INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

LABORATOIRE D'INFORMATIQUE FONDAMENTALE D'ORLEANS

A new framework for code-based encryption and more

J.-C. Deneuville
<jean-christophe.deneuville@insa-cvl.fr>

April the 5th, 2018
Code-based Cryptography Workshop
Fort Lauderdale

Joint work with:

C. Aguilar Melchor — IRIT Toulouse
O. Blazy, P. Gaborit — XLIM-DMI Limoges
G. Zémor — University of Bordeaux

Motivations

[ME78]

Motivations

[ME78] $\longrightarrow$ [Nie86]

Motivations

$$[\text{ME78}] \longrightarrow [\text{Nie86}] \longrightarrow \left.\begin{array}{l}\text{RS}\\ \text{BCH}\\ \text{Goppa}\\ \text{RM}\end{array}\right\} \begin{array}{c}\text{80's}\\ \downarrow\\ \text{00's}\end{array}$$

Motivations

Key Sizes

[ME78] $\longrightarrow$	[Nie86] $\longrightarrow$	RS BCH Goppa RM	80's $\downarrow$ 00's

Security reduction to a standard problem (random codes)

Motivations

Key Sizes

[ME78] $\longrightarrow$	[Nie86] $\longrightarrow$	RS BCH Goppa RM	80's $\downarrow$ 00's	$\longrightarrow$	Other variations Most of them broken

Security proof

Motivations

Key Sizes

Rank
Metric

[Gab91]

[ME78] → [Nie86] → RS, BCH, Goppa, RM } 80's ↓ 00's → Other variations
Most of them broken

Security proof

Motivations

Key Sizes

Rank
Metric

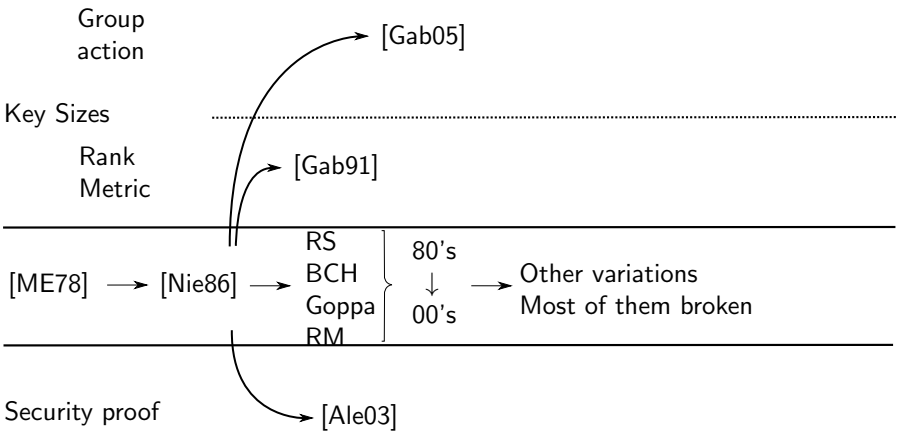
[Gab91]

[ME78] → [Nie86] → RS, BCH, Goppa, RM } 80's ↓ 00's → Other variations
Most of them broken

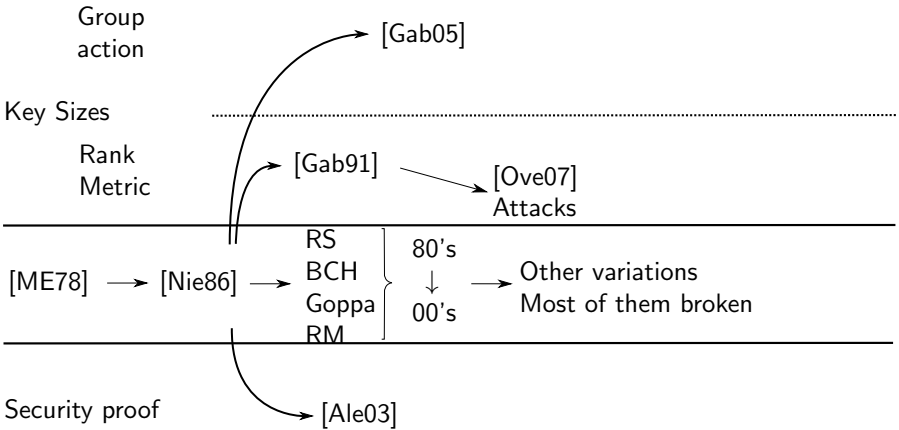
Security proof

[Ale03]

