CODE-BASED CRYPTOGRAPHY: STATE OF THE ART

PART II

Edoardo Persichetti

19 March 2019

CHARLES E. SCHMIDT COLLEGE OF SCIENCE
Florida Atlantic University

In This Talk

- Structured Codes
- Sparse-Matrix Codes
- Rank Metric
- Conclusions

Part I

Structured Codes

Traditional approach at current security levels produces very large keys: several Kb to $\approx$ 1Mb.

(Classic McEliece/NTS-KEM).

STRUCTURED CODES

Traditional approach at current security levels produces very large keys: several Kb to $\approx$ 1Mb.

(Classic McEliece/NTS-KEM).

The problem is: public key is a large matrix, size $\mathcal{O}(n^2)$.

Structured Codes

Traditional approach at current security levels produces very large keys: several Kb to $\approx$ 1Mb.

(Classic McEliece/NTS-KEM).

The problem is: public key is a large matrix, size $\mathcal{O}(n^2)$.

Idea: public matrix with compact description (Gaborit '05).

STRUCTURED CODES

Traditional approach at current security levels produces very large keys: several Kb to $\approx$ 1Mb.

(Classic McEliece/NTS-KEM).

The problem is: public key is a large matrix, size $\mathcal{O}(n^2)$.

Idea: public matrix with compact description (Gaborit '05).

This would allow to describe public-key more efficiently.

STRUCTURED CODES

Traditional approach at current security levels produces very large keys: several Kb to $\approx$ 1Mb.
(Classic McEliece/NTS-KEM).

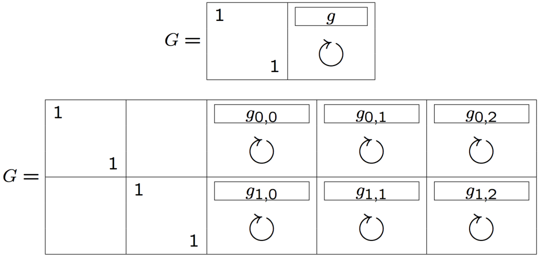
The problem is: public key is a large matrix, size $\mathcal{O}(n^2)$.

Idea: public matrix with compact description (Gaborit '05).

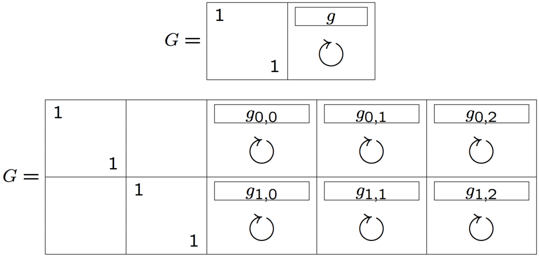
This would allow to describe public-key more efficiently.

Need families of codes with particular automorphism group.

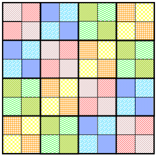
Quasi-Cyclic Codes (Berger, Cayrel, Gaborit, Otmani '09).



Quasi-Cyclic Codes (Berger, Cayrel, Gaborit, Otmani '09).



Quasi-Dyadic Codes (Misoczki, Barreto '09).



Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

SECURITY

Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

Problem: extra structure = extra info for attacker.

Security

Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

Problem: extra structure = extra info for attacker.

Critical algebraic attack (Faugère, Otmani, Perret, Tillich '10).

SECURITY

Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

Problem: extra structure = extra info for attacker.

Critical algebraic attack (Faugère, Otmani, Perret, Tillich '10).

Solve system of equations derived from $H \cdot G^T = 0$ to recover private key.

SECURITY

Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

Problem: extra structure = extra info for attacker.

Critical algebraic attack (Faugère, Otmani, Perret, Tillich '10).

Solve system of equations derived from $H \cdot G^T = 0$ to recover private key.

QC/QD + algebraic structure crucial to reduce number of unknowns of system.

