A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors

Qian Guo Thomas Johansson Paul Stankovski

Dept. of Electrical and Information Technology, Lund University

Outline

Outline

Motivation

- Quantum computers break cryptosystems based on the hardness of factoring and discrete log—e.g., RSA, ECC.
- Post-quantum candidates: lattice-based, code-based, hash-based, multivariate crypto.

Motivation

- Quantum computers break cryptosystems based on the hardness of factoring and discrete log—e.g., RSA, ECC.
- Post-quantum candidates: lattice-based, code-based, hash-based, multivariate crypto.
- Code-based cryptosystems—e.g., McEliece using Goppa codes [McEliece 1978].
- Main drawback: large key-size.

Motivation

- Quantum computers break cryptosystems based on the hardness of factoring and discrete log—e.g., RSA, ECC.
- Post-quantum candidates: lattice-based, code-based, hash-based, multivariate crypto.
- Code-based cryptosystems—e.g., McEliece using Goppa codes [McEliece 1978].
- Main drawback: large key-size.
- An important variant: QC-MDPC [Misoczki, Tillich, Sendrier, Barreto 2013].
  - Much smaller key-size: 4801 bits for 80-bit security.
  - good security arguments (very little structure).
  - easy implementation (including lightweight implementation) [Heyse, von Maurich, Güneysu, 2013].
  - A scheme recommended for further study.

Motivation

- Quantum computers break cryptosystems based on the hardness of factoring and discrete log—e.g., RSA, ECC.
- Post-quantum candidates: lattice-based, code-based, hash-based, multivariate crypto.
- Code-based cryptosystems—e.g., McEliece using Goppa codes [McEliece 1978].
- Main drawback: large key-size.
- An important variant: QC-MDPC [Misoczki, Tillich, Sendrier, Barreto 2013].
  - Much smaller key-size: 4801 bits for 80-bit security.
  - good security arguments (very little structure).
  - easy implementation (including lightweight implementation) [Heyse, von Maurich, Güneysu, 2013].
  - A scheme recommended for further study.
- Our goal: to recover the secret key

Outline

1. Motivation
2. **Background on QC-MDPC**
3. The New Idea Using Decoding Errors
   - Key-Recovery from Distance Spectrum (DS)
   - On Plain QC-MDPC (CPA)
   - On the CCA-Secure Version
   - An Intuitive Explanation
4. Results
5. Discussions and Conclusions

QC-MDPC Codes

Quasi-cyclic Codes

Suppose $n = n_0 r$. An $[n, n-r]$-linear code $\mathcal{C}$ over $\mathbb{F}_2$ is quasi-cyclic if every cyclic shift of a codeword by n_0 steps remains a codeword.

We assume that $n_0 = 2$ throughout the remaining slides.

- For convenience, we write

$$\mathbf{H} = [\mathbf{H_0} | \mathbf{H}_1],$$

$$\mathbf{G} = [\mathbf{I} | \mathbf{P}] = \left[\mathbf{I} | (\mathbf{H}_1^{-1} \mathbf{H}_0)^T \right].$$

  where $\mathbf{H_i}$ are circulant matrices (defined by its first row).
- Operations can be viewed in the polynomial ring $\mathbb{F}_2[x]/\langle x^r - 1 \rangle$.

$$h_0(x), h_1(x), p(x) = h_0(x)/h_1(x), \ldots$$

- The polynomial $h_0(x)$ can also be represented by a vector $\mathbf{h}_0$.

QC-MDPC Codes

LDPC/MDPC Codes

A Low Density Parity-Check Code (LDPC) is a linear code admitting a sparse parity-check matrix, while a Moderate Density Parity-Check Code (MDPC) is a linear code with a denser but still sparse parity-check matrix.

- LDPC codes are with small constant row weights.
- MDPC codes with row weights scale in $O(\sqrt{n \log n})$.

QC-MDPC Codes

A QC-MDPC code is a quasi-cyclic MDPC code with row weight $\hat{w}$.

The QC-MDPC PKC Scheme

- KeyGen():
  - Generate a parity-check matrix $\mathbf{H} = [\mathbf{H_0}|\mathbf{H_1}]$ for a binary QC-MDPC code with row weight $\hat{w}$.
  - Derive the systematic generator matrix $\mathbf{G} = [\mathbf{I}|\mathbf{P}]$, where $\mathbf{P} = (\mathbf{H}_1^{-1}\mathbf{H}_0)^T$.
  - The public key: $\mathbf{G}$. The private key: $\mathbf{H}$.
- $\text{Enc}_{\mathbf{G}}(\mathbf{m})$:
  - Generate a random error vector $\mathbf{e}$ with weight t.
  - The ciphertext is $\mathbf{c} = \mathbf{mG} + \mathbf{e}$.
- $\text{Dec}_{\mathbf{H}}(\mathbf{c})$:
  - Compute the syndrome vector $\mathbf{s} = \mathbf{cH^T} = \mathbf{eH^T}$, and then use an iterative decoder to extract the noise $\mathbf{e}$.
  - Recover the plaintext $\mathbf{m}$ from the first k entries of $\mathbf{mG}$.

