Introduction Applications in Cryptography

Generalized Sparse Matrices and Applications to Decoding and Cryptography

Maxime BROS

University of Limoges (France) XLIM Research Institute, UMR 7252

maxime bros@etu unilim fr

May 23, 2019







2 Decoding codes with sparse parity check matrix

Applications in Cryptography

Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Reminder about Error Correcting Codes

Error correcting codes are used to transmit informations (satellites, DVD, telecommunications, ...) but also for cryptographic purpose.

Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Reminder about Error Correcting Codes

Error correcting codes are used to transmit informations (satellites, DVD, telecommunications, ...) but also for cryptographic purpose.

Code (definition)

A code C is vector space of $GF(q)^n$ of dimension k.

Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Reminder about Error Correcting Codes

Error correcting codes are used to transmit informations (satellites, DVD, telecommunications, ...) but also for cryptographic purpose.

Code (definition)

A code C is vector space of $GF(q)^n$ of dimension k.

$$\begin{array}{cccc} \mathcal{E} \colon GF(q)^k & \longrightarrow & GF(q)^n \\ m & \longmapsto & mG \end{array}$$



Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Reminder about Error Correcting Codes

Parity Check Matrix

H is a parity check matrix for the code C if for every word $c \in GF(q)^n$:

$$c \in \mathcal{C} \iff Hc^T = 0_{n-k}$$

Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Hard Problems in Coding Theory

Decoding Problem (computational)

- Let G be a matrix k × n over a field K, y a vector of length n (with coefficients in K) and ω ∈ N.
- Find m∈ K^k such that weight(y − mG) for a given metric is smaller or equal to ω.

Syndrome Decoding Problem (computational)

Decoding codes with sparse parity check matrix Applications in Cryptography Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Hard Problems in Coding Theory

Decoding Problem (computational)

- Let G be a matrix k × n over a field K, y a vector of length n (with coefficients in K) and ω ∈ N.
- Find m∈ K^k such that weight(y − mG) for a given metric is smaller or equal to ω.

Syndrome Decoding Problem (computational)

- Let H be a matrix (n − k) × n over a field K,
 s a vector of length (n − k) (with coefficients in K) and ω ∈ N.
- Find e ∈ Kⁿ with weight smaller or equal to ω for a given metric such that He^t = s ?

Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Hard Problems in Coding Theory

• These 2 problems are equivalent.

Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Hard Problems in Coding Theory

• These 2 problems are equivalent.

• Proven NP-complete with Hamming metric in 1978 by Berlekamp, McEliece and Tilborg.

Reminder about Error Correcting Codes Decoding and Syndrome Decoding Problems

Hard Problems in Coding Theory

• These 2 problems are equivalent.

- Proven NP-complete with Hamming metric in 1978 by Berlekamp, McEliece and Tilborg.
- Proven to be probabilistically NP-complete with rank metric in 2017 by Gaborit and Zémor.

MDPC Rank Metric LRPC

Decoding codes with sparse parity check matrix

• We will study two codes (MDPC and LRPC) for which one uses the sparsity of their **parity check matrix** to decode.

MDPC Rank Metric LRPC

Decoding codes with sparse parity check matrix

- We will study two codes (MDPC and LRPC) for which one uses the sparsity of their **parity check matrix** to decode.
- The notion of sparsity one uses depends on the chosen metric.



• For MDPC codes sparse means : majority of zeros in the matrix (only $\mathcal{O}(\sqrt{n})$ 1's per row)



• For MDPC codes sparse means : majority of zeros in the matrix (only $\mathcal{O}(\sqrt{n})$ 1's per row)

MDPC

• Only H is sparse \implies "Moderate Density Parity Check".

Gallager's algorithm

Reminder : only the error contributes to the syndrome.



Decoding codes with sparse parity check matrix

MDPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{=0} + He^T$

(1010000000100001100011001) 11010000000000000001100 0110100000001000001100 00110100000001000000110 0001101000000000000000000000000000000	
--	--

 $\int_{y_{20}} /$

MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc}_{-0}^T + He^T$



#(common 1's) =

MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{=0} + He^T$

	$\begin{pmatrix} y_1 \end{pmatrix}$	
	<u>У</u> 2	
(1 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 1)	<i>y</i> 3	
	<i>y</i> 4	(
11010000000000001100	<i>y</i> 5	
01101000000100000110	V6	
	V7	
000110100000000000001	Vo	
00001101000101000000	Vo	=
00000110100110100000	<i>y</i> 9 <i>V</i> 40	
0000011010010010000	<i>y</i> 10	
0000001101001001000	<i>y</i> 11	
1000000110001100100	<i>y</i> 12	
01000000011000110010/	<i>Y</i> 13	\
	<i>Y</i> 14	`
	\y ₂₀ /	

 $\#(\text{common 1's}) = \{1$

0

MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc}_{=0}^T + He^T$

	/ / 1	
	<i>У</i> 2	
/1010000001000011001	<i>y</i> 3	
11010000000000001100	<i>y</i> 4	
0110100000010000110	95 V6	
00110100000010000011	У0 У7	
	<i>y</i> 8	_
000001101000100000000	<i>Y</i> 9	_
00000011010010010000	<i>Y</i> 10	
0000001101001001000	<i>y</i> 11	
1000000110001100100	912 V13	
\010000011000110010/	<i>Y</i> 14	
	\ _{y20} /	

 $\#(\text{common 1's}) = \{1, 1\}$

 (V_1)

MDPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{=0} + He^T$

$ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &$	y2 y3 y4 y5 y5 y7 y7 y7 y7 y7 y10 y11 y12 y13 y14 	=	<pre>(0 1 0 1 1 1 0 1 0 0</pre>
	y ₂₀ /		

 $\#(\text{common 1's}) = \{1, 1, 1\}$

/ V1 \

MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc}_{-0}^T + He^T$



MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc}_{-0}^T + He^T$



MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{-0} + He^T$



MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{} + He^T$ y2 y3 y4+1 y5 y6 y7 0 1 1 1 0 1 *y*8 = *y*9 *Y*10 *Y*11 1 У12 У13 У14 0 $\#(\text{common 1's}) = \{1, 1, 1, 3, 2, 2, 2, 3, 1, 1, 1, 1, 2, 2, 0, 1, 2, 1, 1, 2\}$ Let $\tau = 3$ be our threshold.

MDPC Rank Metric LRPC

Gallager's algorithm

Reminder : only the error contributes to the syndrome. More precisely $Hy^T = \underbrace{Hc^T}_{=0} + He^T$



MDPC Rank Metric LRPC

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

MDPC Rank Metrie LRPC

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

• Compute the syndrome $s = Hy^T$

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

- Compute the syndrome $s = Hy^T$
- For each of the n columns of H, count the number of common 1's between the syndrome and this column

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

- Compute the syndrome $s = Hy^T$
- For each of the n columns of H, count the number of common 1's between the syndrome and this column

MDPC

(a) For a given column *i*, if this number of common 1's is greater than a threshold τ , then change the *i*th bit of *y*

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

- Compute the syndrome $s = Hy^T$
- For each of the n columns of H, count the number of common 1's between the syndrome and this column

- **(a)** For a given column *i*, if this number of common 1's is greater than a threshold τ , then change the *i*th bit of *y*
- Gall this new vector y again, and go back to the first step, until either

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

- Compute the syndrome $s = Hy^T$
- For each of the n columns of H, count the number of common 1's between the syndrome and this column

- **(a)** For a given column *i*, if this number of common 1's is greater than a threshold τ , then change the *i*th bit of *y*
- Gall this new vector y again, and go back to the first step, until either
 - $s = 0_{n-k}$ \implies RETURN the last y, which is the codeword

Gallager's algorithm

One receives a word y := mG + e = codeword + error.

