signature Standard - DSS)
  - Special publication SP 800-56A Rev 2 (key establishment systems based on discrete logarithm)
  - Special publication SP 800-56B (key establishment systems based on integer factorization)

NIST PQcrypto call timeline

- **24-26 February 2016**: Announcement and description of the NIST call.
- **28 April 2016**: NISTIR 8105 report on post-quantum cryptography released.
- **20 December 2016**: Official publication of the call.
- **30 November 2017**: Deadline for submission of candidates (82 submissions).
- **30 January 2019**: Second round admission announced (26 candidates).
- **22 July 2020**: Third round admission announced (7 finalists + 8 alternate candidates).

Security level goals

NIST target (for categories 1, 3, and 5)

Computational effort required on either a classical or a quantum computer to break the AES with a key size of λ bits, $\lambda \in \{128, 192, 256\}$, through an exhaustive key search.

- On a classical computer we have complexity $2^{\lambda} C_{AES}$, where C_{AES} is the binary cost of AES.
- The quantum cost can be estimated taking into account Grover's algorithm and counting the strictly needed Clifford and T gates (which are the most expensive).

NIST Category	AES Key Size (bits)	Classical Cost (binary operations)	Quantum Cost (quantum gates)
1	128	$2^{128} \cdot 2^{14} \cdot 3 = 2^{143.5}$	$1.16 \cdot 2^{81}$
3	192	$2^{192} \cdot 2^{14} \cdot 4 = 2^{208}$	$1.33 \cdot 2^{113}$
5	256	$2^{256} \cdot 2^{14} \cdot 5 = 2^{272.3}$	$1.57 \cdot 2^{145}$

- R. Ueno, S. Morioka, N. Homma, and T. Aoki. A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths - Toward Efficient CBC-Mode Implementation. In B. Gierlichs and A. Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016*, vol. 9813 of LNCS, pages 538–558. Springer, 2016.
- M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In T. Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, vol. 9606 of *LNCS*, pages 29–43. Springer, 2016.

Security level goals

NIST target (for categories 1, 3, and 5)

Computational effort required on either a classical or a quantum computer to break the AES with a key size of λ bits, $\lambda \in \{128, 192, 256\}$, through an exhaustive key search.

- On a classical computer we have complexity $2^{\lambda}\mathtt{C}_{\mathtt{AES}}$, where $\mathtt{C}_{\mathtt{AES}}$ is the binary cost of AES.
- The quantum cost can be estimated taking into account Grover's algorithm and counting the strictly needed Clifford and T gates (which are the most expensive).

NIST Category	AES Key Size (bits)	Classical Cost (binary operations)	Quantum Cost (quantum gates)
1	128	$2^{128} \cdot 2^{14} \cdot 3 = 2^{143.5}$	$1.16 \cdot 2^{81}$
3	192	$2^{192} \cdot 2^{14} \cdot 4 = 2^{208}$	$1.33 \cdot 2^{113}$
5	256	$2^{256} \cdot 2^{14} \cdot 5 = 2^{272.3}$	$1.57 \cdot 2^{145}$

- R. Ueno, S. Morioka, N. Homma, and T. Aoki. A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths - Toward Efficient CBC-Mode Implementation. In B. Gierlichs and A. Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016*, vol. 9813 of LNCS, pages 538–558. Springer, 2016.
- M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In T. Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, vol. 9606 of *LNCS*, pages 29–43. Springer, 2016.

Security level goals

NIST target (for categories 1, 3, and 5)

Computational effort required on either a classical or a quantum computer to break the AES with a key size of λ bits, $\lambda \in \{128, 192, 256\}$, through an exhaustive key search.

- On a classical computer we have complexity $2^{\lambda}\mathtt{C}_{\mathtt{AES}}$, where $\mathtt{C}_{\mathtt{AES}}$ is the binary cost of AES.
- The quantum cost can be estimated taking into account Grover's algorithm and counting the strictly needed Clifford and T gates (which are the most expensive).

NIST Category	AES Key Size (bits)	Classical Cost (binary operations)	Quantum Cost (quantum gates)
1	128	$2^{128} \cdot 2^{14} \cdot 3 = 2^{143.5}$	$1.16 \cdot 2^{81}$
3	192	$2^{192} \cdot 2^{14} \cdot 4 = 2^{208}$	$1.33 \cdot 2^{113}$
5	256	$2^{256} \cdot 2^{14} \cdot 5 = 2^{272.3}$	$1.57 \cdot 2^{145}$

- R. Ueno, S. Morioka, N. Homma, and T. Aoki. A High Throughput/Gate AES Hardware Architecture by Compressing Encryption and Decryption Datapaths - Toward Efficient CBC-Mode Implementation. In B. Gierlichs and A. Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016*, vol. 9813 of LNCS, pages 538–558. Springer, 2016.
- M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In T. Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, vol. 9606 of *LNCS*, pages 29–43. Springer, 2016.

McEliece cryptosystem

- Proposed by Robert McEliece in 1978.
- Irreducible Goppa codes were used in the original proposal.
- Secret irreducible Goppa code:
  - irreducible polynomial of degree t over $GF(2^m)$,
  - length (maximum): $n = 2^m$,
  - dimension: $k \geq n - t \cdot m$,
  - correction capability: t errors.

Robert J. McEliece
(May 21, 1942 – May 8, 2019)

Rationale

1. The number of irreducible polynomials of degree t over $GF(n)$ is $\approx n^t/t$.
2. The probability that a random polynomial is irreducible is $\approx 1/t$, and a fast algorithm exists for testing irreducibility.

R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.

McEliece cryptosystem

- Proposed by Robert McEliece in 1978.
- Irreducible Goppa codes were used in the original proposal.
- Secret irreducible Goppa code:
    - irreducible polynomial of degree t over $GF(2^m)$,
    - length (maximum): $n = 2^m$,
    - dimension: $k \geq n - t \cdot m$,
    - correction capability: t errors.

Robert J. McEliece
(May 21, 1942 – May 8, 2019)

Rationale

1. The number of irreducible polynomials of degree t over $GF(n)$ is $\approx n^t/t$.
2. The probability that a random polynomial is irreducible is $\approx 1/t$, and a fast algorithm exists for testing irreducibility.

► R. McEliece, "Public-Key System Based on Algebraic Coding Theory," DSN Progress Report 44, pp. 114–116, 1978.

McEliece cryptosystem - key generation

Private key

- $k \times n$ generator matrix $\mathbf{G}$ of a secret Goppa code,
- random dense $k \times k$ non-singular "scrambling" matrix $\mathbf{S}$,
- random $n \times n$ permutation matrix $\mathbf{P}$.

Public key

$$\mathbf{G}' = \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{P}$$

- The public code is permutation equivalent to the secret code.
- Security relies on the hardness of decoding a random-like code.

▶ E. Berlekamp, R. McEliece and H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Trans. Inf. Theory, vol. 24, no. 3, pp. 384–386, May 1978.

McEliece cryptosystem - key generation

Private key

- $k \times n$ generator matrix $\mathbf{G}$ of a secret Goppa code,
- random dense $k \times k$ non-singular "scrambling" matrix $\mathbf{S}$,
- random $n \times n$ permutation matrix $\mathbf{P}$.

Public key

$$\mathbf{G}' = \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{P}$$

- The public code is permutation equivalent to the secret code.
- Security relies on the hardness of decoding a random-like code.

E. Berlekamp, R. McEliece and H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Trans. Inf. Theory, vol. 24, no. 3, pp. 384–386, May 1978.

McEliece cryptosystem - key generation

Private key

- $k \times n$ generator matrix $\mathbf{G}$ of a secret Goppa code,
- random dense $k \times k$ non-singular "scrambling" matrix $\mathbf{S}$,
- random $n \times n$ permutation matrix $\mathbf{P}$.

Public key

$$\mathbf{G}' = \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{P}$$

- The public code is permutation equivalent to the secret code.
- Security relies on the hardness of decoding a random-like code.

▶ E. Berlekamp, R. McEliece and H. van Tilborg, "On the inherent intractability of certain coding problems," IEEE Trans. Inf. Theory, vol. 24, no. 3, pp. 384–386, May 1978.

McEliece cryptosystem - encryption

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

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}' + \mathbf{e} = \mathbf{c} + \mathbf{e}$$

Alert

This only provides semantic security!

McEliece cryptosystem - encryption

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

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}' + \mathbf{e} = \mathbf{c} + \mathbf{e}$$

Alert

This only provides semantic security!

