QC-MDPC KEM

Philip Lafrance

ISARA Corporation
<philip.lafrance@isara.com>

April 13, 2018

Background

What is it?

Background

What is it?

- Encryption-based Key Encapsulation Mechanism:

Background

What is it?

- Encryption-based Key Encapsulation Mechanism:
  - Takes as input a public key and a secret seed.

Background

What is it?

- Encryption-based Key Encapsulation Mechanism:
    - Takes as input a public key and a secret seed.
    - Derives and “encapsulates” an *ephemeral* symmetric key K.

Background

What is it?

- Encryption-based Key Encapsulation Mechanism:
  - Takes as input a public key and a secret seed.
  - Derives and “encapsulates” an *ephemeral* symmetric key K.
  - K can be recovered from the ciphertext by using the secret key matching the public key used above.

Background

What is it based on?

Background

What is it based on?

- McEliece Encryption Scheme:

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.

Background

What is it based on?

- McEliece Encryption Scheme:
    - Encrypted messages are of the form: $mG \oplus e$,
    - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.
    - n - codeword length

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.
  - n - codeword length
  - 2^k - cardinality of the code family

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.
  - n - codeword length
  - 2^k - cardinality of the code family
  - k, and $r = k = n/2$ - *dimension* and *co-dimension*

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.
  - n - codeword length
  - 2^k - cardinality of the code family
  - k, and $r = k = n/2$ - *dimension* and *co-dimension*
  - $w \in \mathcal{O}(\sqrt{n \log(n)})$ - *weight* of the rows of the *parity-check matrix H*

Background

What is it based on?

- McEliece Encryption Scheme:
  - Encrypted messages are of the form: $mG \oplus e$,
  - where, m is the message, e is an error vector, and G is the public-key.
- Using Quasi-Cyclic Moderate Density Parity Check codes.
  - n - codeword length
  - 2^k - cardinality of the code family
  - k, and $r = k = n/2$ - *dimension* and *co-dimension*
  - $w \in \mathcal{O}(\sqrt{n \log(n)})$ - *weight* of the rows of the *parity-check matrix H*
  - t - the error-correction threshold

Key Generation

Algorithm 1 QCMDPC.KeyGen

Input: Security parameter $n = 2r$, weight w, and co-dimension r.
Output: Public key G, secret key H.

1: Select $h_0, h_1 \xleftarrow{\$} \{0,1\}^r$, each of odd weight $w/2$.
2: Compute $H_0, H_1 \in \mathbb{F}_2^{r \times r}$ by right circular shifts of h_0 and h_1.
3: Set $H = [H_0 | H_1] \in \mathbb{F}_2^{r \times n}$.
4: Calculate $Q = (H_1^{-1} H_0)^T$
5: Set $G = [I_k | Q]$.
6: **return** (G, H).

Decoding Algorithms

Recall that an encrypted message is of the form:

$$c = mG \oplus e.$$

Decoding Algorithms

Recall that an encrypted message is of the form:

$$c = mG \oplus e.$$

- To recover m, a *decoding algorithm* is required.

Decoding Algorithms

Recall that an encrypted message is of the form:

$$c = mG \oplus e.$$

- To recover m, a *decoding algorithm* is required.
- The choice of decoder does not affect interoperability/functionality.

Decoding Algorithms

Recall that an encrypted message is of the form:

$$c = mG \oplus e.$$

- To recover m, a *decoding algorithm* is required.
- The choice of decoder does not affect interoperability/functionality.
- However, for security reasons, the decoding algorithm must be constant time, and preferably with as low of a decoding failure rate (DFR) as possible.

Encapsulation

We require an error vector derivation function, as well as two key derivation functions.

Encapsulation

We require an error vector derivation function, as well as two key derivation functions.

- $\nu : \{0,1\}^* \to \{0,1\}^n$ – an efficient, deterministic, pseudorandom, one-way function with weight t outputs.

Encapsulation

We require an error vector derivation function, as well as two key derivation functions.

- $\nu : \{0,1\}^* \to \{0,1\}^n$ – an efficient, deterministic, pseudorandom, one-way function with weight t outputs.
- $\mathsf{KDF}_1 : \{0,1\}^* \to \{0,1\}^k$, and

Encapsulation

We require an error vector derivation function, as well as two key derivation functions.

- $\nu : \{0,1\}^* \to \{0,1\}^n$ – an efficient, deterministic, pseudorandom, one-way function with weight t outputs.
- $\mathsf{KDF}_1 : \{0,1\}^* \to \{0,1\}^k$, and
- $\mathsf{KDF}_2 : \{0,1\}^* \to \{0,1\}^{256+\ell}$ – where ℓ is the desired key length.

Encapsulation

Algorithm 2 QCMDPC.Encap

Input: Public key G, and random seed $s \in \mathbb{F}_2^k$.
Output: Symmetric key $K \in \{0,1\}^m$
Output: Ciphertext $C = (C_1, C_2) \in \mathbb{F}_2^{256} \times \mathbb{F}_2^{\ell}$.

$e \quad \nu(s)$ ▷ Compute n-bit error vector
$y \quad \mathrm{KDF}_1(e)$ ▷ Compute k-bit masking value
$x \quad s \oplus y$ ▷ Obtain k-bit plain text
$C_1 \quad xG \oplus e$ ▷ Encrypt x with e
$C_2 || K \quad \mathrm{KDF}_2(s)$
return $(K,\ C = (C_1, C_2))$

Decapsulation

Algorithm 3 QCMDPC.Decap

Input: Secret key H, ciphertext $(C_1, C_2) \in \mathbb{F}_2^{256} \times \mathbb{F}_2^{\ell}$, and dimension k.

Output: Symmetric key $K \in \{0,1\}^{\ell}$ or a decapsulation failure $\perp$.

$((x, e), d_{\text{err}})$ QCMDPC.Decrypt(H, C_1).
y KDF$_1(e)$
s $x \oplus y$
e' $\nu(s)$.
$C_2' || K$ KDF$_2(s)$.
if $e' = e$ **and** $C_2' = C_2$ **and** $d_{\text{err}} =$ False **then**
    return K
else
    return $\perp \leftarrow$
end if

What attacks were considered?

Security

What attacks were considered?

- State-of-the-art key distinguishing, key recovery, and decoding attacks.

Security

What attacks were considered?

- State-of-the-art key distinguishing, key recovery, and decoding attacks.
  - ISD,

Security

What attacks were considered?

- State-of-the-art key distinguishing, key recovery, and decoding attacks.
  - ISD,
  - Prange + Grover, MMT + Quantum Walks (QISD),

Security

What attacks were considered?

- State-of-the-art key distinguishing, key recovery, and decoding attacks.
  - ISD,
  - Prange + Grover, MMT + Quantum Walks (QISD),
- GJS

Security

What attacks were considered?

- State-of-the-art key distinguishing, key recovery, and decoding attacks.
  - ISD,
  - Prange + Grover, MMT + Quantum Walks (QISD),
- GJS
- IND-CPA reduction

Parameters

Security					
Classical	Quantum	n	r	w	t
80	58	9602	4801	90	84
128	86	19714	9857	142	134
256	154	65542	32771	274	264

Table: Parameter sets for classical and quantum security[1].

[1]Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece Variants from Moderate Density Parity-Check Codes, 2012. Cryptology ePrint Archive, Report 2012/409.

Parameters

Using the $(65542, 32771, 274, 264)$ parameter set:

Security				
Classical	Quantum	Public key	Private Key	Ciphertext
256	154	4097	548	8226

Table: Data sizes in bytes.

Thank You.