A New Code Based Public Key Encryption and Signature Scheme based on List Decoding

Presented by
Danilo Gligoroski

joint work with:
Simona Samardjiska and Håkon Jacobsen and Sergey Bezzateev

Department of Telematics,
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology - NTNU, NORWAY
Acknowledgements

- To Tanja Lange for inviting me to attend the workshop *Post-Quantum Cryptography and Quantum Algorithms*, Lorentz Center, 5-9 Nov 2012, where I got the initial idea
- To Christiane Peters for discussing with me the initial idea and for her encouragements
- To Ludovic Perret and Jean-Charles Faugère for inviting me to be a guest at their department at LIP6 for 3 months to analyze the scheme with Groebner bases
- To Nicolas Sendrier (discussing cheap distinguishers for the encryption variant of the scheme) and Jean-Pierre Tillich (discussing the scheme and the statistical properties of the signature part of the scheme) for spending two days at INRIA at Paris-Rocquencourt in discussions with us about different aspects of the security of the scheme
Introduction

• In 1978 Robert McEliece proposed a public key scheme based on Coding Theory
• The scheme could be used only for encryption
• Difficulty of decoding random linear codes
• Not so attractive as RSA (public key bigger than 32KB)
• Scheme was analyzed 35 years and seems solid
  – Some parameters adjusted in 2008
McEliece Public-Key Scheme

- **Key Generation**
  - *Alice* chooses $S$, $G$ and $P$
    - $S$ is a random $(k \times k)$ nonsingular binary matrix.
    - $G$ is a $(k \times n)$ generator matrix of a $t$-error-correcting binary linear code.
    - $P$ is a random $(n \times n)$ permutation matrix.
  - *Alice’s secret key*: $S$, $G$ and $P$
  - *Alice’s public key*: $G’ = SGP$
McEliece Public-Key Scheme (Cont.)

• Encryption
  - *Bob* sends a $k$-bit binary message $m$ to *Alice*, by computing
    $$c = mG' + e$$
  - $e$ is an $n$-bit random error vector of weight $t$. 

A new code based public key encryption and signature scheme based on list decoding
• **Decryption**
  - When Alice receives $c$, she
    1. Calculates $c' = cP^{-1} = mSG + eP^{-1}$
    2. Uses the decoding algorithm of the original code $G$ to obtain $m' = mS$ form $c'$
    3. Recovers message $m$ by computing $m = m'S^{-1}$
McEliece Public-Key Scheme (Cont.)

Parameters

• For 80 bits of security McEliece originally suggested to use Goppa codes and the security parameter sizes of $n=1024, k=524, t=50$

• Recent analysis (Bernstein, Lange and Peters, 2008) suggests parameter sizes of $n=2048, k=1751, t=27$ for 80 bits of security (standard algebraic decoding for the Goppa codes)

• Or $n=1632, k=1269, t=34$ when using list decoding for the Goppa code
McEliece Public-Key Scheme (Cont.)

Parameters

- For 80 bits of security McEliece uses Goppa codes and the parameters are: $n=1024$, $k=524$, $t=50$

- Recent analysis (Bernstein, Lange and Peters, 2008) suggests parameter sizes of $n=2048$, $k=1751$, $t=27$ for 80 bits of security (standard algebraic decoding for the Goppa codes)

- Or $n=1632$, $k=1269$, $t=34$ when using list decoding for the Goppa code

Q: Why Bob must produce just so little errors when he encrypts the messages?
How the codes are decoded in Coding Theory?

The set of all codewords composes the code $C$
How the codes are decoded in Coding Theory?

\[ d \] – Minimum distance of the code \( C \)
How the codes are decoded in Coding Theory?

\[ t = d/2 \]

radius for unique decoding of the code \( C \)
How the codes are decoded in Coding Theory?

\[ t = \frac{d}{2} \]
How the codes are decoded in Coding Theory?

When the dimension of the codewords is $n$, and their number is $2^k$, and the minimum distance is $d$, we talk about $(n, k, d)$ codes.

$t = d/2$
How the codes are decoded in Coding Theory?

\[ t = \frac{d}{2} \]

\[ k \leq n - \log_2 \left( \sum_{i=0}^{t} \binom{n}{i} \right) \]

\[ t = \frac{d}{2} \]
How the codes are decoded in Coding Theory?

A new code based public key encryption and signature scheme based on list decoding
How the codes are decoded in Coding Theory?

\[ c' = c + e \]

\[ c' \text{ can not be decoded if } ||e|| > t \]
Can we do better with some other decoding approach?
Can we do better with some other decoding approach?

- Yes we can :-)  
  - By using **List decoding**  
  - Proposed in 50’ by Elias and Wozencraft (but no efficient algorithm)  
  - Sudan in 1997 proposed an efficient list decoding algorithm with polynomial run-time  
  - Guruswami and Sudan in 1998 significantly improved Sudan’s algorithm
Can we do better with some other decoding approach?

- Yes we can :-)  
  - By using List decoding
  - Proposed in 50’ by Elias
  - Sudan in 1997 proposed polynomial run-time
  - Guruswami and Sudan algorithm

Can we do EVEN BETTER than the classical list decoding, if our coding/decoding is for cryptographical purposes?
Can we do better with some other decoding approach?

