정 보 보 호 학 회 지
제 33 권 제2호, 2023. 4

NIST PQC 공모전 동향 분석 및 표준화 대상 & Round 4 알고리즘 소개

김 동 천*, 김 영 범*, 서 석 충**

요 약

소인수 분해 및 이산대수 문제의 난제를 기반으로 설계된 기존의 공개키 암호 및 전자서명 체계가 1994년 제안된 Shor 알고리즘으로부터 안전성에 대한 위협을 받게 되자, NIST에서는 양자컴퓨팅 환경으로부터 보안성이 유지되는 암호를 선정하기 위해 양자내성암호 공모전을 개최하였다. 총 3 Round를 통해 PQC 표준화 대상 알고리즘을 채택하였으며, 추가로 채택된 양자내성암호의 기반에 대한 다양성을 두기 위해 Round 4를 진행하였다. 따라서 본 논문에서는 표준화 대상으로 선정된 알고리즘(Selected Algorithms 2022)과 현재 Round 4를 진행하고 있는 알고리즘의 기반이 되는 배경지식과 구조를 설명하고, 알고리즘별 주요 사양을 통해 각각의 장단점을 살펴볼 것이다. 나아가 현재 양자내성연구단을 통해 우리나라에서 제안된 KpqC에 대해서도 간단히 소개할 것이다.

Ⅰ. 서 론

기존의 공개키 암호 및 전자서명 체계는 소인수 분해 문제와 이산대수 문제의 수학적 난제를 기반으로 설계되었다.

하지만 1994년 양자 회로상에서의 소인수 분해 알고리즘인 Shor 알고리즘이 발표되었고, 지속적인 양자 회로 및 양자 컴퓨터의 발전에 따라 Shor 알고리즘을 통해 기존 암호 체계가 다항 시간 내에 분석될 수 있음이 알려졌다. 이에 NIST(National Institute of Standards and Technology)는 양자 컴퓨팅 환경에서의 보안 안전성을 도모하기 위해 PQC(Post-Quantum Cryptography) 공모전을 개최하였다.

NIST PQC 공모전은 2016년을 시작으로 다음 해 12월, 공모전을 통과한 69개의 알고리즘 중 철회된 5개를 제외한 64개의 Round 1 알고리즘을 후보로 등록하였고, 이후 2019년 6월, Round 2에서는 Round 1을 통과한 알고리즘 중 NIST PQC 표준화 대상 평가 기준(Security, cost and performance, algorithm and implementation characteristics)에 부합한 26개만을 발표하였다[1].

2020년 7월 Round 3에서는 앞서 언급한 26개의 알고리즘 중 7개의 Finalists와 알고리즘의 안전성에 문제가 생긴 경우 이를 대체할 수 있는 8개의 Alternates 알고리즘만이 후보로 등록되었다.

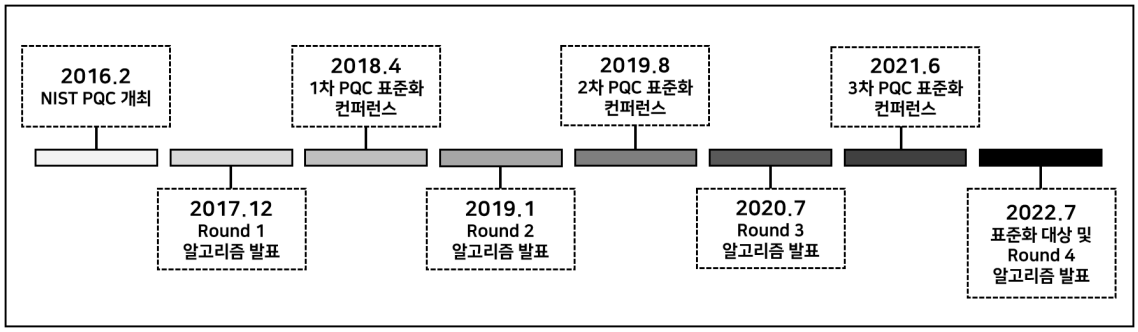
이후 2022년 7월 마침내 양자 내성 암호 PKE/KEM 표준으로 CRYSTALS-KYBER, 전자서명의 표준으로 CRYSTALS-DILITHIUM, FALCON, SPHINCS+가 선정되었다.

나아가 하나의 알고리즘만 채택된 PKE/KEM 표준을 보완하기 위해 다양한 수학적 난제를 기반으로 한 암호 알고리즘을 Round 4를 통해 공모하였고, 이를 통해 코드 기반의 Classic McEliece, BIKE, HQC와 Isogeny 기반의 SIKE가 표준화 대상 알고리즘의 후보로 등록되었다[2].

하지만 이 중 SIKE의 경우 Isogeny Diffie-Hellman protocol(SIDH)을 효율적으로 공격할 수 있는 key recovery attack이 발견되어 안전성에 대한 취약점이 드러났다[3].

