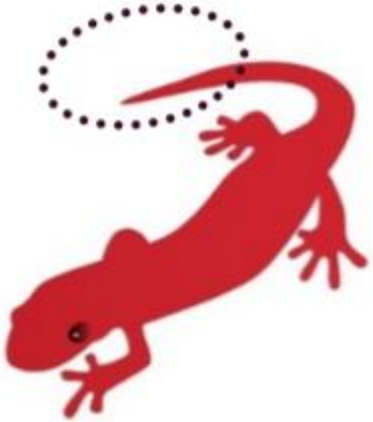
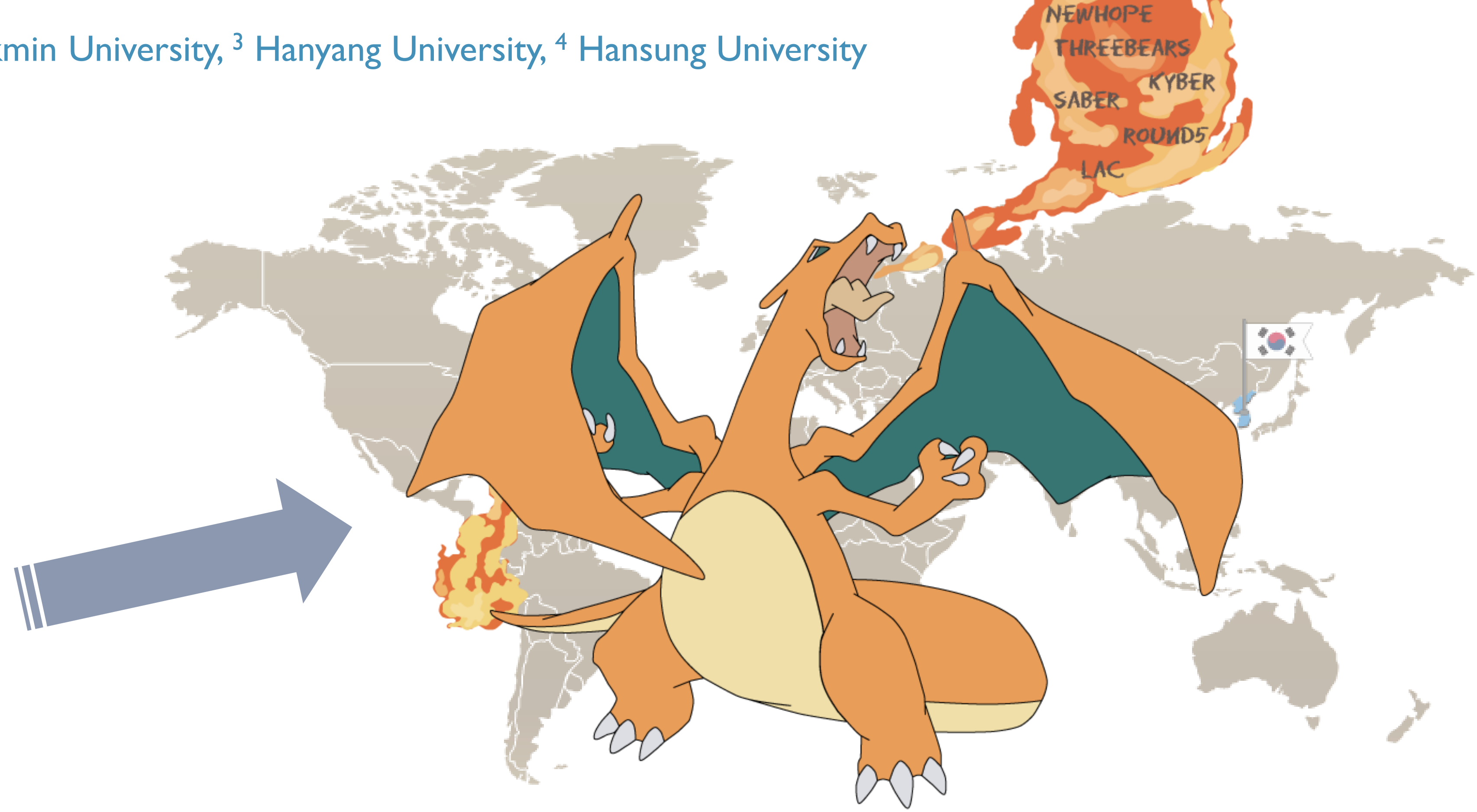
# LizarMong: Excellent KEM based on RLWE and RLWR

2019.12. 5.

