# Using LDGM codes and Sparse Syndromes to Achieve Digital Signatures

M.Baldi*, M. Bianchi*, F. Chiaraluce*,
J. Rosenthal**, D. Schipani**

*Università Politecnica delle Marche
Ancona, Italy

**University of Zurich

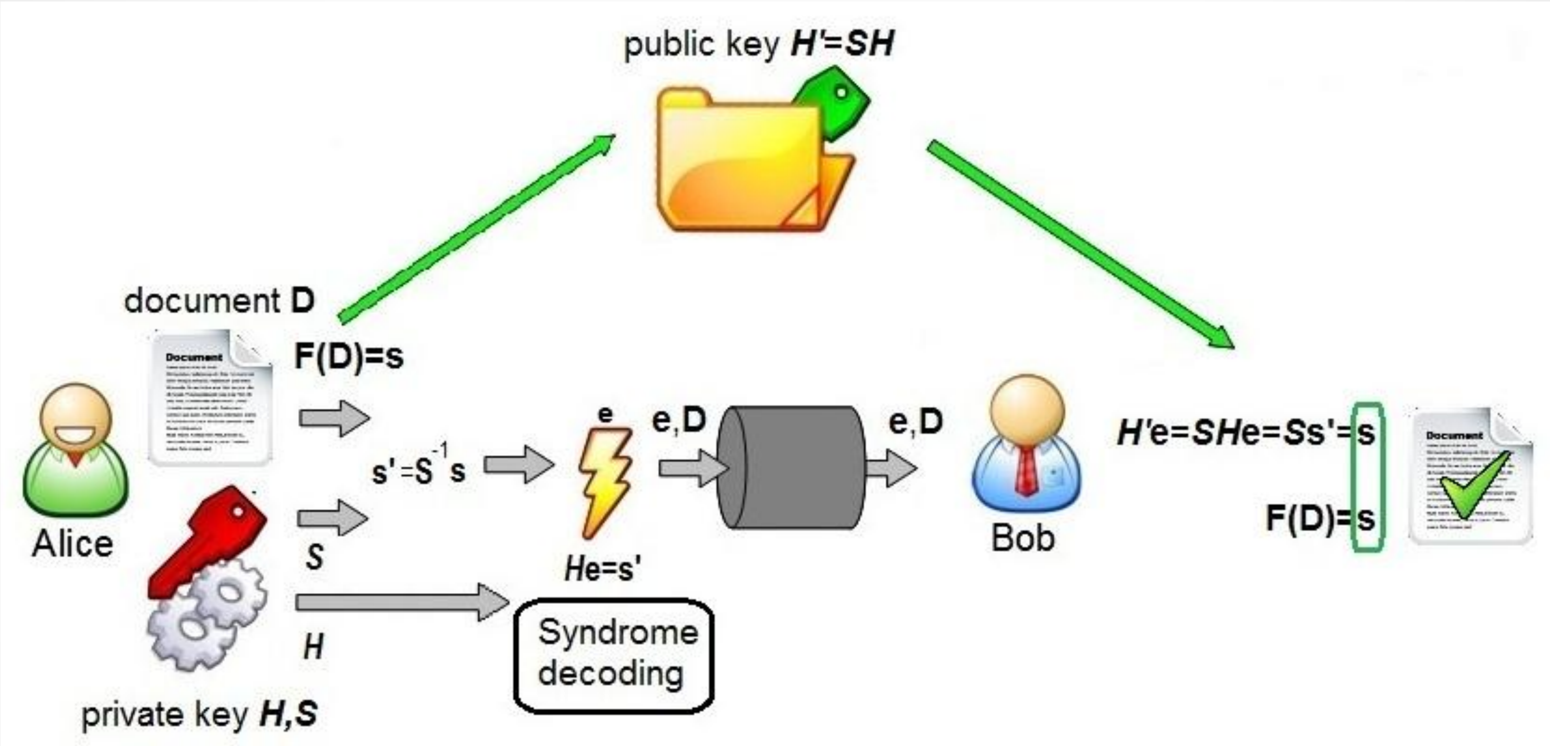
# Code Based Signature Schemes

- 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)

# KKS

- The KKS scheme is quite different from traditional code based cryptosystem
- It is based on two codes, one selecting the subset support of the other
- It does not require a decoding phase
- Majour issue: there is an attack for almost all of the parameter sets

# CFS Sketch




Just a scheme!

