Using sparse codes in cryptographic primitives

Marco Baldi and Marco Bianchi

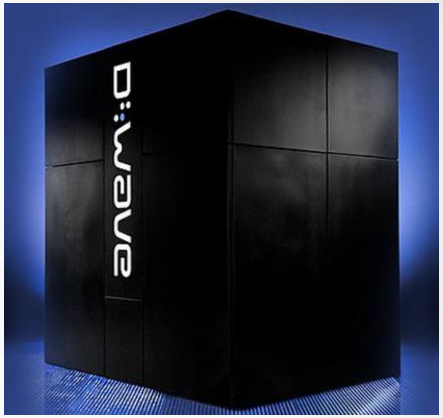
Università Politecnica delle Marche
Ancona, Italy

{m.baldi, m.bianchi}@univpm.it

Code-based Cryptography

- Cryptographic primitives based on the decoding problem (decoding a random-like code)
- McEliece and Niederreiter cryptosystems: public-key cryptosystems based on the decoding problem
- Courtois-Finiasz-Sendrier (CFS) and Kabatianskii-Krouk-Smeets (KKS) systems: digital signature schemes based on the decoding problem

The Quantum Computer Threat



- Quantum computers allow to factorize large integers and to compute discrete logarithms in polynomial time
- They will seriously endanger **RSA**, **DSA**, **ECDSA**…
- ***October 2011***: University of Southern California, Lockheed Martin and D-Wave Systems develop D-Wave One
- ***August 2012***: Harvard Researchers Use D-Wave quantum computer to fold proteins
- ***May 2013***: NASA and Google jointly order a 512 qubit D-Wave Two

McEliece cryptosystem

- Public Key Cryptosystem (PKC) proposed by McEliece in 1978, exploiting the problem of decoding a random linear code

- Private key:

$$\{\mathbf{G}, \mathbf{S}, \mathbf{P}\}$$

  - **G**: 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:

$$\mathbf{G}' = \mathbf{SGP}$$

McEliece cryptosystem (2)

- Encryption map:

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

- Decryption map:

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

all errors are corrected, thus obtaining:

$$\mathbf{u'} = \mathbf{u}\mathbf{S}$$
$$\mathbf{u} = \mathbf{u'}\mathbf{S}^{-1}$$

Goppa codes and key size

- Any degree-t (irreducible) polynomial generates a different Goppa 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, which are reduced to ***rk*** bits with a CCA2 secure conversion [1]
- Despite this, key size is **large** and grows **quadratically** with the code length

[1] 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 capacity-achieving codes under Belief Propagation decoding
- They allow a random-based design, which results in large families of codes with similar characteristics
- The low density of their parity-check matrices could be used to reduce the key size, but this exposes the system to key recovery attacks
- Hence, , the permutation matrix **P** must be replaced with a denser matrix **Q** which makes the public code denser as well

[2] 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.

[3] 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

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

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

- A more efficient way to reduce the key size is to use dense public keys but with structured LDPC codes

- QC-LDPC codes with **H** as a row of circulant matrices:

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

completely described by its first row

- Systematic generator matrix:

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

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

[5] 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

Key Size and 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

[6] M. Baldi, M. Bianchi, F. Chiaraluce, "Security and complexity of the McEliece cryptosystem based on QC-LDPC codes", IET Information Security, in press, http://arxiv.org/abs/1109.5827

Comparison with Goppa codes

- 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 !

- For the **QC-LDPC** code-based system, the key size **grows linearly** with the code length, due to the **quasi-cyclic** nature of the codes, while with Goppa codes it grows **quadratically**

MDPC code-based variant

- 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
- Using randomly designed MDPC codes 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

