

KpqC Competition, KpqC Algorithms, and Current Status

The 27th World Conference on Information Security Applications

Kyung Chul Jeong
August 27, 2026

Secretary, KpqC Research Group
National Security Research Institute (NSR), Korea



양자내성암호연구단

※ $K_{\text{門}}C$

(pq $\approx$ 門, the Chinese character for 'door')

※ K_門C

(pq $\approx$ 門, the Chinese character for 'door')

- ▶ Opening the door to Korean post-quantum crypto research activities, and
- ▶ Closing the door on the quantum security threat

(pq $\approx$ 門, the Chinese character for ‘door’)

- Established in 2021, funded by the Korean government
- Operated the KpqC Competition (Nov. 2021 – Jan. 2025)
- Currently promoting real-world adoption of KpqC algorithms
- Continuing to support foundational research for PQC migration

To talk about post-quantum cryptography,

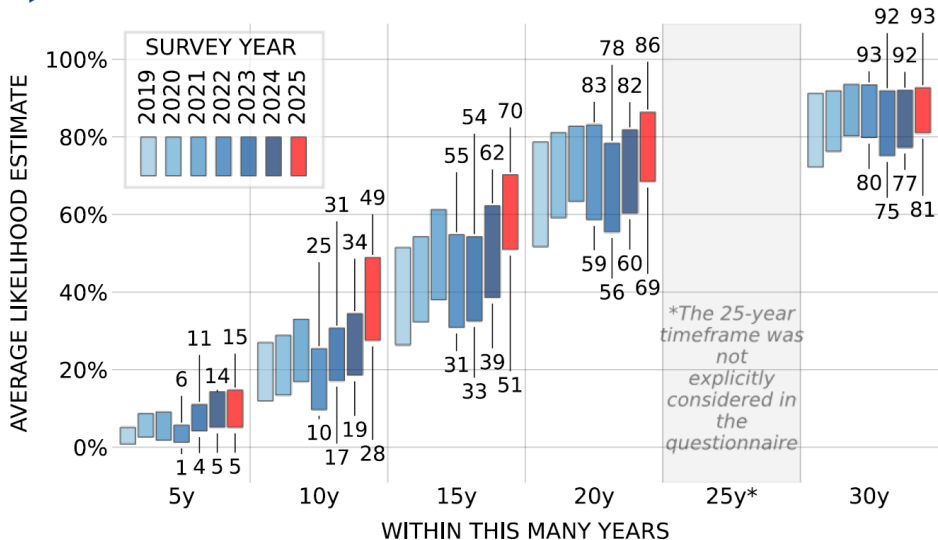
- ▶ *Pre*-quantum cryptography based on integer factoring or discrete logarithm
- ▶ Shor's algorithm and its complexity with quantum HW
- ▶ Various post-quantum hard problems and their hardness
- ▶ HNDL attack, inventory, hybrid approach, crypto agility
- ▶ Migration policy and so on

In this talk,

- ▶ I assume that most of the concepts are already well understood.
- ▶ Just focusing on KpqC
- ▶ Briefly covering one thing, the 'When' question



SURVEY COMPARISON OF THE EXPERTS' ESTIMATES OF THE LIKELIHOOD OF A QUANTUM COMPUTER ABLE TO BREAK RSA-2048 IN 24 HOURS



In Mosca et al. *Quantum Threat Timeline Report 2025*. GRI (March 2026),

Within 10 years

“Half of the respondents felt the likelihood of such a computer within 10 years was ‘about 50%’ or more likely.”

