HQC

NIST Postquantum Cryptography Project

Carl Miller

April 3, 2018

The Basics

- It's a public key encryption (and key encapsulation) scheme.

- HQC = "Hamming Quasi-Cyclic." The scheme is based on the hardness of decoding quasi-cyclic codes. It's proved to be IND-CPA.

Quasi-Cyclic codes

Translation-invariant codes

A linear code $V \subseteq \mathbf{F}_2^n$ is **cyclic** if it is stabilized by the translation operator $T\colon (v_1, \ldots, v_n) \mapsto (v_2, \ldots, v_n, v_1)$.

Translation-invariant codes

Equivalently,

A cyclic linear code in $\mathbf{F}_2^n$ is an ideal of the ring $\mathbf{F}_2[X]/(X^N - 1)$.

f

Translation-invariant codes

The following case (s = 2) is important.

Let $\mathcal{R} := \mathbf{F}_2[X]/(X^N - 1)$, $h \in \mathcal{R}$. Then, let $\mathcal{C}$ be the kernel of the map $\mathcal{R}^2 \to \mathcal{R}$ given by

$$(x, y) \mapsto x + hy.$$

The 2-DQCSD Problem

DQCSD = Decision Quasi-Cyclic code Syndrome Decoding

Fix $w \leq n$. Suppose that a random oracle chooses:

- random $h \in \mathcal{R}$,
- random $x, y \in \mathcal{R}$ each having w monomial terms,

and outputs $(h, x + yh)$.

The PKE Protocol

Public-Key Encryption with QC codes

k = message length, n = output length parameter.

Alice fixes a (known) efficiently decodable [n,k] code with generator matrix **G**. (**G** is the BCH code tensored with the repetition code?)

Alice chooses a random degree <= n polynomial **h** (i.e., a random QC code of index 2).

Public key

Encrypted message

G, h

G, h

Public-Key Encryption with QC codes

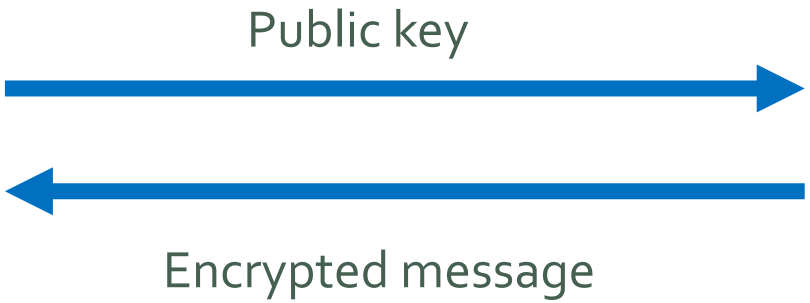
Encrypted message will be encoded with both **G** and **h**, but to Alice it will appear to have only been encoded with G.

Public-Key Encryption with QC codes

1. Alice chooses two low-Hamming weight polynomials **x,y** of degree n.
2. Alice sends **s** := **x** + **hy** to Bob. (Public key.)

   (All arithmetic is mod 2 and mod (X^n – 1).)

G, h, x, y



G, h, x + hy

Public-Key Encryption with QC codes

3. Let **m** = Bob's message (k bits). Bob computes m**G**.

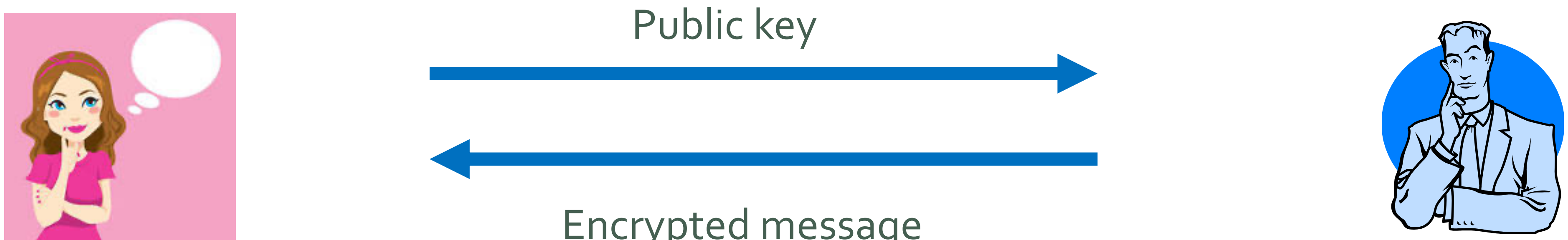
4. Bob computes low-Hamming weight **e, $\mathbf{r_1}$, $\mathbf{r_2}$,** and sends

$$\mathbf{u} := \mathbf{r_1} + \mathbf{h}\mathbf{r_2} \qquad \text{and} \qquad \mathbf{v} := \mathbf{mG} + \mathbf{s}\mathbf{r_2} + \mathbf{e}$$

to Alice.

