

Recent Updates on SOLMAE

김 광조

세계암호학회(IACR) Fellow
국제사이버보안연구원(IRCS) 원장
KAIST 전산학부 명예교수

kkj@kaist.ac.kr

<https://caislab.kaist.ac.kr/kkj>

Contents

1. Overview of SOLMAE
2. Paper at SCIS2023@Kogura, Japan
3. SOLMAE by Prest in RWPQC2023@Tokyo, Japan
4. Paper CISC-S23@Kangwon U., Korea
5. 2 Invited talks to China
6. KpqC paper competition
7. Invited talk to LG U+@Seoul, Korea
8. 2 papers on SOLMAE accepted to AS23
9. Summary

Overview of SOLMAE



<http://solmae-sign.info>

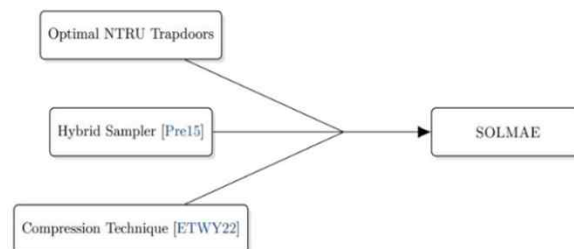
solmae-sign.info



[News](#) [Resources](#) [About us](#) [Publications](#)

SOLMAE: quantum-Secure algOrithm for Long-term Message Authentication and Encryption

This webpage introduces the SOLMAE signature scheme submitted to the Korean Post-Quantum Competition. SOLMAE is a lattice-based signature scheme inspired by several pioneering works and stands for quantum-Secure algOrithm for Long-term Message Authentication and Encryption. At its core, it is based on the hash-then-sign signature paradigm proposed by Gentry, Peikert and Vaikuntanathan[GPV08]. To be efficiently instantiated, this framework needs a class of lattices enjoying efficiently computable trapdoor bases for the signing procedure.