- Yes we can :-)  
  - By using List decoding  
  - Proposed in 50’ by Elias

Can the errors introduced in encryption phase be chosen from some arbitrary set and still be decodable?

Can we do EVEN BETTER than the classical list decoding, if our decoding is for graphical purposes?
Can we do better with some other decoding approach?

- Yes we can :-)  
  - By using List decoding
  - Proposed in 1998

Can we do EVEN BETTER than the classical list decoding, if our encryption is for graphical purposes?

YES!

Can the in encryption phase be chosen from some arbitrary set and still be decodable?

A new code based public key encryption and signature scheme based on list decoding
Our approach

• Instead of the classical approach in coding theory where errors are forming a Hamming sphere with radius $t$ around the codewords
Our approach

• We want to form an arbitrary set of errors around the codewords
• We want also to increase the set of possible errors that the encryptor can introduce
• Still to be capable to decode efficiently and reliably

A new code based public key encryption and signature scheme based on list decoding
Our approach

A classical modeling of an error set around a code word with the Hamming sphere.

An artistic visualization of our idea with an arbitrary error set around the codewords (Escher’s tessellations).
Our approach

- McEliece in the world of Escher
- Cryptology ePrint Archive: Report 2014/360
  http://eprint.iacr.org/2014/360
Some definitions and properties

Definition 1. Let $\ell$ be a positive integer, and let $E_\ell \subseteq \mathbb{F}_2^\ell$ such that $|E_\ell| > 2^{\ell-1}$. We define the density of the set $E_\ell$ as:

$$D(E_\ell) = |E_\ell|^{1/\ell}.$$ 

We will refer to the integer $\ell > 0$ as granulation (when clear from context we will use just $E$).

Proposition 1. 1. Let $E_{\ell_1} \subseteq \mathbb{F}_2^{\ell_1}$, $E_{\ell_2} \subseteq \mathbb{F}_2^{\ell_2}$, for some integers $\ell_1, \ell_2 > 0$. Let $D(E_{\ell_1}) = D(E_{\ell_2}) = \rho$. Then $D(E_{\ell_1} \times E_{\ell_2}) = \rho$.

2. Let $E_{\ell,1}, E_{\ell,2}, \ldots, E_{\ell,m} \subseteq \mathbb{F}_2^\ell$, $\ell > 0$, and $D(E_{\ell,1}) = D(E_{\ell,2}) = \cdots = D(E_{\ell,m}) = \rho$. Then $D(E_{\ell,1} \times E_{\ell,2} \times \cdots \times E_{\ell,m}) = \rho$. 

A new code based public key encryption and signature scheme based on list decoding
Some definitions and properties

• Example

1. Let \( E_2 = \{ x \in \mathbb{F}_2^2 | wt(x) < 2 \} = \{ (0, 0), (0, 1), (1, 0) \} \). Then \( D(E_2) = |E_2|^{1/2} = 3^{1/2} \), and also \( D(E_2^m) = |E_2^m|^{1/4} = 9^{1/4} = 3^{1/2} \) as well as \( D(E_2^m) = 3^{1/2} \) for any positive integer \( m \).

2. Let \( E_{4,1} = \{ x \in \mathbb{F}_2^4 | 2 \leq wt(x) \leq 3 \} \). Then \( D(E_{4,1}) = \left( \sum_{i=2}^{3} \binom{4}{i} \right)^{1/4} = 10^{1/4} \), and also \( D(E_{4,1}^m) = 10^{1/4} \) for any positive integer \( m \). Note that the set \( E_{4,2} = \{ x \in \mathbb{F}_2^4 | wt(x) \leq 2 \} \setminus \{ (0, 0, 0, 0) \} \) has also density \( D(E_{4,2}) = 10^{1/4} \).

3. Let \( E_4 = \{ (0, 1, 0, 0), (0, 0, 0, 1), (0, 1, 0, 1), (1, 0, 0, 1), (0, 0, 1, 0), (0, 1, 1, 0), (1, 0, 1, 0), (1, 1, 1, 0), (0, 1, 1, 1), (1, 1, 1, 1) \} \). The values of \( E_4 \) are chosen without any particular rule in mind. Then \( D(E_4) = |E_4|^{1/4} = 10^{1/4} \) as well as \( D(E_4^m) = 10^{1/4} \) for any positive integer \( m \).
Some definitions and properties

Proposition 3. Let $C$ be any binary $(n, k)$ code and $E \subseteq \mathbb{F}_2^n$ be an error set of density $\rho$. Let $w$ be any word of length $n$, $W_E = \{w + e | e \in E\}$ and $C_{W_E}$ denote the set of codewords in $W_E$. Then:

1. The expected number of codewords in $W_E$ is $\rho^n2^{k-n}$. The probability that $C_{W_E}$ is an empty set is given by $\Pr[C_{W_E} = \emptyset] \leq e^{-(\rho^n2^{k-n+1}-1)^2/(\rho^n2^{k-n+3})}.

2. Suppose there exists a codeword $c \in W_E$. Then the expected number of codewords in $W_E \setminus \{c\}$ is approximately $\rho^n2^{k-n}$ for large enough $n$ and $k$. The probability that $C_{W_E}\setminus\{c\}$ has another element except $c$ is estimated by $\Pr[|C_{W_E}\setminus\{c\}| \geq 1] \leq e^{-(1-\rho^n2^{k-n})^2/(1+\rho^n2^{k-n})}$. 