McEliece cryptosystem - decryption

1. Bob computes

$$\begin{aligned} \mathbf{x}' &= \mathbf{x} \cdot \mathbf{P}^{-1} = \\ &= (\mathbf{u} \cdot \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{P} + \mathbf{e}) \cdot \mathbf{P}^{-1} = \\ &= \mathbf{u} \cdot \mathbf{S} \cdot \mathbf{G} + \mathbf{e} \cdot \mathbf{P}^{-1} \end{aligned}$$

2. Bob decodes the secret code and obtains

$$\mathbf{u}' = \mathbf{u} \cdot \mathbf{S}$$

3. Bob computes $\mathbf{u} = \mathbf{u}' \cdot \mathbf{S}^{-1}$.

Niederreiter cryptosystem - key generation

Private key

- $r \times n$ parity-check matrix $\mathbf{H}$ of a secret code,
- random dense $r \times r$ non-singular "scrambling" matrix $\mathbf{S}$.

Public key

$$\mathbf{H}' = \mathbf{S} \cdot \mathbf{H}$$

H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, 15(2):159–166, 1986.

Niederreiter cryptosystem - key generation

Private key

- $r \times n$ parity-check matrix $\mathbf{H}$ of a secret code,
- random dense $r \times r$ non-singular "scrambling" matrix $\mathbf{S}$.

Public key

$$\mathbf{H}' = \mathbf{S} \cdot \mathbf{H}$$

- H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," Problems of Control and Information Theory, 15(2):159–166, 1986.

Niederreiter cryptosystem - encryption

1. Alice gets Bob's public key $\mathbf{H}'$.
2. She maps each block of the secret message into an error pattern $\mathbf{e}$ with length n and weight t.
3. She encrypts $\mathbf{e}$ as

$$\mathbf{x} = \mathbf{H}' \cdot \mathbf{e}^T = \mathbf{S} \cdot \mathbf{H} \cdot \mathbf{e}^T$$

Alert

We still only have semantic security!

Niederreiter cryptosystem - encryption

1. Alice gets Bob's public key $\mathbf{H}'$.
2. She maps each block of the secret message into an error pattern $\mathbf{e}$ with length n and weight t.
3. She encrypts $\mathbf{e}$ as

$$\mathbf{x} = \mathbf{H}' \cdot \mathbf{e}^T = \mathbf{S} \cdot \mathbf{H} \cdot \mathbf{e}^T$$

Alert

We still only have semantic security!

Niederreiter cryptosystem - decryption

1. Bob computes
$$\mathbf{x}' = \mathbf{S}^{-1} \cdot \mathbf{x} = \mathbf{H} \cdot \mathbf{e}^T$$
2. Bob performs syndrome decoding of the secret code and obtains $\mathbf{e}$ from $\mathbf{x}'$.
3. He demaps $\mathbf{e}$ into the corresponding secret message block.

McEliece/Niederreiter cryptosystems

- GRS codes originally used in Niederreiter were attacked.
- But Goppa codes resisted cryptanalysis for more than 40 years.
- These systems are faster than competing solutions...
- ...but they require large public keys (56 KiB or more for 80-bit security).
- Attacks based on distinguishers pose some threats on high rate Goppa codes.
- They also invalidate all existing McEliece cryptosystem security proofs for high rate Goppa codes.

- D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography*, vol. 5299 of Springer LNCS, pp. 31–46, 2008.
- J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," In Proc. Information Theory Workshop 2011, pp. 282–286, Paraty, Brasil, 2011.

McEliece/Niederreiter cryptosystems

- GRS codes originally used in Niederreiter were attacked.
- But Goppa codes resisted cryptanalysis for more than 40 years.
- These systems are faster than competing solutions...
- ...but they require large public keys (56 KiB or more for 80-bit security).
- Attacks based on distinguishers pose some threats on high rate Goppa codes.
- They also invalidate all existing McEliece cryptosystem security proofs for high rate Goppa codes.

- D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography*, vol. 5299 of Springer LNCS, pp. 31–46, 2008.
- J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," In Proc. Information Theory Workshop 2011, pp. 282–286, Paraty, Brasil, 2011.

McEliece/Niederreiter cryptosystems

- GRS codes originally used in Niederreiter were attacked.
- But Goppa codes resisted cryptanalysis for more than 40 years.
- These systems are faster than competing solutions...
- ...but they require large public keys (56 KiB or more for 80-bit security).
- Attacks based on distinguishers pose some threats on high rate Goppa codes.
- They also invalidate all existing McEliece cryptosystem security proofs for high rate Goppa codes.

- D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography*, vol. 5299 of Springer LNCS, pp. 31–46, 2008.
- J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," In Proc. Information Theory Workshop 2011, pp. 282–286, Paraty, Brasil, 2011.

McEliece/Niederreiter cryptosystems

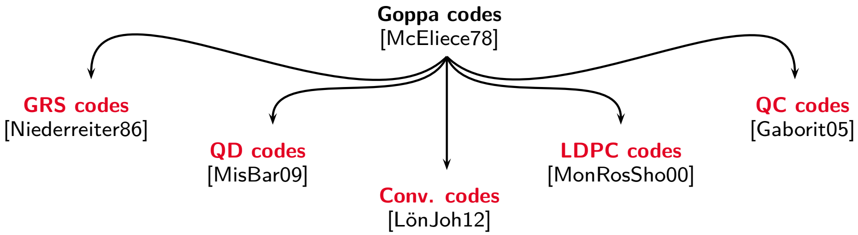
- GRS codes originally used in Niederreiter were attacked.
- But Goppa codes resisted cryptanalysis for more than 40 years.
- These systems are faster than competing solutions...
- ...but they require large public keys (56 KiB or more for 80-bit security).
- Attacks based on distinguishers pose some threats on high rate Goppa codes.
- They also invalidate all existing McEliece cryptosystem security proofs for high rate Goppa codes.

- D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography*, vol. 5299 of Springer LNCS, pp. 31–46, 2008.
- J.-C. Faugère, V. Gauthier, A. Otmani, L. Perret, and J.-P. Tillich, "A distinguisher for high rate McEliece cryptosystems," In Proc. Information Theory Workshop 2011, pp. 282–286, Paraty, Brasil, 2011.

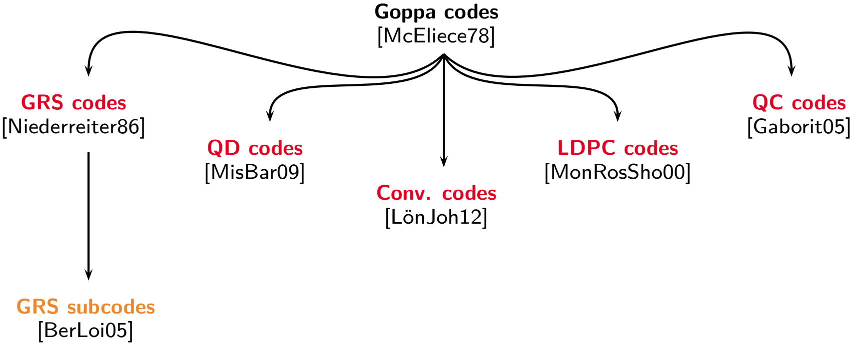
Alternatives to Goppa codes (Hamming metric)

Goppa codes
[McEliece78]

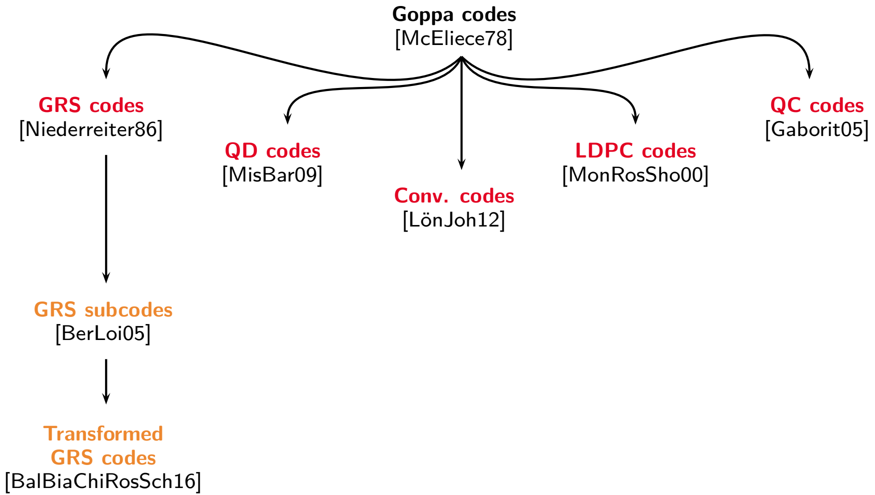
Alternatives to Goppa codes (Hamming metric)