SECURITY

Several families have QC/QD description:
GRS, Goppa, Generalized Srivastava (P. '11).

Problem: extra structure = extra info for attacker.

Critical algebraic attack (Faugère, Otmani, Perret, Tillich '10).

Solve system of equations derived from $H \cdot G^T = 0$ to recover private key.

QC/QD + algebraic structure crucial to reduce number of unknowns of system.

After a few years of fixes and new attacks: keys getting bigger, confidence/interest getting smaller.
(Faugère, Otmani, Perret, de Portzamparc, Tillich '16, Barelli-Couvreur '18).

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Designed in a conservative way.

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Designed in a conservative way.

BIG QUAKE parameters (bytes):

q	m	n	t	PK Size	SK Size	Ciph Size	Security
2	18	10,070	190	149,625	41,804	492	5
2	18	7,410	152	84,132	30,860	406	3
2	12	3,510	91	25,389	14,772	201	1

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Designed in a conservative way.

BIG QUAKE parameters (bytes):

q	m	n	t	PK Size	SK Size	Ciph Size	Security
2	18	10,070	190	149,625	41,804	492	5
2	18	7,410	152	84,132	30,860	406	3
2	12	3,510	91	25,389	14,772	201	1

DAGS: based on Quasi-Dyadic q-ary Generalized Srivastava Codes

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Designed in a conservative way.

BIG QUAKE parameters (bytes):

q	m	n	t	PK Size	SK Size	Ciph Size	Security
2	18	10,070	190	149,625	41,804	492	5
2	18	7,410	152	84,132	30,860	406	3
2	12	3,510	91	25,389	14,772	201	1

DAGS: based on Quasi-Dyadic q-ary Generalized Srivastava Codes

More aggressive choice of parameters.

Case Study: NIST Submissions

BIG QUAKE: based on Quasi-Cyclic Binary Goppa Codes

Designed in a conservative way.

BIG QUAKE parameters (bytes):

q	m	n	t	PK Size	SK Size	Ciph Size	Security
2	18	10,070	190	149,625	41,804	492	5
2	18	7,410	152	84,132	30,860	406	3
2	12	3,510	91	25,389	14,772	201	1

DAGS: based on Quasi-Dyadic q-ary Generalized Srivastava Codes

More aggressive choice of parameters.

DAGS parameters (bytes):

q	m	n	t	PK Size	SK Size	Ciph Size	Security
2^8	2	1,600	176	19,712	6,400	1,632	5
2^8	2	1,216	176	11,264	4,864	1,248	3
2^6	2	832	104	8,112	2,496	656	1

Part II

SPARSE-MATRIX CODES

SPARSE-MATRIX CODES

Family of codes characterized by very sparse parity-check matrix.

.

Sparse-Matrix Codes

Family of codes characterized by very sparse parity-check matrix.

Definition 1 (LDPC Code)

An $[n, k]$ binary linear code which admits a parity-check matrix of constant row weight $w \in O(1)$.

.

Sparse-Matrix Codes

Family of codes characterized by very sparse parity-check matrix.

Definition 1 (LDPC Code)

An $[n, k]$ binary linear code which admits a parity-check matrix of constant row weight $w \in O(1)$.

If we write $H = (H_0 \mid H_1)$ resp. $r \times k$ and $r \times r$ then $G = (I_k \mid H_0^T H_1^{-T})$.

Sparse-Matrix Codes

Family of codes characterized by very sparse parity-check matrix.

Definition 1 (LDPC Code)

An $[n, k]$ binary linear code which admits a parity-check matrix of constant row weight $w \in O(1)$.

If we write $H = (H_0 \mid H_1)$ resp. $r \times k$ and $r \times r$ then $G = (I_k \mid H_0^T H_1^{-T})$.

The non-trivial block is dense, so this is a natural choice of public key for McEliece.

