Online Documentation
05/11/2023

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 4th analysis on the Layered ROLLO scheme. As in the response, the new scheme can successfully prevent the new attack, which is achieved by using new PK regarding the inner modulus P^{(1)}. However, The new PK can be used to additionally reveal some information and new attacks can break the LROLLO-128 and LROLLO-192, where the degree of error polynomials, represented by difference between n^{(1)} and n^{(2)}, are small

In the revised scheme, we increased the parameter of n^{(2)}, where the n^{(2)} increased nearly to 2n^{(1)} for LROLLO-128 and LROLLO-192. However, KEM scheme and parameter of LROLLO-256 are unchanged.

In addition, we notice commented issues on the implementation on inner modulus P^{(1)} , where P^{(1)} is declared as a fixed polynomial as in the initial source code RBC in the ROLLO-I. In this case, the corresponding SL can be lowered from the low-degree polynomial $\frac{P_B}{P^{(1)}}$ when modulus polynomial $P^{(1)}$ is known by attacker. Therefore, We are trying to modify it with minimizing the additional processing speed (i.e. processing cycle) until the end of 2nd submission. There seems to be some options when considering the performance optimization.

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.
- 23/10/22: New attacks and analysis are uploaded in kPQC-bulletin
- 23/11/03: 4th Modification on Layered ROLLO are uploaded on kPQC-bulletin.

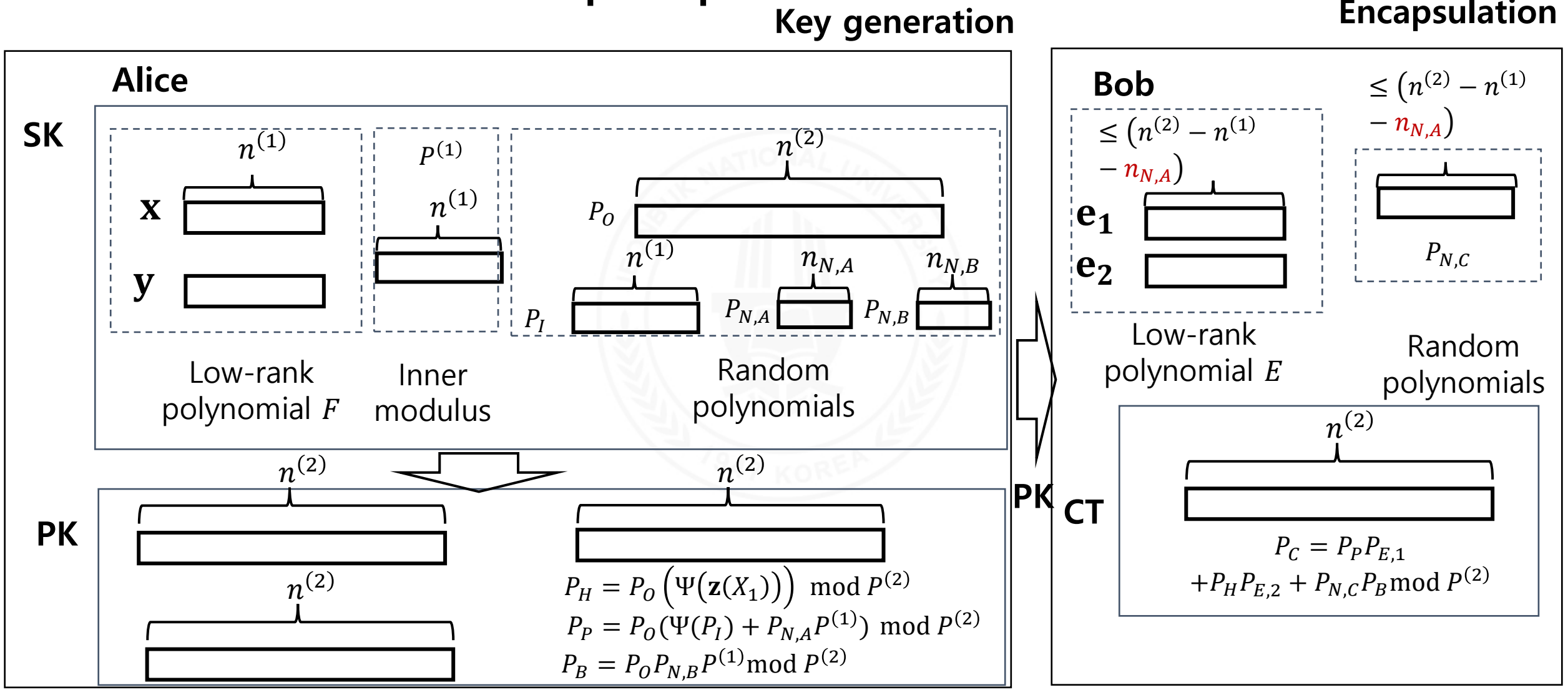
- 23/02/23(Planned): Modified version of Layered ROLLO was submitted to kPQC 2 Round.