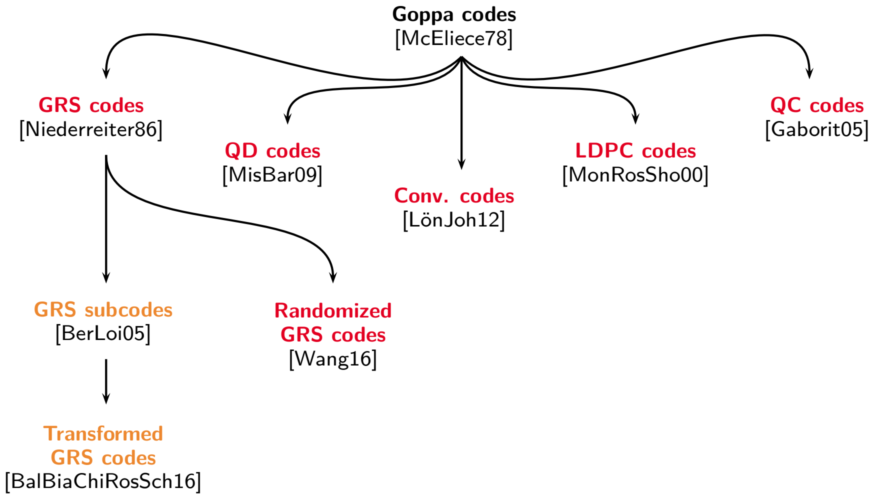
Alternatives to Goppa codes (Hamming metric)



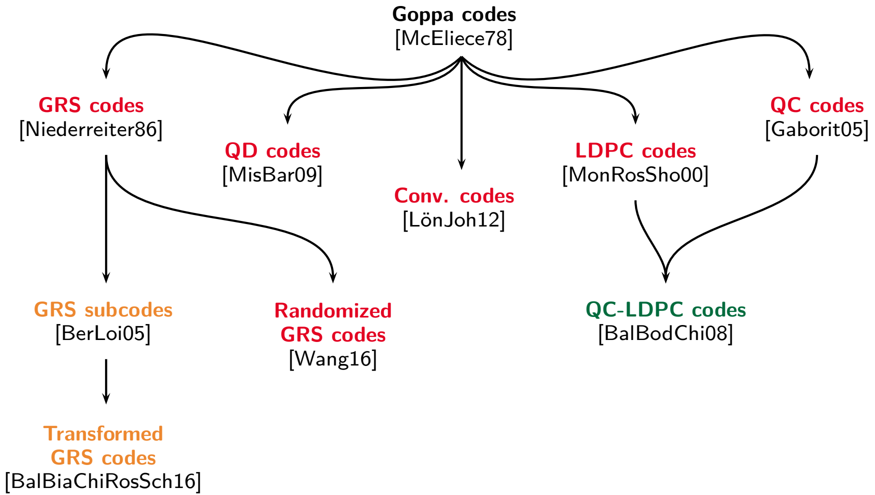
Alternatives to Goppa codes (Hamming metric)



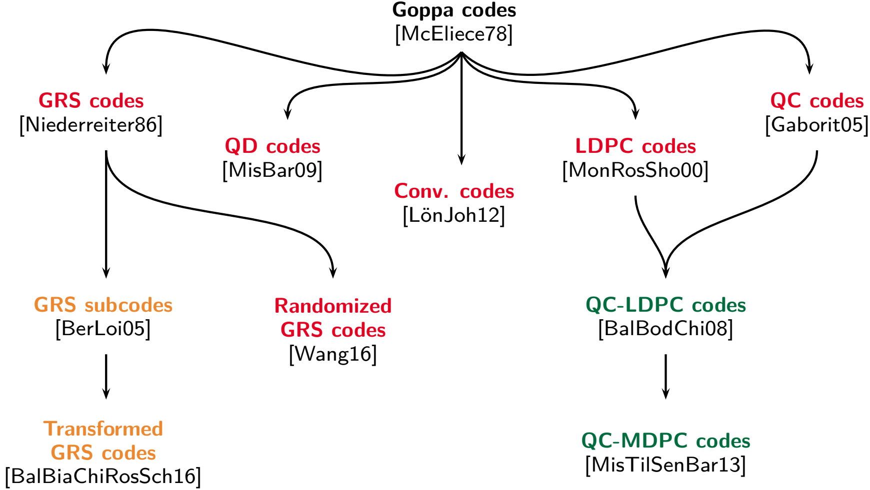
Alternatives to Goppa codes (Hamming metric)



Alternatives to Goppa codes (Hamming metric)



Alternatives to Goppa codes (Hamming metric)



QC-LDPC code-based cryptosystems

- QC-LDPC codes bring important advantages in the framework of McEliece/Niederreiter cryptosystems:
  - The sparsity of their matrices enables very efficient decoding.
  - Quasi-cyclicity enables very compact keys.
- quasi-cyclic low-density parity-check (QC-LDPC) code-based systems introduced in 2008.
- quasi-cyclic moderate-density parity-check (QC-MDPC) code-based variants introduced in 2013.

- M. Baldi, M. Bodrato, F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes", Proc. SCN 2008, vol. 5229 of LNCS, pp. 246–262, 2008.
- R. Misoczki, J.-P. Tillich, N. Sendrier, P.S.L.M. Barreto, "MDPC-McEliece: new McEliece variants from moderate density parity-check codes", Proc. IEEE ISIT 2013, pp. 2069–2073, July 2013.

QC-LDPC code-based cryptosystems

- QC-LDPC codes bring important advantages in the framework of McEliece/Niederreiter cryptosystems:
  - The sparsity of their matrices enables very efficient decoding.
  - Quasi-cyclicity enables very compact keys.
- QC-LDPC code-based systems introduced in 2008.
- QC-MDPC code-based variants introduced in 2013.

- M. Baldi, M. Bodrato, F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes", Proc. SCN 2008, vol. 5229 of LNCS, pp. 246–262, 2008.
- R. Misoczki, J.-P. Tillich, N. Sendrier, P.S.L.M. Barreto, "MDPC-McEliece: new McEliece variants from moderate density parity-check codes", Proc. IEEE ISIT 2013, pp. 2069–2073, July 2013.

QC-LDPC code-based cryptosystems

- QC-LDPC codes bring important advantages in the framework of McEliece/Niederreiter cryptosystems:
  - The sparsity of their matrices enables very efficient decoding.
  - Quasi-cyclicity enables very compact keys.
- QC-LDPC code-based systems introduced in 2008.
- QC-MDPC code-based variants introduced in 2013.

- M. Baldi, M. Bodrato, F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes", Proc. SCN 2008, vol. 5229 of LNCS, pp. 246–262, 2008.
- R. Misoczki, J.-P. Tillich, N. Sendrier, P.S.L.M. Barreto, "MDPC-McEliece: new McEliece variants from moderate density parity-check codes", Proc. IEEE ISIT 2013, pp. 2069–2073, July 2013.

Private QC-LDPC code

The private code is a QC-LDPC code with:

- rate $R = \frac{n_0-1}{n_0}$ (with $n_0 = 2, 3, 4$),
- redundancy r (in the order of some thousands),
- length $n = n_0 \cdot r$,
- dimension $k = (n_0 - 1) \cdot r$.

Secret QC-LDPC matrix

$$\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1 | \ldots | \mathbf{H}_{n_0-1}]$$

- Each $\mathbf{H}_i$ is an $r \times r$ circulant matrix.
- Prime values for r must be chosen to avoid folding attacks.

▶ M. Koochak Shooshtari, M. Ahmadian-Attari, T. Johansson and M. Reza Aref, "Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes," in IET Information Security, vol. 10, no. 4, pp. 194-202, 2016.

Private QC-LDPC code

The private code is a QC-LDPC code with:

- rate $R = \frac{n_0 - 1}{n_0}$ (with $n_0 = 2, 3, 4$),
- redundancy r (in the order of some thousands),
- length $n = n_0 \cdot r$,
- dimension $k = (n_0 - 1) \cdot r$.

Secret QC-LDPC matrix

$$\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1 | \dots | \mathbf{H}_{n_0 - 1}]$$

- Each $\mathbf{H}_i$ is an $r \times r$ circulant matrix.
- Prime values for r must be chosen to avoid folding attacks.