본 논문에서는 위에서 언급한 NIST PQC 공모전 중 Round 3에 대한 주요 결과와 이를 통해 선정된 알고리즘(Selected Algorithms 2022) 및 Round 4에서 후보로

이 성과는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2022-00207416, 안전한 차세대 IoT 통신 환경 구축을 위한 양자내성암호 최적화 및 보안 프로토콜 적용 연구, 100%)
* 국민대학교 금융정보보안학과 (대학원생, kindongsy@kookmin.ac.kr/darania@kookmin.ac.kr)
** 국민대학교 금융정보보안학과 (부교수, scseo@kookmin.ac.kr)



〔그림 1〕 NIST PQC 공모전 연표

등록된 알고리즘에 대해 분석하여 소개할 것이다.

Ⅱ. 표준화 대상 알고리즘 및 Round 4 진출 알고리즘

NIST PQC Round 3에서는 Public-key Encryption (PKE) and Key Encapsulation Mechanism(KEM)과 Digital Signature Algorithms(DSA), 두 관점에서 Finalists 7개[표 1]와 이를 대체할 수 있는 Alternate Candidates 8개[표 2]를 발표하였다.

〔표 1〕 Round 3 Finalists

Round 3 Finalists		
분야	PKE/KEMs	DSA
알고리즘	Classic McEliece, CRYSTALS-KYBER, NTRU, Saber	CRYSTALS-Dilithium, FALCON, Rainbow

〔표 2〕 Round 3 Alternate Candidates

Round 3 Alternate Candidates		
분야	PKE/KEMs	DSA
알고리즘	BIKE, FrodoKEM, HQC, NTRU Prime, SIKE	GeMSS, Picnic, SPHINCS+

2.1. 표준화 대상 알고리즘

이 중 GeMSS는 HFEv- 서명 체계를 기반으로 하는데, HFEv-에 대한 새로운 key recovery attack이 제안되어 주장하는 것만큼 안전하지 않다는 것이 증명되었고[4], Rainbow 역시 개인키에 대한 취약점이 발견되어 표준화 대상에서 제외하였다[5].

FrodoKEM의 경우 다른 KEM(BIKE, HQC, SIKE)보다 일반적으로 성능이 좋지 않은 것이 드러났고, NTRU Prime 역시 다른 알고리즘에 비해 장래성이 부족하다고 판단되어 Round 3에서 그치고 말았다.

한편 KYBER, NTRU, SABER의 경우 모두 격자 기반 알고리즘으로 유사한 KEM 구조로 되어 있는데, 이때 NIST는 MLWE 기반의 알고리즘인 KYBER가 차세대 암호로 더 설득력 있다고 판단했고, 벤치마크에서도 높은 성적을 거두었기에 KYBER를 최종 표준화 알고리즘으로 선택했다.

DSA 관점에서는 Picnic과 SPHINCS+가 작은 공개키 및 큰 서명값을 가지는 부분에서 유사한데, NIST는 SPHINCS+가 보안성이 높고 더 발전되었다고 판단하여 Picnic 대신 SPHINCS+를 선택했다.

더불어 SPHINCS+의 경우 필요한 연산에 있어 격자 기반 서명 알고리즘보다 서명의 크기가 훨씬 크고 느리지만 PQC의 안전성을 격자 기반으로 단순화하지 않기 위해 Round 4를 진행하는 대신 표준화 대상으로 바로 선정하였다.

이런 과정을 거쳐 Round 3 이후 표준화 대상 PQC로 선택된 알고리즘은 다음[표 3]과 같다.

DSA 분야에서 표준화 대상 알고리즘으로 선정된 FALCON과 CRYSTALS-Dilithium은 공통적으로 격자 기반 알고리즘이므로 NIST는 한 가지만 채택하려 했다. 하지만 FALCON의 경우 키 및 서명 생성 시 Dilithium보다 더 많은 자원(논리 회로 및 RAM)을 필요로 하므로 자원이 제한된 환경에서 부채널 공격에 대한 안전성을 갖추기에 적합하지 않았고, Dilithium만을 일률적으로 사용하는 경우 서명 데이터를 단일

〔표 3〕 Selected Algorithms 2022

Selected Algorithms 2022		
Base	Public-Key Encryption / KEMs	Digital Signatures
Lattice	CRYSTALS-KYBER	CRYSTALS-Dilithium, FALCON
Hash	·	SPHINCS+

인터넷 패킷에 담기 위해 응용 프로그램을 수정해야 한다는 어려움이 따랐기 때문에 NIST에서는 위의 상호보완적 관계를 고려하여 두 가지 알고리즘을 모두 채택했다.

2.2. Round 4 진출 알고리즘

나아가 NIST는 KEM 분야에서 표준화 대상 암호 알고리즘의 기반 난제에 대한 다양성을 추구하기 위해 [표 4]와 같이 BIKE, HQC, Classic McEliece, SIKE를 대상으로 Round 4를 진행하였다.

