REDOG

REinforced modified Dual-Ouroboros based on Gabidulin codes

Jon-Lark Kim

Feb. 24, 2023

CEO of DeepHelix Co.
Prof. of Sogang University, S. Korea

Contents

REDOG

1. Preliminaries

Rank metric codes

Let q be a prime power and $\mathbb{F}_{q^m}$ be the finite field with q^m elements. Consider a basis $\{\beta_1, ..., \beta_m\}$ of $\mathbb{F}_{q^m}$ over the base field $\mathbb{F}_q$.

The definition of rank metric linear code and its rank is as follows.

Rank metric linear code

An $[n, k]$ linear code of length n and dimension k is a linear subspace $\mathcal{C}$ of the vector space $\mathbb{F}_{q^m}^n$, i.e. $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$. Let $l \leq k$, then an $[n, l]$ linear subcode $\mathcal{C}'$ is an $[n, l]$ linear code such that $\mathcal{C}' \subseteq \mathcal{C}$.

Rank of rank metric code

Let $\mathbf{x} = (x_1, ..., x_n) \in \mathbb{F}_{q^m}^n$. For each $1 \leq j \leq n$, $x_j = \sum_{i=1}^{m} c_{ij}\beta_i$ where $c_{ij} \in \mathbb{F}_q$. The rank of $\mathbf{x}$ in $\mathbb{F}_q$, denoted by $\mathrm{rk}(\mathbf{x})$ is defined as $\mathrm{rk}(\mathbf{x})$=$\mathrm{rk}(X)$ where $X = [c_{ij}] \in \mathbb{F}_q^{m \times n}$.

Circulant matrix

Let $\mathbf{x} = (x_0, ..., x_{n-1}) \in \mathbb{F}_{q^m}^n$. The circulant matrix $\mathrm{Cir}_n(\mathbf{x})$ induced by $\mathbf{x}$ is defined as

$$\mathrm{Cir}_n(\mathbf{x}) = \left[x_{i-j \pmod n}\right]_{ij} = \begin{bmatrix} x_0 & x_{n-1} & \cdots & x_1 \\ x_1 & x_0 & \cdots & x_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{bmatrix}$$

The $k \times n$-partial circulant matrix induced by $\mathbf{x}$, denoted by $\mathrm{Cir}_k(\mathbf{x})$ is defined as the first k rows of $\mathrm{Cir}_n(\mathbf{x})$.

Partial cyclic code

An $[n, k]$-partial cyclic code $\mathrm{PC}_{n,k}[\mathbf{x}]$ generated by $\mathbf{x} \in \mathbb{F}_{q^m}^n$ is a linear code with generator matrix $\mathrm{Cir}_k(\mathbf{x})$

Moore matrix

Denote $[l] = q^l$ as the lth Frobenius power for an integer l. A matrix $G = [G_{ij}] \in \mathbb{F}_{q^m}^{k \times n}$ is called a *Moore matrix* induced by $\mathbf{g}$ if there exists a vector $\mathbf{g} = (g_1, ..., g_n) \in \mathbb{F}_{q^m}^n$ such that the ith row of G is equal to $\mathbf{g}^{[i-1]} = (g_1^{[i-1]}, ..., g_n^{[i-1]})$ for $1 \leq i \leq k$, i.e., G is of the form

$$G = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^{[1]} & g_2^{[1]} & \cdots & g_n^{[1]} \\ \vdots & \vdots & \vdots & \vdots \\ g_1^{[k-1]} & g_2^{[k-1]} & \cdots & g_n^{[k-1]} \end{bmatrix}. \tag{1}$$

Similarly, we define $G^{[l]} = \left[G_{ij}^{[l]} \right]$. For any set $S \subset \mathbb{F}_{q^m}^n$, we denote $S^{([l])} = \{\mathbf{s}^{[l]} | \mathbf{s} \in S\}$

Then we can define a Gabidulin codes as follow.

Gabidulin codes

Let $\mathbf{g} \in \mathbb{F}_{q^m}^n$ with $\mathrm{rk}(\mathbf{g}) = n \leq m$. The $[n, k]$ Gabidulin code $\mathrm{Gab}_{n,k}(\mathbf{g})$ over $\mathbb{F}_{q^m}^n$ of dimension k with generator vector $\mathbf{g}$ is the code generated by a Moore matrix G induced by $\mathbf{g}$ in the form of Moore matrix (1).