Sparse-Matrix Codes

Family of codes characterized by very sparse parity-check matrix.

Definition 1 (LDPC Code)

An $[n, k]$ binary linear code which admits a parity-check matrix of constant row weight $w \in O(1)$.

If we write $H = (H_0 \mid H_1)$ resp. $r \times k$ and $r \times r$ then $G = (I_k \mid H_0^T H_1^{-T})$.

The non-trivial block is dense, so this is a natural choice of public key for McEliece.

Decodable with very efficient probabilistic "bit flipping" algorithm (Gallager, '63), small decoding failure rate (DFR).

Security

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

This is also a decoding problem! So we have essentially one assumption.

SECURITY

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

This is also a decoding problem! So we have essentially one assumption.

Best attacks: generic “search” algorithms like Information-Set Decoding (ISD).

SECURITY

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

This is also a decoding problem! So we have essentially one assumption.

Best attacks: generic "search" algorithms like Information-Set Decoding (ISD).

MDPC: "relaxed" version of LDPC (Misoczki, Tillich, Sendrier and Barreto '12).

SECURITY

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

This is also a decoding problem! So we have essentially one assumption.

Best attacks: generic "search" algorithms like Information-Set Decoding (ISD).

MDPC: "relaxed" version of LDPC (Misoczki, Tillich, Sendrier and Barreto '12).

Change weight w from very low (≈ 10) to "moderate" ($O(\sqrt{n})$).

Distinguish public matrix $\approxeq$ look for low-weight codewords in the dual.

This is also a decoding problem! So we have essentially one assumption.

Best attacks: generic "search" algorithms like Information-Set Decoding (ISD).

MDPC: "relaxed" version of LDPC (Misoczki, Tillich, Sendrier and Barreto '12).

Change weight w from very low (≈ 10) to "moderate" ($O(\sqrt{n})$).

Still decodable, gain in security makes up for degradation.

Structures Sparse-Matrix Codes

Using "plain" LDPC/MDPC is not practical due to long code lengths.

Using “plain” LDPC/MDPC is not practical due to long code lengths.

Possible to build QC-LDPC/MDPC codes and have compact keys.

Structures Sparse-Matrix Codes

Using "plain" LDPC/MDPC is not practical due to long code lengths.

Possible to build QC-LDPC/MDPC codes and have compact keys.

Matrices formed by circulant blocks

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}$$

Using "plain" LDPC/MDPC is not practical due to long code lengths.

Possible to build QC-LDPC/MDPC codes and have compact keys.

Matrices formed by circulant blocks

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}$$

Correspond to ideals of $\mathcal{R} = \mathbb{F}_2[x]/(x^p - 1)$: describe using ring arithmetic.

Using “plain” LDPC/MDPC is not practical due to long code lengths.

Possible to build QC-LDPC/MDPC codes and have compact keys.

Matrices formed by circulant blocks

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}$$

Correspond to ideals of $\mathcal{R} = \mathbb{F}_2[x]/(x^p - 1)$: describe using ring arithmetic.

Sparse-matrix codes don’t possess inherent algebraic structure.

Structures Sparse-Matrix Codes

Using “plain” LDPC/MDPC is not practical due to long code lengths.

Possible to build QC-LDPC/MDPC codes and have compact keys.

Matrices formed by circulant blocks

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{p-1} \\ a_{p-1} & a_0 & \dots & a_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \dots & a_0 \end{bmatrix}$$

Correspond to ideals of $\mathcal{R} = \mathbb{F}_2[x]/(x^p - 1)$: describe using ring arithmetic.

Sparse-matrix codes don’t possess inherent algebraic structure.

QC property alone does not provide a structural attack.

Key Generation

- Choose h_0, h_1 in $\mathcal{R}$ of combined weight w.
- SK: parity-check matrix formed by circulant blocks h_0, h_1.
- PK: generator matrix formed by identity and $g = h_0 h_1^{-1}$.