Some definitions and properties

Proposition 3. Let $C$ be any binary $(n, k)$ code and $E \subseteq \mathbb{F}_2^n$ be an error set of density $\rho$. Let $w$ be any word of length $n$, $W_E = \{w + e | e \in E\}$ and $C_{W_E}$ denote the set of codewords in $W_E$. Then:

1. The expected number of codewords in $W_E$ is $\rho^n 2^{k-n}$. The probability that $C_{W_E}$ is an empty set is given by $\Pr[C_{W_E} = \emptyset] \leq e^{-\left(\rho^n 2^{k-n} + 1\right)^2 / \left(\rho^n 2^{k-n} + 3\right)}$.

2. Suppose there exists a codeword $c \in W_E$. Then the expected number of codewords in $W_E \setminus \{c\}$ is approximately $\rho^n 2^{k-n}$ for large enough $n$ and $k$. The probability that $C_{W_E \setminus \{c\}}$ has another element except $c$ is estimated by $\Pr[|C_{W_E \setminus \{c\}}| \geq 1] \leq e^{-\left(1 - \rho^n 2^{k-n}\right)^2 / \left(1 + \rho^n 2^{k-n}\right)}$.

1. Let $C$ be a $(1280, 256)$ binary code. The code rate is 0.2. We consider an error set $E$ of density $\rho = 3^{1/2}$. Let $c$ be a codeword and $w = c + e$ for some $e \in E$. Then, from Proposition 3 the decoding list of the word $w$ is of average length $1 + \rho^n 2^{k-n} = 1.00127$. The probability that there is another element in the list except $c$ is 0.37. Note that these parameters may be suitable for building an encryption scheme since we can expect that the list has only one element.

2. Let $C$ be a $(1208, 256)$ binary code. The code rate is 0.211921. We consider an error set $E$ of density $\rho = 3^{1/2}$. Let $w$ be a word of length $n$. Then, the decoding list of the word $w$ is of average length $39.8733$, and the probability that the list is empty is $2^{-28}$. Such parameters are suitable for building a signature scheme since with great confidence we can always expect to have a valid signature. Moreover, the number of valid signatures is relatively small.
Some definitions and properties

- Example of Code where we can decode up to \( n/2 \) errors

Let \( E_2 = \{ x \in \mathbb{F}_2^n | \text{wt}(x) < 2 \} = \{(0, 0), (0, 1), (1, 0)\} \). Then \( D(E_2) = |E_2|^{1/2} = 3^{1/2} \), and also \( D(E_3^2) = |E_3^2|^{1/4} = 9^{1/4} = 3^{1/2} \) as well as \( D(E_3^m) = 3^{1/2} \) for any positive integer \( m \).
Some definitions and properties

• Example of Code where we can decode up to $n/2$ errors

Let $E_2 = \{x \in \mathbb{F}_2^2 \mid \text{wt}(x) < 2\} = \{(0, 0), (0, 1), (1, 0)\}$. Then $D(E_2) = |E_2|^{1/2} = 3^{1/2}$, and also $D(E_2^m) = |E_2^m|^{1/4} = 9^{1/4} = 3^{1/2}$ as well as $D(E_3^m) = 3^{1/2}$ for any positive integer $m$.

Up to 1 error in every two bits AND up to $n/2$ errors
Encryption

\[ m \in \mathbb{F}_2^k, \]

\[ c = mG_{\text{pub}} + e \in \mathbb{F}_2^n \]

\[ e \text{ (drawn from a specific error set } E^m) \]
Encryption

\[ m \in \mathbb{F}_2^k, \]

\[ c = mG_{\text{pub}} + e \in \mathbb{F}_2^n \]

\( e \) (drawn from a specific error set \( E^m \))

The scheme looks similar as LWE schemes, but with different error set and powerful list decoding algorithm.
Decoding

\[ G = I_k \]

\[ B_1 \]

\[ B_2 \]

\[ 0 \]

\[ B_w \]
Decoding

\[ G = I_k \]

\[ x \]

[Diagram showing a matrix structure with submatrices labeled \( B_1 \), \( B_2 \), and \( B_w \).]
A new code based public key encryption and signature scheme based on list decoding

Decoding

\[ G = I_k \]

\[ x \]

\[ e \]
Decoding

\[ G = I_k \]

\[ x \]

\[ e \]

\[ y \]

A new code based public key encryption and signature scheme based on list decoding
Decoding

$$G = I_k$$

$$x$$

$$e$$

$$y$$

A new code based public key encryption and signature scheme based on list decoding
A new code based public key encryption and signature scheme based on list decoding.
Decoding

\[ G = I_k \]

\[ x \]

\[ e \]

\[ y \]

A new code based public key encryption and signature scheme based on list decoding
Decoding
A new code based public key encryption and signature scheme based on list decoding
Decoding

\[
x \xrightarrow{G = I_k} B_1 \cup B_2 \cup \ldots \cup B_w
\]

\[
y \xrightarrow{e}
\]
A new code based public key encryption and signature scheme based on list decoding
Decoding

\[ G = \begin{bmatrix} I_k \end{bmatrix} \]

\[ x \]

\[ e \]

\[ y \]
A nice property of the scheme

• The scheme is both encryption and signature scheme
A nice property of the scheme

• The scheme is both encryption and signature scheme
• Why the ordinary McEliece scheme is so hard to use it for digital signatures?
Recall …

\[ t = d/2 \]
Recall …

The probability that a random syndrome lies in the “decodable areas” is exponentially small.