Theorem

There exists a Moore Matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$ such that H is a parity-check matrix of a Gabidulin code. In other words, the dual of a Gabidulin code also a Gabidulin code.

The error-correcting capability of $\mathrm{Gab}_{n,k}(\mathbf{g})$ is $r = \left\lfloor \frac{n-k}{2} \right\rfloor$. There exist efficient decoding algorithms for Gabidulin codes which are able to correct error up to rank r.

We will introduce about r-Frobenius weak.

r-Frobenius weak

Let C be an $[n, k]$-linear code. We say that C is r-Frobenius weak if for some s relatively prime to m and for a generic $\mathbf{e} \in \mathbb{F}_{q^m}^n$ of rank r, the space U spanned by the elements of rank one in $C_{\text{ext}} = \sum_{i=0}^{r-1} \left(C + \langle \mathbf{e} \rangle_{\mathbb{F}_{q^m}}\right)^{[si]}$, fulfills $C \cap U = \{0\}$.

The algorithm of r-Frobenius weak attack is as follow.

Algorithm : **FrobeniusWeakAttack**

Data : $\mathbf{y} = \mathbf{m}G_{\mathsf{pub}} + \mathbf{e}$ (a ciphertext where $\mathbf{m}$ is the plaintext), the public key $\mathsf{pk} = G_{\mathsf{pub}}$ with parameter $r = \mathrm{rk}(\mathbf{e})$

Result : The plaintext $\mathbf{m}$

1 Construct the matrix

$$G_{\mathsf{pub,ext}} = \begin{bmatrix} G_{\mathsf{pub}} \\ \mathbf{y} \\ \vdots \\ G_{\mathsf{pub}}^{[r-1]} \\ \mathbf{y}^{[r-1]} \end{bmatrix}.$$

2 Compute the space $\mathcal{U}$ generated by the elements of rank one in $\mathcal{C}_{\mathsf{ext}} = \langle G_{\mathsf{pub,ext}} \rangle_{\mathbb{F}_{q^m}}$.

3 Compute $u = \dim_{\mathbb{F}_{q^m}}(\mathcal{U})$.

4 **if** $u \leqslant n - k$ **then**

5 Compute a parity-check matrix $H_U \in \mathbb{F}_q^{(n-u)\times n}$ for $\mathcal{U}$.

6 Solve $\mathbf{y}(H_U)^T - \mathbf{m}[G_{\mathsf{pub}}(H_U)^T]$ for $\mathbf{m}$,

7 **return** $\mathbf{m}$.

8 **else**

9 **return** $\perp$

2. REDOG scheme

REDOG is a reinforced version of the modified Dual-Ouruboros with Gabidulin (DO.Gab-PKE)[1]. We explain how to select the invertible matrix S.

Selection of secret key S

1. If $S \in \mathrm{GL}_{n-k}(\mathbb{F}_{q^m})$ and F is in echelon form, then the decryption algorithm of the modified DO.Gab-PKE is incorrect.
2. If $S \in \mathrm{GL}_{n-k}(\mathbb{F}_q)$, then the decryption algorithm of the modified DO.Gab-PKE can be performed correctly.

[1] J.-L. Kim, Y.-S. Kim, L.E. Galvez, M.J. Kim, A modified Dual-Ouroboros public-key encryption using Gabidulin codes. Appl. Algebra Eng. Commun. Comput. 32, 147—156, (2021).

To overcome this, we employs Loidreau's approach[2]to consider matrix S^{-1} over some λ-dimensional subspace $\Lambda \subset \mathbb{F}_{q^m}$.

Take S^{-1} as an $(n-k) \times (n-k)$ invertible matrix over Λ, where Λ is a λ-dimensional subspace of $\mathbb{F}_{q^m}$ which contains the element 1

Take the error $\mathbf{e}$ as a random vector of rank $t \leq \left\lfloor \frac{r}{\lambda} \right\rfloor$.