Summary on new attacks on Layered ROLLO

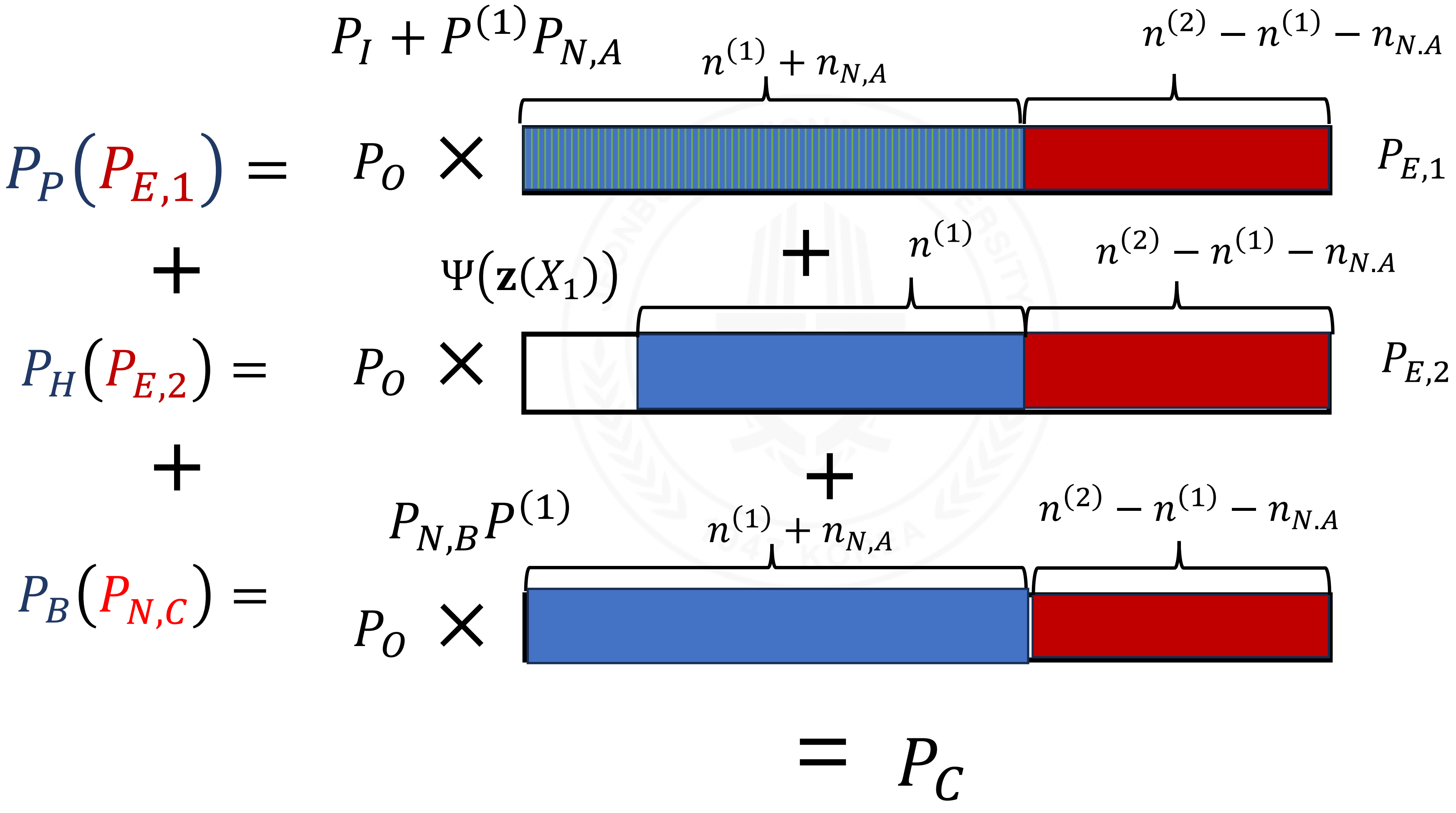
- New attacks are based on the fixed location of error polynomial $P_{E,1}$ and $P_{E,2}$ on the ciphertext
- Solution: Polynomial masking on $P_{E,2}$, which make it harder to find an exact values on $P_{E,1}$ and $P_{E,2}$ similarly to the recovery on P_I

