격자 기반 NIST PQC 알고리즘과 표준(FIPS) 초안 차이점 분석

윤성우, 박지민, 목정현, 강찬영, 이석준

가천대학교

borok2311@gachon.ac.kr, jimin030907@gachon.ac.kr,
johnmok@gachon.ac.kr, 1120cy@gachon.ac.kr, junny@gachon.ac.kr

Analysis of differences between Lattice-based NIST PQC algorithm and their standard (FIPS) drafts

Yun Sungwoo, Park Jimin, Mok Junghyun, Kang Chanyoung, Lee Sokjoon

Gachon University

요 약

2023년 8월 Lattice 기반 공개키 암호 알고리즘에 대한 FIPS 203 표준 초안과 Lattice 기반 전자서명 알고리즘에 대한 FIPS 204 표준 초안(이하 FIPS 문서)이 공개되었다. 이 FIPS 문서들은 각각 2022년 7월 표준화 대상 양자내성암호 알고리즘으로 선정된 CRYSTALS-Kyber와 CRYSTALS-Dilithium(이하 CRYSTALS 알고리즘)을 기반으로 보안성 및 성능 측면에서 CRYSTALS 알고리즘을 일부 보완하여 작성되었다. 본 논문에서는 CRYSTALS 알고리즘과 FIPS 문서의 차이점과 변경된 이유를 분석하여 정리한다.

Ⅰ. 서 론

현대 암호학 관점에서 양자 컴퓨터의 발전은 기존 공개키 암호 시스템에 큰 위협을 주고 있다. 이러한 위협에 대비하기 위해 양자 내성 암호(PQC, Post-Quantum Cryptography)에 대한 중요성이 대두되면서 NIST는 PQC 표준화를 위한 공모전을 개최했다. 2022년 7월 총 3라운드를 거쳐 1개의 PKE/KEM 알고리즘과 3개의 전자서명 알고리즘을 채택하였다. 이후, 2023년 8월 NIST는 Lattice 기반의 PKE/KEM 알고리즘인 CRYSTALS-Kyber[1]를 표준화한 FIPS 203[2] 초안과 Lattice 기반의 전자서명 알고리즘인 CRYSTALS-Dilithium[3]을 표준화한 FIPS 204[4] 초안을 공개하였다. 이 과정에서 CRYSTALS 알고리즘의 일부분을 수정하였다. 본 논문에서는 FIPS 문서와 기존 알고리즘과의 차이점이 무엇인지 알아보고, 어떠한 이유로 해당 차이점을 가지게 되었는지 분석한 것을 기술하고자 한다.

Ⅱ. CRYSTALS-Kyber 알고리즘과 FIPS 203 문서 간 차이점 분석

1. KEM 알고리즘 - Shared Key의 길이 고정

CRYSTALS-Kyber에서는 Shared Key(공유 키)의 길이를 가변으로 설정할 수 있었으나, FIPS 203에서는 256비트로 고정한 것을 알 수 있는데, 이는 NIST 표준[5]에 명시된 다른 대칭 키 암호(TDES, AES)와 함께 사용하는 것을 상정하고 변경한 것임을 알 수 있었다.

2. KEM Encaps 알고리즘 - 해시함수를 사용하는 단계 삭제

Encaps 알고리즘이란 FIPS 203 알고리즘 중 하나로 캡슐화 키를 사용하여 공유 키와 관련된 암호문을 생성하는 알고리즘을 뜻한다. CRYSTALS-Kyber에서는 KEM.enc 알고리즘에 RNG(Random Number Generator)를 사용하는데, RNG는 무작위의 숫자를 생성하는 의사 난수 생성 알고리즘 중 하나로 일부 물리적인 소스(날짜, 시간 등)를 예측할 수 있다는 단점이 있었고 이를 보완하기 위해 해시함수를 사용했다[6]. 하지만 FIPS 203에서는 RNG 대신 RBG(Random Bit Generator)라고 하는 의사 난수 생성 알고리즘을 사용하여 불필요한 해시 함수 사용을 제거했다는 것을 알 수 있다.