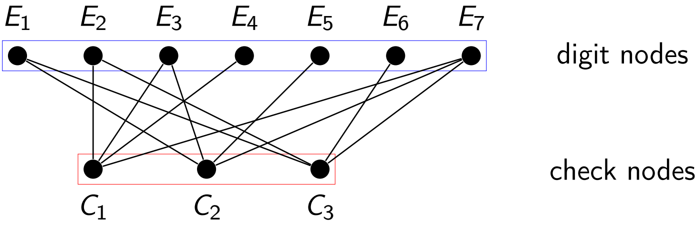
CCA-Secure Version

- Extending the security model beyond CPA:
  - Resend attacks, reaction attacks, chosen ciphertext attacks,...
- To cope with CCA, one can use a CCA conversion, e.g., the one suggested by Kobara, Imai in 2001.
  - The CCA conversion makes the choice of error vector $\mathbf{e}$ "random".

Suggested parameters for 80-bit security:

$$n = 9602, k = r = 4801, \hat{w} = 90, t = 84 \quad \text{public key: 4801 bits}$$

Iterative Decoding: Gallager's Bit-Flipping Strategy



$$\mathbf{c}\mathbf{H}^{\mathbf{T}} = (\mathbf{v}+\mathbf{e})\mathbf{H}^{\mathbf{T}} = \mathbf{e}\mathbf{H}^{\mathbf{T}} = \mathbf{s}$$

- Start with Tanner graph for **H**, initial syndrome **s** and set digit nodes to zero. Add a counter to each digit node.
- For the t^{th} iteration:
  - Run through all parity-check equations and for every digit node connected to an unsatisfied check node, increase its corresponding counter by one.
  - Run through all digit nodes and flip its value if its counter satisfies a certain constraint, e.g., the counter surpasses a threshold.

Outline

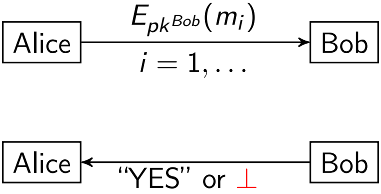
Basic Scenario

$$\boxed{\text{Alice}} \xrightarrow[i = 1, \ldots]{E_{pk^{Bob}}(m_i)} \boxed{\text{Bob}}$$

$$\boxed{\text{Alice}} \xleftarrow[\text{“YES” or } \perp]{} \boxed{\text{Bob}}$$

- In terms of a security model definition, the attack is called a *reaction attack*.
- A weaker model than CCA (a stronger attack).
- Resend and reaction attacks on McEliece PKC have appeared before. However, they have only targeted message recovery.
- Key recovery: to recover $\mathbf{h}_0$.

Basic Scenario



- In terms of a security model definition, the attack is called a *reaction attack*.
- A weaker model than CCA (a stronger attack).
- Resend and reaction attacks on McEliece PKC have appeared before. However, they have only targeted message recovery.
- Key recovery: to recover $\mathbf{h}_0$.
- Show: Decoding error probabilities for different error patterns $\Rightarrow$ the private key $\mathbf{h}_0$.

Key-Related Property: Distance Spectrum (DS)

Distance Spectrum (DS)

The distance spectrum for $\mathbf{h}_0$, denoted $D(\mathbf{h}_0)$, is given as

$$D(\mathbf{h}_0) = \{d : 1 \leq d \leq \lfloor \frac{r}{2} \rfloor, \exists \text{ a pair of ones with distance } d \text{ in } cyc(\mathbf{h}_0)\}.$$

Here $cyc(\mathbf{h}_0)$ includes all cyclic shifts of $\mathbf{h}_0$. Since a distance d can appear many times in $\mathbf{h}_0$, we introduce the multiplicity $\mu(d)$.

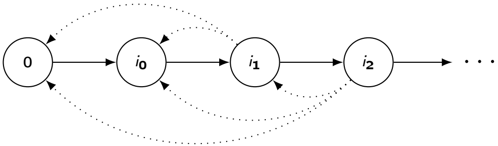
As an example, for the bit pattern $\mathbf{c} = 0011001$ we have $r = 7$ and $1 \leq d \leq 3$. Thus,

$$D(\mathbf{c}) = \{1, 3\},$$

with distance multiplicities $\mu(1) = 1, \mu(2) = 0$ and $\mu(3) = 2$.

- $D(\mathbf{h}_0) \Rightarrow$ the private key $\mathbf{h}_0$.