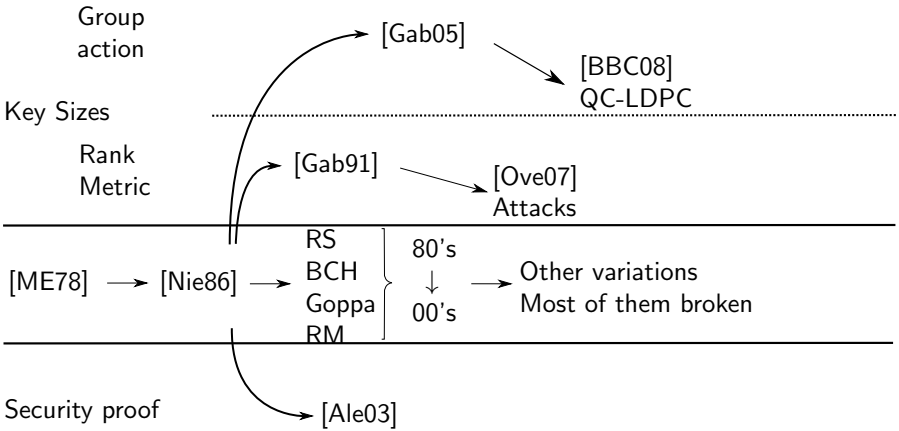
Motivations



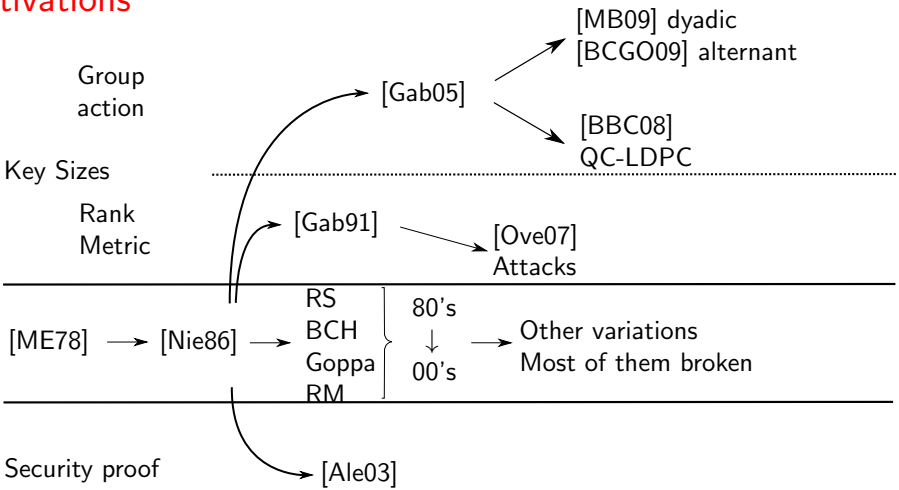
Motivations



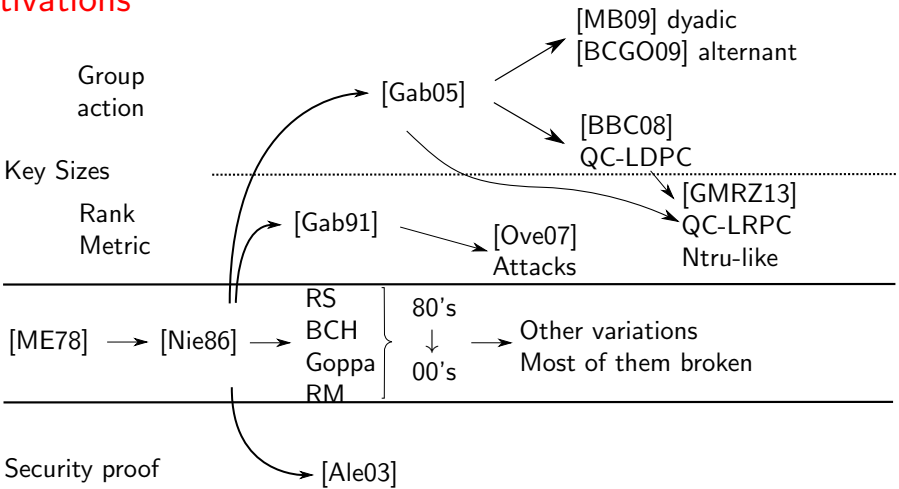
Motivations



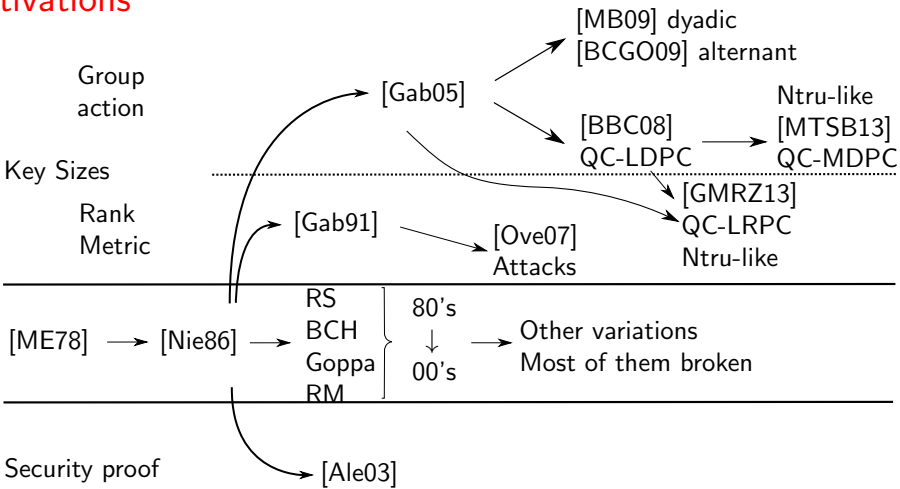
Motivations



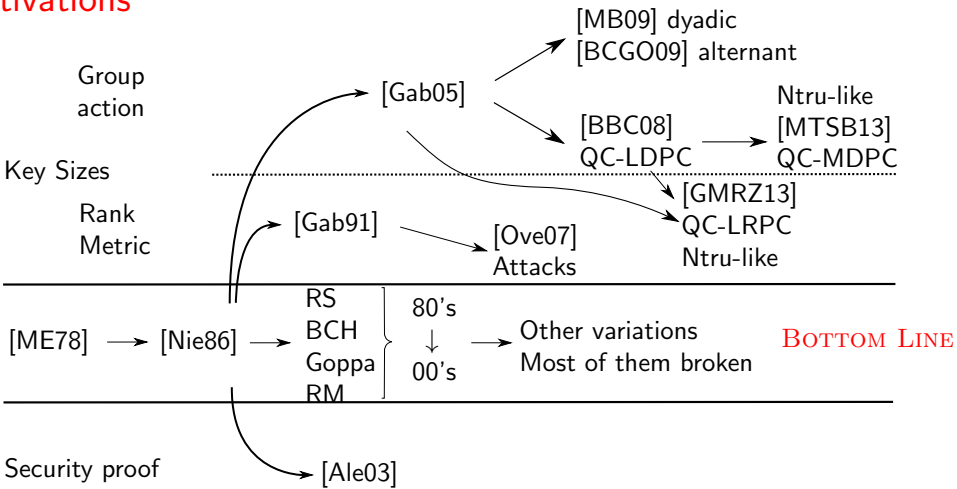
Motivations



Motivations



Motivations



Motivations

Group action

Key Sizes

Rank Metric

Security proof

[ME78] → [Nie86] → RS, BCH, Goppa, RM

80's ↓ 00's → Other variations, Most of them broken

[Nie86] → [Ale03]

[Gab91] → [Ove07] Attacks

[Gab05] → [MB09] dyadic, [BCGO09] alternant

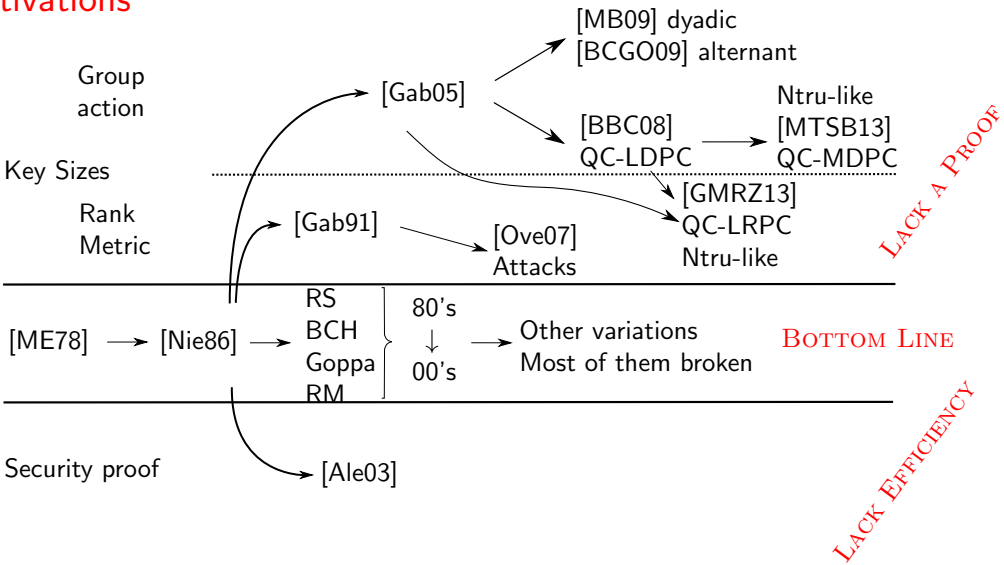
[Gab05] → [BBC08] QC-LDPC → [MTSB13] Ntru-like QC-MDPC

[Gab05] → [GMRZ13] QC-LRPC Ntru-like

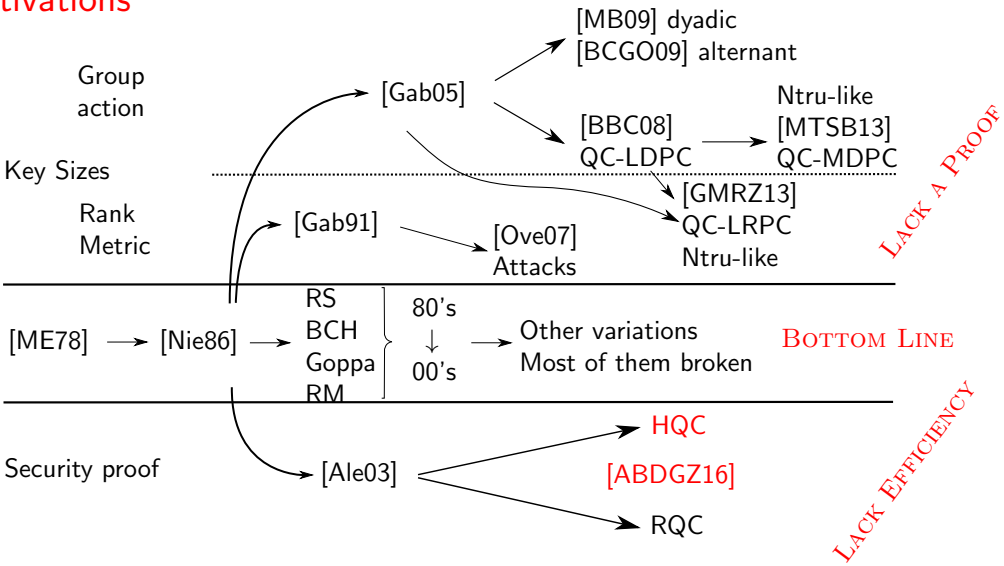
Lack a Proof

Bottom Line

Motivations



Motivations



Outline

Outline

Code-Based Encryption: McEliece

Key Generation :