Algorithm 8 KYBER.CCAKEM.Enc(pk)

Input: Public key $pk \in \mathcal{B}^{12\cdot k\cdot n/8+32}$
Output: Ciphertext $c \in \mathcal{B}^{d_u\cdot k\cdot n/8+d_v\cdot n/8}$
Output: Shared key $K \in \mathcal{B}^*$
1: $m \leftarrow \mathcal{B}^{32}$
2: $m \leftarrow \mathrm{H}(m)$
3: $(\bar{K}, r) := \mathrm{G}(m \| \mathrm{H}(pk))$
4: $c := \text{KYBER.CPAPKE.Enc}(pk, m, r)$
5: $K := \mathsf{KDF}(\bar{K} \| \mathrm{H}(c))$
6: **return** (c, K)

[그림 1] CRYSTALS-Kyber KEM.Encaps 알고리즘

Algorithm 16 ML-KEM.Encaps(ek)

Uses the encapsulation key to generate a shared
Validated input: encapsulation key ek $\in \mathbb{B}^{384k+}$
Output: shared key $K \in \mathbb{B}^{32}$.
Output: ciphertext $c \in \mathbb{B}^{32(d_u k+d_v)}$.
1: $m \xleftarrow{\$} \mathbb{B}^{32}$
2: $(K, r) \leftarrow G(m \| H(\mathsf{ek}))$
3: $c \leftarrow$ K-PKE.Encrypt(ek, m, r)
4: **return** (K, c)

[그림 2] FIPS203 KEM.Encaps 알고리즘

3. 다른 형식의 FO(Fujisaki-Okamoto) 변환 적용

CRYSTALS-Kyber에서는 [그림 1]과 같이 KDF(Key-Derivation Function)를 사용하여 FO 변환을 진행하였지만, FIPS 203에서는 [그림 2]와 같이 KDF를 사용하지 않는 방법으로 FO 변환을 진행하였다. KDF를 사용하지 않음으로써 AVX2 기준 최대 17%의 속도 향상을 확인할 수 있었으며 복호화가 실패할 확률 또한 감소한다는 것을 알 수 있었다.

4. 입력 유효성 검사 단계 포함

CRYSTALS-Kyber에서는 따로 유효성 검사 단계를 명시하지 않았지만, FIPS 203의 경우 암호화를 진행하기 위한 입력값의 조건이 무엇인지 구체적으로 명시하는 입력 유효성 검사 단계를 Encaps 알고리즘과 Decaps 알고리즘에 포함하였다.

Ⅲ. CRYSTALS-Dilithium 알고리즘과 FIPS 204 문서 간 차이점 분석

1. 매개변수 tr의 길이 증가

기존 Dilithium 3.1 버전에서는 매개변수 tr(공개키의 해시 값)의 길이가 기 256비트로 설정되어 있으나 FIPS 204에서는 512비트로 늘어난 것을 확인했다. CRYSTALS-Dilithium 알고리즘에서 사용하는 해시 함수가 충돌 저항성을 만족한다는 전제 하에 서명이 특정 검증키에만 독점적으로 속하는 성질(Malicious-Strong Universal Exclusive Ownership), 서명 후에 메세지를 변경하는 것을 방지하는 성질(Message-Bound Signatures), 재서명을 금지하는 성질(No Re-signing)을 가진다[7]. 따라서 충돌 저항성을 강화하기 위해 tr의 길이를 512비트로 변경하였음을 알 수 있다.

2. 매개변수 $\tilde{c}$의 길이 증가

CRYSTALS-Dilithium에서 언급한 NIST 보안 수준은 2/3/5수준으로 세 가지 버전이 있으며, 이는 FIPS 204에서 각각 ML-DSA-44/ML-DSA-65/ML-DSA-87로 대응된다. CRYSTALS-Dilithium에서는 FIPS 204에서는 $\tilde{c}$을 생성할 때, 256비트의 고정된 크기를 가졌지만, FIPS 204에서는 $\tilde{c}$를 생성할 때 2λ비트 크기로 생성되며, 이때 λ의 크기는 [그림 3]과 같이 보안 수준에 따라 128/192/256비트의 크기를 가지도록 변경되었다. 이는 해시 함수의 충돌 저항성을 강화하기 위한 변경으로, 보안 레벨이 올라감에 따라 더 높은 충돌 저항성을 가질 수 있도록 크기를 조정하였음을 알 수 있었다.