Sparse-matrix McEliece

Key Generation

- Choose h_0, h_1 in $\mathcal{R}$ of combined weight w.
- SK: parity-check matrix formed by circulant blocks h_0, h_1.
- PK: generator matrix formed by identity and $g = h_0 h_1^{-1}$.

Encryption

- Take message $\mu \in \mathcal{R}$.
- Sample vectors e_0, e_1 in $\mathcal{R}$ of combined weight t.
- Output $c = (\mu + e_0, \mu \cdot g + e_1)$.

Key Generation

- Choose h_0, h_1 in $\mathcal{R}$ of combined weight w.
- SK: parity-check matrix formed by circulant blocks h_0, h_1.
- PK: generator matrix formed by identity and $g = h_0 h_1^{-1}$.

Encryption

- Take message $\mu \in \mathcal{R}$.
- Sample vectors e_0, e_1 in $\mathcal{R}$ of combined weight t.
- Output $c = (\mu + e_0, \mu \cdot g + e_1)$.

Decryption

- Set $(e_0, e_1) = Decode_{BitFlipping}(c)$.
- Return $\perp$ if decoding fails.
- Else recover μ (truncate).

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

$1, 2$: CAKE (Barreto, Gueron, Güneysu, Misoczki, P., Sendrier, Tillich, '17).

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

$1, 2$: CAKE (Barreto, Gueron, Güneysu, Misoczki, P., Sendrier, Tillich, '17).

3: Ouroboros (Deneuville, Gaborit, Zémor, '17).

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

$1, 2$: CAKE (Barreto, Gueron, Güneysu, Misoczki, P., Sendrier, Tillich, '17).

3: Ouroboros (Deneuville, Gaborit, Zémor, '17).

BIKE-1: use McEliece and non-systematic generator matrix to avoid polynomial inversion and save time (latency).

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

$1, 2$: CAKE (Barreto, Gueron, Güneysu, Misoczki, P., Sendrier, Tillich, ’17).

3: Ouroboros (Deneuville, Gaborit, Zémor, ’17).

BIKE-1: use McEliece and non-systematic generator matrix to avoid polynomial inversion and save time (latency).

BIKE-2: use Niederreiter and systematic parity-check with (possibly) pre-computed keys to save space (bandwidth).

BIKE

Suite of KEM schemes based on the bit-flipping decoder and QC-MDPC codes.

Three variants, independently published.

$1, 2$: CAKE (Barreto, Gueron, Güneysu, Misoczki, P., Sendrier, Tillich, '17).

3: Ouroboros (Deneuville, Gaborit, Zémor, '17).

BIKE-1: use McEliece and non-systematic generator matrix to avoid polynomial inversion and save time (latency).

BIKE-2: use Niederreiter and systematic parity-check with (possibly) pre-computed keys to save space (bandwidth).

BIKE-3: use "noisy" decoder to have simpler security reduction.

LEDACRYPT

Based on QC-LDPC codes.

LEDACRYPT

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

LEDACRYPT

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

Following a long line of work from Baldi, Chiaraluce et al. (2007-onwards).

LEDACRYPT

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

Following a long line of work from Baldi, Chiaraluce et al. (2007-onwards).

Variable number of blocks $n_0 = 2, 3, 4$.

LEDACRYPT

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

Following a long line of work from Baldi, Chiaraluce et al. (2007-onwards).

Variable number of blocks $n_0 = 2, 3, 4$.

Private key is made dense via secret matrix $Q \longrightarrow \approx$QC-MDPC.

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

Following a long line of work from Baldi, Chiaraluce et al. (2007-onwards).

Variable number of blocks $n_0 = 2, 3, 4$.

Private key is made dense via secret matrix $Q \longrightarrow \approx$QC-MDPC.

Specialized "Q-decoder" provides better decoding performance.

Based on QC-LDPC codes.