Chi-Gon Jung[1], JongHyeok Lee[2], Youngjin Ju[3], Yong-Been Kwon[4], Seong-Woo Kim[1], and Yunheung Peak[1]

[1] Seoul National University, [2] Kookmin University, [3] Hanyang University, [4] Hansung University






<image source: https://csrc.nist.gov/CSRC/media/Presentations/Lizard/images-media/Lizard-April2018.pdf >

# CONTENTS

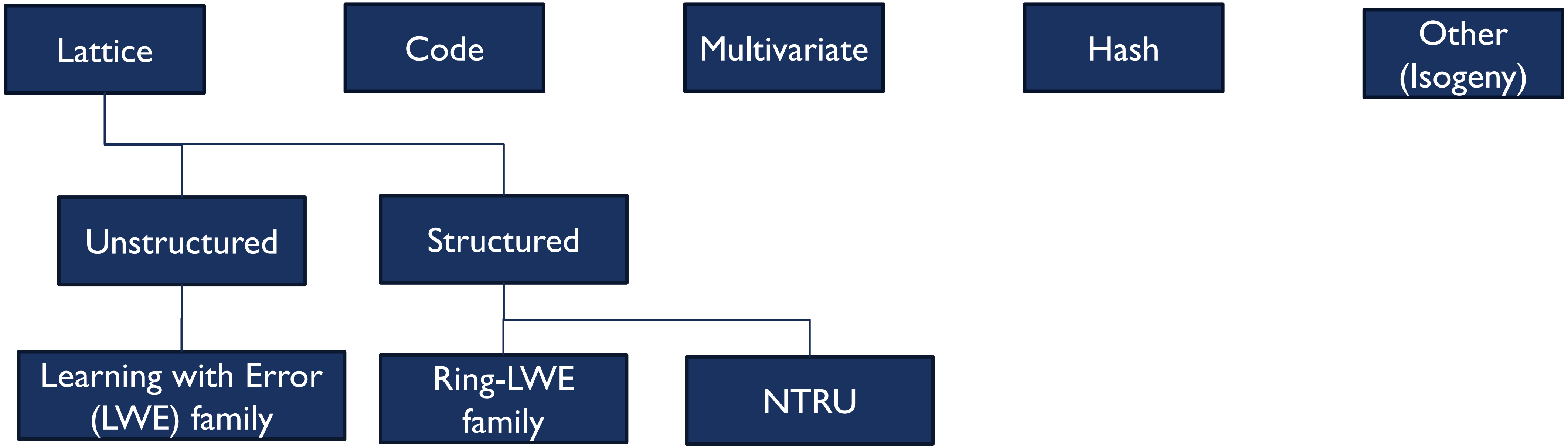
# INTRODUCTION

# NIST Post-Quantum Cryptography Competition

- Goal
  - to develop cryptographic systems (signature, encryption, and key-establishment)
  - that are secure against both quantum and classical computers,
  - and can interoperate with existing communications protocols and networks.
- Progress (2016 ~ ): 2017 - 1st Round Begins / 2019 - 2nd Round Begins
- Evaluation Criteria
  - Security: focus on categories 1, 2, and 3. (i.e. 128-192bit security strength.)
  - Cost and Performance
    - The size of public keys and ciphertext. (In this work called *bandwidth*)
    - Computation efficiency of key generation, public and private key operations. (called *performance*)
    - Probability of decryption failures. (called *correctness*)

