Using LDPC codes in the McEliece cryptosystem

Marco Baldi

Università Politecnica delle Marche
Ancona, Italy

m.baldi@univpm.it

17 Nov. 2010

How it began…

C. Monico, J. Rosenthal,

A. Shokrollahi,

"Using low density parity check codes in the McEliece cryptosystem,"

Proc. ISIT 2000, page 215, Sorrento, Italy, June 2000

…after a 10-year investigation:

Can LDPC codes be used in the McEliece cryptosystem without compromising its security?

ISIT 2000, Sorrento, Italy, June 25-30,2000

Using Low Density Parity Check Codes in the McEliece Cryptosystem

Chris Monico
Department of Mathematics
University of Notre Dame
Notre Dame, Indiana 46556
e-mail: cmonico@nd.edu
http://www.nd.edu/~cmonico/

Joachim Rosenthal[1]
Department of Mathematics
University of Notre Dame
Notre Dame, Indiana 46556
email: Rosenthal.1@nd.edu
http://www.nd.edu/~rosen/

Amin Shokrollahi
Bell Labs
600 Mountain Avenue
Murray Hill, NJ 07974
email:
amin@research.bell-labs.com

Abstract — **We examine the implications of using a Low Density Parity Check Code (LDPCC) in place of the usual Goppa code in McEliece's cryptosystem. Using a LDPCC allows for larger block lengths and the possibility of a combined error correction/encryption protocol.**

I. Introduction

If one wishes to use a LDPCC in the McEliece system, there are several ways to proceed. An efficient way seems to be the following:

As usual, suppose Bob wishes to send Alice a secure message over an insecure channel. Alice chooses a random $(n-k)\times n$ sparse parity check matrix, H, for a binary LDPCC, $\mathcal{C}$, that admits decoding of any pattern of t or fewer errors with, say, belief propagation. She also randomly chooses sparse invertible matrices $S \in GL(k,\mathbb{F}_2)$ and $T \in GL(n-k,\mathbb{F}_2)$. She then calculates $\tilde{H} := TH$ and has keys:

Public Key: $(\tilde{H}, S, t)$

Private Key: (H, T)

Now, if Bob wants to send Alice the message m, he first computes the generator matrix, G, for the code $\mathcal{C}$ in row reduced echelon form, and then computes $\tilde{G} = S^{-1}G$. He then applies the encryption map:

$$m \mapsto m\tilde{G} + e =: y$$

where e is a random error vector of weight at most t. Alice's decryption procedure is then as follows: Since $\tilde{G}$ and G define the same code, $\mathcal{C}$, she can use H to decode the word y to $m\tilde{G} = mS^{-1}G$. Since G is in row reduced echelon form, this reveals mS^{-1} in the k coordinates of $m\tilde{G}$ in which G has only one nonzero entry (i.e., the *systematic coordinates* of G). Right multiplication by S finally recovers Bob's message m.

This seems relatively efficient because the keys consist of sparse matrices, allowing considerable compression. Hence, one could have key sizes comparable to those of a (1024, 512) McEliece system, but for a code of size (16384, 8192).

II. Security

The security of this system is based on two observations:

- If T is chosen with the proper parameters, $\tilde{H}$ will most likely not admit decoding with, e.g. belief propagation, for the correction of up to t errors.
- It *seems* difficult to recover a matrix, H', equivalent to $\tilde{H}$ that admits decoding with, e.g. belief propagation, for the correction of up to t errors. In particular it seems difficult to recover the specific degree structure of the parity check matrix H.

However, a simple observation shows that if T is chosen too sparsely, this latter task is *not* difficult. In what follows, if $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$ are two vectors over $\mathbb{F}_2$, $u * v := (u_1 v_1, \ldots, u_n v_n)$ denotes the intersection of the binary vectors u, v. This is a vector whose support is exactly $\mathrm{supp}(u) \cap \mathrm{supp}(v)$. Equivalently, it can be considered as the 'AND' of u and v.

Let $h_1, \ldots, h_{n-k}$ denote the row vectors of H and $\tilde{h}_1, \ldots, \tilde{h}_{n-k}$ the row vectors of $\tilde{H}$. Notice that the h_i are sparse vectors and each $\tilde{h}_j$ is a linear combination of the h_i. Furthermore, if T is sparse, each $\tilde{h}_j = h_{j_1} + \cdots + h_{j_{w_j}}$ with the w_j small. That is, each $\tilde{h}_j$ is a linear combination of a small number of rows of H. If the w_j are too small (i.e., T is too sparse), then with reasonable probability one has that $\tilde{h}_j * h_{j_m} = h_{j_m}$ for many of the $1 \le j \le n-k$, $1 \le j_m \le j_{w_j}$. In this case, since each h_{j_m} appears in several of the $\tilde{h}_j$, we can, with non-negligible probability, find j_1, j_2 such that

$$\tilde{h}_{j_1} * \tilde{h}_{j_2} = h_i$$

for some i. Thus, in time $k(k-1)/2$, we can recover some of the original rows of H by computing the intersection of all pairs of rows, checking to see if the intersection is in $Rowsp(\tilde{H})$. Having found some of the original rows, we can determine, with high probability, which of the $\tilde{h}_j$ have these rows as components in their linear combinations. We thus subtract each original row from the $\tilde{h}_j$ that have many nonzero coordinates in common with it. Then go back to computing the intersection of all pairs of rows again, and keep repeating until we've found sufficiently many original rows to allow decoding.

III. Conclusion

Empirical evidence has shown this attack and some variants of it, to be effective enough that we consider this system insecure unless T is chosen to be dense. Thus, there seems to be no advantage to using a parity check matrix as the public key. However, this system is still of possible interest in the following case: If one is using a LDPCC for error correction, some security can be added at very little extra cost.

References

[1] R.J. McEliece, "A Public Key Cryptosystem Based on Algebraic Coding Theory," *Technical Report DSN Progress Report #42-44*, Jet Propulsion Laboratory, Pasadena, California, 1978.

[2] R.G. Gallager, "Low Density Parity Check Codes," *MIT Press*,Cambridge, MA, 1963.

[3] T. Richardson, M.A. Shokrollahi, and R. Urbanke. Design of provably good low-density parity check codes. *IEEE Trans. Inform. Theory (submitted)*, 1999.

[1]The research is supported in part by NSF grant DMS-96-10389.