5. Alice computes **v – uy,** which is (**mG** + noise).

6. She decodes **m**.



G, h, x, y

G, h, x + hy

Public-Key Encryption with QC codes

Critical observation: What remains after step 5 is

$$(mG) + e'$$

where

$$e' = [x\, r_1 + r_2\, y + e\,].$$

All of terms on the right are low Hamming weight, so **e′** is low Hamming weight.

Public key

Encrypted message

G, h, x, y

G, h, x + hy

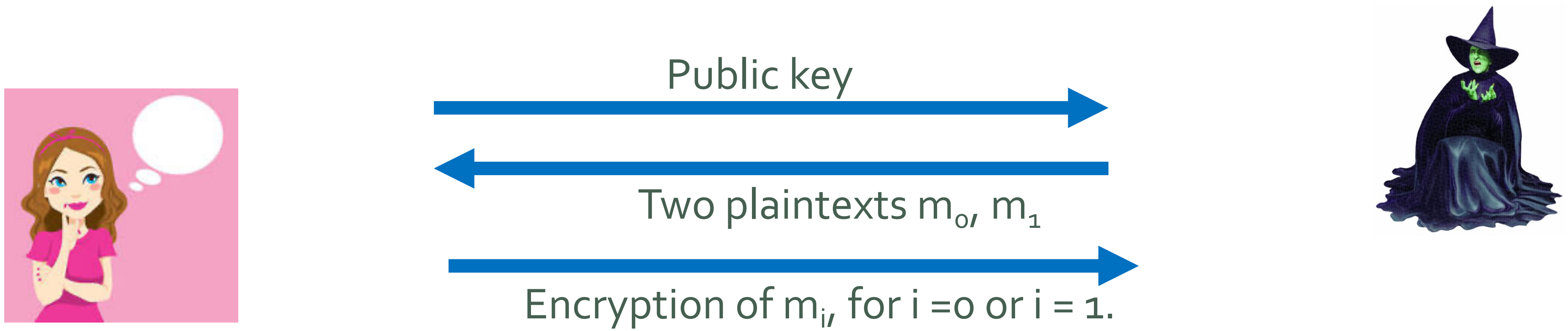
Security Proof (Sketch)

IND-CPA Guessing Game

Suppose that the adversary has an algorithm that successfully guess i.

The user changes the game by instead choosing various data (**s, $\mathbf{r_1}$, $\mathbf{r_2}$, e**) completely at random.

By hardness assumption, the adversary can't tell the difference.



IND-CPA Guessing Game

Lastly, the user changes her choice of i. Now (because of random choices) the adversary can’t tell that this change was made.

Contradiction.

Based on hardness of 2-DQCSD and 3-DQCSD.

Public key

Two plaintexts m_0, m_1

Encryption of m_i, for i =0 or i = 1.

Numerics

Parameters

Encryption size parameter

Message size

Hamming weight parameters

Instance	n_1	n_2	n	k	δ	w	$w_\mathbf{r}$	$w_\mathbf{e}$	security	p_{fail}
Basic-I	766	29	22,229	256	57	67	77	77	128	$< 2^{-64}$
Basic-II	766	31	23,747	256	57	67	77	77	128	$< 2^{-96}$
Basic-III	796	31	24,677	256	57	67	77	77	128	$< 2^{-128}$
Advanced-I	796	51	40,597	256	60	101	117	117	192	$< 2^{-64}$
Advanced-II	766	57	43,669	256	57	101	117	117	192	$< 2^{-128}$
Advanced-III	766	61	46,747	256	57	101	117	117	192	$< 2^{-192}$
Paranoiac-I	766	77	59,011	256	57	133	153	153	256	$< 2^{-64}$
Paranoiac-II	766	83	63,587	256	57	133	153	153	256	$< 2^{-128}$
Paranoiac-III	796	85	67,699	256	60	133	153	153	256	$< 2^{-192}$
Paranoiac-IV	796	89	70,853	256	60	133	153	153	256	$< 2^{-256}$

Decryption Failure probability

Performance

Instance	KeyGen	Encrypt	Decrypt
Basic-I	1.12	1.59	0.71
Basic-II	1.21	1.74	0.77
Basic-III	1.26	1.79	0.79
Advanced-I	2.43	4.14	1.59
Advanced-II	2.58	4.49	1.69
Advanced-III	2.82	4.94	1.83
Paranoiac-I	4.24	7.87	3.02
Paranoiac-II	4.52	8.39	3.22
Paranoiac-III	4.76	8.87	3.40
Paranoiac-IV	5.07	9.42	3.61

Table 2: Timings (in ms) of the reference implementation for different instances of HQC.

Advantages & Limitations

- Reduction to well-studied problem (syndrome decoding).
- Simple protocol.
- Encrypted messages are long.

HQC

NIST Postquantum Cryptography Project

Carl Miller

April 3, 2018