# NIST Post-Quantum Cryptography Competition

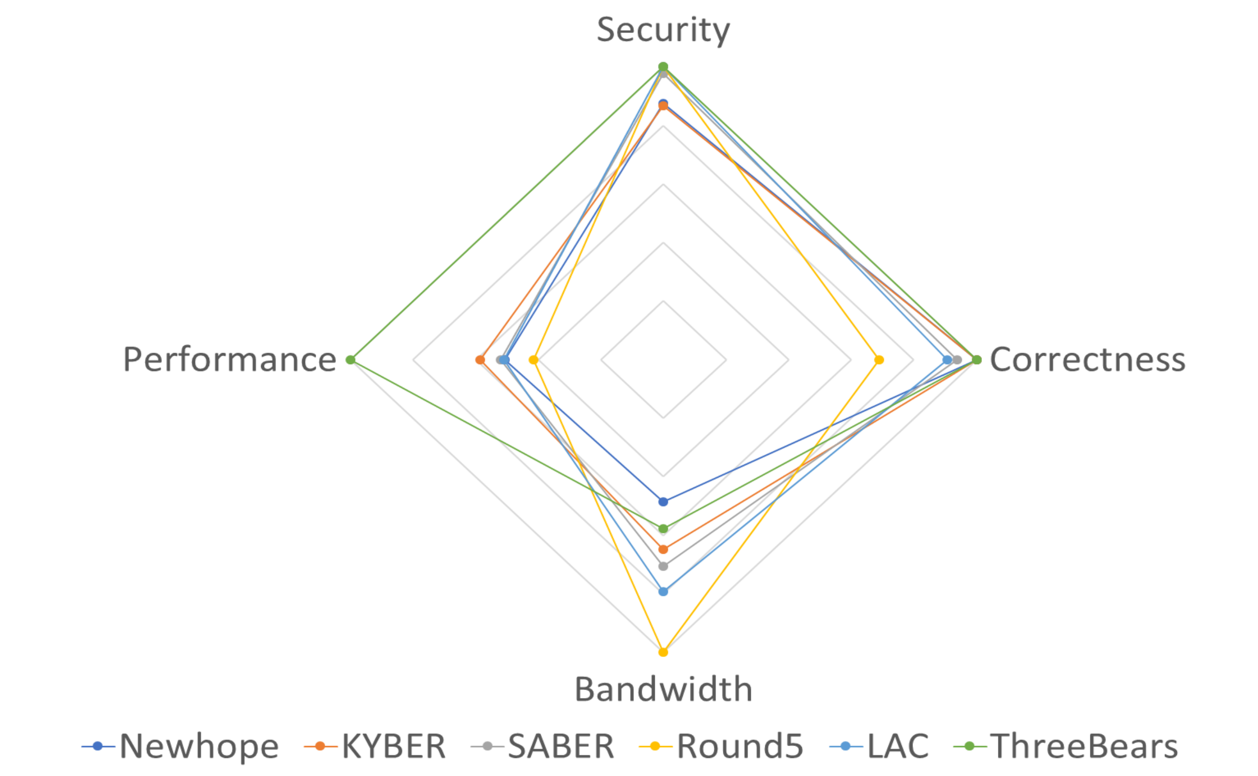
- Main families for which post-quantum primitives




- In this work focused on Ring-LWE family.
  - Simple, efficient, and parallelizable.
  - Provably secure under a worst-case.
  - Relatively well-study.