$t = d/2$
Recall …
Recall …

The probability that a random syndrome lies in a “decodable area” for our codes is almost 1.
A new code based public key encryption and signature scheme based on list decoding

Algorithm 3 Signing

Input: A value $h \in \mathbb{F}_2^n$ to be signed. The private key $S$, $G$ and $P$.
Output: A valid signature $\sigma \in \mathbb{F}_2^k$, so that $\sigma G_{\text{pub}} + h \in E^m \subset \mathbb{F}_2^n$.

Procedure:

1. Compute $y = hP^{-1}$.
2. Decode $y$ using Algorithm 1, to obtain a list $L_w$ of (possibly several) valid decodings.
3. Select any element $x \in L_w$ and compute $\sigma = xS^{-1}$.

Algorithm 4 Verification

Input: A pair $(h, \sigma) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$, and the public key $G_{\text{pub}}$.

Output:

$$\text{Ver}(h, \sigma) = \begin{cases} 
\text{Accept,} & \text{if } \sigma G_{\text{pub}} + h \in E^m \subset \mathbb{F}_2^n. \\
\text{Reject,} & \text{otherwise.}
\end{cases}$$
A nice property of our approach

Recall (we can control the number of elements in the decoding list):

Let $C$ be a $(1208, 256)$ binary code. The code rate is $0.211921$. We consider an error set $E$ of density $\rho = 3^{1/2}$. Let $w$ be a word of length $n$. Then, the decoding list of the word $w$ is of average length $39.8733$, and the probability that the list is empty is $2^{-28}$. Such parameters are suitable for building a signature scheme, since with great confidence we can always expect to have a valid signature. Moreover, the number of valid signatures is relatively small.
A nice property of our approach

**Algorithm 3 Signing**

**Input:** A value $h \in \mathbb{F}_2^n$ to be signed. The private key $S, G$ and $P$.

**Output:** A valid signature $\sigma \in \mathbb{F}_2^k$, so that $\sigma G_{\text{pub}} + h \in E^m \subset \mathbb{F}_2^n$.

**Procedure:**

1. Compute $y = hP^{-1}$.
2. Decode $y$ using Algorithm 1, to obtain a list $L_w$ of (possibly several) valid decodings.
3. Select any element $x \leftarrow L_w$ and compute $\sigma = xS^{-1}$.

**Algorithm 4 Verification**

**Input:** A pair $(h, \sigma) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$, and the public key $G_{\text{pub}}$.

**Output:**

$$\text{Ver}(h, \sigma) = \begin{cases} 
\text{Accept,} & \text{if } \sigma G_{\text{pub}} + h \in E^m \subset \mathbb{F}_2^n, \\
\text{Reject,} & \text{otherwise}.
\end{cases}$$

Recall (we can control the number of elements in the decoding list):

Let $C$ be a $(1208, 256)$ binary code. The code rate is $0.211921$. We consider an error set $E$ of density $\rho = 3^{1/2}$. Let $w$ be a word of length $n$. Then, the decoding list of the word $w$ is of average length $39.8733$, and the probability that the list is empty is $2^{-28}$. Such parameters are suitable for building a signature scheme, since with great confidence we can always expect to have a valid signature. Moreover, the number of valid signatures is relatively small.
Security analysis

- Four types of attacks analyzed
  1. Information Set Decoding for Error Sets of a Given Density
  2. Modeling ISD using Polynomial System Solving
     Groebner bases approach
  3. Rank Attacks
  4. Cheap distinguishers (equivalent keys finders)
1. Information Set Decoding for Error Sets of a Given Density