BIKE, HQC는 격자 기반이 아닌 Codes 기반 알고리즘으로 범용 KEM으로 사용하기 적합하다. Classic McEliece 역시 finalists까지 진출한 Codes 기반 알고리즘이지만, 공개키의 크기가 크기 때문에 NIST에서 당장 널리 사용될 알고리즘으로는 고려되고 있지 않다.

SIKE의 경우 위에서 언급했듯이 Isogeny 기반 알고리즘에 대한 효율적인 키 복구 공격이 발견되어 취약점이 인정되었지만, NIST가 해당 정보를 보관함으로써 다른 연구자에게 도움이 될 것으로 판단하여 철회되지 않고 현재까지도 NIST 공식 홈페이지 Round 4에서 확인할 수 있다.

〔표 4〕 Round 4 Algorithms

Round 4 Algorithms	
Base	Public-Key Encryption / KEMs
Codes	BIKE, HQC, Classic McEliece
Isogeny	SIKE

Ⅲ. 표준화 대상 알고리즘의 기반 난제

여기서는 2단원에서 소개한 Selected Algorithms 2022와 Round 4 알고리즘의 배경이 되는 기반 지식을 소개하고자 한다.

3.1. 격자 기반

격자 기반 암호는 1996년 Ajtai에 의해 제안된 수학적으로 정의된 격자 상에서의 난제를 기반으로 하는 암호이다. 대표적인 난제로 고차원 격자 상에서 0이 아닌 최단 백터를 찾는 SVP(Shortest Vector Problem)와 임의의 점에 가장 가까운 벡터를 찾는 CVP(Closest Vector Problem)가 있다.

최종 선정된 알고리즘 중 CRYSTALS-KYBER, CRYSTALS-Dilithium의 경우 위 난제를 변형한 M-LWE(Module-Learning With Errors)에 안전성을 두고 있고, FALCON은 NTRU 격자 상에 SVP를 적용하여 알고리즘을 고안했다[6][7][8].

3.2. 해시 기반

해시 기반 서명 기법은 1970년대 후반 Lamport에 의해 제2 역상 저항성을 기반으로 단일 공개키/서명키를 통해 일회성 서명/검증을 제공하도록 개발되었다. 하지만 해시 기반 서명의 경우 전통적으로 서명 속도와 크기 면에서 문제가 되었고[9], 이후 Merkle을 통해 제안된 Merkle tree와 이를 기반으로 한 XMSS(eXtended Merkle Signature Scheme)를 통해 서명/검증에 대한 효율성이 향상되었다[10].

PQC 표준화 대상 알고리즘으로 선정된 SPHINCS+ 역시 연산 구조로 해시 기반 전자서명 알고리즘을 사용하였으며, Merkel Tree 및 XMSS 기반으로 설계되었다[11].

3.3. 코드 기반

코드 기반 암호는 1978년 Robert J. McEliece가 통신상에서 사용되는 코딩 이론을 활용하여 새로운 공개키 시스템인 McEliece를 제안하면서 등장하였다.

송신자는 오류 수정이 가능한 생성 행렬 기반의 공개키와 메시지를 조합하여 인코딩한 후 해당 데이터에 의도적으로 오류를 추가하여 수신자에게 전송한다.

전송된 데이터에 대한 올바른 수신자는 오류 수정 코드인 패리티 체크 행렬을 통해 첨부된 오류를 쉽게 제거하고 원본 메시지를 확인할 수 있다[12].

NIST PQC Round 4에 진출한 Claasic McEliece는 Goppa라는 오류 수정 코드를 사용한 코드 기반 PKE 알고리즘이다[13].

Ⅳ. 표준화 대상 알고리즘 소개

본 단원에서는 앞서 설명한 배경지식을 기반으로 NIST PQC 표준화 대상 알고리즘(Selected Algorithms 2022) 각각을 소개하고, 특징 및 장단점을 분석하고자 한다.

다음 [표 5]는 'OPEN QUANTUM SAFE'를 통해 Selected Algorithms의 키 및 암호문/서명의 크기 정보를 정리한 것이며, [표 6], [표 7]은 각각 Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz 환경에서 PKE/KEMs, DSA 알고리즘의 Cycle을 통해 연산 시간을 나타낸 것이다. 이때 SPHINCS+는 해시 함수 및 알고리즘의 종류에 따라 서명 크기가 다르므로 하나의 알고리즘(SPHINCS+ SHA256-128/192/256f simple)을 선택하여 대표적인 수치를 나타내었다.

