Code-Based Cryptography
An Introduction

2021년 11월

조선대학교 정보통신공학부

김영식†

Department of Information and Communication Engineering
Chosun University

Contents

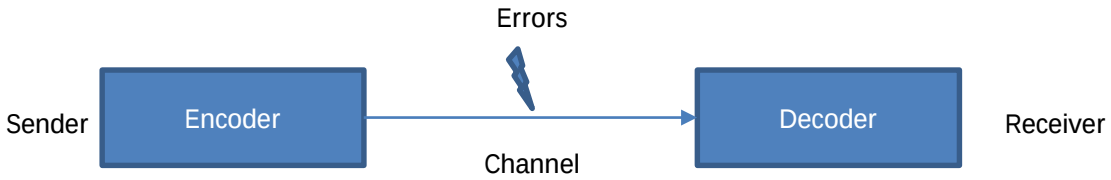
- Introduction to Code
- McEliece Cryptosystems
- Goppa Codes
- Attacks on Code-Based Cryptography
- NIST PQC Finalists: Classic McEliece
- NIST PQC Alternates: BIKE, HQC

CODE-BASED CRYPTOGRAPHY

1. **Introduction to Code**
2. McEliece Cryptosystems
3. Goppa Codes
4. Attacks on Code-Based Cryptography
5. NIST PQC Finalists: Classic McEliece
6. NIST PQC Alternates: BIKE, HQC

Coding Theory

- The sender uses an encoder to transform a message into a codeword by adding redundancy.
  - **Goal**: Protect against errors in a noisy channel

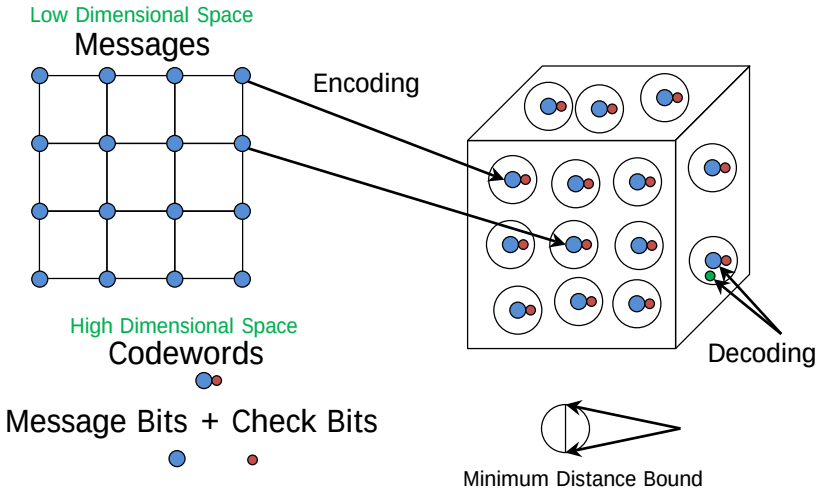


- The receiver uses a decoding algorithm to correct errors which might have occurred during transmission.

Summary of Linear Block Codes

Definition: A linear block code of length n over $GF(q)$ is a k-dimensional subspace of $GF(q)^n$.

→ **Notation**: $[n, k]$ linear block code



It is linear because the linear combinations of valid codewords are also valid codewords.

Summary of Linear Block Codes

- Consider an $[n, k]$ binary linear block code C.

  By the definition, C is the set of all n-bit vectors formed by linear combinations over $GF(2)$ of k linearly independent basis vectors $g_1, g_2, \ldots, g_k$.

  → A codeword vector c is generated by

$$c = m_0 g_1 + m_1 g_2 + \ldots + m_{k-1} g_k = \begin{pmatrix} m_0 & m_1 & \ldots & m_{k-1} \end{pmatrix} \begin{pmatrix} g_1 \\ g_2 \\ \ldots \\ g_k \end{pmatrix} = mG$$

  where $m = k$-bit information (or message) vector, G = generator matrix.

Summary of Linear Block Codes

- An $r \times n$ parity-check matrix H can be formed from the $k \times n$ generator matrix G such that $GH^T = 0$ (or $HG^T = 0$) where $r = n - k$.

  Ex) For a linear block code,

$$G = [I_k ; P] = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \Leftrightarrow H = [-P^T ; I_r] = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

  Syndrome computation of a received vector r:

$$s^T = Hr^T = H\left(c^T + e^T\right) = He^T$$

  where s is the syndrome vector, r is the received vector, c is the transmitted codeword, and e is the channel error vector.

Hamming Metric

- The Hamming distance of $x, y \in GF(2)^n$ is

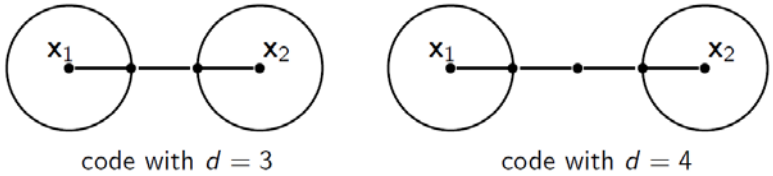
$$\text{dist}(\mathbf{x}, \mathbf{y}) = \#\{i \in \{1, \ldots, n\} \ : x_i \neq y_i\}$$

- The Hamming weight of $x \in GF(2)^n$ is

$$\text{wt}(\mathbf{x}) = \#\{i \in \{1, \ldots, n\} \ : x_i \neq 0\}$$

- The minimum distance of a linear code C is defined as

$$d(C) = \min_{\substack{\mathbf{x},\mathbf{y} \in C \\ \mathbf{x} \neq \mathbf{y}}} \text{dist}(\mathbf{x}, \mathbf{y}) = \min_{\substack{\mathbf{x} \in C \\ \mathbf{x} \neq \mathbf{0}}} \text{wt}(\mathbf{x})$$



code with $d = 3$ code with $d = 4$

Decoding Problem

- **Syndrome Decoding Problem**
  - Given an $(n-k) \times n$ binary matrix H, a vector s in $GF(2)^{n-k}$ and $w \geq 0$, find e in $GF(2)^n$ of weight $\leq w$ such that $He = s$.

> Find the closest codeword $\boldsymbol{x}$ in $\boldsymbol{\Gamma}$ to a given $\boldsymbol{y}$ in $\boldsymbol{F_2^n}$, assuming that there is a unique closest word.