Bit-flipping algorithm

- Compute the syndrome $s = Hy^T$
- For each of the n columns of H, count the number of common 1's between the syndrome and this column

- **(a)** For a given column *i*, if this number of common 1's is greater than a threshold τ , then change the *i*th bit of *y*
- Call this new vector y again, and go back to the first step, until either
 - $s = 0_{n-k}$ \implies **RETURN** the last *y*, which is the codeword
 - a certain number
 of iterations is reached ⇒ RETURN FAIL

MDPC Rank Metric

Rank Decoding Problem

• In 1985, Gabidulin, a Russian researcher, introduced rank-codes over an extension field

(the use of rank metric started in 1951 by Hua and then in 1978 by Delsarte who introduced rank distance for matrix-codes).

MDPC Rank Metric

Rank Decoding Problem

• In 1985, Gabidulin, a Russian researcher, introduced rank-codes over an extension field

(the use of rank metric started in 1951 by Hua and then in 1978 by Delsarte who introduced rank distance for matrix-codes).



Figure: Ernst Gabidulin (1937 (U.S.S.R) - aged 81 today)

MDPC Rank Metric

Rank Decoding Problem

• In 1985, Gabidulin, a Russian researcher, introduced rank-codes over an extension field

(the use of rank metric started in 1951 by Hua and then in 1978 by Delsarte who introduced rank distance for matrix-codes).

• Using this metric instead of the Hamming metric :



Figure: Ernst Gabidulin (1937 (U.S.S.R) - aged 81 today)
MDPC Rank Metric

Rank Decoding Problem

• In 1985, Gabidulin, a Russian researcher, introduced rank-codes over an extension field

(the use of rank metric started in 1951 by Hua and then in 1978 by Delsarte who introduced rank distance for matrix-codes).

- Using this metric instead of the Hamming metric :
 - Decoding Problem \implies Rank Decoding Problem (RD)



Figure: Ernst Gabidulin (1937 (U.S.S.R) - aged 81 today)

MDPC Rank Metric LRPC

Rank Decoding Problem

• In 1985, Gabidulin, a Russian researcher, introduced rank-codes over an extension field

(the use of rank metric started in 1951 by Hua and then in 1978 by Delsarte who introduced rank distance for matrix-codes).

- Using this metric instead of the Hamming metric :
 - Decoding Problem \implies Rank Decoding Problem (RD)
 - Syndrome Decoding Problem \implies Rank Syndrome Decoding Pb. (RSD)



Figure: Ernst Gabidulin (1937 (U.S.S.R) - aged 81 today)

MDPC Rank Metric

Reminder about finite fields



MDPC Rank Metric

Reminder about finite fields



MDPC Rank Metric LRPC

Rank metric

$$m = 4$$
, and let $e \in GF(2^4)^4$

MDPC Rank Metric LRPC

Rank metric

$$m = 4$$
, and let $e \in GF(2^4)^4$

$$e = (1 + \alpha^2, \alpha, 1, \alpha^2)$$
$$M = \frac{1}{\alpha^2} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

MDPC Rank Metric LRPC

Rank metric

$$m = 4$$
, and let $e \in GF(2^4)^4$

$$e = (1 + \alpha^2, \alpha, 1, \alpha^2)$$
$$M = \frac{1}{\alpha^2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

MDPC Rank Metric LRPC

Rank metric

$$m=$$
 4, and let $e\in GF(2^4)^4$

$$e = (1 + \alpha^2, \alpha, 1, \alpha^2)$$
$$M = \frac{1}{\alpha^2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

e can be represented by a matrix M of GF(2)

MDPC Rank Metric LRPC

Rank metric

$$m=$$
 4, and let $e\in GF(2^4)^4$

$$e = (1 + \alpha^2, \alpha, 1, \alpha^2)$$
$$M = \frac{1}{\alpha^2} \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

e can be represented by a matrix M of GF(2)We define Rank(e) = Rank(M) = 3

MDPC Rank Metric LRPC

Rank metric

Notation : all presentation long, q is a power of a prime p and $m, n \in \mathbb{N}^*$.

Rank of a $GF(q^m)^n$ word

Let
$$e = (e_1, e_2, \ldots, e_n)$$
 be a vector of $GF(q^m)^n$.
Given a basis $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_m)$ of $GF(q^m)$ over $GF(q)$, one gets :
 $\forall i \in \{1, \ldots, n\}, \exists ! \ (e_{i,1}, e_{i,2}, \ldots, e_{i,m}) \in GF(q)^m$ such that
 $e_i = \sum_{j=1}^m e_{i,j} \cdot \mathcal{B}_j$.
The rank of e , noted $Rank(e)$, is defined by the rank of the matrix
 $(e_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}}$.

MDPC Rank Metric LRPC

Rank metric

Notation : all presentation long, q is a power of a prime p and $m, n \in \mathbb{N}^*$.

Rank of a $GF(q^m)^n$ word

Let
$$e = (e_1, e_2, \ldots, e_n)$$
 be a vector of $GF(q^m)^n$.
Given a basis $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_m)$ of $GF(q^m)$ over $GF(q)$, one gets :
 $\forall i \in \{1, \ldots, n\}, \exists ! \ (e_{i,1}, e_{i,2}, \ldots, e_{i,m}) \in GF(q)^m$ such that
 $e_i = \sum_{j=1}^m e_{i,j} \cdot \mathcal{B}_j$.
The rank of e , noted $Rank(e)$, is defined by the rank of the matrix
 $(e_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}}$.

Rank distance

$$egin{array}{rcl} d:& {\it GF}(q^m)^n imes {\it GF}(q^m)^n &
ightarrow \mathbb{N} \ & (e_1,e_2) & \mapsto & {\it Rank}(e_1-e_2) \end{array}$$

is a distance (or a metric) over $GF(q^m)^n$.

MDPC Rank Metric LRPC

Code properties in Rank metric

What are the differences between code with Hamming and Rank metric ?

Minimal distance of a code with rank metric

Let C be a [n, k, d]-code over $GF(q^m)$ for the rank metric As long as C is additive :

 $d = min\{Rank(c) \mid c \in C\}$

MDPC Rank Metric LRPC

Code properties in Rank metric

What are the differences between code with Hamming and Rank metric ?