Comment No.	Comments	Modification
1.	New attacks by combining three linear equations by each PK and CT	Modifying the code parameter of LROLLO-128 and LROLLO-192 where the proposed attack is not applied

Sketch of the proposed scheme



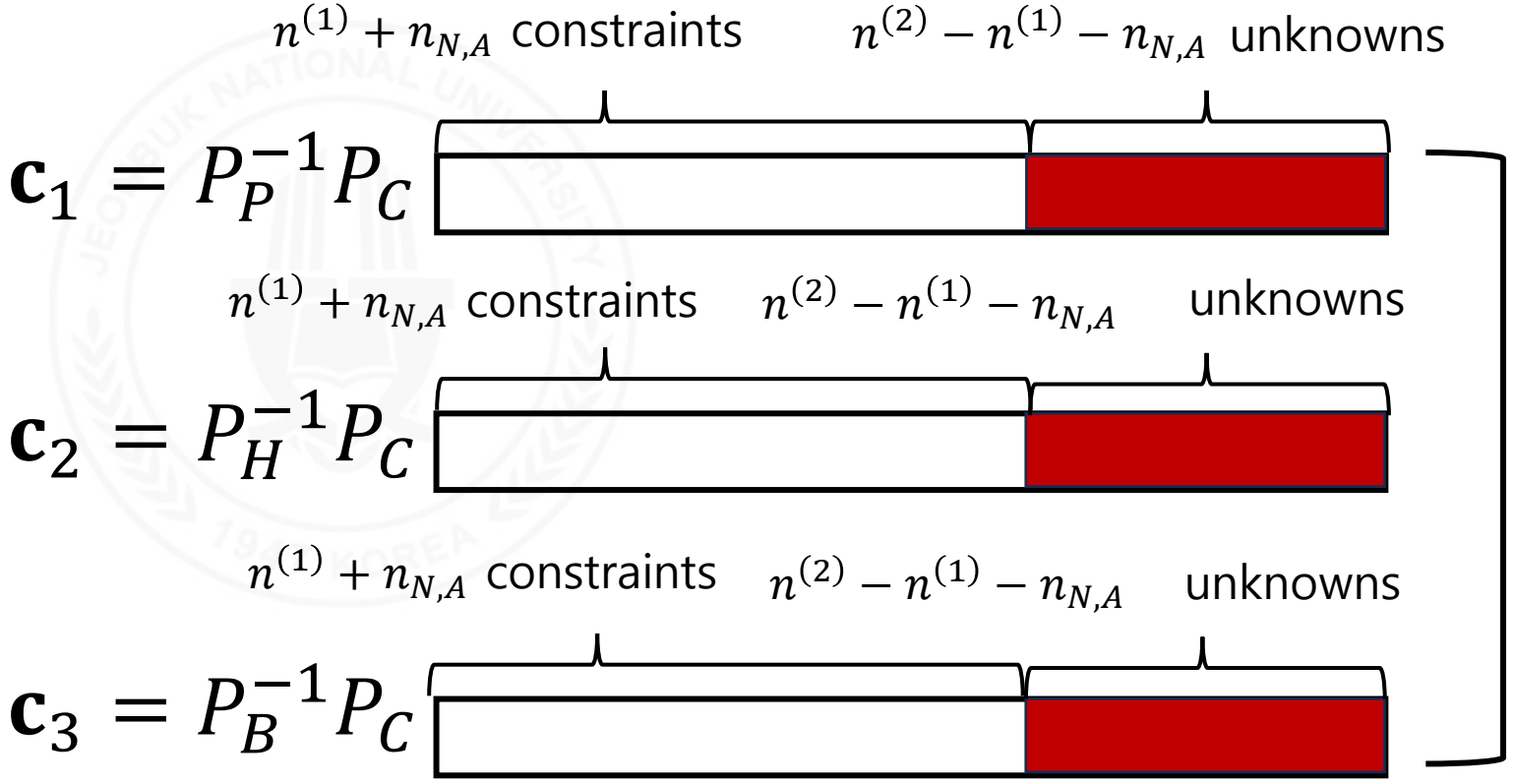
Sketch of the new scheme



Sketch of the new scheme

From the linear combinations between the three CT, $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_3$.