- $\mathcal{C}[n,k]$ linear code, generated by $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, decoding up to t errors
- $\mathbf{S} \xleftarrow{\$} \mathbb{F}_q^{k\times k}$ invertible, $\mathbf{P} \xleftarrow{\$} \mathbb{F}_2^{n\times n}$ permutation

$$\rightarrow \mathsf{pk} = (\tilde{\mathbf{G}} = \mathbf{SGP}, t),\ \mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P})$$

Encryption (of $\mu \in \mathbb{F}_q^k$) :

- $\mathbf{e} \xleftarrow{\$} \mathbb{F}_q^n$, with $\omega(\mathbf{e}) = t$

$$\rightarrow \mathbf{c} = \mu\tilde{\mathbf{G}} + \mathbf{e}$$

Decryption :

- $\tilde{\mu} = \mathcal{C}.\mathrm{Decode}(\mathbf{cP}^{-1})$

$$\rightarrow \tilde{\mu}\mathbf{S}^{-1}$$

Code-Based Encryption: McEliece

Key Generation :

- $\mathcal{C}[n,k]$ linear code, generated by $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, decoding up to t errors
- $\mathbf{S} \xleftarrow{\$} \mathbb{F}_q^{k\times k}$ invertible, $\mathbf{P} \xleftarrow{\$} \mathbb{F}_2^{n\times n}$ permutation

$$\rightarrow \mathsf{pk} = (\tilde{\mathbf{G}} = \mathbf{SGP}, t),\ \mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P})$$

Encryption (of $\mu \in \mathbb{F}_q^k$) :

- $\mathbf{e} \xleftarrow{\$} \mathbb{F}_q^n$, with $\omega(\mathbf{e}) = t$

$$\rightarrow \mathbf{c} = \mu\tilde{\mathbf{G}} + \mathbf{e}$$

Decryption :

- $\tilde{\mu} = \mathcal{C}.\mathrm{Decode}(\mathbf{cP}^{-1})$

$$\rightarrow \tilde{\mu}\mathbf{S}^{-1}$$

Code-Based Encryption: McEliece

Key Generation :

- $\mathcal{C}[n,k]$ linear code, generated by $\mathbf{G} \in \mathbb{F}_q^{k\times n}$, decoding up to t errors
- $\mathbf{S} \xleftarrow{\$} \mathbb{F}_q^{k\times k}$ invertible, $\mathbf{P} \xleftarrow{\$} \mathbb{F}_2^{n\times n}$ permutation

$$\rightarrow \mathsf{pk} = (\tilde{\mathbf{G}} = \mathbf{SGP}, t),\ \mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P})$$

Encryption (of $\mu \in \mathbb{F}_q^k$) :

- $\mathbf{e} \xleftarrow{\$} \mathbb{F}_q^n$, with $\omega(\mathbf{e}) = t$

$$\rightarrow \mathbf{c} = \mu\tilde{\mathbf{G}} + \mathbf{e}$$

Decryption :

- $\tilde{\mu} = \mathcal{C}.\mathrm{Decode}(\mathbf{cP}^{-1})$

$$\rightarrow \tilde{\mu}\mathbf{S}^{-1}$$

Outline

Alekhnovich first cryptosystem (1/2)

Hypothesis: Decisional Decoding Hypothesis with parameter t

Let $0 < R_1 < R_2 < 1$. Consider k, n such that $R_1 \leq k/n \leq R_2$, $\mathcal{C}$ a random code (generated by $\mathbf{G}$), and a vector which is either:

(i) a uniformly random vector $\mathbf{u}$

(ii) $\mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$ is a unif. rand. codeword and $\mathbf{e}$ a unif. rand. error of weight t, ind. of $\mathbf{c}$.

There is no polynomial-time decoding algorithm $\mathcal{A}$ that decides between (i) and (ii) with a non-negligible advantage over random choice.

Key generation

- $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{k \times n}$, $\mathbf{e} \xleftarrow{\$} \mathcal{S}_t^n(\mathbb{F}_2)$
- $\mathbf{y} \leftarrow \mathbf{x}\mathbf{A} + \mathbf{e}$, for $\mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathrm{pk} = \mathbf{H} \leftarrow \begin{pmatrix} \mathbf{A} \\ \mathbf{y} \end{pmatrix}$
- $\mathbf{G} = \mathbf{H}^{\perp}$ generating $\mathcal{C}$
- $\mathrm{sk} = \mathbf{e}$

Alekhnovich first cryptosystem (1/2)

Hypothesis: Decisional Decoding Hypothesis with parameter t

Let $0 < R_1 < R_2 < 1$. Consider k, n such that $R_1 \leq k/n \leq R_2$, $\mathcal{C}$ a random code (generated by $\mathbf{G}$), and a vector which is either:

(i) a uniformly random vector $\mathbf{u}$

(ii) $\mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}$ is a unif. rand. codeword and $\mathbf{e}$ a unif. rand. error of weight t, ind. of $\mathbf{c}$.

There is no polynomial-time decoding algorithm $\mathcal{A}$ that decides between (i) and (ii) with a non-negligible advantage over random choice.

Key generation

- $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{k \times n}$, $\mathbf{e} \xleftarrow{\$} \mathcal{S}_t^n(\mathbb{F}_2)$
- $\mathbf{y} \leftarrow \mathbf{x}\mathbf{A} + \mathbf{e}$, for $\mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathtt{pk} = \mathbf{H} \leftarrow \begin{pmatrix} \mathbf{A} \\ \mathbf{y} \end{pmatrix}$
- $\mathbf{G} = \mathbf{H}^{\perp}$ generating $\mathcal{C}$
- $\mathtt{sk} = \mathbf{e}$

Alekhnovich first cryptosystem (2/2)

Encryption

- $\mathbf{c}_0 = \mathtt{Enc}_{\mathtt{pk}}(0) = \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathbf{c}_1 = \mathtt{Enc}_{\mathtt{pk}}(1) = \mathbf{c} + \mathbf{t}, (\mathbf{c}, \mathbf{t}) \xleftarrow{\$} \mathcal{C} \times \mathcal{S}_t{}^n(\mathbb{F}_2)$

Decryption

- Return $b = \langle \mathbf{e}, \mathbf{c} \rangle$

$(\langle \mathbf{e}, \mathbf{c}_1 \rangle = \langle \mathbf{e}, \mathbf{c} \rangle + \langle \mathbf{e}, \mathbf{t} \rangle = \langle \mathbf{e}, \mathbf{t} \rangle)$

Security reduction

Alekhnovich showed that a distinguisher between $\mathbf{u}$ and $\mathbf{c} + \mathbf{e}$ yields an algorithm to decode up to t errors.

$\rightarrow$ Can be adapted to larger plaintext spaces, but results in inefficient cryptosystems...

