McNie

A code-based public key encryption scheme

Jon-Lark Kim* (presenter)
Young-Sik Kim**
Lucky Galvez*
Myeong Jae Kim*
Nari Lee*
* Sogang University, S. Korea, ** Chosun University, S. Korea

NIST Post-Quantum Cryptography Workshop
April 12 2018

Outline

Advantage

- The McEliece cryptosystem and its variants are well known code-based public key cryptosystems:

$$\mathbf{c} = \mathbf{m}G + \mathbf{e},$$

  where $\mathbf{m}$ is a message, G is a generator matrix for a code which can correct errors $\mathbf{e}$, and $\mathbf{c}$ is a ciphertext.
- However, McEliece cryptosystems with many algebraic codes with good structures have been broken due to their structures except for Goppa codes.
- Our McNie is a new code-based public key cryptosystem which is less vulnerable against currently known structural attacks.
- We can use Hamming weight or rank weight in general.

McNie- Key generation

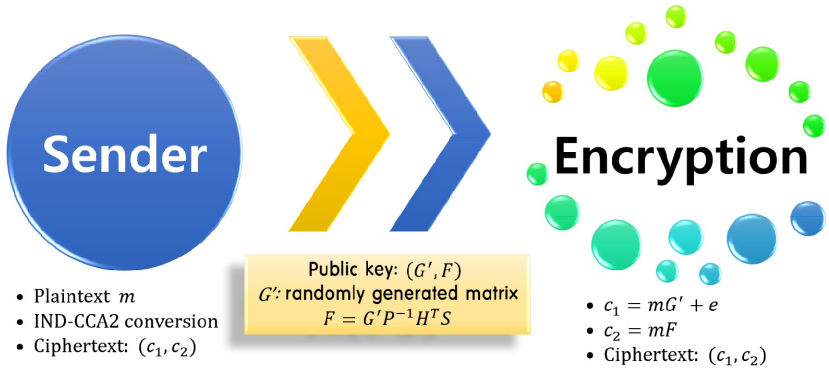
- Consider Hamming weight or rank weight.
- **Secret key:** (H, P, S, Φ_H)
  H: a parity check matrix for an $[n, k]$ code C over $\mathbb{F}_{q^m}$
  P: an $n \times n$ permutation matrix
  S: an $(n-k) \times (n-k)$ invertible matrix over $\mathbb{F}_{q^m}$
  Φ_H: an efficient decoding algorithm for C which corrects errors of weight up to r

- **Public key:** (G', F)
  G': Generator matrix for a random $[n, l]$ code over $\mathbb{F}_{q^m}$
  $F = G' P^{-1} H^T S$

McNie- Encryption

Message: $\mathbf{m} \in \mathbb{F}_{q^m}^{l}$

- Randomly generate $\mathbf{e} \in \mathbb{F}_{q^m}^{n}$ of weight r
- $Enc(\mathbf{m}) = (\mathbf{c_1}, \mathbf{c_2})$
  $\mathbf{c_1} = \mathbf{m}G' + \mathbf{e}$
  $\mathbf{c_2} = \mathbf{m}F = \mathbf{m}G'P^{-1}H^TS$



McNie- Decryption

Received vector: $\mathbf{c} = (\mathbf{c_1}, \mathbf{c_2})$

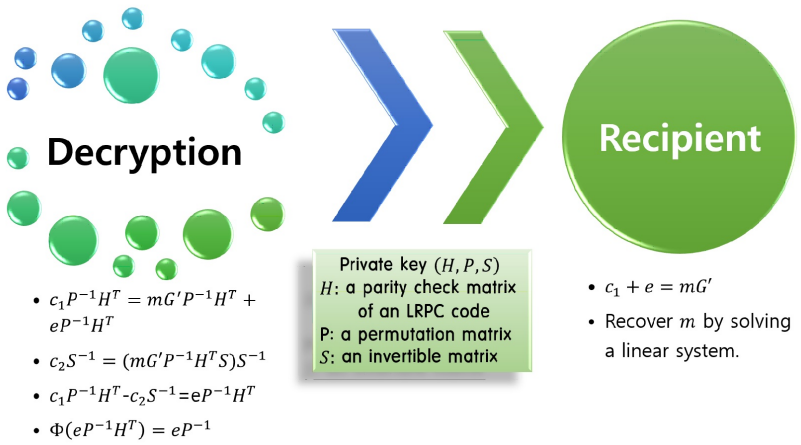
- Compute

$$\begin{aligned} \mathbf{s}' &= \mathbf{c_1} P^{-1} H^T - \mathbf{c_2} S^{-1} \\ &= (\mathbf{m} G' + \mathbf{e}) P^{-1} H^T \\ &\quad -(\mathbf{m} G' P^{-1} H^T S) S^{-1} \\ &= \mathbf{e} P^{-1} H^T \\ \mathbf{e}' &= \Phi_H(\mathbf{s}') = \mathbf{e} P^{-1} \\ \mathbf{e} &= \mathbf{e}' P \end{aligned}$$