Two variants from same basis: KEM (Niederreiter) / PKE (McEliece).

Following a long line of work from Baldi, Chiaraluce et al. (2007-onwards).

Variable number of blocks $n_0 = 2, 3, 4$.

Private key is made dense via secret matrix $Q \longrightarrow \approx$QC-MDPC.

Specialized "Q-decoder" provides better decoding performance.

Sizes comparable to BIKE.

Sample Parameters: Level 1

BIKE offers a noticeable tradeoff.

Sample Parameters: Level 1

BIKE offers a noticeable tradeoff.

BIKE parameters (bytes):

BIKE-#	p	w	t	PK Size	SK Size	Ciph Size
1	10,163	142	134	2,541	267	2,541
2	10,163	142	134	1,271	267	1,271
3	11,027	134	154	2,757	252	2,757

Sample Parameters: Level 1

BIKE offers a noticeable tradeoff.

BIKE parameters (bytes):

BIKE-#	p	w	t	PK Size	SK Size	Ciph Size
1	10,163	142	134	2,541	267	2,541
2	10,163	142	134	1,271	267	1,271
3	11,027	134	154	2,757	252	2,757

Below we present LEDAkem for ease of comparison.

Sample Parameters: Level 1

BIKE offers a noticeable tradeoff.

BIKE parameters (bytes):

BIKE-#	p	w	t	PK Size	SK Size	Ciph Size
1	10,163	142	134	2,541	267	2,541
2	10,163	142	134	1,271	267	1,271
3	11,027	134	154	2,757	252	2,757

Below we present LEDAkem for ease of comparison.

LEDAkem parameters (bytes):

n_0	p	w	t	PK Size	SK Size	Ciph Size
2	15,013	9	143	1,880	468	1,880
3	9,643	13	90	2,416	604	1,208
4	8,467	11	72	3,192	716	1,064

DECODING FAILURES ARE BAD!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Decoding Failures are Bad!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Decoding Failures are Bad!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Solution: use ephemeral keys.

Decoding Failures are Bad!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Solution: use ephemeral keys.

Problem 2: IND-CCA security.

DECODING FAILURES ARE BAD!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Solution: use ephemeral keys.

Problem 2: IND-CCA security.

IND-CCA conversions require perfect correctness or at least trivial DFR ($\approx 2^{-128}$).

Decoding Failures are Bad!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Solution: use ephemeral keys.

Problem 2: IND-CCA security.

IND-CCA conversions require perfect correctness or at least trivial DFR ($\approx 2^{-128}$).

Decoding algorithms have (currently) DFR around 10^{-7} to 10^{-9}.

Decoding Failures are Bad!

Problem 1: reaction attacks (Guo, Johansson, Stankovski, '16).

Observe decryption of several ($\approx$ 300 million) ciphertexts: analyze decoding failures to reconstruct private key (distance spectrum).

Solution: use ephemeral keys.

Problem 2: IND-CCA security.

IND-CCA conversions require perfect correctness or at least trivial DFR ($\approx 2^{-128}$).

Decoding algorithms have (currently) DFR around 10^{-7} to 10^{-9}.

Solution: all variants only claim IND-CPA security.

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

5 out of 7 code-based NIST submissions in Round 2 use QC structure.

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

5 out of 7 code-based NIST submissions in Round 2 use QC structure.

- BIKE
- Classic McEliece
- HQC
- LEDAcrypt
- NTS-KEM
- ROLLO
- RQC

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

5 out of 7 code-based NIST submissions in Round 2 use QC structure.

Is there any other structure we can use? Can we generalize this, do it better/differently?

NEW DEVELOPMENTS

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

5 out of 7 code-based NIST submissions in Round 2 use QC structure.

Is there any other structure we can use? Can we generalize this, do it better/differently?

Use alternative Reproducible Codes (Santini, P., Baldi, '18).

New Developments

New, improved BIKE decoder (Sendrier, Vasseur, '19).

