McEliece cryptosystem based on LDPC codes

Marco Baldi
Università Politecnica delle Marche - DEIT
Ancona, Italy
m.baldi@univpm.it

Bergen – 25 October 2007

Outline

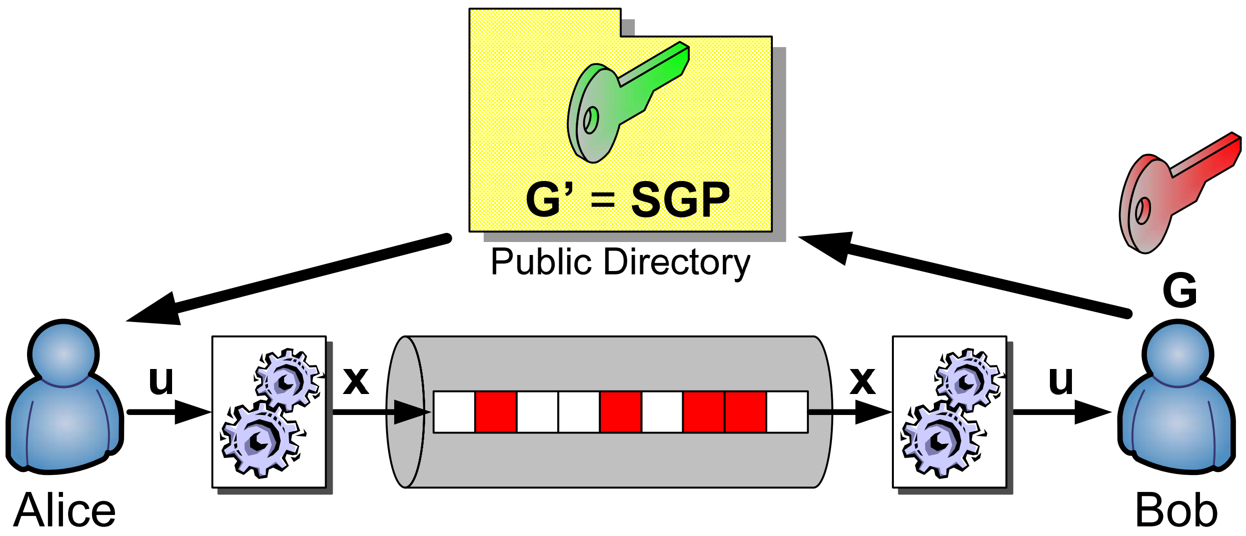
- The McEliece cryptosystem
- LDPC codes
- First LDPC-based version
- QC-LDPC codes
- Cryptanalysis
- Revision of the cryptosystem
- Complexity
- Conclusions

The McEliece Cryptosystem

- Public Key Cryptosystem (PKC) proposed by R. J. McEliece in 1978 [1].
- Based on algebraic coding theory (difficulty of decoding a linear large code with no visible structure).
- Still unbroken!
- Faster than competing solutions, like RSA.
- Adopts 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.

The McEliece Cryptosystem (2)



- **G** is the generator matrix of a *t*-error correcting Goppa code, in systematic form
- **S** is a *k* x *k* non-singular scrambling matrix
- **P** is an *n* x *n* permutation matrix
- The encryption map is:

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

- **e** is a vector of *t* intentional errors

The 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 recovers:

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

- 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 is large enough to avoid any enumeration.
  - An efficient algorithm is known for decoding.
  - A generator (or parity-check) matrix of a permutation equivalent code gives no information on the secret code.

- Main drawbacks:
  - Long keys
  - Low transmission rate

LDPC Codes

- Low-Density Parity-Check (LDPC) codes are state-of-art forward error correcting (FEC) codes.
- Firstly introduced by Gallager in 1962 [2] and recently rediscovered [3].
- They are able to approach the channel capacity under belief propagation (BP) decoding [4].

---

[2] R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.
[3] 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.
[4] 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)

- Many applications and hardware implementations.
- Inclusion in several telecommunications standards.

LDPC Codes are Linear Block Codes

- A binary linear block code is a map:

  $$C(n, k): GF_2^k \rightarrow GF_2^n$$

  with image Γ, a vectorial subspace of GF_2^n.

- It exists a *kxn* generator matrix **G** such that:

  $$\Gamma = \mathrm{Im}\{\mathbf{G}\}$$

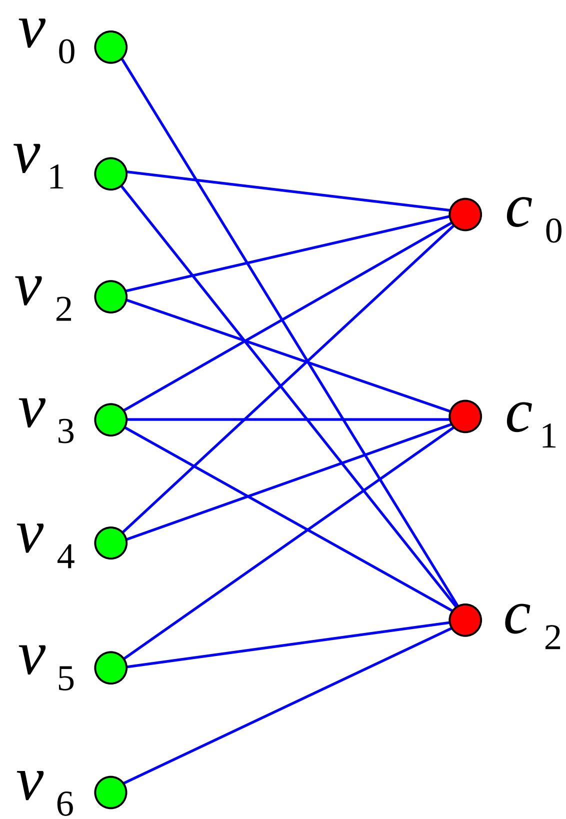
- It exists an *rxn* (*r* = *n* - *k*) parity-check matrix **H** such that:

  $$\Gamma = \mathrm{Ker}\{\mathbf{H}\}$$

- LDPC codes have parity-check matrices with special characteristics.

