LDPC code-based (and other) variants of the McEliece cryptosystem

Marco Baldi

Università Politecnica delle Marche
Ancona, Italy

m.baldi@univpm.it

McEliece cryptosystem

- Public Key Cryptosystem (PKC) proposed by McEliece in 1978 [1]
- Based on the problem of decoding a linear large code with no visible structure

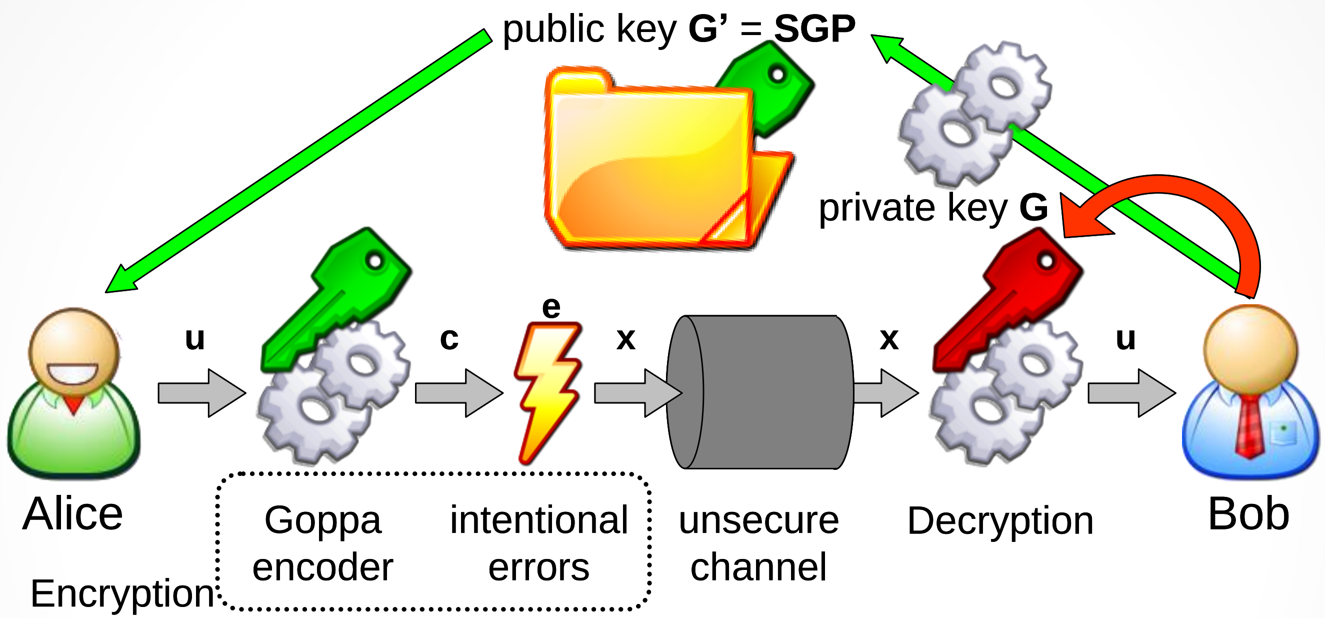
Still unbroken!

- Faster than competing solutions, like RSA.
- The original version uses binary Goppa codes with:
  - length $n = 1024$
  - dimension $k = 524$
  - minimum distance $d_{min} = 101$
  - error correction capability $t = 50$ errors

---

[1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report*, pp. 114–116, 1978.

McEliece cryptosystem (2)



- Private key: {**G**, **S**, **P**}
  - **G**: systematic generator matrix of a t-error correcting Goppa code
  - **S**: k x k non-singular scrambling matrix
  - **P**: n x n permutation matrix
- Public key: **G' = SGP**
- **e**: vector of t intentional errors

encryption map:

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

McEliece cryptosystem (3)

- After receiving **x**, Bob computes:

$$\mathbf{x}' = \mathbf{x}\mathbf{P}^{-1} = \mathbf{u}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}^{-1}$$

- He then corrects all the *t* errors and gets:

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

- Finally, Bob calculates $\mathbf{u}'\mathbf{S}^{-1}$, thus obtaining **u**

- Requisites for the codes:
  - For given *n*, *k* and *t*, the family of codes must be large enough to avoid any enumeration
  - An efficient algorithm must be known for decoding
  - A generator (or parity-check) matrix of a permutation equivalent code must give no information on the secret code

- Main drawback: large **public keys**

Niederreiter cryptosystem

- Exploits the same principle, but uses the code parity-check matrix (**H**) in the place of the generator matrix (**G**)

- Secret key: {**H**, **S**} → Public key: **H'** = **SH**

- Message mapped into a weight-*t* error vector (**e**)

- Encryption: $\mathbf{x} = \mathbf{H}'\mathbf{e}^T$
- Decryption: $\mathbf{s} = \mathbf{S}^{-1}\mathbf{x} = \mathbf{H}\mathbf{e}^T$ → syndrome decoding (**e**)

- Advantages:
  - shorter keys for code rate > ½
  - smaller encryption complexity

Goppa codes [2,3]

- Goppa codes are subfield subcodes of GRS codes
- Given:
  - A degree-t (irreducible) polynomial $g(x)$ in $GF(p^m)[x]$
  - A set of n elements of $GF(p^m)$ (support of the code) which are not zeroes of $g(x)$:

$$a_0, a_1, \ldots, a_{n-1}$$

- A Goppa code is defined as the set of vectors $\mathbf{c} = [c_0, c_1, \ldots, c_{n-1}]$, with c_i in $GF(p)$, such that:

$$\sum_{i=0}^{n-1} \frac{c_i}{x-\alpha_i} \equiv 0 \bmod g(x)$$

[2] V. D. Goppa, “A new class of linear error-correcting codes,” Probl. Peredach. Inform., vol. 6, no. 3, pp. 24-30, Sept. 1970.
[3] V. D. Goppa, “Rational representation of codes and (L,g) codes,” Probl. Peredach. Inform., vol. 7, no. 3, pp. 4149, Sept. 1971.