Minimal distance of a code with rank metric

Let C be a [n, k, d]-code over $GF(q^m)$ for the rank metric As long as C is additive :

$$d = min\{Rank(c) \mid c \in C\}$$

Support of a word in C

The support of an element $x = (x_1, ..., x_n) \in GF(q^m)^n$ is the GF(q)-vector space of $GF(q)^m$ generated by the coordinates x_i of x.

Support of a word

е

$$= (1 + \alpha^{2}, \alpha, 1, \alpha^{2}) \in GF(2^{4})^{4} \text{ and } Rank(e) = 3$$

$$support(e) = < \begin{pmatrix} 1\\0\\1\\0 \end{pmatrix}, \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix}, \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix} >$$

$$= < \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix} >$$

Rank Metric

Support of a word

$$e = (1 + \alpha^{2}, \alpha, 1, \alpha^{2}) \in GF(2^{4})^{4} \text{ and } Rank(e) = 3$$

$$support(e) = < \begin{pmatrix} 1\\0\\1\\0 \end{pmatrix}, \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix}, \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix} >$$

$$= < \begin{pmatrix} 1\\0\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\1\\0\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix}, \begin{pmatrix} 0\\0\\1\\0 \end{pmatrix} >$$

Rank Metric

support(e) is a vector space E of $GF(2)^4$ of dimension 3

MDPC Rank Metric LRPC

• In this context, rank metric, sparse means with values in a vector space of small dimension *d*, as seen before it uses the generalization of the support to rank metric.

- In this context, rank metric, sparse means with values in a vector space of small dimension *d*, as seen before it uses the generalization of the support to rank metric.
- The sparse matrix is still $H \Longrightarrow$ "Low Rank Parity Check".

Decoding LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

• Let *H* be the parity check matrix of an LRPC code, with coefficients in $F \subset GF(2)^N$ of dimension *d*.

Decoding LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

- Let *H* be the parity check matrix of an LRPC code, with coefficients in $F \subset GF(2)^N$ of dimension *d*.
- Let e be the error, with coefficients in $E \subset GF(2)^N$ of dimension r. i.e supp(e) = E.

Decoding LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

- Let *H* be the parity check matrix of an LRPC code, with coefficients in $F \subset GF(2)^N$ of dimension *d*.
- Let e be the error, with coefficients in $E \subset GF(2)^N$ of dimension r. i.e supp(e) = E.
- From the syndrome $s := Hy^T = He^T$, one gets the vector space $S := < s_1, \ldots, s_{n-k} >$

Decoding LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

- Let *H* be the parity check matrix of an LRPC code, with coefficients in $F \subset GF(2)^N$ of dimension *d*.
- Let e be the error, with coefficients in $E \subset GF(2)^N$ of dimension r. i.e supp(e) = E.
- From the syndrome $s := Hy^T = He^T$, one gets the vector space $S := < s_1, \ldots, s_{n-k} >$
- This vector space is a subset of the product space < *EF* > of dimension at most *rd*.

Decoding LRPC

As for the Hamming metric, the knowledge of **the support** of the error is enough to decode.

- Let *H* be the parity check matrix of an LRPC code, with coefficients in $F \subset GF(2)^N$ of dimension *d*.
- Let e be the error, with coefficients in $E \subset GF(2)^N$ of dimension r. i.e supp(e) = E.
- From the syndrome $s := Hy^T = He^T$, one gets the vector space $S := < s_1, \ldots, s_{n-k} >$
- This vector space is a subset of the product space < *EF* > of dimension at most *rd*.
- Since rd < (n k), it is very likely that $S = \langle EF \rangle$.

Decoding LRPC

Let F_i be the *i*th element of a basis of F, let's consider the vector space

LRPC

 $F_i^{-1}S = F_i^{-1} < E_1F_1, E_2F_1, \dots E_rF_1, \\ E_1F_2, E_2F_2, \dots E_rF_2, \\ \dots \dots \dots \dots \\ E_1F_i, E_2F_i, \dots E_rF_i, \\ \dots \dots \dots \dots \\ E_1F_d, E_2F_d, \dots E_rF_d, > E_rF_d$

Decoding LRPC

Let F_i be the i^{th} element of a basis of F, let's considerer the vector space

LRPC

 $F_{i}^{-1}S =$

$$< F_{i}^{-1}E_{1}F_{1}, F_{i}^{-1}E_{2}F_{1}, \dots F_{i}^{-1}E_{r}F_{1}, \\F_{i}^{-1}E_{1}F_{2}, F_{i}^{-1}E_{2}F_{2}, \dots F_{i}^{-1}E_{r}F_{2}, \\\dots & \dots & \dots \\F_{i}^{-1}E_{1}F_{i}, F_{i}^{-1}E_{2}F_{i}, \dots F_{i}^{-1}E_{r}F_{i}, \\\dots & \dots & \dots \\F_{i}^{-1}E_{1}F_{d}, F_{i}^{-1}E_{2}F_{d}, \dots F_{i}^{-1}E_{r}F_{d}, >$$

Decoding LRPC

Let F_i be the i^{th} element of a basis of F, let's considerer the vector space

LRPC

 $F_i^{-1}S =$

Decoding LRPC

Let F_i be the i^{th} element of a basis of F, let's considerer the vector space

I RPC

 $F_{i}^{-1}S = \begin{cases} < F_{i}^{-1}E_{1}F_{1}, & F_{i}^{-1}E_{2}F_{1}, & \dots & F_{i}^{-1}E_{r}F_{1}, \\ F_{i}^{-1}E_{1}F_{2}, & F_{i}^{-1}E_{2}F_{2}, & \dots & F_{i}^{-1}E_{r}F_{2}, \\ \dots & \dots & \dots & \dots \\ E_{1}, & E_{2}, & \dots & E_{r}, \\ \dots & \dots & \dots & \dots \\ F_{i}^{-1}E_{1}F_{d}, & F_{i}^{-1}E_{2}F_{d}, & \dots & F_{i}^{-1}E_{r}F_{d}, \end{cases} >$

So we have $E \subset F_i^{-1}S$, $\forall i \in \{1, \ldots, d\}$.

LRPC

Decoding LRPC

• $E \subset F_i^{-1}S := S_i \quad \forall i \in \{1, \ldots, d\}$

Decoding LRPC

- $E \subset F_i^{-1}S := S_i \quad \forall i \in \{1, \ldots, d\}$
- The co-space of E in $F_i^{-1}S$ is very likely to be different for $i' \neq i$.

LRPC

Decoding LRPC

- $E \subset F_i^{-1}S := S_i \quad \forall i \in \{1, \ldots, d\}$
- The co-space of E in $F_i^{-1}S$ is very likely to be different for $i' \neq i$.
- And so the key to the decoding algorithm of LRPC is that (with high probability)

$$\bigcap_{i=1}^d S_i = E$$

MDPC Rank Metrie LRPC

Decoding LRPC

- $E \subset F_i^{-1}S := S_i \quad \forall i \in \{1, \ldots, d\}$
- The co-space of E in $F_i^{-1}S$ is very likely to be different for $i' \neq i$.
- And so the key to the decoding algorithm of LRPC is that (with high probability)

$$\bigcap_{i=1}^d S_i = E$$

• Knowing the support of the error, one has to solve a linear system with more equations than unknowns to find the error and so to decode *y*.

