An Efficient Attack on a Code-based Signature Scheme

Aurélie Phesso [1] Jean-Pierre Tillich[2]

[1]University of Bordeaux
France.
[2]SECRET Project - INRIA Paris

March 9, 2016

Introduction

Syndrome decoding problem

Input : $\boldsymbol{s} \in \{0,1\}^r$, $\boldsymbol{w} \in \mathbb{N}$ and $\boldsymbol{H} \in \{0,1\}^{r \times n}$
Output : $\boldsymbol{e} \in \{0,1\}^n$, weight$(\boldsymbol{e}) \leq \boldsymbol{w}$ and $\boldsymbol{H}\boldsymbol{e}^T = \boldsymbol{s}^T$

Hardness problem

This problem is **NP**-complete

In the case of the signature

Input : $\boldsymbol{s} = \mathit{Hash}(m) \in \{0,1\}^r$ and $\boldsymbol{w} \in \mathbb{N}$, $\boldsymbol{H} \in \{0,1\}^{r \times n}$
Output : $\boldsymbol{e} \in \{0,1\}^n$: signature, weight$(\boldsymbol{e}) \leq \boldsymbol{w}$ and $\boldsymbol{H}\boldsymbol{e}^T = \mathit{Hash}(m)^T$

Introduction

We can solve this problem only for these codes :

- High-rate Goppa code
- Polar code
- Low Density Generator Matrix code (LDGM)
- Convolutional code

Courtois Finiasz Sendrier signature scheme (CFS)

CFS

The first practical signature scheme based on high-rate Goppa codes.

Courtois Finiasz Sendrier signature scheme (CFS)

CFS

The first practical signature scheme based on high-rate Goppa codes.

But this scheme has drawbacks:

Goppa(m,t)	Public key size	Signature cost	Security
(m, t)	$K = 2^m mt$	$(mt)^2 t!$	$\lambda = 2^{tm/2}$

  For t fixed $\Rightarrow \lambda = K^{t/2}$
- The hypothesis used to give a security proof was not met

New signature scheme proposal

Proposition

Baldi et al. propose a new signature scheme using a Low Density Generator Matrix code (LDGM) .

Definition (LDGM)

The generator matrix $\boldsymbol{G}$ of this code contains only a few 1's compared to the number of 0's.

Remark

In order to reduce keysize, they use QC-LDGM code.

The basic idea of this scheme

Input : $\boldsymbol{s} \in \{0,1\}^r$, weight($\boldsymbol{s}$) = $\boldsymbol{w}$ and $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{P} & \boldsymbol{I} \end{pmatrix}$

it is easy to find a signature $\boldsymbol{e}$ with weight($\boldsymbol{e}$) $\leq \boldsymbol{w}$ such that :

$$\boldsymbol{H}\boldsymbol{e}^T = \boldsymbol{s}^T$$

it is sufficient to take :

$$\boldsymbol{e} = \underbrace{00\cdots0}_{k} || \boldsymbol{s}$$

Obtaining $\boldsymbol{s}$ with weight $\boldsymbol{w}$

$$\mathcal{F} : \quad \{0,1\}^r \rightarrow \{x \in \{0,1\}^r \quad | \quad weight(x) = \boldsymbol{w}\}$$

Strategy to hide $\boldsymbol{H}$

To hide $\boldsymbol{H}$ the authors of this scheme introduced two new matrices :

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$$

Strategy to hide $\boldsymbol{H}$

To hide $\boldsymbol{H}$ the authors of this scheme introduced two new matrices :

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$$

- $\boldsymbol{S}$: sparse matrix with average row and column weight $m_S << n$

Strategy to hide $\boldsymbol{H}$

To hide $\boldsymbol{H}$ the authors of this scheme introduced two new matrices :

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$$

- $\boldsymbol{S}$: sparse matrix with average row and column weight $m_S << n$
- $\boldsymbol{Q}$: weight controlling matrix insures that $\boldsymbol{s}'^T = \boldsymbol{Q}\boldsymbol{s}^T$ has a small weight :

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

  - $\boldsymbol{T}$: sparse matrix with row and column weight $m_T << n$
  - $\boldsymbol{R}$: dense matrix with very low-rank s.t good probability to find $\boldsymbol{R}\boldsymbol{s}^T = 0$

Signature generation

To sign a message **m** :