Goppa codes and key size

- Any degree-t (irreducible) polynomial generates a different code
- So, the number of different codes with same parameters and correction capability is very high
- Their matrices are non-structured, thus their storage requires:
  - kn bits for the McEliece cryptosystem
  - rn bits for the Niederreiter version
- In order to resist message resend attacks, a CCA2 secure conversion should be adopted [4]
- This also allows to store only the non-systematic part of the matrices, that is, rk bits.

---

[4] K. Kobara, H. Imai, "Semantically secure McEliece public-key cryptosystems - conversions for McEliece PKC", Proc. PKC 2001, pp. 19-35.

LDPC Codes

- Low-Density Parity-Check (LDPC) codes are state-of-art forward error correcting (FEC) codes
- Firstly introduced by Gallager in 1962 [5] and recently rediscovered [6]
- They are able to approach the channel capacity under belief propagation (BP) decoding [7]
- Now adopted in many applications and standards

[5] R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.

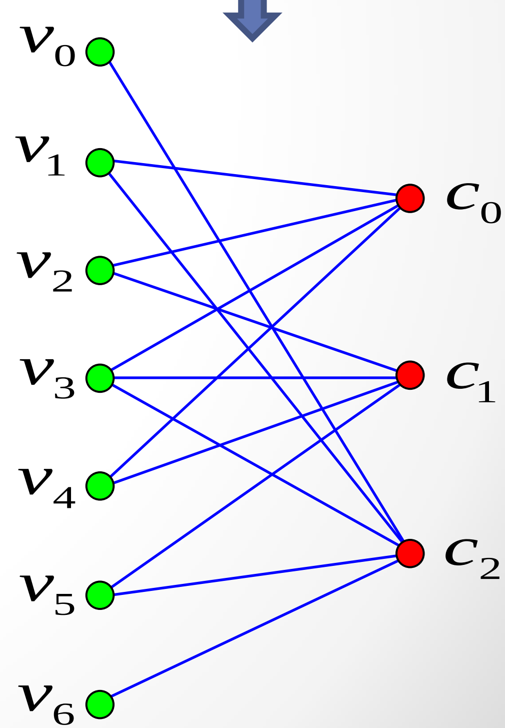
[6] D. J. C. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in Cryptography and Coding. 5th IMA Conference, ser. Lecture Notes in Computer Science, C. Boyd, Ed. Berlin: Springer, 1995, no. 1025, pp. 100–111.

[7] 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.

LDPC Codes (2)

- LDPC codes are linear block codes
  - n: code length
  - k: code dimension
  - $r = n - k$: code redundancy
  - $\mathbf{G}$: $k \times n$ generator matrix
  - $\mathbf{H}$: $r \times n$ parity-check matrix
  - d_v: average **H** column weight
  - d_c: average **H** row weight

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



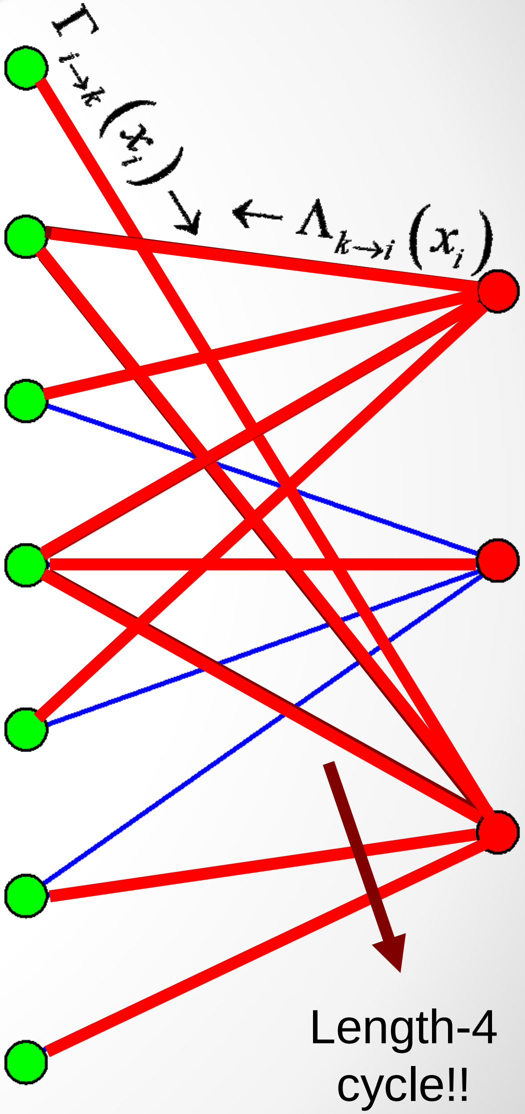
- LDPC codes have parity-check matrices with:
  - Low density of ones ($d_v << r$, $d_c << n$)
  - No more than one overlapping symbol 1 between any two rows/columns
  - No short cycles in the associated **Tanner graph**

LDPC decoding

- LDPC decoding can be accomplished through the Sum-Product Algorithm (SPA) with Log-Likelihood Ratios (LLR)
- For a random variable U:

$$LLR(U) = \ln\left[\frac{\Pr(U=0)}{\Pr(U=1)}\right]$$

- The initial LLRs are derived from the channel
- They are then updated by exchanging messages on the Tanner graph