From the decryption process, since $\mathrm{rk}(\mathbf{e}) = t$ and Λ is a λ-dimensional subspace of $\mathbb{F}_{q^m}$, we have $\mathrm{rk}(\mathbf{e}') = \lambda t \leq r$.

Thus, the decoding algorithm Φ_H can recover $\mathbf{e}'$ correctly.

[2]P. Loidreau, A new rank metric codes based encryption scheme, 8th International Conference on Post-Quantum Cryptography, PQCrypto 2017, May 2017, Utrecht, France.

2. REDOG scheme - Setup

We describe key generation, encryption, and decryption of REDOG as follows.

Setup

Generate global parameters with integers m, n, l, r, k such that $l < n$ and $\lambda t \leq r \leq \left\lfloor \frac{n-k}{2} \right\rfloor$.

Output parameters $= (m, n, l, k, r, \lambda, t)$.

Key generation

Let $[H_1 H_2]$ be a parity check matrix for a $[2n-k, n]$ Gabidulin code $\mathcal{C}$ over $\mathbb{F}_{q^m}$, where $H_2 \in \mathrm{GL}_{n-k}(\mathbb{F}_{q^m})$. Let Φ_H be an efficient decoding algorithm for $\mathcal{C}$ with error correcting capability of $r = \left\lfloor \frac{n-k}{2} \right\rfloor$. Let $\mathcal{H}$ be a hash function from $\mathbb{F}_{q^m}^{2n-k}$ to $\mathbb{F}_{q^m}^{l}$.

Generate a generator matrix G for a random $[n, l]$ code over $\mathbb{F}_{q^m}$. Generate a random $n \times n$ isometric matrix P.

Generate a random λ-dimensional subspace, $\Lambda \subset \mathbb{F}_{q^m}$ such that $1 \in \Lambda$.

Generate a random $(n-k) \times (n-k)$ invertible matrix $S^{-1} \in \mathrm{GL}_{n-k}(\Lambda)$.

Output public key and secret key pair

$\mathsf{pk} = (G, F = GP^{-1}H_1^T[H_2^T]^{-1}S), \mathsf{sk} = (P, H, S, \Phi_H)$.

Encryption

Let $\mathbf{m} \in \mathbb{F}_{q^m}^{l}$ be the plaintext message to be encrypted. Generate randomly vector $\mathbf{e} = (e_1, e_2) \in \mathbb{F}_{q^m}^{2n-k}$ such that $\mathrm{rk}(\mathbf{e})=t$, $e_1 \in \mathbb{F}_{q^m}^{n}$ and $e_2 \in \mathbb{F}_{q^m}^{n-k}$. Let $\mathbf{m}' = \mathbf{m} + \mathcal{H}(\mathbf{e})$. Compute $c_1 = \mathbf{m}'G + e_1, c_2 = \mathbf{m}'F + e_2$.

Output ciphertext $\mathbf{c} = (c_1, c_2)$.

Decryption

Compute

$$
\begin{aligned}
&c_1 P^{-1} H_1^T - c_2 S^{-1} H_2^T \\
&\quad = \mathbf{m}' G P^{-1} H_1^T + e_1 P^{-1} H_1^T - \mathbf{m}' G P^{-1} H_1^T [H_2^T]^{-1} S S^{-1} H_2^T - e_2 S^{-1} H_2^T \\
&\quad = e_1 P^{-1} H_1^T - e_2 S^{-1} H_2^T \\
&\quad = (e_1 P^{-1}, -e_2 S^{-1}) \begin{bmatrix} H_1^T \\ H_2^T \end{bmatrix}
\end{aligned}
$$

Let $\mathbf{e}' = (e_1 P^{-1}, -e_2 S^{-1})$. Since $\mathrm{rk}(\mathbf{e}') \leq r$, apply Φ_H to obtain $\mathbf{e}'$.

Compute $e_1 = e_1 P^{-1} P$ and $e_2 = e_2 S^{-1} S$ to obtain $\mathbf{e} = (e_1, e_2)$.

Finally, solve the system $\mathbf{m}' G = c_1 - e_1$ to recover $\mathbf{m} = \mathbf{m}' - \mathcal{H}(e)$.