Theorem 1. The probability of success of one iteration and the cost of one iteration of the Lee-Brickell variant, Stern variant, Finiasz-Sendrier variant, Bernstein-Lange-Peters variant, May-Meurer-Thomae variant and Becker-Joux-May-Meurer variant adapted to error sets of density $\rho$ are given in Table 1.
<table>
<thead>
<tr>
<th>Variant</th>
<th>$\rho Pr_{V A R}$</th>
<th>$\rho Cost_{V A R} - Cost_{Gauss}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB</td>
<td>$\left( \frac{k}{2^{\ell}} \right) \frac{(\rho^\ell - 1)^p}{\rho^k}$</td>
<td>$\left( \frac{k}{p} \right) (\rho^\ell - 1)^p p n$</td>
</tr>
<tr>
<td>ST</td>
<td>$\left( \frac{k}{2^{\ell}} \right)^2 \frac{(\rho^\ell - 1)^{2p}}{\rho^{k+\lambda}}$</td>
<td>$2\lambda p L + 2pn \frac{L^2}{2^\lambda}$, $L = \left( \frac{k}{2^{\ell}} \right) (\rho^\ell - 1)^p$</td>
</tr>
<tr>
<td>FS</td>
<td>$\left( \frac{k+\lambda}{2^{\ell}} \right)^2 \frac{(\rho^\ell - 1)^{2p}}{\rho^{k+\lambda}}$</td>
<td>$2\lambda p L + 2pn \frac{L^2}{2^\lambda}$, $L = \left( \frac{k+\lambda}{2^{\ell}} \right) (\rho^\ell - 1)^p$</td>
</tr>
<tr>
<td>BLP</td>
<td>$\left( \frac{k}{2^{\ell}} \right)^2 \left( \frac{\lambda_1}{q} \right) \left( \frac{\lambda_2}{q} \right)$</td>
<td>$\left( \frac{k}{p} \right) (\rho^\ell - 1)^p 2(\lambda_1 + \lambda_2)p + \left( \frac{k}{2^{\ell}} \right) \left( \left( \frac{\lambda_1}{q} \right) + \left( \frac{\lambda_2}{q} \right) \right) (\rho^\ell - 1)^p + q(\lambda_1 + \lambda_2)q$</td>
</tr>
<tr>
<td></td>
<td>$\cdot \frac{(\rho^\ell - 1)^{2p+2q}}{\rho^{k+\lambda_1+\lambda_2}}$</td>
<td>$+ \frac{\left( \frac{k}{2^{\ell}} \right)^2 \left( \frac{\lambda_1}{q} \right) \left( \frac{\lambda_2}{q} \right)}{2^{\lambda_1+\lambda_2}} (\rho^\ell - 1)^{2p+2q} 2(p+q)n$</td>
</tr>
<tr>
<td>MMT</td>
<td>$\left( \frac{k+\lambda}{2^{\ell}} \right)^2 \frac{(\rho^\ell - 1)^{2p}}{\rho^{k+\lambda}}$</td>
<td>$2\lambda_2 p L + (2n + \lambda - \lambda_2)p \frac{L^2}{2^{\lambda_2}} + pn \frac{L^4}{2^{\lambda+\lambda_2}}$, $L = \left( \frac{k+\lambda}{p/2} \right) (\rho^\ell - 1)^{p/2}$</td>
</tr>
<tr>
<td>BJMM</td>
<td>$\left( \frac{k+\lambda}{p} \right) \frac{(\rho^\ell - 1)^p}{\rho^{k+\lambda}}$</td>
<td>$4Pr_{coll}^{-4} p_2 \left( L_3 \log_2 R_2 + n \frac{L_2^2}{R_2^2} \right) + 2n \left( p_1 \frac{L_2^2 R_2}{R_1} + p \frac{L_2^2 R_1}{2^\lambda} \right)$, $Pr_{coll} = \left( \frac{k+\lambda}{p_2/2} \right)^2 \left( \frac{k+\lambda}{p_2} \right)^{-1}$, $p_i = \frac{p_{i-1}}{2} + \epsilon_i$, $i = 1, 2, p_0 = p$,</td>
</tr>
<tr>
<td></td>
<td>$L_i = \left( \frac{k+\lambda}{p_i} \right) (\rho^\ell - 1)^{p_i}$, $i = 1, 2$, $L_3 = \left( \frac{k+\lambda}{p_2/2} \right) (\rho^\ell - 1)^{p_2/2}$, $R_i = \left( \frac{p_{i-1}}{p_{i-1/2}} \right) \left( \frac{(k+\lambda)^{\ell}}{\epsilon_i} \right) (\rho^\ell - 1)^{\epsilon_i}$, $i = 1, 2, p_0 = p$</td>
<td></td>
</tr>
</tbody>
</table>
A new code based public key encryption and signature scheme based on list decoding

<table>
<thead>
<tr>
<th>Variant</th>
<th>$\rho P r_{VAR}$</th>
<th>$\rho C o s t_{VAR} - C o s t_{Gauss}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LB$</td>
<td>$(k/\ell)(\rho^\ell - 1)^p$  (\rho^k)</td>
<td>(2\lambda pL + 2pn\frac{L^2}{2\lambda}), (L = \left(\frac{k/2\ell}{\rho}\right)(\rho^\ell - 1)^p)</td>
</tr>
<tr>
<td>$ST$</td>
<td>$(k/2\ell)^2(\rho^\ell - 1)^{2p}$  (\rho^{k+\lambda})</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variant</th>
<th>$LB$</th>
<th>$ST$</th>
<th>$FS$</th>
<th>$BLP$</th>
<th>$MMT$</th>
<th>$BJMM$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 256$</td>
<td>$2^{212}$</td>
<td>$2^{197}$</td>
<td>$2^{183}$</td>
<td>$2^{186}$</td>
<td>$2^{146}$</td>
<td>$2^{123}$</td>
</tr>
<tr>
<td>$k = 512$</td>
<td>$2^{416}$</td>
<td>$2^{381}$</td>
<td>$2^{356}$</td>
<td>$2^{356}$</td>
<td>$2^{279}$</td>
<td>$2^{226}$</td>
</tr>
</tbody>
</table>

