JEONBUK NATIONAL UNIVERSITY

Online Documentation
20/10/2023

Comments and modification on Layered ROLLO on kPQC-forum

Chanki Kim, Jeonbuk Nat'l. Univ., Dept. of Computer Science and Artificial Intelligence
Young-Sik Kim, DGIST, Dept. of Electrical Engineering and Computer Science
Jong-Seon No, Seoul Nat'l. Univ., Dept. of Electrical Computer Engineering

Response letter to kPQC forum

Dear all,

We would like to inform you about our response to the recent analysis on the Layered ROLLO scheme. In the previous round, we modified the Layered ROLLO scheme in order to prevent the attack using PK. On the contrary, new attacks are based on the fixed location of the error polynomial $P\_(E,1)$ and $P\_(E,2)$ on the ciphertext, which can recover the error vector directly. Note that the new attack uses information set decoding from CT and PK and it is different from the previous attack using PK only. Nevertheless, both attacks actually originated from the fixed nonzero element indices by the low-degree polynomials and thus, the corresponding solutions are similarly induced.

For the modified schemes, polynomial masking on CT can be used, which makes it hard to find exact values without guessing the noise term $P_{N,C}$. Accordingly, we apply a parameter change including increased PK size compared to the parameters of layered ROLLO submission.

Note that the implementation codes are also modified and newly uploaded in the web. We also fixed some issues of constant implementation and memory leaks commented from KPQClean.

We would like to express our gratitude to the researchers for their valuable analysis and suggestions regarding the vulnerabilities in the Layered ROLLO scheme.

Sincerely,

Chanki Kim, Young-Sik Kim, and Jong-Seon No

History on Layered ROLLO Scheme

- 22/10/31: Initial version of Layered ROLLO was submitted to kPQC 1 Round.

- 23/04/10: New attacks and analysis are uploaded in kPQC-bulletin
- 23/05/19: 1st Modification on Layered ROLLO are uploaded on kPQC-bulletin.
- 23/09/05: New attacks and analysis are uploaded in kPQC-bulletin
- 23/09/22: 2nd Modification on Layered ROLLO are uploaded on kPQC-bulletin.
- 23/10/03: New attacks and analysis are uploaded in kPQC-bulletin
- 23/10/20: 3rd Modification on Layered ROLLO are uploaded on kPQC-bulletin.

Summary on new attacks on Layered ROLLO

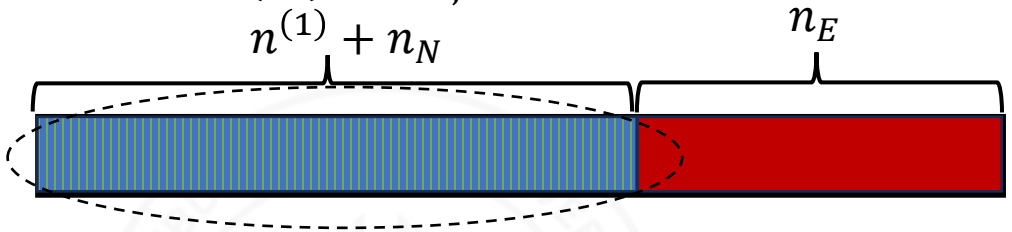
- New attacks are based on the fixed location of error polynomial $P_{E,1}$ and $P_{E,2}$ on the ciphertext
- Solution: Polynomial masking on $P_{E,2}$, which make it harder to find an exact values on $P_{E,1}$ and $P_{E,2}$ similarly to the recovery on P_I

Comment No.	Comments	Modification
1.	new attacks by fixed nonzero location on $P_{E,1}$,$P_{E,2}$	Increasing PK size and adding error polynomial on CT