LDPC decoding for the McEliece PKC

- The McEliece encryption map is equivalent to transmission over a special Binary Symmetric Channel with error probability $p = t/n$
- LLR of *a priori* probabilities associated with the codeword bit at position i:

$$LLR(x_i) = \ln\left[\frac{P(x_i = 0 \mid y_i = y)}{P(x_i = 1 \mid y_i = y)}\right]$$

- Applying the Bayes theorem:

$$LLR(x_i \mid y_i = 0) = \ln\left(\frac{1-p}{p}\right) = \ln\left(\frac{n-t}{t}\right)$$

$$LLR(x_i \mid y_i = 1) = \ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{t}{n-t}\right)$$

Bit flipping decoding

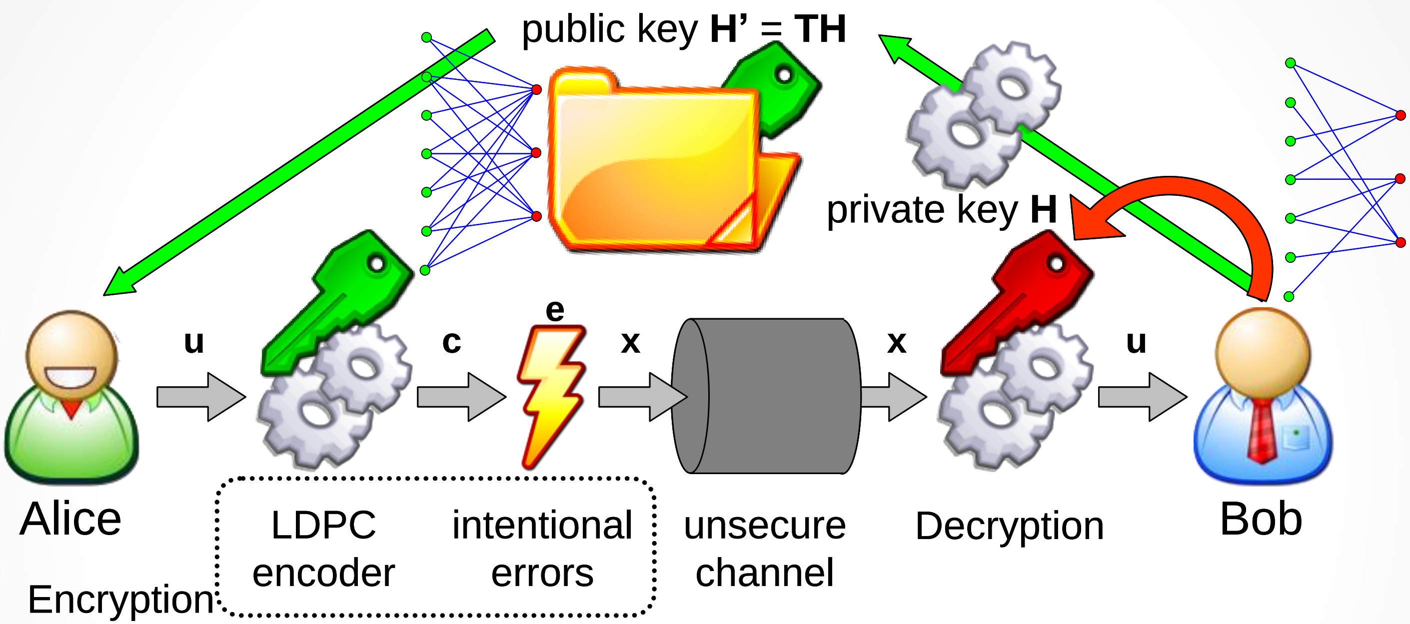
- LDPC decoding can also be accomplished through hard-decision iterative algorithms known as bit-flipping (**BF**)
- During an iteration, every check node sends each neighboring variable node the binary sum of all its neighboring variable nodes, excluding that node
- In order to send a message back to each neighboring check node, a variable node counts the number of unsatisfied parity-check sums from the other check nodes
- If this number overcomes some threshold, the variable node flips its value and sends it back, otherwise, it sends its initial value unchanged
- BF is well suited when soft information from the channel is not available (as in the McEliece cryptosystem)

Decoding threshold

- Differently from algebraic codes, the **decoding radius** of LDPC codes is not easy to estimate
- Their error correction capability is statistical (with a high mean)
- For iterative decoders, the **decoding threshold** of large ensembles of codes can be estimated through density evolution techniques
- The decoding threshold of BF decoders can be found by iterating simple closed-form expressions

n [bits]		12288	15360	18432	21504	24576	27648	30720	33792	36864	39936	43008	46080	49152
$R = 2/3$	$d_v = 13$	190	237	285	333	380	428	476	523	571	619	666	714	762
	$d_v = 15$	192	240	288	336	384	432	479	527	575	622	670	718	766
n [bits]		16384	20480	24576	28672	32768	36864	40960	45056	49152	53248	57344	61440	65536
$R = 3/4$	$d_v = 13$	181	225	270	315	360	405	450	495	540	585	630	675	720
	$d_v = 15$	187	233	280	327	374	421	468	515	561	608	655	702	749

First LDPC-based McEliece PKC [8]



- **H**: private LDPC matrix
- **T**: $r \times r$ transformation matrix
- **H' = TH**: public parity-check matrix (prevents LDPC decoding)
- **T** → must be dense to avoid possible recovering of **H** from **H'**
- **H'** → becomes dense too

[8] C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," in *Proc. IEEE ISIT 2000*, Sorrento, Italy, Jun. 2000, p. 215.