- Solve the system

$$\mathbf{m} G' = \mathbf{c_1} - \mathbf{e}$$

to recover $\mathbf{m}$.



Apply McNie to rank metric codes

Let $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

$$c = (c_1, \ldots, c_n) \in \mathbb{F}_{q^m}^n \Leftrightarrow \bar{c} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}, \quad c_j = \sum_{i=1}^{m} c_{ij}\alpha_i$$

- **rank weight**: $w_R(c) = Rank(\bar{c})$
- **rank distance**: $d_R(c, c') = Rank(\bar{c} - \bar{c}')$

A *rank metric code* is an $[n, k]$ code over $\mathbb{F}_{q^m}$ equipped with the rank metric.

A family of rank metric codes used in McNie:

A *Low Rank Parity Check (LRPC)* code of rank d is an $[n, k]$ code over $\mathbb{F}_{q^m}$ that has for its parity check matrix an $(n - k) \times n$ matrix $H = (h_{ij})$ such that the sub-vector space of $\mathbb{F}_{q^m}$ generated by its coefficients h_{ij} has dimension at most d.

Using 3-quasi-cyclic LRPC codes

We use circulant matrices and construct quasi-cyclic LRPC codes over $\mathbb{F}_{q^m}$ in order to reduce key size.

Let n be a multiple of 3 and block size $blk = \frac{n}{3}$.

- Generate $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3 \in \mathbb{F}_{q^m}^{blk}$ s.t. dim $Supp(\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3) = d$
- Generate $\mathbf{g}_1, \mathbf{g}_2 \in \mathbb{F}_{q^m}^{blk}$.
- Let H_i, G_j be circulant matrices whose first row are $\mathbf{h}_i$ and $\mathbf{g}_j$, resp.
- Let $H = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix}$, $G' = \begin{bmatrix} I_{blk} & 0 & G_1 \\ 0 & I_{blk} & G_2 \end{bmatrix}$
- Take $P = I_n$ and $S = (H_1^T + G_1 H_3^T)^{-1}$ which is also a circulant matrix.
- $F = G' P^{-1} H^T S$ has the following form :

$$F = \begin{bmatrix} I_{\frac{n}{3}} \\ F' \end{bmatrix},$$

where $F' = (H_2^T + G_2 H_3^T)(H_1 + H_3 G_1^T)^{-1}$.

Using 4-quasi-cyclic LRPC codes

Let n be divisible by 4 and block size $blk = \frac{n}{4}$.

- Generate $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_8 \in \mathbb{F}_{q^m}^{blk}$ s.t. dim $Supp(\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_8) = d$.
- Generate vectors $\mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3 \in \mathbb{F}_{q^m}^{blk}$.
- Let $H = \begin{bmatrix} H_1 & H_2 & H_3 & H_4 \\ H_5 & H_6 & H_7 & H_8 \end{bmatrix}$, $G' = \begin{bmatrix} I_{blk} & 0 & 0 & G_1 \\ 0 & I_{blk} & 0 & G_2 \\ 0 & 0 & I_{blk} & G_3 \end{bmatrix}$
- Take $P = I_n$ and $\bar{S} = \begin{bmatrix} S_1 & S_2 \\ S_3 & S_4 \end{bmatrix}$, where S_1, S_2, S_3, S_4 are $blk \times blk$ circulant matrices.
- $\bar{F} = G' P^{-1} H^T S = \begin{bmatrix} F_1 & F_2 \\ F_3 & F_4 \\ F_5 & F_6 \end{bmatrix}$
- Reduce $\bar{F}$ in column echelon form $F = \bar{F}E = \begin{bmatrix} I_{blk} & 0 \\ 0 & I_{blk} \\ F' & F'' \end{bmatrix}$, where

$$E = \begin{bmatrix} E_1 & E_2 \\ E_3 & E_4 \end{bmatrix} = \begin{bmatrix} (F_2^{-1}F_1 - F_4^{-1}F_3)^{-1}F_2^{-1} & (F_4^{-1}F_3 - F_2^{-1}F_1)^{-1}F_4^{-1} \\ -F_4^{-1}F_3E_1 & -F_2^{-1}F_1E_2 \end{bmatrix}.$$