Sketch of the new attack

$$P_P = \Psi(P_I) + P_{N,A}P^{(1)}$$

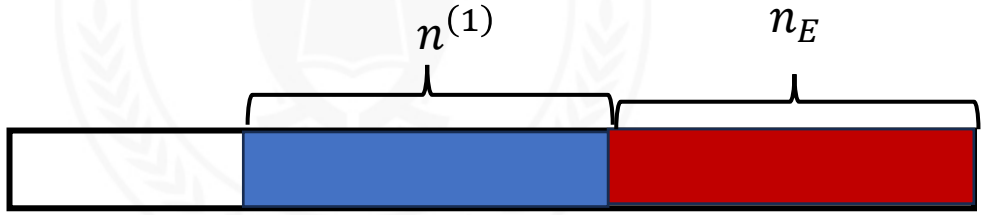
$$P_P P_{E,1} = P_O \times$$



$P_{E,1}$

$+ \quad \Psi(\mathbf{z}(X_1))$ Masked part $+$

$$P_H(P_{E,2}) = P_O \times$$

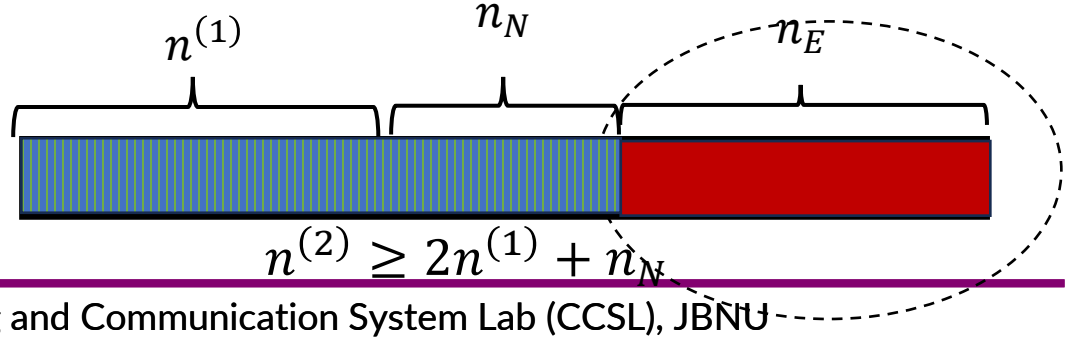


$P_{E,2}$

$=$

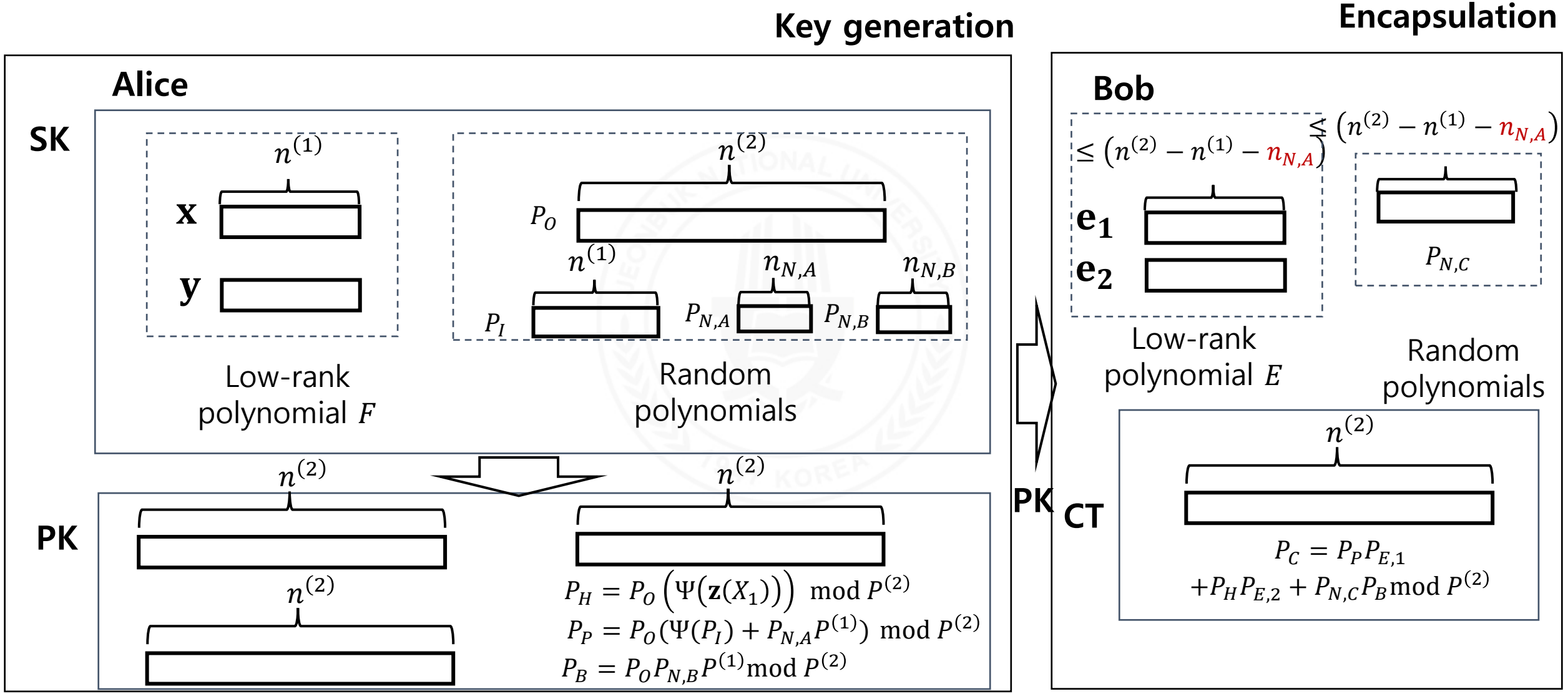
$=$

$$P_C \quad P_O \times$$



The attacker can exploit the location from the equation $P_H^{-1}P_C = P_P P_H^{-1} P_{E,1} + P_{E,2}$ and Prange algorithm

