A QC-LDPC code-based public-key cryptosystem resistant to reaction attacks

Paolo Santini

Università Politecnica delle Marche
p.santini@pm.univpm.it

*Code-Based Cryptography Workshop 2018*

Fort Lauderdale, Florida, USA
April 5-6, 2018
The use of codes for cryptographic purposes was initiated by McEliece in 1978, proposing a cryptosystem based on Goppa codes.

The main drawback of code-based cryptosystems is represented by the dimension of the public key.

In the binary case, the smallest key sizes are reached when quasi-cyclic (QC) sparse codes are used.

Low-density parity-check (LDPC) codes and moderate-density parity-check (MDPC) codes use decoders that are usually characterized by a small (but non negligible) decoding failure rate (DFR).
Reaction attacks on sparse codes

- Low-density parity-check (LDPC) codes and moderate-density parity-check (MDPC) codes use decoders that are usually characterized by a small (but non negligible) decoding failure rate (DFR).

- The decryption failure probability depends on the structure of the secret key: an opponent can estimate such a probability by observing Bob’s reactions during decryption of known ciphertexts.


Avoiding reaction attacks

Different ideas have been proposed, in order to avoid reaction attacks:

- use of ephemeral keys;
- choice of the system parameters in order to achieve negligible DFR values;
- use of decoding strategies that do not leak information about the secret key;

---


J.-P-Tillich,” The decoding failure probability of MDPC codes,” 2018.

Avoiding reaction attacks

Different ideas have been proposed, in order to avoid reaction attacks:

- use of ephemeral keys;
- choice of the system parameters in order to achieve negligible DFR values;
- use of decoding strategies that do not leak information about the secret key;
- decoding with indistinguishable parity check matrices (with respect to reaction attacks).


A monomial code is a QC code whose parity check matrix is in the form

\[
H = \begin{bmatrix}
H_{0,0} & H_{0,1} & \cdots & H_{0,n_0} \\
H_{1,0} & H_{1,1} & \cdots & H_{1,n_0} \\
\vdots & \vdots & \ddots & \vdots \\
H_{r_0-1,0} & H_{r_0-1,1} & \cdots & H_{r_0-1,n_0-1}
\end{bmatrix}
\]

with each \( H_i \) being a circulant with size \( p \) and weight 1.

It can be easily shown that at least \( r_0 + 1 \) rows in \( H \) are linearly dependent on the other rows, hence the code has dimension \( k \geq (n_0 - r_0)p + r_0 + 1 = k_0 p + r_0 + 1 \).
Exponent matrix

- Considering the homomorphism between size-$p$ binary circulant matrices and polynomials in $\mathbb{F}_2[x]/(x^p - 1)$, each circulant block $H_{i,j}$ can be represented as $x^{w_{i,j}}$.
- The exponents of the monomial can be grouped in a matrix $W$, named exponent matrix, which is a compact representation of $H$

$$W = \begin{bmatrix}
w_{0,0} & w_{0,1} & \cdots & w_{0,n_0-1} \\
w_{1,0} & w_{1,1} & \cdots & w_{1,n_0-1} \\
\vdots & \vdots & \ddots & \vdots \\
w_{r_0-1,0} & w_{r_0-1,1} & \cdots & w_{r_0-1,n_0-1}
\end{bmatrix}$$
# Key generation

**Secret Key**
- $r \times n$ parity check matrix $H$;
- $k \times k$ scrambling matrix $S$.

**Public key**
- Let $G$ be a generator matrix for the secret code.
- The public key is $G' = S \cdot G$. 

---

[136x200] Paolo Santini

---

Key generation

**Secret Key**
- $r \times n$ parity check matrix $H$;
- $k \times k$ scrambling matrix $S$.

**Public key**
- Let $G$ be a generator matrix for the secret code.
- The public key is $G' = S \cdot G$.

**Reducing the public key size**
When a secure CCA2 conversion is used, $G'$ can be in systematic form.

---

Encryption

- Alice generates a length-\(n\) vector \(e\) with weight \(t\).
- She encrypts a \(k\)-bit message \(u\) as

\[
x = u \cdot G' + e
\]

Decryption

Bob computes 

\[
s = Hx^T = He^T
\]

He decodes \(s\), corrects \(e\) and recovers \(u\).
Encryption and decryption

**Encryption**

- Alice generates a length-\(n\) vector \(e\) with weight \(t\).
- She encrypts a \(k\)-bit message \(u\) as
  \[
  x = u \cdot G' + e
  \]

**Decryption**

- Bob computes \(s = H \cdot x^T = H \cdot e^T\).
- He decodes \(s\), corrects \(e\) and recovers \(u\).
The matrix $G$ can have the following structure

Each matrix $G_i$ has size $p \times (r_0 + 1)p$ and is in QC form.