Reconstruction of $\mathbf{h}_0$ from DS



Assuming $D(\mathbf{h}_0)$ is known, we can reconstruct $\mathbf{h}_0$.

- Start by assigning the first two ones in a length i_0 vector in position 0 and i_0, where i_0 is the smallest value in $D(\mathbf{h}_0)$.
- Put the third one in a position and test if the two distances between this third one and the previous two ones both appear in the distance spectrum. If they do not, we test the next position for the third bit.
- If they do, we move to test the fourth bit and its distances to the previous three ones, etc.

In expectation, it is efficient.

Main Observation

The Problem

Decoding error probabilities for different error patterns $\Rightarrow D(\mathbf{h}_0)$?

Main Observation

The Problem

Decoding error probabilities for different error patterns $\Rightarrow D(\mathbf{h}_0)$?

Main Observation

For a distance d, consider the error patterns with at least one pair of ones at distance d. Then, the decoding error probability when $d \in D(h_0)$ is smaller than that if $d \notin D(h_0)$.

On Plain QC-MDPC (CPA)

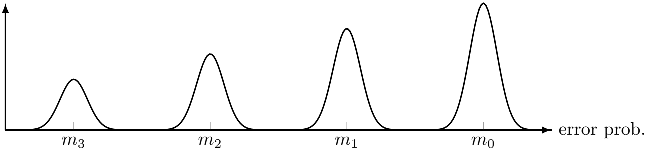
- Ψ_d is the set of all binary vectors of length $n = 2r$ having exactly t ones, where all the t ones are placed as pairs with distance d in the first half of the vector.

$$\mathbf{e} = (00\cdots01\underbrace{00\cdots0}_{d-1}100\cdots01\underbrace{00\cdots0}_{d-1}100\cdots0, 00\cdots0)$$

Attack

- Alice will send messages to Bob, with error selected from Ψ_d.
- When there is a decoding error with Bob, she will record this and after M messages she will be able to compute an empirical decoding error probability for the subset Ψ_d.
- Alice will repeat for $d = 1, 2, \ldots, U$.

How to Decide Multiplicity $\mu(d)$



(a)

(b)

Figure: Classification of distance multiplicities based on decoding error probability. (a): Distribution shape in general. (b): Empirical distribution using $M = 100,000$ decoding trials for each distance (proposed parameters for 80-bit security with $t = 84$).

Computing DS

Input: parameters n, r, w and t of the underlying QC-MDPC scheme, $M =$ trials per distance.
Output: distance spectrum $D(\mathbf{h}_0)$.

For all distances d

- Try M decoding trials using the designed error pattern
- Perform statistical test to decide multiplicity $\mu(d)$
- If $\mu(d) \neq 0$, add d with multiplicity $\mu(d)$ to distance spectrum $D(\mathbf{h}_0)$

The complexity is $O(M \cdot U)$.

Attack on CCA-Secure QC-MDPC

We can no longer control the error.

- Form different subsets with desired error patterns.
  - For a distance d, error patterns that contain at least one occurrence of distance d between error bits are chosen.
- These subsets can still be used to efficiently distinguish whether a certain distance d appears in the distance spectrum of $\mathbf{h_0}$.

Attack in CCA case

Input: a collection of T ciphertexts (denoted Σ).
Output: distance spectrum $D(\mathbf{h}_0)$.

Record decryptability for each $c \in \Sigma$
$\mathbf{s} \leftarrow$ storage for distance spectrum of secret key
For all distances d

  $\Sigma_d \leftarrow \{c \in \Sigma \mid \mu_c(d) \geq 1\}$

  $\mathbf{s}[d] \leftarrow$ multiplicity classification from decryptability rate in Σ_d

Return $\mathbf{s}$

$\mu_c(d)$ is the number of pairs of ones with distance d in the error vector for ciphertext c.

The complexity is $O(T \cdot \frac{r}{2})$.

An Explanation for the Distinguishing Procedure

Error patterns are from Ψ_d. Let $w = wt(\mathbf{h}_0)$.

- The first iteration plays a vital role in the decoding process
- jth parity check : $\sum_{i=0}^{n-1} h_{ij} e_i = s_j$
- If we look at all the r parity checks in $\mathbf{H}$, we will create a total of exactly $t \cdot w$ nonzero terms $h_{ij} e_i$ in the parity checks all together.
- Putting $t \cdot w$ different objects in r buckets and counting the number of objects in each bucket. An even number of objects in a bucket will be helpful in decoding; an odd number of objects will act in opposite.

Table: The relation between the number of nonzero $h_{ij} e_i$'s and that of correctly changed counters in the first decoding iteration.