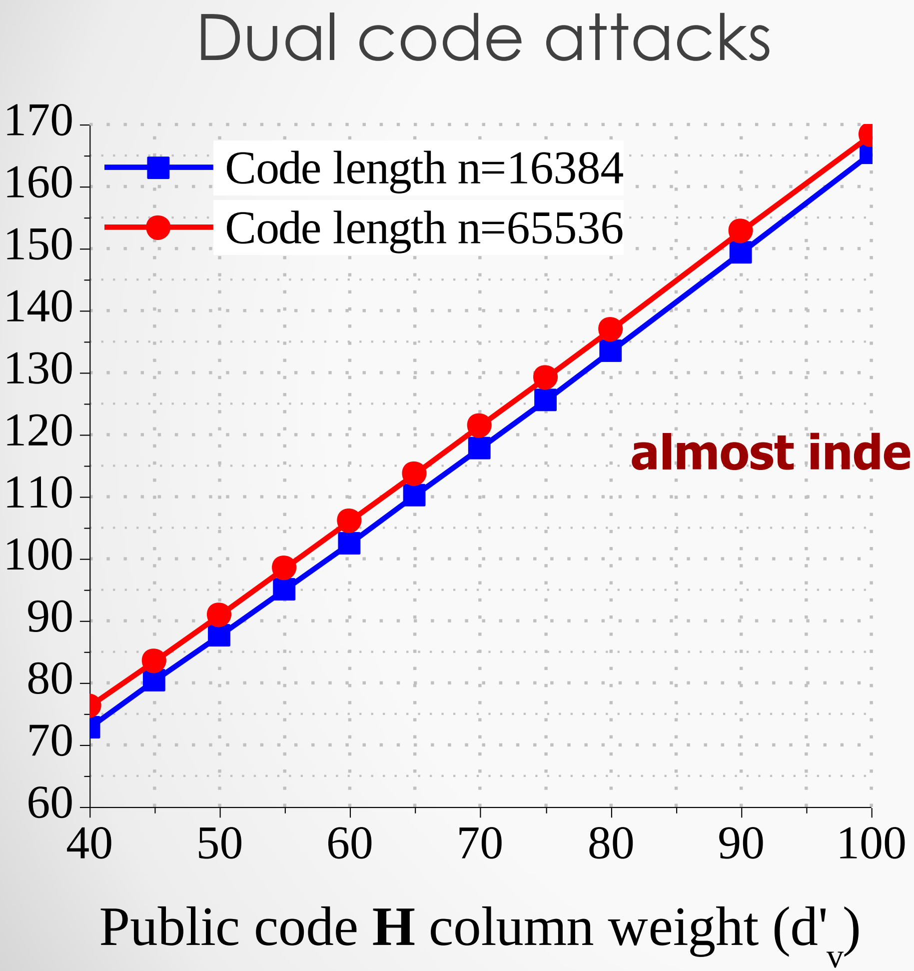
[7] R. Misoczki, J.-P. Tillich, N. Sendrier, P. S. L. M. Barreto, "MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes", cryptology ePrint archive, http://eprint.iacr.org/2012/409

Code Density Optimization

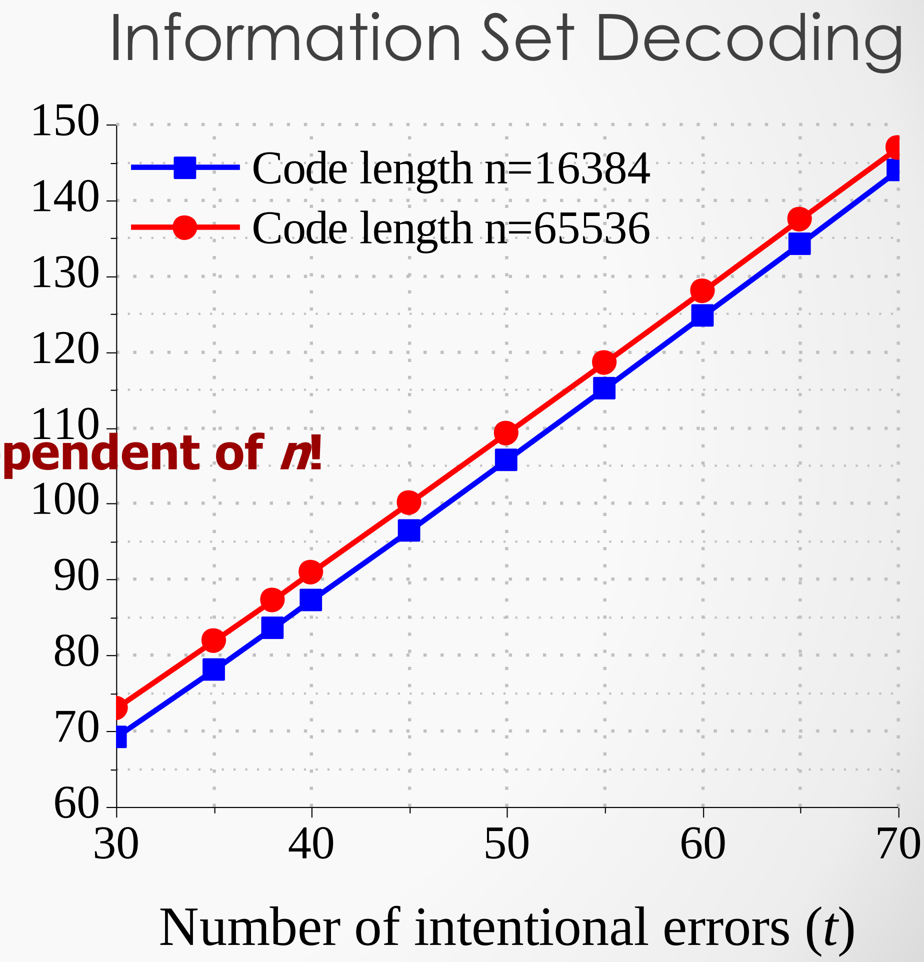
- To use LDPC codes securely, the permutation matrix **P** must be replaced with a matrix **Q** having average row and column weight m, $1 < m << n$
- This avoids the existence of a sparse (and hence weak) representation for the public code…
- …but also increases the number of intentional errors by a factor up to m
- The choice of m can be optimized by using simple tools