1. Take a counter i
2. $\boldsymbol{s}_i = \mathcal{F}(Hash(m\|i))$ until finding $\boldsymbol{R}\boldsymbol{s}_i^T = 0$
3. Compute private syndrome

$$\boldsymbol{s}'^T = \boldsymbol{Q}\boldsymbol{s}_i^T$$

4. Build

$$\boldsymbol{e} = \underbrace{00\cdots0}_{k}||\boldsymbol{s}'$$

Signature generation

To sign a message **m** :

1. Take a counter i
2. $\boldsymbol{s}_i = \mathcal{F}(Hash(m\|i))$ until finding $\boldsymbol{R}\boldsymbol{s}_i^T = 0$
3. Compute private syndrome

$$\boldsymbol{s}'^T = \boldsymbol{Q}\boldsymbol{s}_i^T$$

4. Build

$$\boldsymbol{e} = \underbrace{00\cdots0}_{k}\,||\,\boldsymbol{s}'$$

5. Select a random codeword $\boldsymbol{c} \in \mathscr{C}_{LDGM}$ with small weight w_c
6. The signature is (σ, i)

$$\boldsymbol{\sigma} = (\boldsymbol{e} + \boldsymbol{c})\boldsymbol{S}^T. \tag{1}$$

Verification

The verifier computes

$$\boldsymbol{s}^* \stackrel{\text{def}}{=} \mathcal{F}(Hash(\boldsymbol{m}||i))$$

and checks

$$\boldsymbol{H}'\boldsymbol{\sigma}^T = \boldsymbol{s}^{*T} \quad \text{and} \quad weight(\boldsymbol{\sigma}) \leq (m_T w + w_c)m_S$$

If this is not the case, the signature is discarded.

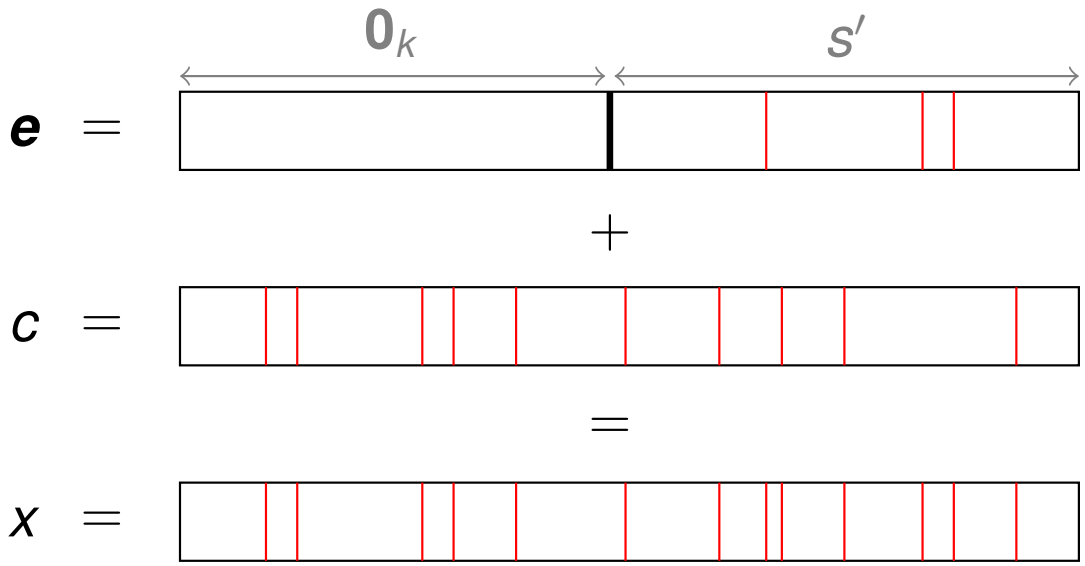
How to break this scheme

- We will show how an attacker can break this scheme using 100 000 signatures
- Idea : Some of the bits of the signatures are correlated
- An attacker can recover enough of the secret structure to forge new signatures

Notation

Let us define :

$$\boldsymbol{x} = (x_1 \dots x_n) \stackrel{\text{def}}{=} (\mathbf{0}_k || \boldsymbol{s}') + \boldsymbol{c}.$$