Decoding LRPC

• This decoding is probabilistic, but the failure probability decreases exponentially according to the parameters.

LRPC

Decoding LRPC

- This decoding is probabilistic, but the failure probability decreases exponentially according to the parameters.
- For instance, with parameters [n = 94, k = 47, N = 47, d = r = 5] for a binary LRPC, $\mathbb{P}(failure) = 2^{-23}$.

iding the structure Ngebraic attacks in the random case

Asymmetric Cryptography

Asymmetric Cryptography (or Public Key Cryptography)



Hiding the structure Algebraic attacks in the random case

Asymmetric Cryptography







Hiding the structure Algebraic attacks in the random case

Asymmetric Cryptography






Hiding the structure Algebraic attacks in the random case







Hiding the structure Algebraic attacks in the random case







Hiding the structure Algebraic attacks in the random case







Hiding the structure Algebraic attacks in the random case







Hiding the structure Algebraic attacks in the random case







Hiding the structure Algebraic attacks in the random case

McEliece cryptosystem

Robert McEliece (1942 - 2019) started code-based cryptography in 1978.

Hiding the structure Algebraic attacks in the random case

McEliece cryptosystem

Robert McEliece (1942 - 2019) started code-based cryptography in 1978.



McEliece cryptosystem

McEliece cryptosystem (or setting)

Let C be a linear code [n, k, d]-code, decodable up to t = (d - 1)/2 errors in polynomial time with an algorithm D.

Private Key: the generator matrix G of C (usually in systematic form), S a non-singular $k \times k$ matrix and P a $n \times n$ permutation matrix. **Public Key**: G' = SGP

Encryption : let *m* be the plaintext (of length *k*), the cipher text is y = mG' + e with $w(e) \le t$ (*e* random). **Decryption** :

- Compute $yP^{-1} = (mS)G + eP^{-1}$
- Decode $mSG = \mathcal{D}(yP^{-1})$
- Recover mS and then m (using S^{-1})

Hiding the structure Algebraic attacks in the random case

Application to Cryptography : hiding the structure

- Attack : recovering the plaintext
 - \implies Decoding Random Code Problem under the assumption that
 - \mathcal{C}^\prime is indistinguishable from a random code.

Hiding the structure Algebraic attacks in the random case

Application to Cryptography : hiding the structure

- Attack : recovering the plaintext
 - \Longrightarrow Decoding Random Code Problem under the assumption that \mathcal{C}' is indistinguishable from a random code.
- This is not the case if C has a "strong" algebraic structure (Reed-Solomon or Gabidulin for instance).

Hiding the structure Algebraic attacks in the random case

Application to Cryptography : hiding the structure

- Attack : recovering the plaintext
 - \Longrightarrow Decoding Random Code Problem under the assumption that \mathcal{C}' is indistinguishable from a random code.
- This is not the case if C has a "strong" algebraic structure (Reed-Solomon or Gabidulin for instance).
- One wants to get rid of the "scrambling" step.

MDPC cryptosystem

New variant of McEliece cryptosystem [MTSB13] et [GMRZ13]

Let C be an **MDPC** [n, k, d]-code, decodable up to t = (d - 1)/2 errors in polynomial time with a **probabilistic** decoding algorithm \mathcal{D}_H (using its sparse parity check matrix $H = (h_0|h_1)$).

Private Key: the parity check matrix H of C**Public Key**: the generator matrix $G = (I|(h_0h_1^{-1})^T)$ of C (systematic form).

Encryption : let *m* be the plaintext (of length *k*), the cipher text is y = mG + e with $w(e) \le t$ (*e* random). **Decryption** :

- Decode $mG = \mathcal{D}_H(y)$
- Recover *m* (extracting the first *k* components of *mG*)

MDPC cryptosystem

New variant of McEliece cryptosystem [MTSB13] et [GMRZ13]

Let C be an **MDPC** [n, k, d]-code, decodable up to t = (d - 1)/2 errors in polynomial time with a **probabilistic** decoding algorithm \mathcal{D}_H (using its sparse parity check matrix $H = (h_0|h_1)$).

Private Key: the parity check matrix H of C**Public Key**: the generator matrix $G = (I|(h_0h_1^{-1})^T)$ of C (systematic form).

Encryption : let *m* be the plaintext (of length *k*), the cipher text is y = mG + e with $w(e) \le t$ (*e* random). **Decryption** :

- Decode $mG = \mathcal{D}_H(y)$
- Recover *m* (extracting the first *k* components of *mG*)

Note : LRPC cryptosystem described in 2013 by Gaborit, Murat, Ruatta and Zémor is similar but uses rank metric and the LRPC decoding previously described.

MDPC cryptosystem

New variant of McEliece cryptosystem [MTSB13] et [GMRZ13]

Let C be an **MDPC** [n, k, d]-code, decodable up to t = (d - 1)/2 errors in polynomial time with a **probabilistic** decoding algorithm \mathcal{D}_H (using its sparse parity check matrix $H = (h_0|h_1)$).

Private Key: the parity check matrix H of C**Public Key**: the generator matrix $G = (I|(h_0 h_1^{-1})^T)$ of C (systematic form).

Encryption : let *m* be the plaintext (of length *k*), the cipher text is y = mG + e with $w(e) \le t$ (*e* random). **Decryption** :

- Decode $mG = \mathcal{D}_H(y)$
- Recover *m* (extracting the first *k* components of *mG*)

Note : LRPC cryptosystem described in 2013 by Gaborit, Murat, Ruatta and Zémor is similar but uses rank metric and the LRPC decoding previously described.

Disclaimer : this system has to be modified to be used in practice, since the systematic form would lead to leaks of the original message.

Hiding the structure

Hiding the structure Algebraic attacks in the random case

Advantages of MDPC and LRPC :

Hiding the structure

Advantages of MDPC and LRPC :

• Almost no algebraic structure.

Thus the security doesn't rely on masking the structure but only on the hardness of decoding random codes **under the assumption of indistinguability of MDPC and LRPC**.

Hiding the structure

Hiding the structure

Advantages of MDPC and LRPC :

• Almost no algebraic structure.

Thus the security doesn't rely on masking the structure but only on the hardness of decoding random codes **under the assumption of indistinguability of MDPC and LRPC**.

Hiding the structure

• Thus, the "scrambling" step is replaced by going from the **sparse** *H* **to the dense** *G*.