Possible to adjust block length to achieve desired DFR.

BIKE will feature IND-CCA version with static keys in Round 2.

5 out of 7 code-based NIST submissions in Round 2 use QC structure.

Is there any other structure we can use? Can we generalize this, do it better/differently?

Use alternative Reproducible Codes (Santini, P., Baldi, '18).

Can possibly negate DOOM speedup and reaction attacks.

Part III

Rank Metric

Rank Metric

One of alternative metrics used in Coding Theory.

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).

$d_R(x, y) = wt_R(x - y)$.

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).

$d_R(x, y) = wt_R(x - y)$.

So rank metric codes are matrix codes.

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.
$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).
$d_R(x, y) = wt_R(x - y)$.

So rank metric codes are matrix codes.

$[n, k]$ rank metric linear code over $\mathbb{F}_{q^m}$

A subspace of dimension k of $\mathbb{F}_{q^m}^n$ (Gabidulin, '85).

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).

$d_R(x, y) = wt_R(x - y)$.

So rank metric codes are matrix codes.

$[n, k]$ rank metric linear code over $\mathbb{F}_{q^m}$

A subspace of dimension k of $\mathbb{F}_{q^m}^n$ (Gabidulin, '85).

A subspace of dimension k of $\mathbb{F}_q^{m \times n}$ (Delsarte, '78).

Rank Metric

One of alternative metrics used in Coding Theory.

Rank metric

Let $x \in \mathbb{F}_{q^m}^n$ and $\beta = (\beta_1, \ldots, \beta_m)$ basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$wt_R(x) = Rank(\phi_\beta(x))$, where ϕ_β is projection over $\mathbb{F}_q$ (columns).

$d_R(x, y) = wt_R(x - y)$.

So rank metric codes are matrix codes.

$[n, k]$ rank metric linear code over $\mathbb{F}_{q^m}$

A subspace of dimension k of $\mathbb{F}_{q^m}^n$ (Gabidulin, '85).

A subspace of dimension k of $\mathbb{F}_q^{m \times n}$ (Delsarte, '78).

Support of a word

$Supp(x) = span < x_1, \ldots, x_n >_{\mathbb{F}_q}$.

Possible to “translate” many concepts from Hamming metric.

Possible to “translate” many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Few families with efficient decoding algorithm.

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Few families with efficient decoding algorithm.

- Gabidulin codes: $\approx$Reed-Solomon.

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Few families with efficient decoding algorithm.

- Gabidulin codes: $\approx$Reed-Solomon.
- Low-Rank Parity-Check codes (LRPC): $\approx$LDPC.

The Rank Metric World

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Few families with efficient decoding algorithm.

- Gabidulin codes: $\approx$Reed-Solomon.
- Low-Rank Parity-Check codes (LRPC): $\approx$LDPC.

Generic attack: rank equivalent of ISD, combinatorial (Chabaud, Stern, '96).

The Rank Metric World

Possible to "translate" many concepts from Hamming metric.

- Singleton Bound on largest minimum distance (MRD codes).
- GV Bound on size of spheres.
- Syndrome Decoding Problem (RSD): proved to be NP-Hard.

Few families with efficient decoding algorithm.

- Gabidulin codes: $\approx$Reed-Solomon.
- Low-Rank Parity-Check codes (LRPC): $\approx$LDPC.

Generic attack: rank equivalent of ISD, combinatorial (Chabaud, Stern, '96).

Structural attacks exist (Gibson, '95, '96, Overbeck, '05, Debris-Alazard, Tillich, '18).

Case Study: NIST Submissions

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.
- LOCKER: PKE version of LAKE.

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, ≈BIKE-2.
- LOCKER: PKE version of LAKE.
- Rank-Ouroboros: rank version of Ouroboros (BIKE-3).