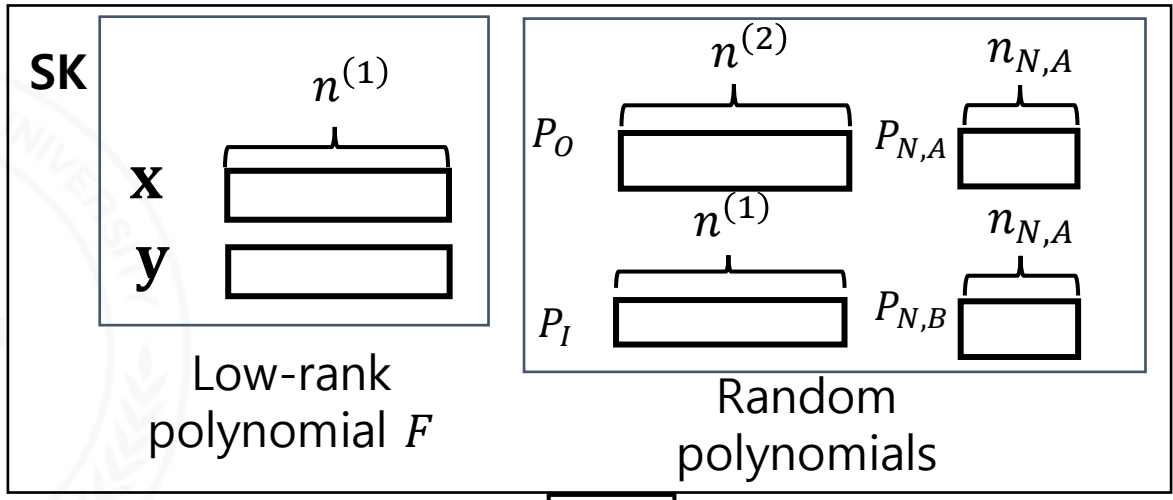
- The attacker may make a three overdetermined equations with unknowns $3\left(n^{(2)} - n^{(1)} - n_{N,A}\right)$ and constraints $3\left(n^{(1)} + n_{N,A}\right)$.
- Actually, CT has at most $n^{(2)}$ linearly independent equations and thus, the equation is determined when $3\left(n^{(2)} - n^{(1)} - n_{N,A}\right) < n^{(2)}$
- Otherwise, the attack should require an additional complexity by guessing if $3\left(n^{(2)} - n^{(1)} - n_{N,A}\right) > n^{(2)}$