0-7803-5857-0/00/$10.00 ©2000 IEEE.

A 10-year story

(of the non-Goppa armada)

- C. Monico, J. Rosenthal, A. Shokrollahi, "**Using low density parity check codes in the McEliece cryptosystem**," Proc. IEEE ISIT 2000, p. 215, Sorrento, Italy, June 2000.
- P. Gaborit, "**Shorter keys for code based cryptography**," Proc. Int. Workshop on Coding and Cryptography (WCC 2005), pp. 81–90, Bergen, Norway, March 2005.
- M. Baldi, F. Chiaraluce, "**Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes**," Proc. IEEE ISIT 2007, pp. 2591–2595, Nice, France, June 2007.
- A. Otmani, J.P. Tillich, L. Dallot, "**Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes**," Proc. First International Conference on Symbolic Computation and Cryptography (SCC 2008), pp. 69-81, Beijing, China, April 2008.
- M. Baldi, M. Bodrato, F. Chiaraluce, "**A new analysis of the McEliece cryptosystem based on QC-LDPC codes**," in Security and Cryptography for Networks, Lecture Notes in Computer Science, vol. 5229, pp. 246–262, 2008.
- T.P. Berger, P.-L. Cayrel, P. Gaborit, A. Otmani, "**Reducing key length of the McEliece cryptosystem**," Proc. AfricaCrypt 2009, Lecture Notes in Computer Science, vol. 5580, pp. 77-97, 2009.
- M.K. Shooshtari, M. Ahmadian, A. Payandeh, "**Improving the security of McEliece-like public key cryptosystem based on LDPC codes**," Proc. 11th International Conference on Advanced Communication Technology (ICACT 2009), pp. 1050-1053, Phoenix Park, Korea, Feb. 2009.
- M. Baldi, "**LDPC codes in the McEliece cryptosystem: attacks and countermeasures**," in Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes, NATO Science for Peace and Security Series (D: Information and Communication Security), vol. 23, pp. 160-174, IOS Press, 2009.
- R. Misoczki, P.S.L.M. Barreto, "**Compact McEliece keys from Goppa codes**," Cryptology ePrint Archive, Report 2009/187, 2009.
- V.G. Umana and G. Leander, "**Practical key recovery attacks on two McEliece variants**," Cryptology ePrint Archive, Report 2009/509, 2009.
- K. Kobara, "**Flexible quasi-dyadic code-based public-key encryption and signature**," Cryptology ePrint Archive, Report 2009/635, 2009.
- J.-C. Faugère, A. Otmani, L. Perret, J.-P. Tillich, "**Algebraic cryptanalysis of McEliece variants with compact keys**," Proc. 29th International Conference on Cryptology (EUROCRYPT 2010), Nice, France, June 2010.
- C. Wieschebrink, "**Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes**," Proc. 3rd International Workshop on Post-Quantum Cryptography (PQCrypto 2010), Lecture Notes in Computer Science, vol. 6061, pp. 61-72, 2010.
- J.-C. Faugère, A. Otmani, L. Perret, J.-P. Tillich, "**Algebraic cryptanalysis of compact McEliece's variants - toward a complexity analysis**," Proc. 2nd International Conference on Symbolic Computation and Cryptography (SCC 2010), pp. 45-56, Egham, UK, June 2010.
- D.J. Bernstein, T. Lange, C. Peters, "**Wild McEliece**," Proc. Selected Areas in Cryptography (SAC 2010), Waterloo, Canada, August 2010.

The McEliece Cryptosystem

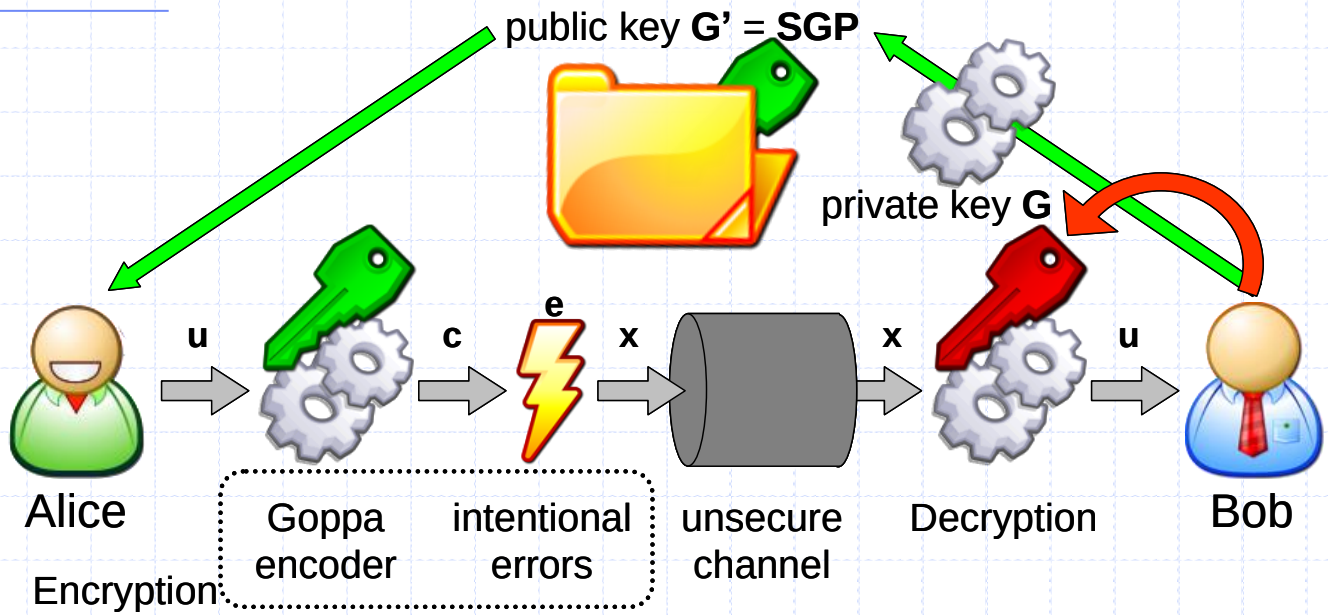
- Public Key Cryptosystem (PKC) proposed by McEliece in 1978 [1]
- Based on the problem of decoding a linear large code with no visible structure

Still unbroken!

