정 보 보 호 학 회 지
제 33 권 제2호, 2023. 4

NIST PQC 벤치마크 플랫폼 최신 동향

권 혁 동*, 심 민 주*, 송 경 주*, 이 민 우**, 서 화 정***

요 약

미국 국립표준기술연구소에서 개최한 양자내성암호 표준화 공모전이 2022년 표준화 선정을 마치고 4라운드 추가 표준 선정에 진입하였다. 공모전에 제출된 후보 알고리즘은 각자의 성능 지표를 제시하였지만, 서로 다른 환경에서 성능을 측정하였기에 객관적인 비교가 어려웠다. 이를 해결하기 위해 양자내성암호 벤치마크 플랫폼이 등장하였다. 양자내성암호 벤치마크 플랫폼은 양자내성암호 후보 알고리즘을 모두 동일한 환경에서 벤치마크하여 객관적인 결과를 제공하는데 집중하였다. 이때 각 알고리즘이 지니는 다양한 구동 의존성 때문에 알고리즘을 동일 환경에서 가동하는 데에 제한이 있다. 따라서 벤치마크 플랫폼은 알고리즘들의 의존성을 제거하고 동일한 환경을 제공하는데 집중하였으며 이를 통해 양자내성암호의 성능에 대한 접근성을 높여 사람들의 양자내성암호 연구에 큰 도움을 주었다.

Ⅰ. 서 론

양자컴퓨터의 발전에 따라 현대 암호 체계는 보안 관점에서 많은 위협을 받고 있다. 양자 알고리즘인 Grover 알고리즘은 높은 검색 능력을 통해 대칭키 알고리즘의 보안강도를 절반으로 낮추며[1], Shor 알고리즘은 다항시간안에 소인수분해가 가능하도록 하여 공개키 알고리즘에 위협적이다[2]. 특히 공개키 암호는 보안강도를 높이기 위해서는 전혀 다른 암호화 알고리즘인 양자내성암호를 적용해야 한다. 따라서 양자컴퓨터가 상용화되기 이전에 양자내성암호로의 빠른 이행이 요구되고 있다.

양자 컴퓨터가 모든 수학적 난제에 대한 해답을 다항시간 안에 해결할 수 있는 것은 아니다. 따라서 양자컴퓨터가 해결하기 어려운 수학적 문제에 기반을 둔 양자내성암호가 활발히 연구되고 있다. 그러나 새로운 암호 체계를 도입하기 위해서는 오랜 시간을 소요된다. 즉 양자내성암호의 개발부터 안전성 검증, 표준화 그리고 양자내성암호 전환까지 순차적인 절차가 필요하다[3].

미국 국립표준기술연구소(National Institute of Standards and Technology)는 미 상무부 소속의 비 관리기관으로, 산업 기술, 측정 기술 및 표준의 개발을 통해 미국 경제 성장을 촉진할 목적으로 설립되었다[4]. 현재 NIST에서 암호 기술의 표준화를 담당하고 있으며 2017년 초 양자 내성 암호 알고리즘(PQC, Post Quantum Cryptography) 공개 모집을 진행하였다.

본 기고에서는 양자내성암호 표준화 공모전에 대한 내용과 해당 암호화 알고리즘을 객관적으로 평가하기 위해 사용된 양자내성암호 벤치마크 플랫폼을 소개한다. 본 기고의 남은 부분 구성은 다음과 같다. 2장에서 양자내성암호 공모전에 대한 간략한 내용을 살펴보고 벤치마크 플랫폼에 탑재된 알고리즘에 대해서 확인한다. 3장에서 양자내성암호 벤치마크 플랫폼 2종인 PQClean 라이브러리와 pqm4에 대해서 알아본다. 4장에서 본 기고의 결론을 맺는다.

본 연구는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (<Q|Crypton>, No.2019-0-00033, 미래컴퓨팅 환경에 대비한 계산 복잡도 기반 암호 안전성 검증 기술개발, 50%) 그리고 본 연구는 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천기술 연구, 50%).

* 한성대학교 정보컴퓨터공학과 (대학원생, korlethean@gmail.com, minjoos9797@gmail.com, thdrudwn98@gmail.com)
** 한성대학교 IT융합공학부 (대학원생, minunejip@gmail.com)
*** 한성대학교 융합보안학과 (부교수, hwajeong84@gmail.com)

Ⅱ. 양자내성암호 표준화 공모전

2.1. NIST 양자내성암호 표준화 공모전

NIST는 다가올 미래 컴퓨팅 시대를 대비하여 2016년에 PQC 공모전에 대한 소개를 진행하였고, 이후 2017년에 1라운드, 2019년 2라운드, 2020년 3라운드를 진행하였다 [5]. 그리고 2022년엔 표준 암호들을 선정하였으며, 현재 추가 암호들에 대한 4라운드를 진행하고 있다.