A lot of details are to be considered

# CFS (1)

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

- Public:
  - A hash function $\mathcal{H}(D)$
  - A function $\mathcal{F}(h, \ldots)$ able to transform the hash $h$ into a vector that becomes a correctable syndrome for the secret code C, when multiplied *by* $\boldsymbol{S}^{-1}$
- 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 $s=\mathcal{F}(\mathcal{H}(D),\ldots)$
  - $s' = s(\boldsymbol{S}^T)^{-1}$
  - 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 $s=\mathcal{F}(\mathcal{H}(D),\ldots)$
  - 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 $\mathcal{F}(h,\ldots)$ in such a way not to endanger the system
- 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)

- It is particularly efficient when we can choose among more than one correct answers (multiple instances)

- In GBA, the columns of ***H'*** summing in the desired vector are selected by partial zero-summing

# CFS (5)

- Using GBA, 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 the original CFS parameters, the average correct decoding probability is quite high

# 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

- We need a way to perform syndrome decoding without imposing too many restrictions on code parameters and error weight

- 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 $r$ (the redundancy of the code)

# Idea (2)

- Differently from CFS not only decodable syndrome are used
- However it is simple to impose that syndromes are decodable from the secret codes (just impose a maximum syndrome weight $w$ equal to the code error correction capability)
- It is not straightforward to ensure that those syndromes are uniquely decodable through the public code
- We need to check that $e$ has a relatively low weight, otherwise it is easy to find $e'$ such that $e'\mathbf{H}'^{T}=s$ and the weight of $e'$ is about $n/2$
- I.e.

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

# Proposed Scheme

- Use LDGM codes, fixing a target weight $w_c$
- Use ***H*** with an identity block somewhere (i.e. on the right end)
- $\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$
- ***S*** is a sparse, not singular, matrix with row and column weight $m_s$

# The Q-matrix

- $\boldsymbol{Q} = \boldsymbol{R}+\boldsymbol{T}$
- $\boldsymbol{T}$ is a sparse, not singular, matrix with row and column weight $m_T$
- $\boldsymbol{R}$ is build upon two matrices, $\boldsymbol{a}$ and $\boldsymbol{b}$ having dimension (z x $r$)
- 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 $\boldsymbol{H}$: $e\boldsymbol{H}^T=s'$, 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 $e'\boldsymbol{H}'^{T} = s$

# Using QC-Codes

- The scheme can be designed using Quasi-Cyclic codes as already proposed for QC-LDPC based McEliece Cryptosystem

$$G_{QC} = \begin{bmatrix} C_{0,0} & C_{0,1} & C_{0,2} & \cdots & C_{0,n_0-1} \\ C_{1,0} & C_{1,1} & C_{1,2} & \cdots & C_{1,n_0-1} \\ C_{2,0} & C_{2,1} & C_{2,2} & \cdots & C_{2,n_0-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{k_0-1,0} & C_{k_0-1,1} & C_{k_0-1,2} & \cdots & C_{k_0-1,n_0-1} \end{bmatrix}$$

- If the circulant blocks have dimension *l* x *l*, it implies factor *l* reduction in the key dimension

# Rationale

- Removing the request for high rate codes makes GBA unfeasible even taking advantage of the quasi-cyclic nature of the codes
- The known ISD algorithms are not able to find errors of moderately high weight in reasonable time for the proposed parameters
- The insertion of the codeword c is necessary to make the system not-linear (it is an affine map)
- The use of **Q** reinforces the system against the most dangerous known attack (Support Intersection Attack)

# 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)

# Future Works

- Build new attacks
- Is it possible to increase the ISD efficiency taking advantage of the QC nature of the codes?
- Is it possible to reduce the problem to a known NP-problem? (…we know it is not the end of the story)