Sketch of the new scheme



Sketch of the new scheme

$P_P = P_O(\Psi(P_I) + P_{N,A}P^{(1)})$

$P_B = P_O P_{N,B} P^{(1)}$

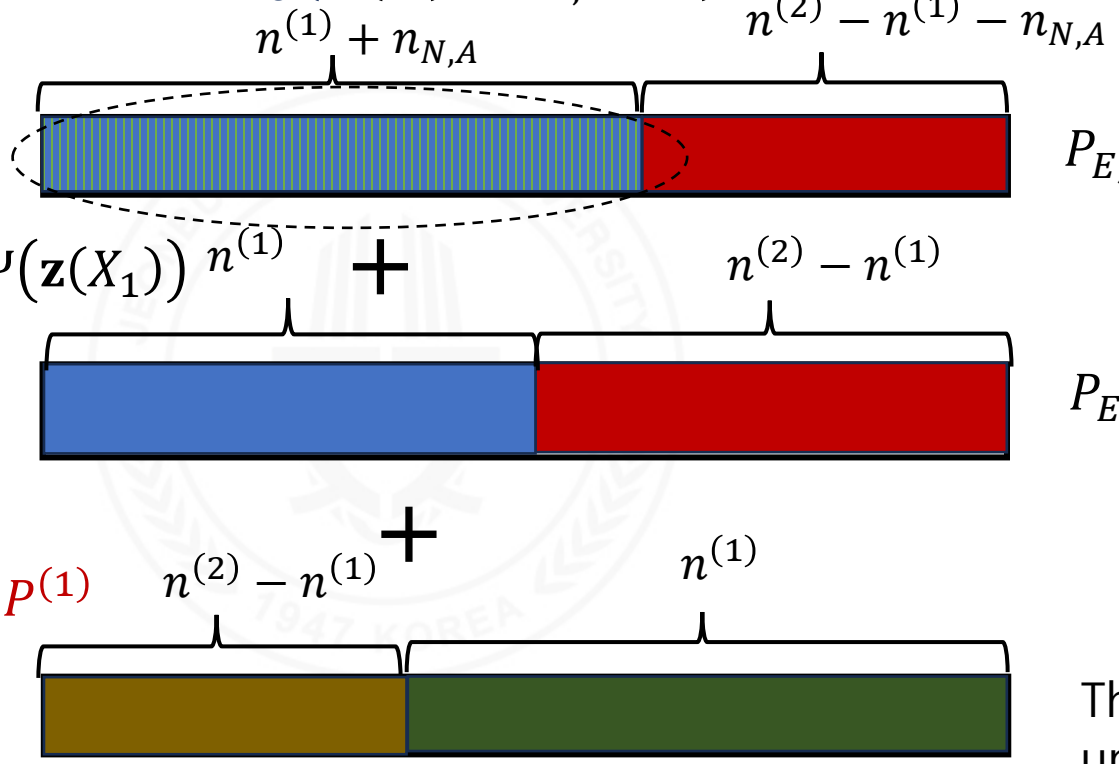
$P_P P_{E,1} = P_O \times$

$+$

$P_H(P_{E,2}) = P_O \times$

$+$

$P_B(P_{N,C}) = P_O \times$



$= P_C$

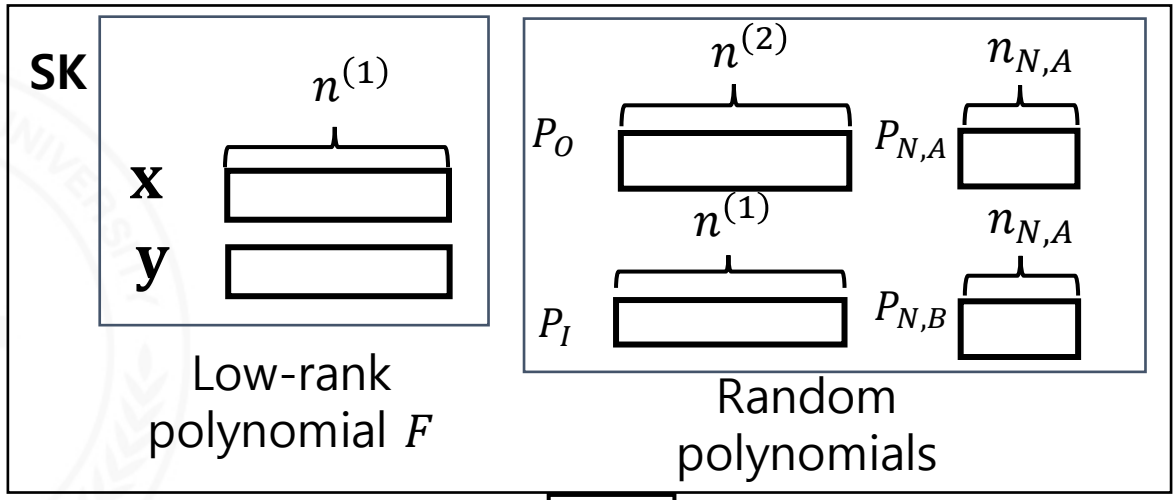
The attacker cannot find a unique support of the error vector

$P_H^{-1}P_C = P_H^{-1}P_P P_{E,1} + P_{E,2} + P_H^{-1}P_B P_{N,C}$

Modified Layered ROLLO-I: Procedures

1. Key generation

- Select two $n^{(1)}$ -degree and $n^{(2)}$-degree primitive polynomials $P^{(1)}$, and $P^{(2)}$
- Generate two random vectors $\mathbf{x}$ and $\mathbf{y}$ from the low-rank $\mathbb{F}_q$-subspace $F \in (\mathbb{F}_{q^m})^{n^{(1)}}$ with rank weight d
- Generate $(n^{(1)})$-degree, $(n_{N,A})$-degree, $(n_{N,B} = n_{N,A})$-degree and $(n^{(2)})$-degree random polynomials P_I, $P_{N,A}$, $P_{N,B}$, P_O
- Generate $\mathbf{z}$ and P_H as
  - $\mathbf{z} = (P_I \mathbf{x}^{-1} \mathbf{y}) \bmod P^{(1)}$
  - $P_H = \left(P_O \Psi(\mathbf{z}(X_1))\right) \bmod P^{(2)}$
- Finally, construct SK and PK as
  - **PK:** P_H, $P_P = P_O(\Psi(P_I) + P_{N,A}P^{(1)}) \bmod P^{(2)}$, $P_B = P_O P_{N,B} P^{(1)}$ (NOTE: We use an additional key size by P_P, which amounts to $\left\lceil \frac{n \log_2 m}{8} \right\rceil$[Byte])
  - **SK:** $\mathbf{x}$, $\mathbf{y}$, P_O, P_I, and $P^{(1)}$



$$\mathbf{z} = P_I \mathbf{x}^{-1} \mathbf{y} \quad \bmod P^{(1)}$$
$$P_H = P_O\left(\Psi(\mathbf{z}(X_1))\right) \quad \bmod P^{(2)}$$
$$P_P = P_O(\Psi(P_I) + P_{N,A}P^{(1)}) \quad \bmod P^{(2)}$$
$$P_B = P_O P_{N,B} P^{(1)} \quad \bmod P^{(2)}$$