- Survey based annual report from 2019
 - Likelihood estimates for CRQC realization
 - ▶ Cryptographically Relevant Quantum Computer
- ex.) A quantum computer able to break RSA-2048 in 24 hours

Fixing '50%+ likelihood', comparison to '*Quantum Threat Timeline Report 2024*'

- A 5-year shift (from 15 years to 10 years) in just one year
 - ▶ The year at '50%+ likelihood' majority: 15 years

Fixing '50%+ likelihood', comparison to '*Quantum Threat Timeline Report 2024*'

- A 5-year shift (from 15 years to 10 years) in just one year
 - ▶ The year at '50%+ likelihood' majority: 15 years
 - ※ It was 15 years from the first 2019 report through 2024.
 - ※ No option between 10 and 15

Fixing '50%+ likelihood', comparison to '*Quantum Threat Timeline Report 2024*'

- A 5-year shift (from 15 years to 10 years) in just one year
 - ▶ The year at '50%+ likelihood' majority: 15 years
 - ※ It was 15 years from the first 2019 report through 2024.
 - ※ No option between 10 and 15
- From 31% to 50% in just one year
 - ▶ The share of respondents at '50%+ likelihood' for 10 years: 31%

Google Quantum Computing Roadmap

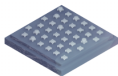
Our focus is to unlock the full potential of quantum computing by developing a large-scale computer capable of complex, error-corrected computations. We're guided by a roadmap featuring six milestones that will lead us toward top-quality quantum computing hardware and software for meaningful applications.



MILESTONE 1

BEYOND CLASSICAL

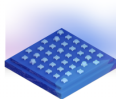
Physical Qubits: 54
Logical Qubit Error Rate: -



MILESTONE 2

QUANTUM ERROR CORRECTION

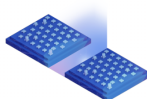
Physical Qubits: 10^2
Logical Qubit Error Rate: 10^{-2}



MILESTONE 3

BUILDING A LONG-LIVED LOGICAL QUBIT

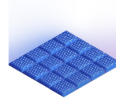
Physical Qubits: 10^3
Logical Qubit Error Rate: 10^{-6}



MILESTONE 4

CREATING A LOGICAL GATE

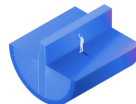
Physical Qubits: 10^4
Logical Qubit Error Rate: 10^{-8}



MILESTONE 5

ENGINEERING SCALE UP

Physical Qubits: 10^5
Logical Qubit Error Rate: 10^{-8}



MILESTONE 6

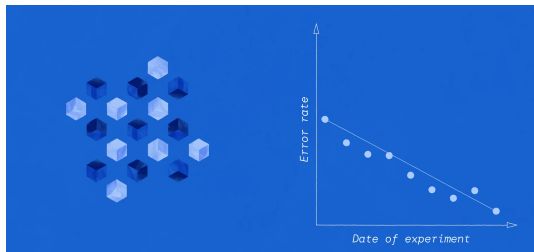
LARGE ERROR-CORRECTED QUANTUM COMPUTER

Physical Qubits: 10^6
Logical Qubit Error Rate: 10^{-13}

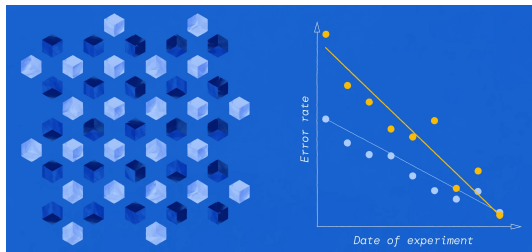


Google Quantum AI. “**Suppressing Quantum Errors by Scaling a Surface Code Logical Qubit**”. In: *Nature* 614.7949 (Feb. 2023)

‘These results mark an experimental demonstration in which **quantum error correction begins to improve performance** with increasing qubit number, illuminating the path to reaching the logical error rates required for computation.’



(from <https://quantumai.google/>)