  - Decoding a generic binary code of length n and without knowing anything about its structure requires about $2^{(0.5+o(1))n/\log_2(n)}$ binary operations (if code rate is ½).
  - Coding theory deals with efficiently decodable codes, not random ones as in McEliece crypto.

Decoding Problem

- **Structured Codes for Efficient Decoding**
  - There are lots of code families with fast decoding algorithms
    - ✓Goppa codes/alternant codes, RS codes, Gabidulin codes, RM codes, algebraic-geometric codes, convolutional codes, LDPC codes etc.
  - All those decoding algorithms use information on the structure of the code.
- **Decoding for random codes are very hard**
  - Random code often outperforms structured codes.
  - Decoding is exponentially complicated.

"All codes are good, except those that we know of" Shannon Lecture in 1995

David Forney

CODE-BASED CRYPTOGRAPHY

1. Introduction to Code
2. **McEliece Cryptosystems**
3. Goppa Codes
4. Attacks on Code-Based Cryptography
5. NIST PQC Finalists: Classic McEliece
6. NIST PQC Alternates: BIKE, HQC

McEliece Cryptography

- In 1971, Goppa proposed a family of codes with fast decoders for many matrices *H*.
- In 1978, McEliece introduced a public key cryptosystem based on error correcting codes.
- There is no efficient structural attacks that might distinguish between a permuted Goppa code used by McEliece and a random code.
  - Original parameter designed for 2^{64} security.
  - Easily scale up for higher security.

McEliece

- The problem of syndrome decoding is known as NP-hard.
  - E.R. Berlekamp, R.J. McEliece, and H.C. van Tilborg, “On the inherent intractability of certain coding problems” *IEEE Trans. Inf. Theory*, vol. 24, no. 3, May 1978.

History of Code-Based Cryptography

- 1978 McEliece: binary Goppa codes
- 1986 Niederreiter variant based on GRS codes
- 1991 Gabidulin, Paramonov, Tretjakov: Gabidulin codes
- 1994 Sidelnikov: Reed-Muller codes
- 1996 Janwa-Moreno: Algebraic Geometric codes
- 1990s many propositions with LDPC codes
- 2003 Alekhnovich: Alekhnovich system
- 2005 Berger-Loidreau: subcodes of GRS codes
- 2006 Wieschebrink, GRS codes + random columns in the generator matrix

History of Code-Based Cryptography

- 2008 Baldi-Bodrato-Chiaraluce: MDPC codes
- 2010 Bernstein, Lange, Peters: non-binary wild Goppa codes
- 2012 Misoczki-Tillich-Barreto-Sendrier: MDPC codes
- 2012 Londahl-Johansson: convolutional codes
- 2013 Gaborit, Murat, Ruatta, Zemor: LRPC codes
- 2014 Shrestha, Kim: polar codes
- 2014 Hooshmand, Shooshtari, Eghlidos, Aref: subcodes of polar codes
- 2017 NIST submissions

McEliece Cryptography

- **Based on binary Goppa code**
  - Let C be a length n binary Goppa code Γ of dimension k with minimum distance $2t + 1$ where $t = (n - k)/\log_2(n)$
    - ✓Original parameters: n = 1024, k = 524, t = 50 (**not enough**)

McEliece Cryptography

- **Private Key**
  - The McEliece private (secret) key consists of **a generator matrix G** for Γ, **an efficient t-error correcting decoding algorithm** for Γ; an **$n \times n$ permutation matrix P** and **a nonsingular $k \times k$ matrix S**.
  - n, k, t are public; but Γ, P, S are randomly generated secrets
- **Public Key**
  - The McEliece public key is the **$k \times n$ matrix $G' = SGP$**
- **Ciphertext**
  - A codeword to which some errors have been added

McEliece Cryptography

- **Encryption Algorithm**
  - **Input**: m, $K_{pub} = (G', t)$
  - **Output**: Cipher-text c

1. Encode the message m as a binary string of length k
2. $c' \leftarrow mG'$
3. Generate a random n-bit error vector z containing at most t ones
4. $c = c' + z$

  - Simple matrix multiplication with the input message at step 2
    - ✓Efficiently implement
  - Distribute t random errors on the resulting codeword at step 4
    - ✓Need an appropriate random number generator

McEliece Cryptography

- **Decryption Algorithm**

$$c = mSGP + z$$

  - **Input**: c, $K_{sec} = (P^{-1}, G, S^{-1})$
  - **Output**: Plain-text m

1: $c' \leftarrow cP^{-1}$

2: Use a decoding algorithm for the code Γ to decode c' to $m' = mS$

3: $m \leftarrow m'S^{-1}$

4: return m

$$cP^{-1}H^T = mSGH^T + zP^{-1}H^T = zP^{-1}H^T$$

  - Require more complex operations in binary extension fields

McEliece Cryptography

- **Advantages**
  - The **encryption and decryption are fast**
  - Security reductions are tight
  - Efficient key generation with the growth of the key size, the security grows much faster
  - The encryption and decryption have a **low complexity**
  - A signature scheme can be constructed based on the Niederreiter scheme
- **Disadvantages**
  - **The private and public keys are large matrices**
    - ✓The public key is 100 kB to several MB long
  - This is why the algorithm has been rarely used in practice

Niederreiter Encryption Scheme

- **Proposed by Niederreiter in 1986**
- **Key Generation**
  - Given a t-error correcting linear $[n, k, d]$-code C with parity check matrix H
  - **Private key**: (S, H, P) where S is a scrambling matrix and P is a permutation matrix
  - **Public key**: $H' = SHP$

Niederreiter Encryption Scheme

- **Encryption**
  - Message m is represented as an error vector e in F_2^n, $wt(e) \leq t$
  - $s' \leftarrow H'e^T$: **Syndrome decoding problem**

$$s' = SHPe^T$$

- **Decryption**
  - Let Φ_H be a t-error correcting decoding algorithm of C with H
  - $e \leftarrow P^{-1}\Phi_H(S^{-1}s')$

CODE-BASED CRYPTOGRAPHY

1. Introduction to Code
2. McEliece Cryptosystems
3. **Goppa Codes**
4. Attacks on Code-Based Cryptography
5. NIST PQC Finalists: Classic McEliece
6. NIST PQC Alternates: BIKE, HQC

Binary Goppa Code

- **Binary Goppa Code**
  - Let $q = 2^m$
  - A binary Goppa code is often defined by
    - ✓A list $L = (a_1, \dots, a_n)$ of n distinct elements in F_q, called the **support**
    - ✓A degree-t polynomial $g(x) \in F_q[x]$ such that $g(a) \neq 0$ for all $a \in L$.
      - $g(x)$ is called the **Goppa polynomial**
  - The corresponding binary Goppa code $\Gamma(L, g)$ is

$$\{ c \in F_2 | S(c) = \frac{c_1}{x - a_1} + \frac{c_2}{x - a_2} + \cdots + \frac{c_n}{x - a_n} \equiv 0 \; mod \; g(x) \}$$

  - This code is linear, i.e., $S(b + c) = S(b) + S(c)$ and has length n.