Hiding the structure Algebraic attacks in the random case

Hiding the structure

Few visual examples : *M* of size 20×20 , coefficients in $GF(2^{20}) \approx 1$ million elements, dim(F) = 2.

neneesseenseesse Nentrier Ma																			
[w*362254	9	w^362254	w^362254	e	w*362254	w^671186	w^362254	w*362254	9	9	w^362254	w*362254	9	w^362254	#^733998	w*733998	w^671186	8	61
W*362254	w*733998	w^471186	0	w*733998	w*362254	w^362254	w^362254	w*471186	w*733998	ė	w^362254	w*733998	w*362254	w^733990	₩^471186	w*471186	9	w^362254	w^471186]
6 3	9	w^471186	w^733998	0	w^733998	w^733998	0	w*733998	w*471186	0	w^362254	9	w*471186	0	w^362254	w*733998	w^733998	w^471186	01
[w^471186	w*471186	w^733998	w^471186	w*362254	w*733998	w^362254	ē	w*471186	8	w^733998	w^362254	w*471186	8	w^733990	₩^471186	w*471186	8	w^362254	w^3622541
6 3	w*362254	w^471186	0	w*471186	w*733998	w^733998	w^362254	w*471186	w*362254	w^471186	0	w*471186	e	w^362254	0	w*733998	w*471186	w^733998	w^3622541
[w*362254	w*362254	9	w^471186	w*733998	9	w^362254	w^471186	w*733998	w*471186	9	w^733998	9	w^471186	w^733998	9	w*733998	9	w^733998	w^733990]
[w*362254	w*733998	9	w^471186	w*471186	9	w^733998	*733990	w*362254	w*471106	w^733990	9	w*362254	w*471186	w^362254	w^36225 4	w*471186	w^733998	9	w^362254]
[w*733990	w*362254	9	w^471186	6	9	9	w^471186	w*471186	9	9	w^733998	w*733998	w*471186	w^362254	9	w*471186	9	9	w^362254]
[w*733990	9	w^733998	w^471186	9	9	w^471106	w^471186	9	9	w^362254	■^36225 4	w*471186	w*362254	9	w^36225 4	9	9	w^471106	w^733990]
E 0	w^733998	w^471186	8	w*362254	w*471186	9	8	6	9	9	■^471186	w*471186	9	9	w^362254	w*362254	w^733998	8	6]
[w*471186	9	w^471186	w^733990	w*471186	w*733998	w^733998	w^471186	w*362254	w*733998	9	w^733990	w*733998	9	w^733998	9	w*471186	9	w^471106	0]
[w*471106	w*733998	w^733998	w^733998	w*733998	w*362254	w^362254	w^471186	w*733998	w*362254	w^362254	0	w*733998	9	w^733990	₩^36225 4	w*733998	w^471186	w^362254	w^733990]
(0	w^733998	w^471186	w^362254	w*733990	w*471186	w^733990	9	w*471186	w*471186	w^733990	9	9	w^362254	w^362254	w^733990	w*733998	9	w^471186	w^362254]
E 0	w*471106	9	w^471186	w*733990	9	w^362254	w^362254	w*733990	9	w^471186	*733998	w*471106	w*362254	w^471186	₩^471186	9	w^733998	w^362254	w^7339901
[w^471186	w*471186	w^471186	8	w*471186	w^471186	w^471186	w^471186	w*733998	w^733998	w^362254	w^471186	w*733998	w*471186	w^362254	w^733990	9	w^362254	w^362254	w^362254]
[w*471186	w*471106	w^733990	m^733990	w*471186	w*362254	w^471106	w^471186	w*362254	w*733998	w^362254	■^733990	w*362254	w*362254	9	w^733990	w*471106	w*471106	9	w^3622541
[w*362254	w*362254	w^362254	w^733998	w*471186	w*733998	9	w^471186	w*362254	w*471186	w^362254	w^733990	w*471186	w*471186	9	w^471186	w*471186	w^362254	w^471106	w^362254]
L 0	9	9	w^362264	0	w*471186	w^362254	w^471186	w*733998	9	w^362254	w^362254	w*362254	w*362254	w^733990	0	w*362254	w*471186	w^733990	w^733990]
[w*362254	w*733998	w^733998	9	6	9	w^733998	9	6	w^733998	w^362254	9	6	w*362254	9	9	w*733998	w^733998	w^733990	w^471186]
[w*733990	w*471186	9	w^733990	w*471186	w*471186	9	w^733990	w*362254	w*362254	9	w^733990	w*733990	w^733998	9	w^362254	w*733998	9	w^362254	w^362254J
*oaxooxxoox	eonnoon																		
Matrice MAL	-1)=																		
L W*86448	W-988123	W*863625	W-898785	W-139452	W-023210	W-594637	W*804/52	W-105343	W-28264	W-1645633	W*888/8/	w-61/636	W-265974	M T0A562	W-94/41	W-801/59	W-52989	W**00/008	#*000572]
[w*333957	***742554	w^11EA26	W-010724	-1000929	w-446446	W-704400	mA1(2991	-1040753	w 757233	w-768287	#*************************************	w-176259	w-350565	W-304173	W-037537	w1092779	w111491	W-774009	#**70(4591
[w1911167	w522150	W 110424	#^E47097	w1921522	w1221147	w 100307	#142001 #6449201	w1207180	w 033366	w0527210	# 039272	wA00700	w1602048	w^252626	#^412266	w152/16	w1551022	W 210700	#19759461
[w*311167	w-23150	wh(78103	W-567867	W-831522	w-22110/	w^111700/	m-408361	w-297189	W-764454	W-53/310	m*120237	m.99786	w-69590/	w^{(253634	W-691839	W-524141	w-551022	W-223229	W*875040]
[w19/1010	W 73772	W^127024	#A272201	w141722	********	wh269262	#^346945	w1262619	m112022	WAE0EE20	#A67792	w1614572	#1212000	wf100E909	#400924	w1526670	#A283208	W 270322	wf1051201
[w*500067	** 796729	wh225021	w^301100	w*116930	#166222	w1912786	= 344840 = 010120	w1994021	w1155916	w1600012	w15/5/00	w*159510	w1588020	m^705221	w*A05753	w*517264	w1028085	w^260061	wf6637061
[w*336628	w*692788	w^546219	#^362368	w*463793	w^163169	w*31191	#^596868	#*44884	w^933333	w^236622	#^977827	w*797738	w^777339	w^266168	#^826518	** 425298	w^916518	w^1818882	w^289761
[w*727887	w*537798	w^387187	#^622828	m^98354	w*175592	w^578483	#^758515	w*624998	w^1821246	w^715397	#*345528	w*915186	w*455894	w^868478	#*886395	w*913268	w*347761	w132665	w^991671
w*262635	w*384261	w^266488	w^587969	w*548688	w*268454	w^268662	#^767743	w*566267	w*281989	w^1889597	#*577284	w*549731	w^238258	w^522123	w^491763	w*994584	w^766994	w^652178	w^193461
[w*787261	w*787284	w^577188	#^944193	m^76889	w^72114	w^168346	#^188787	w1007068	w*78632	w^543358	#*734432	#^33828	w1617557	w^755685	#^123481	w*848974	w^717828	w^239878	#^4582731
w^653534	w*254962	w^613788	w*1829328	w*731945	w*737731	w^257473	#^482995	w*282381	w*918616	w^295557	w^889336	w*351942	w*388277	w^125269	w^188947	#^12813	w*69692	w^816486	w*18831681
w*828747	w*472469	w^375392	w^928977	w*812964	w^318578	w^966941	w^935842	w*1827964	w*619944	w^858971	w^296038	w*683987	w*423541	w^184752	₩^654787	w*379296	w^732234	w^948721	w^361762]
[w*110971	w*688696	w^923340	w^214480	w*747262	w*926785	w^489888	w^952228	w*445776	w*423943	w^649872	w^22829	w*356036	w*786585	w^939812	w^920848	w*394764	w^1024427	w^678614	w^7235101
[w*625737	w*796362	w^516193	w^823319	w*487968	w^792252	w^784791	#^442528	w*954389	w*636815	w^499427	w^651880	w*325788	w*982786	w^259759	#^784348	w*738285	w^423883	w^788285	w^8648591
[w*637454 ·	w^1085277	w^634523	w^597556	w*899547	w*50212	w^771166	w^176848	w*739499	w^1012145	w^894991	w^297488	w*401139	w*445943	w^868647	w^985819	w*586967	w*555279	w*11988	w^3938681
[w*491731	w*105205	w^838924	w^523184	w*133971	w*621235	w^376245	₩^675466	w*253911	w^239981	w^1012958	w^945889	w*722834	w^1847441	w^191877	w^127222	w*163984	w^286531	w^1005897	w^628569]
[w*525867	w*931838	w^738973	w^215016	w*658662	w*195917	w^891334	w^983857	w*441472	w*344288	w^417181	₩^577925	w*705133	w*41732	w^482983	₩*837857	w*465912	w^268532	w^478886	w^988553]
[w^93536	w*751673	w^536866	w^729978	w*123721	w*756989	w^668630	₩^636987	w*454234	w*51525	w^574824	w^308632	w*461191	w^137995	w^836411	₩^667454	w*403991	w^986525	w^553448	w^160312]
[w*269887	w*696612	w^755898	w^178280	w*828283	w^733597	w^859384	w^569348	w*146891	w^116635	w^1004028	w^759877	w*155728	w*889492	w^303886	w*1014978	w*798897	w^325774	w^228381	w^278762]
ROANBONNOAN	*****																		