To public, including Bob P_H, P_P

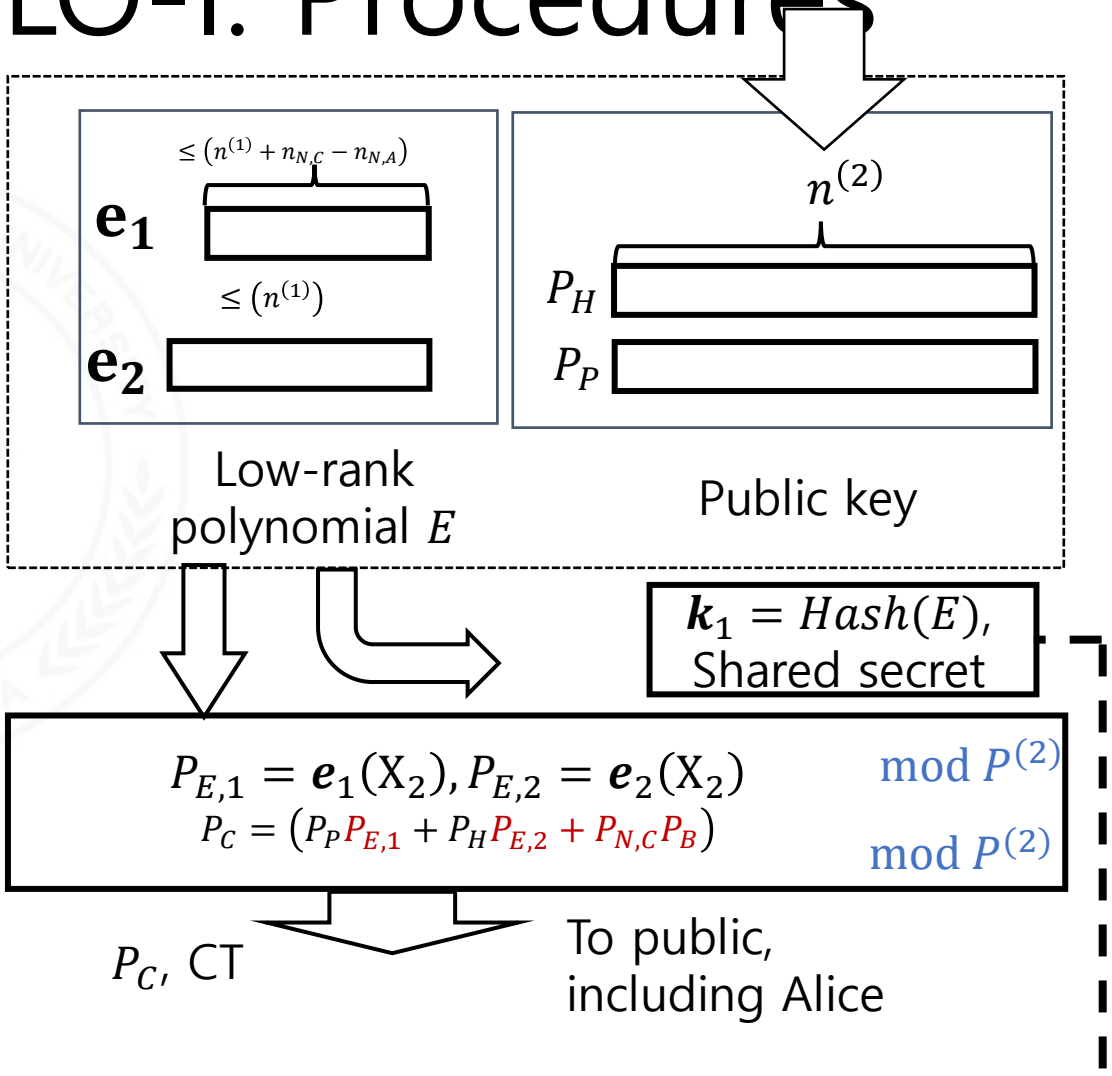
Modified Layered ROLLO-I: Procedures

2. Encapsulation: Adding polynomial masking on the error vector $P_{N,C}P_B$

- Generate low-rank $\mathbb{F}_q$-subspace $E \in \left(\mathbb{F}_{q^m}\right)^{n^{(2)}}$ with rank weight r with the last $n^{(2)} - n^{(1)} + n_{N,A}$ nonzero elements
- Generate two error vectors $\boldsymbol{e}_1, \boldsymbol{e}_2 \in E$ and corresponding $\left(n^{(2)} - n^{(1)} - n_{N,A}\right)$-degree polynomials
- $P_{E,1} = \mathbf{e}_1(X_2)$ and $P_{E,2} = \mathbf{e}_2(X_2)$
- Obtain CT polynomial P_C as

$P_C = \left(P_P P_{E,1} + P_H P_{E,2} + P_{N,C}P_B\right) \bmod P^{(2)}$ for $(n^{(2)} - n^{(1)} - n_{N,A})$ –degree polynomials $P_{N,C}$

- Obtain $\mathbf{k}_1 = Hash(E)$ to have a shared secret (SS)