- Faster than competing solutions, like RSA.
- 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**: generator matrix of a t-error correcting Goppa code, in systematic form
- **S**: k x k non-singular scrambling matrix
- **P**: n x n permutation matrix
- **e**: vector of t intentional errors

encryption map:

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

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 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 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 more recently rediscovered [3]
- They are able to approach the channel capacity under belief propagation (BP) decoding [4]
- Now adopted in many applications and standards

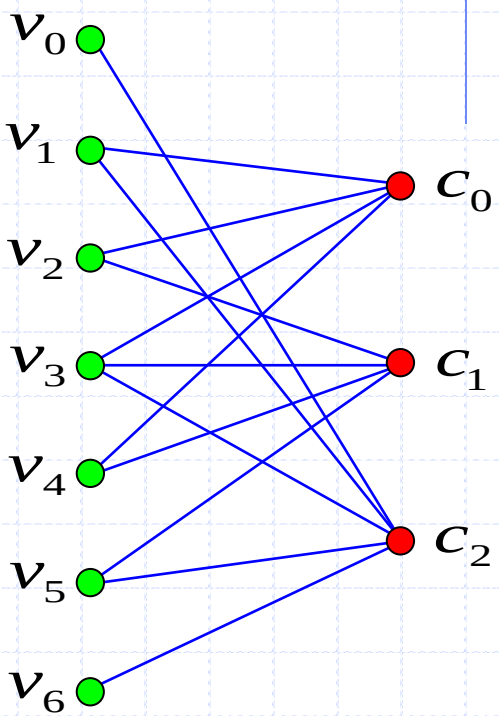
[2] R. G. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.

[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)

- LDPC codes are linear block codes
  - n: code length
  - k: code dimension
  - $r = n - k$: code redundancy
  - **G**: $k \times n$ generator matrix
  - **H**: $r \times n$ parity-check matrix



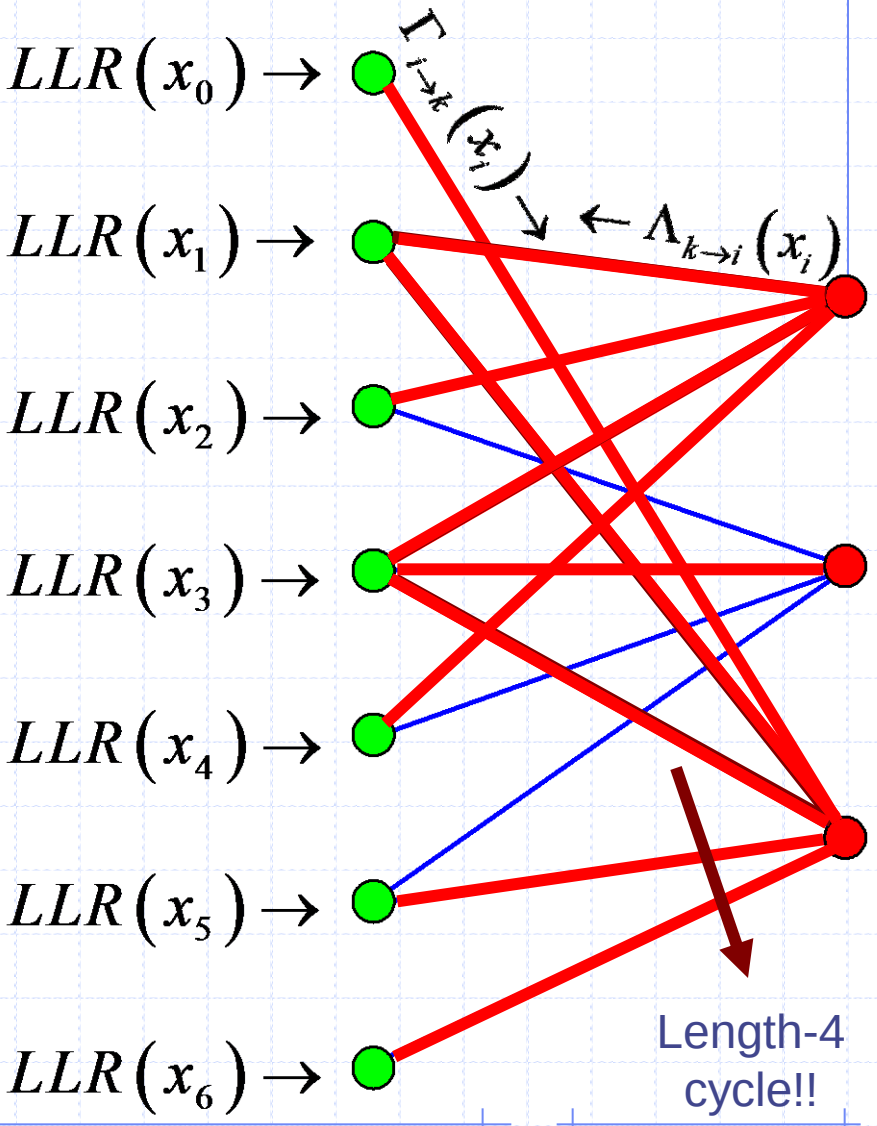
- LDPC codes have parity-check matrices with:
  - Low density of 1 symbols
  - No more than 1 overlapping 1 symbol between any two rows/columns
  - Possibly long local cycles in the associated Tanner graph