✓ 국제사이버보안연구원, 한국형 양자내성 전자서명 방식 '솔매' 개발

<https://www.boannews.com/media/view.asp?idx=119368&page=1&kind=3>

quantum-Secure algOrithm for Long-term Message Authentication and Encryption



- ▶ Korean PI

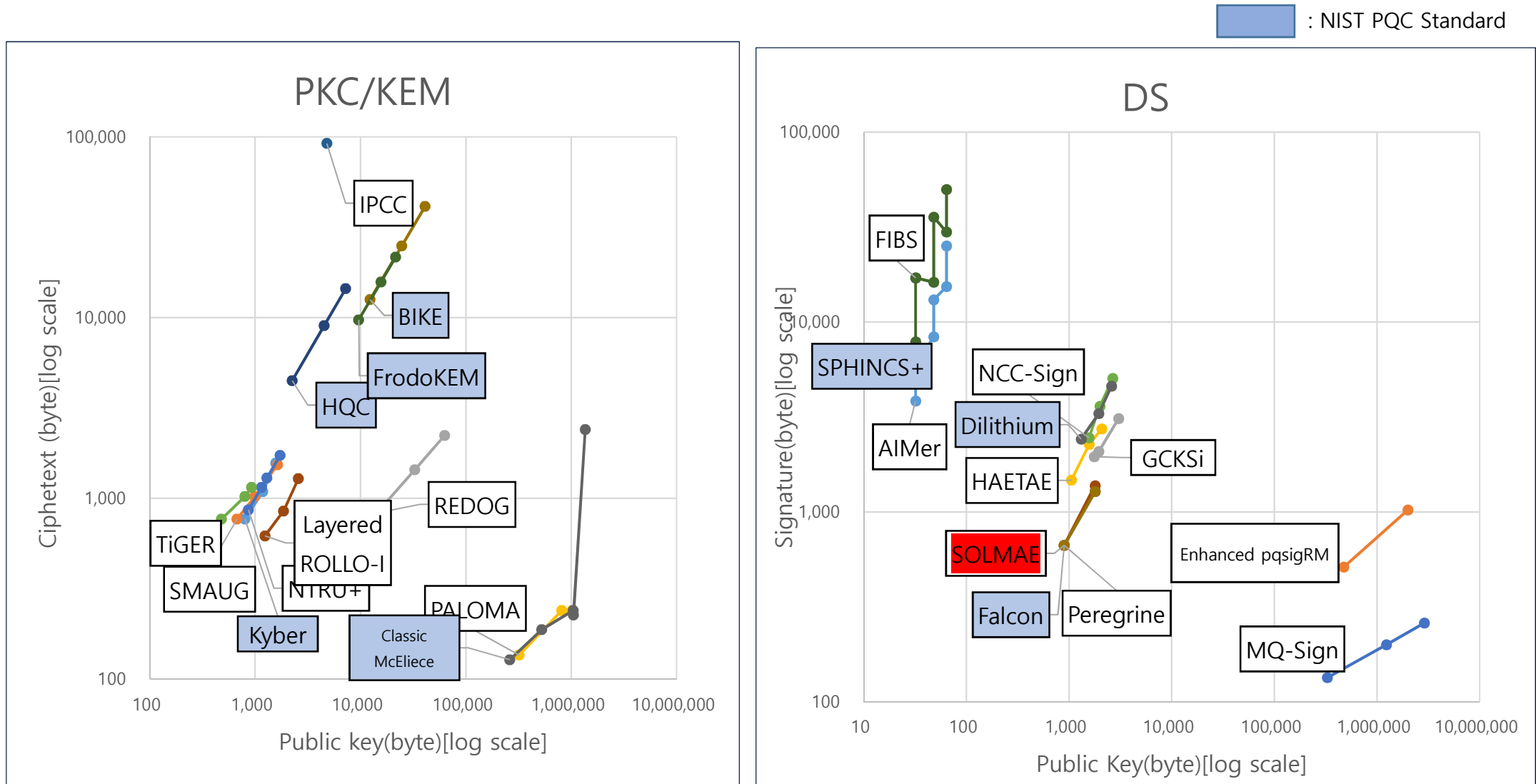
- ▶ [Kwangjo Kim](https://caislab.kaist.ac.kr/~kkj) (President@IRCS, Emeritus Prof.@KAIST),
<https://caislab.kaist.ac.kr/~kkj>

- ▶ Member (7 persons)

- ▶ [Mehdi Tibouchi](https://www.normalesup.org/~tibouchi/) (NTT, Japan), <https://www.normalesup.org/~tibouchi/>
 - ▶ [Thomas Espitau](https://espitau.github.io/) (PQshield, UK), <https://espitau.github.io/>
 - ▶ [Alexandre Wallet](https://awallet.github.io/) (INRIA Rennes, France), <https://awallet.github.io/>
 - ▶ [Yang Yu](https://yuyang-crypto.github.io/) (Tsinghua University, China), <https://yuyang-crypto.github.io/>
 - ▶ [Akira Takahashi](https://akiratk0355.github.io/) (U. of Edinburgh, Scotland), <https://akiratk0355.github.io/>
 - ▶ [Sylvain Guilley](https://perso.telecom-paristech.fr/guilley/) (Secure-IC, France), <https://perso.telecom-paristech.fr/guilley/>
 - ▶ [Seungki Kim](https://sites.google.com/view/seungki/home) (U. of Cincinnati, USA), <https://sites.google.com/view/seungki/home>



Comparison of KpqC R1 Algorithms*



*from KRnet2023 presentation by NSR

SOLMAE was proposed to KpqC Competition

Prof. Kwangjo Kim
IACR Fellow

Emeritus Professor@KAIST / President@IRCS

Acknowledgement : My travel to SCIS2023 was fully supported by Prof. Kouichi Kakurai, Kyushu Univ.

1st RWPQC2023@Tokyo in Mar. 26, 2023

<https://rwpqc.org/>



We are the Real World PQC workshop.

Our aim is to bring together the industry, academia and standardization bodies to think about the tasks we will encounter ahead of integrating post-quantum algorithms to the networks, protocols and systems we use today. Our aim will be informed by sharing knowledge about the latest advances of cryptanalysis, the state of different algorithms, and implementation concerns.

We believe it is very important to start organizing around this topic now as we begin to face the challenges of migrating our systems to post-quantum cryptography. Welcome to the first installment of our workshop!