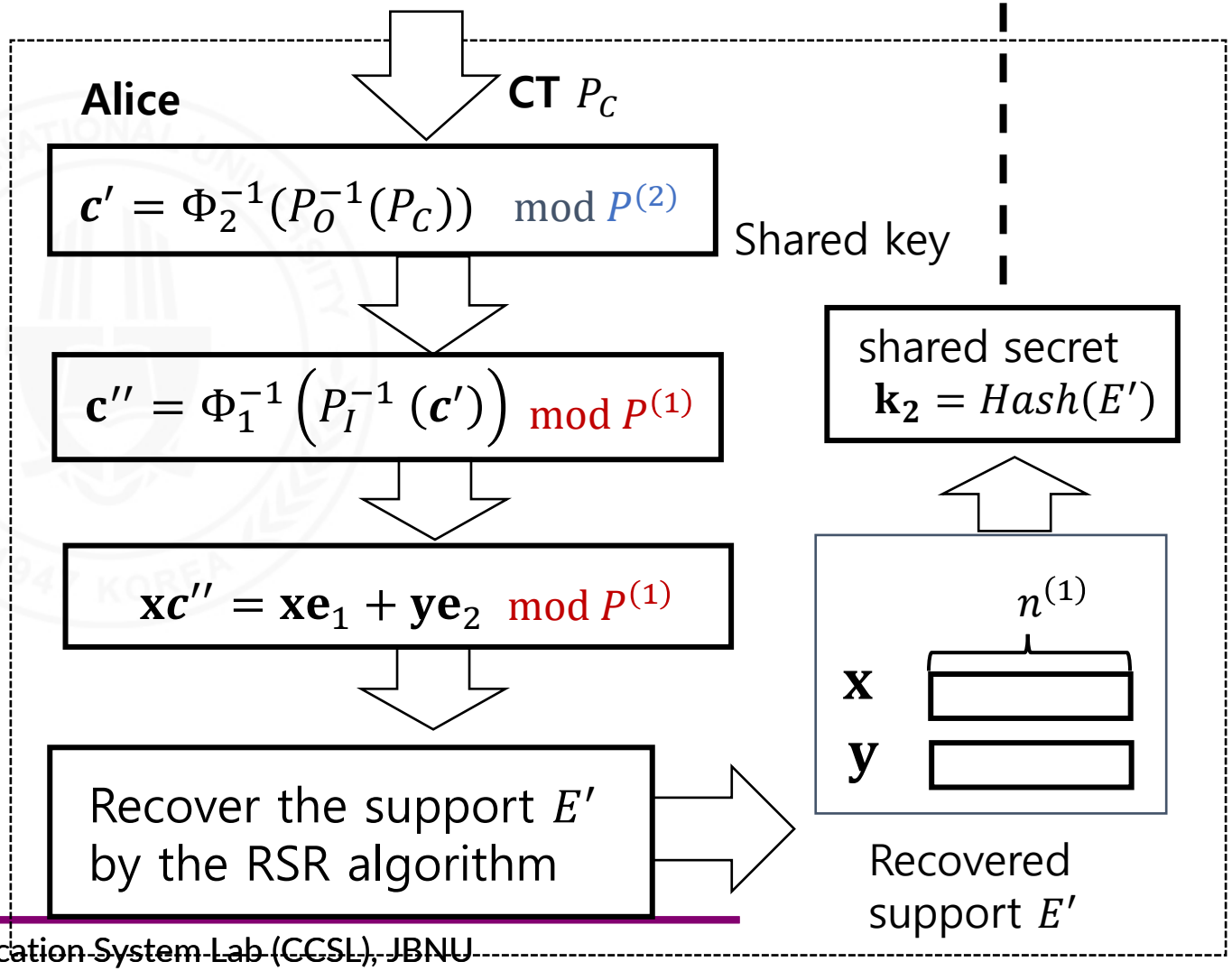


Modified Layered ROLLO-I: Procedures

3. Decapsulation: Maintained

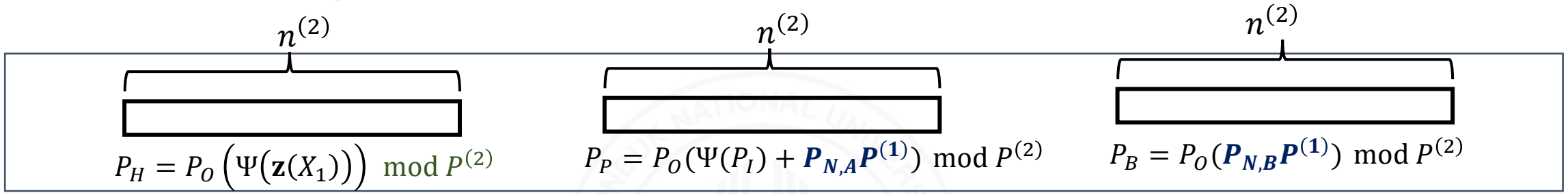
- Obtain the codeword $\mathbf{xc}'' = \mathbf{xe}_1 + \mathbf{ye}_2 \bmod P^{(1)}$ from P_C by
  - $\boldsymbol{c}' = \Phi_2^{-1}(P_O^{-1}(P_C)) \bmod P^{(2)}$
  - $\boldsymbol{c}'' = \Phi_1^{-1}\left(P_I^{-1}(\boldsymbol{c}') \bmod P^{(1)}\right)$
  - $\mathbf{xc}'' \bmod P^{(1)} = \mathbf{xe}_1 + \mathbf{ye}_2 \bmod P^{(1)}$

- From $\mathbf{xc}'' \bmod P^{(1)}$, recover the support E' by the RSR algorithm
- Derive shared key $\mathbf{k_2} = Hash(E')$



Summary on the new attacks in 2nd Round

PK



1. The 1st direct attack: $P_P^{-1}P_H \bmod P^{(1)} = \mathbf{x}^{-1}\mathbf{y} \bmod P^{(1)}$
→ Fixed by introducing two independent moduli (2023. 5)
2. The 2nd direct attack 2: deriving determined equation from $\left(P_P^{-1}P_H \bmod P^{(2)}\right)P_P \bmod P^{(2)} = \mathbf{x}^{-1}\mathbf{y} \bmod P^{(2)}$
→ Fixed by using polynomial masking on $P_P \rightarrow P_P + P_{N,A}P^{(1)}$ with SL $S_D = O\left(q^{m(n_{N,A})}, q^{m(2n^{(1)}-n^{(2)}+n_{N,A})}\right)$ (2023. 9)

CT

$$P_C = \left(P_P P_{E,1} + P_H P_{E,2} + P_{N,C}P_B\right) \bmod P^{(2)}$$

3. RSD attack for CT on small rank weight r on error vectors $P_{E,1}$, $P_{E,2}$
→ Fixed by increasing the rank weight (2023. 4, 2023. 9), with SL S_R
4. New ciphertext attack: deriving determined equation from $\left(P_P^{-1}P_H \bmod P^{(2)}\right)P_C \bmod P^{(2)} \rightarrow P_{E,1}, P_{E,2}$
→ Managed by using polynomial masking on $P_C \rightarrow P_{N,C}P_B$ (2023. 10)

On the new attacks

- **Scenario 1:**

Among P_H, P_P, P_B, consider the equation between P_H and P_P(Actually, the other are also similar)

$$(\Psi(P_I) + \boldsymbol{P_{N,A}P^{(1)}})\left(P_P^{-1}P_H \bmod P^{(2)}\right)\bmod P^{(2)} = \boldsymbol{z} \bmod P^{(2)}, \rightarrow \mathbf{v}M = \mathbf{w} \bmod P^{(2)},$$

$$\boldsymbol{z} \bmod P^{(2)}\left(P_H^{-1}P_P \bmod P^{(2)}\right)\bmod P^{(2)} = (\Psi(P_I) + \boldsymbol{P_{N,A}P^{(1)}}), \rightarrow \mathbf{w}M^{-1} = \mathbf{v} \bmod P^{(2)}$$