LDPC decoding

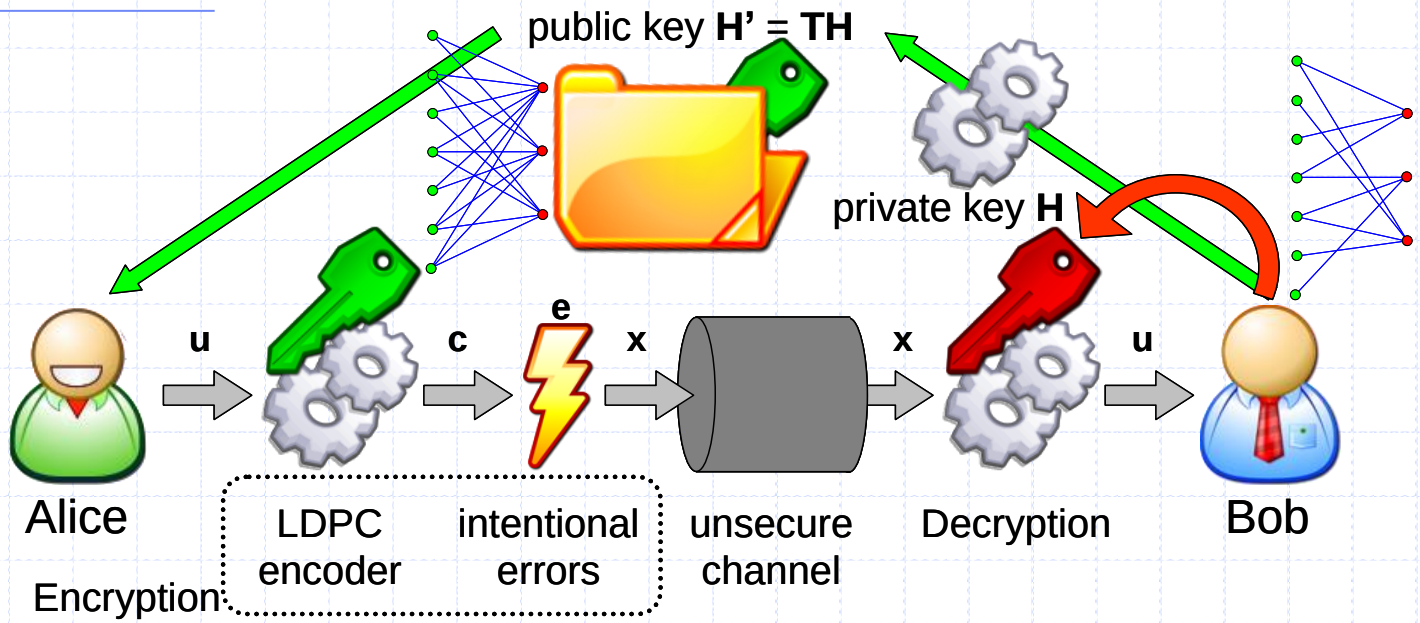
- The LLR-SPA decoder uses likelihood values on the logarithmic scale
- For a random variable U:

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

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



First LDPC-based McEliece PKC [5]



- **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

[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)

- The high density of **H'** helps preventing Eve from the effective usage of LDPC decoders
- 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 [6]:
  1. Its dimension and length are 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 (can be extended to the non-systematic case)
  3. 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 alternative forms:
  - Circulant of blocks
  - Block of circulants

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

Block of circulants form for **H**

- **H** is formed by r_0 x n_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 $r_0 = 1$, 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:

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

- 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:

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

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

QC-LDPC codes based on DFs

- A difference family (DF) 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 p x p circulant matrix (its elements give the positions of the 1 symbols in the first row of the matrix)
- If a difference family with $\lambda = 1$ is used to obtain a QC-LDPC matrix in the form:

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

then **H** is free of length-4 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" (RDF) a random multi-set with the properties of a difference family
- The random-based approach allows to design large families of codes with fixed parameters
- Let us fix n_0, p and d_v (variable nodes degree)
- An estimation of the number of different codes (based on combinatorial arguments) is as follows:

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

QC-LDPC codes based on RDFs (2)

- The number of different codes can be very high:
  - $M(n_0 = 3, d_v = 13, p = 5120) \geq 2^{351}$
  - $M(n_0 = 4, d_v = 13, p = 7168) \geq 2^{497}$
  - $M(n_0 = 4, d_v = 13, p = 8192) \geq 2^{512}$

- The error correction performance of codes based on (n_0, d_v, p)-RDFs is the same, 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!!

LDPC decoding for the McEliece PKC

- LDPC decoding can be accomplished through the Sum-Product Algorithm (SPA) with Log-Likelihood Ratios (LLR)
- This application of LDPC codes can be modeled as 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:

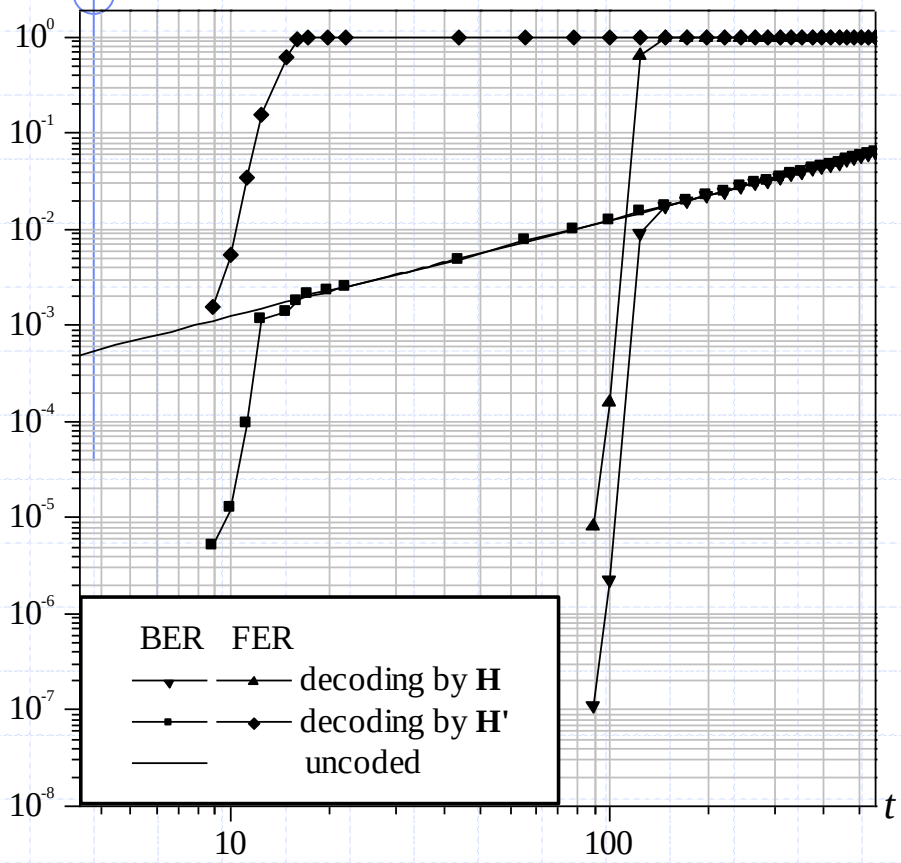
$$LLR(x_i) = \ln\left[\frac{P(x_i = 0 \mid y_i = y)}{P(x_i = 1 \mid y_i = y)}\right]$$