$G_a$ contains additional $r' = n - k - r_0p$ rows, needed in order to compensate the rank deficiency. These rows do not depend on the parity check matrix entries.

The scrambling matrix $S$ has the following structure:

- Each matrix $S_i$ is a dense circulant of size $p$.
- The presence of the identity $I_{r'}$ is due to $G_a$.

With these choices $G'$ has the following structure

$$
\begin{align*}
G'_{0} & \\
G'_{1} & \\
\vdots & \\
G'_{K_{0-1}} & \\
\end{align*}
$$

with $G'_i = S_i \cdot G_i$.

The block $G_a$ is not part of the public key.
A particular choice - Public key

- With these choices $G'$ has the following structure

$$
\begin{array}{c}
G_0 \\
G_1' \\
\vdots \\
G_{k_0-1}' \\
G_a
\end{array}
$$

with $G_i' = S_i \cdot G_i$.

- The block $G_a$ is not part of the public key.

Public key size

With blocks $G_i'$ in systematic form, the public key size is

$$KS = k_0r_0p$$
Distance spectrum

- Given two ones at positions $v_1$ and $v_2$, the corresponding cyclic distance is

$$\delta(v_1, v_2) = \min \{ \pm(v_1 - v_2) \mod p \}$$

- The **distance spectrum** of a circulant matrix $A$ is the set of distances produced by couples of ones in a row of $A$.

- We say that a distance $d$ has multiplicity $\mu(d)$ if there are $\mu(d)$ distinct couples of ones at distance $d$.

Given two ones at positions $v_1$ and $v_2$, the corresponding cyclic distance is

$$\delta(v_1, v_2) = \min \{ \pm (v_1 - v_2) \mod p \}$$

The distance spectrum of a circulant matrix $A$ is the set of distances produced by couples of ones in a row of $A$.

We say that a distance $d$ has multiplicity $\mu(d)$ if there are $\mu(d)$ distinct couples of ones at distance $d$.

The decoding failure rate (DFR) depends on the number of common distances between $e$ and $H$.

Relation between distance spectrum and DFR

- We can write $\mathbf{e} = [\mathbf{e}_0, \cdots, \mathbf{e}_{n_0-1}]$ and $\mathbf{s} = [\mathbf{s}_0, \cdots, \mathbf{s}_{r_0-1}]$, with $s_j = \sum_{i=0}^{n_0-1} H_{j,i} \cdot e_i^T$.
- Common distances between $\mathbf{e}$ and $\mathbf{H}$ cause cancellations in the computation of the blocks $s_j$. 
We can write $e = [e_0, \cdots, e_{n_0-1}]$ and $s = [s_0, \cdots, s_{r_0-1}]$, with $s_j = \sum_{i=0}^{n_0-1} H_{j,i} \cdot e_i^T$.

Common distances between $e$ and $H$ cause cancellations in the computation of the blocks $s_j$.

**Observation #1**

The DFR depends on the syndrome weight.
Relation between distance spectrum and DFR

- We can write $e = [e_0, \cdots, e_{n_0-1}]$ and $s = [s_0, \cdots, s_{r_0-1}]$, with $s_j = \sum_{i=0}^{n_0-1} H_{j,i} \cdot e_i^T$.
- Common distances between $e$ and $H$ cause cancellations in the computation of the blocks $s_j$.

**Observation #1**
The DFR depends on the syndrome weight.

**Observation #2**
Since the whole error vector contributes to the computation of every syndrome block, an opponent cannot know the positions of blocks where cancellations occurred.
The distances in $H$ are uniquely defined by $W$.

Distances can only be defined when considering two different columns in $W$.

Let $\lambda_{i,j}(W)$ be the set of distances between the $i$-th and $j$-th columns of $W$: the distance spectrum $\Lambda(W)$ is defined as the array containing all the sets $\lambda_{i,j}(W)$. 