# Which is the best among NIST candidate algorithms?

< Evaluating of 128-bit security-level KEM >

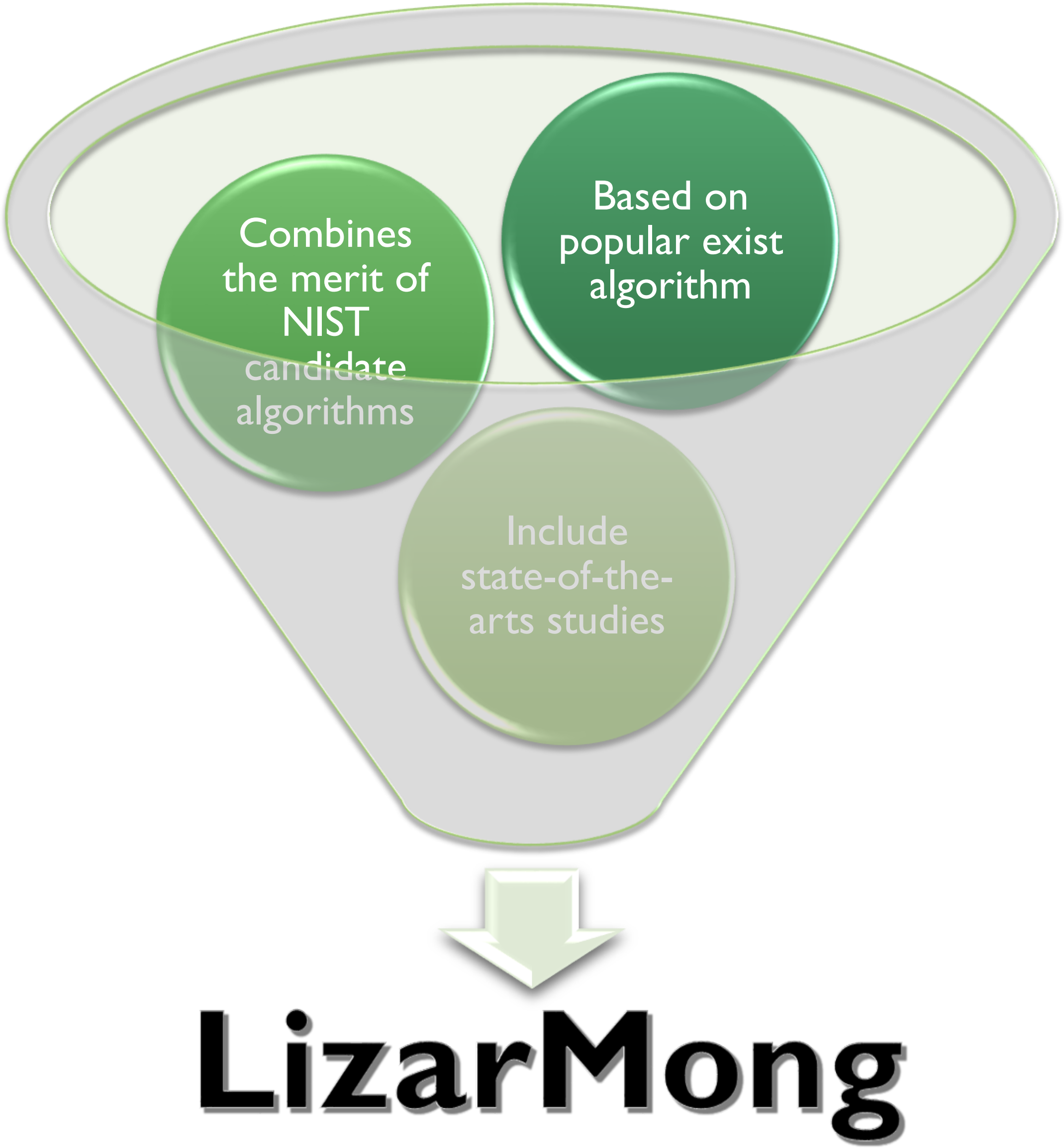



- All evaluation criteria are important.
  - NIST said "Still open to mergers."
- Most of latest studies are not included.
  - Side-channel attacks.
  - Errors in each bit occur dependently.
  - More efficient QROM for IND-CCA2.

**★Goal: Making an excellent key encapsulation mechanism of all aspects ★**

# DETAIL TO LIZARMONG

# Overview




- Based on *RLizard*[CKLS18] (was 1st round candidate).
  - RLWE + RLWR and Sparse ternary secret.
  - Good performance, security, and correctness.
  - But, relatively large bandwidth.
- Combines
  - Bytes Modulus[LAC], Error correction code[Round5], and Centered binomial distribution[NewHope].
  - Public-key and ciphertext compress technique.
- Includes
  - Countermeasures against known side-channel attacks.
  - Error dependency[DVV19].
  - Improved QROM for IND-CCA2[JZC+18].

# Specification of LizarMong

## Design elements

- Reduce the bandwidth and maintain the RLizard's strengths.
- Minimized known side-channel attack points.

| Compare | Underlying Problem | Ring | Compress | Modulus | ECC | Distributions | |
|---|---|---|---|---|---|---|---|
| | | | | | | Secret | Error |
| LizarMong | RLWE+RLWR | $\mathbb{Z}_q/X^n+1$ | **Yes** | **Small (fixed $2^8$)** | **XE5** | Uniform sparse ternary | **Binomial (std≈0.7)** |
| RLizard | 〃 | 〃 | No | Small ($2^{10\sim12}$) | None | 〃 | Gaussian (std≈1.15) |
| Why? | Key: conservative Enc/Dec: Fast | Fast / secure | Bandwidth | Bandwidth, Performance | Correctness, Side-channel | Correctness, Performance | Side-channel Correctness, Performance |
| Proved | - | - | Common in NIST's Alg. | [PRSD17] | [Saa17] | - | [ADPS16] |

# Specification of LizarMong

## ☐ IND-CCA2 KEM

Bob
(Server)

Alice
(Client)

① Key generate

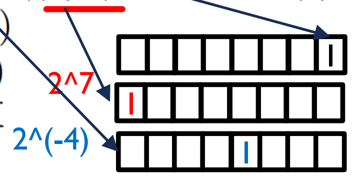
$$\begin{aligned}
&Seed_a \xleftarrow{\$} \{0,1\}^{256} \\
&\mathbf{a} \leftarrow \texttt{SHAKE256}(Seed_a, n/8) \\
&\mathbf{s} \xleftarrow{\$} HWT_n(h_s) \text{ and } \mathbf{e} \xleftarrow{\$} \psi_{cb}^{n} \\
&\mathbf{b} \leftarrow -\mathbf{a} * \mathbf{s} + \mathbf{e} \\
&pk \leftarrow (Seed_a \parallel \mathbf{b}) \text{ and } sk_{cpa} \leftarrow \mathbf{s} \\
&\mathbf{u} \xleftarrow{\$} R_2 \\
&\textbf{return} \quad pk, sk \leftarrow (sk_{cpa} \parallel \mathbf{u})
\end{aligned}$$