LDPC matrices

- The parity-check matrix **H** is asssociated with a bipartite (Tanner) graph.
- It has *n* variable nodes and *r* control nodes.
- The BP decoding algorithm works on the Tanner graph.



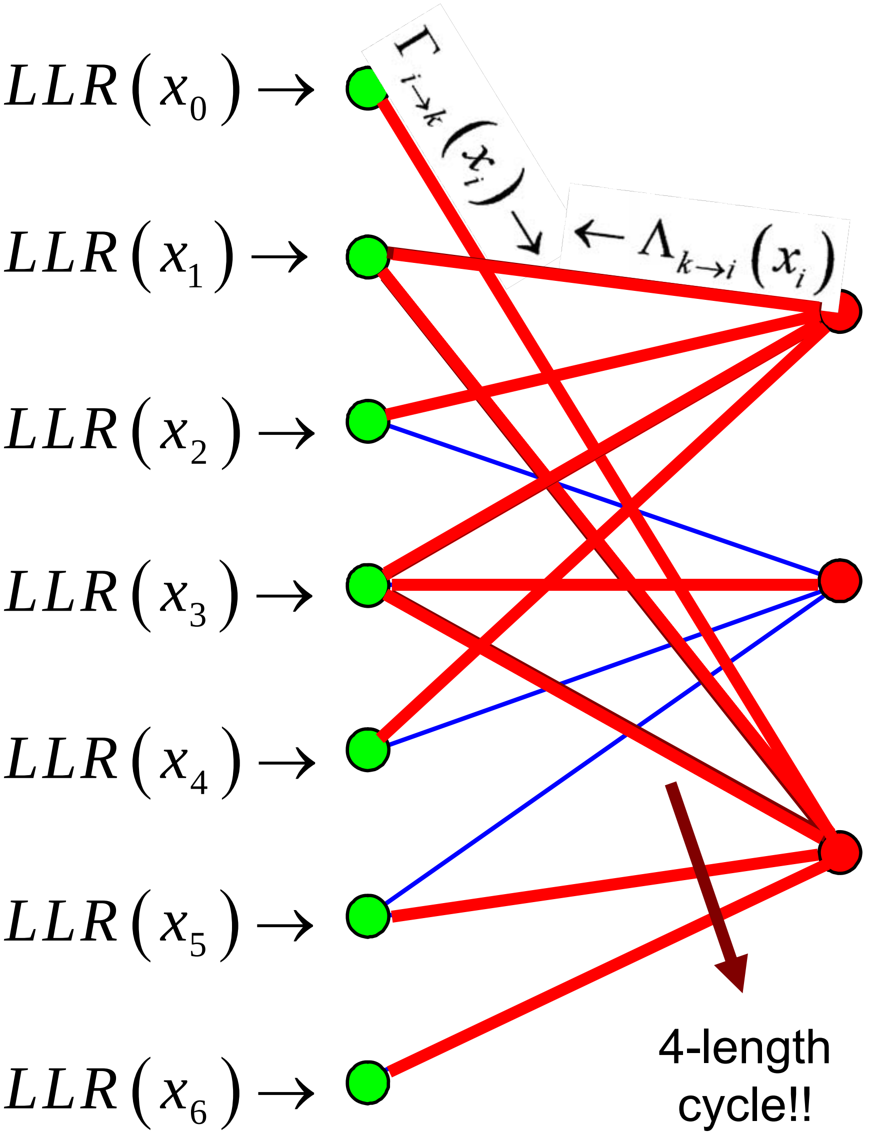
- In order to reach optimality, BP needs a graph free of short cycles.
- This can be achieved in sparse graphs → sparse **H** matrices.