〔표 5〕 Selected Algorithms Spec

키 및 암호문/서명 크기(Bytes)					
구분	보안 수준	KYBER	Dilithium	FALCON	SPHINCS+
공개키 크기	1	800	1,312	897	32
	3	1,184	1,952	·	48
	5	1,568	2,592	1,793	64
개인키 크기	1	1,632	2,528	1,281	64
	3	2,400	4,000	·	96
	5	3,168	4,864	2,305	128
암호문/서명 크기	1	768	2,420	690	17,088
	3	1,088	3,293	·	35,664
	5	1,568	4,595	1,330	49,856

〔표 6〕 Selected Algorithms PKE/KEMs 연산 시간

PKE/KEMs 연산 시간(x86_64-ref, Cycles)					
구분	Algorithm	보안 수준	KeyGen	EnCap	DeCap
PKE/KEMs	KYBER	1	111,348	133,673	154,819
		3	185,553	220,579	250,792
		5	287,000	315,066	353,167

〔표 7〕 Selected Algorithms DSA 연산 시간

DSA 연산 시간(x86_64-ref, Cycles)					
구분	Algorithm	보안 수준	KeyPair	Sign	Verify
DSA	Dilithium	1	267,188	1,216,321	286,937
		3	473,932	1,909,159	455,024
		5	704,467	2,300,371	731,223
	FALCON	1	46,206,441	14,141,048	142,588
		3	·	·	·
		5	142,207,860	31,098,436	285,087
	SPHINCS+	1	4,392,566	108,723,181	6,124,417
		3	6,397,572	177,995,594	9,086,434
		5	17,200,129	363,025,642	9,093,812

4.1. CRYSTALS-KYBER

Public-Key Encryption / KEMs 알고리즘인 CRYSTALS-KYBER는 암호화 과정에서 에러가 추가된 선형방정식으로 이루어진 오라클로부터 얻은 결괏값을 통해 다항식을 다항 시간(PPT, Probabilistic Polynomial Time) 내에 해결할 수 없다는 원리를 이용한 LWE 난제를 다항식 환(Ring)을 통해 연산 효율을 높인 Module-LWE를 기반으로 하고 있다[14].

Module-LWE를 기반으로 R_q 상의 다항식을 원소로 가지는 공개행렬 A, 비밀 벡터 s, 에러 e를 이용하여 $A \cdot s + e$ 구조의 다항식 곱셈 및 덧셈을 통해 구현된다[15].

이때 가장 연산 시간이 긴 다항식 곱셈의 연산 시간을 줄이기 위해 NTT를 활용하며, 정방행렬 A의 크기인 k값을 통해 보안 수준을 결정한다[16].

CRYSTALS-KYBER 알고리즘은 다항식의 차수인 n과 다항식 계수가 정의되는 소수 q를 고정으로 사용하며 k값만 변경하여 보안 수준을 변경할 수 있어서 확장성이 유연하다는 장점이 있다[17].

4.2. CRYSTALS-Dilithium

Digital Signature 분야 표준화 대상 알고리즘인 CRYSTALS-Dilithium은 MLWE(Module Learning

With Errors)와 주어진 샘플들에 대해 특정 Constraint를 만족하는 짧은 정수해를 찾는 문제인 MSIS(Module Short Integer Solution)의 어려움을 바탕으로 설계되었다[18][19].

CRYSTALS-Dilithium은 크게 키 생성 과정, 서명 과정, 검증과정으로 이루어져 있다.

키 생성 과정은 R_q 상에서 다항식에 대한 공개행렬 A를 생성하고, 무작위의 비밀 키 벡터인 s_1와 s_2를 Sampling 함수를 통해 추출하여 $t = As_1 + s_2$ 값을 계산한다.

서명 과정은 다항식 y의 마스킹 벡터를 생성하는 과정으로, 서명자는 Ay를 계산하고 메시지와 Ay를 해싱하여 c를 생성한다. 이후 c와 s_1를 통해 생성되는 $z = y + cs_1$가 서명 값이 된다.

검증과정에서는 공개키(A,t)와 서명값(c,z)을 통해 $Az - ct$를 계산하여 z와 c가 서명 과정에서 생성된 값과 일치하는지 확인한다. 이후 서명자가 생성한 $Ay - cs_2$와 검증자가 계산하는 $Az - ct$를 비교하여 일치한 경우 검증을 완료한다[20].

CRYSTALS-Dilithium 역시 핵심 연산은 다항 환 상에서 행렬 A에 대한 곱셈 과정이므로 CRYSTALS-KYBER와 같이 연산의 부하를 감소시키기 위해 NTT를 사용하였다.

4.3. FALCON

FALCON은 NTRU 격자 상에서 SIS 문제의 어려움에 안전성을 둔 전자서명 알고리즘으로, 서명 및 검증 속도가 빠르고 서명 길이가 상대적으로 짧다는 장점이 있다[21].

CRYSTASL-KYBER, Dilithium과는 달리 NTRU 격자 기반으로, 보안 수준을 높이긴 위해선 다항식의 차수를 증가시킨다.

FALCON 또한 전자서명이므로 크게 세 과정으로 구분된다.

키 생성 과정은 다항식 f,g가 주어졌을 때 NTRU 방정식 $fG - gF = q \bmod \phi$를 만족하는 두 다항식 F,G를 찾는 과정으로, 개인키는 f,g,F,G이며 공개키는 $h = gf^{-1} \bmod q$가 된다.