*Public-key (pk)*

*544byte (in Comfort) or*
*1056byte (in Strong)*

② Encapsulation

$$\begin{aligned}
&\delta \xleftarrow{\$} \{0,1\}^{sd} \\
&\mathbf{r} \leftarrow H(\delta) \\
&\delta' \leftarrow \mathrm{eccENC}(\delta) \\
&\mathbf{c_1} \leftarrow \lfloor (p/q) \cdot \mathbf{a} * \mathbf{r} \rceil \\
&\mathbf{c_2} \leftarrow \lfloor \underline{(k/q)} \cdot ((\underline{q/2}) \cdot \delta' + \mathbf{b} * \mathbf{r}) \rceil \\
&\mathbf{c} \leftarrow (\mathbf{c_1} \parallel \mathbf{c_2}) \\
&\mathbf{K} \leftarrow G(\mathbf{c}, \delta') \\
&\textbf{return} \quad \mathbf{c}, \mathbf{K}
\end{aligned}$$




*Ciphertext (c)*

*640byte (in Comfort) or*
*1280byte (in Strong)*

③ Decapsulation

$$\begin{aligned}
&\mathbf{c_1}, \mathbf{c_2} \leftarrow \mathrm{Parsing}(\mathbf{c}) \\
&sk_{cpa}, \mathbf{u} \leftarrow \mathrm{Parsing}(sk) \\
&\hat{\delta}' \leftarrow \lfloor (2/p) \cdot ((p/k) \cdot \mathbf{c_2} + \mathbf{c_1} * sk_{cpa}) \rceil \\
&\hat{\delta} \leftarrow \mathrm{eccDEC}(\hat{\delta}') \\
&\hat{\mathbf{r}} \leftarrow H(\hat{\delta}) \\
&\hat{\delta}'' \leftarrow \mathrm{eccENC}(\hat{\delta}) \\
&\hat{\mathbf{c}} \leftarrow \lfloor (p/q) \cdot \mathbf{a} * \hat{\mathbf{r}} \rceil \parallel \lfloor (k/q) \cdot ((q/2) \cdot \hat{\delta}'' + \mathbf{b} * \hat{\mathbf{r}}) \rceil \\
&\textbf{if } \mathbf{c} \neq \hat{\mathbf{c}} \textbf{ then } \mathbf{K} \leftarrow G(\mathbf{c}, \mathbf{u}) \textbf{ else } \mathbf{K} \leftarrow G(\mathbf{c}, \hat{\delta}'') \\
&\textbf{return} \quad \mathbf{K}
\end{aligned}$$

< Detail parameters for each security-level >

| *parameters* | $n$ | $q$ | $p$ | $k$ | $h_s$ | $h_r$ | $d$ | $sd$ | $cb$ |
|---|---|---|---|---|---|---|---|---|---|
| Comfort (128-bit) | 512 | 256 | 64 | 16 | 128 | 128 | 256 | 256 | 1 |
| Strong (256-bit) | 1024 | 256 | 64 | 16 | 128 | 128 | 512 | 512 | 1 |

# Security analysis

## Security Proof

RLWE / RLWR

↓ Lemma 1.

RLizard
(IND-CPA PKE)

↓ Theorem 1.
Under the assumption that SHAKE256 is ROM

LizarMong
(IND-CPA PKE)

↓ Jiang's Transformation

LizarMong
(IND-CCA2 KEM)

## Cryptanalytic attacks

- BKZ lattice basis reduction algorithm.
- The core SVP. (ignores repeated calls for SVP oracle)
- Use 'online LWE estimator' [Alb17].
- Consider Dual and Primal attack like RLizard.

Table 3: Computational complexity of best RLWE and RLWR attacks

| Parameters | Claim Security | Attacks | | Classical | Quantum |
|---|---|---|---|---|---|
| Comfort | NIST Category 1 (AES 128-bit) | Primal | RLWE | **133** | **121** |
| | | | RLWR | 144 | 131 |
| | | Dual | RLWE | 165 | 154 |
| | | | RLWR | 180 | 170 |
| Strong | NIST Category 5 (AES 256-bit) | Primal | RLWE | **256** | **236** |
| | | | RLWR | 269 | 249 |
| | | Dual | RLWE | 304 | 275 |
| | | | RLWR | 328 | 301 |

# Correctness analysis

- Estimating the Correctness considering the dependency of each bit error.
  - The correctness of all RLWE estimates on the assumption that errors occur independently.
  - The independent assumption was disproved [DVV19]; Especially improper using ECC.