[8] M. Baldi, M. Bianchi, F. Chiaraluce, "Optimization of the parity-check matrix density in QC-LDPC code-based McEliece cryptosystems", to be presented at IEEE ICC 2013, http://arxiv.org/abs/1303.2545

Attacks Work Factor ($\log_2$)



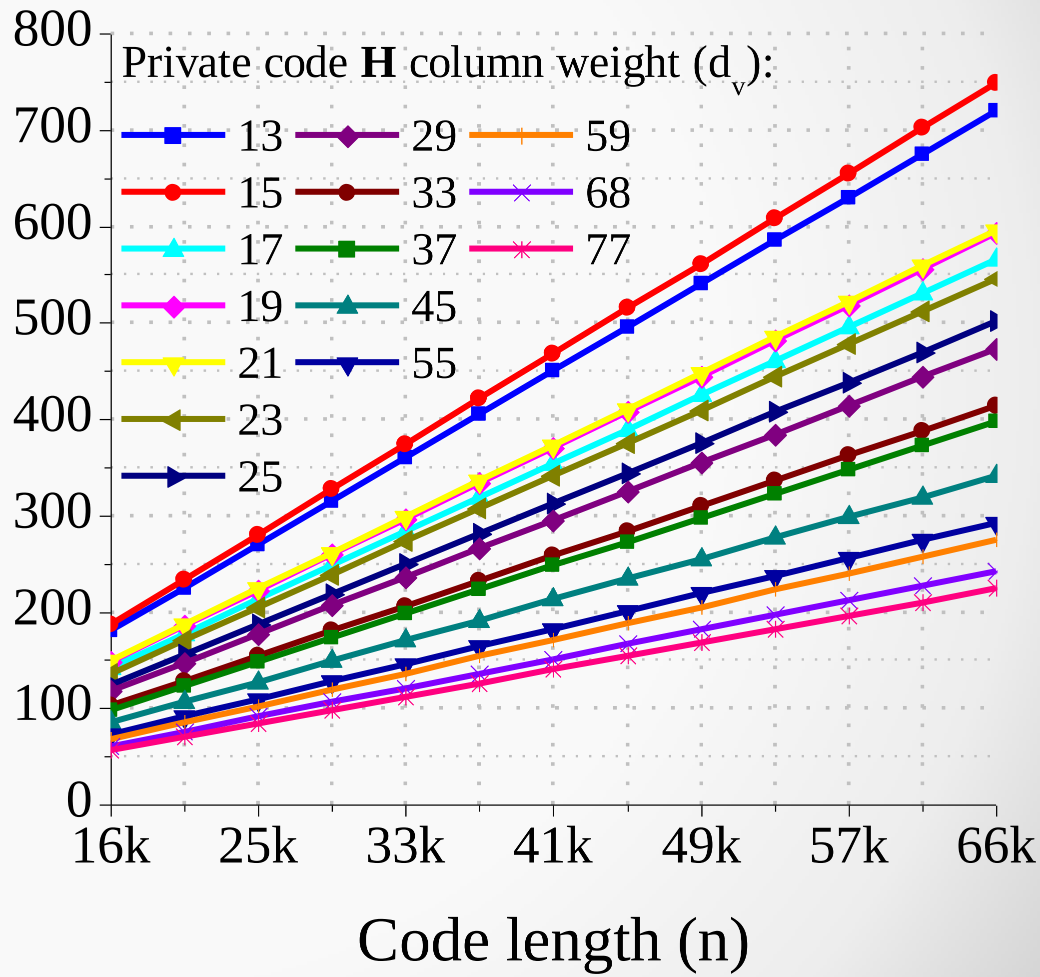
almost independent of *n*!