# $(h_{ij} e_i = 1)$	#(right change)	#(wrong change)
0	w	0
1	1	$w - 1$
2	$w - 2$	2
3	3	$w - 3$
$\vdots$	$\vdots$	$\vdots$

An Explanation for the Distinguishing Procedure

- If $\mathbf{h}_0$ contains two ones with distance d inbetween (CASE-1), we have "artificially" created cases where we know that we have at least two nonzero terms $h_{ij}e_i$ in the parity check.
- This "artificial" creation of pairs of nonzero terms $h_{ij}e_i$ in the same check equation changes the distribution of the number of nonzero terms $h_{ij}e_i$ in parity checks.

Table: The distinct distributions of the number of nonzero terms $h_{ij}e_i$'s for the error patterns from Ψ_d using the QC-MDPC parameters for 80-bit security and assuming that the weight of $\mathbf{h}_0$ is exactly 45.

$\#\ (h_{ij}e_i = 1)$	Probability	
	CASE-0	CASE-1
0	0.4485	0.4534 $\uparrow$
1	0.3663	0.3602 $\downarrow$
≥ 2	0.1852	0.1864

Outline

1. Motivation
2. Background on QC-MDPC
3. The New Idea Using Decoding Errors
   - Key-Recovery from Distance Spectrum (DS)
   - On Plain QC-MDPC (CPA)
   - On the CCA-Secure Version
   - An Intuitive Explanation
4. **Results**
5. Discussions and Conclusions

Results—Reconstruction of $\mathbf{h}_0$ from DS

80 bit security: $n = 9602, k = r = 4801, \hat{w} = 90, t = 84$ with (simplest) Gallager bit-flipping

Reconstruction of $\mathbf{h}_0$ from the DS:

- It takes in expectation 2^{35} operations.
- It can be slow in the worst-case.

In practice:

- We perform 3000 trials using a single core of a personal computer.
- The implementation is unoptimised.
- It takes 144 seconds on average.
- The worst case: 49 minutes.

Results—Obtaining DS in the CPA Case

80 bit security: $n = 9602, k = r = 4801, \hat{w} = 90, t = 84$ with (simplest) Gallager bit-flipping

Table: Decoding error rates when using the original Gallager's bit-flipping algorithm and the designed error pattern Ψ_d with $t = 84$ and $t = 90$. The number of decoding trials in a group is $M = 100,000$ and $M = 10,000$, respectively.

	$t = 84$		$t = 90$	
multiplicity	error rate	σ	error rate	σ
0	0.0044099	0.00003868	0.415395	0.000830
1	0.0009116	0.00001304	0.248642	0.000729
2	0.0001418	0.00000475	0.121623	0.000529
3	0.0000134	0.00000112	0.048330	0.000299

$U = 2400$. The complexity of determining the DS for $t = 84$ (or $t = 90$) is 2^{28} (or 2^{25}).

Results—Obtaining DS in the CCA Case

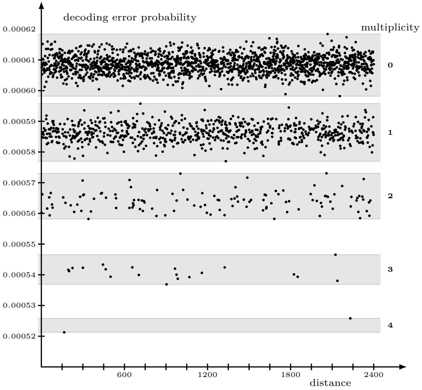


Figure: Classification intervals for the $t = 84$ worst-case simulation after 356M ciphertexts. All 2400 data points plotted.

The complexity is less than 2^{40} for the proposed security parameters for 80-bit security using the Gallager's original bit-flipping decoder.

Outline

Discussions: Using Other Decoders

- In implementation the original Gallager's bit-flipping algorithm is employed (error rate 5×10^{-4}).
- The state-of-the-art variants can improve upon it with a factor of $2^{15.6}$ (error rate 10^{-8}).
- Reasonable guess: the attack time when using one of these better decoders is the complexity when using the original one $\times 2^{15.6}$. That is 2^{44} (or 2^{55}) for the CPA (or CCA) case when using the suggested parameters for 80-bit security.

Final Remarks

- A reaction-type key-recovery attack against QC-MDPC has been presented.
- This attack can break the CCA-secure version using the suggested parameters.
- Countermeasure: make the decoding error probability small, like 2^{-80} for 80-bit security.
- The attack may still be applicable in e.g. side-channel attacks.

Thank you for your attention!

Questions?

Table

Table: The relation between the number of nonzero $h_{ij}e_i$'s and that of correctly changed counters in the first decoding iteration.

# $(h_{ij}e_i = 1)$	s_j	$\hat{s}_j$	#(right change)	#(wrong change)
0	0	0	w	0
1	1	0	1	$w-1$
2	0	0	$w-2$	2
3	1	0	3	$w-3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$