| MMT | $(k+\lambda)/2\ell)^2(\rho^\ell - 1)^{2p}$  \(\rho^{k+\lambda}\) | $2\lambda_2 pL + (2n + \lambda - \lambda_2)p\frac{L^2}{2\lambda_2} + pn\frac{L^4}{2\lambda_2}$, \(L = \left(\frac{(k+\lambda)/\ell}{p/2}\right)(\rho^\ell - 1)^{p/2}\) |
| BJMM | $(k+\lambda)/\ell)(\rho^\ell - 1)^p$  \(\rho^{k+\lambda}\) | $4Pr_{coll}^{-4}p_2 \left(L_3 \log_2 R_2 + n\frac{L^2}{R_2}\right) + 2n \left(p_1\frac{L^2}{R_1} + p\frac{L^2}{2\lambda}\right)$, \(L = \left(\frac{(k+\lambda)/\ell}{p_2/2}\right)(\rho^\ell - 1)^{p_2/2}\) |

$$Pr_{coll} = \left(\frac{(k+\lambda)/\ell}{p_2/2}\right)^2\left(\frac{(k+\lambda)/\ell}{p_2}\right)^{-1}, p_i = \frac{p_i-1}{2} + \epsilon_i, i = 1, 2, p_0 = p,$$

$$L_i = \left(\frac{(k+\lambda)/\ell}{p_i}\right)(\rho^\ell - 1)^{p_i}, i = 1, 2, L_3 = \left(\frac{(k+\lambda)/\ell}{p_2/2}\right)(\rho^\ell - 1)^{p_2/2},$$

$$R_i = \left(\frac{p_i-1}{p_i-1/2}\right)^{\frac{(k+\lambda)/\ell-p_i-1}{\epsilon_i}}(\rho^\ell - 1)^{\epsilon_i}, i = 1, 2, p_0 = p$$
2. Modeling ISD using Polynomial System Solving
   Groebner bases approach

Example 1.
Let the error set be $E_2 = \{(0,0), (0,1), (1,0)\}$. This set is completely described by the following quadratic equation:

$$e_1 e_2 = 0,$$

i.e. the solutions of that equation coincides with the error set.

Example 2.
Let the error set be $E_4 = \{(0,1,0,0), (0,0,0,1), (0,1,0,1), (1,0,0,1), (0,0,1,0), (0,1,1,0), (1,0,1,0), (1,1,1,0), (0,1,1,1), (1,1,1,1)\}$. This set is completely described by the following cubic equation:

$$e_2 + e_3 + e_4 + e_1 e_2 + e_2 e_3 + e_2 e_4 + e_1 e_2 e_3 = 1.$$
Security analysis

2. Modeling ISD using Polynomial System Solving

Groebner bases approach

Given a public generator matrix $G_{pub}$ and a ciphertext $c$, we can form $n$ linear equations

$$xG_{pub} + y = c,$$

where $x$ denotes the $k$ unknown bits of the message, and $y$ is the $n$-bit unknown error. Clearly, we don’t have enough equations to find the correct solution efficiently. However, from the known structure of the error vector we can derive additional equations of higher degree that describe exactly the error set. If we denote these equations as $P(y) = 0$, then a solution of the system

$$xG_{pub} + y = c$$

$$P(y) = 0$$

will give the same solution for the message and the error vector as the decoding algorithm with the knowledge of the private key.

We emphasize that any error set can be described by a system of equations, including the set of errors of a bounded weight used in the McEliece system. The efficiency of this approach strongly depends on the error structure.
Security analysis

2. Modeling ISD using Polynomial System Solving
   Groebner bases approach

Furthermore, it is possible to introduce an optimization parameter in the form of a guess of some of the errors, or a guess of linear equations for the errors. In what follows we present the modeling of an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$.

Let $E_\ell$ be an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$. Without loss of generality, we can assume that $E_\ell = \{(00), (01), (10)\}$. Let $(e_1, e_2) \in E_\ell$. Then, the equation $e_1 e_2 = 0$ describes completely the error set $E_\ell$. Hence, the system turns into:

$$(x_1, \ldots, x_k)G_{\text{pub}} + (y_1, \ldots, y_n) = c$$

$$y_1 y_2 = 0$$

$$\ldots$$

$$y_{n-1} y_n = 0$$

The system can be easily transformed to the following form:

$$A_1(x_1, \ldots, x_k)A_2(x_1, \ldots, x_k) = 0$$

$$\ldots$$

$$A_{n-1}(x_1, \ldots, x_k)A_n(x_1, \ldots, x_k) = 0$$

(14)

where $A_i$ are some affine expressions in the variables $x_1, \ldots, x_k$. 
2. Modeling ISD using Groebner bases approach

Furthermore, it is possible to introduce the ISD with its errors, or a guess of linear equations for the errors. In what follows we present the modeling of an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$.

Let $E_\ell$ be an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$. Without loss of generality, we can assume that $E_\ell = \{(00), (01), (10)\}$. Let $(e_1, e_2) \in E_\ell$. Then, the equation $e_1 e_2 = 0$ describes completely the error set $E_\ell$. Hence, the system (13) turns into:

$$(x_1, \ldots, x_k)G_{\text{pub}} + (y_1, \ldots, y_n) = c$$

$$y_1 y_2 = 0$$

$$\ldots$$

$$y_{n-1} y_n = 0$$

The system can be easily transformed to the following form:

$$A_1(x_1, \ldots, x_k)A_2(x_1, \ldots, x_k) = 0$$

$$\ldots$$

$$A_{n-1}(x_1, \ldots, x_k)A_n(x_1, \ldots, x_k) = 0$$

(14)

where $A_i$ are some affine expressions in the variables $x_1, \ldots, x_k$. 