Private Code Density Design

- Design procedure:
  - Fix the security level
  - Obtain d_v' and t
  - Fix n
  - Find m such that there is a length-n code with $d_v = d_v'/m$ and able to correct $t' = tm$ errors
- The higher m, the lower decoding complexity
- Hence, LDPC codes are advantageous over MDPC codes

Number of correctable errors



Irregular Codes

- Irregular LDPC codes achieve higher error correction than regular ones
- This can be exploited to increase the system efficiency by reducing the code length…
- …although the QC structure and the need to avoid enumeration impose some constraints

160-bit security

QC-LDPC code type	n_0	d_v'	t	d_v	n	Key size (bytes)
regular	4	97	79	13	54616	5121
irregular	4	97	79	13	46448	4355 -15%

[9] M. Baldi, M. Bianchi, N. Maturo, F. Chiaraluce, "Improving the efficiency of the LDPC code-based McEliece cryptosystem through irregular codes", to be presented at IEEE ISCC 2013

Code Based Signature Schemes

- Standard signature schemes rely on classic cryptographic primitives as RSA and DSA
- They will be endangered by quantum computers as well as RSA and DSA
- Code-based cryptographic primitives could be used for digital signatures
- Two main schemes were proposed for code based signatures:
  - Kabatianskii-Krouk-Smeets (KKS)
  - Courtois-Finiasz-Sendrier (CFS)

CFS (1)

- Close to the original McEliece Cryptosystem
- It is based on Goppa codes

- Public:
  - A hash function $\mathcal{H}(D)$
  - A function $\mathcal{F}(C,h)$ able to transform the hash h into a correctable syndrome through the code C
- Initialization:
  - The signer chooses a Goppa code G able to decode t errors and a parity check matrix $\boldsymbol{H}$ that allows decoding
  - He chooses also a scrambling matrix $\boldsymbol{S}$ and publishes $\boldsymbol{H'}=\boldsymbol{SH}$

CFS (2)

- Signing the document D:
  - The signer computes $s=\mathcal{F}(G,\mathcal{H}(D))$
  - $s' = s(\boldsymbol{S}^T)^{-1}$
  - He decodes the syndrome s' through the secret parity check matrix $\boldsymbol{H}$: $e\boldsymbol{H}^T=s'$
  - The error e is the signature

- Verification:
  - The verifier computes $s=\mathcal{F}(G,\mathcal{H}(D))$
  - He checks that $e\boldsymbol{H}'^T=e(\boldsymbol{H}^T\boldsymbol{S}^T)= s(\boldsymbol{S}^T)^{-1}\boldsymbol{S}^T =s$

CFS (3)

- The main problem is to find an efficient function $\mathcal{F}(C,h)$
- For Goppa codes two techniques were proposed:
  - Appending a counter to $\mathcal{H}(D)$ until a valid signature is generated
  - Performing complete decoding
- Both these methods require codes with very special parameters:
  - very high rate
  - very small error correction capability

CFS (4)