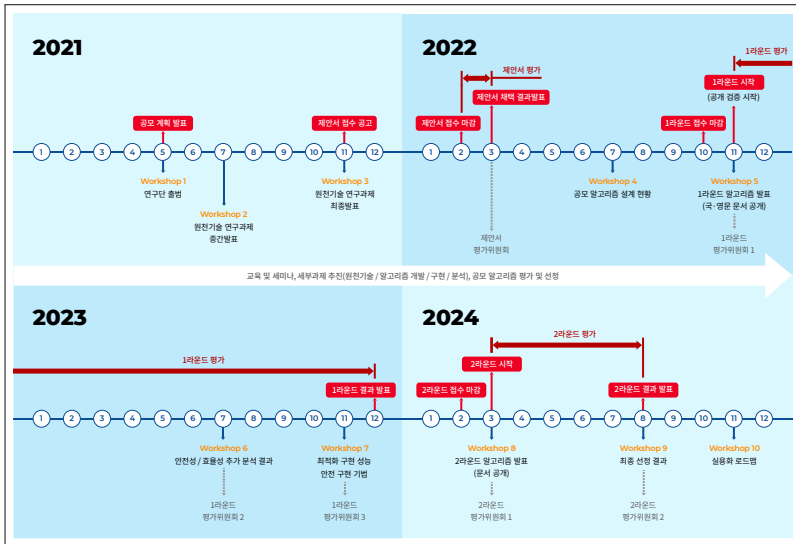
KpqC Competition

KpqC Algorithms

Current Status of KpqC

1. KpqC Competition

Roadmap of the KpqC Competition



- To have some competitive Korean PQC algorithms was valuable. But,
 - ▶ Fostering domestic PQC research was our greater priority.
- Began with 0th Round of collecting '*design and development **plan***'s. And,
 - ▶ Algorithm development was supported through a commissioned R&D project.
- Functional Categories of Algorithms
 - ▶ Key Encapsulation Mechanism (KEM) / Public Key Encryption (PKE)
 - ▶ Digital Signature Algorithm (DSA)

- The algorithm development teams
- The researchers who cryptanalyzed and implemented the algorithms
- A team at TU/e was our main international partner.
 - ▶ Thanks to Prof. Tanja Lange, who served as project leader on their side.
- ※ “*The NIST PQC Standardization Process*” was certainly a key model. But,
 - ▶ No authority to issue official standards

1st Round started in November 2022.

Underlying Problem	KEM / PKE	DSA	Total
Lattice	3	5	8
Code	3	1	4
Multivariate	—	1	1
Isogeny	—	1	1
Other	1	1	2
Total	7	9	16

Round 1 Candidates (2)



16 candidates across various underlying hard problems

Underlying Problem	KEM / PKE	DSA	Total
Lattice	NTRU+, SMAUG, TiGER	GCKSign, HAETAE, NCC-Sign, Peregrine, SOLMAE	8
Code	Layered ROLLO-I, PALOMA, REDOG	Enhanced pqsigRM	4
Multivariate	—	MQ-Sign	1
Isogeny	—	FIBS	1
Other	IPCC	AIMer	2
Total	7	9	16

- Address: <https://groups.google.com/g/kpqc-bulletin>
- The need for a platform
 - ※ Some research results reached us only informally.
 - ▶ To share **cryptanalytic results** and provide **credit** and a **timestamp** for them
 - ▶ To share information such as algorithm **revisions** and **benchmarks**
 - ▶ To provide a **channel** for discussion

cf.) NIST pqc-forum

(<https://groups.google.com/a/list.nist.gov/g/pqc-forum>)

- (2022-12, IPCC)
A flaw in the design was identified, allowing efficient plaintext recovery.
- (2022-12, GCKSign)
A mistake in the quantitative security estimation was found.

- (2023-03, MQ-Sign)
Practical key-recovery or forgery attacks for some variants
 - Aulbach et al. *Practical Key-Recovery Attack on MQ-Sign and More*. PQC (2024)
 - Ikematsu et al. *A Security Analysis on MQ-Sign*. WISA (2024)
- (2023-04, Layered ROLLO-I)
A back-and-forth of attacks and revisions
 - Chee et al. *Analysis of Layered ROLLO-I: A BII-LRPC code-based KEM*. IACR CiC (2024)

- (2023-07, REDOG)
Identifying a decryption flaw and providing a message-recovery attack
→ Lange et al. *On the security of REDOG*. ICISC (2024)
- (2023-10, Peregrine)
Practical key recovery exploiting leaked information from signatures
→ Lin et al. *Cryptanalysis of the Peregrine Lattice-Based Signature Scheme*. PKC (2024)

- (2023-06, NTRU+)
Theoretical IND-CCA attack exploiting an ambiguity in SOTP Inv. function
→ Lee et al. *Cryptanalysis on “NTRU+: Compact Construction of NTRU Using Simple Encoding Method”*. IEEE TIFS (2024)
- In response, a check routine was added within the Inv. function.