LDPC decoding

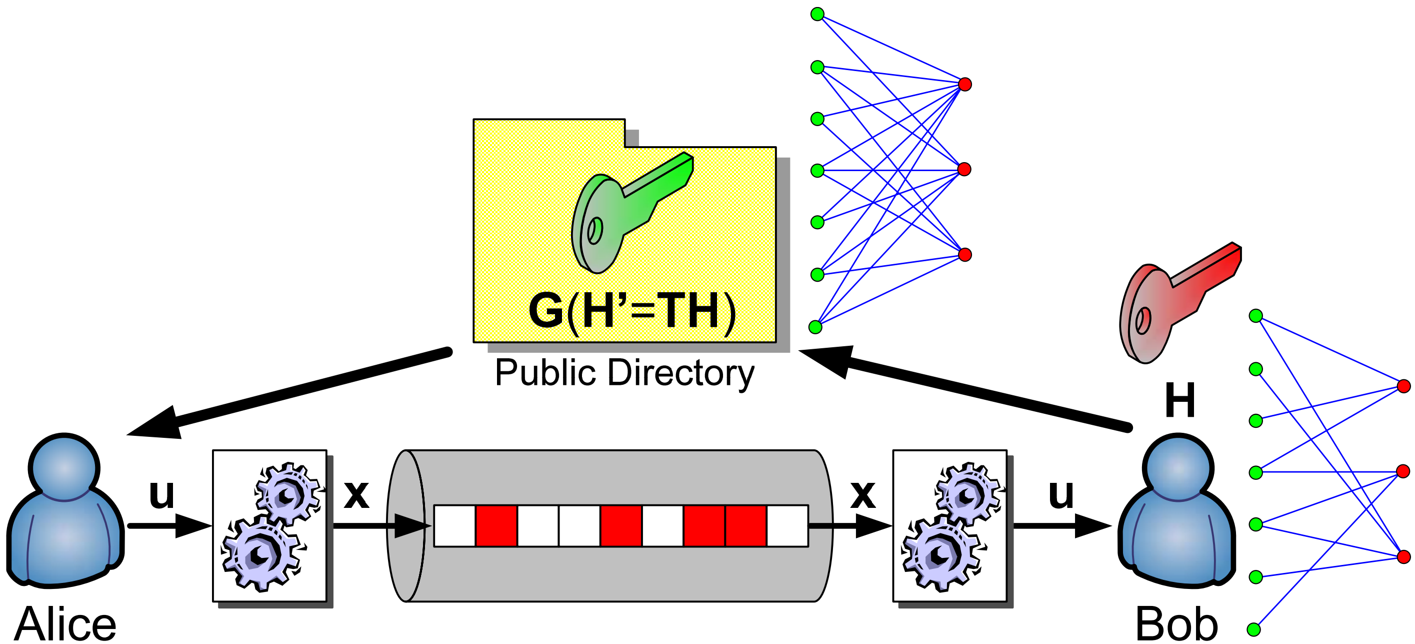
- The LLR-SPA decoder uses likelihood values on the logarithmic scale.
- For a random variable U:

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

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



First LDPC-based McEliece PKC



- Basically derived from the proposal in [5]
- **H** is the private LDPC matrix
- **H' = TH** is the public parity-check matrix (must be dense)
- **G** is a generator matrix derived from **H'**

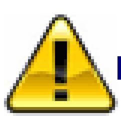
[5] C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," in *Proc. IEEE ISIT 2000*, Sorrento, Italy, Jun. 2000, p. 215.

First LDPC-based McEliece PKC (2)

- Also this version uses a scrambling matrix **S**.
- Alice calculates **G’** = $\mathbf{S}^{-1}\mathbf{G}$ and uses the standard encryption map:

$$\mathbf{x} = \mathbf{uG'} + \mathbf{e}$$

- The BP decoder works only on sparse and short cycle free Tanner graphs.
- Bob, who knows **H**, can correct all the *t* errors and apply the decryption map.
- An eavesdropper only knows **H’**, that is unsuitable for BP decoding.



- However, the secret code is completely exposed (**G** is a valid generator matrix for it)…
- …while in the original system it was hidden.

Choice of *t* for LDPC Codes

- This application of LDPC codes can be modeled as transmission over a particular BSC channel with error probability $p = t/n$.
- Log-likelihood ratio of *a priori* probabilities associated with the codeword bit at position *i*:

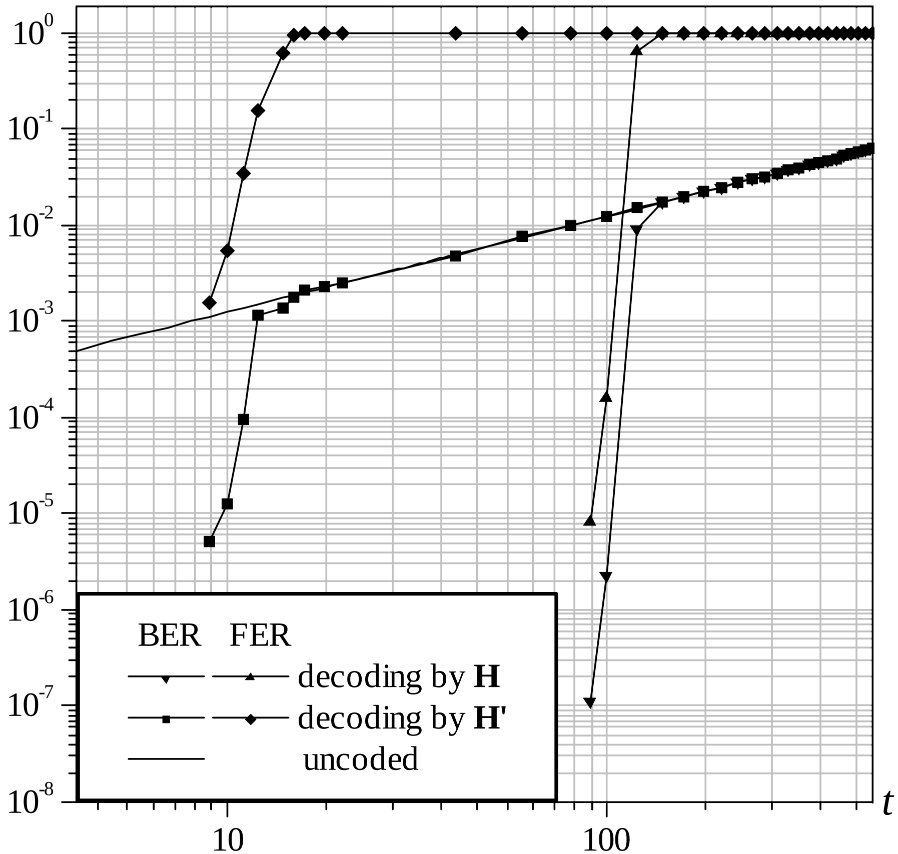
$$LLR(x_i) = \ln\left[\frac{P\left(x_i = 0 \mid y_i = y\right)}{P\left(x_i = 1 \mid y_i = y\right)}\right]$$