Hiding the structure Algebraic attacks in the random case

Hiding the structure

Few visual examples : M of size 20×20 , coefficients in $GF(2^{20}) \approx 1$ million elements, dim(F) = 2. $dim(support(M^{-1})) = 19$.

*****	*****																		
Matrice M=																			
L W-302254		W-302254	M305524		W-362254	W-471166	#307524	W-362254			M305524	W-362254		W-302254	m.1333446	W-733990	M411100		01
[W*362254	W*733998	W"4/1105	8	W~733998	W*362254	W*362254	W*362254	W^4/1186	W*733998	9	W*362254	W~733998	W*352254	Mu133449	W*4/1186	W*4/1186	8	W*362254	W^4/1186]
L 0		w^4/1185	mv133008		w*733998	w~733998	8	w~733998	w*471186	9	₩^362254		w*471186	9	₩^362254	w*733998	w~733998	w^4/1186	61
[W^4/1186	W*4/1186	Mu133AA9	W*4/1186	W*302254	W*733998	W"362254	8	W^4/1186	8	Mu133AAB	W*362254	W^4/1186	9	Mu133AAA	W^4/1186	W*4/1186	8	W*362254	W*362254]
L 0	w*362254	w^471186	9	w^471186	w*733998	w^733998	w^362254	w^471186	w*362254	w^471186	9	w*471186	9	w^362254	8	w*733998	w*471186	w^733998	w^3622541
L W*302254	W*362254	9	W^4/1186	W*/33998	9	W*362254	W^4/1186	W*733998	w~4/1186	9	#r/33448	9	w~4/1186	Mu 133009	9	W*733998	9	Mu133AA9	Mv133AA61
L w*362254	w*733998	9	w^471186	w^471186	9	w^733998	w^733998	w*362254	w*471106	w^733998	9	w*362254	w*471186	w^362254	w^362254	w*471186	w*733998	9	w^362254]
L w*/33990	w*362254	9	W^4/1186	6	9	9	#^4/1186	w^4/1186	9	9	#v133448	w~733998	w~4/1186	Wn362254	6	w~4/1106	9	9	w^362254]
L w*733990	9	w^733998	w^471186	9	9	w^471106	w^471186	9	9	w^362254	₩^362254	w*471186	w*362254	9	w^362254	9	9	w^471106	w^733990]
1 0	w~733998	w^4/1185	6	w*302254	w~4/1186	9	6	6	9	9	#^4/1186	w^4/1106	9	9	w^362254	w*302254	w~733998	9	61
[w*471186	9	w^471186	w^733990	w*471186	w*733998	w^733998	w^471186	w*362254	w*733998	9	w^733990	w*733998	9	w^733990	9	w*471186	9	w^471186	6)
L w*4/1186	w*733998	w~733998	mv133008	w*733998	w*362254	wh362254	w^4/1186	w~733998	w*362254	w^362254	6	w*733998		w~733998	₩^362254	w*733998	w*471186	w^362254	#*733998]
L 0	w*733998	w^471186	w^362254	w*733998	w*471186	w^733998	9	w*471186	w*471186	w^733998	9	9	w*362254	w^362254	w^733990	w*733998	9	w^471186	w^362254]
1 0	w*471106	9	w^4/1188	w*733998	9	w^362254	#^362254	w*733998	9	w^4/1106	mv133666	w^4/1106	w*362254	w^4/1106	₩^4/1188	6	w~733998	w^362254	mv1336661
[w*471186	w*471186	w^471186	9	w*471186	w*471186	w^471186	w^471186	w*733998	w*733998	w^362254	w^471186	w*733998	w*471186	w^362254	w^733990	9	w^362254	w^362254	w^362254]
L w*471186	w*471106	w^733998	w^733998	w^471186	w*362254	w^471186	w^471188	w*362254	w*733998	w^362254	mv133666	w*362254	w*362254	9	w^733990	w*471186	w*471106	9	w^3622541
[w*362254	w*362254	w^362254	w^733998	w*471186	w*733998	9	w^471186	w*362254	w*471186	w^362254	w^733998	w*471186	w ⁴ 471186	9	w^471186	w*471186	w^362254	w^471106	w^362254]
L 0	9	9	w^362254	9	w*471106	w^362254	w^471186	w*733990	9	w^362254	₩^362254	w*362254	w*362254	w^733990	9	w*362254	w*471106	w^733990	w^7339901
[w*362254	w*733998	w^733998	0	6	9	w^733998	0	6	w^733998	w^362254	0	6	w*362254	9	9	w*733998	w^733998	w^733990	w^471106]
[w*733990	w*471106	9	w^733990	w*471186	w*471106	9	w^733990	w*362254	w*362254	9	w^733990	w*733990	w^733998	9	w^362254	w*733990	9	w^362254	w^362254J
*OANCOANCOAN	ROANDON																		
Natrice M^(-1)=																		
L w^88448	w*988123	w^863025	w^898785	w*139452	w*623210	w^594837	w^864752	w*105343	w*28264	w^1845833	w^888787	w*617636	w*265974	w^189285	W^94741	w*801759	w*52989	w^687858	w^699552]
[W^3333957	W*942554	Wn/886/8	W^618924	Mu 1986252	M., 448440	W-964468	8-119946	W*1846953	W*959233	W"/6828/	W*824678	W*//6259	W* 356585	W*3841/3	₩^63/539	w*63/281	W*111491	W"//4659	W-982573]
L w*147497	w*781692	w^115424	w^811426	w*649856	w*728662	w^156367	w^142081	w*186927	w*633388	w*63598	₩^639272	w*220801	w*862548	w^417167	w^412288	w*982778	w*844957	w^218985	w^784658]
[w*311167	w*23150	w^296538	w^567087	w*831522	w*221167	w^288209	w^468301	w*297189	w*764454	w^537310	w^720237	w^99788	w*695987	w^253634	w^691839	w*524141	w^551022	w^223229	w^875846]
L w*993583	w*93992	w^478192	w^810565	w*141922	w*942386	w^117984	w^334543	w*819778	w^5759	w^855873	₩^181661	w*888342	w*243147	w^447472	w^465924	w*763151	w*897715	w^278322	w^361146J
[w^841810	w*/56/62	w~12/936	w^2/3301	w*547858	w~1/2200	Mu346393	#*344845	w*263618	w*13923	W^595539	w^57782	w*6145/2	w*312908	m.1662868	w*/5/103	w*536479	w~283298	Wn351286	w^195139]