Modified Layered ROLLO-I: Procedures

1. Key generation: Unchanged

- 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

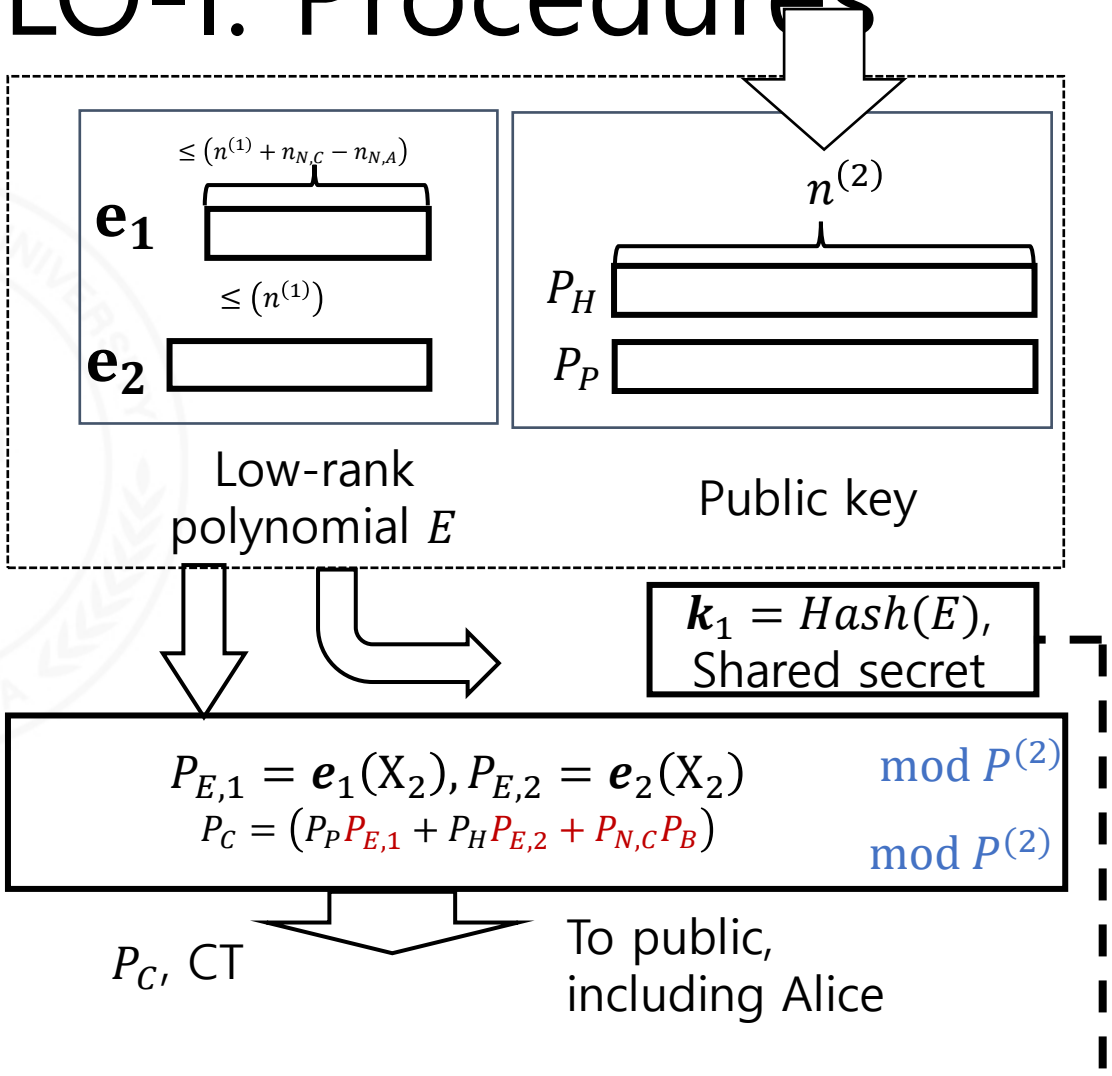
Modified Layered ROLLO-I: Procedures

2. Encapsulation: Unchanged

- 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)