3. Parameter and Security

3. Parameter and Security

To implement our REDOG cryptosystem, we used the following software and hardware platforms:

- SageMATH 9.2 version
- Python 3.7.7 version
- Visual studio 2019
- 3.8GHz Intel(R) Core(TM) i7 processor with 32GB of memory

3. Parameter and Security

We present our proposed parameters for REDOG in Table 1.

The public key size is $\text{size}_{\text{pk}} = m(n + l(n-k))/8$ bytes, the secret key size is $\text{size}_{\text{sk}} = (n^2 + (3n-2k)m)/8$ bytes, and the ciphertext size is $\text{size}_{\text{ct}} = (2n-k)m/8$ bytes.

Table 1: Proposed parameters for REDOG

Instance	$(n, k, l, q, m, r, \lambda, t)$	size_{pk}	size_{sk}	size_{ct}	Security level
REDOG-1	(44,8,37,2,83,18,3,6)	14.25KB	1.45KB	0.83KB	128
REDOG-2	(58,10,49,2,109,24,3,8)	32.84KB	2.52KB	1.44KB	192
REDOG-3	(72,12,61,2,135,30,3,10)	62.98KB	3.89KB	2.23KB	256

3. Parameter and Security

The performance results of implementing the REDOG cryptosystem using the platform described above are as follows.

Table 2: Performance of REDOG

Instance	$(n, k, l, q, m, r, \lambda, t)$	$\text{KeyGen}_{\text{time}}$	Enc_{time}	Dec_{time}	Security level
REDOG-1	(44,8,37,2,83,18,3,6)	2.5 sec	0.035 sec	1.434 sec	128
REDOG-2	(58,10,49,2,109,24,3,8)	4.7 sec	0.06 sec	3.254 sec	192
REDOG-3	(72,12,61,2,135,30,3,10)	10.0 sec	0.1 sec	6.366 sec	256

3. Parameter and Security

We checked about following attacks. Detail of cost of known attacks are from[3].

1. IND − CPA security: REDOG achieves IND − CPA security.
2. Key recovery attack
3. Our plaintext recovery attack
4. Message recovery attacks
5. r-Frobenius weak attack

[3] T. S. C. Lau, C. H. Tan, T. F. Prabowo, On the security of the modified Dual-ouroboros PKE using Gabidulin codes, Appl. Algebra Eng. Commun. Comput. 32, 681–699, (2021).

We compare security of REDOG with other cryptosystems BIKE, HQC, and Classic McEliece.

Table 3: Security level and key sizes of BIKE[4]

Quantity	Size	AES-128	AES-192	AES-256
Private key	$w\lceil log_2(r)\rceil$	2,130bits	2,296bits	4,384bits
Public key	n	20,326bits	43,786bits	65,498bits
Ciphertext	n	20,326bits	43,786bits	65,498bits

[4]N. Aragon, et al. "BIKE: bit flipping key encapsulation." (2017).

We compare security of REDOG with other cryptosystems BIKE, HQC, and Classic McEliece.

Table 4: Security level and key sizes of HQC[5]

Instance	pk size	sk size	ct size
hqc-128	2,249bytes	40bytes	4,481bytes
hqc-192	4,522bytes	40bytes	9,026bytes
hqc-256	7,245bytes	40bytes	14,469bytes

[5]C. Aguilar Melchor, et al. "Hamming quasi-cyclic (HQC)" NIST PQC Round 2.4 (2018): 13.

3. Parameter and Security

We compare security of REDOG with other cryptosystems BIKE, HQC, and Classic McEliece.

Table 5: Parameters, security level and key sizes of Classic McEliece[6]

Variant	n	m	t	$k = n - mt$	pk size	sk size	Security level
mceliece6960119	6960	13	119	5413	1047KB	13.6KB	128
mceliece8192128	8192	13	128	6528	1358KB	13.75KB	256

[5]H. Singh, "Code based cryptography: Classic mceliece," arXiv preprint arXiv:1907.12754 (2019).

4. Conclusion

4. Conclusion

The parameters of REDOG are reasonably good when compared with BIKE, HQC, and Classic McEliece which is in the 4th round of NIST PQC competition.

Also HQC and BIKE algorithms have decryption failure which is a disadvantage although their key sizes are much smaller than REDOG.

Therefore, we believe that REDOG can be a strong candidate for the KPQC standardization.