- Applying the Bayes theorem:

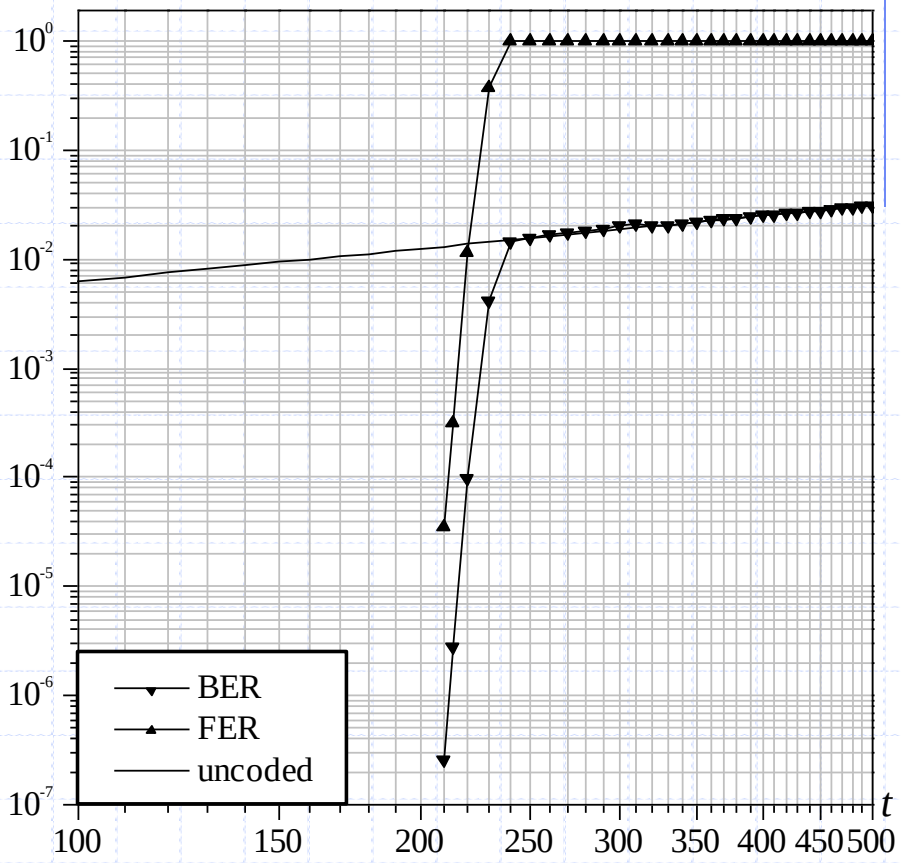
$$LLR(x_i \mid y_i = 0) = \ln\left(\frac{1-p}{p}\right) = \ln\left(\frac{n-t}{t}\right)$$

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

LDPC decoding for the McEliece PKC (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.

First LDPC-based McEliece PKC

- So, in principle, the need for a dense **T** could not be a problem…
- …but is it enough to provide robustness to the LDPC-based system?
- Main issue:
  - Can still be possible to recover the secret representation of the code?

- First approach:
  - Can be possible to recover **H** from **H'** even when **T** is dense?
- Second approach:
  - Disclosing **H'** gives access to the secret code (while in the original version it was hidden through a permutation). Can be possible to recover its sparse representation?

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)

$\mathbf{H' = TH = TZZ^{-1}H = T_dH_d}$

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

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

$\mathbf{P}$ $\qquad$ $\mathbf{Z}$

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

$\mathbf{H}_b$ has the same density as $\mathbf{H}$ **(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 $\mathbf{Z}^*$

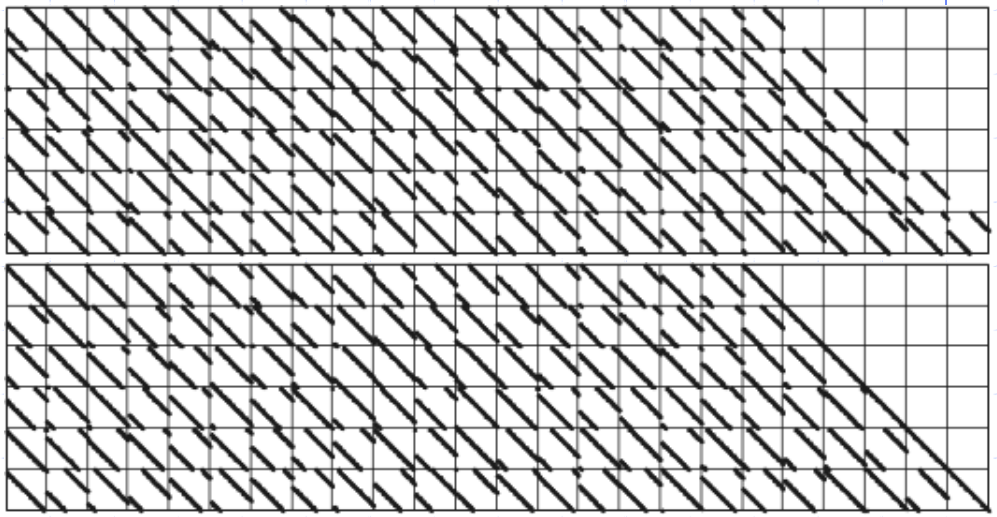
$$\mathbf{H}_d = \mathbf{Z}^{-1}\mathbf{H} = [\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{I}]$$

