

NTRU+: KpqC 공모전 선정 격자 기반 공개키 암호

곽 현 지*, 이 미 라*, 김 종 현**, 박 종 환***

요 약

NTRU+는 NTRU를 기반으로 설계된 격자 기반 공개키 암호로, 2025년 1월 KpqC 공모전에서 최종 선정되었다. NTRU는 NTRU의 속도 개선을 위해 NTT를 도입했으나, 그 과정에서 암호문 크기가 증가하는 문제가 발생했다. 이를 완화하고자 매우 작은 복호화 실패 확률을 허용했으나, 실패 확률이 메시지 선택에 따라 달라져 CCA 공격에 취약한 문제로 인해 암호문 크기를 충분히 줄이지 못했다. 이 문제를 해결하고자 최약 복호화 실패 확률을 평균 수준에 가깝게 낮추는 ACWC 변환이 제안되었으나, NTRU에 적용 시 부채널 공격에 취약한 분포를 사용해야 한다는 단점이 있었다. NTRU+는 ACWC 변환의 이러한 한계를 극복한 ACWC₂ 변환을 도입했으며, 여기에 재암호화없이 CCA 안전성을 확보할 수 있는 변형된 FO 변환을 적용하여 효율성을 더욱 개선하였다. NTRU+는 Kyber 및 SMAUG-T보다는 암호문 크기는 다소 크지만, 동일 보안 수준에서 암호복호화에 필요한 CPU 사이클이 현저히 적어 실제 구현 성능 면에서도 우수함을 입증했다.

I. 서 론

양자컴퓨터의 발전은 기존 공개키 암호체계의 근본적인 전환을 요구하고 있다. 1994년 Shor가 소인수분해와 이산대수 문제를 다항 시간에 해결할 수 있는 양자 알고리즘[1]을 발표했으며, 이는 두 문제의 계산 복잡도에 기반해 안전성을 확보해 온 RSA와 ECDSA 같은 기존 공개키 암호시스템의 보안을 직접적인 위협에 노출시켰다. 비록 현재 양자컴퓨터 기술이 초기 단계에 머물러 있지만, 장기적인 관점에서 기존 암호를 대체할 양자내성암호(Post-Quantum Cryptography, PQC)를 준비하는 일은 필수적이다.

이에 따라 미국 국립표준기술연구소(NIST)를 비롯한 여러 국가와 기관이 PQC 표준화 작업을 활발히 추진하고 있다. NIST는 2016년 12월 PQC 표준화 프로젝트를 착수했으며, 2022년 7월 CRYSTALS-Kyber와 CRYSTALS-Dilithium 등을 첫 표준 알고리즘으로 확정했다. 국내에서는 국가보안기술연구소가 2022년 11월 ‘양자내성암호 국가공모전(KpqC Competition)’을 개최했고, 2025년 1월 NTRU+[2]가 공개키 암호(Public Key Encryption, PKE)/키 캡슐화(Key

Encapsulation Mechanism, KEM) 부문 최종 알고리즘 중 하나로 선정되었다.

NTRU+는 1998년 Hoffstein, Pipher, 그리고 Silverman이 제안한 격자 기반 암호인 NTRU[3]를 기반으로 설계된 공개키 암호이다. NTRU는 다항식환(polynomial ring) $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ 상에서 정의되며, 비밀키는 f 로, 공개키는 $h = pg/f$ 로 설정된다. 여기서 f 와 g 는 작은 절댓값의 계수를 가지는 R_q 상의 다항식이며, p 는 q 보다 작은 양의 정수로 q 와 서로소이다. m 에 대한 암호문은 $c = hr + m \in R_q$ 로 생성한다. 이때, 메시지 다항식 m 과 난수 다항식 r 은 모두 절댓값이 작은 계수를 갖는 R_q 상의 다항식이다. 암호문 c 로부터 메시지 다항식 m 을 복호화하는 과정은 $((cf \bmod q) \bmod p) \cdot f^{-1} \bmod p$ 연산을 통해 수행된다. 이때, $cf \equiv pgr + fm \pmod{q}$ 를 만족하는 다항식 $pgr + fm$ 의 모든 계수 절댓값이 $q/2$ 보다 작아야 복호화 과정에 성공하게 된다.