First LDPC-based McEliece PKC (2)

- The high density of **H'** helps preventing Eve from using the iterative LDPC decoder
- But a dense (and unstructured) **H'** gives no advantage in terms of key size over Goppa matrices
- Can we use structured LDPC codes (like Quasi-Cyclic LDPC codes) to "compensate" the need for dense matrices?

Quasi-Cyclic codes

- A linear block code is a Quasi-Cyclic (QC) code if [9]:
  1. Its dimension and length are both multiple of an integer p ($k = k_0 p$ and $n = n_0 p$)
  2. Every cyclic shift of a codeword by n_0 positions yields another codeword
  3. Each block of n_0 bits in a codeword is formed by k_0 information bits followed by $r_0 = n_0 - k_0$ parity bits (can be extended to the non-systematic case)

- The generator and parity-check matrices of a QC code can assume two alternative forms:
  - Circulant of blocks
  - Block of circulants

[9] R. Townsend, E. Jr. Weldon, "Self-orthogonal Quasi-Cyclic codes," IEEE Trans. Inform. Theory, vol. 13, no. 2, pp. 183–195, April 1967.

QC-LDPC codes with rate $(n_0 - 1)/n_0$

- For $r_0 = 1$, we obtain a particular family of codes with length $n = n_0 p$, dimension $k = k_0 p$ and rate $(n_0 - 1)/n_0$
- **H** assumes the form of a single row of circulants:

$$\mathbf{H} = \left[\mathbf{H}_0^c \quad \mathbf{H}_1^c \quad \cdots \quad \mathbf{H}_{n_0-1}^c \right]$$

completely described by its first row

- In order to be non-singular, **H** must have at least one non-singular block (suppose the last)
- In this case, **G** (in systematic form) is easily derived:

$$\mathbf{G} = \left[\mathbf{I} \quad \begin{array}{c} \left[\left(\mathbf{H}_{n_0-1}^c \right)^{-1} \cdot \mathbf{H}_0^c \right]^T \\ \left[\left(\mathbf{H}_{n_0-1}^c \right)^{-1} \cdot \mathbf{H}_1^c \right]^T \\ \vdots \\ \left[\left(\mathbf{H}_{n_0-1}^c \right)^{-1} \cdot \mathbf{H}_{n_0-2}^c \right]^T \end{array} \right]$$

completely described by its $(k + 1)$-th column

Random-based design

- We define "Random Difference Family" (RDF) a series of subsets of a finite group G such that every non-zero element of G appears no more than once as a difference of two elements in a subset
- An RDF can be used to obtain a QC-LDPC matrix free of length-4 cycles in the form:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0^c & \mathbf{H}_1^c & \cdots & \mathbf{H}_{n_0-1}^c \end{bmatrix}$$

- The random-based approach allows to design large families of codes with fixed parameters
- The codes in a family share the characteristics that mostly influence LDPC decoding, thus they have equivalent error correction performance

An example

- RDF over Z_{13}:
  - {1, 3, 8} (differences: 2, 11, 7, 6, 5, 8)
  - {5, 6, 9} (differences: 1, 12, 4, 9, 3, 10)

- Parity-check matrix (n_0 = 2, p = 13):

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

Using QC-LDPC codes

- The need for dense public matrices is no longer a problem if we exploit the structured nature of QC-LDPC codes
- However, if we use the classical McEliece setting, the public code is permutation equivalent to the private one
- Can it be possible to recover the secret representation of the code (differently from Goppa codes)?

Attack to the Dual Code

- The dual of the secret code has at least *r* codewords with very low weight
- An opponent can directly search for them, thus recovering **H**
- Stern's algorithm (or one of its more recent variants) searches for low weight codewords through an iterative procedure [10]
- Some values of work factor (W) for code rate 3/4:
  - $n = 16000$, $d_v = 13 \rightarrow W = \mathbf{2^{37.5}}$
  - $n = 32000$, $d_v = 17 \rightarrow W = \mathbf{2^{43.7}}$
  - $n = 64000$, $d_v = 21 \rightarrow W = \mathbf{2^{50.4}}$
- Even though long codes (and rather dense matrices) are adopted, the system is highly exposed to a total break!