- Applying the Bayes theorem:

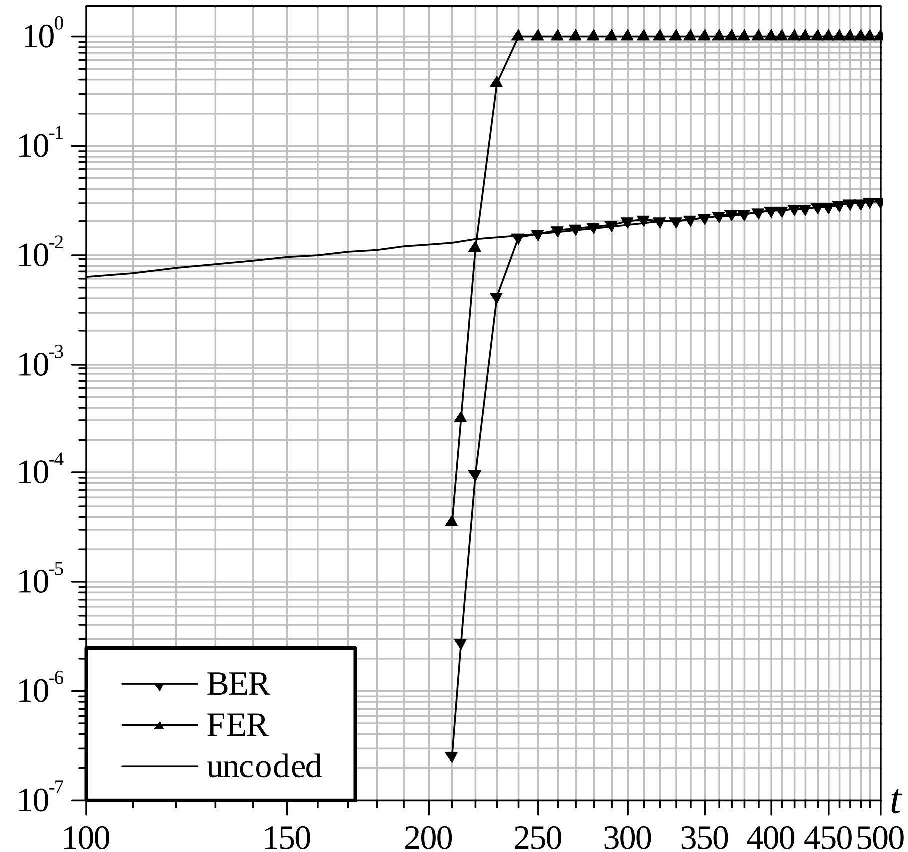
$$LLR(x_i \mid y_i = 0) = \ln\left(\frac{1-p}{p}\right) = \ln\left(\frac{n-t}{t}\right)$$

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

Choice of *t* for LDPC Codes (2)



QC-LDPC code with n = 8000, k = 6000 and d_v = 13. Decoding by **H** and by **H'**.



QC-LDPC code with n = 16128, k = 12096 and d_v = 13, under q = 6 bit quantization.

Quasi-Cyclic codes

- A linear block code is a Quasi-Cyclic (QC) code if [6]:
  1. It has dimension and length both multiple of an integer p ($k = k_0 p$ and $n = n_0 p$).
  2. 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.
  3. Every cyclic shift of a codeword by n_0 positions yields another codeword.

- Property 2 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

[6] R. Townsend and E. Jr. Weldon, Self-orthogonal Quasi-Cyclic codes. IEEE Trans. Inform. Theory, 13(2):183–195, April 1967.

Block of circulants form for **H**

- **H** is formed by r_0xn_0 blocks $\mathbf{H}_{ij}^c$:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{00}^c & \mathbf{H}_{01}^c & \cdots & \mathbf{H}_{0(n_0-1)}^c \\ \mathbf{H}_{10}^c & \mathbf{H}_{11}^c & \cdots & \mathbf{H}_{1(n_0-1)}^c \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{(r_0-1)0}^c & \mathbf{H}_{(r_0-1)1}^c & \cdots & \mathbf{H}_{(r_0-1)(n_0-1)}^c \end{bmatrix},$$

- Each $\mathbf{H}_{ij}^c$ is a *p*x*p* circulant matrix:

$$\mathbf{H}_{ij}^c = \begin{bmatrix} h_0^{ij} & h_1^{ij} & \cdots & h_{p-1}^{ij} \\ h_{p-1}^{ij} & h_0^{ij} & \cdots & h_{p-2}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ h_1^{ij} & h_2^{ij} & \cdots & h_0^{ij} \end{bmatrix}$$

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

- For $r_0 = 1$, a particular family of codes with length $n = n_0 p$, dimension $k = k_0 p$ and rate $(n_0 - 1)/n_0$ is derived.

- **H** assumes the form of a single row of circulants:

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

- 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:

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

completely described by its $(k + 1)$-th column, *i.e.* by ***k* bits** (key length)

QC-LDPC codes based on DFs

- A difference family is a series of subsets of a finite group *G* (base-blocks) such that every non-zero element of *G* appears exactly λ times as a difference of two elements from a base-block.
- If $G \equiv Z_p$, each base-block can be associated to a *pxp* circulant matrix (its elements give the positions of the 1 symbols in the matrix first row).
- If a difference family with λ = 1 is used to obtain a QC-LDPC matrix in the form:

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

H is free of 4-length cycles [7].

[7] S. Johnson and S. Weller, "A family of irregular LDPC codes with low encoding complexity," IEEE Commun. Lett., vol. 7, pp. 79–81, Feb. 2003.