NIST의 PQC 표준 공모전은 기존의 AES, SHA-2/SHA-3 표준 공모전과는 다르게 하나의 표준 암호 제정이 아닌 다양한 환경에서 여러 유형의 안전성과 효율성을 갖춘 PQC 표준 후보군 제정에 그 목적을 두고 있다 [6]. 2017년 12월에 진행된 1 라운드에선 NIST가 제시한 성능 조건을 만족하는 다양한 PQC 후보들 중 철회된 알고리즘 5개를 제외한 64개가 [표 1]과 같이 등록되었다.

2019년 6월에 진행된 2 라운드에선 알고리즘의 실용성과 안전성에 대한 평가가 이루어졌다. 당초 64개였던 PQC 후보들 중 38개가 효율성, 구조 및 알려진 공격법 등에 대한 문제들로 탈락하였고 [표 2]와 같이 26개의 알고리즘만이 후보만이 선정되었다. 이 중 대부분 알고리즘은 코드 기반과 격자 기반의 문제를 어려움으로 하는 알고리즘들이다.

2020년 7월에 진행된 3 라운드에서는 26개의 후보

〔표 1〕 NIST PQC 1라운드 알고리즘 통계

Round 1	Signature	KEM/ENC
Code	2	17
Lattice	5	21
Hash	3	0
Multivariate	7	2
ETC.	2	5

〔표 2〕 NIST PQC 2라운드 알고리즘 통계

Round 2	Signature	KEM/ENC
Code	0	7
Lattice	3	9
Hash	2	0
Multivariate	4	0
ETC.	0	1

〔표 3〕 NIST PQC 3라운드 알고리즘 통계

Round 3	Signature	KEM/ENC
Code	0	1(2)
Lattice	2	3(2)
Hash	0(2)	0
Multivariate	1(1)	0
ETC.	0	(1)

중 11개가 효율성 및 추가로 알려진 공격법 등에 의한 문제로 탈락하여 총 15개의 알고리즘이 후보로 선정되었다. 후보들은 7개의 최종 후보들과 8개의 대체 후보들로 이루어졌으며, 대체 후보들은 최종 후보들에 문제가 생길 시 대체를 위해 존재한다[5]. 3 라운드의 후보 알고리즘들은 [표 3]과 같으며, 대체 후보 알고리즘의 통계는 괄호 표시를 통해 명시하였다.

2022년 7월에 표준 암호 선정이 이루어졌다. KEM/ENC의 표준으로는 격자 기반의 어려움을 기반으로한 CRYSTALS-KYBER가 선정되었다. 전자 서명의 표준으로는 격자 기반의 어려움 기반인 CRYSTALS-DILITHIUM과 FALCON, 그리고 대칭/해시의 어려움 기반인 SPHINCS+가 지정되었다[7]. KEM/ENC 부문에서는 단일 알고리즘만이 선정되었다. 이에 NIST에서는 4라운드를 추가로 진행하여 Classic McEliece, BIKE, HQC, SIKE를 후보로 선정되었다.

2.2. 공개키 알고리즘

2.2.1. HQC

HQC는 Hamming Quasi-Cyclic의 약자로, NIST 양자내성암호 공모전의 Round 4 대체후보군에 진출한 코드 기반 공개키 알고리즘이다[8]. HQC는 Hamming metric에서 무작위 quasi-cyclic code를 디코딩하는 문제를 기반으로 하고 있다 [9]. [표 4]에서 HQC의 매개변수들을 확인할 수 있다..

2.2.2. CRYSTALS-KYBER

Kyber는 NIST 양자내성암호 공모전에서 표준 암호로 채택된 알고리즘으로 Module-Learning With Errors (M-LWE) 문제를 기반으로 한다[7]. Kyber는

〔표 4〕 HQC 매개변수, n: 벡터의 길이, n_1: Reed-Solomon code의 길이, n_2: Reed-Muller code의 길이, w: 벡터 x, y의 가중치, w_r: 벡터 r_1, r_2, e의 가중치, $[n, k, d]$: (code의 길이, code의 차수, code의 최소 거리).