Parameters of Goppa Code

- **Notation $[n, k, d]$ Goppa code**
  - n: the length of the codeword c
  - k: dimension
  - d: minimum distance
- **Theorem**
  - The Goppa code $\Gamma(L, g(z))$ of size n is a linear code over $GF(q)$ with the properties
    - ✓The dimension of the code satisfies $k \geq n - mt$
    - ✓The minimum distance of the code satisfies $d \geq t + 1$
- **Separable polynomials**
  - The polynomials only with roots of multiplicity 1 (Square Free)
  - Square-free: If $g = g_1^{d_1} g_2^{d_2} \cdots g_l^{d_l}$, then $d_i = 1$ for all i.

Binary Goppa Code

- **Property of $\Gamma(\mathbf{L}, \mathbf{g})$**
  - Dimension $k \geq n - mt$ (usually equality holds).
  - Parity check matrix

$$\begin{pmatrix} 1/g(a_1) & 1/g(a_2) & \cdots & 1/g(a_n) \\ a_1/g(a_1) & a_2/g(a_2) & & a_n/g(a_n) \\ \vdots & & \ddots & \vdots \\ a_1^{t-1}/g(a_1) & a_2^{t-1}/g(a_2) & \cdots & a_n^{t-1}/g(a_n) \end{pmatrix}$$

  - **Minimum distance**: $d \geq t + 1$
  - **Theorem**
    - ✓ Let $\Gamma(L, g(z))$ be a binary Goppa code with a ***separable*** polynomial $g(z)$ of degree t. Then $\Gamma(L, g(z))$ has a minimum distance $d \geq 2t + 1$.
  - There exist efficient t-error decoding algorithms when g is square-free.

Binary Goppa Code

- **Dimension $k \geq n - mt$**
  - $g(a_i) \neq 0$ implies $\gcd\big(x - a_i, g(x)\big) = 1$, thus get polynomials

$$\frac{1}{x - a_i} \equiv f_i(x) = \sum_{j=0}^{t-1} f_{i,j} x^j \bmod g(x),\ f_i(x) \in F_{2^m}[x]$$

  - $\sum_{i=1}^{n} c_i f_i(x) \equiv 0 \bmod g(x) \Rightarrow \sum_{i=1}^{n} c_i f_i(x) = 0$, in other words,

$$\begin{pmatrix} f_{1,0} & f_{2,0} & \cdots & f_{n,0} \\ f_{1,1} & f_{2,1} & & f_{n,1} \\ \vdots & & \ddots & \vdots \\ f_{1,t-1} & f_{2,t-1} & \cdots & f_{n,t-1} \end{pmatrix} \cdot (c_1 \quad \ldots \quad c_n)^T = (0 \quad \ldots \quad 0)^T \in F_{2^m}^t$$

  - These are t conditions over F_{2^m}, so tm conditions over F_2.
  - Some conditions might be linearly dependent, so $k \geq n - tm$.

Binary Goppa Code

- **Minimum Distance $d \geq t + 1$**
  - Put $s(x) = S(c) = \sum_{i=1}^{n} c_i/(x - a_i)$

$$s(x) = \sum_{c_i=1} \frac{1}{x - a_i} = \frac{\left(\sum_{c_i=1} \prod_{j \neq i}(x - a_j)\right)}{\prod_{c_i=1}(x - a_i)}$$

$$= \frac{f'(x)}{f(x)} \equiv 0 \; mod \; g(x)$$

  - Note that $\deg(f) = wt(c) \Rightarrow \deg\left(f'(x)\right) \leq wt(c) - 1$
  - Multiply both sides of $\equiv$ by $f(x)$ to obtain

$$f'(x) = \sum_{c_i=1} \prod_{j \neq i} (x - a_j) \equiv 0 \; mod \; g(z)$$

  - $f'(x)$ is a multiple of $g(x)$. Is $f'(x) = 0$?
    - ✓Note that $\gcd(f'(x), x - a_i) = 1$ if $c_i = 1$, so
    - ✓$\gcd\left(f'(x), f(x)\right) = 1 \Rightarrow \; f'(x) \neq 0$
  - Therefore, $f'(x)$ has degree $\geq t \Rightarrow wt\,(c) \geq t + 1$

Binary Goppa Code

- **Minimum Distance $d \geq 2t+1$ for square-free $g(x)$**
  - Recall that $f'(x) \equiv 0 \bmod g(x)$
  - Let $wt(c) = w$, so $\deg(f) = w$ and $\deg(f') \leq w-1$
  - Observe that over F_{2^m}
    - ✓ $\left(f_{2i+1}x^{2i+1}\right)' = f_{2i+1}x^{2i}$, $\left(f_{2i}x^{2i}\right)' = 0 \cdot f_{2i}x^{2i-1} = 0$
    - ✓ thus, $f'(x)$ contains only terms of even degree
  - Note that over F_{2^m}: $x^{2i} + x^{2j} = \left(x^i + x^j\right)^2$ and in general

$$f'(x) = \sum_{i=0}^{\lfloor (w-1)/2 \rfloor} f_{2i+1}x^{2i} = \left(\sum_{i=0}^{\lfloor (w-1)/2 \rfloor} \sqrt{f_{2i+1}}x^i \right)^2 = F^2(x)$$

  - Recall that $f'(x) \neq 0 \Rightarrow F(x) \neq 0$
  - Since $g(x)$ is square-free, $g(x) | F^2(x) \Rightarrow g(x) | F(x)$, thus
  - $\left\lfloor \frac{w-1}{2} \right\rfloor \geq t \Rightarrow w \geq 2t+1$