We know that

$$|\mathbf{v}| = n^{(1)} + n_{N,A}, (\text{unknown})\ n^{(2)} - |\mathbf{w}| = n^{(2)} - n^{(1)}(\text{constraints}).$$

$$|\mathbf{w}| = n^{(1)}, (\text{unknown})\ \ n^{(2)} - |\mathbf{v}| = n^{(2)} - n^{(1)} - n_A(\text{constraints}).$$

Thus, we still have $\left(n^{(1)} + n_{N,A}\right) - (n^{(2)} - n^{(1)})$ undetermined elements, and the corresponding complexity(SL) is

$$S_W = O\left(q^{(2n^{(1)} - n^{(2)} + n_{N,A})m}\right)$$

On the new attacks

- **Scenario 2:**

- We had an assumption that $O\left(q^{2n^{(1)}-n^{(2)}+n_{N,A}}\right) > SL$ by scenario 1.
- For the scenario 2, we have $P_H^{-1}P_C$($P_P^{-1}P_C, P_B^{-1}P_C$ can be considered similarly)

$$P_H^{-1}P_C = P_P P_H^{-1} P_{E,1} + P_B P_H^{-1} P_{N,C} + P_{E,2} \bmod P^{(2)}$$
$$\rightarrow \mathbf{s} = G_1\mathbf{e}_1' + G_2\mathbf{e}_2' + \mathbf{e}_3' \bmod P^{(2)}, \mathbf{e}_3' = \mathbf{s} - (G_1\mathbf{e}_1' + G_2\mathbf{e}_2')$$

for $G_1 = P_P P_H^{-1}, G_2 = P_B P_H^{-1}$ and $|\mathbf{e}_1'| = |\mathbf{e}_2'| = |\mathbf{e}_3'| = n^{(2)} - n^{(1)} - n_{N,A}$.

- The next step is to find $(G_1\mathbf{e}_1' + G_2\mathbf{e}_2')$, but we did not find an unique pair $(\mathbf{e}_1', \mathbf{e}_2')$ satisfying

$$\mathbf{s}_{[n^{(1)}+n_{N,A}+1, n^{(2)}]} = (G_1\mathbf{e}_1' + G_2\mathbf{e}_2')_{[n^{(1)}+n_{N,A}+1, n^{(2)}]}.$$

- The required complexity to find a correct $(\mathbf{e}_1', \mathbf{e}_2')$ is $S_C = O\left(q^{(n^{(2)}-n^{(1)}-n_{N,A})m}\right)$

Parameter selection for new-LROLLO

In summary, the new attack scenarios have SL either of

1) $\mathrm{S}_R = O\left(RSD\ attacks\ for\ CT\ with\ long\ codelength\ n^{(2)}\right)$, or

2) $S_D = O\left(q^{(2n^{(1)}-n^{(2)}+n_{N,A})m} \times \mathrm{S}_{R,S}\left(RSD\ attacks\ for\ CT\ with\ short\ codelength\ n^{(2)}\right)\right)$

(e.g.) For the second attacks on d, we have

$$= O\left(q^{(2n^{(1)}-n^{(2)}+n_{N,A})m} \times \left(n^{(1)}m\right)^3 q^{d\left\lfloor\frac{m}{2}\right\rfloor-m-n^{(1)}}\right)$$

3) $\mathrm{S}_C = O\left(q^{(n^{(2)}-n^{(1)}-n_{N,A})m}\right)$(New ciphertext attack)

For the existing attacks on r, d from the guessed P_I, P_O, the required complexity for finding the correct P_I, P_O are $O\left(q^{mn^{(1)}}\right)$ and $O\left(q^{mn^{(2)}}\right)$, respectively, which is sufficiently high compared to the other scenario even for low d.

New Suggested Parameters

- **Suggested parameter for the existing ROLLO-I**

Name	q	n	m	r	d	SL (Conv. Gen)	SL (Conv. Stru.)	DFR	Pk size	ct size
ROLLO-I-128	2	83	67	**7**	8	207.648687	155.3233859	-27	696	696
ROLLO-I-192	2	97	79	**8**	8	278.968813	178.7110808	-33	958	958
ROLLO-I-256	2	113	97	**9**	9	383.623107	266.7602754	-32	1371	1371

- **Suggested parameter for the first submission(2022.10)**

Name	q	$n^{(1)}$	$n^{(2)}$	n_N	n_I	m	r	d	SL. (S_R)	SL. (S_C)	DFR	Pk size	ct size
LROLLO-128	2	37	**74**	N.A.	**2**	67	**2**	2	27.89	Broken	-31	1240	620
LROLLO-192	2	43	**86**	N.A.	**2**	79	**3**	2	42.38	Broken	-35	1857	979
LROLLO-256	2	53	**106**	N.A	**2**	97	**3**	2	44.37	Broken	-38	2571	1286

- **Suggested parameter for the new Layered ROLLO-I(2023.10),**

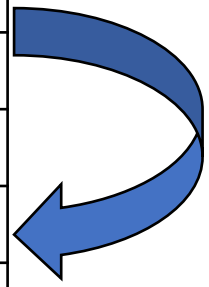
Name	q	$n^{(1)}$	$n^{(2)}$	$n_{N,A}$	m	r	d	SL. (S_R)	SL. (S_W)	SL. (S_C)	DFR	Pk size	ct size
New-128	2	37	**61**	**4**	67	**7**	2	**154.6**	**≥268**	**1608**	**-23**	1533	511
New-192	2	43	**71**	**4**	79	**9**	2	**199.68**	**≥ 316**	**2212**	**-25**	2106	702
New-256	2	53	**103**	**4**	97	**12**	2	**273.4**	**≥ 388**	**4850**	**-29**	3750	1250

- From the larger $n^{(2)}$, the PK and CT size becomes larger than the first submission.