RWC2023 by IACR was held Mar. 27-29, Tokyo, Japan

SCHEDULE

Activity

Workshop Kick-Off – Daniel Apon (MITRE)

Opening Remarks – Charles Clancy (MITRE)

Invited Talk #1

NIST and PQC – Dustin Moody (NIST)

Lessons Learned Talks of the selected candidates

Kyber – Peter Schwabe (Max Planck Institute for Security and Privacy)

Silithium – Vadim Lyubashevsky (IBM Research)

Falcon – Thomas Prest (PQShield)

SPHINCS+ – Andreas Hölsing (Eindhoven University of Technology)

Coffee Break

Invited Talk #2

IETF Efforts in PQC – Douglas Stebila (University of Waterloo)

Roundtable #1 – Implementation and side channels

Moderator: James Howe

Participants: Thomas Pornin (NCC), Tobias Schneider (NXP), Daniel Genkin (Georgia Tech), Mélissa Rossi (ANSSI), Patrick Longa (Microsoft Research)

Lunch Break

Invited Talk #3

ANSSI's recommendations on the migration plan – Mélissa Rossi (ANSSI)

Roundtable #2 – Industry side discussion

Moderator: Marc Manzano

Participants: Carlos Aguilar (SandboxAQ), Rafael Misoczki (Meta), Craig Costello (Microsoft), Panos Kampanakis (AWS), Tancrede Lepoint (AWS), Scott Fluhrer (Cisco)

Coffee Break

Roundtable #3 – Current state of cryptanalysis

Moderator: Daniel Apon

Participants: Ward Beullens (IBM Research), Chloe Martindale (University of Bristol), Martin Albrecht (SandboxAQ, King's College London), Leo Ducas (CWI), Ray Periner (NIST)

Invited Talk #4

The bright present and future of lattice-based zero-knowledge proofs – Vadim Lyubashevsky (IBM Research)

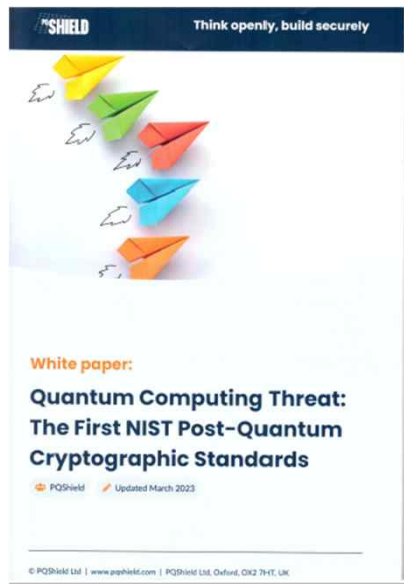
Recent Results / Lightning Talks: GlowingMoltenHotRealPQC

(5 minute slots) Submissions closed: please see below for information.

Moderator: Sofia Celi

Closing remarks – Sofia Celi (Brave)

Prest's presentation@1st RWPQC



4 Falcon (Secondary standard)

Type: Signature
Paradigm: Hash-then-sign
Family: Lattices
Hard Problems: NTRU
Sym. primitives: SHAKE-256
Randomness: Noncentered discrete Gaussians
Specification: [PPH+22]
Website: <https://falcon-sign.info/>
Related Works: [p4IP+03, GPV08, SS13, DUP14, DP16, OS4G19]

NIST's overall assessment [NIST22b]
 "Falcon was chosen for standardization because NIST has confidence in its security (under the assumption that it is correctly implemented) and because its small bandwidth may be necessary in certain applications."

Design
 Falcon is based on the GPU framework (GPV08) for obtaining hash-then-sign schemes over lattices. As done in [SS13, DUP14], the design is instantiated over the very compact class of NTRU lattices [PPH+22] in order to minimize the bandwidth cost. Falcon is the selected standard with the smallest communication cost (public key + signature).

Algorithmic optimisations
 Falcon exploits the algebraic structure of cyclotomic rings in order to optimize its efficiency, notably via the use of a Fast Fourier Sampling algorithm [DP14] in the signing procedure, and of a tower-of-rings algorithm [PPH+22] during key generation. Both algorithms yield a $O(n)$ factor improvement compared to previous algorithms, in being the degree of the base ring $\mathbb{Z}[X]/(X^k + 1)$.