- (2023-07, AIMer)
A number of algebraic attacks
 - Liu et al. *Algebraic Attacks on RAIN and AIM Using Equivalent Representations*. ToSC (2023)
 - Zhang et al. *Algebraic Attacks on Round-Reduced Rain and Full AIM-III*. ASIACRYPT (2023)
- In response, the underlying primitive AIM was replaced with AIM2.

- Jolijn Cottaar et al. ***Report on evaluation of KpqC candidates.*** Cryptology ePrint Archive, Paper 2023/1853. 2023
 - ▶ Written by a team of 10 researchers from Eindhoven University of Technology
 - ▶ Analyzing all Round 1 candidates
 - ▶ Including analysis results not separately published elsewhere

2nd Round started in April 2024.

Underlying Problem	KEM / PKE	DSA	Total
Lattice	2	2	4
Code	2	—	2
Multivariate	—	1	1
Symmetric Key	—	1	1
Total	4	4	8

8 candidates of 2nd Round

Underlying Problem	KEM / PKE	DSA	Total
Lattice	NTRU+, SMAUG-T [*]	HAETAE, NCC-Sign	4
Code	PALOMA, REDOG	—	2
Multivariate	—	MQ-Sign	1
Symmetric Key	—	AIMer	1
Total	4	4	8

^{*} merger of SMAUG and TiGER

- (2024-09, REDOG)
A message recovery attack exploiting the low rank weight of the ciphertext
→ Pellegrini et al. *Analysis of REDOG: The Pad Thai Attack*. PQC (2025)
- (2024-11, MQ-Sign)
A forgery attack exploiting the secret key structure of a variant
→ Ran et al. *"Shifting Our Knowledge of MQ-Sign Security"*. PQC (2025)

- Daniel J. Bernstein et al. ***Report on evaluation of KpqC Round-2 candidates.***
Cryptology ePrint Archive, Paper 2024/2077. 2024
 - ▶ Written by a team of 17 researchers
 - ▶ Analyzing all Round 2 candidates
 - ▶ SUPERCOP benchmarks and patent issues
- More (cryptanalytic and non-cryptanalytic) results on the KpqC-bulletin
 - ▶ Notably, a number of implementation-related results

In January 2025,

Underlying Problem	KEM	DSA
Lattice	NTRU+, SMAUG-T	HAETAE
Symmetric Key	—	AIMer

2. KpqC Algorithms

- Specifications of *the KpqC algorithms* were provided in January 2026.
 - ▶ The competition was officially concluded.
 - ▶ The algorithms could be revised before being standardized.
 - ▶ Current versions are available on each algorithm's website.

Algorithm		Website
KEM	NTRU+	https://www.ntruplus.org/
	SMAUG-T	https://kpqc.cryptolab.co.kr/smaug-t
Signature	AIMer	https://aimer-signature.org/
	HAETAE	https://kpqc.cryptolab.co.kr/haetae

NTT-Friendly NTRU Scheme Using Simple Encoding Method

- KEM based on the standard NTRU structure
- Using NTT-friendly rings over cyclotomic trinomials
- SOTP (Semi-generalized One-Time Pad), a new message encoding method

† Jonghyun Kim et al. **“NTRU+: Compact Construction of NTRU Using Simple Encoding Method”**. In: *IEEE TIFS* 18 (2023)

Ring $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$, where $f(x) = x^n - x^{n/2} + 1$

Parameter	n	q
NTRU+768	768	3457
NTRU+864	864	
NTRU+1152	1152	

Message encoding method SOTP

- Message distribution conversion
 - ▶ From arbitrary binary distribution
 - ▶ To (ternary) CBD^a
- Key features in security proof
 - ▶ Average-Case to Worst-Case
 - ▶ OW-CPA $\rightarrow$ IND-CPA
 - ▶ Tight reduction