- Decryption failure is when satisfied $|e * r + s * f + g| \geq \frac{q}{4} - \frac{q}{2p}$.
  - $f = a * r - (q/p)c_1; g = v - \hat{v}; v = \lfloor (p/q) \cdot ((q/2) \cdot \mathbf{M}' + \mathbf{b} * \mathbf{r} \rceil$ , $\hat{v} = v \ll (\log p - \log k)$

- $\Pr[Fail] \approx \sum_{\|S\|,\|C\|} \left(1 - Binom(d, l_m, p_b)\right) \cdot \Pr[\|S\|] \cdot \Pr[\|C\|]$
  - $S = (\mathbf{s}, \mathbf{e})^T, C = (\mathbf{f}, \mathbf{r})^T$ , $Binom(k, n, p) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$ , $p_b = \Pr[F_0 \mid \|S\|, \|C\|]$

| Prameters | without ECC | with XE5(5bit ECC) |
|---|---|---|
| Comfort | $2^{-37}$ | $2^{-179}$ |
| Strong | $2^{-68}$ | $2^{-302}$ |

# Resistance to known side-channel attacks

- We investigated the known major side-channel attacks and the points they exploited.

| Attack methods | Attacks | Attack Points | |
|---|---|---|---|
| Timing Attacks | [PH16] | ~~Modulus operation doing or not.~~ | AND and ADD instead of Modulus Op. |
| | [KH18] | ~~CDT sampling's branch.~~ | |
| Differential Attacks | [PPM17] | ~~INV NTT operation~~ | Do not use NTT |
| | [ATT+18] | ~~Multiplication using secrets.~~ | Devised sparse polynomial multiplication with Hiding |
| | [HCY19] | ~~Multiplication using secrets.~~ | |
| Template Attacks | [BFM+18] | ~~Multiplication using secrets.~~ | |
| Fault Attacks | [EFGT18] | ~~Error sampling function.~~ | Check the final loop index |
| | [RRB+19] | ~~Same distribution for secret and error sampling.~~ | Each distribution for error and secret |
| Cache Attacks | [BHLY16] | ~~CDT sampling's table look up.~~ | Replaced with centered binomial distribution |

- Our strategy
  - First, ruled out the targeted by the known attacks during the design element selection.
  - Second, internalizes efficient countermeasures for unavoidable vulnerabilities.

# Evaluation

- Compare with RLizard,
  - Band.: up to 85% smaller / Perfor.: 3.3x faster
- Compare with NIST candidate Algorithms,
  - Band.: 5~42% smaller / Perfor.: 1.2~4.1x faster