- Codes with small t and high rate could be decoded, with good probability, through the Generalized Birthday Paradox Algorithm (GBA)
- In GBA, the columns of ***H'*** summing in the desired vector are selected by partial zero-summing
- Decoding is not guaranteed (it is guaranteed in ISD decoding)
- GBA works with random vectors, for code-based algorithms the vectors are ***H'*** columns: lack of randomness requires extra-effort
- However, for CFS parameters, the average correct decoding probability is astonishing close to 1

LDGM codes

- LDGM codes are codes with low density in the generator matrix **G**
- They are known for other applications like concatenated decoding
- We will consider LDGM generator matrix in the form:

$$\boldsymbol{G} = [\boldsymbol{I}_k \mid \boldsymbol{A}]$$

- A valid parity check matrix is:

$$\boldsymbol{H} = [\boldsymbol{A}^T \mid \boldsymbol{I}_r]$$

- **G** row weight is w_G

Idea

- Using $\boldsymbol{H}$ in triangular form, it is trivial to find a vector e such that $e\boldsymbol{H}^T=s$, for every s: it is just $e=[\boldsymbol{0}\,|\,s]$
- In this simplified scenario e has maximum weight equal to r
- Differently from CFS not only decodable syndrome are used (every weight is permitted for s)
- We need to check that e has a relatively low weight, otherwise it is easy to find e' such that $e'\boldsymbol{H}^T=s$ and the weight of e' is about $n/2$
- I.e.

$$e' = \left(\left(\boldsymbol{H}^T(\boldsymbol{H}\,\boldsymbol{H}^T)^{-1}\right)s^T\right)^T$$

Proposed Scheme

- Use LDGM codes, fixing a target weight w_c
- Use $\boldsymbol{H}$ with an identity block somewhere (i.e. on the right end)
- $\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$
- $\boldsymbol{S}$ is a sparse, not singular, matrix with row and column weight m_s
- $\boldsymbol{Q} = \boldsymbol{R} + \boldsymbol{T}$
- $\boldsymbol{T}$ is a sparse, not singular, matrix with row and column weight m_T
- $\boldsymbol{R} = \boldsymbol{a}^T\boldsymbol{b}$, with $\boldsymbol{a},\boldsymbol{b}$ ($z \times r$) matrices
- Our $\mathcal{F}(h,p)$ function has to transform an hash into a vector s such that $\boldsymbol{b}s=\boldsymbol{0}$ depending on the parameter p

Signing

- The signer chooses secret ***H*, *Q*** and ***S***
- He computes $s=\mathcal{F}(\mathcal{H}(D),p)$, it requires 2^z attempts in the average case
- $s' = \boldsymbol{Q}s$
- He decodes the syndrome s' through the secret parity check matrix ***H***: $e\boldsymbol{H}^T=s'$, that is $e=[\boldsymbol{0}|s']$
- He chooses a random low-weight codeword c having weight w_c that is (close to) a small multiple of w_G, w_c is made public
- The signature is the couple $[p, e'=(e+c)\boldsymbol{S}^T]$

Verification

- The verifier computes the vector $s=\mathcal{F}(\mathcal{H}(D),\mathrm{p})$ having weight w
- The verifier checks that the weight of e’ is equal or smaller than $(m_T w+w_c)m_s$
- He checks that $e'\boldsymbol{H}'^{\boldsymbol{T}} = s$

Rationale

- Removing the request for high rate codes makes GBA unfeasable
- ISD algorithms are not able to find errors of moderately high weight
- The insertion of the codeword c is needed to make the system not-linear (it becomes an affine map)
- The use of **Q** reinforces the system against the most dangerous known attack (Support Intersection Attack)
- We can use Quasi Cyclic codes in order to keep the public key size small

Parameters

SL (bits)	n	k	p	w	w_g	w_c	z	m_T	m_S	A_{w_c}	N_s	S_k (KiB)
80	9800	4900	50	18	20	160	2	1	9	$2^{82.76}$	$2^{166.10}$	117
120	24960	10000	80	23	25	325	2	1	14	$2^{140.19}$	$2^{242.51}$	570
160	46000	16000	100	29	31	465	2	1	20	$2^{169.23}$	$2^{326.49}$	1685

- For the same security levels (SL), CFS requires Key Sizes (S_k) in the range 1.25-20 MiB (parallel version) or greater than 52 MiB (standard version)

ESCAPADE research project

http://escapade.dii.univpm.it