^aCentered Binomial Distribution

Encode($x \in \mathcal{M}'$, $u \leftarrow U^{2n}$)

- 1: $u = (u_1, u_2) \in \{0, 1\}^n \times \{0, 1\}^n$
- 2: $y = (x \oplus u_1) - u_2 \in \{-1, 0, 1\}^n$
- 3: **return** y

Decode($y \in \mathcal{M}$, $u \in U^{2n}$)

- 1: $u = (u_1, u_2) \in \{0, 1\}^n \times \{0, 1\}^n$
- 2: **if** $y + u_2 \notin \{0, 1\}^n$, **then**
- 3: **return** $\perp$
- 4: $x = (y + u_2) \oplus u_1 \in \{0, 1\}^n$
- 5: **return** x

Size and Performance Comparison (of the NTRU+ specification document)

Scheme	Size (bytes)			Optimized C (K cycles)		
	pk	ct	sk	Gen	Encap	Decap
NTRU+768	1152	1152	2336	99	75	78
NTRU+864	1296	1296	2624	111	87	90
NTRU+1152	1728	1728	3488	176	118	126
Kyber512	800	768	1632	135	159	203
Kyber768	1184	1088	2400	232	258	318
Kyber1024	1568	1568	3168	334	388	463
ntruhs2048509	699	699	935	9296	624	1657
ntruhrss701	1138	1138	1450	17414	1060	3119
ntruhs2048677	930	930	1234	16148	1071	2900
ntruhs4096821	1230	1230	1590	21879	2003	4261

Key Encapsulation Mechanism based on MLWE and MLWR

- Merger of SMAUG and TiGER
- KEM based on the standard LWE scheme
- Key Generation based on MLWE, Encryption based on MLWR

† Jung Hee Cheon et al. **“SMAUG (-T), Revisited: Timing-Secure, More Compact, Less Failure”**. In: *IEEE Access* 12 (2024)

Rank- k module over $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ and rounding modulus p, p'

Parameter	n	k	q	(p, p')
TiMER	256	2	1024	(256, 8)
SMAUG-T Mode 1		2	1024	(256, 32)
SMAUG-T Mode 3		3	2048	(512, 16)
SMAUG-T Mode 5		4	2048	(512, 128)

Evolution of SMAUG

- Lizard

- ▶ Cheon et al. *Lizard: Cut Off the Tail! A Practical Post-quantum Public-Key Encryption from LWE and LWR*. SCN (2018)

→ RLizard

- ▶ Lee et al. *RLizard: Post-Quantum Key Encapsulation Mechanism for IoT Devices*. IEEE Access (2019)

→ SMAUG

- ▶ Cheon et al. *SMAUG: Pushing Lattice-Based Key Encapsulation Mechanisms to the Limits*. SAC (2023)

Evolution of TiGER

- Lizard

→ LizarMong

- ▶ Jung et al. *LizarMong: Excellent Key Encapsulation Mechanism Based on RLWE and RLWR*. ICISC (2019)

→ TiGER

- ▶ Park et al. *TiGER: Tiny bandwidth key encapsulation mechanism for easy miGration based on RLWE(R)*. IACR ePrint (2022)

Size and Performance Comparison (of the SMAUG-T specification document)