서명 과정은 서명 (r,s)를 생성하는 과정으로, 메시지 m과 랜덤값인 r에 대한 해시값 c를 계산한다. 이후 개인키를 사용하여 $s_1 + s_2 h = c \bmod q$를 만족하는 s_1, s_2를 생성한다. 이때 가우시안 분포에 따라 랜덤한 라운딩을 수행하는 ffsampling 함수가 사용되며 이는 서명 과정에서 가장 큰 부하를 차지한다.

서명 검증 과정에서는 전달받은 서명 값 (r,s)를 서명 생성의 역과정을 통해 s_1, s_2를 복구하고, 이 값이 올바른 범위 내에 있는지 검증하는 것이다. 서명과 동일하게 m과 r을 통해 c를 생성하고, Decompress 함수와 $s_1 = c - s_2 h \bmod q$를 통해 s_1, s_2를 생성한다[22].

FALCON 역시 다항식 환 상에서의 곱셈 연산이 주요 연산이기 때문에 FFT와 NTT를 사용하여 연산 효율을 향상했다.

추가적으로 FALCON의 경우 NIST 보안 수준 중 3이 제외된 Level 1, 5에 해당하는 FALCON-512, FALCON-1024만이 제안되었다.

4.4. SPHINCS+

SPHINCS+는 OTP, WOTS+(Winternitz One Time Signature+), FORS(Forest Of Random Subset), XMSS 등 기존 해시 기반 전자서명의 전반적인 방식을 통합한 형태로 구성되어 있고, Haraka, SHA-2, SHAKE 등의 다양한 해시 함수를 사용한다[23].

반면 SPHINCS+는 기존의 해시 기반 전자서명과는 달리 상태 비저장 구조 형태를 가지기 때문에 분산 서버 및 백업 운용에도 적용 가능한 장점이 있다.

SPHINCS+는 여러 개의 트리가 결합한 하이퍼 트리 구조를 사용하여 많은 수의 Few Time Signature(FTS)의 키 쌍 인증을 수행하는 구조이다.

각 세부 계층은 XMSS 또는 Merkle Hash Tree로 구성된 서브 트리 형태이며, 각 서브 트리의 서명은 WOTS+ 방안을 사용한다. 더불어 서명 메시지 원본에 대한 1차 서명은 FORS를 이용하고, 이후 해당 서명값이 SPHINCS+ 전체 구조의 입력값으로 구성된다.

서명 생성 과정에서 입력 메시지 m은 비트 단위로 분할된 $m = (m_0, m_1, \ldots\ldots, m_{\leq n_1 - 1})$으로 구성되고, $C = \sum_{i=1}^{\leq n_1} (w - 1 - m_i)$를 통해 checksum이 계산된다. 서명 생성 과정에서 w를 통해 해당 checksum을 $C = (C_0, C_2, \ldots\ldots, C_{\leq n_2 - 1})$가 만족하도록 재분할한

후, $msg = m \mid\mid C = (msg_0, ..., msg_{\leq n-1})$를 구성하기 위해 C와 m을 병합한다.

이후 각각의 msg_i값에 대한 서명 값은 $sig = (Hash_{msg_0}(sk_0), Hash_{msg_1}(sk_1), \ldots, Hash_{msg_{\leq n-1}}(sk_{\leq n-1}))$를 통해 생성된다.

서명 검증 과정의 메시지에 대한 checksum 생성 및 msg생성 과정은 동일하게 진행된다. 이후 전달받은 $verify = (Hash_{w-1-msg_i}(sig_i)), sig_i = Hash_{msg_i}(sk_i)$ $with\,(0 \leq i \leq n-1)$과정을 통해 서명 검증을 진행한다[24].

SPHINCS+의 경우 [표 5]와 같이 공개키 및 개인키의 크기가 작은 장점을 가진 반면에, 서명 크기가 확연히 큰 단점을 가지고 있다.

Ⅴ. Round 4 진출 알고리즘 소개

본 단원에서는 NIST PQC Round 4에 진출한 각각의 알고리즘을 소개하고, 특징 및 장단점을 분석하고자 한다. 다만 SIKE의 경우 위에서 언급했듯이 취약점이 발견되었으므로, 본 논문에서는 분석 대상에서 제외하였다.