Scheme	n	w	w_r	security	p_{fail}	Reed-Solomon $[n, k, d]$	Reed-Muller $[n, k, d]$
HQC128	17,669	66	75	128	$< 2^{-128}$	[46, 16, 15]	[384, 8, 192]
HQC192	35,851	100	114	192	$< 2^{-192}$	[56, 24, 16]	[640, 8, 320]
HQC256	57,637	131	149	256	$< 2^{-256}$	[90, 32, 29]	[640, 8, 320]

수식 1의 ring 상에서 연산을 진행한다. 수식 2는 ring의 매개변수이다. Kyber는 cycloiomic ring이 적용되고, Fujisaki-Okamoto 변환을 사용한다[11].

$$R_q = Z_{7681}/(X^{256}+1) \tag{1}$$

$$q = 7681 = 2^{13} - 2^9 + 1 \tag{2}$$

Kyber는 Number Theoretic Transform를 사용하기 때문에 빠른 연산이 가능하며, 추가적인 메모리 요구가 없다. [표 5]는 Kyber에서 사용되는 매개변수를 나열한 것이다. Kyber는 매개변수의 크기가 대부분 작기에 효율적으로 연산과 관리가 가능하다.

2.2.3. Classic McEliece

Classic McEliece는 NIST 양자내성암호 공모전의 Round 4 대체후보군에 진출한 코드기반 알고리즘이다. 1978년 McEliece 암호 시스템을 기반으로 하고 있으며, McEliece의 Niederreiter 변형을 기반으로 한다. Classic McEliece의 비밀키는 길이가 n인 바이너리 Goppa code를 수식 3과 같이 정의한다.

$$\Gamma_2(g, \alpha_1, \cdots, \alpha_n) = \{c \in \mathbb{F}_2^n \mid \sum_i c_i/(x-\alpha_i) \equiv 0 \bmod g \tag{3}$$

[표 6]은 Classic McEliece에서 사용되는 매개변수를 나열한 것이다. Classic McEliece는 매우 짧은 암호문 길이를 가지며, 빠른 시간 안에 암호화와 복호화가 가능하다. 하지만 공개키의 길이가 256KB~1.3MB까지로 매우 크며, 키 생성 과정이 느리다.

2.3. 전자서명 알고리즘

2.3.1. CRYSTALS-Dilithium[12]

CRYSTALS-Dilithium은 격자 기반 디지털 서명 알고리즘으로써 Module-LWE (Learning With Errors) 및 SIS (Short Integer Solution)의 수학적 난제를 암호에 사용한다. Module lattice 기반으로 설계되어 파라미터의 크기 크고 속도가 느리며 같은 CRYSTALS 상의 KYBER 와 기본 특성 및 구조가 유사하다. Dilithium에서 사용되는 NTT 에서는 $Z_q[x]/(X^n+1)$ $n=256$, 그리고 $q=8380417=2^{23}-2^{13}+1$ 으로 정의된 다항식 링 R_q으로 연산을 진행한다. Dilithium은 빠른 다항식 곱셈을 위해 n-point NTT (Number Theoretic Transform)를 사용하며 NTT 도메인에서 다항식의 인수를 모두 NTT 도메인으로 변환 후 곱셈을 진행하고 결과를 다시 역변환하여 최종 곱 다항식을 구한다. [표 7]은 Dilithium에서 사용되는 매개변수를 보여준다. Dilithium은 $(k\times l)(l\times 1)$의 행렬-벡터 다항식 곱셈이 키 생성, 서명, 검증에서 주된 연산이 된다. 이러한 연산의 반복이 Dilithium 연산의 시간에 큰

〔표 5〕 Kyber 파라미터, k: NIST 권장 보안 강도, q: n을 만족하는 작은 소수, η_1: s와 e의 노이즈, η_2: e_1과 e_2의 노이즈, d_u, d_v: 라운딩 파라미터, δ: 캡슐화 해제에 실패할 확률.

Scheme	n	k	q	η_1	η_2	(d_u, d_v)	δ
KYBER512	256	2	3329	3	2	(10, 4)	2^{-139}
KYBER768	256	3	3329	2	2	(10, 4)	2^{-164}
KYBER1024	256	4	3329	2	2	(11, 5)	2^{-174}

〔표 6〕 Classic McEliece 매개변수, m: 코드 길이, n: 바이너리 필드의 크기, t: 수정 가능한 오류 수, level: NIST 권장 보안 강도.