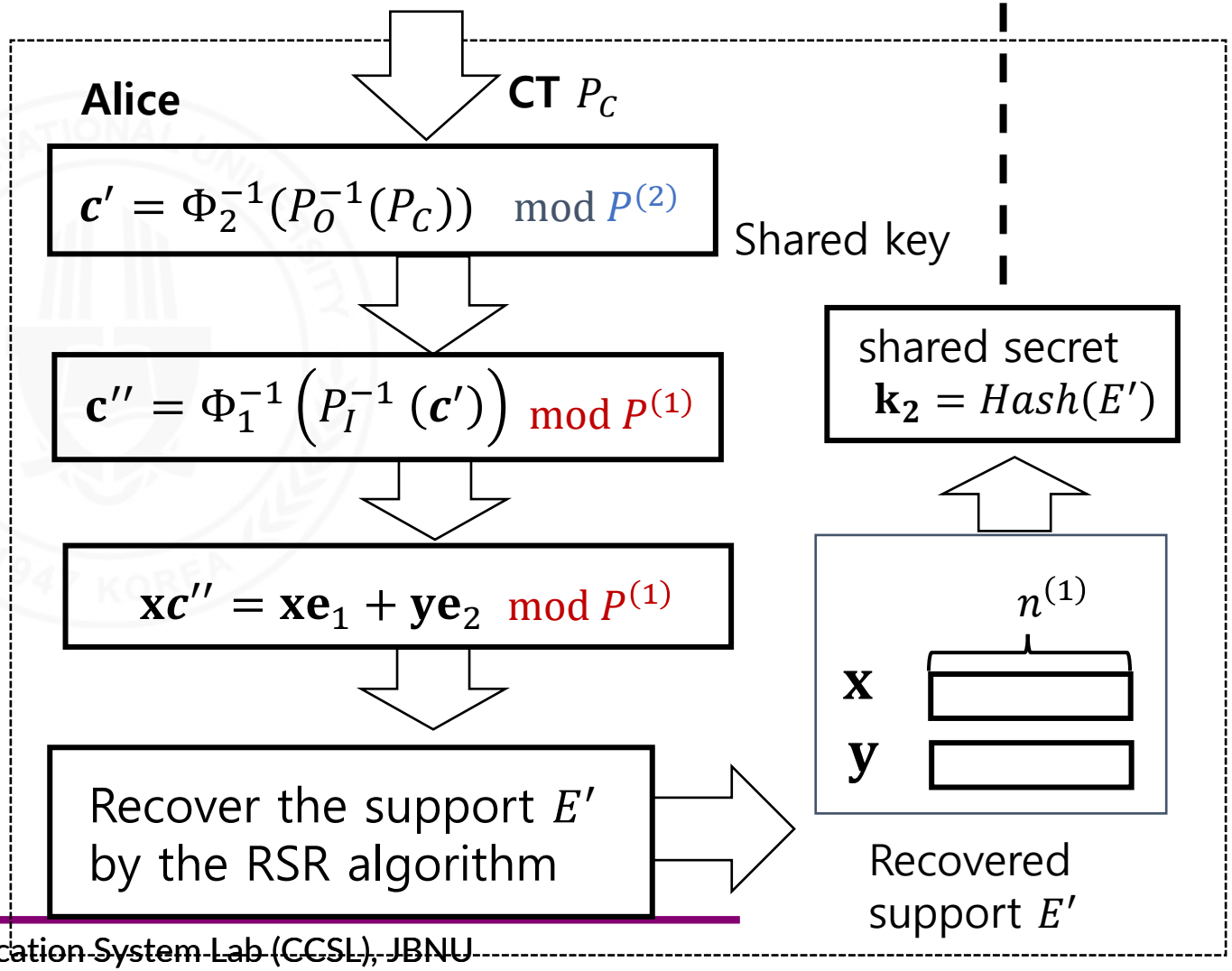


Modified Layered ROLLO-I: Procedures

3. Decapsulation: Unchanged

- 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 $\mathbf{k_2} = Hash(E')$



Parameter selection for new-LROLLO

In summary, the new attack scenarios have SL either of

1) Attacks by rank support decoding S_R
2) First direct attack(Using two PKs, P_H, P_P, and other pairs)

$$S_{D1} = O\left(q^{(2n^{(1)}-n^{(2)}+n_{N,A})m}\right)$$

3) Second direct attack(Using two PKs (P_H, P_P and other pairs) and CT(P_C))

$$S_{D2} = O\left(q^{(n^{(2)}-n^{(1)}-n_{N,A}-1)m}\right)$$

4) Third direct attack(Using three PKs and CT, P_H, P_P, P_B and CT(P_C)))

$$S_{D3} = O\left(q^{(3(n^{(2)}-n^{(1)}-n_{N,A}-1)-n^{(2)})m}\right)$$

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(2023.11),**

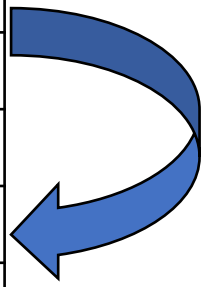
Name	q	$n^{(1)}$	$n^{(2)}$	$n_{N,A}$	m	r	d	SL (S_R)	SL. (S_{D1})	SL. (S_{D2})	SL. (S_{D3})	DFR	Pk size	ct size
New-128	2	37	**71**	**4**	67	**7**	2	**154.6**	**469**	**1943**	**1072**	**-23**	1755	585
New-192	2	43	**83**	**4**	79	**9**	2	**199.68**	**553**	**2765**	**1738**	**-25**	2460	820
New-256	2	53	**103**	**4**	97	**12**	2	**273.4**	**679**	**4365**	**3104**	**-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
  - CPU: Intel(R) Xeon(R) w7-3465X, 256GB, x4 4800GB/s, Hynix DDR5