IND-CCA2 Conversion

Notations

$Prep(\mathbf{m})$	:	Preprocessing to a message $\mathbf{m}$, such as data-padding, etc. Its inverse is $Prep^{-1}()$.
$Hash(\mathbf{x})$	:	One-way hash function of an arbitrary length binary string x to a fixed length binary string.
$Conv(\bar{\mathbf{z}})$	:	Bijective function which converts a vector $\bar{\mathbf{z}}$ over $\mathbb{F}_{q^m}$ into the corresponding error vector $\mathbf{z}$ of length n with a constant rank weight r. Its inverse is $Conv^{-1}()$.
$Gen(\mathbf{x})$	:	Generator of a cryptographically secure pseudo random sequences of arbitrary length from a fixed length seed x.
$Msb_{x_1}(x_2)$	:	The left x_1 bits of x_2.
$Lsb_{x_1}(x_2)$	:	The right x_1 bits of x_2.
$Const$	:	Predetermined constant used in public.
$Rand$	:	Random source which generates a truly random (or computationally indistinguishable pseudo random) sequence.
$\mathcal{E}^{McNie}(x, z)$	:	Encryption of x using McNie PKC with an error vector z.
$\mathcal{D}^{McNie}(x)$	:	Decryption of x using McNie PKC.

IND-CCA2 Conversion based on Kobara-Imai generic CCA-2 conversion [4]

Encryption of m:		**Decryption of c:**	
$\mathbf{r}$	$:= Rand$	y_5	$:= Msb_{n-l}(\mathbf{c})$
$\bar{\mathbf{m}}$	$:= Prep(\mathbf{m})$	z	$:= Lsb_{2n}(\mathbf{c})$
y_1	$:= Gen(\mathbf{r}) \oplus (\bar{\mathbf{m}}\|\|Const)$	$\mathbf{c_1}$	$:= Msb_n(z)$
y_2	$:= \mathbf{r} \oplus Hash(y_1)$	$\mathbf{c_2}$	$:= Lsb_n(z)$
$(y_5\|\|y_4\|\|y_3)$	$:= (y_2\|\|y_1)$	y_3	$:= \mathcal{D}^{McNie}(\mathbf{c_1}\|\|\mathbf{c_2})$
$\mathbf{e}$	$:= Conv(y_4)$	$\mathbf{e}$	$:= y_3 G' \oplus \mathbf{c_1}$
$(\mathbf{c_1}\|\|\mathbf{c_2})$	$:= \mathcal{E}^{McNie}(y_3, \mathbf{e})$	y_4	$:= Conv^{-1}(\mathbf{e})$
$\mathbf{c}$	$:= (y_5\|\|\mathbf{c_1}\|\|\mathbf{c_2})$	$(y_2\|\|y_1)$	$:= (y_5\|\|y_4\|\|y_3)$
return $\mathbf{c}$		$\mathbf{r}$	$:= y_2 \oplus Hash(y_1)$
		$\bar{\mathbf{m}}\|\|Const'$	$:= Gen(\mathbf{r}) \oplus y_1$
		If	$Const' = Const$
			return $Prep^{-1}(\bar{\mathbf{m}})$
		Otherwise	reject $\mathbf{c}$

Suggested parameters

Parameter	n	k	l	blk	d	r	m	q	Category
`encrypt/3Q_128_1`	93	62	62	31	3	5	37	2	1
`encrypt/3Q_128_2`	105	70	70	35	3	5	37	2	1
`encrypt/3Q_192_1`	111	74	74	37	3	7	41	2	3
`encrypt/3Q_192_2`	123	82	82	41	3	7	41	2	3
`encrypt/3Q_256_1`	111	74	74	37	3	7	59	2	5
`encrypt/3Q_256_2`	141	94	94	47	3	9	47	2	5