Notation

We have :

$$\boldsymbol{\sigma} = \boldsymbol{x}\boldsymbol{S}^T$$

Let us first observe that :

$$\sigma_i = \sum_{j:S_{ij}=1} x_j \tag{2}$$

x =

prob$(x_i = 1)$ is close to 0

Example of Correlation

σ_i and σ_j are correlated if they share a common x_t

Correlated

σ_i and σ_j are correlated :

$$\begin{cases} \sigma_i = x_1 + x_3 \\ \sigma_j = x_2 + x_3 + x_4 \end{cases}$$

Example of Correlation

σ_i and σ_j are correlated if they share a common x_t

Correlated

σ_i and σ_j are correlated :

$$\begin{cases} \sigma_i = x_1 + x_3 \\ \sigma_j = x_2 + x_3 + x_4 \end{cases}$$

Independent

σ_i and σ_j are independent :

$$\begin{cases} \sigma_i = x_1 + x_3 \\ \sigma_j = x_{13} + x_{27} + x_{40} \end{cases}$$

Proposition

Proposition

Let X_1, X_2, X_3 be independent Bernoulli variables such that

$$\mathbf{prob}(X_i = 1) = p_i$$

$$\sigma_1 \stackrel{\text{def}}{=} X_1 + X_3 \quad \text{and} \quad \sigma_2 \stackrel{\text{def}}{=} X_2 + X_3$$

Then if

$$p_3 \notin \{0, 1\} \quad \text{and} \quad p_1, p_2 \neq \frac{1}{2}$$

We have that σ_1 and σ_2 are correlated with

$$\text{Cov}(\sigma_1, \sigma_2) \stackrel{\text{def}}{=} \mathbb{E}(\sigma_1\sigma_2) - \mathbb{E}(\sigma_1)\mathbb{E}(\sigma_2) = p_3(1 - p_3)(1 - 2p_1)(1 - 2p_2)$$

Estimating $\mathrm{Cov}(\sigma_i, \sigma_j)$

$\mathrm{Cov}(\sigma_i, \sigma_j)$ estimated by a large number of signatures :

σ_i , σ_j	Independent	Correlated
$\mathrm{Cov}(\sigma_i, \sigma_j)$	$-3.47 \cdot 10^{-4}$	$9.11 \cdot 10^{-3}$

Recovering $\boldsymbol{S}$ permuted

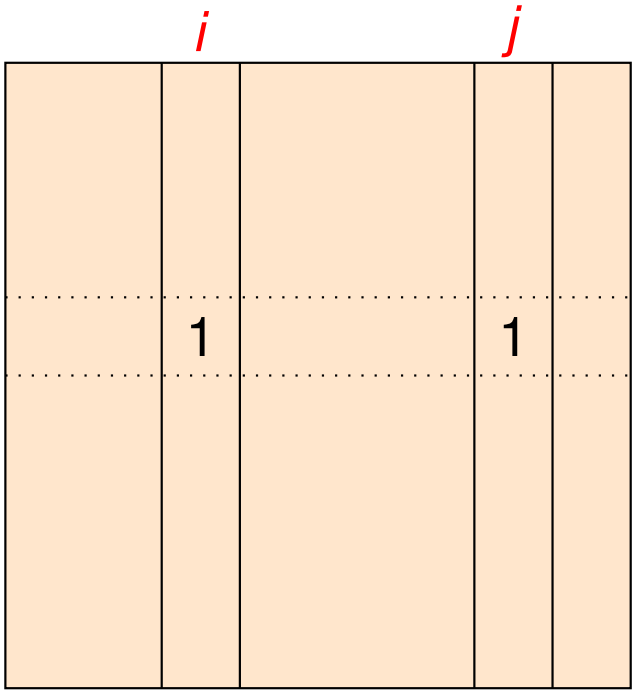
Computing all $\mathrm{Cov}(\sigma_i, \sigma_j)$ allows to recover most of rows $\boldsymbol{S}^T$. We can recover :

$$\boldsymbol{S}_p = \boldsymbol{S}\boldsymbol{\Pi}$$

σ_i and σ_j are correlated implies that :

$$\exists k \quad \text{s.t} \quad \boldsymbol{S}^T[k, i] = \boldsymbol{S}^T[k, j] = 1$$

$\boldsymbol{S}^T =$



Second source of correlations

Let us remember that :

$$\boldsymbol{\sigma} = (\boldsymbol{e} + \boldsymbol{c})\boldsymbol{S}^T \quad \text{where} \quad \boldsymbol{c} = \sum_{s=1}^{\boldsymbol{w}_c/m_G} g_{i_s}$$

$\boldsymbol{G}$ generator matrix of the LDGM code.