Alekhnovich first cryptosystem (2/2)

Encryption

- $\mathbf{c}_0 = \mathtt{Enc}_{\mathtt{pk}}(0) = \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathbf{c}_1 = \mathtt{Enc}_{\mathtt{pk}}(1) = \mathbf{c} + \mathbf{t}, (\mathbf{c}, \mathbf{t}) \xleftarrow{\$} \mathcal{C} \times \mathcal{S}_t{}^n(\mathbb{F}_2)$

Decryption

- Return $b = \langle \mathbf{e}, \mathbf{c} \rangle$

$(\langle \mathbf{e}, \mathbf{c}_1 \rangle = \langle \mathbf{e}, \mathbf{c} \rangle + \langle \mathbf{e}, \mathbf{t} \rangle = \langle \mathbf{e}, \mathbf{t} \rangle)$

Security reduction

Alekhnovich showed that a distinguisher between $\mathbf{u}$ and $\mathbf{c} + \mathbf{e}$ yields an algorithm to decode up to t errors.

$\rightarrow$ Can be adapted to larger plaintext spaces, but results in inefficient cryptosystems...

Alekhnovich first cryptosystem (2/2)

Encryption

- $\mathbf{c}_0 = \mathtt{Enc}_{\mathtt{pk}}(0) = \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathbf{c}_1 = \mathtt{Enc}_{\mathtt{pk}}(1) = \mathbf{c} + \mathbf{t}, (\mathbf{c}, \mathbf{t}) \xleftarrow{\$} \mathcal{C} \times \mathcal{S}_t{}^n(\mathbb{F}_2)$

Decryption

- Return $b = \langle \mathbf{e}, \mathbf{c} \rangle$

$(\langle \mathbf{e}, \mathbf{c}_1 \rangle = \langle \mathbf{e}, \mathbf{c} \rangle + \langle \mathbf{e}, \mathbf{t} \rangle = \langle \mathbf{e}, \mathbf{t} \rangle)$

Security reduction

Alekhnovich showed that a distinguisher between $\mathbf{u}$ and $\mathbf{c} + \mathbf{e}$ yields an algorithm to decode up to t errors.

$\rightarrow$ Can be adapted to larger plaintext spaces, but results in inefficient cryptosystems...

Alekhnovich first cryptosystem (2/2)

Encryption

- $\mathbf{c}_0 = \mathtt{Enc}_{\mathtt{pk}}(0) = \mathbf{u} \xleftarrow{\$} \mathbb{F}_2^n$
- $\mathbf{c}_1 = \mathtt{Enc}_{\mathtt{pk}}(1) = \mathbf{c} + \mathbf{t}, (\mathbf{c}, \mathbf{t}) \xleftarrow{\$} \mathcal{C} \times \mathcal{S}_t{}^n(\mathbb{F}_2)$

Decryption

- Return $b = \langle \mathbf{e}, \mathbf{c} \rangle$

$(\langle \mathbf{e}, \mathbf{c}_1 \rangle = \langle \mathbf{e}, \mathbf{c} \rangle + \langle \mathbf{e}, \mathbf{t} \rangle = \langle \mathbf{e}, \mathbf{t} \rangle)$

Security reduction

Alekhnovich showed that a distinguisher between $\mathbf{u}$ and $\mathbf{c} + \mathbf{e}$ yields an algorithm to decode up to t errors.

$\rightarrow$ Can be adapted to larger plaintext spaces, but results in inefficient cryptosystems...

Outline

HQC Encryption Scheme [ABD+18]

Encryption scheme in **H**amming metric, using **Q**uasi-**C**yclic Codes

- Notation: Secret data - Public data - One-time Randomness
- **G** is the generator matrix of some public code $\mathcal{C}$.

Alice		Bob
$\mathsf{seed_h} \xleftarrow{\$} \{0,1\}^\lambda$, $\mathbf{h} \xleftarrow{\mathsf{seed_h}} \mathbb{F}_2^n$		
$\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{h}\mathbf{y}$	$\xrightarrow{\mathsf{seed_h}, \mathbf{s}}$	
		$\mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\epsilon \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$
		$\mathbf{v} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$, $\quad \rho \leftarrow \mu\mathbf{G} + \mathbf{s}\mathbf{r}_2 + \epsilon$
	$\xleftarrow{\mathbf{v}, \rho}$	
$\mu \leftarrow \mathcal{C}.\mathsf{Decode}(\rho - \mathbf{v}\mathbf{y})$		

HQC Encryption Scheme [ABD+18]

Encryption scheme in **H**amming metric, using **Q**uasi-**C**yclic Codes

- Notation: Secret data - Public data - One-time Randomness
- **G** is the generator matrix of some public code $\mathcal{C}$.

Alice		Bob
$\mathsf{seed_h} \xleftarrow{\$} \{0,1\}^\lambda$, $\mathbf{h} \xleftarrow{\mathsf{seed_h}} \mathbb{F}_2^n$ $\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{h}\mathbf{y}$	$\xrightarrow{\mathsf{seed_h}, \mathbf{s}}$	$\mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\epsilon \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ $\mathbf{v} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$, $\rho \leftarrow \mu\mathbf{G} + \mathbf{s}\mathbf{r}_2 + \epsilon$
$\mu \leftarrow \mathcal{C}.\mathsf{Decode}(\rho - \mathbf{v}\mathbf{y})$	$\xleftarrow{\mathbf{v}, \rho}$	

HQC Encryption Scheme [ABD+18]

Encryption scheme in **H**amming metric, using **Q**uasi-**C**yclic Codes

- Notation: Secret data - Public data - One-time Randomness
- $\mathbf{G}$ is the generator matrix of some public code $\mathcal{C}$.

Alice		Bob
$\mathsf{seed_h} \xleftarrow{\$} \{0,1\}^\lambda$, $\mathbf{h} \xleftarrow{\mathsf{seed_h}} \mathbb{F}_2^n$ $\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{h}\mathbf{y}$	$\xrightarrow{\mathsf{seed_h}, \mathbf{s}}$	$\mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\boldsymbol{\epsilon} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ $\mathbf{v} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$, $\quad \boldsymbol{\rho} \leftarrow \boldsymbol{\mu}\mathbf{G} + \mathbf{s}\mathbf{r}_2 + \boldsymbol{\epsilon}$
$\boldsymbol{\mu} \leftarrow \mathcal{C}.\mathsf{Decode}(\boldsymbol{\rho} - \mathbf{v}\mathbf{y})$	$\xleftarrow{\mathbf{v}, \boldsymbol{\rho}}$	

HQC Encryption Scheme [ABD+18]

Encryption scheme in **H**amming metric, using **Q**uasi-**C**yclic Codes

- Notation: Secret data - Public data - One-time Randomness
- $\mathbf{G}$ is the generator matrix of some public code $\mathcal{C}$.

Alice		Bob
$\mathsf{seed}_{\mathbf{h}} \xleftarrow{\$} \{0,1\}^\lambda$, $\mathbf{h} \xleftarrow{\mathsf{seed}_{\mathbf{h}}} \mathbb{F}_2^n$ $\mathbf{x}, \mathbf{y} \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{h}\mathbf{y}$	$\xrightarrow{\mathsf{seed}_{\mathbf{h}}, \mathbf{s}}$	
		$\mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$, $\boldsymbol{\epsilon} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ $\mathbf{v} \leftarrow \mathbf{r}_1 + \mathbf{h}\mathbf{r}_2$, $\quad \boldsymbol{\rho} \leftarrow \boldsymbol{\mu}\mathbf{G} + \mathbf{s}\mathbf{r}_2 + \boldsymbol{\epsilon}$
$\boldsymbol{\mu} \leftarrow \mathcal{C}.\mathsf{Decode}(\boldsymbol{\rho} - \mathbf{v}\mathbf{y})$	$\xleftarrow{\mathbf{v}, \boldsymbol{\rho}}$	

Correctness

Correctness Property

$$\text{Decrypt}\left(\mathsf{sk}, \text{Encrypt}\left(\mathsf{pk}, \boldsymbol{\mu}, \theta\right)\right) = \boldsymbol{\mu}$$