NIST level	(SK) (bytes)	(PK) (bytes)	(Sig) (bytes)	KG (cycles)	Sign (cycles)	Verify (cycles)
1	-	897	668	19872000	386678	82329
3	-	-	-	-	-	-
5	-	1793	1280	63133000	789564	168498

© PQShield Ltd | www.pqshield.com 13 of 32

VARIANTS
 A few variants of Falcon have been proposed, such as a module version, ModFalcon [DP+20], as well as two variants based on a different sampling algorithm: Mitaka and SOLMAE (Korean PQC submission). In addition, a ring signature variant has been proposed: Rapier [JA219].

Implementation
 Falcon uses floating-point arithmetic (FPA), which can make its implementation delicate on platforms that don't support FPA natively. In this case, FPA needs to be emulated. [DP16, PPH+18] have proposed implementations of Falcon on ARM Cortex-M4; both use memory-laziness tricks in order to reduce its memory footprint.

Physical attacks
 While the side-channel resistance of Falcon has been less studied than for Dilithium, side-channel attacks against unprotected implementations of Falcon have been proposed recently. The first one [KA21, GMR22] targets floating-point multiplications, the second and third ones [GMR22, ZYVW22] are side-channel assisted variations of the hidden parallelized attack [NR06].

Specification
 → NIST draft standard: 2023-2024?
 → IETF draft?

Design evolution
 → SOLMAE [KTW+22] [Korean PQC submission]

* [SOLMAE] uses the same simple, fast, parallelizable signing algorithm as Mitaka [...]. However, by leveraging a novel key generation algorithm [...], SOLMAE achieves the same high security and short key and signature sizes as Falcon. *

Suggestion are welcome!

PS: feel free to grab a physical copy of our white paper 📄
 "The First NIST Post-Quantum Cryptographic Standards"

There is one nice design evolution of Falcon that I like.

MITRE
 SANDBOXAQ

<https://www.youtube.com/watch?v=J0QpSV2xSvM&t=17347s>

How SOLMAE was designed

Prof. Kwangjo Kim

[IACR Fellow](#)

President@ [IRCS](#) / Emeritus Professor@ [KAIST](#)

Key Features of RSA and SOLMAE

Table 1: Key Features of RSA and SOLMAE

Item	RSA	SOLMAE
Mathematics	Number Theory	Algebra
Basic operation	Mod. Exp.	Polynomial
Trapdoor	Mul. Inverse	NTRU
Verification	$Left = Right$	$Left \leq Right$
Gaussian sampling	No	Yes
Security assumption	Integer Fact.	SIS
Worst to avg. red.	No	Yes
Classical attack	No	No
Quantum attack	Yes	No

Comparison of SOLMAE and ECDSA

Table 3: Comparison of SOLMAE and ECDSA

		SOLMAE ECDSA	
Specification		512	P256r1
Size(Bytes)	pk	1,792	65
	sgn	1,375	32
Time	KeyGen(ms)	30.21	2.53
	Sign(μ s)	288.2	2,582.8
	Verif(μ s)	55.6	7,744.7

A blue starburst-shaped callout containing the text '10X Faster' in red.

10X
Faster

How SOLMAE was developed to meet KpqC Competition

Prof. Kwangjo Kim

[IACR Fellow](#)

President@ [IRCS](#) / Emeritus Professor@ [KAIST](#)

Current Status of KpqC Standard Competition in Korea

IACR Fellow Kwangjo Kim

President@IRCS/Emeritus

Prof@KAIST

kkj@kaist.ac.kr

<https://mp.weixin.qq.com/s/vsjRK24NPWWDKzz6ryTKoQ>

✓ 家：越早 后量子公 究， 于中 得相 主 越有利*

* Domestic experts: The sooner you participate in post-quantum public key research, the more beneficial it will be for China to gain the relevant initiative.