QC-LDPC codes based on RDFs

- We define "Random Difference Family" a random multi-set with the properties of a difference family.
- The random-based approach permits to design large family of codes with fixed parameters.
- Given n_0, p and d_v (degree of variable nodes), the number of different codes is:

$$N\left(n_0, d_v, p\right) \geq \frac{1}{p} \binom{p}{d_v}^{n_0} \prod_{l=0}^{n_0-1} \prod_{j=1}^{d_v-1} \frac{p - j\left[2 - p \bmod 2 + \left(j^2 - 1\right)/2 + l d_v \left(d_v - 1\right)\right]}{p - j}$$

QC-LDPC codes based on RDFs (2)

- The number of different codes is very high:

$$\begin{cases} N\left(n_0 = 4, d_v = 11, p = 4032\right) \geq 2^{391} \\ N\left(n_0 = 4, d_v = 13, p = 4032\right) \geq 2^{94} \end{cases} \longleftarrow \text{estimated through sub-RDFs}$$

- The error correction performance of codes based on (n_0, d_v, p)-RDFs is equivalent, since they share:
  - code length and rate
  - parity check matrix density
  - nodes degree distributions
  - girth length distribution

- They are good candidates for the use in the McEliece cryptosystem!!

QC-LDPC codes in the McEliece PKC

- QC-LDPC codes based on RDFs seem able to overcome the main drawbacks of the original McEliece PKC.

- We consider QC-LDPC codes with:
  - $p = 4032$
  - $r_0 = 1$
  - $n_0 = 4$ (rate R = 3/4)
  - $n = n_0 p = 16128$
  - $k = k_0 p = 12096$

- Their applicability must be subject to cryptanalysis.

Information Set Decoding Attacks

- An eavesdropper could select only k elements of $\mathbf{x}$ and $\mathbf{e}$, chosen at fixed positions (the first k, for example), together with the corresponding k columns of $\mathbf{G'}$.
- The encryption map for this "information set" would be:

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

- If, by random choice, it is $\mathbf{e}_k = \mathbf{0}$, $\mathbf{u}$ can be easily obtained as $\mathbf{x}_k\mathbf{G'}_k^{-1}$ (assuming $\mathbf{G'}_k$ non-singular).
- Lee and Brickell generalized this attack by exploiting also the case $\mathbf{e}_k \neq \mathbf{0}$ [8].
- Considering a subset of all the possible $\mathbf{e}_k$ vectors (those with weight $\leq j$) can be convenient for the eavesdropper.

[8] P. Lee and E. Brickell, "An observation on the security of McEliece's public-key cryptosystem," EUROCRYPT 88, pp. 275–280.

Information Set Decoding Attacks

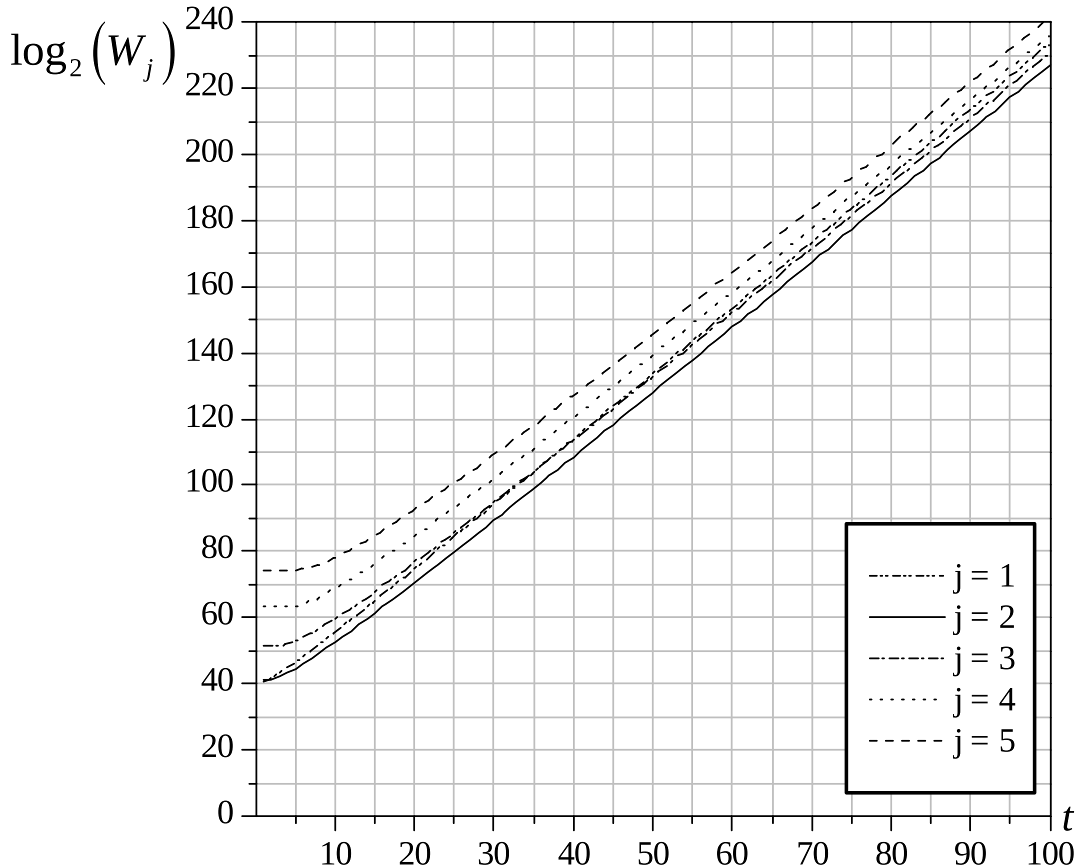
- Binary Work Factor (average number of binary operations required by a successful attack):

$$W_j = T_j\left(\alpha k^3 + N_j \beta k\right)$$

$$T_j = 1/\sum_{i=0}^{j}\binom{t}{i}\binom{n-t}{k-i}/\binom{n}{k}$$

$$N_j = \sum_{i=0}^{j}\binom{k}{i}$$

$$\alpha = \beta = 1$$



- For n_0 = 4, d_v = 13, p = 4032, the minimum work factor is achieved for j = 2.