$\mathcal{C}$.Decode correctly decodes $\rho - \mathbf{v} \cdot \mathbf{y}$ whenever

the error term is **not too big**

$$\omega\left(\mathbf{s} \cdot \mathbf{r}_2 - \mathbf{v} \cdot \mathbf{y} + \epsilon\right) \leq \delta$$

$$\omega\left(\left(\mathbf{x} + \mathbf{h} \cdot \mathbf{y}\right) \cdot \mathbf{r}_2 - \left(\mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2\right) \cdot \mathbf{y} + \epsilon\right) \leq \delta$$

$$\omega(\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \epsilon) \leq \delta$$

Error distribution analysis $\rightarrow$ Decryption failure probability better understood

Correctness

Correctness Property

$$\text{Decrypt}\left(\mathsf{sk}, \text{Encrypt}\left(\mathsf{pk}, \boldsymbol{\mu}, \theta\right)\right) = \boldsymbol{\mu}$$

$\mathcal{C}$.Decode correctly decodes $\boldsymbol{\rho} - \mathbf{v} \cdot \mathbf{y}$ whenever

the error term is **not too big**

$$\omega\left(\mathbf{s} \cdot \mathbf{r}_2 - \mathbf{v} \cdot \mathbf{y} + \boldsymbol{\epsilon}\right) \leq \delta$$

$$\omega\left(\left(\mathbf{x} + \mathbf{h} \cdot \mathbf{y}\right) \cdot \mathbf{r}_2 - \left(\mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2\right) \cdot \mathbf{y} + \boldsymbol{\epsilon}\right) \leq \delta$$

$$\omega(\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}) \leq \delta$$

Error distribution analysis $\rightarrow$ Decryption failure probability better understood

Correctness

Correctness Property

$$\text{Decrypt}\left(\mathsf{sk}, \text{Encrypt}\left(\mathsf{pk}, \boldsymbol{\mu}, \theta\right)\right) = \boldsymbol{\mu}$$

$\mathcal{C}$.Decode correctly decodes $\boldsymbol{\rho} - \mathbf{v} \cdot \mathbf{y}$ whenever

the error term is **not too big**

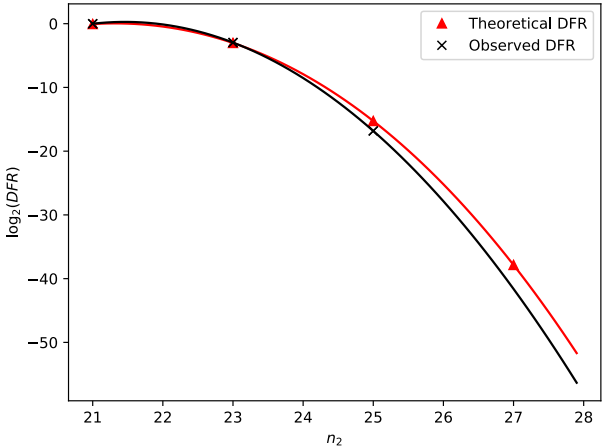
$$\omega\left(\mathbf{s} \cdot \mathbf{r}_2 - \mathbf{v} \cdot \mathbf{y} + \boldsymbol{\epsilon}\right) \leq \delta$$

$$\omega\left(\left(\mathbf{x} + \mathbf{h} \cdot \mathbf{y}\right) \cdot \mathbf{r}_2 - \left(\mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2\right) \cdot \mathbf{y} + \boldsymbol{\epsilon}\right) \leq \delta$$

$$\omega(\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}) \leq \delta$$

Error distribution analysis $\rightarrow$ Decryption failure probability better understood

Decryption failure rate



- In red: Theoretical DFR with a reasonable assumption on the error distribution
- In black: Observed/Empirical DFR, obtained by running 10^5 encryptions/decryptions over 10^3 codes with $n_1 = 766$, $k_1 = 256$, $\delta_1 = 57$, $w = 67$, $w_{\mathbf{r}} = 77$ and varying n_2.

Parameters

Instance	Cryptosystem Parameters								
	n_1	n_2	$n \approx n_1 n_2$	k_1	δ_1	w	$w_{\mathbf{r}} = w_{\mathbf{e}}$	security	p_{fail}
Basic-I	766	29	22,229	256	57	67	77	128	$< 2^{-64}$
Basic-II	766	31	23,747	256	57	67	77	128	$< 2^{-96}$
Basic-III	796	31	24,677	256	57	67	77	128	$< 2^{-128}$
Advanced-I	796	51	40,597	256	60	101	117	192	$< 2^{-64}$
Advanced-II	766	57	43,669	256	57	101	117	192	$< 2^{-128}$
Advanced-III	766	61	46,747	256	57	101	117	192	$< 2^{-192}$
Paranoiac-I	766	77	59,011	256	57	133	153	256	$< 2^{-64}$
Paranoiac-II	766	83	63,587	256	57	133	153	256	$< 2^{-128}$
Paranoiac-III	796	85	67,699	256	60	133	153	256	$< 2^{-192}$
Paranoiac-IV	796	89	70,853	256	60	133	153	256	$< 2^{-256}$

Outline

Security Model and Hybrid Argument

- Key exchange as an encryption scheme
- Same as Ding *et al.* [Din12, DXL12], Peikert's [Pei14], BCNS [BCNS15] and NewHope [ADPS16]
- Usual game:

$\mathbf{Exp}^{\mathsf{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$
3. $(\epsilon_0, \epsilon_1) \leftarrow \mathcal{A}(\mathtt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \epsilon_b, \theta)$
5. $b' \leftarrow \mathcal{A}(\mathtt{GUESS} : \mathbf{c}^*)$
6. $\mathtt{RETURN}\ \ b'$

- Hybrid argument:
  1. Construct a sequence of games transitioning from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$
  2. Prove they are indistinguishable one from another

Security Model and Hybrid Argument

- Key exchange as an encryption scheme
- Same as Ding *et al.* [Din12, DXL12], Peikert's [Pei14], BCNS [BCNS15] and NewHope [ADPS16]

- Usual game:

$\mathbf{Exp}^{\mathsf{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$
1. $\mathsf{param} \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{param})$
3. $(\epsilon_0, \epsilon_1) \leftarrow \mathcal{A}(\mathtt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \epsilon_b, \theta)$
5. $b' \leftarrow \mathcal{A}(\mathtt{GUESS} : \mathbf{c}^*)$
6. $\mathtt{RETURN}\ \ b'$

- Hybrid argument:
  1. Construct a sequence of games transitioning from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$
  2. Prove they are indistinguishable one from another

Security Model and Hybrid Argument

- Key exchange as an encryption scheme
- Same as Ding *et al.* [Din12, DXL12], Peikert's [Pei14], BCNS [BCNS15] and NewHope [ADPS16]
- Usual game:

$\mathbf{Exp}^{\mathrm{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$
1. param $\leftarrow$ Setup(1^λ)
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow$ KeyGen(param)
3. $(\epsilon_0, \epsilon_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow$ Encrypt$(\mathsf{pk}, \epsilon_b, \theta)$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN b'

- Hybrid argument:
  1. Construct a sequence of games transitioning from Enc(ϵ_0) to Enc(ϵ_1)
  2. Prove they are indistinguishable one from another

Security Model and Hybrid Argument

- Key exchange as an encryption scheme
- Same as Ding *et al.* [Din12, DXL12], Peikert's [Pei14], BCNS [BCNS15] and NewHope [ADPS16]
- Usual game:

$\mathbf{Exp}^{\mathrm{ind}-b}_{\mathcal{E},\mathcal{A}}(\lambda)$
1. param $\leftarrow$ Setup(1^λ)
2. $(\mathsf{pk}, \mathsf{sk}) \leftarrow$ KeyGen(param)
3. $(\epsilon_0, \epsilon_1) \leftarrow \mathcal{A}(\texttt{FIND} : \mathsf{pk})$
4. $\mathbf{c}^* \leftarrow$ Encrypt$(\mathsf{pk}, \epsilon_b, \theta)$
5. $b' \leftarrow \mathcal{A}(\texttt{GUESS} : \mathbf{c}^*)$
6. RETURN b'

- Hybrid argument:
  1. Construct a sequence of games transitioning from Enc(ϵ_0) to Enc(ϵ_1)
  2. Prove they are indistinguishable one from another

Outline

Security

Definition (SD Distribution)

For positive integers, n, k, and w, the *SD(n, k, w) Distribution* chooses $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ and $\mathbf{x} \xleftarrow{\$} \mathbb{F}^n$ such that $\omega(\mathbf{x}) = w$, and outputs $(\mathbf{H}, \mathbf{H}\mathbf{x}^\top)$.