L w*590967	w*786728	w^225921	w^381198	w*114839	w*44222	w^812786	w^910129	w*884021	w*155814	w^488812	₩^545498	w*158519	w*588939	w^795331	w^495753	w*517364	w*938985	w^340051	w^443794]
L w*334420	w*892788	w^546219	w^342348	w*443793	w~163168	w*31191	m-226868	W^44884	w-933333	w^234422	w^977827	w*797738	w*777338	w^244168	₩^826510	w*425288	w~916518	w^1010002	w^28974]
[W*/2/08/	W*537798	W"38/18/	W*622828	W"98354	W*1/5592	W"578483	W*/58515	W-024998	W"1021246	W"/1539/	W*345528	W*915186	W*455894	W"868478	M-8893AP	W*913260	W*34//61	W*32665	M., AA191]
L w*262635	w*384261	w^266488	w^587969	w*548688	w*268454	w^268662	#*767743	w*566267	w*281989	w^1009597	w^577284	w*549731	w^230250	w^522123	₩^491763	w*994584	w*766994	w^652178	w^19346]
[W*/8/261	W*/8/284	W*5//188	8-944193	Mu 1998A	W^72114	W"168346	W*188/8/	W-198/868	W*/8632	Mu243328	#*/34432	W*33828	W*61/55/	W"/55685	W^123481	W*8488/4	W*/1/020	W*239878	W^4582/3]
L w*653534	w*254962	w^613788	w*1029328	w*731945	w*737731	w^257473	w^402995	w*202381	w*918616	w^295557	₩^889336	w*351942	w*380277	w^125269	w^100947	w^12813	w*69492	w^816486	w*1003168]
L W*828747	w*4/2469	W*375392	W^928977	W*812964	W*318578	W-966941	#°935842	w~1827964	W*619944	W-8284/1	#~296838	M-993A91	W*423541	W~184/52	W-054/8/	W*3/9296	W*732234	W*948/21	W^361/62]
l w*110971	w*6889696	w^923340	w^214480	w*747262	w*926785	w^489888	w^952228	w*445776	w*423943	w^649872	w^22829	w*356836	w*786585	w^939812	₩^920848	w*394764	w^1024427	w^670614	w^723510]
L w^625/3/	w*796362	w~516193	w^823319	W*48/968	w*792252	W~784791	#*442528	w*954389	w*636815	W*499427	₩^651888	w*325780	w~982786	W^259759	8~784348	w*/38285	w*423883	wn/88285	#~864859]
l w*637454	w^1005277	w^634523	w^597556	w*899547	w*50212	w^771166	w^176848	w*739499	w^1012145	w^894991	w^297488	w*401139	w*445943	w^868647	w^985819	w*586967	w*555279	w*11988	w^393868]
L w^491731	w~105205	w^838924	w^523184	w^133971	w*621235	wn3/6245	w^0/5466	w^253911	w^239981	w^1012958	w^y45889	w*/22834	w^1847441	w^191877	w^127222	w*103984	w^286531	w^1005897	w^0285691
l w*525867	w*931838	w^738973	w^215916	w*658662	w*195917	w^891334	w^903857	w*441472	w*344288	w^417181	w^577925	w*785133	w*41732	w^482983	₩^837857	w*465912	w^268532	w^478886	w^988553]
L w^93536	w*751673	wn536866	w^729978	w*123721	w^756989	w^668630	₩^636987	w*454234	w^51525	w^574824	₩^388632	w~461191	w~137995	w^836411	w^007454	w~403991	w1986525	w^553448	w^168312]
[w*269887	w*696612	w^755898	w^178280	w*828283	w*733597	w^859384	w^569348	w*146891	w*116635	w^1084828	w^759877	w*155720	w*889492	w^303886	w*1014978	w*798897	w*325774	w^228381	w^278762]
*******	*****																		

Hiding the structure Algebraic attacks in the random case

Hiding the structure

```
mathis-130:beamer bros$ magma
                           Wed May 22 2019 10:06:37
Magma V2.23-1 (STUDENT)
                                                        [Seed = 1608908847]
Type ? for help. Type <Ctrl>-D to guit.
> load "H.magma";
Loading "H.magma"
n =
250
r =
10
affichage =
Ø
Trial 1
                  => dimension support(M) = 10, dimension support(M<sup>(-1)</sup>) = 250 (max=250)
Trial 2
                  => dimension support(M) = 10, dimension support(M^(-1)) = 250 (max=250)
Trial 3
                  => dimension support(M) = 10, dimension support(M^{(-1)}) = 250 (max=250)
Trial 4
                 => dimension support(M) = 10, dimension support(M<sup>(-1)</sup>) = 250 (max=250)
Trial 5
                  => dimension support(M) = 10, dimension support(M^{(-1)}) = 250 (max=250)
Trial 6
                  => dimension support(M) = 10, dimension support(M^{(-1)}) = 250 (max=250)
Trial 7
                 = dimension support(M) = 10, dimension support(M^(-1)) = 250 (max=250)
Trial 8
                  => dimension support(M) = 10, dimension support(M^(-1)) = 250 (max=250)
Trial 9
                  => dimension support(M) = 10, dimension support(M^(-1)) = 250 (max=250)
Trial 10
                  => dimension support(M) = 10, dimension support(M^(-1)) = 250 (max=250)
```

Conclusion about sparsity and coding theory :

• One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)

- One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)
- But a sparse parity check matrix is enough :

 the sparser its parity check matrix is, the more a code can decode.