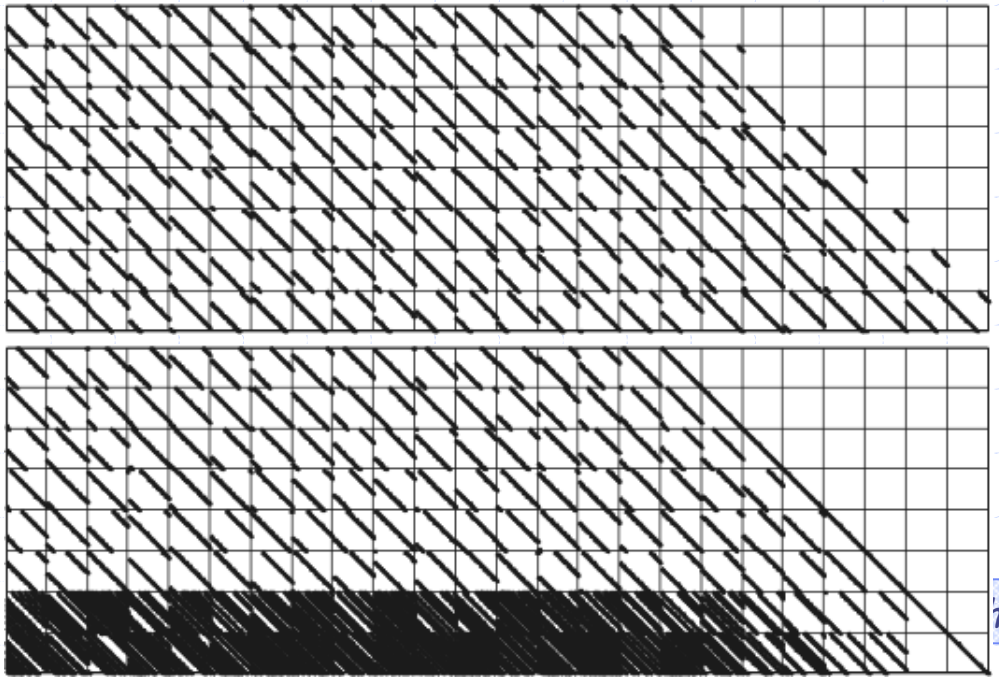
$$\mathbf{H}' = \mathbf{T}_d\mathbf{H}_d = [\mathbf{T}_d\mathbf{Z}^{-1}\mathbf{P} \mid \mathbf{T}_d]$$

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 $n_0 = 24$, $r_0 = 8$, $p = 40$

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 code has, at least, $A_{d_c} \geq r$ codewords with weight $d_c = d_v/(1-R)$
- Since $d_c << n$, we can consider $A_{d_c} \sim r$
- Stern's algorithm searches for low weight codewords through an iterative procedure [8]
- Probability of finding, in one iteration, a (supposed unique) w-weight codeword of the dual code:

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

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

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:

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

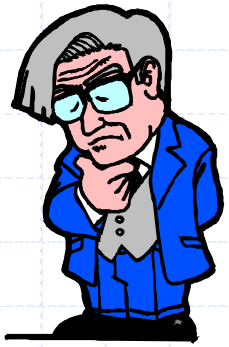
- The total work factor can be estimated as $W = cN$
- Some examples with code rate 3/4:
  - $n = 16000$, $d_v = 13$ → $W = \mathbf{2^{37.5}}$ (min for $g = 3$, $l = 42$)
  - $n = 32000$, $d_v = 17$ → $W = \mathbf{2^{43.7}}$ (min for $g = 3$, $l = 47$)
  - $n = 64000$, $d_v = 21$ → $W = \mathbf{2^{50.4}}$ (min for $g = 3$, $l = 51$)
- Even though long codes (and rather dense matrices) are adopted, the system is highly exposed to a total break!

Lesson learned

- We may be tempted to exploit sparsity of LDPC matrices to reduce the key size
- But sparse public keys are weak
- We can hide their sparsity in the public keys (through structured codes)…
- …but it is still "intrinsic" in the code (and can be recovered by attacking its dual)

so…

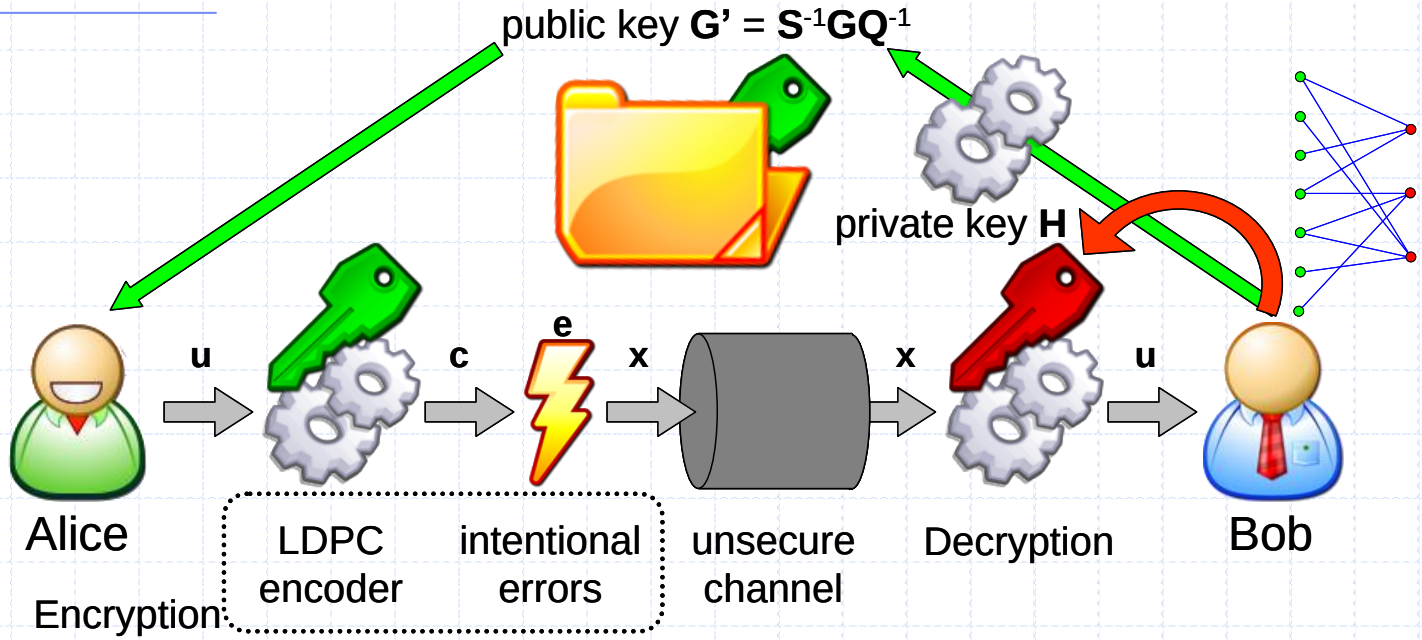
- We should find the way of hiding the code itself, but:
  - How?
  - In such case, are LDPC codes still advantageous?



Some facts

- Hiding the secret code in the public key has some cost
- LDPC codes are able to correct a high number of errors
- Maybe we could trade some error correcting capability for security!