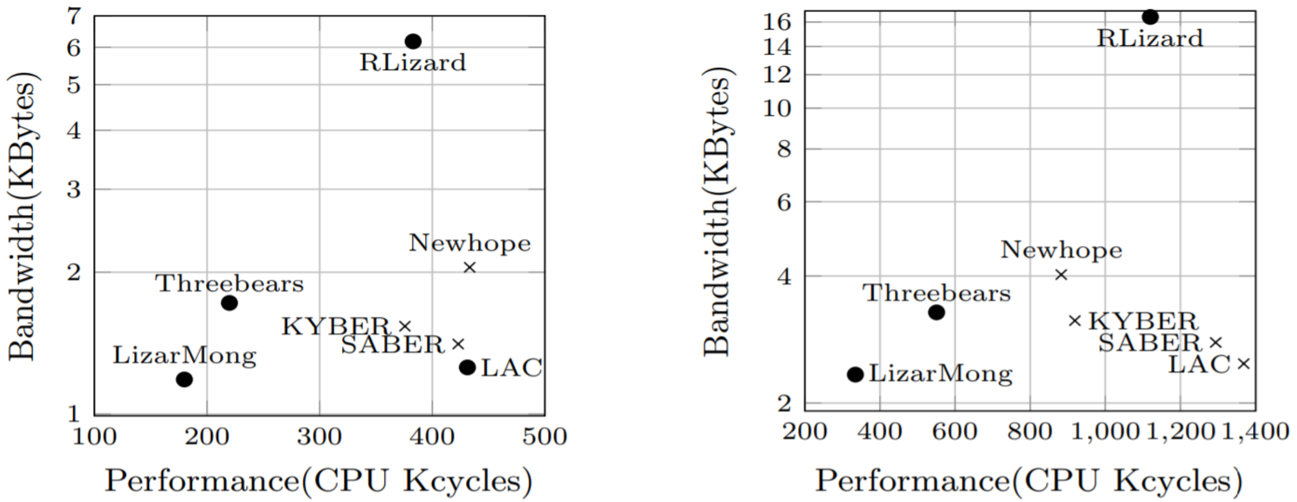
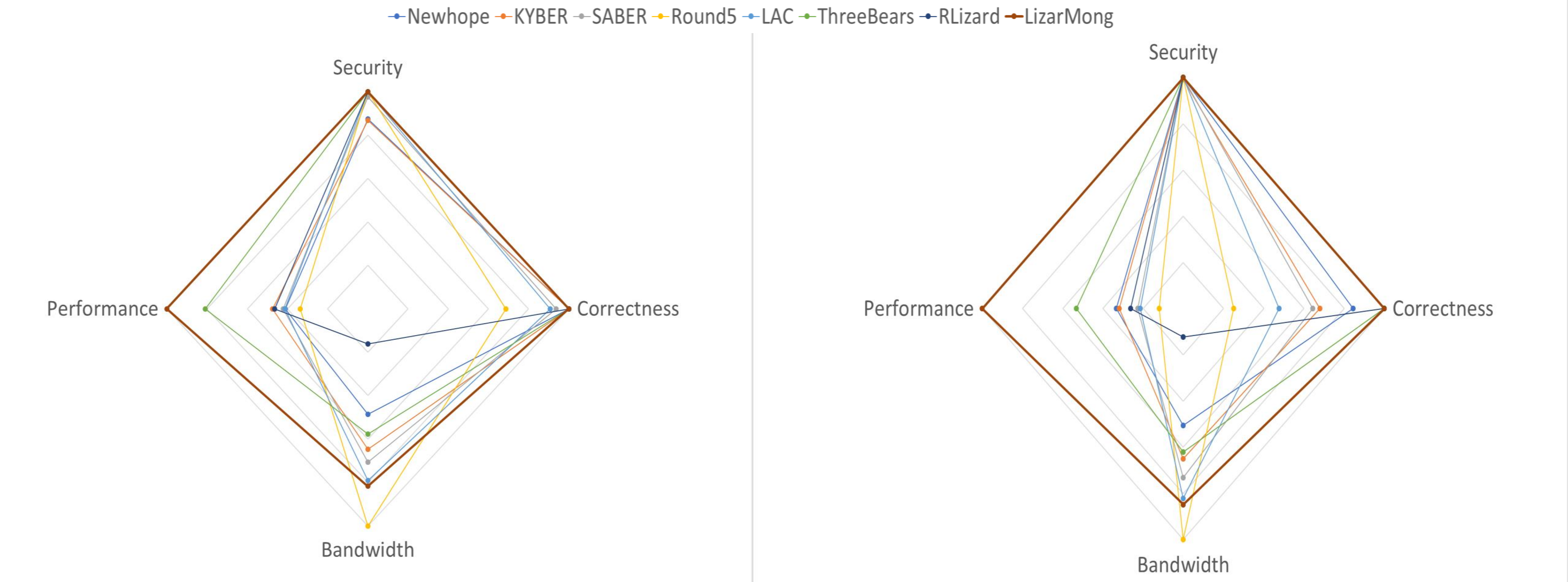



Fig. 1: Comparison of bandwidth and performance based on IND-CCA2 KEM. (left) 128-bit security level (right) 256-bit security level (Note: ● are algorithms with security and correctness similar to each security level, and × are not.)

Table 5: Comparison KEM with NIST candidate algorithms and RLizard

| Algorithms | Security (log) | Correctness (log) | Bandwidth (Bytes) | Performance (K cycles) Enc+Dec | Performance (K cycles) KeyGen |
|---|---|---|---|---|---|
| LizarMong | 133 | −179 | 1,184 | 137.5 | 42.4 |
| | 256 | −302 | 2,336 | 272.7 | 61.8 |
| RLizard | 147 | −188 | 6,176 | 217.8 | 165.3 |
| | 195 | −246 | 8,240 | 416.9 | 232.7 |
| | 318 | −306 | 16,448 | 737.3 | 382.7 |
| NewHope | 112 | −213 | 2,048 | 329.6 | 103.6 |
| | 257 | −216 | 4,032 | 673.5 | 209.2 |
| KYBER | 111 | −178 | 1,536 | 278.2 | 97.5 |
| | 181 | −164 | 2,272 | 463.6 | 174.3 |
| | 254 | −174 | 3,136 | 656.0 | 263.1 |
| SABER | 125 | −120 | 1,408 | 316.9 | 106.1 |
| | 203 | −136 | 2,080 | 587.6 | 213.6 |
| | 283 | −165 | 2,784 | 934.8 | 359.2 |
| LAC | 147 | −116 | 1,256 | 341.2 | 90.0 |
| | 286 | −143 | 2,244 | 840.1 | 235.6 |
| | 320 | −122 | 2,480 | 1,101.6 | 266.6 |
| Round5 (IND-CPA) | 128 | −88 | 994 | 384.4 | 114.6 |
| | 193 | −117 | 1,639 | 857.2 | 311.3 |
| | 256 | −64 | 2,035 | 1,794.9 | 643.4 |
| Threebears | 154 | −156 | 1,721 | 167.8 | 52.1 |
| | 235 | −206 | 2,501 | 271.4 | 91.9 |
| | 314 | −256 | 3,281 | 402.5 | 148.2 |