Case Study: NIST Submissions

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.
- LOCKER: PKE version of LAKE.
- Rank-Ouroboros: rank version of Ouroboros (BIKE-3).

RQC: based on random codes $\approx$HQC.

Case Study: NIST Submissions

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.
- LOCKER: PKE version of LAKE.
- Rank-Ouroboros: rank version of Ouroboros (BIKE-3).

RQC: based on random codes $\approx$HQC.

Advantage: higher attack complexity $\mathcal{O}((n-k)^3 m^3 q^{t\lceil \frac{(k+1)m}{n} \rceil - m})$.

Case Study: NIST Submissions

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.
- LOCKER: PKE version of LAKE.
- Rank-Ouroboros: rank version of Ouroboros (BIKE-3).

RQC: based on random codes $\approx$HQC.

Advantage: higher attack complexity $\mathcal{O}((n-k)^3 m^3 q^{t\lceil \frac{(k+1)m}{n} \rceil - m})$.

Choose much smaller parameters, get smaller sizes.

Case Study: NIST Submissions

ROLLO: merge of 3 slightly different proposals on QC-LRPC codes.

- LAKE: rank-Niederreiter, $\approx$BIKE-2.
- LOCKER: PKE version of LAKE.
- Rank-Ouroboros: rank version of Ouroboros (BIKE-3).

RQC: based on random codes $\approx$HQC.

Advantage: higher attack complexity $\mathcal{O}((n-k)^3 m^3 q^{t\lceil\frac{(k+1)m}{n}\rceil - m})$.

Choose much smaller parameters, get smaller sizes.

No DFR for RQC.

Rank Metric Parameters

ROLLO: large amount of parameter sets, not easy to read through, some info missing. We chose here Rank-Ouroboros.

Rank Metric Parameters

ROLLO: large amount of parameter sets, not easy to read through, some info missing. We chose here Rank-Ouroboros.

Rank-Ouroboros parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	127	67	8	2,128	2,128	2,128	5
2	101	59	8	1,490	1,490	1,490	3
2	89	53	6	1,180	1,180	1,180	1

Rank Metric Parameters

ROLLO: large amount of parameter sets, not easy to read through, some info missing. We chose here Rank-Ouroboros.

Rank-Ouroboros parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	127	67	8	2,128	2,128	2,128	5
2	101	59	8	1,490	1,490	1,490	3
2	89	53	6	1,180	1,180	1,180	1

DFR for above parameters is still too low ($2^{-36}, 2^{-42}$) for e.g. IND-CCA security.

Rank Metric Parameters

ROLLO: large amount of parameter sets, not easy to read through, some info missing. We chose here Rank-Ouroboros.

Rank-Ouroboros parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	127	67	8	2,128	2,128	2,128	5
2	101	59	8	1,490	1,490	1,490	3
2	89	53	6	1,180	1,180	1,180	1

DFR for above parameters is still too low ($2^{-36}, 2^{-42}$) for e.g. IND-CCA security.

RQC parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	139	101	8	3,510	3,510	3,574	5
2	113	97	7	2,741	2,741	2,805	3
2	89	67	6	1,491	1,491	1,555	1

Rank Metric Parameters

ROLLO: large amount of parameter sets, not easy to read through, some info missing. We chose here Rank-Ouroboros.

Rank-Ouroboros parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	127	67	8	2,128	2,128	2,128	5
2	101	59	8	1,490	1,490	1,490	3
2	89	53	6	1,180	1,180	1,180	1

DFR for above parameters is still too low ($2^{-36}, 2^{-42}$) for e.g. IND-CCA security.

RQC parameters (bytes):

q	m	p	t	PK Size	SK Size	Ciph Size	Security
2	139	101	8	3,510	3,510	3,574	5
2	113	97	7	2,741	2,741	2,805	3
2	89	67	6	1,491	1,491	1,555	1

Sizes can be further compressed using seed expanders (also in other schemes).