Given a concrete encryption $c$, we model the algebraic system to solve it by Groebner bases as a MQ system where the unknown variables are from the message and the error.
Security analysis

2. Modeling ISD using Groebner bases approach

Furthermore, it is possible to introduce the algebraic system to solve it by Groebner bases as a MQ system where the unknown variables are from the message and the error.

We can look at this scheme as the first public key encryption and signature multivariate (quadratic, cubic, ...) scheme that is not given in a form

$$S \cdot P \cdot T$$

Given a concrete encryption $c$, we model the algebraic system to solve it by Groebner bases as a MQ system where the unknown variables are from the message and the error.

The system can be described as:

$$A_1(x_1, \ldots, x_k)A_2(x_1, \ldots, x_k) = 0$$

$$\cdots$$

$$A_{n-1}(x_1, \ldots, x_k)A_n(x_1, \ldots, x_k) = 0$$

where $A_i$ are some affine expressions in the variables $x_1, \ldots, x_k$. 

A new code based public key encryption and signature scheme based on list decoding
2. Modeling ISD using Polynomial System Solving

Groebner bases approach

We can introduce an optimization parameter $p$ as follows. Suppose we have made a correct guess that the equation $y_{2t-1} + y_{2t} = b_t, b_t \in \{0, 1\}$ holds for $p$ pairs $(y_{2t-1}, y_{2t})$ of coordinates of the error vector. Adding these $p$ new equations to the system reduces the complexity of solving it. Note that it is enough to correctly guess $k$ equation to obtain a full system of $k$ unknowns. The probability of making the correct guess is $Pr = (2/3)^p$. Under the natural constrain $0 \leq p \leq k$, we can roughly estimate the complexity to

$$Comp = (3/2)^p \cdot \left( \left( \frac{k-p}{Dreg_{k-p}} \right) + p \right)^\omega$$
Security analysis

2. Modeling ISD using Polynomial System Solving

Groebner bases approach

We can introduce an optimization parameter $p$ as follows. Suppose we have made a correct guess that the equation $y_{2t-1} + y_{2t} = b_t$, $b_t \in \{0, 1\}$ holds for $p$ pairs $(y_{2t-1}, y_{2t})$ of coordinates of the error vector. Adding these $p$ new equations to the system reduces the complexity of solving it. Note that it is enough to correctly guess $k$ equation to obtain a full system of $k$ unknowns. The probability of making the correct guess is $Pr = \left(\frac{2}{3}\right)^p$. Under the natural constrain $0 \leq p \leq k$, we can roughly estimate the complexity to

$$\text{Comp} = \left(\frac{3}{2}\right)^p \cdot \left(\frac{k-p}{D_{reg_{k-p}}} + p\right)^\omega$$

where $D_{reg_{k-p}}$ denotes the degree of regularity of a system of $k-p$ variables of the form (14).

We performed some experiments using the $F_4$ algorithm implemented in MAGMA, and based on rather conservative projections of the degree of regularity, we give the following table with a rough estimate of the lower bound of the complexity.

**Table 3.** Estimated complexity of solving $\rho ISD$ using the $F_4$ algorithm for $\ell = 2$, $\rho = 3^{1/2}$.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>$2^{84}$</td>
</tr>
<tr>
<td>256</td>
<td>$2^{152}$</td>
</tr>
<tr>
<td>512</td>
<td>$2^{237}$</td>
</tr>
</tbody>
</table>
Security analysis

3. Rank Attacks (on the code and on the dual code)

\[ G = \begin{pmatrix}
  I_k & \cdot \\
  \cdot & \cdot \\
\end{pmatrix} \]
Security analysis

3. Rank Attacks (on the \textit{code} and on the \textit{dual code})

If $k_1$ is too small, decoding is fast but system is unsecure.
Security analysis

3. Rank Attacks (on the code and on the dual code)

If $k_1$ is too small, decoding is fast but system is unsecure.

If $k_1$ is small, decoding is affordably fast, and for the encoding scheme the security can be increased by increasing the $n_w$. 

A new code based public key encryption and signature scheme based on list decoding
Security analysis

3. Rank Attacks (on the code and on the dual code)

If $k_1$ is too small, decoding is fast but system is unsecure.

If $k_1$ is small, decoding is affordably fast, and for the encoding scheme the security can be increased by increasing the $n_w$.

But $n_w$ can not be too big, because Groebner attacks will break the system.
Security analysis

3. Rank Attacks (on the code and on the dual code)

If $k_1$ is too small, decoding is fast but system is unsecure.

If $k_1$ is small, decoding is affordably fast, and for the encoding scheme the security can be increased by increasing the $n_w$.

But $n_w$ can not be too big, because Groebner attacks will break the system.

Code rates 1:4, 1:5, 1:6
Security analysis

3. Rank Attacks (on the code and on the dual code)

For encryption, $k_1$ values in range 20 – 34 and code rates 1:4, 1:5, 1:6 offer security against known attacks in range $2^{80} - 2^{160}$
Security analysis

3. Rank Attacks (on the code and on the dual code)

For encryption, $k_1$ values in range 20 – 34 and code rates 1:4, 1:5, 1:6 offer security against known attacks in range $2^{80} - 2^{160}$

For signatures, $k_1$ values in range 256 – 1024 and code rates around 1:2 offer security against known attacks in range $2^{80} - 2^{160}$
3. Rank Attacks (on the code and on the dual code)

\[ Pr_{\text{rank}} = \left( \frac{n/\ell - (K_t/\ell + 1)}{N_t/\ell - (K_t/\ell + 1)} \right) \left( \frac{n/\ell}{N_t/\ell} \right)^{-1} \]

The rank computation takes approximately \( k(K_t + \ell)^{\omega^{-1}} \) operations, where \( \omega \) is the linear algebra constant.