Definition (Decisional s-QCSD Problem)

For positive integers n, k, w, s, a random parity check matrix $\mathbf{H}$ of a QC code $\mathcal{C}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{F}^n$, the *Decisional s-Quasi-Cyclic SD Problem s-DQCSD(n, k, w)* asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^\top)$ came from the s-QCSD(n, k, w) distribution or the uniform distribution over $\mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{n-k}$.

Theorem

HQC is IND-CPA *under the* 2-*DQCSD and* 3-*DQCSD assumptions.* $\rightarrow$ sketch of proof

Security

Definition (SD Distribution)

For positive integers, n, k, and w, the *SD(n, k, w) Distribution* chooses $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ and $\mathbf{x} \xleftarrow{\$} \mathbb{F}^n$ such that $\omega(\mathbf{x}) = w$, and outputs $(\mathbf{H}, \mathbf{H}\mathbf{x}^\top)$.

Definition (Decisional s-QCSD Problem)

For positive integers n, k, w, s, a random parity check matrix $\mathbf{H}$ of a QC code $\mathcal{C}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{F}^n$, the *Decisional s-Quasi-Cyclic SD Problem s-DQCSD(n, k, w)* asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^\top)$ came from the s-QCSD(n, k, w) distribution or the uniform distribution over $\mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{n-k}$.

Theorem

HQC is IND-CPA *under the* 2-*DQCSD and* 3-*DQCSD assumptions.* $\rightarrow$ sketch of proof

Security

Definition (SD Distribution)

For positive integers, n, k, and w, the *SD(n, k, w) Distribution* chooses $\mathbf{H} \xleftarrow{\$} \mathbb{F}^{(n-k)\times n}$ and $\mathbf{x} \xleftarrow{\$} \mathbb{F}^n$ such that $\omega(\mathbf{x}) = w$, and outputs $(\mathbf{H}, \mathbf{H}\mathbf{x}^\top)$.

Definition (Decisional s-QCSD Problem)

For positive integers n, k, w, s, a random parity check matrix $\mathbf{H}$ of a QC code $\mathcal{C}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{F}^n$, the *Decisional s-Quasi-Cyclic SD Problem s-DQCSD(n, k, w)* asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^\top)$ came from the s-QCSD(n, k, w) distribution or the uniform distribution over $\mathbb{F}^{(n-k)\times n} \times \mathbb{F}^{n-k}$.

Theorem

HQC is IND-CPA *under the* 2*-DQCSD and* 3*-DQCSD assumptions.* $\rightarrow$ sketch of proof

Outline

A particular decoding

- HQC requires $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}$ to be "small" to correctly decode
- Ouroboros further exploits the shape of the error [DGZ17]

Cyclic Error Decoding (CED) Problem

- Let $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$ with $w = \mathcal{O}(\sqrt{n})$, and $\mathbf{e} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ a random error vector.
- Given $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ and $\mathbf{e}_c \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \mathbf{e}$ such that $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$, find $(\mathbf{r}_1, \mathbf{r}_2)$.

- This is essentially a *noisy* SD problem

A particular decoding

- HQC requires $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}$ to be "small" to correctly decode
- Ouroboros further exploits the shape of the error [DGZ17]

Cyclic Error Decoding (CED) Problem

- Let $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$ with $w = \mathcal{O}(\sqrt{n})$, and $\mathbf{e} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ a random error vector.
- Given $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ and $\mathbf{e}_c \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \mathbf{e}$ such that $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$, find $(\mathbf{r}_1, \mathbf{r}_2)$.

- This is essentially a *noisy* SD problem

A particular decoding

- HQC requires $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}$ to be "small" to correctly decode
- Ouroboros further exploits the shape of the error [DGZ17]

Cyclic Error Decoding (CED) Problem

- Let $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$ with $w = \mathcal{O}(\sqrt{n})$, and $\mathbf{e} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ a random error vector.
- Given $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ and $\mathbf{e}_c \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \boldsymbol{e}$ such that $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$, find $(\mathbf{r}_1, \mathbf{r}_2)$.

- This is essentially a *noisy* SD problem

A particular decoding

- HQC requires $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}$ to be "small" to correctly decode
- Ouroboros further exploits the shape of the error [DGZ17]

Cyclic Error Decoding (CED) Problem

- Let $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$ with $w = \mathcal{O}(\sqrt{n})$, and $\mathbf{e} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ a random error vector.
- Given $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ and $\mathbf{e_c} \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \boldsymbol{e}$ such that $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$, find $(\mathbf{r}_1, \mathbf{r}_2)$.

- This is essentially a *noisy* SD problem

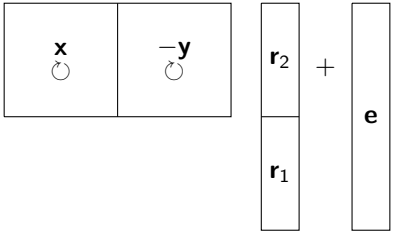
A particular decoding

- HQC requires $\mathbf{x} \cdot \mathbf{r}_2 - \mathbf{r}_1 \cdot \mathbf{y} + \boldsymbol{\epsilon}$ to be "small" to correctly decode
- Ouroboros further exploits the shape of the error [DGZ17]

Cyclic Error Decoding (CED) Problem

- Let $\mathbf{x}, \mathbf{y}, \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{S}_w^n(\mathbb{F}_2)$ with $w = \mathcal{O}(\sqrt{n})$, and $\mathbf{e} \xleftarrow{\$} \mathcal{S}_{cw}^n(\mathbb{F}_2)$ a random error vector.
- Given $(\mathbf{x}, \mathbf{y}) \in (\mathcal{S}_w^n(\mathbb{F}_2))^2$ and $\mathbf{e_c} \leftarrow \mathbf{x}\mathbf{r}_2 - \mathbf{y}\mathbf{r}_1 + \boldsymbol{e}$ such that $\omega(\mathbf{r}_1) = \omega(\mathbf{r}_2) = w$, find $(\mathbf{r}_1, \mathbf{r}_2)$.

- This is essentially a *noisy* SD problem



Outline

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Hard Decision Decoding: BitFlipping

- Iterative decoding for **L**ow **D**ensity **P**arity **C**heck codes [Gal62]
- Decoding capacity increase linearly with the code length (for LDPC)

Intuition

1. Compute the number of unsatisfied parity-check equations for each bit of the message
2. If this number is greater than some *threshold*, flip the bit and go to 1.
3. Stop when the syndrome is null (or after a certain number of iterations).

- Easy to understand, implement, and natively pretty efficient
- The threshold value is crucial [CS16]

Ouroboros (*aka.* BIKE-3)

The BitFlipping algorithm can be modified to handle noisy syndrome (for almost free!).