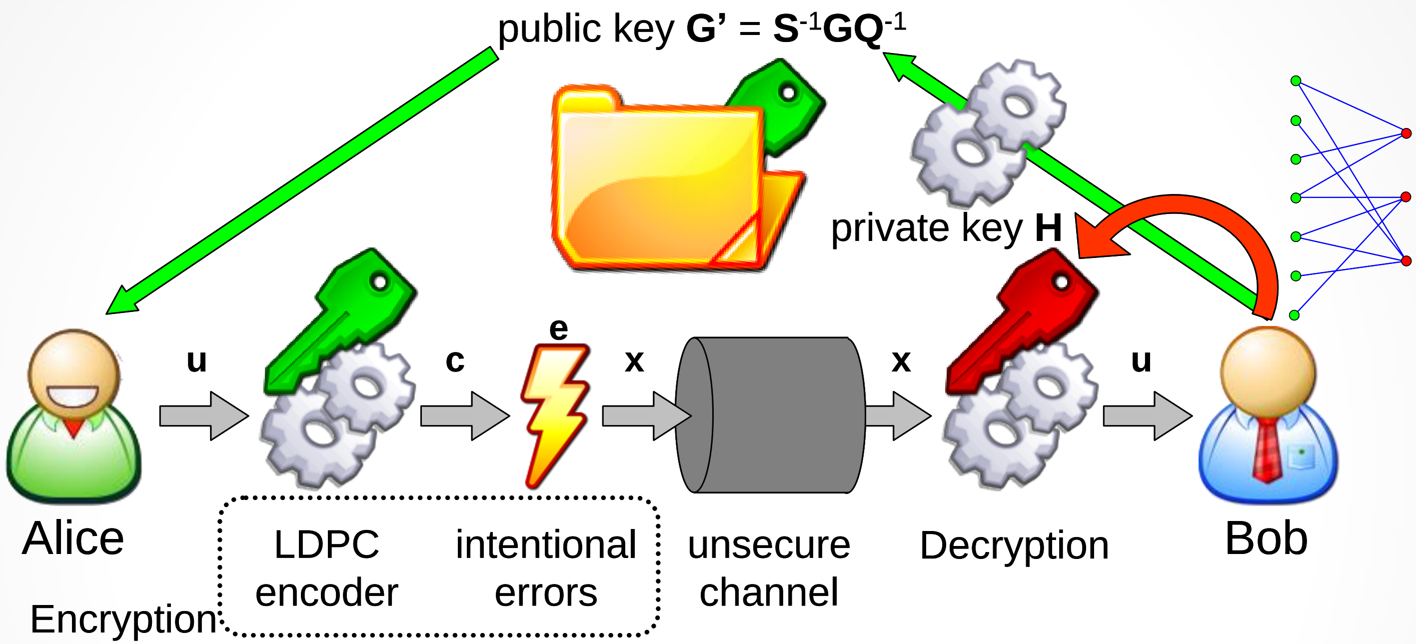
[10] J. Stern, "A method for finding codewords of small weight," LNCS, 1989, pp. 106–113.

Disguising the secret code [11]

- When using LDPC codes, we cannot expose a public code which is permutation equivalent to the private code
- To avoid this, we can replace the permutation matrix **P** with a more general (but still sparse) transformation matrix **Q**
- This way, we trade some error correcting capability for security

[11] M. Baldi, M. Bodrato, F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes", in Security and Cryptography for Networks, Vol. 5229 of Lecture Notes in Computer Science, pp. 246–262, Springer Berlin / Heidelberg, 2008.

New System Proposal



- **Q** is formed by n_0 x n_0 circulant blocks with size p
- The public code has parity-check matrix **H'** = **HQ**^{T}
- **Q** has column weight m
- The row weight of **H'** is ~ $m \cdot n_0 \cdot d_v$ → increased weight
- The QC-LDPC code must be able to correct $t = t'm$ errors (t' are those added by Alice)

New System Proposal (2)

- The permutation matrix used in the original McEliece is replaced by a (denser) transformation matrix **Q**
- The transformation must be inverted before LDPC decoding
- This causes an “error spreading” phenomenon during decryption...
- ...but it is compensated by the high correction capability of LDPC codes
- This prevents all attacks based on the code “sparsity”
- But a bad choice of **S** and **Q** can still expose the system to dangerous attacks

Attack to the dual code

- In the new system, the dual of the public code does not have low-weight codewords

- The dual code has codeword weight $\leq m \cdot n_0 \cdot d_v$

- Due to the matrix sparsity, it is highly probable that the minimum weight approaches $m \cdot n_0 \cdot d_v$

- Even a small m is sufficient to make searching for those codewords too difficult for an attacker

System parameters and key size

- Public key size (in bytes), considering a CCA2 secure conversion $[(n_0 - 1)p]$:

p [bits]	4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840	4096
$n_0 = 4$	1536	1920	2304	2688	3072	3456	3840	4224	4608	4992	5376	5760	6144

- The key size increases linearly in the code length!

Decoding Attacks

- Given an intercepted ciphertext **x**, the linear block code generated by:

$$\mathbf{G''} = \begin{bmatrix} \mathbf{G'} \\ \mathbf{x} \end{bmatrix}$$

  contains only one minimum weight codeword, and this coincides with the error vector **e**

- So, the problem of finding **e** translates into that of finding the minimum weight codeword of a linear block code

- We refer to the algorithm in [12] for the search of minimum weight codewords in a linear block code (with no visible structure)

[12] D. J. Bernstein, T. Lange, C. Peters, "Attacking and defending the McEliece cryptosystem," In Post-Quantum Cryptography, vol. 5299 of LNCS, pages 31–46. Springer Berlin / Heidelberg, 2008.

Decoding Attacks (2)

- Every blockwise cyclically shifted version of the ciphertext **x** is still a valid ciphertext
- Eve can continue extending **G''** by adding shifted versions of **x**, and can search for as many shifted versions of the error vector

Security level

- Minimum attack WF for *m* = 7:

p [bits]		4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	$d_v = 13$	2^{54}	2^{63}	2^{73}	2^{84}	2^{94}	2^{105}	2^{116}	2^{125}	2^{135}	2^{146}	2^{157}	2^{161}	2^{161}
	$d_v = 15$	2^{54}	2^{64}	2^{75}	2^{85}	2^{94}	2^{105}	2^{116}	2^{126}	2^{137}	2^{146}	2^{157}	2^{168}	2^{179}
$n_0 = 4$	$d_v = 13$	2^{60}	2^{73}	2^{85}	2^{98}	2^{109}	2^{121}	2^{134}	2^{146}	2^{153}	2^{154}	2^{154}	2^{154}	2^{154}
	$d_v = 15$	2^{62}	2^{75}	2^{88}	2^{100}	2^{113}	2^{127}	2^{138}	2^{152}	2^{165}	2^{176}	2^{176}	2^{176}	2^{176}