다음 [표 8]는 'OPEN QUANTUM SAFE'를 통해 Round 4에 진출한 알고리즘의 키 및 암호문/서명 크기 정보를 정리한 것이며, [표 9]는 Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz 환경에서 Cycle을 통해 연산 시간을 나타낸 것이다. 이때 Classic

〔표 8〕 Round 4 Algorithms Spec

Round 4 Algorithms Spec				
	보안 수준	BIKE	HQC	Classic McEliece
공개키 크기 (Bytes)	1	1,540	2,249	261,120
	3	3,082	4,522	524,160
	5	5,122	7,245	1,044,992
개인키 크기 (Bytes)	1	280	40	6452
	3	418	40	13,568
	5	580	40	13,892
암호문 /서명 크기 (Bytes)	1	1,572	4,481	128
	3	3,114	9,026	188
	5	5,154	14,469	240

〔표 9〕 Round 4 연산 시간

PKE/KEMs 연산 시간(x86_64-ref, Cycles)					
구분	Algorithm	보안 수준	KeyGen	EnCap	DeCap
PKE/ KEMs	BIKE	1	10,885,172	784,889	13,270,177
		3	34,223,772	2,283,365	39,676,145
		5	86,995,286	5,707,899	98,626,475
	HQC	1	302,077	609,249	956,009
		3	743,204	1,568,088	2,339,842
		5	1,421,144	3,046,888	4,563,838
	Classic McEliece	1	344,505,761	512,297	543,031
		3	1,191,088,068	1,085,856	1,215,800
		5	1,414,943,246	2,077,481	1,355,641

McEliece의 경우 연산 시 사용되는 파라미터의 종류에 따라 키 및 암호문의 크기가 다르므로 하나의 알고리즘(Classic-McEliece-348864/460896/6688128)을 선택해 대표적인 수치만을 나타내었다.

5.1. BIKE

BIKE는 코드 기반 KEM 알고리즘으로, 디코딩 방식으로 비트플립 기술을 사용하는 QC-MDPC (Quasi-Cyclic Moderate Density Parity-Check)와 패리티 체크 행렬을 갖춘 Niederreiter 암호를 기반으로 하고 있다[25].

기존 코드 기반 알고리즘인 McEliece와 비교하면 작은 키 크기를 통해서도 합리적인 보안 강도를 가질 수 있다는 장점이 있다.

다항식 환을 통해 비밀키 h_0, h_1과, 메시지 공간 M에서 σ를 각각 추출하고, 이를 통해 $h = h_1 h_0^{-1}$을 계산해 공개키를 생성한다.

이후 Encapsulation 과정에서는 먼저 메시지 공간에서 추출한 m에 해시 함수 H를 적용하여 e_0, e_1을 계산한다. 그리고 e_0, e_1, m,공개키 h와 해시 함수 L을 통해 $c = (e_0 + e_1 h, m \oplus L(e_0, e_1))$를 계산한 뒤, 한 번 더 다른 해시 함수인 $K(m, c)$를 이용하여 공유 키인 K를 계산한다.

Decapsulation 과정에서는 Black-Gray-Flip decoder를 기반으로 비밀키인 h_0, h_1, σ 그리고 c값을 이용해

다시 K값을 도출해낸다[26].

5.2. HQC

HQC는 Random Quasi-Cyclic Code의 decoding이 어렵다는 점에 착안하여 설계되었다.

HQC의 연산은 Binary Field 상에서 이루어지므로 Binary Field 상의 곱셈에 대한 최적화를 위해 현재 3-Way Tom-Cook과 3-Way Karatsuba 알고리즘을 사용하고 있다.

HQC PKE는 Setup, KeyGen, Encrypt, Decrypt로 구성된다.

먼저 Setup에서는 파라미터 (n,k,σ,w,w_r,w_e)를 생성한다. 이후 KeyGen 에서는 $R=F_2[X]/(X^n-1)$을 기반으로 h와 생성 행렬 $G\in F_2^{k\times n}$와, $w(x)=w(y)=w$를 만족하는 비밀키인 $sk=(x,y)$를 추출한 뒤 $pk=(h,s=x+hy)$를 계산하여 고정된 크기의 sk와 pk를 생성한다.

Encrypt 단계에선 $w(e)=w_e$를 R에서 추출하고, $w(r_1)=w(r_2)=w_r$를 만족하는 e와 $r=(r_1,r_2)$을 R^2에서 추출한 뒤, $u=r_1+hr_2$와$v=mG+sr_2+e$를 계산하여 $c=(u,v)$를 생성한다. 마지막으로 Decrypt 단계에서는 $Decode(v-uy)$연산을 진행함으로써 원문을 확인할 수 있다[27]. 이때 HQC의 Decode 연산은 Berlekamp-Massey 알고리즘을 통해 효율적으로 구현할 수 있다.

5.3. Classic McEliece

Classic McEliece는 1978년 McEliece에 의해 제안된 선형 오류 수정 코드의 일종인 이진 Goppa 코드를 기반으로 설계되었다.

공개키는 패리티 검사 행렬을 기반으로 생성하며, 개인키는 이진 Goppa 코드를 기반으로 한다.

키생성 과정에서는 먼저 $n-bit$ string s를 생성한 뒤 Random Monic Irreducible Polynomial $g(x)\in F_q[x]$ of degree t 및 Random Sequence $(\alpha_1,\alpha_2,\cdots,\alpha_n)$ of n distinct elements of F_q를 생성한다. 이후 Goppa code $\Gamma=(g,\alpha_1,\alpha_2,\cdots,\alpha_n)$을 정의하여 $(T,c_{n-k-\mu+1},\cdots,c_{n-k},\Gamma')=MatGen(\Gamma)$을 계산한다. 이때 T가 공개키가 되고, (Γ',s)가 비밀키가 된다[28].