Scheme	m	n	t	$n-t$	level
348864	12	3488	64	768	1
348864**f**	12	3488	64	768	1
460896	13	4608	96	1248	3
460896**f**	13	4608	96	1248	3
6688128	13	6688	128	1664	5
6688128**f**	13	6688	128	1664	5
6960119	13	6960	119	1667	5
6960119**f**	13	6960	119	1667	5
8192128	13	8192	128	1664	5
8192128**f**	13	8192	128	1664	5

영향을 주며 내부 연산에서 SHAKE-256을 통해 연산 효율을 증가시켰다.

2.3.2. Falcon[13]

Falcon은 Sort Integer Solution (SIS) 문제를 기반으로 한 격자 기반 전자서명 알고리즘이다. Falcon은 NTRU 격자와 Fast Fourier SAMPLING 과 함께 2008년 Gentry, Peilert, Vaikuntanathan이 제안한 GPV framework를 사용하여 설계되었다[14].

Falcon은 n=512, n=1024 의 두 가지 규격으로 제공되며 각 규격에 대한 배개변수는 [표 8]과 같다. KeyGen/ms는 밀리초 당 키 생성, SIGN/s는 초 당 서명 생성, Verify/s는 초당 검증을 나타내고 pk, sig는 각각 BYTE 단위의 공개키 크기 및 서명 크기를 나타낸다. Falcon 내부에서는 NTT가 아닌 FFT(Fast Fourier Transform)을 활용하여 정수를 부동 소수점으로 변환시킨다. 키 생성 알고리즘에서 사용되는 ϕ는 다항식 $\phi = x^n + 1$, q는 모듈러 소수를 나타내며 모듈러 소수의 권장 크기는 $q = 12289$ 이다.

키 생성은 크게 두 가지 동작으로 구성된다. 첫 번째로 NTRU 방정식을 검증하는 다항식 f, g, F, $G \in Z[x]/(\phi)$을 계산한다. 그 다음 생성된 적절한 다항식 f, g, F, G를 통해 Falcon tree를 계산한다.

Falcon의 서명 과정은 다음과 같다. 메시지 m과 개인키 sk를 통해 salt r과 서명 s를 출력한다. 높은 수준에서 서명 생성 알고리즘은 메시지 m과 salt r을 통해 해시값 $c \in Z_q[x]/(\phi)$을 계산하고 비밀키에 대한 정보 f, g, F, G 를 사용하여 $s_1 + s_2 h = c \bmod q$을 만족하는 두 개의 short value s_1, s_2을 계산한다. HashToPoint 에서는 메시지를 해시 값을 다항식으로 바꾸고 ffSampling은 t의 점 위치를 가까운 격자 위의 점으로 옮기고 차이 값을 압축하여 그 값을 서명에 사용한다. Compress 단계는 서명을 압축한다.

2.3.3. SPHINCS+[15]

SPHINCS+는 SPHINCS의 속도 및 서명 크기를 개선한 stateless 해시 기반 서명 알고리즘이다. SPHINCS+의 주요 기여는 Forest of Random Subsets (FORS) 방식의 도입 및 leaf node 선택 방법이며 내부에서 암호화 속성이 있는 함수를 사용한다. [표 9]는 SPHINCS+에 대한 매개변수를 보여준다. 표의 Scheme에서 s/f는 각각 small 과 fast를 나타낸다. 그

〔표 7〕 Dilithium의 매개변수, (k,l): 행렬 크기, η: 샘플링 경계 값, β, ω: 거부 임계 값, pk: 공개키 크기(Byte), sig: 서명 크기, Level: NIST가 제시한 권장 보안 강도.

Scheme	(k,l)	η	β	ω	pk(Bytes)	sign(Bytes)	d	Level
Dilithium-2	(4, 4)	2	78	80	1,312	2,420	13	2
Dilithium-3	(6, 5)	4	196	55	1,952	3,293	13	3
Dilithium-4	(8, 7)	2	120	75	2,592	4,595	13	5

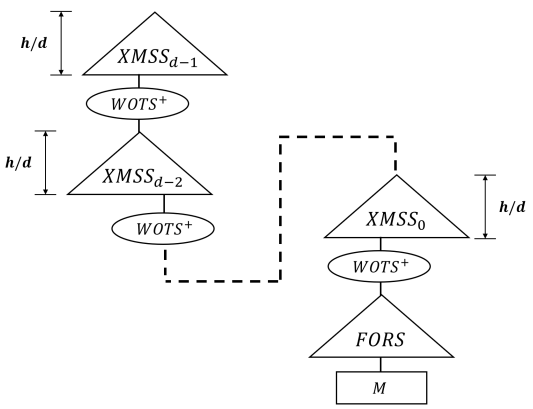
〔표 8〕 Falcon의 파라미터

Scheme	Keygen/ms	SIGN/s	Verify/s	pk(BYTES)	SIG(BYTES)
Falcon-512	8.64	5948.1	27933.0	897	666
Falcon-1024	27.45	2913.0	13650.0	1793	1289