Goppa code Examples

- **Finite Field $GF(2^4)$**
  - Irreducible polynomial $x^4 + x + 1$

$$
\begin{array}{lcccccccccl}
1 & = & 1 & & & & & & & = & (1,0,0,0)^T; \\
\alpha & = & & & \alpha & & & & & = & (0,1,0,0)^T; \\
\alpha^2 & = & & & & & \alpha^2 & & & = & (0,0,1,0)^T; \\
\alpha^3 & = & & & & & & & \alpha^3 & = & (0,0,0,1)^T; \\
\alpha^4 & = & 1 & + & \alpha & & & & & = & (1,1,0,0)^T; \\
\alpha^5 & = & & & \alpha & + & \alpha^2 & & & = & (0,1,1,0)^T; \\
\alpha^6 & = & & & & & \alpha^2 & + & \alpha^3 & = & (0,0,1,1)^T; \\
\alpha^7 & = & 1 & + & \alpha & & & + & \alpha^3 & = & (1,1,0,1)^T; \\
\alpha^8 & = & 1 & & & + & \alpha^2 & & & = & (1,0,1,0)^T; \\
\alpha^9 & = & & & \alpha & & & + & \alpha^3 & = & (0,1,0,1)^T; \\
\alpha^{10} & = & 1 & + & \alpha & + & \alpha^2 & & & = & (1,1,1,0)^T; \\
\alpha^{11} & = & & & \alpha & + & \alpha^2 & + & \alpha^3 & = & (0,1,1,1)^T; \\
\alpha^{12} & = & 1 & + & \alpha & + & \alpha^2 & + & \alpha^3 & = & (1,1,1,1)^T; \\
\alpha^{13} & = & 1 & & & + & \alpha^2 & + & \alpha^3 & = & (1,0,1,1)^T; \\
\alpha^{14} & = & 1 & & & & & + & \alpha^3 & = & (1,0,0,1)^T.
\end{array}
$$

Goppa code Examples

- **Goppa polynomial**
  - $g(z) = (z + \alpha)(z + \alpha^{14}) = z^2 + \alpha^7 z + 1$
  - $L = \{\alpha^i | 2 \leq i \leq 13\}$
- **Parameters**
  - In this case, $q = 2$, $m = 4$, $n = 12$, and $t = 2$
  - In addition, k must be at least 12 – 8 = 4 and d is at least 5
  - $g_1 = \alpha^7$, $g_2 = 1$
  - $h_i = g(\alpha_i)^{-1}$

$$
\begin{aligned}
h_1 &= g(\alpha^2)^{-1} = (\alpha^4 + \alpha^9 + 1)^{-1} \\
&= ((1,1,0,0)^T + (0,1,0,1)^T + (1,0,0,0)^T)^{-1} \\
&= ((0,0,0,1)^T)^{-1} = \alpha^{-3} = \alpha^{12}, \\
h_2 &= g(\alpha^3)^{-1} = (\alpha^6 + \alpha^{10} + 1)^{-1} \\
&= ((0,0,1,1)^T + (1,1,1,0)^T + (1,0,0,0)^T)^{-1} \\
&= ((0,1,0,1)^T)^{-1} = \alpha^{-9} = \alpha^6, \text{ etc.}
\end{aligned}
$$

Goppa code Examples

- **Parity check matrix H**

$$H = \begin{pmatrix} -(\alpha^2+\alpha^7)h_1 & \dots & -(\alpha^{13}+\alpha^7)h_{12} \\ -h_1 & \dots & -h_{12} \end{pmatrix}$$

$$H = \begin{pmatrix} \alpha^9 & \alpha^{10} & \alpha^9 & \alpha^{14} & \alpha^6 & 0 & \alpha^{10} & \alpha^8 & \alpha^2 & \alpha^7 & \alpha^{14} & \alpha^6 \\ \alpha^{12} & \alpha^6 & \alpha^6 & \alpha & \alpha^{11} & 1 & \alpha^{14} & \alpha^8 & \alpha^{11} & \alpha^{14} & \alpha^{12} & \alpha \end{pmatrix}$$

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ - & - & - & - & - & - & - & - & - & - & - & - \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Decoding of Goppa Codes

- Let y be a received word, containing r errors, with $2r + 1 \leq d$, or $r \leq \lfloor \frac{t}{2} \rfloor$ in the case that $d = t + 1$ (the lower bound on the minimum distance)

$$(y_1, \ldots, y_n) = (c_1, \ldots, c_n) + (e_1, \ldots, e_n)$$

  with $e_i \neq 0$ in exactly r places.
- To correct errors, we have to find the error locations and corresponding error values.
  - The set of error locations $E = \{i \mid e_i \neq 0\}$
  - The corresponding error values e_i for i in E.
- **Definition**: The error locator polynomial $\sigma(z)$ and the error evaluator polynomial $w(z)$ are

$$\sigma(z) = \prod_{i \in B} (z - a_i)$$

$$\omega(z) = \sum_{i \in B} e_i \prod_{j \in B, j \neq i} (z - a_j)$$

  - The error locations directly follow from the roots of $\sigma(z)$.

Decoding of Goppa Codes

- For correcting errors in a codeword, one has to solve the **key equation** $\sigma(z)s(z) = w(z)(mod\ g(z))$

- Since $g(z)$ is known and the syndrome $s(z)$ can be computed, one has to solve a system of t equations with $2r$ unknowns, namely $\sigma_0, \sigma_1, \ldots, \sigma_{r-1}$, and $w_0, w_1, \ldots, w_{r-1}$, where $\sigma(z) = \sigma_0 + \sigma_1 + \cdots + \sigma_{r-1}z^{r-1} + z^r$ and $w(z) = w_0 + w_1 z + \cdots + w_{r-1}z^{r-1}$.
- Because $2r \leq t$, we know there is a unique solution.

Binary Goppa Code

- **Decoding of $c + e$**
  - Fix e. Recall $\sigma(x) = \prod_{i,e_i \neq 0}(x - \alpha_i)$ [Error Locator Polynomial]
  - Split into odd and even terms: $\sigma(x) = A^2(x) + xB^2(x)$
  - Note $s(x) = \sigma'(x)/\sigma(x)$ and $\sigma'(x) = B^2(x)$
  - Thus
    - ✓ $B^2(x) \equiv \sigma(x)s(x) \equiv \left(A^2(x) + xB^2(x)\right)s(x) \bmod g(x)$
    - ✓ $B^2(x)\left(x + \frac{1}{s(x)}\right) \equiv A^2(x) \bmod g(x)$
  - Put $v(x) \equiv \sqrt{x + \frac{1}{s(x)}} \bmod g(x)$, then
    - ✓ $A(X) = B(x)v(X) \bmod g(x)$
  - Use Extended EA on v and g, stop when
    - ✓ $A(x) = B(x)v(x) + h(x)g(x)$,

    with $\deg(A) \leq \lfloor \frac{t}{2} \rfloor$, $\deg(B) \leq \lfloor \frac{t-1}{2} \rfloor$