<table>
<thead>
<tr>
<th>( \hat{k} )</th>
<th>256</th>
<th>512</th>
<th>768</th>
<th>1024</th>
<th>1280</th>
<th>1536</th>
<th>1792</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_1 )</td>
<td>20</td>
<td>22</td>
<td>24</td>
<td>26</td>
<td>28</td>
<td>30</td>
<td>32</td>
<td>34</td>
</tr>
<tr>
<td>Complexity of the attack</td>
<td>( 2^{82.9} )</td>
<td>( 2^{103.7} )</td>
<td>( 2^{118.9} )</td>
<td>( 2^{132.4} )</td>
<td>( 2^{144.9} )</td>
<td>( 2^{156.9} )</td>
<td>( 2^{168.5} )</td>
<td>( 2^{180} )</td>
</tr>
</tbody>
</table>
Security analysis

4. Cheap distinguishers or equivalent keys finders (brought to us by Nicolas Sendrier)

From the elements in $L_1$ we build up the temporary list $T_1$ of all possible decodings of $y_0$ having length $4 + 1 = 5$:

$$T_1 = \{(101001)\}.$$ 

Thus, in this case we obtain a unique decoding.

The efficiency of the list decoding algorithm depends on the size of the lists $L_0, L_1, \ldots, L_w$, and whether during the decoding process each new list has a smaller size than the previous one. If the size of the lists decreases, the overall complexity is dominated by the size of the initial list $L_0$. Therefore, given a parameter $k_1$ (which determines $L_0$), we want to impose constrains on the values of $n_i/k_i$ in order to avoid “blow-up” of the list sizes.

Proposition 3. Let $E[|L_i|]$ denote the expected value of the size of the lists $L_1, L_2, \ldots, L_w$. Then $|L_0| \geq E[|L_1|] \geq \cdots \geq E[|L_w|]$ if and only if

$$\frac{n_i}{k_i} \geq \frac{\log_2 \rho}{1 - \log_2 \rho} \text{ for all } 2 \leq i \leq w.$$
Choosing Parameters

One important issue with any cryptographic primitive is its efficiency for a given level of claimed security. For public-key primitives, this can be examined by analyzing the sizes of the private and public key, and the number of operations necessary for encryption, decryption, signing and verification.

![Graphs showing public key size and complexity for encryption and signature](image)

**Fig. 5.** Comparison between the complexity of decoding and the distinguishing attacks for encryption and signature. Dashed horizontal lines denote three security levels: $2^{80}$, $2^{96}$ and $2^{128}$. 
Some concrete parameter proposals

<table>
<thead>
<tr>
<th>Variant</th>
<th>Security level</th>
<th>$n$</th>
<th>$k$</th>
<th>Public key size [KB]</th>
<th>Decoding operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>$2^{80}$</td>
<td>1280</td>
<td>256</td>
<td>40</td>
<td>$2^{19}$</td>
</tr>
<tr>
<td>Encryption</td>
<td>$2^{96}$</td>
<td>2560</td>
<td>512</td>
<td>160</td>
<td>$2^{20.6}$</td>
</tr>
<tr>
<td>Encryption</td>
<td>$2^{128}$</td>
<td>5120</td>
<td>1024</td>
<td>640</td>
<td>$2^{23.8}$</td>
</tr>
<tr>
<td>Signature</td>
<td>$2^{80}$</td>
<td>1208</td>
<td>256</td>
<td>37.7</td>
<td>$2^{19}$</td>
</tr>
<tr>
<td>Signature</td>
<td>$2^{96}$</td>
<td>2416</td>
<td>512</td>
<td>151</td>
<td>$2^{20.6}$</td>
</tr>
<tr>
<td>Signature</td>
<td>$2^{128}$</td>
<td>4832</td>
<td>1024</td>
<td>604</td>
<td>$2^{23.8}$</td>
</tr>
</tbody>
</table>
Potential of these Staircase-Generator Codes for other applications

- *Approaching Maximum Embedding Efficiency on Small Covers Using Staircase-Generator Codes*, S. Samardjiska and D. Gligoroski, 2015 IEEE International Symposium on Information Theory, June 14-19, 2015, Hong Kong

- By applying a similar approach as in the signature variant of the public key scheme, but used for **steganographic matrix embedding**, these codes achieve almost the upper theoretical bound of the embedding efficiency for sizes of the covers in the range of 1000 – 1500.

- Other steganographic schemes based on matrix embedding that offer embedding efficiency close to the theoretical bound are based on the low-density generator matrix (LDGM) codes, and they achieve that bound for large covers in the range $10^5 – 10^6$.

- These Staircase-Generator Codes achieve the upper theoretical bound with **two or three orders of magnitude** smaller covers.
Thank you for your attention!