Code performance analysis

- **Performance measure:** The number of CPU processing cycle for key generation, encapsulation, and decapsulation on the simulation environments on CPU
  - ➢ The new schemes still have good performance compared to the firstly proposed layered ROLLO-I

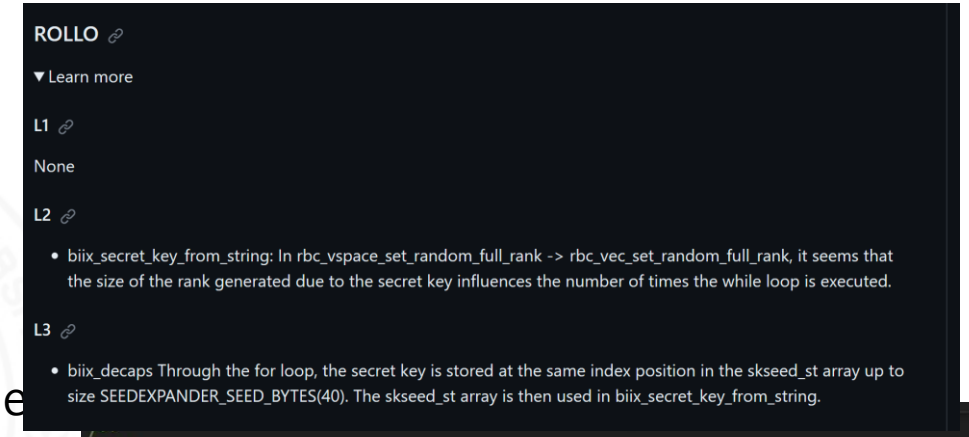
Instance	Key generation	Encapsulation	Decapsulation	Total
ROLLO-I-128	597,908	80,334	980,128	1,658,370
ROLLO-I-192	577,567	69,266	1,292,923	1,939,756
ROLLO-I-256	787,431	86,257	2,199,476	3,073,164
New-128	362,315	156,007	682,282	1,200,604
New-192	352,268	158,901	668,341	1,179,510
New-256	408,562	157,835	1,126,821	1,693,218



30-50% cycle reduction

On the improvements on Implementation Codes

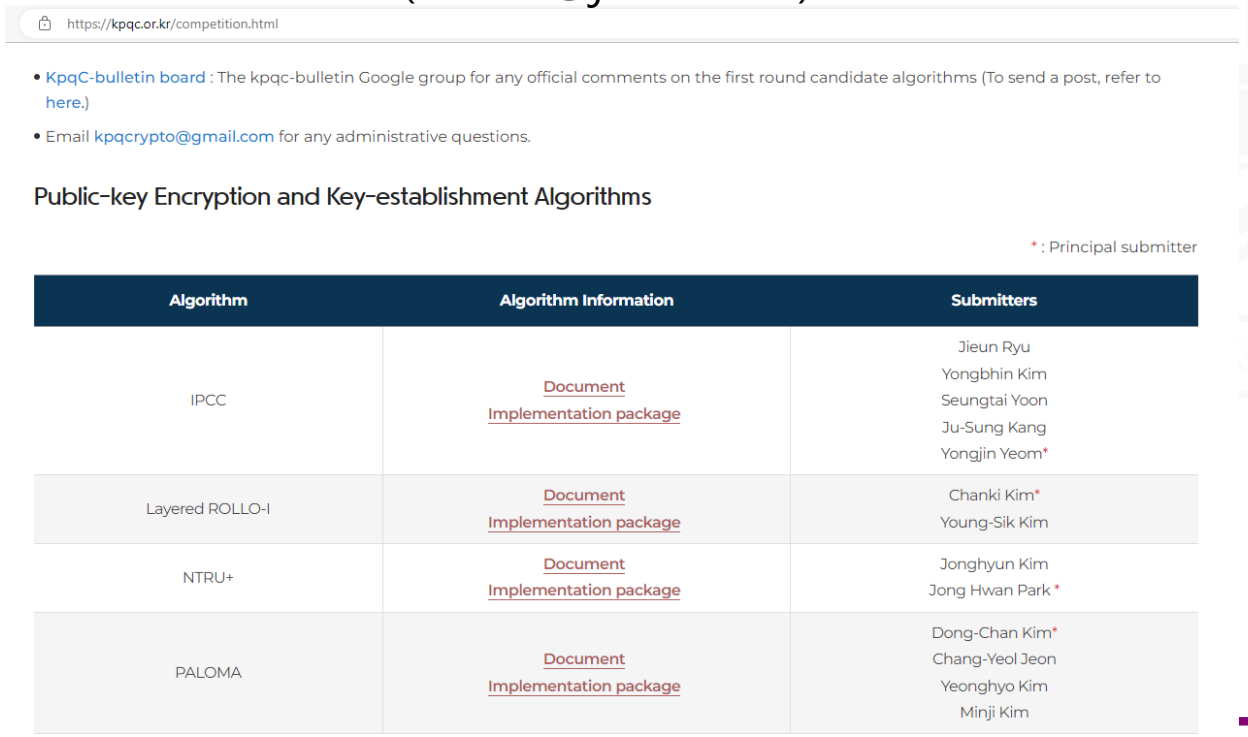
- We refined source codes based on the KPQClean comments
- **For constant-time implementation**
  - L2
    - In the function low-rank vector is generated by adding a random vector and checking if it increase the rank weight, which makes a variable number of iteration.
    - In fact, the case when dependent random vector is generated is rarely occurred
    - For the constant implementation, we added an iteration margin called WHILE_MAX
  - L3
    - Secret key is operated within our intention, which seems to be operated with constant-time.
- **For memory leaks:** We find some variables which is not freed in the function and fixed it




```
/**
 * \fn void rbc_vec_set_random_full_rank_with_one(random_source* ctx, rbc_vec o, uint32_t size)
 * \brief This function sets a vector with random values using the NIST seed expander. The vecto
 *
 * \param[out] ctx random source
 * \param[out] o rbc_vec
 * \param[in] size Size of the vector
 */
void rbc_vec_set_random_full_rank_with_one(random_source* ctx, rbc_vec o, uint32_t size) {
  int32_t rank_max = RBC_FIELD_M < size ? RBC_FIELD_M : size;
  int32_t rank = -1; int32_t cnt = 0;
  int32_t WHILE_MAX = size + 2;
  while(rank != rank_max || cnt < WHILE_MAX)
  {
    rbc_vec_set_random(ctx, o, size - 1);
    rbc_elt_set_one(o[size - 1]);
    rank = rbc_vec_get_rank(o, size);
    cnt++;
  }
}
```