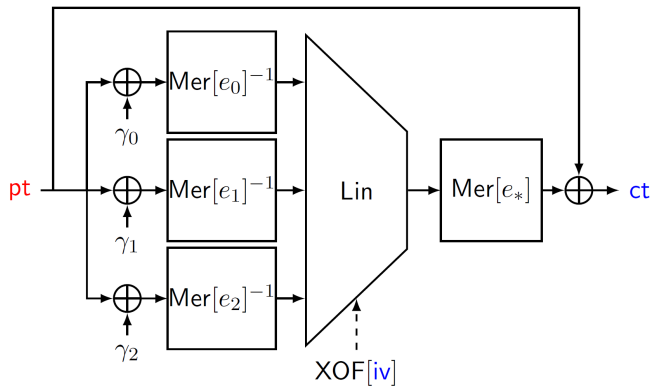
Scheme	Size (bytes)			Reference (K cycles)		
	pk	ct	sk	KeyGen	Encap	Decap
TiMER	672	608	832	110	100	135
SMAUG-T Mode 1	672	672	832	110	100	136
Kyber512	800	768	1632	128	158	187
SMAUG-T Mode 3	1088	992	1312	219	204	253
Kyber768	1184	1088	2400	209	255	286
SMAUG-T Mode 5	1440	1376	1728	357	334	414
Kyber1024	1568	1568	3168	321	369	414

Signature from ZKP of Preimage Knowledge for a One-way Function

- Security relies on symmetric-key primitive.
 - ▶ One-way function AIM2 (a dedicated block cipher)
- Based on MPCitH (MPC-in-the-Head) paradigm
- Customized BN++ ZKP (Zero Knowledge Proof)

† Seongkwang Kim et al. “**AIM: Symmetric Primitive for Shorter Signatures with Stronger Security**”. In: *ACM CCS* (2023)

AIM2 with $l = 3$



Cryptanalysis

Target	Security	Time	Memory
AIM2-I	128	$2^{103.66}$	$2^{100.33}$
AIM2-III	192	$2^{147.46}$	$2^{144.17}$
AIM2-V	256	$2^{213.19}$	$2^{209.91}$

- ▶ Alex Biryukov et al. **“Magic Pot: Cryptanalysis of Full AIM2 in the Standard and Related-/reused-Key Settings Using New Elimination Framework”**. In: *EUROCRYPT* (2026)

→ Updated version released on 2026-08-19

Shorter Lattice-Based Fiat-Shamir Signature

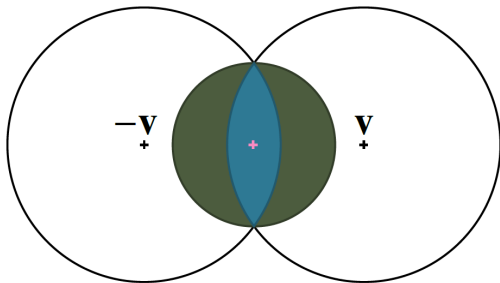
- Fiat-Shamir with aborts paradigm
- Based on MLWE and MSIS
- Bimodal Distributions & Hyperball Sampling
 - size reductions
 - ※ ML-DSA: Unimodal Distributions & Hypercube Sampling

† Jung Hee Cheon et al. “**HAETAE: Shorter Lattice-Based Fiat-Shamir Signatures**”. In: *TCHES* 2024.3 (2024)

$k \times l$ module over $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$

Parameter	n	k	l	q
HAETAE-2	256	2	4	64513
HAETAE-3		3	6	
HAETAE-5		4	7	

HAETAE (Hyperball bimodal module rejection signature scheme)



The HAETAE eyes (from the specification document)



(from the website of HAETAE)

Identifying two issues requiring revision, through their own review

- In the discrete Gaussian Sampler,
 - ▶ CDT table: insufficient range and precision
 - ▶ Function approximation: insufficient polynomial degree
- No attack was found but a proof-level fix was required.

→ Updated version released on 2026-08-25

3. Current Status of KpqC

NTRU+: Updated in July 2026

- Specification (2026-07-10)
 - ▶ Add a note on page 42 to reject non-canonical Decode_q outputs.

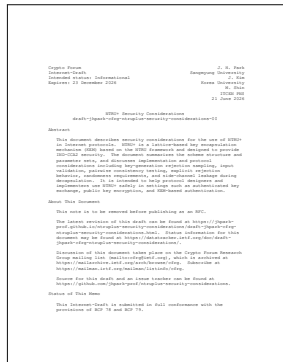
SMAUG-T: Updated in May 2026

- Specification (2026-05-21)
 - ▶ Add a pair of parentheses in Figure 4, spCBD_3/16, step 2, to clarify the order of computation.