이를 통해 송신자는 Encapsulation 함수에서 오류 벡터와 공개키로 암호문을 생성 및 해당 정보들로 세션키를 계산하고, 수신자는 Decapsulation 함수를 통해 암호문을 개인키로 복호화하고 오류 벡터를 복구하며, 송신자와 같이 동일한 세션키를 계산한다. 이때 Encapsulation과 Decapsulation 함수는 동일한 해시 함수를 사용하며, Decode 과정에서는 Berlekamp-Massey 알고리즘을 사용한다[29].

다만 Classic McEliece는 현재 높은 안전성을 만족하기 위해 큰 공개키를 사용한다는 단점이 있는데, 이런 공개키를 생성하는 부분에서 가장 높은 부하가 요구된다.

Ⅵ. 결 론

본 논문에서는 NIST PQC 공모전의 진행 상황을 설명하고, PQC 표준화 대상 알고리즘(Selected Algorithms) 및 Round 4에 진출한 알고리즘들을 소개하였다. 나아가 NIST는 2024년까지 공모전을 통해 채택된 알고리즘에 대해 상용화 가능한 표준화 초안을 발표할 계획이다.

한편 우리나라에서도 양자내성암호연구단을 통해 KpqC 공모전이 진행되고 있다.

2023년 12월 공모전 1라운드 결과 발표를 바탕으로 2024년 3월에 2라운드 알고리즘 발표를 준비하고 있

〔표 10〕 KpqC Competition Round 1

Base	PKE/KEMs	DSA
Codes	Layered ROLLO-I, PALOMA, REDOG	Enhanced pqsingRM
Lattice	SMAUG, NTRU+, TiGER	GCKSign, HAETAE, NCC-Sign, Peregrine, SOLMAE
Multivariate -Quadratic Equation	·	MQ-Sign
Algebraic S-boxes	·	AIMer
Isogeny	·	FIBS
Graph	IPCC	·

으며, KpqC Competition Round 1에서 발표된 알고리즘은 다음[표 10]과 같다.

PQC 알고리즘들은 현재의 공개키 및 전자서명보다 많은 연산 부하와 큰 크기를 가지고 있기 때문에, PQC를 상용화하기 위해서는 프로토콜 및 임베디드/서버 장치에서 최적화가 이루어져야 할 것이다.

따라서 향후 연구에서는 현재 사용 중인 암호 및 전자서명에 대한 알고리즘을 GPU 기반 서버 관점에서 최적화에 관한 연구를 진행할 것이다.

참 고 문 헌

[1] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, “Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process”, NIST, February 2020.

[2] 이태호, 조영진, 박준우, ”NIST PQC 공모전 동향 연구“, 한국통신학회, 한국통신학회 학술대회논문집, pp. 575-577, November 2022

[3] Wouter Castryck, Thomas Decru, “An efficient key recovery attack on SIDH(Preliminary version)”, Cryptology ePrint Archive, July 2022

[4] Chengdong Tao, Albrecht Petzoldt, Jintai Ding, “Improved Key Recovery of the HFEv- Signature Scheme”, Cryptology ePrint Archive, November 2020

[5] Ward Beullens, IBM Research – Zurich, “Breaking Rainbow Takes a Weekend on a Laptop”, Advances in Cryptology-CRYPTO 2022, August 2022

[6] 이정환, 김규상, 김희석, “NIST PQC Round 3 격자 기반 암호 KEM에 대한 부채널 분석 기법 동향 분석”, 한국정보보호학회, 정보보호학회지, pp. 47-56, February 2022

[7] 박찬희, 윤영여, 박해룡, 최은영, 김호원, “격자기반 양자내성 키 교환 알고리즘 구현”, 한국정보보호학회, 정보보호학회지, pp. 11-16, June 2020

[8] Daniele Miccianocio, Shafi Goldwasser, “Shortest Vector Problem”, Complexity of Lattice Problems: A Cryptographic Perspective, pp. 69-90, 2002

[9] 박태환, 배봉진, 김호원, “Stateless 해시 기반 서명 기법 동향 및 전망”, 한국정보처리학회, 학술대회 논문집, pp. 268-270, January 2016

[10] Daniel J. Bernstein, Andreas Hulsing, Stefan Kolbl, Ruben Niederhagen, Joost Rijneveld, Peter Schwabe, “The SPHINCS+Signature Framework”, CCS ‘19, November 2019

[11] 최호진, “GPU 환경에서 해시 기반 PQC 알고리즘 SPHINCS+의 효율적인 구현 방안”, 국민대학교 일반대학원 학위 논문, February 2022