- Key size (in bytes):

p [bits]	4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840	4096
$n_0 = 4$	1536	1920	2304	2688	3072	3456	3840	4224	4608	4992	5376	5760	6144

Encryption complexity

- Encryption complexity (computing the product $\mathbf{u} \cdot \mathbf{G}'$ and adding the intentional error vector):

$$C_{enc} = C_{mul}(\mathbf{u} \cdot \mathbf{G}') + n$$

- Naïve computation: $C_{mul}(\mathbf{u} \cdot \mathbf{G}') = n \cdot k/2$

- Strong reduction by exploiting circulant matrices (Toom-Cook algorithm, Winograd convolution)

Binary operations for each encrypted bit

p [bits]	4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	726	823	919	1005	1092	1178	1236	1351	1380	1524	1510	1697	1639
$n_0 = 4$	956	1081	1206	1321	1437	1552	1624	1783	1811	2013	1984	2244	2157

Decryption complexity

- Decryption complexity can be split into three parts:
  - calculating the product $\mathbf{x} \cdot \mathbf{Q}$
  - decoding the secret LDPC code
  - calculating the product $\mathbf{u'} \cdot \mathbf{S}$

$$C_{dec} = C_{mul}(\mathbf{x} \cdot \mathbf{Q}) + C_{LDPC} + C_{mul}(\mathbf{u'} \cdot \mathbf{S})$$

- Concerning LDPC decoding, the SPA already has low complexity, and BF decoding further reduces it

Binary operations for each decrypted bit (BF decoding)

p [bits]		4096	5120	6144	7168	8192	9216	10240	11264	12288	13312	14336	15360	16384
$n_0 = 3$	$d_v = 13$	1476	1544	1611	1668	1726	1784	1827	1899	1928	2014	2014	2130	2101
	$d_v = 15$	1626	1694	1761	1818	1876	1934	1977	2049	2078	2164	2164	2280	2251
$n_0 = 4$	$d_v = 13$	1598	1694	1790	1877	1963	2050	2107	2223	2252	2396	2381	2569	2511
	$d_v = 15$	1731	1828	1924	2010	2097	2183	2241	2356	2385	2529	2515	2702	2644

Some comparison

- Comparison considering the Niederreiter version with 80-bit security (CCA2 secure conversion)

Solution	n	k	t	Key size [bytes]	Enc. compl.	Dec. compl.
Goppa based	1632	1269	33	57581	48	7890
QC-LDPC based	24576	18432	38	2304	1206	1790 (BF)

1/25 !

- Goppa code parameters proposed in [12]

- The QC-LDPC based system scales favourably when larger keys are needed, since the key size grows linearly with the code length, due to the quasi-cyclic nature of the codes

Generalization of the approach

- An even stronger disguisement of the secret code can be achieved by choosing:

$$\mathbf{Q} = \mathbf{R} + \mathbf{T}$$

- **T** is still a non-singular sparse component, which also has the (undesired) effect of propagating the intentional errors
- **R** is a singular disguisement matrix, whose effect on the intentional errors can be rendered null
- This stronger disguisement allows to revitalize the use of GRS codes in the McEliece cryptosystem, which have always incurred security flaws until now

Generalization of the approach (2)

- A toy solution is to obtain **R** as $\mathbf{a}^T\mathbf{b}$, with **a** and **b** two randomly chosen $1\times n$ vectors
- **a** is disclosed, **b** kept secret
- The intentional error vectors are selected such that $\mathbf{a}\mathbf{e}^T = 0$, thus $\mathbf{R}\mathbf{e}^T = \mathbf{0}$ and there is no error propagation due to **R**
- Actually, disclosing **a** generates a flaw
- A more clever scheme can be used, which exploits the same principle, but with some variants
- Known distinguishers are not able to tell the public matrix obtained from a GRS code from a random matrix

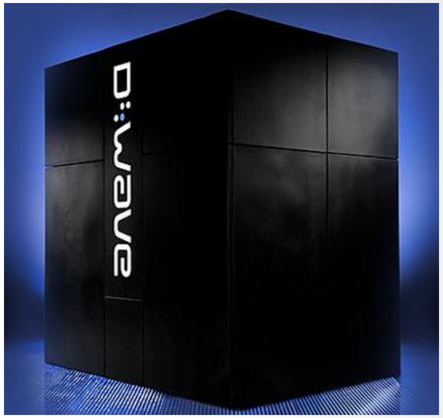
Using MDPC codes [13]

- A recent follow-up uses Moderate-Density Parity-Check (MDPC) codes in the place of LDPC codes
- With MDPC codes, the public code can still be permutation equivalent to the private code without incurring attacks to the dual code
- In addition, the correction capability of these codes remains the same even if some short cycles are present
- Thus, the design of MDPC codes can be completely random
- This has permitted to obtain the first security reduction (to the random linear code decoding problem) for these schemes
- On the other hand, decoding MDPC codes is more complex than for LDPC codes

[13] R. Misoczki, J.-P. Tillich, N. Sendrier, P. S. L. M. Barreto, "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes", cryptology ePrint archive, http://eprint.iacr.org/2012/409

A present (and future) challenge

- Quantum computers allow to factorize large integers and compute discrete logarithms in polynomial time
- They will break many widespread cryptographic and digital signature systems (RSA, DSA…)
- They will also endanger systems based on elliptic curves (like ECDSA)



D-Wave One™

- **October 2011**: first commercial and operational quantum computing academic center (University of Southern California, Lockheed Martin and D-Wave Systems)

Possible applications

- Code-based cryptography can be used to:
  - Provide security against attacks based on **quantum computers**
  - Implement lightweight encryption and decryption for resource-limited and **mobile** devices
  - Provide fast and up-to-date security tools for **cloud** platforms