Comparison of SOLMAE vs. FALCON by Python

Theoretical and Empirical Analysis of FALCON and SOLMAE using their Python Implementation

Anonymous Authors

No Institute Given

Abstract. Since NIST has recently selected FALCON as one of quantum-resistant digital signatures which uses the hash-and-sign paradigm in the style of Gentry–Peikert–Vaikuntanathan framework and instantiated over NTRU lattices, SOLMAE as a variant of FALCON was submitted to KpqC standard competition by taking all the pros of FALCON and MITAKA and reducing their cons as much as possible. In this paper, we suggest the asymptotic computational complexity of FALCON and SOLMAE take $\Theta(n \log n)$ in their **KeyGen**, **Sign** and **Verif** procedures simultaneously, but our computer experiments using their Python implementation exhibit empirically that **KeyGen** of FALCON-512 takes longer time than that of SOLMAE-512 by about a second while the other two procedures are running almost the same time. We show a sample execution of FALCON-512 and SOLMAE-512 with their real value are described in detail for the educational purpose to understand FALCON and SOLMAE easily. We also checked the Gaussian randomness of *N-Sampler* and *UnifCrown* samplers used in SOLMAE only.

Keywords: Lattice-based cryptography · Hash-and-sign paradigm · NTRU trapdoors · Discrete Gaussian sampling · Python implementation

1 Introduction

When Shor [16] has proposed an efficient randomized algorithm on a hypothetical quantum computer a 1999 to integer factorization and discrete logarithm problems in a polynomial time, it was beyond our imagination building for the powerful computing environment at that time. Currently the threat of attacking the current (or classical) secure system by using the quantum computer is expected to be right at our fingertips due to the aggressive road map by IBM quantum computing. We are very concerned about so called *Harvest now, decrypt later* attack [17] which is a surveillance strategy that relies on the acquisition and long-term storage of currently unreadable encrypted data awaiting possible breakthroughs in decryption technology that would render it readable in the future.

Due to the substantial amount of research on quantum computers, large-scale quantum computers built, can break many public-key cryptosystems based on the number-theoretic hard problems in use. In 2016, NIST [14] has initiated Post Quantum Cryptography(PQC) project to solicit, evaluate, and standardize one or more quantum-resistant cryptographic algorithms for Key Encapsulation Mechanism(KEM) and Digital Signature(DS) worldwide. After several rounds, NIST has finally elected CRYSTALS-Kyber for KEM and CRYSTALS-Dilithium, FALCON, and SPHINCS+ for DS in 2022.

Influenced by this NIST PQC project, Korean cryptographic society led by KpqC task force [11] as called for soliciting Korean PQC standard candidates by the end of Oct. in 2022. By the due of submission, 7 candidates KEM and 8 candidates DS for KpqC competition were submitted and their details are available at <https://kpqc.or.kr/>.

SOLMAE¹ was submitted to KpqC Competition as one of DS candidate algorithms which is a lattice-based signature scheme inspired by several pioneering works based on the hash-then-sign signature paradigm proposed by Gentry, Peikert and Vaikuntanathan [6].

SOLMAE is inspired from FALCON's design. Some of the new theoretical foundations were laid out in the presentation of Mitaka [1]. At a high level, it removes the inherent technicality of the sampling procedure, and most of its induced complexity from an implementation standpoint, or free, that is with no loss of efficiency. This simplicity translates into faster operations while

¹ This is an acronym of quantum-Secure algorithm for Long-term Message Authentication and Encryption.

8 Concluding Remarks

FALCON is claimed to have the advantage of providing short public keys and signatures as well as high-security levels; plagued by a contrived signing algorithm, not very fast for signing and hard to parallelize; very little flexibility in terms of parameter settings. However, SOLMAE has a simple, fast, parallelizable signing algorithm, with flexible parameters with its novel key generation algorithm.

In this paper, after giving a brief description of the specification of FALCON and SOLMAE, we found that their asymptotic computational complexity of **KeyGen**, **Sign** and **Verif** procedures take $\Theta(n \log n)$ simultaneously. Also, our computer experiments using their Python implementation exhibit empirically that **KeyGen** of FALCON-512 only takes longer time than that of SOLMAE-512 by about a second. But we can say that this is not an exact evaluation of their performance by Python implementation.

Further work such as elaborated analysis of computational complexity on FALCON and SOLMAE asymptotically is left to do next.

Acknowledgement

Omitted for anonymous submission.