Outline

Reduction Compliant Parameters

	Ouroboros Parameters					
Instance	n	w	w_e	threshold	security	DFR
Low-I	$5,851$	47	94	30	80	$0.92 \cdot 10^{-5}$
Low-II	$5,923$	47	94	30	80	$2.3 \cdot 10^{-6}$
Medium-I	$13,691$	75	150	45	128	$0.96 \cdot 10^{-5}$
Medium-II	$14,243$	75	150	45	128	$1.09 \cdot 10^{-6}$
Strong-I	$40,013$	147	294	85	256	$4.20 \cdot 10^{-5}$
Strong-II	$40,973$	147	294	85	256	$< 10^{-6}$

Table: Parameter sets for Ouroboros

Optimized Parameters (wrt. Best Know Attacks)

	Ouroboros Optimized Parameters					
Instance	n	w	w_e	threshold	security	DFR
Low-I	$4,813$	41	123	27	80	$2.23 \cdot 10^{-5}$
Low-II	$5,003$	41	123	27	80	$2.60 \cdot 10^{-6}$
Medium-I	$10,301$	67	201	42	128	$1.01 \cdot 10^{-4}$
Medium-II	$10,837$	67	201	42	128	$< 10^{-7}$
Strong-I	$32,771$	131	393	77	256	$< 10^{-4}$
Strong-II	$33,997$	131	393	77	256	$< 10^{-7}$

Table: Optimized parameter sets for Ouroboros in Hamming metric

Outline

NIST's call for standardization of post-quantum algorithms

National **I**nstitute of **S**tandards and **T**echnologies

- 3rd call for standardization
- Asks for post-quantum cryptographic algorithms
- 3 categories :
  - Encryption
  - Key exchange
  - Signature
- Many candidates:
  - Error correcting codes,
  - Lattices,
  - Multivariate,
  - Hash functions,
  - ...

- November 2016: anouncement
- November 2017: submission deadline (82 submissions)
- December 2017: 1st round (02-20-18: 66 concurrents)
- April 2018: 1st standardization conference
- ...
- In 3 to 5 years, several algorithms will eventually be standardized

NIST's call for standardization of post-quantum algorithms

NIST **N**ational **I**nstitute of **S**tandards and **T**echnologies

- 3rd call for standardization
- Asks for post-quantum cryptographic algorithms
- 3 categories :
  - Encryption
  - Key exchange
  - Signature
- Many candidates:
  - Error correcting codes,
  - Lattices,
  - Multivariate,
  - Hash functions,
  - ...

- November 2016: anouncement
- November 2017: submission deadline (82 submissions)
- December 2017: 1st round (02-20-18: 66 concurrents)
- April 2018: 1st standardization conference
- ...
- In 3 to 5 years, several algorithms will eventually be standardized

NIST's call for standardization of post-quantum algorithms

NIST **N**ational **I**nstitute of **S**tandards and **T**echnologies

- 3rd call for standardization
- Asks for post-quantum cryptographic algorithms
- 3 categories :
  - Encryption
  - Key exchange
  - Signature
- Many candidates:
  - Error correcting codes,
  - Lattices,
  - Multivariate,
  - Hash functions,
  - ...

- November 2016: anouncement
- November 2017: submission deadline (82 submissions)
- December 2017: 1st round (02-20-18: 66 concurrents)
- April 2018: 1st standardization conference
- ...
- In 3 to 5 years, several algorithms will eventually be standardized

HQC

HQC Home Documentation Implementation Contact

Overview

HQC (**H**amming **Q**uasi-**C**yclic) is a code-based public key encryption scheme designed to provide security against attacks by both classical and quantum computers. It relies on quasi-cyclic codes as well as BCH codes. HQC has been submitted to the NIST's Post-Quantum Cryptography Standardization Project.

Submitters (alphabetical order)

- Carlos Aguilar Melchor, University of Toulouse (FR)
- Nicolas Aragon, University of Limoges (FR)
- Slim Bettaieb, Worldline (FR)
- Loïc Bidoux, Worldline (FR)
- Olivier Blazy, University of Limoges (FR)
- Jean-Christophe Deneuville, INSA-CVL & University of Limoges (FR)
- Philippe Gaborit, University of Limoges (FR)
- Edoardo Persichetti, Florida Atlantic University (US)
- Gilles Zémor, University of Bordeaux (FR)

https://pqc-hqc.org/

Ouroboros *aka.* BIKE-3

BIKE - Bit Flipping Key Encapsulation

Home

Specification

Implementation

KAT

Authors

Acknowledgments

Contact

Welcome to the BIKE Website

This website will be used by the BIKE team as its **official communication media**.

BIKE is a code-based key encapsulation suite based on QC-MDPC (Quasi-Cyclic Moderate Density Parity-Check) codes, which was submitted to the NIST standardization process on post-quantum cryptography. The BIKE suite consists of three variants: BIKE-1, BIKE-2 and BIKE-3. Each variant offers different performance trade-offs.

Timeline

12/20/2017 - NIST accepts BIKE as a "complete and proper" submission.

11/30/2017 - BIKE is submitted to the NIST standardization process.

Specification Document

The specification document of BIKE-1, BIKE-2 and BIKE-3 can be found here.

http://bikesuite.org/

Other submissions

- Encryption schemes:
  - RQC (rank metric version of HQC) → https://pqc-rqc.org/
  - LOCKER (uses LRPC codes), no (public) website yet
- Key exchange protocols:
  - Ouroboros-R (rank metric version of Ouroboros) → https://pqc-ouroborosr.org/
  - LAKE (uses LRPC codes), no (public) website yet

All 1st round submissions available here.

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Conclusion

In this talk

- HQC: *efficient* encryption from random quasi-cyclic codes, *without masking* the underlying structure
- Ouroboros: a *secure*, *simple*, and *efficient* code-based key exchange protocol
- Competitive parameters

Further Improvements

- Improve BitFlipping threshold [CS16]
- Switching to Rank metric drastically improves parameters! → interlude?
- Optimize implementation
- More fancy code combinations for HQC
- Other codes? Other metrics? Lattices?

Thanks!

HQC (& RQC) available @ http://unil.im/HQC-RQC

Ouroboros available @ http://unil.im/ouroboros

Thanks!

Carlos Aguilar, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor.
Efficient encryption from random quasi-cyclic codes.
IEEE Transactions on Information Theory, 2018.

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe.
Post-quantum key exchange - A new hope.
In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343. USENIX Association, 2016.

Joppe W. Bos, Craig Costello, Michael Naehrig, and Douglas Stebila.
Post-quantum key exchange for the TLS protocol from the ring learning with errors problem.
In *2015 IEEE Symposium on Security and Privacy*, pages 553–570. IEEE Computer Society Press, May 2015.

Julia Chaulet and Nicolas Sendrier.
Worst case qc-mdpc decoder for mceliece cryptosystem.
In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 1366–1370. IEEE, 2016.

Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor.
Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory.
In *International Workshop on Post-Quantum Cryptography*, pages 18–34. Springer, 2017.

Jintai Ding.
New cryptographic constructions using generalized learning with errors problem.
Cryptology ePrint Archive, Report 2012/387, 2012.

Jintai Ding, Xiang Xie, and Xiaodong Lin.
A simple provably secure key exchange scheme based on the learning with errors problem.
Cryptology ePrint Archive, Report 2012/688, 2012.

Chris Peikert.
Lattice cryptography for the internet.
In Michele Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, volume 8772 of *Lecture Notes in Computer Science*, pages 197–219. Springer, 2014.

Nicolas Sendrier.
Encoding information into constant weight words.
In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 435–438. IEEE, 2005.

HQC (& RQC) available @ http://unil.im/HQC-RQC

Ouroboros available @ http://unil.im/ouroboros

Rank Metric Interlude (1/2)

Rank metric defined over (finite) extensions of finite fields

- $\mathbb{F}_q$ a finite field with q a power of a prime.
- $\mathbb{F}_{q^m}$ an extension of degree m of $\mathbb{F}_q$.
- $\mathbb{F}_{q^m}$ can be seen as a vector space on $\mathbb{F}_q$.
- $\mathcal{B} = (b_1, ..., b_m)$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Let $\mathbf{v} = (v_1, \ldots, v_n)$ be a word of length n in $\mathbb{F}_{q^m}$.