New System Proposal



- **Q** is formed by n_0 x n_0 circulant blocks with size p
- The public code has parity-check matrix $\mathbf{H'} = \mathbf{HQ}^T$
- **Q** has column weight m
- The row weight of **H'** is $\sim m \cdot n_0 \cdot d_v$ → 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

System parameters

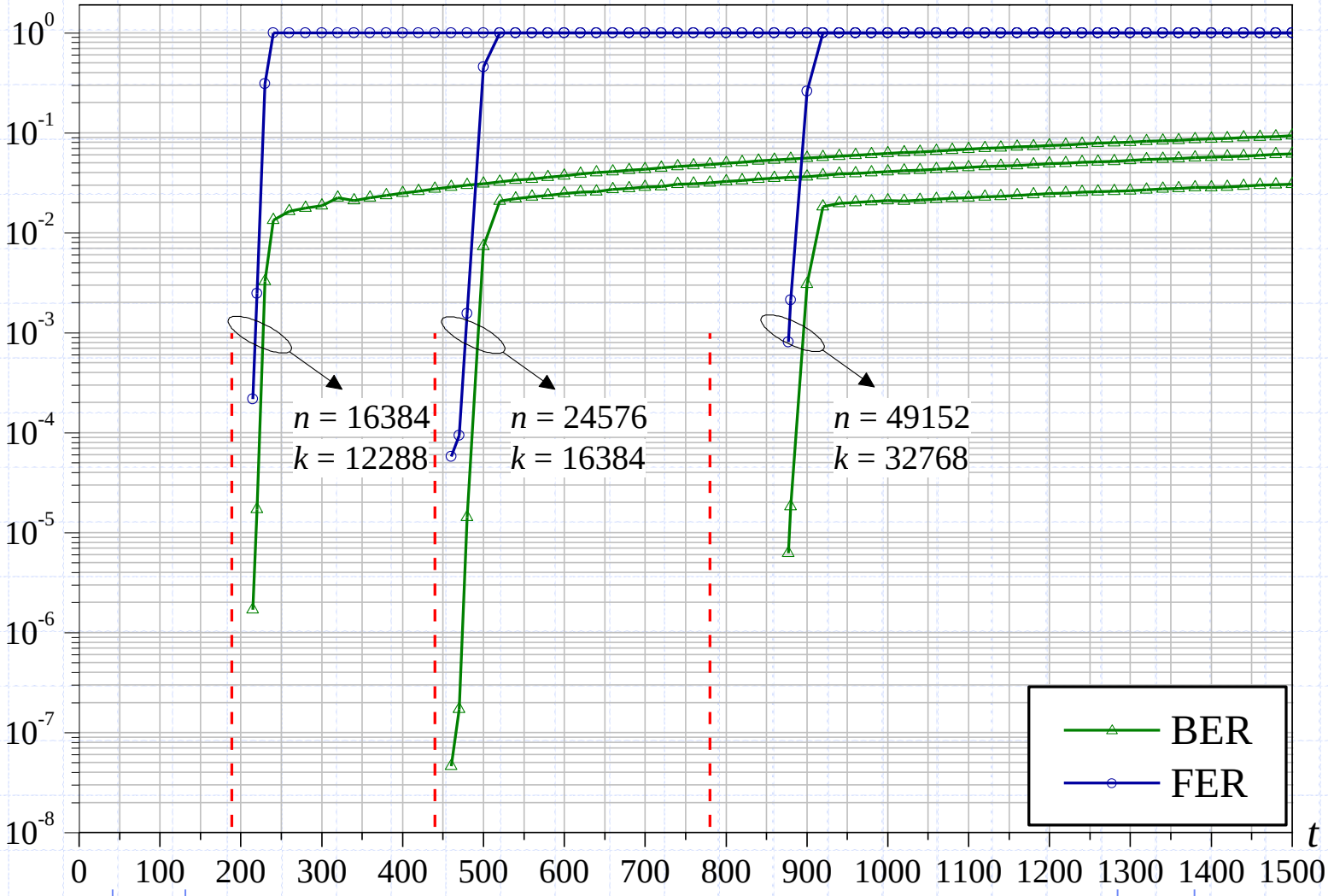
- Some possible choices for the revised system [9]:

System	n_0	d_v	p	m	t'	Key size (bytes)
1	4	13	4096	7	27	6144
2	3	13	8192	11	40	6144
3	3	15	16384	13	60	12288

- The secret LDPC code must be able to correct $t = t'm$ errors
- For the three codes considered:
  - $t = 189$
  - $t = 440$
  - $t = 780$

[9] M. Baldi, "LDPC codes in the McEliece cryptosystem: attacks and countermeasures," in Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes, NATO Science for Peace and Security Series (D: Information and Communication Security), vol. 23, pp. 160-174, IOS Press, 2009.

System parameters (2)



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 matrix sparsity, it is highly probable that the minimum weight approaches $m \cdot n_0 \cdot d_v$
- However, a considerably lower weight provides a sufficiently high work factor for the attack

System	n_0	d_v	p	m	$m \cdot n_0 \cdot d_v$	Max WF	w(WF $\geq 2^{80}$)
1	4	13	4096	7	364	2^{153}	179
2	3	13	8192	11	429	2^{250}	127
3	3	15	16384	13	585	2^{340}	124

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 Stern's algorithm for the search of minimum weight codewords in a linear block code (with no visible structure)