Example: for $p = 13$ and $W = \begin{bmatrix} 1 & 4 & 5 \\ 3 & 11 & 0 \end{bmatrix}$, 

$\Lambda(W) = \begin{bmatrix} 2 & 4 & 13 \\ 4 & 13 & 11 \\ 11 & 13 & 5 \\ 13 & 5 & 3 \\ 5 & 3 & 1 \\ 3 & 1 & 4 \\ 1 & 4 & 5 \\ 4 & 11 & 0 \\ 11 & 0 & 3 \\ 0 & 3 & 11 \\ 3 & 11 & 0 \\ 11 & 0 & 3 \\ 0 & 3 & 11 \\ 3 & 11 & 0 \end{bmatrix}$.
Distance spectrum for monomial codes

- The distances in $\mathbf{H}$ are uniquely defined by $\mathbf{W}$.
- Distances can only be defined when considering two different columns in $\mathbf{W}$.
- Let $\lambda_{i,j}(\mathbf{W})$ be the set of distances between the $i$-th and $j$-th columns of $\mathbf{W}$: the distance spectrum $\Lambda(\mathbf{W})$ is defined as the array containing all the sets $\lambda_{i,j}(\mathbf{W})$.
- Example: for $p = 13$ and $\mathbf{W} = \begin{bmatrix} 1 & 4 & 5 \\ 3 & 11 & 0 \end{bmatrix}$

$$
\Lambda(\mathbf{W}) = \begin{bmatrix}
- & \{3, 5\} & \{3, 4\} \\
\{3, 5\} & - & \{1, 2\} \\
\{3, 4\} & \{1, 2\} & -
\end{bmatrix}
$$
The knowledge of the spectrum $\Lambda(\mathbf{W})$ can be used to build a matrix $\hat{\mathbf{H}} = \mathbf{\Pi} \cdot \mathbf{H}$, with $\mathbf{\Pi}$ being a permutation matrix.

$\hat{\mathbf{H}}$ can be used to decode intercepted cyphertexts:

1. the opponent computes

$$\hat{s} = \hat{\mathbf{H}} \cdot \mathbf{x}^T =$$

$$= \mathbf{\Pi} \cdot \mathbf{H} (\mathbf{u} \cdot \mathbf{G}' + \mathbf{e})^T =$$

$$= \mathbf{\Pi} \cdot \mathbf{H} \cdot \mathbf{e}^T$$

2. decoding of $\hat{s}$ through $\hat{\mathbf{H}}$ returns $\mathbf{e}$. 
The knowledge of the spectrum $\Lambda(W)$ can be used to build a matrix $\hat{H} = \Pi \cdot H$, with $\Pi$ being a permutation matrix.

$\hat{H}$ can be used to decode intercepted ciphertexts:

1. the opponent computes

$$\hat{s} = \hat{H} \cdot x^T = \Pi \cdot H (u \cdot G' + e)^T = \Pi \cdot H \cdot e^T$$

2. decoding of $\hat{s}$ through $\hat{H}$ returns $e$.

Let $\hat{W}$ be the exponent matrix associated to $\hat{H}$: this matrix can be reconstructed from the distance spectrum $\Lambda(W)$. 
Reconstructing the exponent matrix

\[ G \leftarrow \text{graph with node 0} \]
\[
\text{for } j \in \{0, 1, \cdots, n_0 - 1\}, \ d \in \lambda_{0,j}(H) \ 	ext{do}
\]
\[
\text{for } b \in \{0, 2, \cdots, 2r_0 - 2\} \ 	ext{do}
\]
\[
z_j^{(b)} = (j - 1)p + [(p - d) \mod p]
\]
\[
z_j^{(b+1)} = (j - 1)p + d
\]

Augment \( G \) with nodes \( z_j^{(b)}, z_j^{(b+1)} \)

Augment \( G \) with edges \( \left(0, z_j^{(b)}\right), \left(0, z_j^{(b+1)}\right) \)

\[
\text{for } i \in \{1, \cdots, n_0 - 2\}, j \in \{i + 1, \cdots, n_0\} \ 	ext{do}
\]
\[
\text{for } b_i \in \{0, 1, \cdots, 2r_0 - 1\}, b_j \in \{0, 1, \cdots, 2r_0 - 1\} \ 	ext{do}
\]
\[
\text{if } \delta \left(z_i^{(b_i)}, z_j^{(b_j)}\right) \in \lambda_{i,j} \ 	ext{then}
\]

Augment \( G \) with edge \( \left(z_i^{(b_i)}, z_j^{(b_j)}\right) \)
Graph properties

- The algorithm builds the graph associated to the matrix $\hat{W}$ having all zeros in the first column; this matrix can be called the *standard form* of $W$ and denoted as $W^*$. 
Graph properties

- The algorithm builds the graph associated to the matrix $\hat{W}$ having all zeros in the first column; this matrix can be called the *standard form* of $W$ and denoted as $W^*$.