CODE-BASED CRYPTOGRAPHY

1. Introduction to Code
2. McEliece Cryptosystems
3. Goppa Codes
4. **Attacks on Code-Based Cryptography**
5. NIST PQC Finalists: Classic McEliece
6. NIST PQC Alternates: BIKE, HQC

A Long-Studied Problems

- **Correct t errors in a code of length n and dimension k has cost $O(2^{\alpha\left(\frac{k}{n},\frac{t}{n}\right)n})$**

Authors	Year	Max α(R, τ)
Prange	1962	0.1207
Stern	1988	0.1164
Dumer	1991	0.1162
May-Meurer-Thomae	2011	0.1114
Becker-Joux-May-Meurer	2012	0.1019
May-Ozerov	2015	0.0966
Both-May	2017	0.0953
Both-May	2018	0.0885

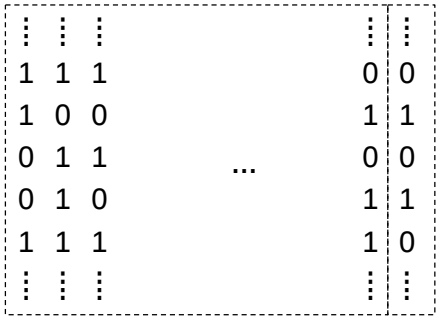
Generic Decoder

- **Build a decoder which gets as input**
  - A parity-check matrix H
  - A ciphertext y in $GF(2)^n$
  - An integer $w \geq 0$
  - The algorithm tries to **determine an error vector e** of weight = w such that

$$s = Hy = He$$

Generic Decoder

- **Problem**
  - **Goal**: find w columns of H produce s after XOR.
  - Given an $r \times n$ matrix, a syndrome s.

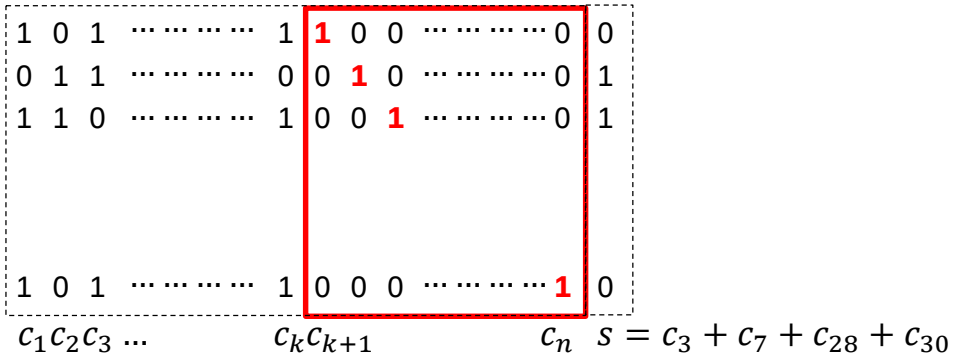


$s = c_2 + c_3 + c_{18} + c_{20} + c_{24}$

  - Can arbitrarily permute rows/columns without changing the problem.

Generic Decoder

- **Problem**
  - **Goal**: find w columns of H produce s after XOR.
  - Given an $r \times n$ matrix, a syndrome s.



  - Can add one row to another and build identity matrix.

Information Set Decoding

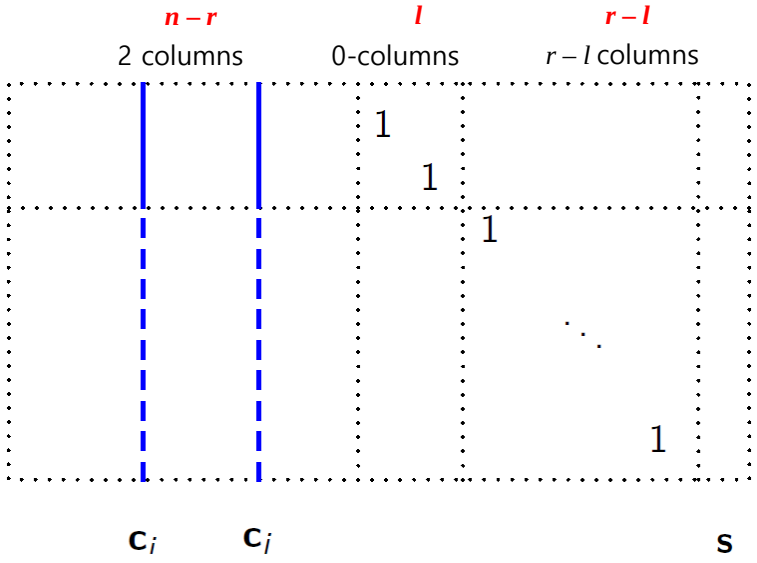
- **Prange in 1962**
  - Perhaps XOR involves none of the first $n - r$ columns.
  - If so, immediately see that s is constructed from w columns from the identity submatrix.
  - If not, re-randomize and restart.
  - This is a probabilistic algorithm.
  - Expect about $\binom{n}{w} / \binom{r}{w}$ iterations.

Information Set Decoding

- **Lee–Brickell**
  - More likely that XOR involves exactly p of the first $n - r$ columns.
  - Check for each p-tuple $(i_1, i_2, \ldots, i_p)$ with $1 < i_1 < i_2 < \cdots < i_p \leq k$ if $s + c_{i_1} + c_{i_2} + \cdots + c_{i_p}$ has weight $w - p$.
  - Expect about $\binom{n}{w} / \binom{n-r}{p}\binom{r}{w-p}$ iterations, each checking sum $s + c_{i_1} + c_{i_2} + \cdots + c_{i_p}$
  - Cost for computing these sums grows with p.
  - Choosing $p = w/2$ would minimize # iterations but increase cost of each iterations enormously → $p = 2$ is often used.

Information Set Decoding

- **Leon, Krouk, Stern**
  - Check for each pair (i,j) with $1 < i < j \leq n - r$ if $s + c_i + c_j$ has weight $w - 2$ and the first l bits all zero.
  - Early abort if $s + c_i + c_j \neq 0$ on first l bits.