Notation

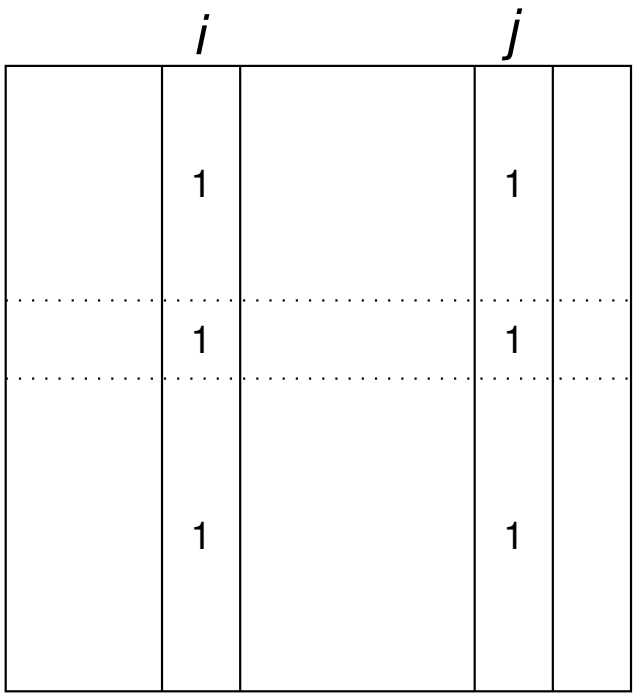
$$\boldsymbol{G}' = \boldsymbol{G}\boldsymbol{S}^T$$

Second source of correlations

σ_i and σ_j are correlated when there are several k_s :

$$s.t \quad \boldsymbol{G}'[k_s, i] = \boldsymbol{G}'[k_s, j] = 1, \quad 1 \leq s \leq n$$

$\boldsymbol{G}' =$



How to use this correlation ?

Public code $\mathscr{C}_{\text{pub}}$

The public code $\mathscr{C}_{\text{pub}}$ has :

- Parity-check matrix $\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}$
- Generator matrix $\boldsymbol{G}'$

Remark

We can recover rows of $\boldsymbol{G}'$.

Observation 1

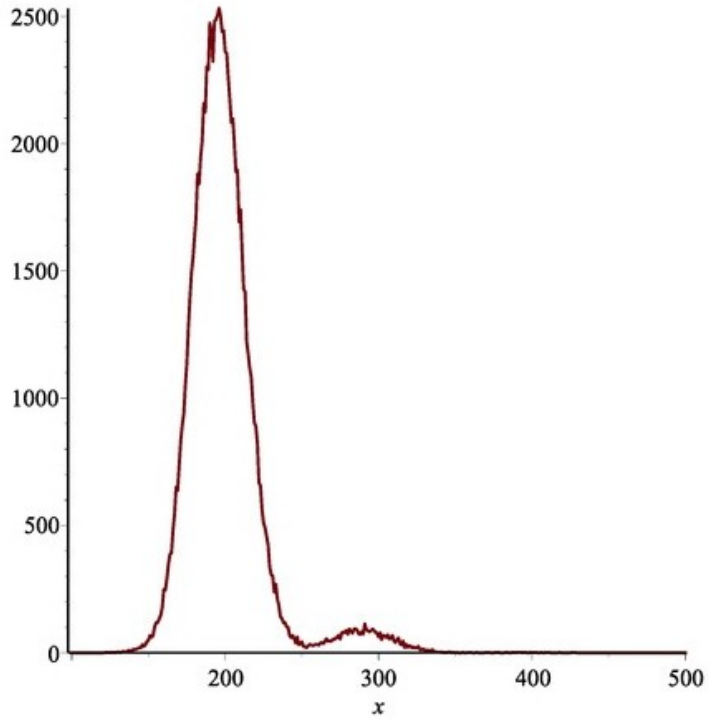


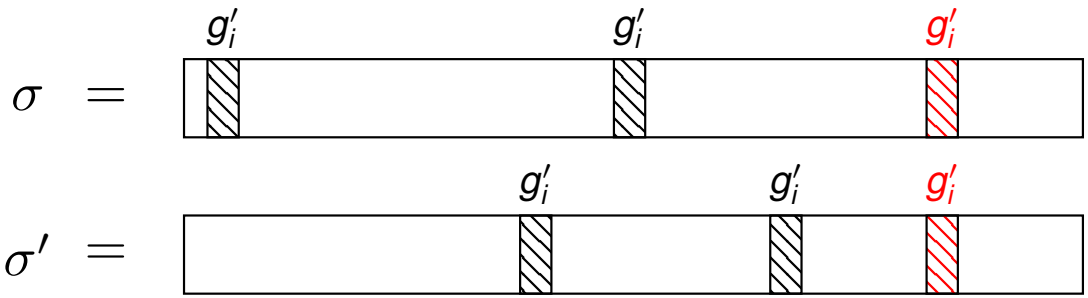
Figure : Distribution of the weight of the intersection of large number of σ