〔표 9〕 SPHINCS+의 매개변수, n: Byte 단위의 보안 매개변수, w: Winternitz 매개변수, h: 하이퍼 트리의 높이, d: 하이퍼 트리의 레이어 수, k: FORS의 트리 수, t: FORS 트리의 leaf 수, bitsec: 〔SPHINCS〕의 Section 9를 통해 미리 계산된 비트 보안, sec level: 〔15〕의Section 4.A.5에서 지정된 보안 수준

Scheme	n	h	d	$\log(t)$	k	w	bitsec	sec level	sig bytes
SPHINCS+ -128s	16	64	8	15	10	16	133	1	8,080
SPHINCS+ -128f	16	60	20	9	30	16	128	1	16,976
SPHINCS+ -192s	24	64	8	16	14	16	196	3	17,064
SPHINCS+ -192f	24	66	22	8	33	16	194	3	35,664
SPHINCS+ -256s	32	64	8	14	22	16	255	5	29,792
SPHINCS+ -256f	32	68	17	10	30	16	254	5	49,216

림 1은 SPHINCS+의 구조를 보여준다. [그림 1]은 높이 h의 hyper-tree이며 d개의 tree로 구성된다. 각 tree의 높이는 h/d가 되며 이때, d는 서명 시간 및 서명 크기와 관련된다. Hyper-tree에서 $(d-1)$ layer 은 single- tree를 가지며 $(d-2)$ layer 은 $2^{(h/d)}$-tree를 가진다. Layer 0 WOTS+의 키 쌍은 HORST를 개선한 방식인 FORS 를 공개 키 서명에 사용된다. FORS는 ka-bit $(k, t=2^a)$의 정수로 정의 되어 ka-bit 문자열을 서명하는데 사용한다. FORS 개인키는 kt-bit의 랜덤 비트로 구성되며 k세트의 t 값으로 나뉜다. FORS의 공개키를 얻기 위해서 k 이진 해시 트리는 개인 키 요소들의 세트로 구성되며 각 t값은 각 leaf 노드에서 사용되고 높이가 a인 k개의 이진 해시 트리가 생성된다. WOTS+를 사용하면 루트 노드가



〔그림 1〕 SPHINCS, SPHINCS+ 알고리즘 구조

Tweakable hash function (Th_k)을 통해 압축된다. Th_k은 공개 매개변수인 P와 tweak T을 사용하여 α-bit 메시지 M을 λ-bit의 해시 값 MD로 매핑 시키는 효율적인 함수이며 수식 4로 작성된다.

$$Th : P \times T \times \{0,1\}^{\alpha} \rightarrow \{0,1\}^{\lambda}$$
$$Th(P,T,M) = H(\| tvert Tvertvert M) \quad (4)$$

Ⅲ. 양자내성암호 벤치마크 플랫폼

3.1. pqclean[16]

NIST 양자내성암호 공모전의 투고 알고리즘은 공통적으로 외부 라이브러리 의존성을 형성하는 문제점이 있었다. 이러한 의존성은 초기 구현 관점에서는 편리하지만, 다른 환경에서 재활용시에 다양한 구현 환경 설정을 요구하며 이런 부분에서 실질적 활용성이 떨어지게 된다. PQClean 라이브러리는 NIST 양자내성암호에 대한 의존성이 제거된 라이브러리를 제공하고자 구현되었다[17].

PQClean은 자체적인 벤치마크 기능을 포함하고 있지 않으나, 다른 플랫폼이나 라이브러리에 쉽게 통합될 수 있도록 설계되었다. 이는 PQClean 라이브러리의 가장 큰 특징으로, 외부 의존성이 제거되었기 때문에 가능하다. 가령 PQClean의 암호 알고리즘을 벤치마크 플랫폼인 SUPERCOP[18]에 통합시켜서 PQClean의 벤치마킹을 손쉽게 진행할 수 있도록 하였다. 비슷하게 다른 암호 라이브러리나 임베디드 플랫

폼 등에도 PQClean 코드를 포함시킬수 있다. 이는 PQClean 라이브러리의 가장 큰 장점으로, 의존성 제거를 통해 외부 라이브러리에 손쉽게 이식을 할 수 있게 하였다. 즉 뛰어난 확장성을 가지고 있어서 양자내성암호의 활용을 용이하게 만들었다.