Information Set Decoding

- **Leon (1989), Krouk (1989)**
  - Check each p-tuple $(i_1, i_2, \ldots, i_p)$ with $1 < i_1 < i_2 < \cdots < i_p \leq k$

    if $s + c_{i_1} + c_{i_2} + \cdots + c_{i_p}$ has weight $w - p$ and the first l bits

    all zero.
  - Fast to test, iteration cost decreases.
  - Expect about $\binom{n}{w} / \binom{n-r}{p}\binom{r-l}{w-p}$ iterations
  - Only small improvement from Lee–Brickell.

Information Set Decoding

- **Stern in 1989**
  - Enforce 0's on first l bits using a meet-in-the-middle approach
    → square-root improvement.
  - Split first $n-r$ columns in two disjoint sets of equal size
  - Draw c_i's from the left, c_j's from the right set.
  - Find collisions between first l bits of $s + c_{i_1} + \cdots + c_{i_{p/2}}$ and the first l bits of $c_{j_1} + \cdots + c_{j_{p/2}}$.
  - For each collision, check if $s + c_{i_1} + \cdots + c_{i_{\frac{p}{2}}} + c_{j_1} + \cdots + c_{j_{p/2}}$ has weight $w - p$.
  - Expect about $\binom{n}{w} / \binom{(n-r)/2}{p/2}^2 \binom{r-l}{w-p}$ iterations

CODE-BASED CRYPTOGRAPHY

1. Introduction to Code
2. McEliece Cryptosystems
3. Goppa Codes
4. Attacks on Code-Based Cryptography
5. **NIST PQC Finalists: Classic McEliece**
6. NIST PQC Alternates: BIKE, HQC

Post Quantum Cryptography

- **3rd Round KEM Candidates for NIST PQC Standard**

Finalists

No.	Name	Category	Category	Code
1	**Classic McEliece**	**Code**	Hamming	Goppa
2	**Crystals-Kyber**	**Lattice**	Module-LWE	
3	**NTRU**	**Lattice**	NTRU	
4	**SABER**	**Lattice**	Module-LWR	

Alternates

No.	Name	Category	Category	Code
5	**BIKE**	**Code**	Hamming	QC-MDPC
6	**HQC**	**Code**	Hamming	QC Code
7	**FrodoKEM**	**Lattice**	LWE	
8	**NTRU Prime**	**Lattice**	NTRU	
9	**SIKE**	**Isogeny**	Isogeny	

Classic McEliece

- **Niederriter Type KEM**
  - Conservative code-based cryptography
  - Almost nothing has changed in more than 40 years
  - Short cipher-text (128~240 bytes)
- **IND-CCA KEM**
  - Generate key once and use it many times.
- **Security Foundation**
  - Given random parity-check matrix H and syndrome s, can an attacker efficiently find e with $s = He$?

Classic McEliece

- **Key Generation**
  - Generate a uniform random monic irreducible polynomial $g(x) \in F_q[x]$ of degree t and select a uniform random sequence $(a_1, a_2, \ldots, a_n)$ of n distinct elements of F_q, where $q = 2^m$. [Parameters of Goppa code]
  - Compute the $t \times n$ matrix $H = \{h_{i,j}\}$ over F_q, where $h_{i,j} = a_j^{i-1}/g(a_j)$ for $i = 1, \ldots, t$ and $j = 1, \ldots, n$.
  - Form an $mt \times n$ matrix H over F_2 by replacing each entry $c_0 + c_1 z + \cdots + c_{m-1} z^{m-1}$ of H with a column of t bits $c_0, c_1, \ldots, c_{m-1}$.

Classic McEliece

- **Key Generation (Cont'd)**
  - Apply Gaussian elimination to H to reduce H to systematic form $(I_{n-k}|\boldsymbol{T})$, where I_{n-k} is an $(n-k) \times (n-k)$ identity matrix, where $k = n - mt$.
  - Generate a uniform random n-bit string s.
  - Put $\Gamma = (g, a_1, a_2, \ldots, a_n)$ and output (s, Γ) as private key and T as public key.

Classic McEliece

- **Encapsulation**
  - Generate a uniform random vector e in F_2^n with weight t
  - Use encoding subroutine on e and public key T to compute C_0
    [Syndrome]
  - Compute $C_1 = H(2, e)$ and put $C = (C_0, C_1)$
  - Compute $K = H(1, e, C)$
  - Output session key K and ciphertext C

Classic McEliece

- **Decapsulation**
  - Split the ciphertext C as (C_0, C_1) with $C_0 \in F_2^{n-k}$ and $C_1 \in F_2^l$
  - Set $b \leftarrow 1$
  - Use the decoding subroutine on C_0 and private key Γ to compute e. If the subroutine returns $\perp$, set $e \leftarrow s$ and $b \leftarrow 0$
  - Compute $C_1' = H(2, e)$
  - If $C_1' \neq C_1$, set $e \leftarrow s$ and $b \leftarrow 0$
  - Compute $K = H(b, e, C)$ and output session key K

Classic McEliece

- **Pros:**
  - Security asymptotic unchanged by more than 40 years of cryptanalysis
  - Efficient and straightforward conversion of OW-CPA PKE into IND-CCA2 KEM
  - Very short ciphertexts
  - Constant-time software implementations
  - Fast encapsulation and decapsulation
  - Open-source (public domain) implementations
  - Patent-free
- **Cons:**
  - Very big public keys
  - Slow key generation (but keys can be reused)