Observation 2

Observation :

$$\boldsymbol{\sigma} = \sum_{s=1}^{w_c/m_G} \boldsymbol{g}'_{i_s} + \boldsymbol{e}\boldsymbol{S}^T \tag{3}$$

This peak is explained by :



Recovering $\boldsymbol{Q}$ up to a column permutation

We have :

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1} \quad \text{and} \quad \boldsymbol{H} = \begin{pmatrix} \boldsymbol{P} & | & \boldsymbol{I} \end{pmatrix}$$

Recovering $\boldsymbol{Q}$ up to a column permutation

We have :

$$\boldsymbol{H}' = \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1} \quad \text{and} \quad \boldsymbol{H} = (\boldsymbol{P} \mid \boldsymbol{I})$$

Observe $\boldsymbol{Q}^{-1}$ has this form :

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Recovering $\boldsymbol{Q}$ up to a column permutation

We have found :

$$\boldsymbol{S}_p = \boldsymbol{S}\boldsymbol{\Pi}$$

By multiplying $\boldsymbol{H}'$ on the right by $\boldsymbol{S}_p$ we obtain :

$$\begin{aligned} \boldsymbol{H}'\boldsymbol{S}_p &= \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{S}^{-1}\boldsymbol{S}_p \\ &= \left(\boldsymbol{Q}^{-1}\boldsymbol{P} \mid \boldsymbol{Q}^{-1}\right)\boldsymbol{\Pi} \end{aligned}$$

To recover $\boldsymbol{Q}^{-1}$ permuted

The columns of $\boldsymbol{Q}^{-1}$ can be detected.

Resume

We have obtained :

$$\begin{aligned} \boldsymbol{S}_p &= \boldsymbol{S}\boldsymbol{\Pi} \\ \boldsymbol{Q}_p &= (\boldsymbol{Q}^{-1}\boldsymbol{\Pi}_r)^{-1} = \boldsymbol{\Pi}_r^{-1}\boldsymbol{Q} \end{aligned}$$

where

$$\boldsymbol{\Pi} = \begin{pmatrix} \boldsymbol{\Pi}_l & | & \boldsymbol{\Pi}_r \end{pmatrix}$$

Forging new signatures

Forgery is performed by using the pair of matrices $(\boldsymbol{Q}_p, \boldsymbol{S}_p)$ instead of the pair $(\boldsymbol{Q}, \boldsymbol{S})$.

1. Take counter i
2. Compute $\boldsymbol{s}_i = \mathcal{F}(Hash(\boldsymbol{m}||i))$
3. Compute private syndrome

$$\boldsymbol{s'}^T = \boldsymbol{Q}_p \boldsymbol{s}_i^T$$

4. Build sparse

$$\boldsymbol{e} = \underbrace{00\cdots0}_{k}||\boldsymbol{s'}$$

5. Compute

$$\sigma = \boldsymbol{e}\boldsymbol{S}_p{}^T + g'_k \quad \text{s.t} \quad \boldsymbol{H'} g'_k = 0$$

6. $\boldsymbol{H'}\sigma^T = \boldsymbol{s}^T \Leftrightarrow \boldsymbol{R}\boldsymbol{s}^T = 0$

Forging new signatures

The verifier computes

$$\boldsymbol{s}^* \stackrel{\text{def}}{=} \mathcal{F}(Hash(\boldsymbol{m}\|i))$$

and checks

$$\boldsymbol{H}'\boldsymbol{\sigma}^T = \boldsymbol{s}^{*T} \quad \text{and} \quad weight(\boldsymbol{\sigma}) \leq (m_T w + w_c)m_S$$

Experimental Results

Running the whole attack was performed on the parameters suggested for 80 bits of security

n	k	p	w	w_g	w_c	z	m_T	m_s
9800	4900	50	18	20	160	2	1	9

Table : Parameters for 80 bits of security.

We have used $100,000$ signatures to perform the attack which was implemented in `Sage` and took about one hour on a 6-core Intel® Xeon® running at 3.20 GHz.

Conclusion

- Gaborit Ruatta Schrek Zémor signature scheme : security proof that signatures do not leak information
- An attacker can forge new signatures using 100 000 signatures
- Maybe, actually this scheme can be used for one-time signature scheme