[12] 송경주, 강예준, 장경배, 서화정, “코드기반암호에 대한 ISD 공격 알고리즘 연구 동향”, 한국정보처리학회, 춘계학술발표대회 논문집 제28권 제1호, 167-170, May 2021

[13] 장경배, 심민주, 서화정, “코드기반암호를 활용한 IoT 환경 보안 프로토콜 설계”, 한국정보처리학회, 춘계학술발표대회 논문집 제27권 제1호, May 2020

[14] 유준수, 윤지원, “LWE와 완전동형암호에 대한 분석 및 동향”, 한국정보보호학회, 정보보호학회지, 111-119, October 2020

[15] Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M.Schanck, Pter Schwabe, Gregor Seiler, Demien Stehle, “CRYSTALS-KYBER(Algorithm Specifications And Supporting Documentations (version 3.0)), NIST PQC Round, October 1 2020

[16] 김성재, “양자내성암호를 위한 고성능 Crystals-Kyber 암호 아키텍처”, 인하대학교 공학 대학원 석사학위논문, February 2022

[17] 김영범, “8-bit AVR 마이크로 컨트롤러에서의 양자내성암호 Crystals-Kyber 최적화 구현 연구”, 국민대학교 일반대학원 석사학위논문, February 2022

[18] 김일주, “격자 기반 전자서명 qTESLA, Dilithium의 부채널 분석에 관한 연구”, 국민대학교 일반대학원 논문, February 2020

[19] 구자현, 이선용, 노종선, “서로 다른 소수를 모듈러스로 갖는 짧은 정수해 문제들 사이의 관계”, 한국

통신학회, 학술대회논문집, February 2020

[20] Shi Bai, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler and Damien Stehle, “CRYSTALS-Dilithium”, Algorithm Specifications and Supporting Documentation, October 1 2020

[21] 김규상, 박동준, 홍석희, “NIST PQC Round 3 FALCON 전자서명 알고리즘의 전력 분석 취약점 연구”, 한국정보보호학회, 정보보호학회지, pp. 57-64, February 2021

[22] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, “FALCON : Fast-Fourier Lattice-based Compact vSignatures over NTRU”, Submission to the NIST’s post-quantum cryptography standardization process, October 1 2020

[23] 강태구, “해시기반 포스트 양자 서명 기법 분석”, 명지대학교 대학원 석사학위논문, February 2020

[24] Jean-Philippe Aumasson, Daniel J. Bernstein, Ward Beullens, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan-Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe, Bas Westerbaan, “SPHICS+ Submission to the NIST post-quantum project, v.3”, NIST Submission, October 2020

[25] Alexandr Kuznetsov, Maria Lutsenko, Nastya Kiian, Tymur Makushenko, Tetiana Kuznetsova, “Code-based key encapsulation mechanisms for post-quantum standardization”, IEEE DESSERT, May 2018

[26] Nicolas Aragon, Paulo S. L. M. Barreto, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Guneysu, Carlos Aguilar Melchor, Rafael Misoczke, Edoardo Persichetti, Jan Richter-Brockmann, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, Gilles Zemor, “BIKE:Bit Flipping Key Encapsulation(Round 4 Submission)”, NIST Submission, October 2022

[27] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, Gilles Zémor, “Hamming Quasi-Cyclic(HQC) (Fourth round version)”, NIST Submission, October 2022

[28] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan, Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G., Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen, Jung Tjhai, Martin Tomlinson, Wen Wang, “Classic McEliece:conservative code-based cryptography:cryptosystem specification”, NIST Submission, October 2022

[29] 최장혁, 박민진, 김동찬, “Classic McEliece 규격 및 파라미터별 성능에 관한 연구”, 한국통신학회, 학술대회논문집, pp. 1513-1514 June 2021

〈저자소개〉

김 동 천 (Dong Cheon Kim)

2023년 2월 : 국민대학교 정보보안암호수학과 졸업

2023년 3월~현재 : 국민대학교 금융정보보안학과 암호 최적화 및 응용 연구실 석사과정

<관심분야> 현대대수학, 암호최적화, 양자내성암호, 정보보호, 네트워크

김 영 범 (Young Beom Kim)

학생회원

2021년 2월 : 국민대학교 정보보안암호수학과 졸업

2023년 2월 : 국민대학교 금융정보보안학과 석사

2023년 3월~현재 : 국민대학교 금융정보보안학과 암호 최적화 및 응용 연구실 박사과정

<관심분야> 암호최적화, 양자내성암호

서 석 충 (Seong Chung Seo)

정회원

2011년 8월 : 고려대학교 정보보호대학원 박사

2013년 11월 : 삼성전자 종합기술원 전문연구원

2014년 4월 : 삼성전자 DMC 연구소 책임연구원

2019년 2월 : 국가보안기술연구소 선임연구원

2019년 3월~현재 : 국민대학교 금융정보보안학과 부교수

<관심분야> 암호최적화, 공개키 암호, 양자내성암호, 암호모듈검증, 네트워크보안