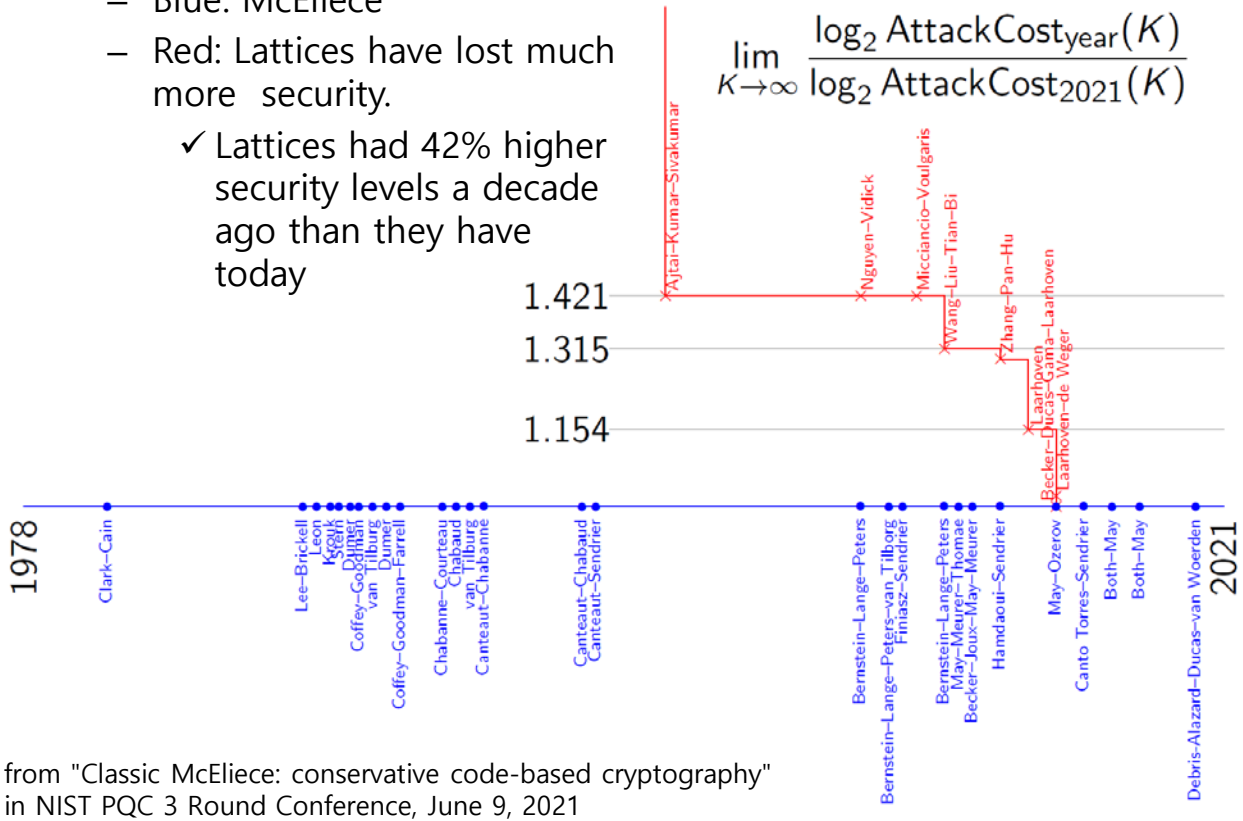
Classic McEliece

	McEliece 348864	McEliece 460896	McEliece 6688128	McEliece 6960119	McEliece 8192128
(n, m, t)	(3488,12,64)	(4608,13,96)	(6688,13,128)	(6960,13,119)	(8192,13,128)
Public Key (bytes)	261,120	524,160	1,044,992	1,047,319	1,357,824
Private Key (bytes)	6,452	13,568	13,892	13,908	14,080
Ciphertext (bytes)	128	188	240	226	240
KeyGen (cycles)	36,627,388	116,914,656	284,468,140	246,291,008	316,116,640
Encapsulation (cycles)	43,832	115,540	149,080	159,116	177,480
Decapsulation (cycles)	134,184	270,856	322,988	300,688	325,744

Timings on Haswell by the submitters

Classic McEliece

- **McEliece Security Stability**
  - Blue: McEliece
  - Red: Lattices have lost much more security.
    - ✓ Lattices had 42% higher security levels a decade ago than they have today



from "Classic McEliece: conservative code-based cryptography" in NIST PQC 3 Round Conference, June 9, 2021

CODE-BASED CRYPTOGRAPHY

1. Introduction to Code
2. McEliece Cryptosystems
3. Goppa Codes
4. Attacks on Code-Based Cryptography
5. NIST PQC Finalists: Classic McEliece
6. **NIST PQC Alternates: BIKE, HQC**

Niederreiter Encryption Scheme: Revisit

- **Key Generation**
  - Given a t-error correcting $[n, k, d]$ code C with parity check matrix H
  - **Private key**: H
  - **Public key**: $H' = [Q|I_{n-k}] \leftarrow H$ in systematic form for shorter key size
  - Equivalently $G' = [I_k|Q^T]$
  - No need for S and P
  - Message m should be randomized using a one-time random token

Niederreiter Encryption Scheme: Revisit

- **Encryption**
  - Message m is represented as an error vector e in F_2^n, $wt(e) \leq t$
  - $s' \leftarrow H'e^T$

$$\boldsymbol{s' = H'e^T}$$

- **Decryption**
  - Let Φ_H be an t-error correcting decoding algorithm
  - $e \leftarrow \Phi_H(s')$

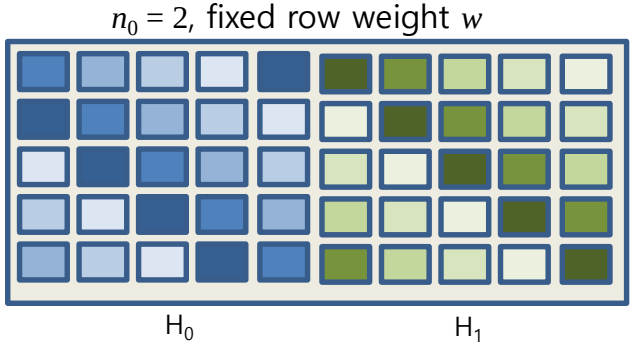
Code Based Cryptography

- **Selection of the employed code is critical**
  - Properties of code determine key size, short keys are essential
  - Structures in codes reduce key size, but can enable attacks
  - Encoding is a fast operation (it is just a matrix-vector multiplication)
  - Efficient decoding required in terms of time and memory
- **Quasi-cyclic moderate density parity check codes [MTSB13]**
  - Short keys due to quasi-cyclic structure
  - Efficient decoding in software and hardware

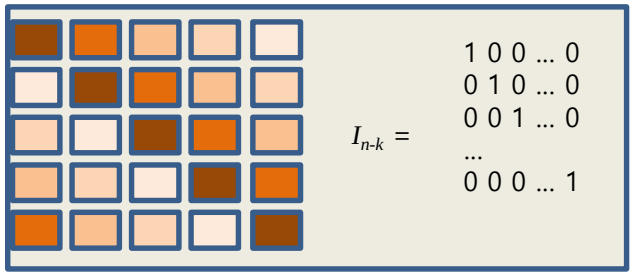
* R. Misoczki, J.P. Tillich, N. Sendrier, and P.S. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *Proc. IEEE ISIT 2013*, pp. 2069-2073.