References

1. Espitau, T., Fouque, P.A., Gérard, F., Rossi, M., Takahashi, A., Tibouchi, M., Wallet, A., Yu, Y.: Mitaka: a simpler, parallelizable, maskable variant of falcon. *Advances in Cryptology, Proc. of EUROCRYPTO 2022, Part III* pp. 222–253 (2022) 1, 8
2. Espitau, T., Tibouchi, M., Wallet, A., Yu, Y.: Shorter hash-and-sign lattice-based signatures. *Advances in Cryptology, Proc. of CRYPTO 2022, Part II* pp. 245–275 (2022) 8
3. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over ntru, <https://falcon-sign.info/> 5, 6, 12, 16
4. Fouque, P.A., Kirchner, P., Tibouchi, M., Wallet, A., Yu, Y.: Key recovery from gram-schmidt norm leakage in hash-and-sign signatures over ntru lattices. *Cryptology ePrint Archive, Paper 2019/1180* (2019), <https://eprint.iacr.org/2019/1180>, <https://eprint.iacr.org/2019/1180> 6
5. G. E. P. Box, M.E.M.: A note on the generation of random normal deviates. *Ann. Math. Statist.* pp. 610–611 (1958) 15
6. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). <https://doi.org/10.1145/1374376.1374407> 1, 5
7. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. *Advances in Cryptology, Proc. of Crypto 1997* pp. 112–131 (1997) 5
8. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21–25, 1998. Lecture Notes in Computer Science*, vol. 1423, pp. 267–288. Springer (1998) 4
9. Kim, K., Tibouchi, M., Espitau, T., Takashima, A., Wallet, A., Yu, Y., Guilley, S., Kim, S.: Solmae : Algorithm specification. Updated SOLMAE, IRCS Blog (2023), <https://ircs.re.kr/?p=1714> 7, 8, 9, 10, 14, 18
10. Kim, W.: *Mathematical Statistics*(in Korean). Minyoungsa, Seoul, Korea (2021) 15
11. KpqC: Korean post-quantum cryptography (2020), <https://kpqc.or.kr/> 1
12. Min, S., Yamamoto, G., Kim, K.: Weak property of malleability in ntru-sign. *Proc. of ACISP 2004, LNCS 3108* pp. 379–390 (2004) 5
13. Nguyen, P.Q., Regev, O.: Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures. *Journal of Cryptology* **22**(2), 139–160 (2009) 5, 14
14. NIST: Post-quantum cryptography (2016), <https://csrc.nist.gov/projects/post-quantum-cryptography> 1
15. Pornin, T., Prest, T.: More efficient algorithms for the NTRU key generation using the field norm. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 504–533. Springer, Heidelberg (Apr 2019). https://doi.org/10.1007/978-3-030-17259-6_17 5, 10

Invited Talk to LG U+

LGUPLUS

양자내성암호 기술 세미나

디지털 암호에서 양자내성 암호로의 변혁
(Paradigm shift from Digital Cryptography
to Post-Quantum Cryptography)

연사 김광조 교수님

- 세계암호학회 Fellow
- KAIST 전산학부 명예교수
- (사)국제사이버보안연구원 원장



일시 2023년 11월 01일 (수)
14:00 ~ 16:00

장소 마곡사옥 E9 B1층 인사이트홀
ZOOM 영상회의

내용 암호의 발전 과정
양자내성암호의 개요 및 활용



문의: 유선망개발팀 김연준님 (NW 공지사항 참고)



2023
양자내성암호
기술 세미나

디지털 암호에서 양자내성 암호로의 변혁

- | 일시_ 11월 1일 오후 2시
- | 장소_ B1 인사이트홀
- | 주최_ 김광조 교수
KAIST 명예교수
- | 내용_ 암호의 발전 과정
양자내성암호의 활용



2 Papers@Asiacrypt2023

25. Antrag: Annular NTRU Trapdoor Generation

Thomas Espitau, Thi Thu Quyen Nguyen, Chao Sun, Mehdi Tibouchi, Alexandre Wallet

PQShield, Inria Rennes, Osaka University, NTT Social Informatics Laboratories, Univ Rennes, Inria, CNRS, IRISA

Table 2. Practical parameter selection, power-of-two case

d	$q = 12289$		$q = 3329$	
	512	1024	512	1024
Quality α	1.15	1.23	1.23	1.48
Repetition rate M	3	4	4	4
Bit security (C/Q)	124/113	264/240	21/110	265/240
Verification key size (bytes)	896	1792	768	1536
Signature size (bytes)	646	1260	591	1176

Table 3. Practical parameter selection for ANTRAG, 3-smooth conductor case.