Table: Suggested parameters using 3-quasi-cyclic LRPC codes

Parameter	n	k	l	blk	d	r	m	q	Category
`encrypt/4Q_128_1`	60	30	45	15	3	5	37	2	1
`encrypt/4Q_128_2`	72	36	54	18	3	5	37	2	1
`encrypt/4Q_192_1`	76	38	57	19	3	7	41	2	3
`encrypt/4Q_192_2`	84	42	63	21	3	7	41	2	3
`encrypt/4Q_256_1`	76	38	57	19	3	7	53	2	5
`encrypt/4Q_256_2`	88	44	66	22	3	8	47	2	5

Table: Suggested parameters using 4-quasi-cyclic LRPC codes

Key sizes for suggested parameters

Parameter	Decryption		Public Key Size (bytes)	Private Key Size (bytes)	Message Size (bytes)	Ciphertext Size (bytes)
	failure 1	failure 2				
`encrypt/3Q_128_1`	-17	-34	431	194	314	579
`encrypt/3Q_128_2`	-20	-34	486	218	358	653
`encrypt/3Q_192_1`	-17	-26	569	247	454	764
`encrypt/3Q_192_2`	-20	-26	631	274	505	846
`encrypt/3Q_256_1`	-17	-62	819	337	636	1097
`encrypt/3Q_256_2`	-20	-22	829	348	699	1110

Table: Key sizes for the suggested parameters for McNie using 3-QC-LRPC codes

Parameter	Decryption		Public Key Size (bytes)	Private Key Size (bytes)	Message Size (bytes)	Ciphertext Size (bytes)
	failure 1	failure 2				
`encrypt/4Q_128_1`	-16	-34	347	340	215	422
`encrypt/4Q_128_2`	-21	-34	417	401	264	505
`encrypt/4Q_192_1`	-18	-26	487	465	336	590
`encrypt/4Q_192_2`	-21	-26	539	512	373	651
`encrypt/4Q_256_1`	-18	-50	630	584	432	761
`encrypt/4Q_256_2`	-20	-30	647	601	461	781

Table: Key sizes for the suggested parameters for McNie using 4-QC-LRPC codes

McNie vs other cryptosystems

Security Level	McNie		DC-LRPC [3]	DC-MDPC [5]	QD-Goppa [6]	Goppa [2]
	3-quasi	4-quasi				
128	**3441**	**2775**	2809	9857	32768	1537536
192	**4551**	**3895**	-	-	45056	4185415
256	**6549**	**5035**	-	32771	65536	7667855

Table: Key-size (bits) comparison with other code-based cryptosystems

Security Level	McNie		NTRU	RSA	ECC	ECC AWC
	3-quasi	4-quasi				
128	**3441**	**2775**	4939	3072	256	277280
192	**4551**	**3895**	6523	7680	384	936618
256	**6549**	**5035**	8173	15360	512	1595434

Table: Comparison of key sizes (bits)

Recent attack on McNie based on 3,4-QC LRPC codes by P. Gaborit

- Let $\mathbf{m} = (m_1, m_2, \ldots, m_l)$
- From $\mathbf{c_2} = \mathbf{m}F$, we obtain $n-k$ linear relations of the m_i's. Hence, all the m_i's can be expressed in terms of some fixed $l-(n-k)$ coordinates.
- We can rewrite c_1 as

$$c_1 = \mathbf{m}'G'' + \mathbf{e}$$

  where G'' is of dimension $l-(n-k)$.
- So we attack a code of dimension $l-(n-k)$ instead of a code of dimmension l.

Improvement on generic attacks on RSD(Rank Syndrome Decoding) by Aragon, Gaborit, Hauteville, Tillich [1]

The attack is an adaptation of the ISD attack to RSD.

This improvement uses the $\mathbb{F}_{q^m}$-linearity of the code.

The main idea is to consider the code $C' = C + \mathbb{F}_{q^m}\mathbf{e}$. The problem is then reduced to finding a weight r codeword in C'.

Instead of looking for the support E of the error $\mathbf{e}$, we can look for a multiple αE, $\alpha \in \mathbb{F}_{q^m}^*$, of the support. This attack has complexity

$$\mathcal{O}(n-k)^3 m^3 q^{r\frac{(k+1)m}{n}-m}.$$

Updated parameters for 3,4-QC LRPC codes

n	l	k	d	r	m	q	failure	Key Size (bytes)	security
120	80	80	3	8	53	2	-23	795	128
138	92	92	3	10	67	2	-25	1156	192
156	104	104	3	12	71	2	-27	1385	256

Table: New suggested parameters for McNie using 3-quasi-cyclic LRPC code

n	l	k	d	r	m	q	failure	Key Size (bytes)	security
92	46	69	3	10	59	2	-36	849	128
112	56	84	3	13	67	2	-38	1173	192
128	64	96	3	16	73	2	-36	1460	256

Table: New suggested parameters for McNie using 4-quasi-cyclic LRPC code

Some parameters for McNie using Gabidulin codes

n	l	k	q	m	r	Key size (bits)	Security
42	36	26	2	42	8	87696	128
48	44	30	2	48	9	139392	192
55	50	33	2	55	11	211750	256

H. Rashwan, E. M. Gabidulin and B. Honary — **Security of the GPT cryptosystem**

Table III. Comparison of the three public-key cryptosystems.