CONSIDERATIONS

Sizes: very promising.

Considerations

Sizes: very promising.

Speed: a little behind other code-based schemes.

CONSIDERATIONS

Sizes: very promising.

Speed: a little behind other code-based schemes.

Cryptanalysis: a lot behind.

Considerations

Sizes: very promising.

Speed: a little behind other code-based schemes.

Cryptanalysis: a lot behind.

At least 25 publications on ISD and improvements (see Classic McEliece document).

CONSIDERATIONS

Sizes: very promising.

Speed: a little behind other code-based schemes.

Cryptanalysis: a lot behind.

At least 25 publications on ISD and improvements (see Classic McEliece document).

Only a handful on rank metric

(Ourivski, Johansson, '02, Gaborit, Ruatta, Schrek, '16, Aragon, Gaborit, Hauteville, Tillich, '18).

CONSIDERATIONS

Sizes: very promising.

Speed: a little behind other code-based schemes.

Cryptanalysis: a lot behind.

At least 25 publications on ISD and improvements (see Classic McEliece document).

Only a handful on rank metric
(Ourivski, Johansson, '02, Gaborit, Ruatta, Schrek, '16, Aragon, Gaborit, Hauteville, Tillich, '18).

Several aspects and details unclear or unexplored.

CONSIDERATIONS

Sizes: very promising.

Speed: a little behind other code-based schemes.

Cryptanalysis: a lot behind.

At least 25 publications on ISD and improvements (see Classic McEliece document).

Only a handful on rank metric
(Ourivski, Johansson, '02, Gaborit, Ruatta, Schrek, '16, Aragon, Gaborit, Hauteville, Tillich, '18).

Several aspects and details unclear or unexplored.

More investigation needed.

Part IV

Conclusions

CONCLUSIONS

Code-based cryptography is prominent candidate for standardization.

CONCLUSIONS

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Conclusions

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

Conclusions

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

Conclusions

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

- Conservative (binary Goppa, no structure)

CONCLUSIONS

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

- Conservative (binary Goppa, no structure)
- Sparse-matrix (LDPC/MDPC, QC structure...for now)

Conclusions

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

- Conservative (binary Goppa, no structure)
- Sparse-matrix (LDPC/MDPC, QC structure...for now)
- Rank metric (LRPC, QC structure)

CONCLUSIONS

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

- Conservative (binary Goppa, no structure)
- Sparse-matrix (LDPC/MDPC, QC structure...for now)
- Rank metric (LRPC, QC structure)

HQC/RQC: theoretical security advantage (CCA).

CONCLUSIONS

Code-based cryptography is prominent candidate for standardization.

Several distinctive strengths (and few well-known drawbacks).

Suitable for KEM: key exchange + encryption.

NIST has identified three macro-areas, each with their own pros/cons:

- Conservative (binary Goppa, no structure)
- Sparse-matrix (LDPC/MDPC, QC structure...for now)
- Rank metric (LRPC, QC structure)

HQC/RQC: theoretical security advantage (CCA).

Round 2: protocol refinements, re-parametrizations, new/improved implementations.

Follow the NIST Competition

FAU has been funded by NIST for PQC project.

Follow the NIST Competition

FAU has been funded by NIST for PQC project.

Detailed competition wiki/database.

Follow the NIST Competition

FAU has been funded by NIST for PQC project.

Detailed competition wiki/database.

Will include parameters, sizes, security assumptions etc. + challenges.

Follow the NIST Competition

FAU has been funded by NIST for PQC project.

Detailed competition wiki/database.

Will include parameters, sizes, security assumptions etc. + challenges.

“Living” resource with external contributions.

Follow the NIST Competition

FAU has been funded by NIST for PQC project.

Detailed competition wiki/database.

Will include parameters, sizes, security assumptions etc. + challenges.

"Living" resource with external contributions.

Work in progress, first draft nearly ready - stay tuned!

Thank you