Minor revisions during the standardization process

- (2026-03) IETF CFRG^a ‘KEM Security Considerations’

- ▶ Adopting per-KEM security evaluation documents
“The proposed output is an adoption of a group of documents that provides security considerations for protocols.”
- ▶ 7 drafts are currently active, including NTRU+.
- ※ draft-jhpark-cfrg-**ntruplus**-security-considerations-00



NTRU+ Security Considerations

^aCrypto Forum Research Group

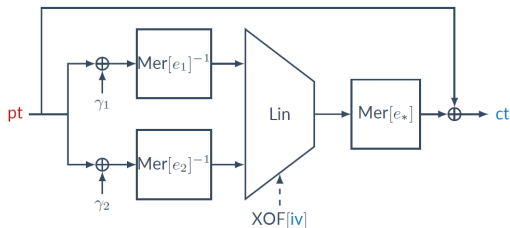
Updated in August 2026

- Specification (2026-08-19)

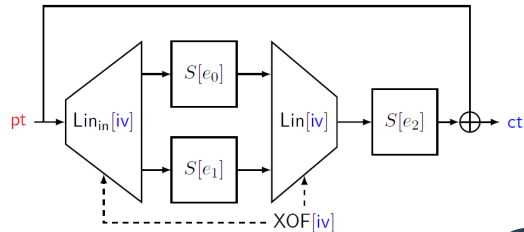
- The symmetric primitive AIM2 was replaced by AIM3.

- Enhanced S-box: $\text{Mer}[e](x) = x^{2^e-1} \rightarrow S[e](x) = x^{2^e-1} + x^{-1}$
 - Fixed constants adding $\rightarrow$ iv-dependent affine layer

AIM2 with $l = 2$



AIM3 with $l = 2$



Parameters of AIM3

	λ	l	e_0	e_1	e_2	e_3
AIM3-I	128	2	3	7	11	-
AIM3-III	192	2	11	23	47	-
AIM3-V	256	3	3	7	11	19

Size (in bytes) Comparison (of the AIMer specification document)

Parameters	pk	sk	Sig. (v2.1)	Sig. (v3.0)	SLH-DSA
AIMer-128f	32	48	5,888	6,944	17,088
AIMer-128s			4,160	4,704	7,856
AIMer-192f	48	72	13,056	15,408	35,664
AIMer-192s			9,120	10,320	16,224
AIMer-256f	64	96	25,120	31,360	49,856
AIMer-256s			17,056	20,224	29,792

※ Performance evaluation is in progress.

Updated in August 2026

- Specification (2026-08-25)
 - ▶ More precision in the discrete Gaussian Sampler
 - CDT table: 64 entries of 16-bit precision with 4σ tail
→ 166 entries of 83-bit precision with 10.375σ tail
 - Function approximation: degree-7 polynomial → 10-degree polynomial

Size and Performance Comparison (of the HAETAE specification document)

	Size (bytes)		Reference (K cycles)		
	pk	Sig.	KeyGen	Sign	Verify
Dilithium-2	1,312	2,420	221	1014	241
Dilithium-3	1,952	3,293	423	1602	383
Dilithium-5	2,592	4,595	599	1958	623
HAETAE-2	992	1,474	1249	6482	179
HAETAE-3	1,472	2,349	2055	8058	308
HAETAE-5	2,080	2,948	1381	11741	361
Falcon-512	897	666	43736	12860	66
Falcon-1024	1,792	1,280	127702	28068	140

- NTRU+, SMAUG-T
 - ▶ Proposed for KS (Korean Industrial Standard) establishment
 - ▶ Standardization is underway, hopefully completed by the end of 2026.
- AIMer, HAETAE
 - ▶ Algorithms have been revised.
 - ▶ KS proposal to follow after appropriate verification



Thank You.

Information

- (Website) <https://www.kpqc.or.kr/>
- (YouTube) <https://www.youtube.com/@kpqcrypto>

Contact

- **<mailto:kpqcrypto@gmail.com>**