M. Koochak Shooshtari, M. Ahmadian-Attari, T. Johansson and M. Reza Aref, "Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes," in IET Information Security, vol. 10, no. 4, pp. 194-202, 2016.

Private QC-LDPC code

The private code is a QC-LDPC code with:

- rate $R = \frac{n_0 - 1}{n_0}$ (with $n_0 = 2, 3, 4$),
- redundancy r (in the order of some thousands),
- length $n = n_0 \cdot r$,
- dimension $k = (n_0 - 1) \cdot r$.

Secret QC-LDPC matrix

$$\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1 | \dots | \mathbf{H}_{n_0 - 1}]$$

- Each $\mathbf{H}_i$ is an $r \times r$ circulant matrix.
- Prime values for r must be chosen to avoid folding attacks.

▶ M. Koochak Shooshtari, M. Ahmadian-Attari, T. Johansson and M. Reza Aref, "Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes," in IET Information Security, vol. 10, no. 4, pp. 194-202, 2016.

Private QC-MDPC code

- Pick an $n \times n$ sparse matrix $\mathbf{Q}$ with average row/column weight $1 \leq m \ll n$.
- $\mathbf{Q}$ is in QC form, i.e., formed by $n_0 \times n_0$ circulant blocks.

Secret QC-MDPC matrix

$$\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q} = \left[\mathbf{H}'_0 | \mathbf{H}'_1 | \ldots | \mathbf{H}'_{n_0-1}\right]$$

- $\mathbf{H}'$ has density m times greater than $\mathbf{H}$, thus it describes a QC-MDPC code.
- QC-LDPC code-based systems pick a random $\mathbf{H}$, a random $\mathbf{Q}$ and $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}$ as private key.
- QC-MDPC code-based systems pick a random $\mathbf{H}'$ (i.e. $\mathbf{H} = \mathbf{H}'$ and $\mathbf{Q} = \mathbf{I}$) as private key.

Private QC-MDPC code

- Pick an $n \times n$ sparse matrix $\mathbf{Q}$ with average row/column weight $1 \leq m \ll n$.
- $\mathbf{Q}$ is in QC form, i.e., formed by $n_0 \times n_0$ circulant blocks.

Secret QC-MDPC matrix

$$\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q} = \left[\mathbf{H}'_0 | \mathbf{H}'_1 | \ldots | \mathbf{H}'_{n_0-1}\right]$$

- $\mathbf{H}'$ has density m times greater than $\mathbf{H}$, thus it describes a QC-MDPC code.
- QC-LDPC code-based systems pick a random $\mathbf{H}$, a random $\mathbf{Q}$ and $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}$ as private key.
- QC-MDPC code-based systems pick a random $\mathbf{H}'$ (i.e. $\mathbf{H} = \mathbf{H}'$ and $\mathbf{Q} = \mathbf{I}$) as private key.

Private QC-MDPC code

- Pick an $n \times n$ sparse matrix $\mathbf{Q}$ with average row/column weight $1 \leq m \ll n$.
- $\mathbf{Q}$ is in QC form, i.e., formed by $n_0 \times n_0$ circulant blocks.

Secret QC-MDPC matrix

$$\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q} = \left[\mathbf{H}'_0 | \mathbf{H}'_1 | \ldots | \mathbf{H}'_{n_0-1}\right]$$

- $\mathbf{H}'$ has density m times greater than $\mathbf{H}$, thus it describes a QC-MDPC code.
- QC-LDPC code-based systems pick a random $\mathbf{H}$, a random $\mathbf{Q}$ and $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}$ as private key.
- QC-MDPC code-based systems pick a random $\mathbf{H}'$ (i.e. $\mathbf{H} = \mathbf{H}'$ and $\mathbf{Q} = \mathbf{I}$) as private key.

Private QC-MDPC code

- Pick an $n \times n$ sparse matrix $\mathbf{Q}$ with average row/column weight $1 \leq m \ll n$.
- $\mathbf{Q}$ is in QC form, i.e., formed by $n_0 \times n_0$ circulant blocks.

Secret QC-MDPC matrix

$$\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q} = \left[\mathbf{H}'_0 | \mathbf{H}'_1 | \ldots | \mathbf{H}'_{n_0-1}\right]$$

- $\mathbf{H}'$ has density m times greater than $\mathbf{H}$, thus it describes a QC-MDPC code.
- QC-LDPC code-based systems pick a random $\mathbf{H}$, a random $\mathbf{Q}$ and $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}$ as private key.
- QC-MDPC code-based systems pick a random $\mathbf{H}'$ (i.e. $\mathbf{H} = \mathbf{H}'$ and $\mathbf{Q} = \mathbf{I}$) as private key.

Private QC-MDPC code

- Pick an $n \times n$ sparse matrix $\mathbf{Q}$ with average row/column weight $1 \leq m \ll n$.
- $\mathbf{Q}$ is in QC form, i.e., formed by $n_0 \times n_0$ circulant blocks.

Secret QC-MDPC matrix

$$\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q} = \left[\mathbf{H}'_0 | \mathbf{H}'_1 | \ldots | \mathbf{H}'_{n_0-1}\right]$$

- $\mathbf{H}'$ has density m times greater than $\mathbf{H}$, thus it describes a QC-MDPC code.
- QC-LDPC code-based systems pick a random $\mathbf{H}$, a random $\mathbf{Q}$ and $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}$ as private key.
- QC-MDPC code-based systems pick a random $\mathbf{H}'$ (i.e. $\mathbf{H} = \mathbf{H}'$ and $\mathbf{Q} = \mathbf{I}$) as private key.

Public code

Public key

Systematic **G** for the private QC-MDPC code:

$$\mathbf{G} = \left[\mathbf{I} \quad \begin{array}{c} \left(\mathbf{H'}^{-1}_{n_0-1} \cdot \mathbf{H'}_0\right)^T \\ \left(\mathbf{H'}^{-1}_{n_0-1} \cdot \mathbf{H'}_1\right)^T \\ \vdots \\ \left(\mathbf{H'}^{-1}_{n_0-1} \cdot \mathbf{H'}_{n_0-2}\right)^T \end{array} \right]$$

- **G** is dense:
  - It allows deriving dense parity-check matrices, which are unsuitable for iterative decoding.
  - Retrieving a sparse parity-check matrix requires an unfeasible computational effort.
- **G** can be in systematic form (owing to CCA2 secure conversion):
  - Public key size $= (n_0 - 1) \cdot r$ bits.

Public code

Public key

Systematic **G** for the private QC-MDPC code:

$$\mathbf{G} = \left[\mathbf{I} \quad \begin{array}{c} \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_0\right)^T \\ \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_1\right)^T \\ \vdots \\ \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_{n_0-2}\right)^T \end{array} \right]$$

- **G** is dense:
  - It allows deriving dense parity-check matrices, which are unsuitable for iterative decoding.
  - Retrieving a sparse parity-check matrix requires an unfeasible computational effort.
- **G** can be in systematic form (owing to CCA2 secure conversion):
  - Public key size $= (n_0 - 1) \cdot r$ bits.

Public code

Public key

Systematic **G** for the private QC-MDPC code:

$$\mathbf{G} = \left[\mathbf{I} \quad \begin{array}{c} \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_0\right)^T \\ \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_1\right)^T \\ \vdots \\ \left(\mathbf{H}'^{-1}_{n_0-1} \cdot \mathbf{H}'_{n_0-2}\right)^T \end{array} \right]$$

- **G** is dense:
  - It allows deriving dense parity-check matrices, which are unsuitable for iterative decoding.
  - Retrieving a sparse parity-check matrix requires an unfeasible computational effort.
- **G** can be in systematic form (owing to CCA2 secure conversion):
  - Public key size $= (n_0 - 1) \cdot r$ bits.

Encryption

- Alice has to encrypt a k-bit vector $\mathbf{u}$.
- She fetches Bob's public key $\mathbf{G}$.
- She generates a random binary intentional error vector $\mathbf{e}$ with weight t.

Encryption map

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G} + \mathbf{e} = \mathbf{c} + \mathbf{e}$$

- $\mathbf{c}$ is a codeword in the public code.
- Addition is modulo-2.
- Hence, all intentional errors are bit flipping errors.

Encryption

- Alice has to encrypt a k-bit vector $\mathbf{u}$.
- She fetches Bob's public key $\mathbf{G}$.
- She generates a random binary intentional error vector $\mathbf{e}$ with weight t.