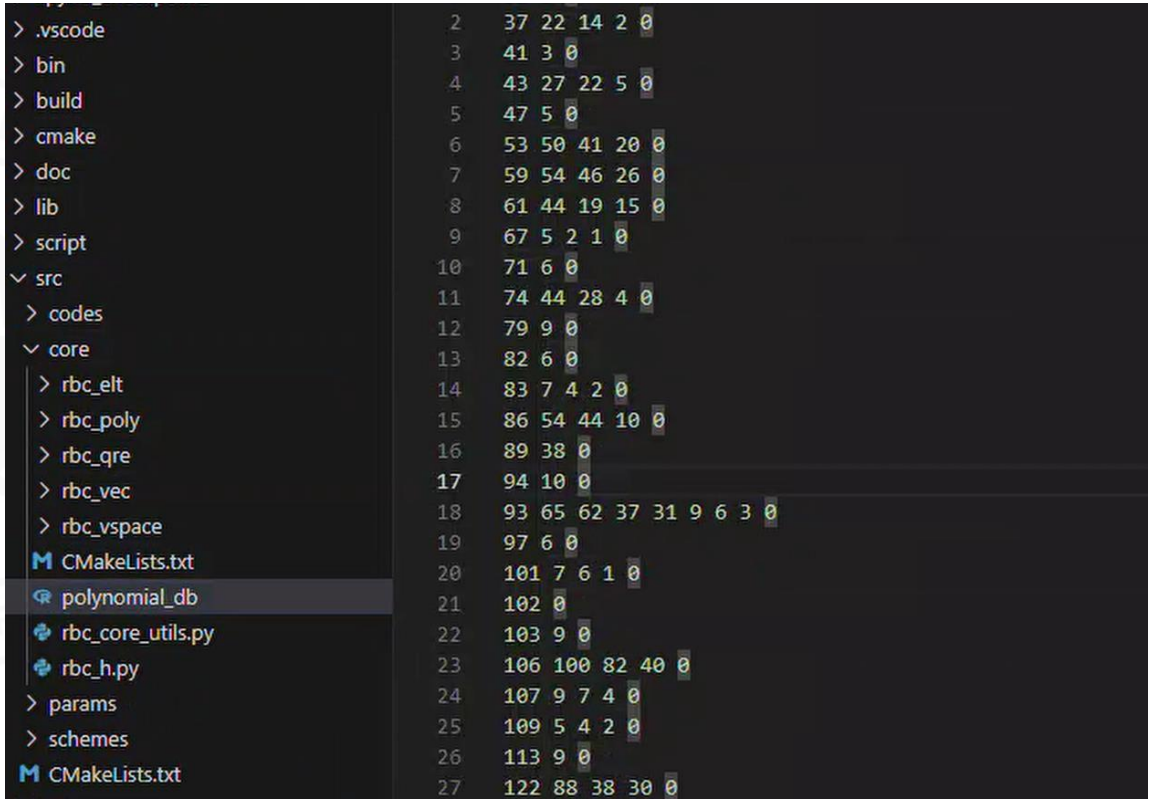
Instance	Key generation	Encapsulation	Decapsulation	Total
ROLLO-I-128	2,331,728	327,392	4,737,172	7,396,292
ROLLO-I-192	2,537,876	546,316	6,248,864	9,333,056
ROLLO-I-256	3,436,412	428,696	10,493,536	14,358,644
New-128	1,417,548	495,412	2,440,012	4,352,972
New-192	1,848,076	823,500	2,789,864	5,461,440
New-256	2,224,488	767,080	4,263,728	7,255,296



30-50% cycle reduction

On the implementation of modulus $P^{(1)}$ operation

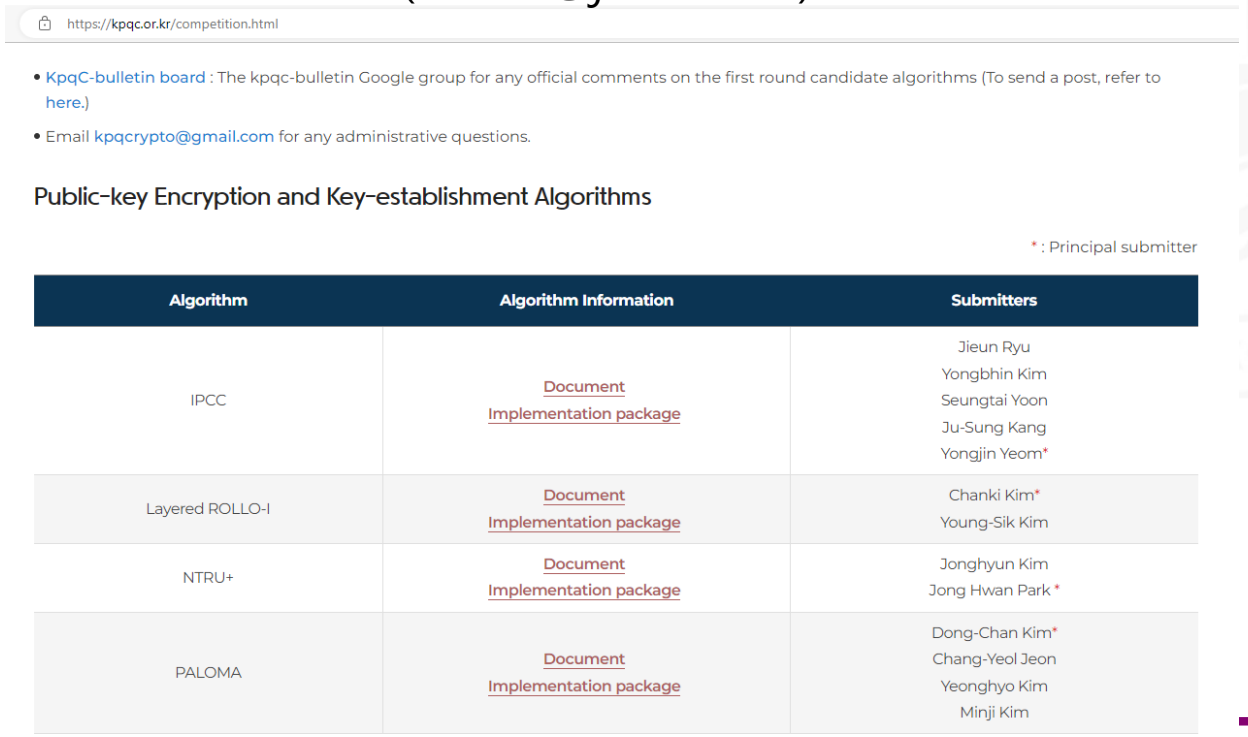
- From the key generation phase, the inner modulus polynomial $P^{(1)}$ is classified as a class of SK, which can be a fixed or random value for each KEM operation.
- Note that the corresponding SL can be lowered from the low-degree polynomial $\frac{P_B}{P^{(1)}}$ when modulus polynomial $P^{(1)}$ can be found by attacker.
- For the fixed case, a new attack for recovering $P^{(1)}$ can be devised by collecting prior PKs or CTs and thus, using random values for $P^{(1)}$ is desirable for security.
- Still, $P^{(1)}$ in the corresponding C code is implemented as a fixed polynomial, which follows the initial source code RBC(Rank-based cryptography) in the ROLLO-I.
- We are trying to modify it with minimizing the additional processing speed (i.e. processing cycle) until the end of 2nd submission. There seems to be some options to be applied for performance optimization.