PQClean은 외부 종속성 문제를 해결하기 위해서 일부 알고리즘을 내부에 이식하였다. 이에 해당되는 알고리즘은 SHA2, SHAKE, AES, DRBG 등이 있다. 기존 양자내성암호 알고리즘은 OpenSSL과 같은 외부 라이브러리에서 알고리즘을 참조한 반면, PQClean은 내부에 이식하는 것으로 외부 종속성을 제거하였다. 이를 통해 PQClean 라이브러리를 사용하는 작업자는 외부 종속성 없이 편리하게 작업할 수 있다.

PQClean 라이브러리는 양자내성암호 후보 알고리즘에서 단순히 종속성을 제거한 라이브러리가 아니다. 안정적으로 구현될 수 있도록 다양한 C언어 요구사항을 제시하고 이를 철저하게 지키는 방향으로 개발되었다. PQClean에서 확인되어야 할 요구사항은 [표 10]과 같다. PQClean에서는 구현의 편의를 위해 자동적으로 확인되는 요구사항과 직접 확인해야 하는 요구사항 두 가지로 분류하였다.

〔표 10〕 PQClean C 구현물에 대한 요구사항

Automatically checked	Manually checked
Code is valid C99	Minimalist Makefiles
Passes function tests	No stringification macros
API functions do not write outside provided buffers	Output-parameter pointers in functions are on the left
api.h cannot include external files	All exported symbols are namespaced in place
Compiles with -Wall -Wextra -Wpedantic -Werror -Wmissing-prototypes with gcc and clang	Integer types are of fixed size where relevant, using stdint.h types (optional, recommended)
#if/#ifdefs only for header encapsulation	Integers used for indexing memory are of size size_t (optional, recommended)
Consistent test vectors across runs and the other 15 requirements	Variable declarations at the beginning (except in for (size_t i=...) (optional, recommended)

PQClean 라이브러리는 스탠드얼론으로 라이브러리를 사용하기 위하기보다는 다른 플랫폼에 통합할 수 있는 소스코드를 지향한다. 때문에 PQClean의 내부에는 다양한 빌드에 도움을 줄 수 있는 테스트 도구들이 포함되어있다. 또한 이 부분에서 의존성이 발생하고 있는데, 여기에는 gcc, clang, make, python3, python-yam library, valgrind, astyle과 같은 것들이 있다.

기존 PQClean 라이브러리는 NIST 양자내성암호 공모전 Round 3의 알고리즘을 모두 구현되어 있다. 하지만 2022년 양자내성암호 표준안이 발표되고 Round 4에 진입하면서 탈락한 알고리즘은 라이브러리에서 제외되었다. 따라서 PQClean 라이브러리에서 제공하는 알고리즘은 공개키 알고리즘에서 CRYSTALS-Kyber, HQC, Classic McEliece, 전자서명 알고리즘에서 CRYSTALS-Dilithium, Falcon, SPHINCS+ 총 6개의 알고리즘만 제공되고 있다. 양자내성암호 표준화 공모전 4라운드의 결과에 따라 라이브러리의 내용은 수정될 가능성이 존재한다.

3.2. pqm4

pqm4 벤치마크 플랫폼은 NIST 양자내성암호 공모전의 후보 알고리즘들의 벤치마크를 위해 제작되었다. 특히 pqm4는 ARM Cortex-M4 프로세서를 대상으로 하는데, Cortex-M4는 임베디드 프로세서로 많이 활용되는 32-bit 프로세서이다.

기존 양자내성암호 벤치마크 결과가 존재함에도 pqm4 벤치마크가 등장한 이유는 크게 두 가지가 존재한다. 첫째는 기존 양자내성암호 벤치마크는 데스크탑 위주의 고성능 환경에서 진행되었다는 점이다. 양자내성암호는 양자컴퓨터로 인해 사용하기 어려워진 현대 암호체계를 대체하기 위해 개발되었다. 그런데 양자내성암호를 데스크탑 위주로 성능을 측정하고, 임베디드 상에서의 성능 측정이 없다면, 실제로 양자내성암호를 활용할 때 사물인터넷 기기와 같은 부분은 제대로 알고리즘을 가동하기 어렵거나 가동 효율이 매우 떨어질 수가 있다. pqm4는 임베디드 프로세서 상에서 양자내성암호의 벤치마크 결과를 제공하여 대부분의 환경에서 양자내성암호가 가동될 수 있음을 나타내는 지표로 활용가능하다.