Encryption map

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G} + \mathbf{e} = \mathbf{c} + \mathbf{e}$$

- $\mathbf{c}$ is a codeword in the public code.
- Addition is modulo-2.
- Hence, all intentional errors are bit flipping errors.

Encryption

- Alice has to encrypt a k-bit vector $\mathbf{u}$.
- She fetches Bob's public key $\mathbf{G}$.
- She generates a random binary intentional error vector $\mathbf{e}$ with weight t.

Encryption map

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G} + \mathbf{e} = \mathbf{c} + \mathbf{e}$$

- $\mathbf{c}$ is a codeword in the public code.
- Addition is modulo-2.
- Hence, all intentional errors are bit flipping errors.

Decryption

- To recover $\mathbf{u}$ from $\mathbf{x}$, Bob first computes the syndrome $\mathbf{s}$ through the secret QC-MDPC matrix.

Syndrome computation

$$\mathbf{s} = \mathbf{H}' \cdot \mathbf{x}^T = \mathbf{H}' \cdot (\mathbf{c} + \mathbf{e})^T = \mathbf{H}' \cdot \mathbf{e}^T$$

$$\mathbf{s} = (\mathbf{HQ}) \cdot \mathbf{e}^T = \mathbf{H} \cdot (\mathbf{eQ}^T)^T = \mathbf{H} \cdot \mathbf{e}'^T$$

- $\mathbf{e}' = \mathbf{eQ}^T$ is a binary vector with weight $\leq t' = tm$.
- From $\mathbf{s}$ Bob can recover:
  - $\mathbf{e}$ by decoding through the secret QC-MDPC $\mathbf{H}'$, or
  - $\mathbf{e}'$ by decoding through the secret QC-LDPC $\mathbf{H}$ (when $\mathbf{Q} \neq \mathbf{I}$).
- After correcting all intentional errors, Bob easily recovers $\mathbf{u}$.

Decryption

- To recover $\mathbf{u}$ from $\mathbf{x}$, Bob first computes the syndrome $\mathbf{s}$ through the secret QC-MDPC matrix.

Syndrome computation

$$\mathbf{s} = \mathbf{H}' \cdot \mathbf{x}^T = \mathbf{H}' \cdot (\mathbf{c} + \mathbf{e})^T = \mathbf{H}' \cdot \mathbf{e}^T$$

$$\mathbf{s} = (\mathbf{HQ}) \cdot \mathbf{e}^T = \mathbf{H} \cdot (\mathbf{eQ}^T)^T = \mathbf{H} \cdot \mathbf{e}'^T$$

- $\mathbf{e}' = \mathbf{eQ}^T$ is a binary vector with weight $\leq t' = tm$.
- From $\mathbf{s}$ Bob can recover:
  - $\mathbf{e}$ by decoding through the secret QC-MDPC $\mathbf{H}'$, or
  - $\mathbf{e}'$ by decoding through the secret QC-LDPC $\mathbf{H}$ (when $\mathbf{Q} \neq \mathbf{I}$).
- After correcting all intentional errors, Bob easily recovers $\mathbf{u}$.

Decryption

- To recover **u** from **x**, Bob first computes the syndrome **s** through the secret QC-MDPC matrix.

Syndrome computation

$$\mathbf{s} = \mathbf{H}' \cdot \mathbf{x}^T = \mathbf{H}' \cdot (\mathbf{c} + \mathbf{e})^T = \mathbf{H}' \cdot \mathbf{e}^T$$

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

- $\mathbf{e}' = \mathbf{e}\mathbf{Q}^T$ is a binary vector with weight $\leq t' = tm$.
- From **s** Bob can recover:
  - **e** by decoding through the secret QC-MDPC $\mathbf{H}'$, or
  - $\mathbf{e}'$ by decoding through the secret QC-LDPC **H** (when $\mathbf{Q} \neq \mathbf{I}$).
- After correcting all intentional errors, Bob easily recovers **u**.

Decryption

- To recover **u** from **x**, Bob first computes the syndrome **s** through the secret QC-MDPC matrix.

Syndrome computation

$$\mathbf{s} = \mathbf{H}' \cdot \mathbf{x}^T = \mathbf{H}' \cdot (\mathbf{c} + \mathbf{e})^T = \mathbf{H}' \cdot \mathbf{e}^T$$

$$\mathbf{s} = (\mathbf{HQ}) \cdot \mathbf{e}^T = \mathbf{H} \cdot (\mathbf{eQ}^T)^T = \mathbf{H} \cdot \mathbf{e}'^T$$

- $\mathbf{e}' = \mathbf{eQ}^T$ is a binary vector with weight $\leq t' = tm$.
- From **s** Bob can recover:
  - **e** by decoding through the secret QC-MDPC $\mathbf{H}'$, or
  - $\mathbf{e}'$ by decoding through the secret QC-LDPC **H** (when $\mathbf{Q} \neq \mathbf{I}$).
- After correcting all intentional errors, Bob easily recovers **u**.

QC-LDPC/MDPC codes in the NIST contest

- LEDAcrypt (Low-dEnsity parity-check coDe-bAsed cryptographic systems), providing:
  - A Niederreiter-based KEM with IND-CPA and ephemeral keys.
  - A Niederreiter-based KEM with IND-CCA2 and long-term keys.
  - A McEliece-based PKC with IND-CCA2.
- BIKE (Bit Flipping Key Encapsulation), providing:
  - Two McEliece/Niederreiter-based KEMs with IND-CPA and ephemeral keys.

- https://www.ledacrypt.org/
- https://bikesuite.org/

QC-LDPC/MDPC codes in the NIST contest

- LEDAcrypt (Low-dEnsity parity-check coDe-bAsed cryptographic systems), providing:
  - A Niederreiter-based KEM with IND-CPA and ephemeral keys.
  - A Niederreiter-based KEM with IND-CCA2 and long-term keys.
  - A McEliece-based PKC with IND-CCA2.
- BIKE (Bit Flipping Key Encapsulation), providing:
  - Two McEliece/Niederreiter-based KEMs with IND-CPA and ephemeral keys.

- https://www.ledacrypt.org/
- https://bikesuite.org/

Main attacks

Decoding attacks

Aimed at decrypting one or more ciphertexts without knowing the private key.

- An information set decoding (ISD) algorithm can be exploited to perform decoding of the public code.

Key recovery attacks

Aimed at recovering the private key from the public key.

- For any linear code, the rows of the parity-check matrix are codewords of its dual code.
- For QC-LDPC and QC-MDPC codes, these rows have low weight and can be searched through ISD algorithms.
- The quantum speedup due to Grover's algorithm must be taken into account.

▶ 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," in *Advances in Cryptology - EUROCRYPT 2012*, vol. 7237 of Springer LNCS, pp. 520–536, 2012.

Main attacks

Decoding attacks

Aimed at decrypting one or more ciphertexts without knowing the private key.

- An ISD algorithm can be exploited to perform decoding of the public code.

Key recovery attacks

Aimed at recovering the private key from the public key.

- For any linear code, the rows of the parity-check matrix are codewords of its dual code.
- For QC-LDPC and QC-MDPC codes, these rows have low weight and can be searched through ISD algorithms.
- The quantum speedup due to Grover's algorithm must be taken into account.

▶ 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," in *Advances in Cryptology - EUROCRYPT 2012*, vol. 7237 of Springer LNCS, pp. 520–536, 2012.

Main attacks

Decoding attacks

Aimed at decrypting one or more ciphertexts without knowing the private key.

- An ISD algorithm can be exploited to perform decoding of the public code.

Key recovery attacks

Aimed at recovering the private key from the public key.

- For any linear code, the rows of the parity-check matrix are codewords of its dual code.
- For QC-LDPC and QC-MDPC codes, these rows have low weight and can be searched through ISD algorithms.
- The quantum speedup due to Grover's algorithm must be taken into account.

▶ 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," in *Advances in Cryptology - EUROCRYPT 2012*, vol. 7237 of Springer LNCS, pp. 520–536, 2012.

Main attacks

Decoding attacks

Aimed at decrypting one or more ciphertexts without knowing the private key.

- An ISD algorithm can be exploited to perform decoding of the public code.

Key recovery attacks

Aimed at recovering the private key from the public key.

- For any linear code, the rows of the parity-check matrix are codewords of its dual code.
- For QC-LDPC and QC-MDPC codes, these rows have low weight and can be searched through ISD algorithms.
- The quantum speedup due to Grover's algorithm must be taken into account.