- Complexity of Stern's algorithm can be evaluated in closed form

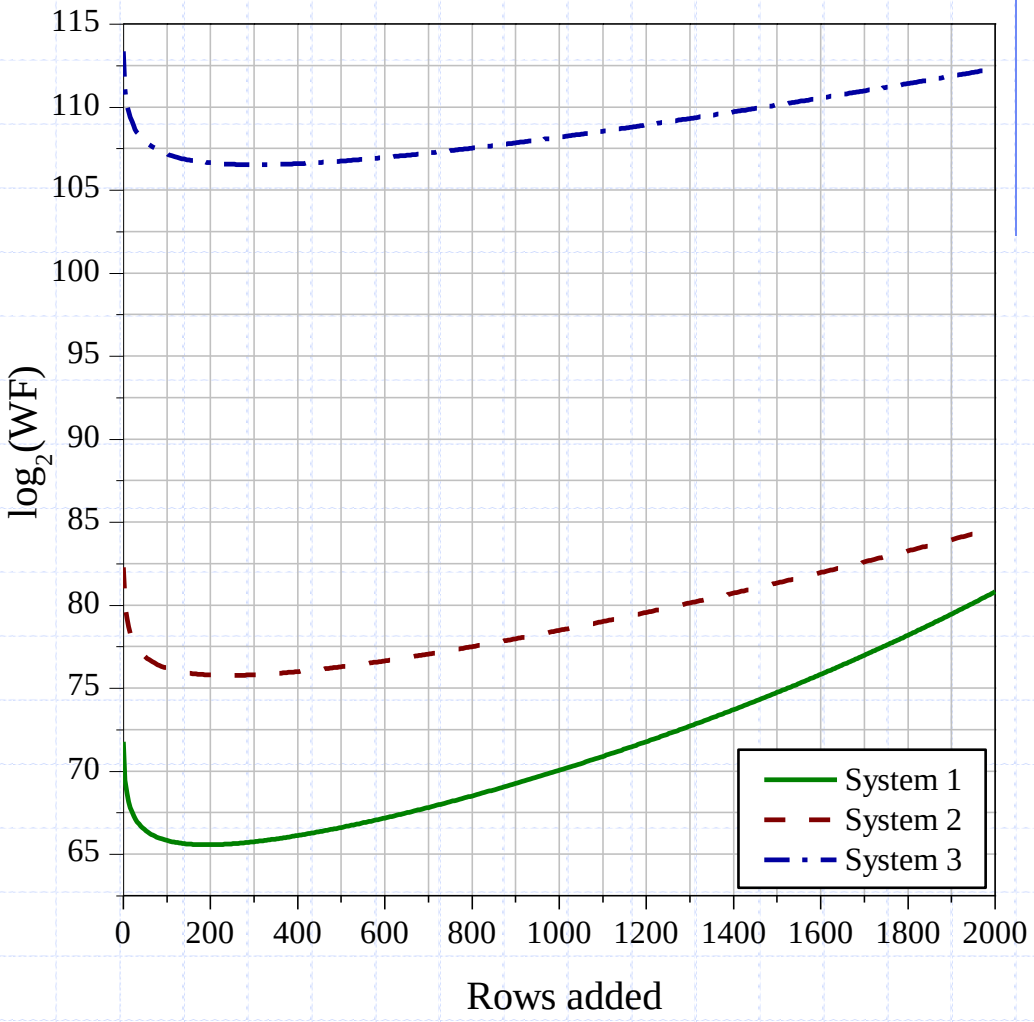
Decoding Attacks (2)

- Recently Stern's algorithm has been further improved [10]
- Estimating the work factor of such modified version is more complex
- For our purposes, it is enough to consider that it could result in a reduction of a factor about 12 ($2^{3.6}$) in the work factor [10]
- An extra speedup results from the quasi-cyclic nature of the codes
- 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

[10] 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 (3)

- The three choices of the system parameters give minimum work factor:
  - $2^{65.6}$
  - $2^{75.8}$
  - $2^{106.5}$
- This is the smallest work factor reached by currently known attacks
- So it can be considered as the security level of the three cryptosystems



What to avoid

- In our first version [11] 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:

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

[11] 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) [12]
- 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 $\mathbf{QS}_{i,j}$, 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:

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

[12] 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 $(\mathbf{Q}\mathbf{S}_{i,0})^{-1} \cdot \mathbf{R}_i$

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

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 [13]
- 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 [13]
- **Q** must be sparse to allow correction of all intentional errors
- A block-diagonal **Q** is weak, so it is advisable to avoid it

[13] 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

Encryption complexity

- Number of binary operations needed for LDPC encoding:
$$C_{\text{mul}}(\mathbf{u} \cdot \mathbf{G}')$$
- Further n operations for addition of the intentional error vector
- Encryption complexity:
$$C_{\text{enc}} = C_{\text{mul}}(\mathbf{u} \cdot \mathbf{G}') + n$$
- Naïve computation: $C_{\text{mul}}(\mathbf{u} \cdot \mathbf{G}') = n \cdot k/2$
- $C_{\text{mul}}(\mathbf{u} \cdot \mathbf{G}')$ can be reduced by exploiting the isomorphism between circulant matrices and polynomials in GF(2)$[x]/(x^p + 1)$:
  - Toom-Cook algorithm
  - Winograd convolution

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

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

- As for LDPC decoding [14]:

$$C_{SPA} = I_{ave} \cdot n \cdot [q \cdot (8d_v + 12R - 11) + d_v]$$

  - I_{ave}: average number of decoding iterations
  - q: number of quantization bits
  - R: code rate

[14] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," Proc. IEEE GLOBECOM '01, San Antonio, TX, 2001

Comparison with other PKCs

	McEliece (1024, 524)	Niederreiter (1024, 524)	RSA 1024-bit mod. public exp. 17	QC-LDPC McEliece 1	QC-LDPC McEliece 2	QC-LDPC McEliece 3
Key Size	67072	32750	256	6144	6144	12288
Rate	0.51	0.57	1	0.75	0.67	0.67
k	524	284	1024	12288	16384	32768
C_{enc}/k	514	50	2402	658	776	1070
C_{dec}/k	5140	7863	738112	4678	8901	12903

- The three QC-LDPC-based systems have shorter keys and higher rates than the original McEliece cryptosystem and the Niederreiter version
- RSA has shorter keys and unitary rate, but higher complexity
- The 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

Some comments

- The adoption of LDPC codes in the McEliece cryptosystem can help to overcome its drawbacks
- But the sparse nature of the LDPC matrices can expose the system to classic and newly developed attacks
- The misuse of sparse transformation matrices can expose the system to total break attacks
- Dense transformation matrices avoid such attacks
- The quasi-cyclic nature of the codes allows to:
  - reduce the key size
  - exploit efficient algorithms for polynomial multiplication

Future work

- The last QC-LDPC-based version proposed has not been still cryptanalyzed by other authors
- This may mean that:
  - Nobody had time to study/attack it
  - Successful attacks are not easy to find
- We could:
  - Optimize the system and its complexity
  - Search for new possible attacks

A possible hint

- We have considered only Stern's algorithm for attempting decoding of a structured (non-algebraic) code
- Do more clever solutions exist for better exploiting the quasi-cyclic nature of the public code?