둘째는 통일된 환경에서 벤치마크 결과가 없었다는 점이다. 양자내성암호 공모전에 투고된 알고리즘들은 각자 연구팀 자체 벤치마크 결과를 제공하고 있으나, 서로 다른 환경에서 벤치마크를 진행했기 때문에 온전한 성능 비교가 어려운 점이 있었다. pqm4는 이러한 맹점을 지적하며, 모든 알고리즘을 Cortex-M4 프로세서 상에서 성능 비교를 진행하였다[19].

pqm4에서는 4가지 종류의 구현 결과물을 제공하며 이는 [표 11]과 같다. 이렇게 다양한 구현 결과물을 제공하는 이유는 기본적인 동작 확인 외에도 Cortex-M4의 특징을 활용한 결과물의 벤치마크도 제공하기 위함이다. 레퍼런스 구현물은 기본적인 성능만을 제공하지만, ARM 어셈블리를 사용한 특수한 구현물은 알고리즘 설계에 따라 성능 차이가 크게 발생할 수 있다. 즉 ARM 프로세서에 적합한 구현물들에 대한 비교 분석이 가능하다는 특징이 있다[20].

〔표 11〕 pqm4의 구현물 버전

Type	Description
ref	The reference implementation submitted to NIST.
clean	Clean reference implementation from PQClean.
opt	An optimized implementation in plain C.
m4	An implementation with Cortex-M4 specific optimizations
m4f	An implementation with Cortex-M4F specific optimizations. (floting-point registers used.)

〔표 12〕 pqm4 구현물에 대한 요구사항

Type	RAM	Flash	ETC.
STM32f4 Discovery	192KB	1MB	Default option.
nucleo-1476rg	128KB	1MB	Does not require USB serial interface converter.
CW308t-STM32f3	40KB	256KB	-
nucleo-l4r5zi	640KB	2MB	Does not require USB serial interface converter.
mps2-an386	4MB	-	Has 2 RAM blocks, one used in lieu of Flash one as RAM

구현 결과물 버전과 비슷하게, 다양한 보드도 설정할 수 있다. pqm4가 대상으로 하는 보드는 [표 12]과 같다. 이렇게 다양한 보드를 제공하는 이유 역시 보드별로 가지는 하드웨어 자원이 다르기에 성능이 다르게 나오기 때문이다.

Ⅳ. 결 론

본 기고에서는 양자내성암호 벤치마크 플랫폼과 해당 플랫폼에 탑재된 알고리즘들에 대해서 확인하였다. PQClean 라이브러리는 외부 의존성을 제거한 스탠드얼론 형식의 라이브러리이다. 해당 라이브러리는 단독으로 가동하기에 용이한 점이 있으며 다른 라이브러리에 이식하여 더욱 효과적으로 가동할 수 있는 장점이 있다.

pqm4 라이브러리는 PQClean 라이브러리를 활용한 벤치마크 플랫폼으로, Cortex-M4 프로세서를 대상으로 양자내성암호 공모전 후보 알고리즘의 벤치마크 결과를 제공한다. pqm4에서 제공하는 벤치마크 결과물은 단순한 레퍼런스 구현물 뿐만 아니라 최적 구현물과 Cortex-M4의 명령어를 활용한 전용 구현물까지 제공한다.

이와 같이 대표적인 양자내성암호 벤치마크 플랫폼에 대해 확인해 보았다. 통일된 환경에서 제공되는 벤치마크 결과는 양자내성암호의 명확한 성능 비교가 되며, 의존성 제거 라이브러리는 암호 알고리즘의 활용에 있어서 큰 도움을 제공해주고 있다. 해당 실용적인 연구 결과를 바탕으로 실제 환경에서 양자내성암호의 활용도가 높아 질 것으로 기대해 본다.

참 고 문 헌

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search." *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212-219, 1996.

[2] P. W. Shor, "Algorithms for quantum

computation: discrete logarithms and factoring." *Proceedings 35th annual symposium on foundations of computer science*, pp. 124-134, Ieee, 1994.

[3] K. B. Jang, and H. J. Seo, “Quantum Computer and Standardization trend of NIST Post-Quantum Cryptography,” *Proceedings of the Korea Information Processing Society Conference*, 26(1), pp. 129-132, 2019.

[4] S. J. Baek, Y. J. Jeon, H. G. Kim, and J. S. Kim, “NIST 경량암호 공모 사업 동향,” Review of KIISC, 30(3), pp 17-24, 2020.