► 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," in *Advances in Cryptology - EUROCRYPT 2012*, vol. 7237 of Springer LNCS, pp. 520–536, 2012.

Reaction attacks

Observations

1. Iterative decoding algorithms do not have a deterministic decoding radius, which entails a non-zero decoding failure rate (DFR).
2. The DFR depends on the structure of the secret matrix.
3. Eve can estimate the DFR by observing Bob's reactions.

- In the CPA case, Eve performs many encryptions with suitably chosen error vectors and observes Bob's reactions during decryption.
- In the CCA2 case, the error vectors cannot be chosen by Eve, who must exploit those resulting from each encryption to make her deductions.

Countermeasure

Make the DFR negligible (i.e., $\leq 2^{-\lambda}$).

- Q. Guo, T. Johansson, and P. Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016*, vol. 10031 of *LNCS*, pages 789–815. Springer Berlin Heidelberg, 2016.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson. A reaction attack on the QC-LDPC McEliece cryptosystem. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017*, pages 51–68. Springer, Utrecht, The Netherlands, June 2017.
- T. Fabšič, V. Hromada, and P. Zajac. A reaction attack on LEDApkc. *IACR Cryptology ePrint Archive*, 2018:140, 2018.

Reaction attacks

Observations

1. Iterative decoding algorithms do not have a deterministic decoding radius, which entails a non-zero DFR.
2. The DFR depends on the structure of the secret matrix.
3. Eve can estimate the DFR by observing Bob's reactions.

- In the CPA case, Eve performs many encryptions with suitably chosen error vectors and observes Bob's reactions during decryption.
- In the CCA2 case, the error vectors cannot be chosen by Eve, who must exploit those resulting from each encryption to make her deductions.

Countermeasure

Make the DFR negligible (i.e., $\leq 2^{-\lambda}$).

- Q. Guo, T. Johansson, and P. Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016*, vol. 10031 of *LNCS*, pages 789–815. Springer Berlin Heidelberg, 2016.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson. A reaction attack on the QC-LDPC McEliece cryptosystem. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017*, pages 51–68. Springer, Utrecht, The Netherlands, June 2017.
- T. Fabšič, V. Hromada, and P. Zajac. A reaction attack on LEDApkc. *IACR Cryptology ePrint Archive*, 2018:140, 2018.

Reaction attacks

Observations

1. Iterative decoding algorithms do not have a deterministic decoding radius, which entails a non-zero DFR.
2. The DFR depends on the structure of the secret matrix.
3. Eve can estimate the DFR by observing Bob's reactions.

- In the CPA case, Eve performs many encryptions with suitably chosen error vectors and observes Bob's reactions during decryption.
- In the CCA2 case, the error vectors cannot be chosen by Eve, who must exploit those resulting from each encryption to make her deductions.

Countermeasure

Make the DFR negligible (i.e., $\leq 2^{-\lambda}$).

- Q. Guo, T. Johansson, and P. Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016*, vol. 10031 of *LNCS*, pages 789–815. Springer Berlin Heidelberg, 2016.
- T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson. A reaction attack on the QC-LDPC McEliece cryptosystem. In T. Lange and T. Takagi, editors, *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017*, pages 51–68. Springer, Utrecht, The Netherlands, June 2017.
- T. Fabšič, V. Hromada, and P. Zajac. A reaction attack on LEDApkc. *IACR Cryptology ePrint Archive*, 2018:140, 2018.

CCA security and δ-correctness

- QC-LDPC and QC-MDPC code-based cryptosystems alone provide semantic security only.
- According to [HHK2017], a KEM can be built having IND-CCA2 reduced to the OW-CPA security of the underlying deterministic public key cryptosystem.
- It is required that the decryption failure probability of the OW-CPA scheme is $\delta \leq 2^{-\lambda}$.

δ-correctness

Probability that a (possibly unbounded) adversary is able to induce a decryption failure on a valid ciphertext, taken as the **maximum** over all possible plaintexts, and averaged over all the keypairs.

DFR

Decoding failure probability of a given code (i.e., keypair) **averaged** over all the possible plaintexts (i.e., error vectors).

▶ D. Hofheinz, K. H ovelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017.

CCA security and δ-correctness

- QC-LDPC and QC-MDPC code-based cryptosystems alone provide semantic security only.
- According to [HHK2017], a KEM can be built having IND-CCA2 reduced to the OW-CPA security of the underlying deterministic public key cryptosystem.
- It is required that the decryption failure probability of the OW-CPA scheme is $\delta \leq 2^{-\lambda}$.

δ-correctness

Probability that a (possibly unbounded) adversary is able to induce a decryption failure on a valid ciphertext, taken as the **maximum** over all possible plaintexts, and averaged over all the keypairs.

DFR

Decoding failure probability of a given code (i.e., keypair) **averaged** over all the possible plaintexts (i.e., error vectors).

▶ D. Hofheinz, K. H ovelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017.

CCA security and δ-correctness

- QC-LDPC and QC-MDPC code-based cryptosystems alone provide semantic security only.
- According to [HHK2017], a KEM can be built having IND-CCA2 reduced to the OW-CPA security of the underlying deterministic public key cryptosystem.
- It is required that the decryption failure probability of the OW-CPA scheme is $\delta \leq 2^{-\lambda}$.

δ-correctness

Probability that a (possibly unbounded) adversary is able to induce a decryption failure on a valid ciphertext, taken as the **maximum** over all possible plaintexts, and averaged over all the keypairs.

DFR

Decoding failure probability of a given code (i.e., keypair) **averaged** over all the possible plaintexts (i.e., error vectors).

- D. Hofheinz, K. H ovelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017.

CCA security and δ-correctness

- QC-LDPC and QC-MDPC code-based cryptosystems alone provide semantic security only.
- According to [HHK2017], a KEM can be built having IND-CCA2 reduced to the OW-CPA security of the underlying deterministic public key cryptosystem.
- It is required that the decryption failure probability of the OW-CPA scheme is $\delta \leq 2^{-\lambda}$.

δ-correctness

Probability that a (possibly unbounded) adversary is able to induce a decryption failure on a valid ciphertext, taken as the **maximum** over all possible plaintexts, and averaged over all the keypairs.

DFR

Decoding failure probability of a given code (i.e., keypair) **averaged** over all the possible plaintexts (i.e., error vectors).

- D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017.

CCA security and δ-correctness

- QC-LDPC and QC-MDPC code-based cryptosystems alone provide semantic security only.
- According to [HHK2017], a KEM can be built having IND-CCA2 reduced to the OW-CPA security of the underlying deterministic public key cryptosystem.
- It is required that the decryption failure probability of the OW-CPA scheme is $\delta \leq 2^{-\lambda}$.

δ-correctness

Probability that a (possibly unbounded) adversary is able to induce a decryption failure on a valid ciphertext, taken as the **maximum** over all possible plaintexts, and averaged over all the keypairs.

DFR

Decoding failure probability of a given code (i.e., keypair) **averaged** over all the possible plaintexts (i.e., error vectors).

- D. Hofheinz, K. H ovelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017.

The DFR problem

Issue 1

Iterative decoders are algorithmic $\Rightarrow$ no closed form formula for their error correction capability.

Issue 2

Mathematical models of iterative decoding algorithms work under some idealistic assumptions (e.g., i.i.d. variables).

Issue 3

Performance curves may be simulated (Monte Carlo) down to DFR $\approx 10^{-9}$.

The DFR problem

Issue 1

Iterative decoders are algorithmic ⇒ no closed form formula for their error correction capability.

Issue 2

Mathematical models of iterative decoding algorithms work under some idealistic assumptions (e.g., i.i.d. variables).

Issue 3

Performance curves may be simulated (Monte Carlo) down to DFR $\approx 10^{-9}$.

The DFR problem

Issue 1

Iterative decoders are algorithmic $\Rightarrow$ no closed form formula for their error correction capability.

Issue 2

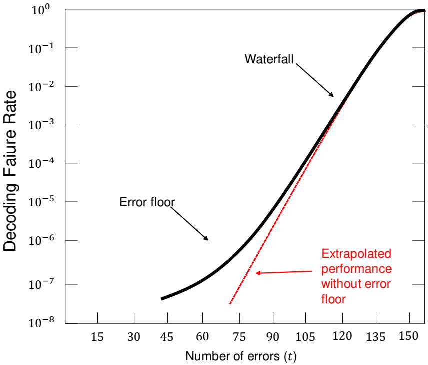
Mathematical models of iterative decoding algorithms work under some idealistic assumptions (e.g., i.i.d. variables).