- Every row of $W^*$ is identified by a size-$n_0$ clique in $G$ which contains the node 0, and corresponds to at least two cliques $\zeta = \{z_0, z_1, \ldots, z_{n_0-1}\}$ and $\zeta^* = \{z_0^*, z_1^*, \ldots, z_{n_0-1}^*\}$, such that

$$z_i^* = p \left\lfloor \frac{z_i}{p} \right\rfloor + [(p - z_i) \mod p]$$
Graph properties

- The algorithm builds the graph associated to the matrix $\hat{W}$ having all zeros in the first column; this matrix can be called the *standard form* of $W$ and denoted as $W^*$.

- Every row of $W^*$ is identified by a size-$n_0$ clique in $G$ which contains the node 0, and corresponds to at least two cliques $\zeta = \{z_0, z_1, \ldots, z_{n_0-1}\}$ and $\zeta^* = \{z_0^*, z_1^*, \ldots, z_{n_0-1}^*\}$, such that

$$z_i^* = p \left\lfloor \frac{z_i}{p} \right\rfloor + [(p - z_i) \mod p]$$

- The graph $G$ cannot contain an edge between two nodes $z_i$ and $z_j$ such that $\left\lfloor \frac{z_i}{p} \right\rfloor = \left\lfloor \frac{z_j}{p} \right\rfloor$: the maximum number of size-$n_0$ cliques in the graph is equal to $p^{n_0-1}$. 
Graph properties

Example of the graph $G$ associated to a code with $n_0 = 5$, $r_0 = 2$, $p = 67$, with exponent matrix $W = \begin{bmatrix} 47 & 18 & 6 & 46 & 63 \\ 2 & 3 & 55 & 21 & 2 \end{bmatrix}$.
**A special class of monomial codes**

**Exponent matrix generation**

1. choose $p$ as a prime;
2. randomly pick $y = \left[ y_0, y_1, \cdots, y_{\lceil \frac{p}{2} \rceil - 1} \right]$, with $y_i \in \mathbb{N}, y_i \in [0; p - 1]$;
3. randomly pick a permutation $s = [s_0, s_1, \cdots, s_{p-1}]$ of the vector $[0, 1, \cdots, p - 1]$;
4. randomly pick a permutation $q = \left[ q_0, q_1, \cdots, q_{\lceil \frac{p}{2} \rceil - 1} \right]$ of the vector $[0, 1, \cdots, \lfloor \frac{p}{2} \rfloor]$;
5. for $i = 0, 1, \cdots, p$, compute the $i$-th column of $W$ as
   \[ y^T + s_i q^T \mod p \]
**Theorem:** all the exponent matrices constructed with the previous procedure satisfy the property

$$\lambda_{i,j}(W) = \left\{0, 1, 2, \cdots, \left\lfloor \frac{p}{2} \right\rfloor \right\}, \ \forall i, j$$

Any two nodes $z_i > 0$ and $z_j > 0$ such that $\lfloor z_i p \rfloor \neq \lfloor z_j p \rfloor$ are connected by an edge: the associated graph has $p$ cliques of size $n_0$. Indistinguishable secret keys

The distance spectrum is the same for all the secret exponent matrices.
**Theorem:** all the exponent matrices constructed with the previous procedure satisfy the property

\[ \lambda_{i,j}(W) = \left\{ 0, 1, 2, \cdots, \left\lfloor \frac{p}{2} \right\rfloor \right\}, \ \forall i, j \]

Any two nodes \( z_i \geq 0 \) and \( z_j > 0 \) such that \( \left\lfloor \frac{z_i}{p} \right\rfloor \neq \left\lfloor \frac{z_j}{p} \right\rfloor \) are connected by an edge: the associated graph has \( p^{n_0-1} \) cliques of size \( n_0 \).
**Theorem:** all the exponent matrices constructed with the previous procedure satisfy the property

\[ \lambda_{i,j}(W) = \left\{ 0, 1, 2, \ldots, \left\lfloor \frac{p}{2} \right\rfloor \right\}, \quad \forall i, j \]

Any two nodes \( z_i \geq 0 \) and \( z_j > 0 \) such that \( \left\lfloor \frac{z_i}{p} \right\rfloor \neq \left\lfloor \frac{z_j}{p} \right\rfloor \) are connected by an edge: the associated graph has \( p^{n_0-1} \) cliques of size \( n_0 \).