- A practical example: **Cloud Wallet™**
  - App to securely save passport, bank and credit card details, photos, voice recordings and other sensitive information
  - Combines 256-bit AES encryption with Post-Quantum Secure McEliece encryption (Goppa-based)
  - Built on top of Dropbox to provide cloud storage

Preprint papers

- M. Baldi, M. Bianchi, F. Chiaraluce, "Security and complexity of the McEliece cryptosystem based on QC-LDPC codes"

  http://arxiv.org/abs/1109.5827

- M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, D. Schipani, "Enhanced public key security for the McEliece cryptosystem"

  http://arxiv.org/abs/1108.2462

ESCAPADE research project

http://escapade.dii.univpm.it

Backup slides

Attack to Circulant Permutation Matrices

- QC-LDPC codes based on circulant permutation blocks are widespread (also included in the IEEE 802.16e standard)
- Without null blocks, their parity-check matrices cannot have full rank
- Null blocks are commonly inserted in such a way to impose the lower triangular (or quasi-lower triangular) form
- A total-break attack is possible, in the form of a global deduction (find $\mathbf{T}_d$ and $\mathbf{H}_d$ such that $\mathbf{H}' = \mathbf{T}_d \cdot \mathbf{H}_d$ and $\mathbf{H}_d$ is suitable for BP decoding)
- It does not depend on the **T** density

Attack to Circulant Permutation Matrices (2)

$\mathbf{H}' = \mathbf{TH} = \mathbf{TZZ}^{-1}\mathbf{H} = \mathbf{T}_d\mathbf{H}_d$

$$\mathbf{H} = \left[\mathbf{P} \mid \mathbf{Z}\right]$$

$$\mathbf{H}' = \mathbf{T}\cdot\mathbf{H} = \begin{bmatrix} \mathbf{T}_{00} & \mathbf{T}_{01} & \mathbf{T}_{02} \\ \mathbf{T}_{10} & \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{20} & \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \Bigg[\underbrace{\begin{matrix} \mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}_{02} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{20} & \mathbf{P}_{21} & \mathbf{P}_{22} \end{matrix}}_{\mathbf{P}} \ \Bigg| \ \underbrace{\begin{matrix} \mathbf{P}_{03} & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_{13} & \mathbf{P}_{14} & \mathbf{0} \\ \mathbf{P}_{23} & \mathbf{P}_{24} & \mathbf{P}_{25} \end{matrix}}_{\mathbf{Z}} \Bigg]$$

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{P}_{03} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{14} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{25} \end{bmatrix}$$

$\mathbf{H}_b$ has the same density as **H** **(total break)!**

$$\mathbf{Z}^{-1} = \begin{bmatrix} \mathbf{V}_{00} & \mathbf{0} & \mathbf{0} \\ \mathbf{V}_{10} & \mathbf{V}_{11} & \mathbf{0} \\ \mathbf{V}_{20} & \mathbf{V}_{21} & \mathbf{V}_{22} \end{bmatrix}$$

weight 1

weight 1

weight 2

through correlation operations on $\mathbf{H}_d$ a further step is possible, that results in $\mathbf{H}_b$ corresponding to **Z***

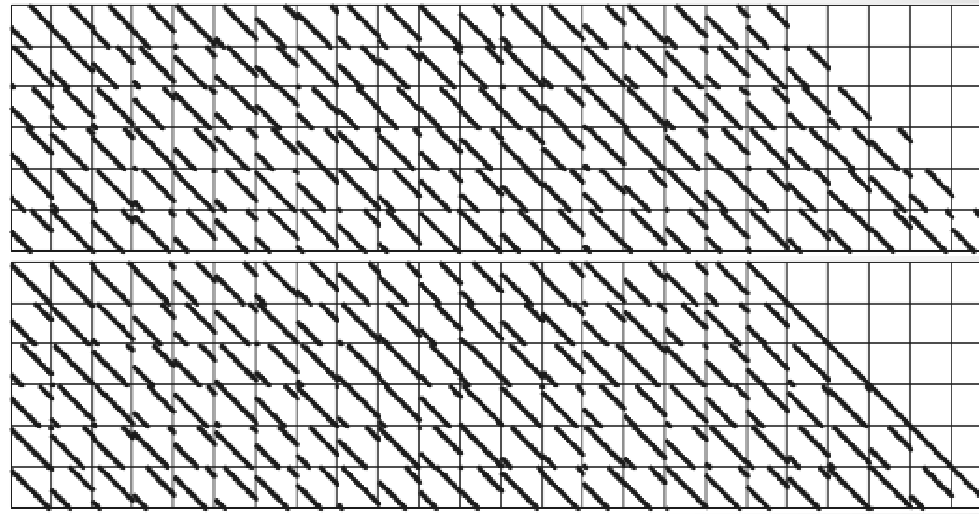
$$\mathbf{H}_d = \mathbf{Z}^{-1}\mathbf{H} = \left[\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{I}\right]$$