Any coordinate $v_j = \sum_{i=1}^{m} v_{ij} b_i$ with $v_{ij} \in \mathbb{F}_q$.

$$\mathbf{v} = (v_1, ..., v_n) \rightarrow \mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix}$$

Rank weight of word

$\mathbf{v}$ has rank $r = \mathrm{rank}(\mathbf{v})$ iff the rank of $\mathbf{V} = (v_{ij})_{ij}$ is r.
Equivalently $\mathrm{rank}(\mathbf{v}) = r \Leftrightarrow v_j \in V_r \subset \mathbb{F}_{q^m}^n$ with $\dim(V_r)$=r.

Rank Metric Interlude (1/2)

Rank metric defined over (finite) extensions of finite fields

- $\mathbb{F}_q$ a finite field with q a power of a prime.
- $\mathbb{F}_{q^m}$ an extension of degree m of $\mathbb{F}_q$.
- $\mathbb{F}_{q^m}$ can be seen as a vector space on $\mathbb{F}_q$.
- $\mathcal{B} = (b_1, ..., b_m)$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Let $\mathbf{v} = (v_1, \ldots, v_n)$ be a word of length n in $\mathbb{F}_{q^m}$.

Any coordinate $v_j = \sum_{i=1}^{m} v_{ij} b_i$ with $v_{ij} \in \mathbb{F}_q$.

$$\mathbf{v} = (v_1, ..., v_n) \rightarrow \mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix}$$

Rank weight of word

$\mathbf{v}$ has rank $r = \mathrm{rank}(\mathbf{v})$ iff the rank of $\mathbf{V} = (v_{ij})_{ij}$ is r.
Equivalently $\mathrm{rank}(\mathbf{v}) = r \Leftrightarrow v_j \in V_r \subset \mathbb{F}_{q^m}^n$ with $\dim(V_r)$=r.

Rank Metric Interlude (1/2)

Rank metric defined over (finite) extensions of finite fields

- $\mathbb{F}_q$ a finite field with q a power of a prime.
- $\mathbb{F}_{q^m}$ an extension of degree m of $\mathbb{F}_q$.
- $\mathbb{F}_{q^m}$ can be seen as a vector space on $\mathbb{F}_q$.
- $\mathcal{B} = (b_1, ..., b_m)$ a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Let $\mathbf{v} = (v_1, \ldots, v_n)$ be a word of length n in $\mathbb{F}_{q^m}$.

Any coordinate $v_j = \sum_{i=1}^{m} v_{ij} b_i$ with $v_{ij} \in \mathbb{F}_q$.

$$\mathbf{v} = (v_1, ..., v_n) \rightarrow \mathbf{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix}$$

Rank weight of word

$\mathbf{v}$ has rank $r = \mathrm{rank}(\mathbf{v})$ iff the rank of $\mathbf{V} = (v_{ij})_{ij}$ is r.
Equivalently $\mathrm{rank}(\mathbf{v}) = r \Leftrightarrow v_j \in V_r \subset \mathbb{F}_{q^m}^n$ with $\dim(V_r)$=r.

Rank Metric Interlude (2/2)

- Best Known Attacks have worse complexity in rank metric ($2^{\mathcal{O}(n^2)}$) than in Hamming metric ($2^{\mathcal{O}(n)}$)
- Consequence: worse attacks $\Rightarrow$ better parameters

Instance	Ouroboros-R Parameters						
	key size (bits)	n	m	q	w	security	decoding failure
Ouroboros-R-I	1,591	37	43	2	5	100	10^{-4}
Ouroboros-R-II	2,809	53	53	2	5	128	10^{-8}
Ouroboros-R-III	3,953	59	67	2	6	192	10^{-7}
Ouroboros-R-IV	5,293	67	79	2	7	256	10^{-5}
Ouroboros-R-V	5,618	53	53	4	6	256	10^{-10}

Parameter sets for Ouroboros-R in rank metric.

back to conclusion

Rank Metric Interlude (2/2)

- Best Known Attacks have worse complexity in rank metric ($2^{\mathcal{O}(n^2)}$) than in Hamming metric ($2^{\mathcal{O}(n)}$)
- Consequence: worse attacks $\Rightarrow$ better parameters

Instance	Ouroboros-R Parameters						
	key size (bits)	n	m	q	w	security	decoding failure
Ouroboros-R-I	1,591	37	43	2	5	100	10^{-4}
Ouroboros-R-II	2,809	53	53	2	5	128	10^{-8}
Ouroboros-R-III	3,953	59	67	2	6	192	10^{-7}
Ouroboros-R-IV	5,293	67	79	2	7	256	10^{-5}
Ouroboros-R-V	5,618	53	53	4	6	256	10^{-10}

Parameter sets for Ouroboros-R in rank metric.

back to conclusion

Rank Metric Interlude (2/2)

- Best Known Attacks have worse complexity in rank metric ($2^{\mathcal{O}(n^2)}$) than in Hamming metric ($2^{\mathcal{O}(n)}$)
- Consequence: worse attacks $\Rightarrow$ better parameters

Instance	Ouroboros-R Parameters						
	key size (bits)	n	m	q	w	security	decoding failure
Ouroboros-R-I	1,591	37	43	2	5	100	10^{-4}
Ouroboros-R-II	2,809	53	53	2	5	128	10^{-8}
Ouroboros-R-III	3,953	59	67	2	6	192	10^{-7}
Ouroboros-R-IV	5,293	67	79	2	7	256	10^{-5}
Ouroboros-R-V	5,618	53	53	4	6	256	10^{-10}

Parameter sets for Ouroboros-R in rank metric.

back to conclusion

Sketch of proof

Sequence of games from $\mathrm{Enc}(\epsilon_0)$ to $\mathrm{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathrm{Enc}(\epsilon_0) \arrow[r] & \mathrm{Enc}_{s^\star}(\epsilon_0) \arrow[r] & \mathrm{Enc}_{s^\star,r^\star}(\epsilon_0) \arrow[d] \\
\mathrm{Enc}(\epsilon_1) & \mathrm{Enc}_{s^\star}(\epsilon_1) \arrow[l] & \mathrm{Enc}_{s^\star,r^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d] \\
\mathsf{Enc}(\epsilon_1) & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[l] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d] \\
\mathsf{Enc}(\epsilon_1) & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[l] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d] \\
\mathsf{Enc}(\epsilon_1) & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[l] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d] \\
\mathsf{Enc}(\epsilon_1) & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[l] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d] \\
\mathsf{Enc}(\epsilon_1) & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[l] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1) \arrow[l]
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow, "\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda)"] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d, leftrightarrow] \\
\mathsf{Enc}(\epsilon_1) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1)
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security

Sketch of proof

Sequence of games from $\mathsf{Enc}(\epsilon_0)$ to $\mathsf{Enc}(\epsilon_1)$

$$\begin{tikzcd}
\mathsf{Enc}(\epsilon_0) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow, "\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda)"] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_0) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_0) \arrow[d, leftrightarrow] \\
\mathsf{Enc}(\epsilon_1) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star}(\epsilon_1) \arrow[r, leftrightarrow] & \mathsf{Enc}_{\mathbf{s}^\star,\mathbf{r}^\star}(\epsilon_1)
\end{tikzcd}$$

$$\mathbf{Adv}^{\mathsf{ind}}_{\mathcal{E},\mathcal{A}}(\lambda) \leq 2 \cdot \left(\mathbf{Adv}^{\text{2-DQCSD}}(\lambda) + \mathbf{Adv}^{\text{3-DQCSD}}(\lambda)\right)$$

back to security