**Indistinguishable secret keys**

The distance spectrum is the same for all the secret exponent matrices.
The $i$-th column of $W^*$ can be expressed as

$$w_i^* = v_i^* q^T \mod p$$

where $v_0^* = 0$ and $[v_1^*, v_2^*, \cdots, v_{p-1}^*]$ corresponds to a permutation of the integers $\{1, 2, \cdots, p - 1\}$.

The vector $q^T$ is a permutation of the integers in $[0, 1, \cdots, \left\lfloor \frac{p}{2} \right\rfloor]$: different configurations of $q$ result in row permuted versions of $W^*$. 
The $i$-th column of $W^*$ can be expressed as

$$w_i^* = v_i^* q^T \mod p$$

where $v_0^* = 0$ and $[v_1^*, v_2^*, \cdots, v_{p-1}^*]$ corresponds to a permutation of the integers $\{1, 2, \cdots, p - 1\}$.

The vector $q^T$ is a permutation of the integers in $[0, 1, \cdots, \lfloor \frac{p}{2} \rfloor]$: different configurations of $q$ result in row permuted versions of $W^*$.

The number of standard exponent matrices is equal to $N_W = (p - 1)!$. 
Theorem: Let $W^{(0)}$ and $W^{(1)}$ be two exponent matrices generated according to the previous procedure, with $v^{(1)} \neq v^{(1)}$, and let $W^*(0)$ and $W^*(1)$ be their corresponding matrices in standard form. Then, $W^*(0)$ and $W^*(1)$ cannot be row permuted versions of the same matrix.
Secret key cardinality

- **Theorem:** Let $W^{(0)}$ and $W^{(1)}$ be two exponent matrices generated according to the previous procedure, with $v^{(1)} \neq v^{(1)}$, and let $W^*(0)$ and $W^*(1)$ be their corresponding matrices in standard form. Then, $W^*(0)$ and $W^*(1)$ cannot be row permuted versions of the same matrix.

- Only one matrix, among all the possible $N_W = (p - 1)!$ ones, is a parity check matrix of the public code.
Theorem: let \( W^{(0)} \) and \( W^{(1)} \) be two exponent matrices generated according to the previous procedure, with \( v^{(1)} \neq v^{(1)} \), and let \( W^{*,(0)} \) and \( W^{*,(1)} \) be their corresponding matrices in standard form. Then, \( W^{*,(0)} \) and \( W^{*,(1)} \) cannot be row permuted versions of the same matrix.

Only one matrix, among all the possible \( N_W = (p - 1)! \) ones, is a parity check matrix of the public code.

Brute-force equivalent security

The opponent cannot obtain information about the secret key: the only way of distinguishing the secret key is testing all possible candidates, whose number is equal to \( N_W \).
Proposed parameters require a number of operations $\geq 2^\lambda$, $\lambda \in \{80, 128, 256\}$, to run attacks on a classical computer.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>$p$</th>
<th>$n_0$</th>
<th>$r_0$</th>
<th>$t$</th>
<th>$N_W$</th>
<th>$K_s$ (kB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>103</td>
<td>103</td>
<td>52</td>
<td>84</td>
<td>$2^{538}$</td>
<td>34.14</td>
</tr>
<tr>
<td>128</td>
<td>137</td>
<td>137</td>
<td>69</td>
<td>132</td>
<td>$2^{773}$</td>
<td>80.36</td>
</tr>
<tr>
<td>256</td>
<td>257</td>
<td>257</td>
<td>129</td>
<td>261</td>
<td>$2^{1684}$</td>
<td>530.45</td>
</tr>
</tbody>
</table>
Conclusions and future works

- The proposed system achieves security against known reaction attacks, even with a non negligible DFR.
- The resulting key sizes are smaller than the ones of Goppa codes, but still too large with respect to other QC codes based systems.
Thanks for the attention