Issue 3

Performance curves may be simulated (Monte Carlo) down to DFR $\approx 10^{-9}$.

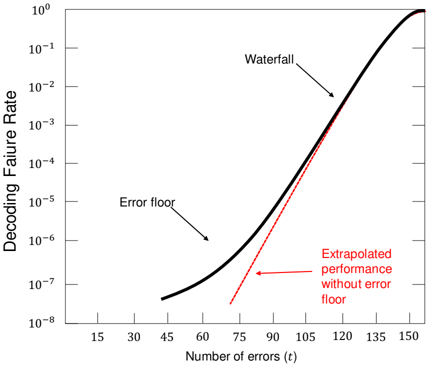
DFR extrapolation?

- One way to extend DFR curves is by extrapolation.
- This is the approach followed in BIKE.
- However, the curve slope may change (error floor) due to:
  - code structural properties,
  - code representation properties,
  - decoding algorithm properties.
- Hence, performance curves can hardly be extrapolated outside the simulated region.



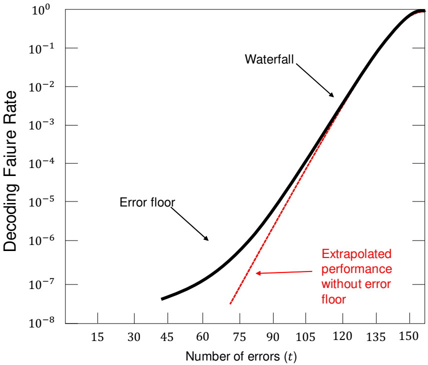
DFR extrapolation?

- One way to extend DFR curves is by extrapolation.
- This is the approach followed in BIKE.
- However, the curve slope may change (error floor) due to:
  - code structural properties,
  - code representation properties,
  - decoding algorithm properties.
- Hence, performance curves can hardly be extrapolated outside the simulated region.



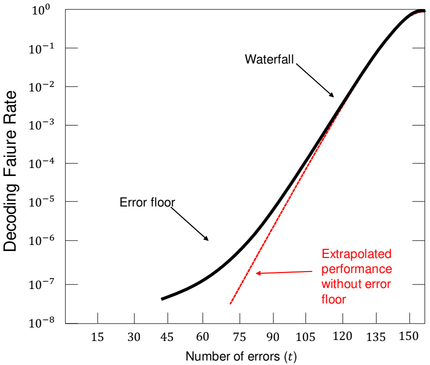
DFR extrapolation?

- One way to extend DFR curves is by extrapolation.
- This is the approach followed in BIKE.
- However, the curve slope may change (error floor) due to:
    - code structural properties,
    - code representation properties,
    - decoding algorithm properties.
- Hence, performance curves can hardly be extrapolated outside the simulated region.



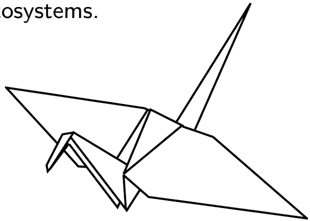
DFR extrapolation?

- One way to extend DFR curves is by extrapolation.
- This is the approach followed in BIKE.
- However, the curve slope may change (error floor) due to:
  - code structural properties,
  - code representation properties,
  - decoding algorithm properties.
- Hence, performance curves can hardly be extrapolated outside the simulated region.



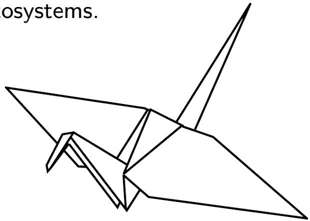
LEDAcrypt

- Suite of low-density parity-check code-based cryptosystems.
- Among the 26 second round candidates of the NIST pqcrypto competition.
- Proposing team:
  - Marco Baldi (Univpm, Italy)
  - Alessandro Barenghi (Polimi, Italy)
  - Franco Chiaraluce (Univpm, Italy)
  - Gerardo Pelosi (Polimi, Italy)
  - Paolo Santini (Univpm, Italy)
- Official website (`https://www.ledacrypt.org/`):
  - First and second round specifications.
  - Full ANSI-C99 codebase.
  - Upcoming updates.
- Upcoming hardware implementation.

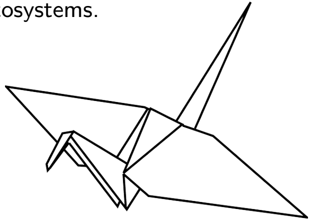


LEDAcrypt

- Suite of low-density parity-check code-based cryptosystems.
- Among the 26 second round candidates of the NIST pqcrypto competition.
- Proposing team:
  - Marco Baldi (Univpm, Italy)
  - Alessandro Barenghi (Polimi, Italy)
  - Franco Chiaraluce (Univpm, Italy)
  - Gerardo Pelosi (Polimi, Italy)
  - Paolo Santini (Univpm, Italy)
- Official website (https://www.ledacrypt.org/):
  - First and second round specifications.
  - Full ANSI-C99 codebase.
  - Upcoming updates.
- Upcoming hardware implementation.



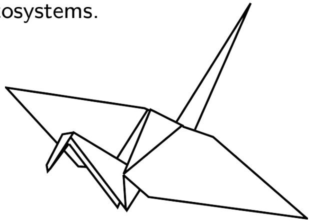
LEDAcrypt



- Suite of low-density parity-check code-based cryptosystems.
- Among the 26 second round candidates of the NIST pqcrypto competition.
- Proposing team:
  - Marco Baldi (Univpm, Italy)
  - Alessandro Barenghi (Polimi, Italy)
  - Franco Chiaraluce (Univpm, Italy)
  - Gerardo Pelosi (Polimi, Italy)
  - Paolo Santini (Univpm, Italy)
- Official website (`https://www.ledacrypt.org/`):
  - First and second round specifications.
  - Full ANSI-C99 codebase.
  - Upcoming updates.
- Upcoming hardware implementation.

LEDAcrypt

- Suite of low-density parity-check code-based cryptosystems.
- Among the 26 second round candidates of the NIST pqcrypto competition.
- Proposing team:
  - Marco Baldi (Univpm, Italy)
  - Alessandro Barenghi (Polimi, Italy)
  - Franco Chiaraluce (Univpm, Italy)
  - Gerardo Pelosi (Polimi, Italy)
  - Paolo Santini (Univpm, Italy)
- Official website (`https://www.ledacrypt.org/`):
  - First and second round specifications.
  - Full ANSI-C99 codebase.
  - Upcoming updates.
- Upcoming hardware implementation.



LEDAcrypt features

1. Both KEM and PKC modes.
2. Closed-form upper bound on the DFR.
3. Solves the mismatch between δ-correctness and DFR:
   - Mechanism to prevent the adversary from exploiting the advantage coming from the selection of input messages/error vectors.
   - This makes the maximum probability of a decryption failure over all plaintexts equal to the average failure probability over all plaintexts.
4. Algorithmic approach to the design of parameter sets.
5. Instances with:
   - Ephemeral keys and a DFR in the order of 10^{-9}, or
   - Long-term keys and a DFR of 2^{-64} or smaller than $2^{-\lambda}$, with $\lambda = 128, 192, 256$.

LEDAcrypt features

1. Both KEM and PKC modes.
2. Closed-form upper bound on the DFR.
3. Solves the mismatch between δ-correctness and DFR:
   - Mechanism to prevent the adversary from exploiting the advantage coming from the selection of input messages/error vectors.
   - This makes the maximum probability of a decryption failure over all plaintexts equal to the average failure probability over all plaintexts.
4. Algorithmic approach to the design of parameter sets.
5. Instances with:
   - Ephemeral keys and a DFR in the order of 10^{-9}, or
   - Long-term keys and a DFR of 2^{-64} or smaller than $2^{-\lambda}$, with $\lambda = 128, 192, 256$.

LEDAcrypt features

1. Both KEM and PKC modes.
2. Closed-form upper bound on the DFR.
3. Solves the mismatch between δ-correctness and DFR:
    - Mechanism to prevent the adversary from exploiting the advantage coming from the selection of input messages/error vectors.
    - This makes the maximum probability of a decryption failure over all plaintexts equal to the average failure probability over all plaintexts.
4. Algorithmic approach to the design of parameter sets.
5. Instances with:
    - Ephemeral keys and a DFR in the order of 10^{-9}, or
    - Long-term keys and a DFR of 2^{-64} or smaller than $2^{-\lambda}$, with $\lambda = 128, 192, 256$.