- The choice t > 25 implies $W_2 > 2^{80}$.

Brute force attacks

- Excluded, since every enumeration attempt is too demanding…
- …even considering each circulant block of $\mathbf{H}$ ($\mathbf{H}_i$)

Message-Resend and Related-Message Attacks

- Berson proved that ISD attacks become very easy in such cases [9].
- Bob's LDPC decoder may (very rarely) need message resending.
- Attacks can be avoided through a simple modification of the encryption/decryption map based on a hash function:

$$\mathbf{x} = [\mathbf{u} + h(\mathbf{e})]\mathbf{G}' + \mathbf{e} \qquad\qquad \mathbf{u} = [\mathbf{u} + h(\mathbf{e})] + h(\mathbf{e})$$

---

[9] T. A. Berson, "Failure of the McEliece public-key cryptosystem under message-resend and related-message attack," CRYPTO '97.

Minimum Weight Codewords 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.

- A clever probabilistic algorithm to find minimum weight codewords is due to Stern [10].

[10] J. Stern, "A method for finding codewords of small weight," Lecture Notes in Computer Science, 1989, pp. 106–113.

Stern Algorithm

- Probability of finding, in one iteration, a codeword with weight *w* (supposed unique):

$$P_w = \frac{\binom{w}{g}\binom{n-w}{k/2-g}}{\binom{n}{k/2}} \frac{\binom{w-g}{g}\binom{n-k/2-w+g}{k/2-g}}{\binom{n-k/2}{k/2}} \frac{\binom{n-k-w+2g}{l}}{\binom{n-k}{l}}$$

where *g* and *l* are two parameters to optimize.

- Average number of iterations: $c = P_w^{-1}$.
- Total work factor: $W = cB$, with *B* (binary operations per iteration):

$$B = \frac{r^3}{2} + kr^2 + 2gl\binom{k/2}{g} + \frac{2gr\binom{k/2}{g}^2}{2^l}$$

Minimum Weight Codewords Attacks

- The original McEliece PKC adopts Goppa codes with $n = 1024$, $k = 524$ and $w = t = 50$.
- In this case, the minimum work factor is $W \sim 2^{64}$, found with $(g, l) = (3, 28)$.
- Adopting longer codes increases the work factor.
- For $n = 16128$, $k = 12096$ and $w = t = 27$ it reaches 2^{72} (minimum for $g = 3$ and $l = 46$).
- High enough for a local deduction attack.
- The choice of a small t does not compromise security.

Density Reduction Attacks

- Already conceived for the original LDPC-based McEliece PKC.
- If matrix **T** is sparse, matrix **H'** is sparse too.
- It is highly probable that sparse vectors are orthogonal.
- The rows of **H'** are linear combinations of those of **H**.
- When a row of **H** is involved in two rows of (a sparse) **H'** their product could directly reveal the row of **H**.
- The solution consists in adopting dense **T** matrices to avoid rows orthogonality.
- Dense **H'** matrices have no advantage on the key size.
- We propose to use QC-LDPC codes to fill the gap.

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 so that 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 $\mathbf{H'} = \mathbf{T}_d \mathbf{H}_d$ and $\mathbf{H}_d$ is suitable for BP decoding).
- It does not depend on the T density.

Attack to Circulant Permutation Matrices (2)

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

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

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

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

H$_b$ has the same density of **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

…

correlation operations on the sparse **H**$_d$ can permit to derive **H**$_b$, that corresponds to **Z***

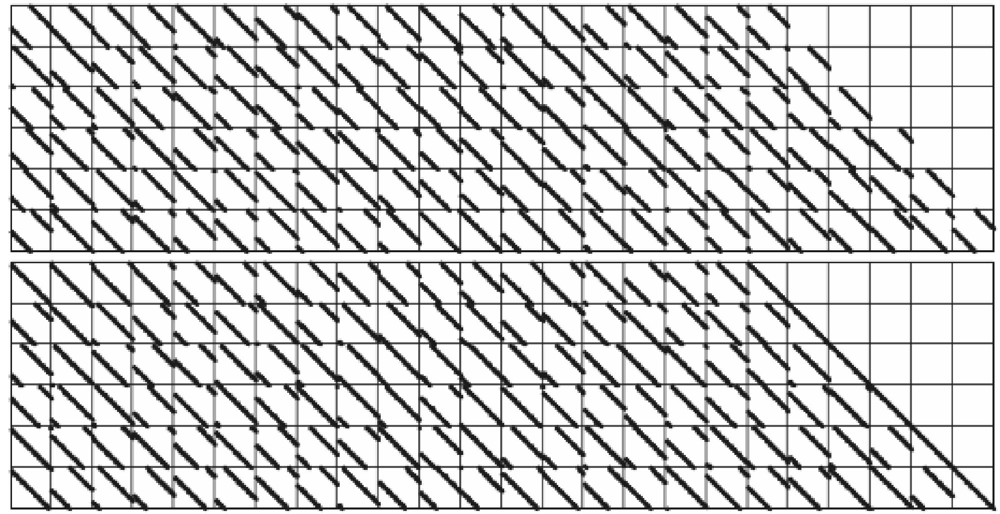
$$\mathbf{H}_d = \mathbf{Z}^{-1}\mathbf{H} = \left[\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{I}\right]$$