List of fixed primitive polynomial in the implementation codes

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)



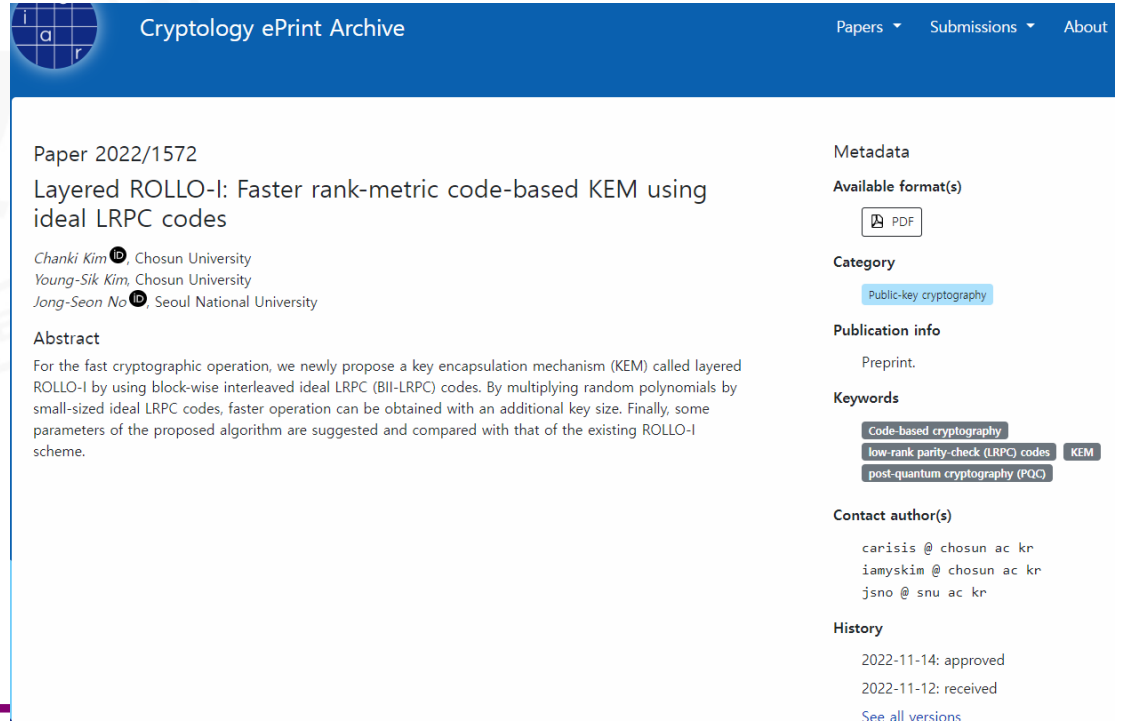
https://kpqc.or.kr/competition.html

- 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