LEDAcrypt features

1. Both KEM and PKC modes.
2. Closed-form upper bound on the DFR.
3. Solves the mismatch between δ-correctness and DFR:
    - Mechanism to prevent the adversary from exploiting the advantage coming from the selection of input messages/error vectors.
    - This makes the maximum probability of a decryption failure over all plaintexts equal to the average failure probability over all plaintexts.
4. Algorithmic approach to the design of parameter sets.
5. Instances with:
    - Ephemeral keys and a DFR in the order of 10^{-9}, or
    - Long-term keys and a DFR of 2^{-64} or smaller than $2^{-\lambda}$, with $\lambda = 128, 192, 256$.

LEDAcrypt features

1. Both KEM and PKC modes.
2. Closed-form upper bound on the DFR.
3. Solves the mismatch between δ-correctness and DFR:
   - Mechanism to prevent the adversary from exploiting the advantage coming from the selection of input messages/error vectors.
   - This makes the maximum probability of a decryption failure over all plaintexts equal to the average failure probability over all plaintexts.
4. Algorithmic approach to the design of parameter sets.
5. Instances with:
   - Ephemeral keys and a DFR in the order of 10^{-9}, or
   - Long-term keys and a DFR of 2^{-64} or smaller than $2^{-\lambda}$, with $\lambda = 128, 192, 256$.

LEDAcrypt parameters

Table: Key, ciphertext and transmitted data sizes for **LEDAcrypt-KEM-CPA** instances with ephemeral keys.

NIST Category	n_0	p	d_v	t	Private Key (B)	Public Key(B)	Ciphertext (B)	Shared Secret (B)	Transmitted data (B)
1	2	10,883	71	133	1,160	1,368	1,392	32	2,760
	3	8,237	79	84	1,920	2,064	1,056	32	3,120
	4	7,187	83	67	2,680	2,712	928	32	3,640
3	2	21,011	103	198	1,680	2,632	2,664	48	5,296
	3	15,373	117	125	2,840	3,856	1,960	48	5,816
	4	13,109	123	99	3,968	4,920	1,672	48	6,592
5	2	35,339	137	263	2,232	4,424	4,464	64	8,888
	3	25,603	155	166	3,760	6,416	3,248	64	9,664
	4	21,611	163	132	5,256	8,112	2,744	64	10,856

BIKE

- BIKE parameters:

Security	r	w	t	DFR
Level 1	12,323	142	134	2^{-128}
Level 3	24,659	206	199	2^{-192}

- Private key, public key and ciphertext sizes (in bits):

Quantity	Size	Level 1	Level 3
Private key	$\ell + w \cdot \lceil \log_2(r) \rceil$	$2,244$	$3,346$
Public key	r	$12,323$	$24,659$
Ciphertext	$r + \ell$	$12,579$	$24,915$

Weak keys in LEDAcrypt

- Presented by Daniel Apon, Corbin McNeill, Ray Perlner and Angela Robinson at the 2019 Quantum Cryptanalysis Dagstuhl Seminar.
- Leverage the product structure of the public code parity-check matrix ($\mathbf{H}' = \mathbf{HQ}$).

Rationale

Making guesses separately on $\mathbf{H}$ and $\mathbf{Q}$ (and projecting them onto $\mathbf{H}'$) accelerates ISD with respect to making them directly on $\mathbf{H}'$.

- One key is weak if
  - Occurs with probability 2^{-x}.
  - Requires the equivalent of 2^y AES operations for ISD.
  - $x + y < \lambda$, being λ the claimed security level.

▶ D. Apon, R. Perlner, A. Robinson, and P. Santini, "Cryptanalysis of LEDAcrypt," Cryptology ePrint Archive, Report 2020/455, 2020.

Weak keys in LEDAcrypt

- Presented by Daniel Apon, Corbin McNeill, Ray Perlner and Angela Robinson at the 2019 Quantum Cryptanalysis Dagstuhl Seminar.
- Leverage the product structure of the public code parity-check matrix ($\mathbf{H}' = \mathbf{HQ}$).

Rationale

Making guesses separately on $\mathbf{H}$ and $\mathbf{Q}$ (and projecting them onto $\mathbf{H}'$) accelerates ISD with respect to making them directly on $\mathbf{H}'$.

- One key is weak if
  - Occurs with probability 2^{-x}.
  - Requires the equivalent of 2^y AES operations for ISD.
  - $x + y < \lambda$, being λ the claimed security level.

▶ D. Apon, R. Perlner, A. Robinson, and P. Santini, "Cryptanalysis of LEDAcrypt," Cryptology ePrint Archive, Report 2020/455, 2020.

Weak keys in LEDAcrypt

- Presented by Daniel Apon, Corbin McNeill, Ray Perlner and Angela Robinson at the 2019 Quantum Cryptanalysis Dagstuhl Seminar.
- Leverage the product structure of the public code parity-check matrix ($\mathbf{H}' = \mathbf{HQ}$).

Rationale

Making guesses separately on $\mathbf{H}$ and $\mathbf{Q}$ (and projecting them onto $\mathbf{H}'$) accelerates ISD with respect to making them directly on $\mathbf{H}'$.

- One key is weak if
  - Occurs with probability 2^{-x}.
  - Requires the equivalent of 2^y AES operations for ISD.
  - $x + y < \lambda$, being λ the claimed security level.

D. Apon, R. Perlner, A. Robinson, and P. Santini, "Cryptanalysis of LEDAcrypt," Cryptology ePrint Archive, Report 2020/455, 2020.

Weak key examples

$n_0 = 2$, cat. 5, IND-CPA

$x \approx 44$, $y \approx 52$

$n_0 = 4$, cat. 1, IND-CPA

$x \approx 40$, $y \approx 50$

- This attack works well when:
  - n_0 is small and
  - the weights of $\mathbf{H}$ and $\mathbf{Q}$ are well balanced.
- Countermeasures:
  - increase n_0,
  - choose unbalanced weights for $\mathbf{H}$ and $\mathbf{Q}$.
- Cautious choice: $\mathbf{Q} = \mathbf{I}$ and $\mathbf{H}' = \mathbf{H}$.

Weak key examples

$n_0 = 2$, cat. 5, IND-CPA

$x \approx 44$, $y \approx 52$

$n_0 = 4$, cat. 1, IND-CPA

$x \approx 40$, $y \approx 50$

- This attack works well when:
  - n_0 is small and
  - the weights of $\mathbf{H}$ and $\mathbf{Q}$ are well balanced.
- Countermeasures:
  - increase n_0,
  - choose unbalanced weights for $\mathbf{H}$ and $\mathbf{Q}$.
- Cautious choice: $\mathbf{Q} = \mathbf{I}$ and $\mathbf{H}' = \mathbf{H}$.

Weak key examples

$n_0 = 2$, cat. 5, IND-CPA

$x \approx 44$, $y \approx 52$

$n_0 = 4$, cat. 1, IND-CPA

$x \approx 40$, $y \approx 50$

- This attack works well when:
  - n_0 is small and
  - the weights of $\mathbf{H}$ and $\mathbf{Q}$ are well balanced.
- Countermeasures:
  - increase n_0,
  - choose unbalanced weights for $\mathbf{H}$ and $\mathbf{Q}$.
- Cautious choice: $\mathbf{Q} = \mathbf{I}$ and $\mathbf{H}' = \mathbf{H}$.

Weak key examples

$n_0 = 2$, cat. 5, IND-CPA

$x \approx 44$, $y \approx 52$

$n_0 = 4$, cat. 1, IND-CPA

$x \approx 40$, $y \approx 50$

- This attack works well when:
  - n_0 is small and
  - the weights of $\mathbf{H}$ and $\mathbf{Q}$ are well balanced.
- Countermeasures:
  - increase n_0,
  - choose unbalanced weights for $\mathbf{H}$ and $\mathbf{Q}$.
- Cautious choice: $\mathbf{Q} = \mathbf{I}$ and $\mathbf{H}' = \mathbf{H}$.

Open research challenges

We still need to work on:

- Weak keys deriving from the product structure and their countermeasures
- Iterative decoders and their theoretical modeling (DFR)
- Cryptanalysis exploiting sparse and structured matrices
- QC-LDPC and QC-MDPC code-based signature schemes

End of presentation

Thank you!

www.univpm.it/marco.baldi

m.baldi@univpm.it