Table 1. ML-DSA Parameter sets

Parameters (see sections 5 and 6 of this document)	Values assigned by each parameter set		
	ML-DSA-44	ML-DSA-65	ML-DSA-87
q - modulus [see §5]	8380417	8380417	8380417
d - # of dropped bits from **t** [see §5]	13	13	13
τ - # of ± 1's in polynomial c [see §6]	39	49	60
λ - collision strength of $\tilde{c}$ [see §6]	128	192	256
γ_1 - coefficient range of **y** [see §6]	2^{17}	2^{19}	2^{19}
γ_2 - low-order rounding range [see §6]	$(q-1)/88$	$(q-1)/32$	$(q-1)/32$
(k,ℓ) - dimensions of **A** [see §5]	(4,4)	(6,5)	(8,7)
η - private key range [see §5]	2	4	2
$\beta = \tau \cdot \eta$ [see §6]	78	196	120
ω - max # of 1's in the hint **h** [see §6]	80	55	75
Challenge entropy $\log \binom{256}{\tau} + \tau$ [see §6]	192	225	257
Repetitions (see explanation below)	4.25	5.1	3.85
Claimed security strength	Category 2	Category 3	Category 5

[그림 3] FIPS 204 ML-DSA 매개변수 파라미터

3. 위협 회피 모드(hedged mode) 추가

CRYSTALS-Dilithium의 Sign 알고리즘은 무작위 시드 값 K, 매개 변수 tr, 메시지 M을 사용하여 매개변수 p'을 생성한다. 하지만 FIPS 204에서는 rnd라는 매개 변수를 추가하여 p'을 생성한다. FIPS 204에서는 rnd값에 따라 Sign 알고리즘이 두 가지 버전으로 나뉘는데 rnd의 값을 RBG를 사용해 생성하는 버전을 위협회피(Hedged) 버전이라고 하고 상수로 고정하는 버전을 결정론적(Deterministic) 버전이라고 한다. 위협회피(Hedged) 버전은 비교적 부채널 공격의 위험으로부터 안전하지만 결정론적(Deterministic) 버전은 난수 발생기를 사용할 수 없는 플랫폼에서 선택적으로 사용될 수 있지만 부채널 공격에 비교적 취약하다. 따라서 FIPS 204에서는 가능한 위협회피(Hedged) 버전의 사용을 권장한다.

Ⅳ. 결론

본 논문에서는 CRYSTALS 알고리즘과 FIPS 문서의 비교·분석 과정을 통하여 FIPS 203은 알고리즘 위주, FIPS 204는 매개변수 위주의 수정이 이루어진 것을 확인할 수 있었다. 2장과 3장에서 CRYSTALS 알고리즘과 FIPS 문서의 차이점과 변경 이유를 정리하였으며, 해당 사항들이 Lattice 기반 암호 알고리즘의 보안성 및 효율성을 향상시키고, 모호한 부분을 보다 명확하게 정리하였음을 알 수 있었다. 현재 양자내성암호의 중요성이 주목받고 있는 상황에서 이러한 비교 분석은 미래 양자컴퓨터에 의한 위협으로부터 대비하는 시작 단계가 될 것으로 보인다. 추후 연구과제로 FIPS 205와 해시 암호 기반의 전자서명 알고리즘 표준에 대해서도 비교 및 분석을 진행할 계획이다.

ACKNOWLEDGMENT

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. NRF-2022R1F1A1073211, 실제 양자컴퓨터 환경을 고려한 격자기반 양자내성암호 양자안전성 분석 연구)과 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2023-00225201, 국방 무인이동체 역이용 방지 제어권 보호기술 개발)

참 고 문 헌

[1] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.02)", 2021

[2] NIST FIPS 203 (Draft), "Module-Lattice-based Key-Encapsulation Mechanism Standard", 2023

[3] S. Bai, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler and D. Stehlé, "CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation (Version 3.1)", 2021

[4] NIST FIPS 204 (Draft), "Module-Lattice-Based Digital Signature Standard", 2023

[5] E. Barker and A. Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths", NIST SP 800-131A Revision 2, 2019

[6] A. Rukhin and J. Soto, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", NIST SP 800-22 Revision 1, 2010

[7] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin and C. Janson, "BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures", IEEE Symposium on Security and Privacy, 2021