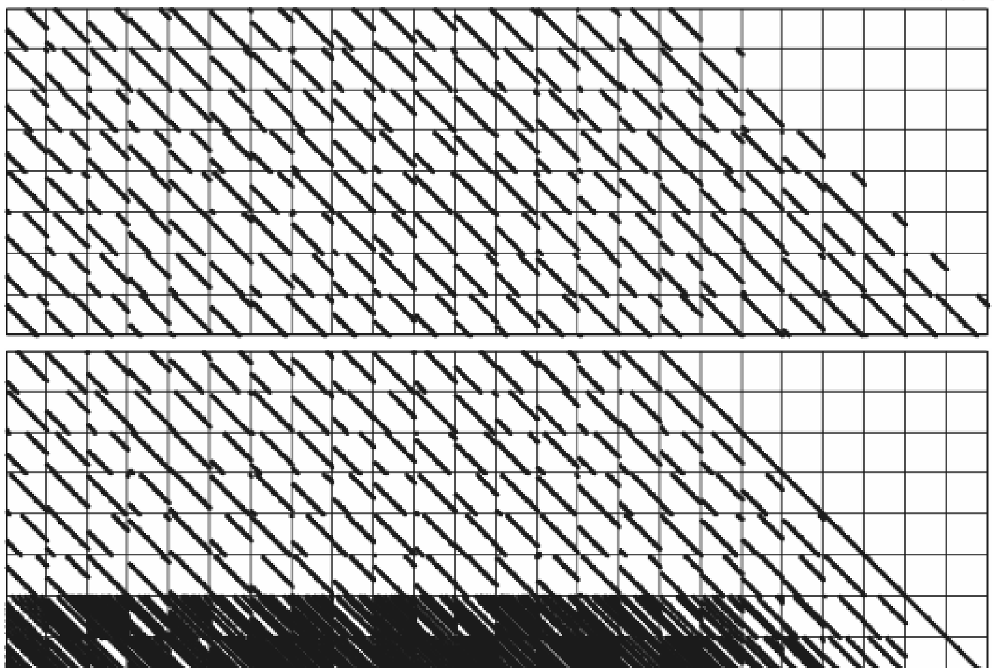
$$\mathbf{H}' = \mathbf{T}_d\mathbf{H}_d = \left[\mathbf{T}_d\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{T}_d\right]$$

knowing **T**$_d$, **H**$_d$ can be derived, that is sparse

Attack to CPMs - Examples

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





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

Attack to the Dual Code

- The dual of the secret code has very low weight codewords.
- An opponent can directly search for them, thus recovering **H**.
- The dual of the secret code has, at least, $A_{d_c} \geq r$ codewords with weight $d_c = d_v/(1\text{-}R)$.
- Since $d_c \ll n$, we can consider $A_{d_c} \sim r$.
- Stern algorithm searches for low weight codewords through an iterative procedure.
- Probability of finding, in one iteration, a (supposed unique) w-weight codeword of the dual code:

$$P_w = \frac{\binom{w}{g}\binom{n-w}{r/2-g}}{\binom{n}{r/2}} \cdot \frac{\binom{w-g}{g}\binom{n-r/2-w+g}{r/2-g}}{\binom{n-r/2}{r/2}} \cdot \frac{\binom{n-r-w+2g}{l}}{\binom{n-r}{l}}$$

Attack to the Dual Code (2)

- If the code contains A_w codewords with weight w, it is $P_{w,A_w} \leq A_w P_w$.

- Average number of iterations needed to find one of them:

$$c \geq P_{w,A_w}^{-1}$$

- Each iteration requires N binary operations:

$$N = \frac{k^3}{2} + rk^2 + 2gl\binom{r/2}{g} + \frac{2gk\binom{r/2}{g}^2}{2^l}$$

- The total work factor is $W = cN$.

  - $p = 4032$ ($A_{d_c} = r = 4032$)
  - $n_0 = 4$ ($n = 16128$)
  - $w = d_c = 52$

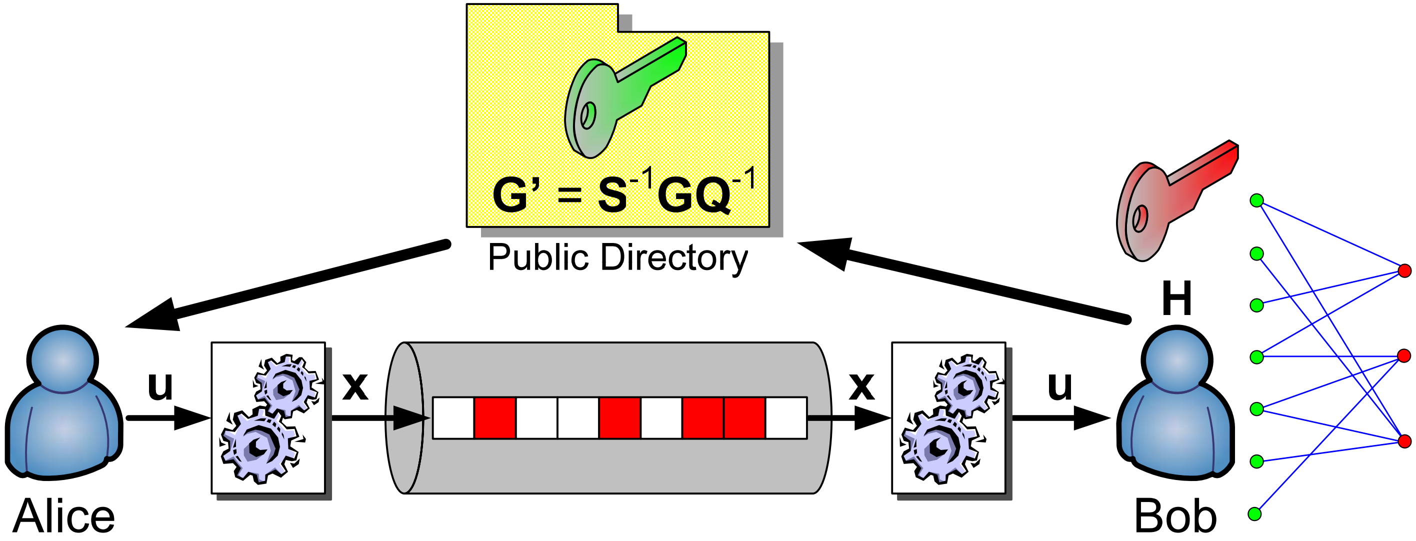
$W = 2^{37.5}$ (minimum for $g = 3$, $l = 43$)