QC-MDPC Codes

- Private key: QC parity check matrix H



- Public key: systematic parity check matrix H



QC-MDPC Codes for Niederreiter

- **Quasi-cyclic moderate density parity-check codes for Niederreiter**
  - t-error correcting $[n, r, w]$ QC-MDPC code of length $n = n_0 r$, $r = n - k$
  - Parity-check matrix H consists of n_0 blocks with fixed row weight w

QC-MDPC Codes for Niederreiter

- **Key Generation**
  - Generate n_0 first rows of parity-check matrix blocks H_i

    $$H_i \in_R F_2^r \text{ of weight } d_v = w/n_0$$

  - Obtain remaining rows by $r-1$ quasi-cyclic shifts of h_i
  - If H_{n_0-1} is not invertible, start over
  - Private key
    - ✓ $H = [H_0|H_1|\dots|H_{n_0-1}]$
  - Systematic public key
    - ✓ $H' = \left[H_{n_0-1}^{-1} \cdot H\right] = \left[H_{n_0-1}^{-1} \cdot H_0 \middle| H_{n_0-1}^{-1} \cdot H_1 \middle| \dots | \boldsymbol{I_r}\right]$

BIKE

- **Niederreiter-based KEM instantiated with QC-MDPC codes**
  - Leverage Fujisaki-Okamoto CCA Transform
  - State-of-the-art QC-MDPC Decoding Failure Rate analysis
  - Black-Gray-Flip Decoder implemented in constant time

BIKE

- **BIKE Key Encapsulation Mechanism**

KeyGen : $() \mapsto (h_0, h_1, \sigma), h$

Output: $(h_0, h_1, \sigma) \in \mathcal{H}_w \times \mathcal{M}$, $h \in \mathcal{R}$

1: $(h_0, h_1) \xleftarrow{\$} \mathcal{H}_w$
2: $h \leftarrow h_1 h_0^{-1}$
3: $\sigma \xleftarrow{\$} \mathcal{M}$

Encaps : $h \mapsto K, c$

Input: $h \in \mathcal{R}$

Output: $K \in \mathcal{K}$, $c \in \mathcal{R} \times \mathcal{M}$

1: $m \xleftarrow{\$} \mathcal{M}$
2: $(e_0, e_1) \leftarrow \mathbf{H}(m)$
3: $c \leftarrow (e_0 + e_1 h, m \oplus \mathbf{L}(e_0, e_1))$
4: $K \leftarrow \mathbf{K}(m, c)$

Decaps : $(h_0, h_1, \sigma), c \mapsto K$

Input: $((h_0, h_1), \sigma) \in \mathcal{H}_w \times \mathcal{M}$, $c = (c_0, c_1) \in \mathcal{R} \times \mathcal{M}$

Output: $K \in \mathcal{K}$

1: $e' \leftarrow \mathtt{decoder}(c_0 h_0, h_0, h_1)$ ▷ $e' \in \mathcal{R}^2 \cup \{\perp\}$
2: $m' \leftarrow c_1 \oplus \mathbf{L}(e')$ ▷ with the convention $\perp = (0, 0)$
3: **if** $e' = \mathbf{H}(m')$ **then** $K \leftarrow \mathbf{K}(m', c)$ **else** $K \leftarrow \mathbf{K}(\sigma, c)$

$\mathcal{M} = \{0,1\}^\ell$ Message space

$\mathcal{K} = \{0,1\}^\ell$ Key space

NOTATION	
$\mathbb{F}_2$:	Binary finite field.
$\mathcal{R}$:	Cyclic polynomial ring $\mathbb{F}_2[X]/(X^r - 1)$.
$\mathcal{H}_w$:	Private key space $\{(h_0, h_1) \in \mathcal{R}^2 \mid \lvert h_0\rvert = \lvert h_1\rvert = w/2\}$
$\mathcal{E}_t$:	Error space $\{(e_0, e_1) \in \mathcal{R}^2 \mid \lvert e_0\rvert + \lvert e_1\rvert = t\}$
$\lvert g\rvert$:	Hamming weight of a binary polynomial $g \in \mathcal{R}$.
$u \xleftarrow{\$} U$:	Variable u is sampled uniformly at random from the set U.
$\oplus$:	exclusive or of two bits, componentwise with vectors

BIKE

- **Parameters and Performance**

	Level 1	Level 3	Level 5
Public Key (bits)	12,323	24,659	40,973
Private Key (bits)	2,244	3,346	4,640
Ciphertext (bits)	12,579	24,915	41,229
KeyGen (kcycles)	600k	1,780k	-
Encapsulation (cycles)	220k	465k	-
Decapsulation (cycles)	2,220k	6,610k	-

HQC

- **Stand for Hamming metric and Quasi-Cyclic Code**
  - G is he generator matrix of some public code C
    - ✓BCH codes, RS codes, RM codes
  - Secret key: $sk = (x, y)$
  - Public key: $pk = (h, s)$
  - Ciphertext: $ct = (u, v)$
  - $S_w^n(F_2) = \{x \in F_2^n | \omega(x) = w\}$
- **Key Generation**
  - $\text{seed}_h \leftarrow_{\$} \{0,1\}^{\lambda}$, $h \leftarrow \text{seed}_h \in F_2^n$
  - $x, y \leftarrow_{\$} S_w^n(F_2)$, $s \leftarrow x + hy$

HQC

- **Encapsulation**
  - $r_1, r_2 \leftarrow S_w^n(F_2),\ e \leftarrow_{\$} S_w^n(F_2)$
  - $u \leftarrow r_1 + hr_2,\ v \leftarrow mG + sr_2 + e$
- **Decapsulation**
  - $m \leftarrow$ Decode $(v - uy)$ with the error HW$(xr_2 - r_1 y + e\ ) \leq \delta$

HQC

- **Important features:**
  - IND-CPA code-based PKE
  - Reduction to a well-known and difficult problem:
    - ✓Decoding random quasi-cyclic codes
  - No hidden trap in the code
  - Efficient decoding
  - Precise DFR analysis
    - ✓Decrease the size of our public keys

	HQC128	HQC192	HQC256
Public Key (bytes)	2,249	4,522	7,245
Private Key (bytes)	4,481	9,026	14,469
KeyGen (cycles)	83k	200k	400k
Encapsulation (cycles)	197k	456k	887k
Decapsulation (cycles)	349l	740k	1478k

감사합니다

Professor Young-Sik Kim

Department of Inform. & Commun. Eng.,
IT Building, Room 8111, Chosun University

E-mail: iamyskim@chosun.ac.kr,
mypurist@gmail.com
TEL +82-62-230-7032
Mobile: +82-10-3251-1418
https://sites.google.com/site/mypurist/