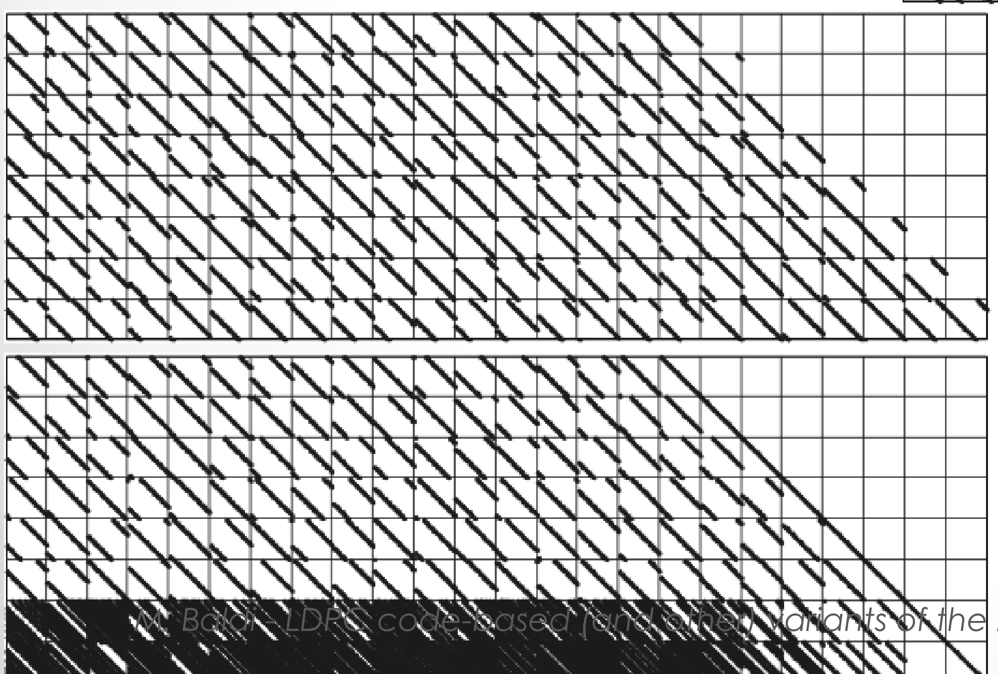
$$\mathbf{H}' = \mathbf{T}_d\mathbf{H}_d = \left[\mathbf{T}_d\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{T}_d\right]$$

by knowing $\mathbf{T}_d$, $\mathbf{H}_d$ can be calculated (as $\mathbf{T}_d^{-1}\mathbf{H}'$) and it is sparse (since $\mathbf{H}_d = \mathbf{Z}^{-1}\mathbf{H}$)

Attack to CPMs - Examples

Successful global deduction
for $n_0 = 24$, $r_0 = 6$, $p = 40$





Unsuccessful global deduction
for $n_0 = 24$, $r_0 = 8$, $p = 40$

What to avoid

- In our first version [15] we chose:
  - d_v = 13
  - p = 4032
  - m = 7
  - t' = 27
- This choice allows to resist all standard attacks
- For reducing complexity, both **S** and **Q** were chosen sparse, with non-null blocks having row/column weight *m* (that is small)
- Q was in diagonal form:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_{n_0-1} \end{bmatrix}$$

[15] M. Baldi, F. Chiaraluce, "Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes," Proc. IEEE ISIT 2007, Nice, France (June 2007) 2591–2595

OTD attack

- A new attack was formulated by Otmani et al. (OTD) [16]
- It is based on the fact that, by selecting the first k columns of $\mathbf{G'}$, an eavesdropper gets

$$\mathbf{G}'_{\leq k} = \mathbf{S}^{-1} \cdot \begin{bmatrix} \mathbf{Q}_0^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Q}_{n_0-2}^{-1} \end{bmatrix}$$

- By inverting $\mathbf{G'}_{\leq k}$ and considering its block at position (i, j), he can obtain $\mathbf{Q}_i\mathbf{S}_{i,j}$, that corresponds to the polynomial

$$g_{i,j}(x) = q_i(x) \cdot s_{i,j}(x) \bmod (x^p + 1)$$

- If both $\mathbf{Q}_i$ and $\mathbf{S}_{i,j}$ are sparse (with row/col weight m), it is highly probable that $g_{i,j}(x)$ has exactly m^2 non-null coefficients and its support contains at least one shift:

$$x^d \cdot q_i(x),\ 0 \leq d \leq p-1$$

[16] A. Otmani, J.P. Tillich, L. Dallot, "Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes," Proc. SCC 2008, Beijing, China (April 2008)

OTD attack (2)

- Three attack strategies
- First strategy: enumerate and validate all m-tuples belonging to the support of $g_{i,j}(x)$

$$\text{WF} = 2^{50.3}$$

- Second strategy: calculate all possible Hadamard products $g^d_{i,j}(x) \otimes g_{i,j}(x)$ and check whether the resulting polynomial has support with weight m

$$\text{WF} = 2^{36}$$

- Third strategy: consider the i-th row ($\mathbf{R}_i$) of the inverse of $\mathbf{G'}_{\leq k}$ and search for low weight codewords in the code generated by $(\mathbf{Q}_i\mathbf{S}_{i,0})^{-1} \cdot \mathbf{R}_i$

$$\text{WF} = \mathbf{2^{32}}$$

Countermeasures

- OTD attacks exploit the sparse nature of **S** and **Q** and the block-diagonal form of **Q**
- They can be countered by adopting dense **S** matrices [17]
- With dense **S**, Eve cannot obtain $\mathbf{Q}_i$ and $\mathbf{S}_{i,j}$, even knowing $\mathbf{Q}_i\mathbf{S}_{i,j}$
- The choice of a dense **S** influences decoding complexity
- But efficient algorithms for circulant matrices can be adopted [17]
- **Q** must be sparse to allow correction of all intentional errors
- A block-diagonal **Q** is weak, so it is advisable to avoid it

[17] M. Baldi, M. Bodrato, F. Chiaraluce, "A New Analysis of the McEliece Cryptosystem based on QC-LDPC Codes," Proc. SCN 2008, Amalfi, Italy, vol. 5229 of LNCS., Springer (2008) 246–262