(a) Modulus $q = 12289$

d	648	768	864	972
Quality α	1.17	1.19	1.21	1.22
Repetition rate M	4	3	3	4
Bit security (C/Q)	166/151	196/178	222/201	251/227
Verification key size (bytes)	1134	1344	1512	1701
Signature size (bytes)	808	952	1069	1200

(b) Various moduli. For $d = 768, 864, 972$, the right column shows moduli of [14].

Modulus q	$d = 648$		$d = 768$		$d = 864$		$d = 972$	
	3889	9721	3329	18433	3727	10369	4373	17497
Quality α	1.32	1.19	1.39	1.16	1.40	1.23	1.40	1.18
Expected repetitions	4	4	4	3	4	3	4	4
Bit security (C/Q)	159/144	164/149	192/174	195/177	220/200	222/201	254/230	250/227
Verification key size (bytes)	972	1134	1152	1440	1296	1512	1580	1823
Signature size (bytes)	747	796	883	977	1000	1058	1133	1225

Table 4. Performance comparison with FALCON and MITAKA.

d	FALCON [33]		MITAKA [14]		This paper	
	512	1024	512	1024	512	1024
Quality α	1.17	1.17	2.04	2.33	1.15	1.23
Classical sec.	123	284	102	233	124	264
Key size (bytes)	896	1792	896	1792	896	1792
Sig. size (bytes)	666	1280	713	1405	646	1260
keygen speed (Mcycles)	—	—	—	—	15.4	55.2
keygen speed (ms)	4.7	13.8	1657*	6214*	5.7	20.5
sign speed (kcycles)	—	—	340	661	334	655
sign speed (μ s)	204	412	127	246	124	243
verif speed (kcycles)	—	—	23	46	23	46
verif speed (μ s)	21	43	9	17	9	17

* Timings for the optimized SageMath implementation (excluding NTRUSolve), since no C implementation exists.

85. On Gaussian sampling, smoothing parameter and application to signatures

Thomas Espitau, Alexandre Wallet, Yang Yu

PQShield, France, Univ Rennes, Inria, CNRS, IRISA, Tsinghua University, Beijing, China

Deliverables of SOLMAE Round 2 (if OK)



SOLMAE_R2 submission					almost
					N/A
Document	SOLMAE_v2.pdf				
extra					
	C				
		SOLMAE_512_test	**c		
		SOLMAE_1024_test	**c		
	Python				
		SOLMAE_512_py	**py		
		SOLMAE_1024_py	**py		
	etc				
		**py, **sage			
KAT					
	generator				
		katrng.{h,c}, PQCgenKAT_sign.c			
Reference Implementation					
	SOLMAE_512				
		SOLMAE_512_v_key			
		**c, **h	build		
			**o. kat_s512.exe	**req, **rsp	
	SOLMAE_1024				
		SOLMAE_1024_v_key			
		**c, **h	build		
			**o. kat_s1024.exe	**req, **rsp	
	SOLMAE_Intermediate				
Optimized Implementation					
	SOLMAE_512				
		SOLMAE_512fpu			
		SOLMAE_512avx2?			
		SOLMAE_512cmx4?			
	SOLMAE_1024				
		SOLMAE_1024fpu			
		SOLMAE_1024avx2?			
		SOLMAE_1024avx2?			
README.txt					

Summary

- After Round 1 submission of SOLMAE package,
The followings were done:
 1. SOLMAE web page (<https://solmae-sign.info>) was opened
 2. International (Japan and China) invited & domestic talks/
papers
 3. Python Implementation of SOLMAE(https://github.com/kjkim0410/SOLMAE_python_512) and comparison with FALCON
- 2 papers on SOLMAE accepted to Asiacrypt2023@China
- International reputations on SOLMAE are rapidly increasing
- Still lots of job are remaining!!