Further Information

- KPQC Hompage: https://kpqc.or.kr/competition.html(Documents and source code for 1 round submission)
- Cryptography Arxiv: Layered ROLLO-I: Faster rank-metric code-based KEM using ideal LRPC codes (iacr.org)
- Kpqc-Bulletin: https://groups.google.com/g/kpqc-bulletin/
- Layered-ROLLO-I Homepage(https://sites.google.com/view/ccsl-jbnu/research/layered-rollo) or contact me (carisis@jbnu.ac.kr)



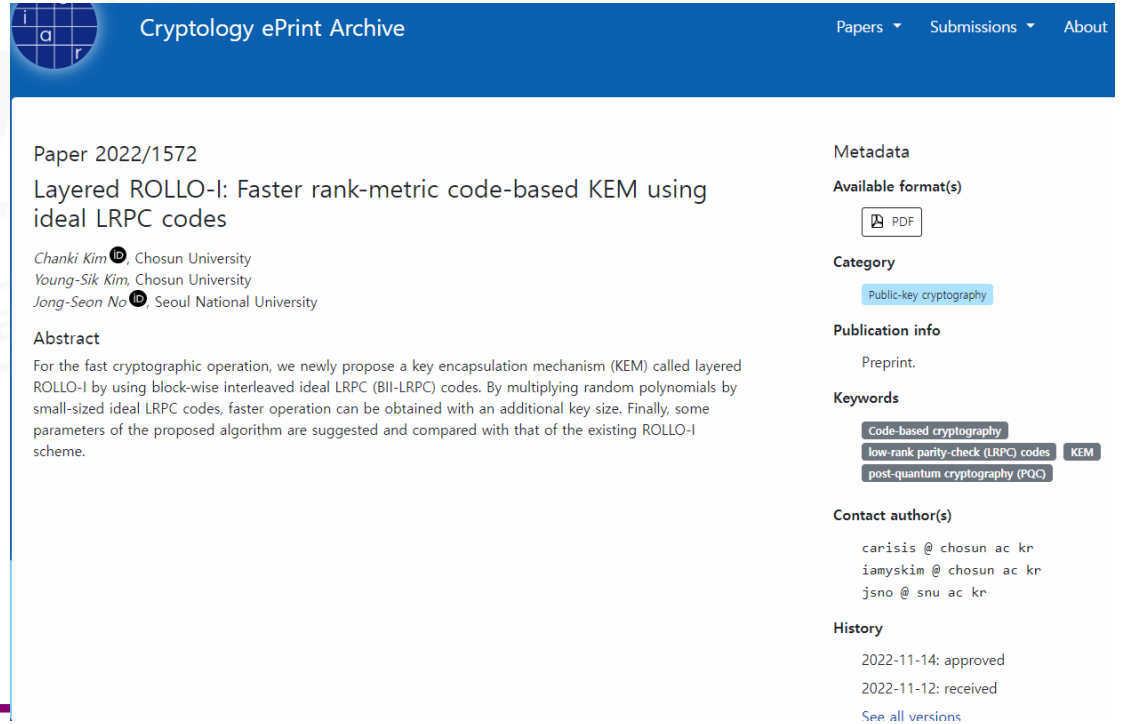
https://kpqc.or.kr/competition.html

- KpqC-bulletin board : The kpqc-bulletin Google group for any official comments on the first round candidate algorithms (To send a post, refer to here.)
- Email kpqcrypto@gmail.com for any administrative questions.

Public-key Encryption and Key-establishment Algorithms

* : Principal submitter

Algorithm	Algorithm Information	Submitters
IPCC	Document Implementation package	Jieun Ryu Yongbhin Kim Seungtai Yoon Ju-Sung Kang Yongjin Yeom*
Layered ROLLO-I	Document Implementation package	Chanki Kim* Young-Sik Kim
NTRU+	Document Implementation package	Jonghyun Kim Jong Hwan Park *
PALOMA	Document Implementation package	Dong-Chan Kim* Chang-Yeol Jeon Yeonghyo Kim Minji Kim



Cryptology ePrint Archive

Papers Submissions About

Paper 2022/1572

Layered ROLLO-I: Faster rank-metric code-based KEM using ideal LRPC codes

Chanki Kim, Chosun University
Young-Sik Kim, Chosun University
Jong-Seon No, Seoul National University

Abstract

For the fast cryptographic operation, we newly propose a key encapsulation mechanism (KEM) called layered ROLLO-I by using block-wise interleaved ideal LRPC (BII-LRPC) codes. By multiplying random polynomials by small-sized ideal LRPC codes, faster operation can be obtained with an additional key size. Finally, some parameters of the proposed algorithm are suggested and compared with that of the existing ROLLO-I scheme.

Metadata

Available format(s)

PDF

Category

Public-key cryptography

Publication info

Preprint.

Keywords

Code-based cryptography
low-rank parity-check (LRPC) codes
KEM
post-quantum cryptography (PQC)

Contact author(s)

carisis @ chosun ac kr
iamyskim @ chosun ac kr
jsno @ snu ac kr

History

2022-11-14: approved
2022-11-12: received

See all versions

Thank you