[5] H. J, Kim, J. H. Park, H. D. Gwon and H. J. Seo, “NIST 암호 표준화 공모전 동향” Review of KIISC, 30(6), pp. 117-123, 2020.

[6] T. H. Park, H. J. Seo, and H. W. Kim, “NIST 양자내성암호 표준공모전 제출물 분석 및 향후 연구전망,” *Korea Information Processing Society Review*, 24(6), pp. 55-63, 2017.

[7] NIST, “Selected Algorithms 2022,” Online: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[8] NIST, “Round 4 Submissions,” Online: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions, 2022.

[9] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, and I. C. Bourges, “Hamming quasi-cyclic (HQC),” *NIST PQC Round 2.4*, pp 13, 2018.

[10] S. Deshpande, M. Nawan, K. Nawaz, J. Szefer, and C. Xu, “Towards a fast and efficient hardware implementation of hqc,” *Cryptology ePrint Archive*, 2022.

[11] H.Becker, V.Hwang, M.J.Kannwischer, B.Y.Yang, and S.Y.Yang, "Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1," Cryptology ePrint Archive, Nov 2021.

[12] L. Ducas, E. Kilitz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, “Crystals-dilithium: A lattice-based digital signature scheme,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238-268, 2018.

[13] P. A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, “Falcon: Fast-Fourier lattice-based compact signatures over NTRU,” *Submission to the NIST’s post-quantum cryptography standardization process*, 36(5), 2018.

[14] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 197-206, 2008.

[15] D. J. Bernstein, A. Hülsing, S. Kölbi, R. Niederhagen, J. Rijneveld, and P. Schwabe, “The SPHINCS+ signature framework,” *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pp. 2129-2146, 2019.

[16] M. J. Kannwischer, P. Schwabe, D. Stebila, T. Wiggers, “Improving software quality in cryptography standardization projects,” *2022 IEEE European Symposium on Security and Privacy Workshops*, pp. 13-30, 2022.

[17] M. Kannwischer, J. Rijneveld, P. Schwabe, D. Stebila, and T. Wiggers, “PQClean: clean, portable, tested implementations of post quantum cryptography,” Online: https://github.com/PQClean/PQClean.

[18] D. J. Bernstein, and T. Lange, “eBACS: ECRYPT Benchmarking of Cryptographic Systems,” Online: http://bench.cr.yp.to.

[19] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, “pqc4: Testing and Benchmarking NIST PQC on ARM Cortex-M4,” 2019.

[20] H. J. Seo, “저성능 사물인터넷 상에서의 양자 내성 암호 구현,” *Review of KIISC*, 30(1), pp. 13-21, 2020.

〈저 자 소 개〉

권 혁 동 (Hyeok-Dong Kwon)

학생회원

2018년 2월 : 한성대학교 IT융합공학부 졸업

2020년 2월 : 한성대학교 IT융합공학부 석사

2020년 3월~현재 : 한성대학교 정보컴퓨터공학과 박사과정

<관심분야> 정보보안, 암호구현

심 민 주 (Min-Joo Sim)

학생회원

2021년 2월 : 한성대학교 IT융합공학부 졸업

2023년 2월 : 한성대학교 IT융합공학부 석사

2023년 3월~현재 : 한성대학교 정보컴퓨터공학과 박사과정

<관심분야> 정보보안, 암호구현

송 경 주 (Gyeong-Ju Song)

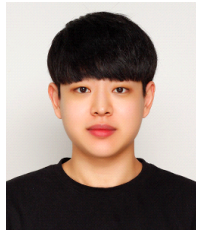
학생회원

2021년 2월 : 한성대학교 IT융합공학부 졸업

2023년 2월 : 한성대학교 IT융합공학부 석사

2023년 3월~현재 : 한성대학교 정보컴퓨터공학과 박사과정

<관심분야> 양자컴퓨팅, 암호구현, 정보보안



이 민 우 (Min-Woo Lee)

학생회원

2023년 2월 : 한성대학교 IT융합공학부 졸업

2023년 3월~현재 : 한성대학교 융합보안학과 석사과정

<관심분야> 정보보안, 암호구현

서 화 정 (Hwa-Jeong Seo)

종신회원

2010년 2월 : 부산대학교 컴퓨터공학과 졸업

2012년 2월 : 부산대학교 컴퓨터공학과 석사

2016년 2월 : 부산대학교 컴퓨터공학과 박사

2017년 4월~2023년 2월 : 한성대학교 IT융합공학부 조교수

2023년 3월~현재 : 한성대학교 융합보안학과 부교수

<관심분야> 정보보안, 암호구현