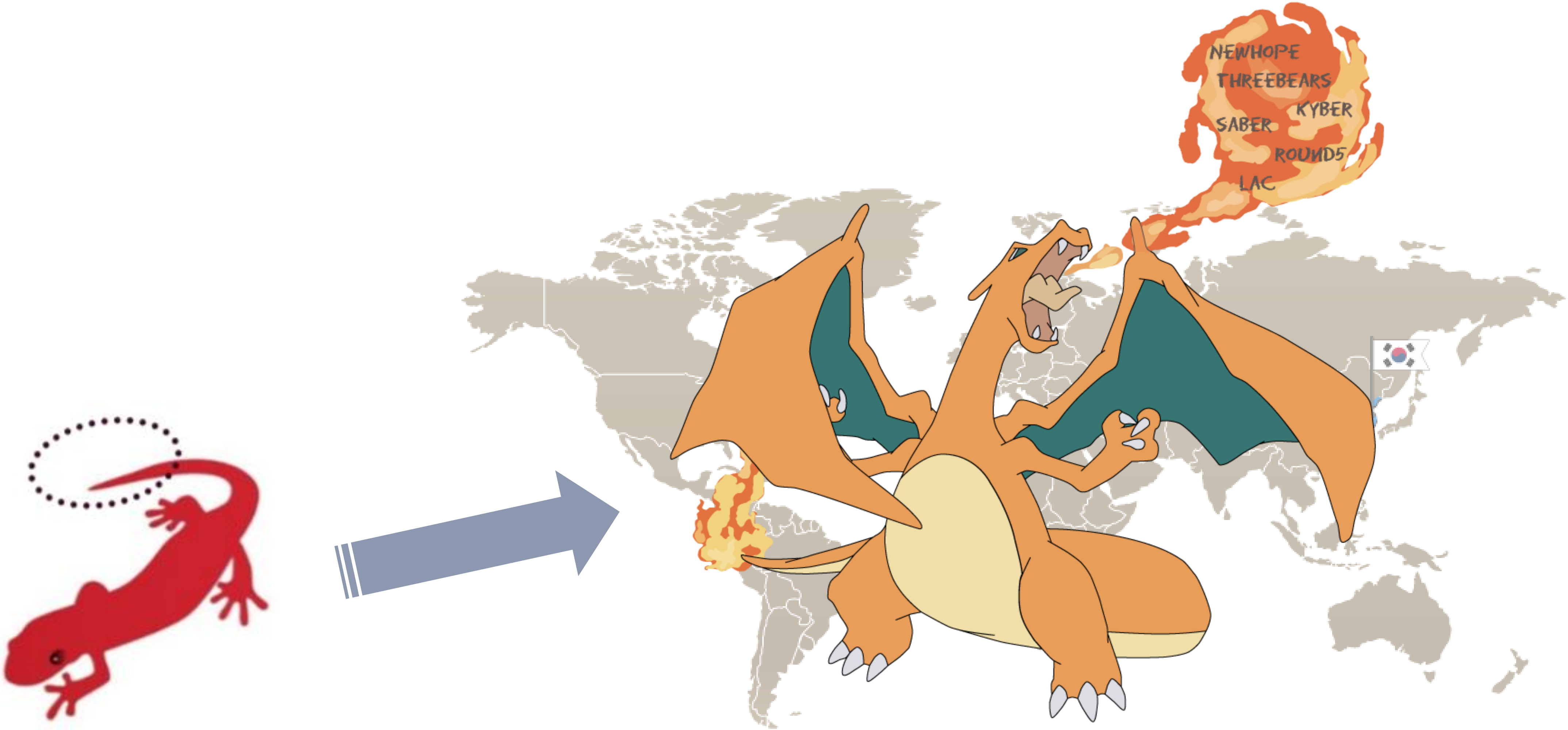
# Conclusion




< 128bit security >　　　　< 256bit security >

**★ LizarMong is excellent key encapsulation mechanism of all aspect! ★**

# Have any Questions? Thank you!




<fig. refer: https://csrc.nist.gov/CSRC/media/Presentations/Lizard/images-media/Lizard-April2018.pdf >