PKC	Parameters	Public Key size (bits)	Complexity	Security
McEliece	Binary, $n = 1024$, $K = 534$, $t = 50$.	2^{19}	$> 2^{64}$	Insecure Bernstein Attack
McEliece Modified 1	Binary, $n = 1632$, $K = 1269$, $t = 33$.	2^{19}	$>2^{64}$	Secure
McEliece Modified 2	Binary, $n = 2960$, $K = 2288$, $t = 56$.	2^{21}	$>2^{64}$	Secure
McEliece Modified 3	Binary, $n = 6624$, $K = 5129$, $t = 115$.	2^{23}	$>2^{64}$	Secure
Niederreiter	q-array, $n = 128$, $d = 64$, $r = 63$.	32,000	$>2^{21}$	Insecure
Niederreiter Modified 1	q-array, $n = 128$, $d = 64$, $r = 63$.	32,000	$>2^{75}$	Secure
GPT Modified	$q = 2^{20}$, $n = 20$, $d = 9$, $k = 12$, $t_1 = 1$.	4800	$>2^{46}$	Insecure Gibson Attack
GPT Modified 1	$q = 2^{20}$, $n = 20$, $d = 9$, $k = 12$, $t_1 = 2$.	4800	$>2^{66}$	Insecure Gibson Attack
GPT Modified 2	$q = 2^{20}$, $n = 20$, $d = 11$, $k = 10$, $t_1 = 3$.	4800	$>2^{86}$	Insecure Gibson Attack
GPT Modified 3	$q = 2^{29}$, $n = 29$, $d = 15$, $k = 15$, $t = 7$, $t_1 = 4$, $s = 1$.	19 Kbits	$>2^{160}$	Insecure Overbeck's Attack
GPT Modified 4	$q = 2^{32}$, $n = 32$, $d = 17$, $k = 16$, $t = 8$, $t_1 = 4$, $p = 3$.	13 Kbits	$>2^{164}$	Insecure Overbeck's Attack
GPT Reducible Rank codes	$q = 2^{20}$, $N = 20$, $n = 40$, $k = 24$, $t = 4$, $t_1 = 3$, $m = 10$, $r = 2$.	10 Kbits	$>2^{70}$	Insecure Overbeck's Attack
Instrumental approach	$q = 2^{20}$, $n = 20$, $d = 11$, $k = 10$, $t = 5$, $t_1 = 4$,	4 Kbits	$>2^{80}$	Secure against all known Attack

References


Aragon, N., Gaborit, P., Hauteville, H., Tillich, J.-P.: Improvement of generic attacks on the rank-syndrome decoding problem. 2017. <hal-01618464>


Bernstein, D.J., Lange, T., and Peters, C.: Attacking and defending the McEliece cryptosystem. In Proceedings of the 2nd International Workshop on Post-Quantum Cryptography, PQCrypto '08, pp. 31–46, Springer-Verlag, Berlin, Heidelberg (2008).


Gaborit, P., Ruatta, O., Schrek, J., Tillich, J. P., Zémor, G.: Rank based Cryptography: a credible post-quantum alternative to classical crypto. In NIST 2015: Workshop on Cybersecurity in a Post-Quantum World 2015 (2015).


Kobara, K., Imai, H.: Semantically secure McEliece public-key crptosystems-conversions for McEliece PKC. Public Key Cryptography vol. 1992, pp. 19-35 (2001).


Misoczki, R., Tillich, J. P., Sendrier, N. and Barreto, P. S.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes. IEEE International Symposium on Information Theory - ISIT 2013, pp. 2069-2073 (2013).


Misoczki, R., and Barreto, P. S.: Compact McEliece keys from Goppa codes. In Selected Areas in Cryptography, pp. 376–392 (2009)

THANK YOU
VERY MUCH!