- One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)
- But a sparse parity check matrix is enough :

 the sparser its parity check matrix is, the more a code can decode.
- For instance, LDPC (*H* with $\mathcal{O}(1)$ 1' per row) can decode up to $\mathcal{O}(n)$ errors.

- One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)
- But a sparse parity check matrix is enough :

 the sparser its parity check matrix is, the more a code can decode.
- For instance, LDPC (*H* with $\mathcal{O}(1)$ 1' per row) can decode up to $\mathcal{O}(n)$ errors.
- Nevertheless, in cryptography, we prefer to use MDPC codes :

- One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)
- But a sparse parity check matrix is enough :

 the sparser its parity check matrix is, the more a code can decode.
- For instance, LDPC (*H* with $\mathcal{O}(1)$ 1' per row) can decode up to $\mathcal{O}(n)$ errors.
- Nevertheless, in cryptography, we prefer to use MDPC codes :
 - We do not want to correct errors but to increase the complexity of attacks.

- One could think that the more structured a code is, the more it can decode (Reed-Solomon, Reed-Muller, Gabidulin, ...)
- But a sparse parity check matrix is enough :

 the sparser its parity check matrix is, the more a code can decode.
- For instance, LDPC (*H* with $\mathcal{O}(1)$ 1' per row) can decode up to $\mathcal{O}(n)$ errors.
- Nevertheless, in cryptography, we prefer to use MDPC codes :
 - We do not want to correct errors but to increase the complexity of attacks.
 - For instance, attacks on LDPC codes would be too easy due to a small private key size.

Hiding the structure Algebraic attacks in the random case

Algebraic Attack against RSD

As seen previously, the security of rank-based cryptosystems relies on the RSD problem.

From now on, I will consider code over $GF(2^N)$.

Hiding the structure Algebraic attacks in the random case

Algebraic Attack against RSD

As seen previously, the security of rank-based cryptosystems relies on the RSD problem.

From now on, I will consider code over $GF(2^N)$.

• From $He^T = s$ one gets a system of equations.

As seen previously, the security of rank-based cryptosystems relies on the RSD problem.

From now on, I will consider code over $GF(2^N)$.

- From $He^T = s$ one gets a system of equations.
- Very particular form : Quadratic, quasi-bihomogeneous

As seen previously, the security of rank-based cryptosystems relies on the RSD problem.

From now on, I will consider code over $GF(2^N)$.

- From $He^T = s$ one gets a system of equations.
- Very particular form : Quadratic, quasi-bihomogeneous

$$\begin{array}{l} {\sf E}_1 = 5 * e = 4 = 1 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e = 4 \\ {\sf E}_1 = 2 * e \\ {\sf$$

Hiding the structure Algebraic attacks in the random case

Algebraic Attack against RSD

• Take advantage of the small rank of *e*

 \implies over-constrained system of equations

Hiding the structure Algebraic attacks in the random case

- Take advantage of the small rank of *e*
 - \implies over-constrained system of equations
- Number of unknowns : Nr+nr. Nr for the basis of E, nr for e.

- Take advantage of the small rank of e
 ⇒ over-constrained system of equations
- Number of unknowns : Nr+nr. Nr for the basis of E, nr for e.
- Number of equations : (n k) equations in $GF(2^N)$, so N(n k) equations.

- Take advantage of the small rank of e
 ⇒ over-constrained system of equations
- Number of unknowns : Nr+nr. Nr for the basis of E, nr for e.
- Number of equations : (n k) equations in $GF(2^N)$, so N(n k) equations.
- Usually n = N, k = n/2 and $r = O(\sqrt{n})$, so $2n^{3/2}$ unknowns for $(1/2)n^2$ equations.

- Take advantage of the small rank of e
 - \Longrightarrow over-constrained system of equations
- Number of unknowns : Nr+nr. Nr for the basis of E, nr for e.
- Number of equations : (n k) equations in $GF(2^N)$, so N(n k) equations.
- Usually n = N, k = n/2 and $r = O(\sqrt{n})$, so $2n^{3/2}$ unknowns for $(1/2)n^2$ equations.
- One way to solve this system is to use Gröbner basis computation, but their complexity is poorly known for system which are not semi-regular.

Hiding the structure Algebraic attacks in the random case

Algebraic Attack against RSD

 More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- But in our case, the system is **over-constrainted** and it has a strong structure coming from the structure of *GF*(2^{*N*}).

- More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- But in our case, the system is **over-constrainted** and it has a strong structure coming from the structure of *GF*(2^{*N*}).
- Nowadays, GB computation complexity for over-constrainted systems over *GF*(2) is still an open question.

- More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- But in our case, the system is **over-constrainted** and it has a strong structure coming from the structure of *GF*(2^{*N*}).
- Nowadays, GB computation complexity for over-constrainted systems over *GF*(2) is still an open question.
- Conjecture/Heuristic for the case "quadratically" over-constrainted.

- More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- But in our case, the system is **over-constrainted** and it has a strong structure coming from the structure of $GF(2^N)$.
- Nowadays, GB computation complexity for over-constrainted systems over *GF*(2) is still an open question.
- Conjecture/Heuristic for the case "quadratically" over-constrainted.
- Observations on small RSD instances

- More precisely, if the system were random with *# unknowns* ≈ *# equations*, we would have some complexity bounds ([Bar04]).
- But in our case, the system is **over-constrainted** and it has a strong structure coming from the structure of $GF(2^N)$.
- Nowadays, GB computation complexity for over-constrainted systems over *GF*(2) is still an open question.
- Conjecture/Heuristic for the case "quadratically" over-constrainted.
- Observations on small RSD instances
 - \implies drop in complexity in comparison to random systems.

Introduction Decoding codes with sparse parity check matrix Applications in Cryptography

Hiding the structure Algebraic attacks in the random case

Algebraic Attack against RSD

[unknowns, equations]	d _{reg} random system	d _{reg} for RSD systems
[48, 48]	8 (Bardet)	$[8,8,2,3] \implies 5$
[48, 48]	8 (Bardet)	$[12, 12, 8, 2] \implies 4$
[58, 70]	7 or 8 (*)	$[10, 10, 2, 4] \implies 5$

(*) 7 is a prediction using my conjecture for over-constrained system whereas 8 would be the result of Bardet's method for semi-regular system.

Introduction Decoding codes with sparse parity check matrix Applications in Cryptography

Hiding the structure Algebraic attacks in the random case

Merci ! Thank you !

:-)

References : [McE78], [MTSB13], [GMRZ13], [Bar04].

References



Magali Bardet.

Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.

Theses, Université Pierre et Marie Curie - Paris VI, 2004.



Philippe Gaborit, Gaétan Murat, Olivier Ruatta, and Gilles Zémor. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC*, volume 2013, 2013.



Robert J McEliece.

A public-key cryptosystem based on algebraic coding theory. *Coding Thv*, 4244:114–116, 1978.



Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo SLM Barreto.

MDPC-McEliece: New McEliece variants from moderate density parity-check codes.

In *2013 IEEE international symposium on information theory*, pages 2069–2073. IEEE, 2013.