- Unless very long codes and low rates are adopted, the system is highly exposed to a total break!

New System Proposal

- Cryptosystems based on LDPC codes must avoid to expose the secret code or a permuted version of it.
- Neither the previous LDPC-based proposal nor the original cryptosystem are suitable.
- We propose a new cryptosystem version.
- It recovers the original version but replaces the permutation matrix **P** with a sparse circulant matrix **Q**.
- The new system still adopts QC-LDPC codes in order to reduce the key length.

New System Proposal (2)



- **Q** is formed by n_0 x n_0 circulants of size *p.*
- The public code has parity-check matrix **H' = HQ**^{T}.
- **Q** has column weight *m* and block diagonal form.
- The row weight of **H'** is ~ md_c → increased weight.
- The QC-LDPC code must be able to correct $t = t'm$ errors (t' are those added by Alice).

Choice of the System Parameters

- QC-LDPC codes based on RDFs can still be adopted.
- We propose the following code parameters:
  - p = 4032
  - n_0 = 4 (R = 3/4, n = 16128, k = 12096)
  - d_v = 13 ($d_c = n_0 d_v$ = 52)
- The choice of t' = 27 protects against previous attacks.
- The choice of m = 7 protects against brute-force attempts on the blocks of **Q**.
- We propose the same row/column weight for the blocks of **S** → $s = mk_0$ = 21.
- Using Stern's algorithm to search for codewords with weight md_c = 364 is too demanding (w = 200 → $W = 2^{88.1}$).
- The QC-LDPC codes must correct up to $t'm$ = 190 errors.

Complexity

- **G'** has no more systematic form → n_0 columns are needed to describe it (key length = n_0k).

- Encryption complexity:
  - For a generic **G** matrix (dense): $C_{enc} = nk/2 + n$
  - For a QC **G** matrix the Toom-Cook algorithm can be applied:

$$C_{enc} = n_0\left[k_0 C_{pm}(p) + (k_0 - 1)p\right] + n$$

$C_{pm}(p)$ = binary operations for polynomial multiplication over $GF_2[x]\text{mod}(x^p + 1)$ ($C_{pm}(4032) = 1.68 \times 10^6$ with the Toom-Cook algorithm)

- Decryption complexity (SPA operations [11] + sparse matrix multiplications):

$$C_{dec} = n \cdot m + I_{ave}\left\{n\left[q\left(8d_v + 12R - 11\right) + d_v\right]\right\} + k \cdot s$$

[11] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient Implementations of the SPA for decoding LDPC codes," (*GLOBECOM '01)*

Comparison with other PKCs

	McEliece (original) [12]	Niederreiter [12]	RSA [12]	McEliece (QC-LDPC)
Key Size (bytes)	67072	32750	256	6048
Information Bits	524	276	1024	12096
Transmission Rate	0.5117	0.5681	1	0.75
Enc Ops per bit	514	50	2402	1671
Dec Ops per bit	5140	7863	738112	4197

- Improved key length and transmission rate with respect to McEliece and Niederreiter.
- RSA has shortest keys and highest rate, but highest complexity.
- The new cryptosystem seems a good trade-off between the original McEliece and the RSA PKCs.

[12] A. Canteaut and F. Chabaud, “A new algorithm for finding minimum-weight words in a linear code: application to McEliece’s cryptosystem…,” IEEE Trans. Inform. Theory, vol. 44, pp. 367–378, Jan. 1998.

Conclusions and future work

- The McEliece cryptosystem has long key and low rate.
- Can LDPC codes overcome such issues?
- Random-based LDPC codes do not permit to reduce the key length.
- QC-LDPC codes can hit the target, but not if based on permutation matrices.
- QC-LDPC codes based on DFs can overcome the drawbacks of the original system, while ensuring a good level of security…
- …but a “killer” attack exists based on the dual code.

Conclusions and future work (2)

- We have proposed a revised version of the McEliece PKC that can:
  - Successfully employ QC-LDPC codes based on RDFs
  - Resist the attack to the dual code
  - Overcome the main drawbacks of the original system
- Do new attacks exist specifically conceived for the proposed PKC?
- Besides QC-LDPC codes, are other codes suitable for this framework?

For more details…

arXiv:0710.0142 [ps, pdf, other]

LDPC Codes in the McEliece Cryptosystem

Marco Baldi, Franco Chiaraluce

Comments: Submitted to the IEEE Transactions on Information Theory

Subjects: **Information Theory (cs.IT)**

http://arxiv.org/abs/0710.0142