NTRU는 수십 년간 다양한 암호학적 분석과 공격을 견디며 안전성을 입증해 왔다. 이러한 장점으로 인해 NTRU는 NIST PQC 표준화 공모전 3라운드 최종 후

본 연구는 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2021-0-00518, 블록체인 데이터 암호화 기반의 프라이버시 보호 기술개발).

* 고려대학교 (대학원생, {ijnuyh01, mrlee01}@korea.ac.kr)

** 고려대학교 (박사 후 연구원, yoswuk@korea.ac.kr)

*** 상명대학교 (정교수, jhpark@smu.ac.kr)

보로 선정되었으나, CRYSTALS-Kyber에 비해 낮은 효율성으로 인해 최종 알고리즘으로는 채택되지 못하였다. 이러한 원인은 다항식환 연산을 고속화에 사용되는 Number Theoretic Transform(NTT)을 효율적으로 적용하기 어렵다는 구조적 제약 때문이다. 이는 q 와 p 라는 서로 다른 정수를 기반으로 정의된 다항식환 R_q 와 R_p 간에 변환이 필요하다는 점에서, 복호화 구조에 NTT를 효율적으로 적용하기 어려웠기 때문이다. 이러한 문제를 해결하기 위해 2019년 Lyubashevsky 등은 비밀키를 $f = pf' + 1$ 로 설정하여 $(cf \bmod q) \bmod p$ 연산만으로 복호화가 가능한 기존 NTRU 변형을 기반으로 NTT를 효율적으로 적용한 NTTRU[4]를 제안하였다.

하지만 변형 NTRU에서는 $pgr + fm$ 보다 더욱 큰 $pgr + pf'm + m$ 의 모든 계수의 절댓값이 $q/2$ 보다 작아야 한다는 제약 조건을 만족해야 하므로, 모듈러스 q 를 더욱 크게 설정해야 했고, 그 결과 암호문 크기가 커지는 문제가 발생했다. 이러한 문제를 완화하기 위해 작은 q 를 유지하면서도 소량의 복호화 실패를 허용하는 방향으로 전환되었다. 그러나 NTRU 계열 암호는 복호화 성공 확률이 메시지 m 에 의존하므로, 일부 복호화 실패를 허용하면 선택 암호문 공격(Chosen Ciphertext Attack, CCA)에서 공격자가 더욱 유리해진다. 공격자는 다양한 (메시지, 난수, 암호문) 목록을 생성한 후, 그중 복호화 실패가 발생한 항목을 골라 분석함으로써 비밀키를 복원할 수 있는데[5], 특히 복호화 실패 발생 확률이 높은 메시지 m 을 전략적으로 선택하면 공격 효과를 더욱 극대화할 수 있다.

이러한 문제를 해결하기 위해서는 복호화 실패 발생 확률이 가장 높은 메시지에 대해서도 복호화 실패 확률을 낮출 수 있는 설계가 필요해졌다. 2023년 Duman 등은 이러한 요구를 충족하기 위해 복호화 실패 확률이 높은 메시지에 대해서도 평균 복호화 실패 확률과 크게 다르지 않도록 보장하는 ACWC 변환을 제안하였다[6]. ACWC 변환은 메시지를 그대로 암호화하는 대신 GOTP(Generalized One-Time Pad)라는 사전 인코딩을 통해 모든 메시지의 계수가 특정 분포를 따르도록 강제함으로써, 최악의 메시지에 대해서도 복호화 실패율을 효과적으로 제어한다. 그러나 Duman 등이 제안한 변형 NTRU에 ACWC 변환을 적용한 NTRU-B 기법[6]은 인코딩 이전의 메시지 및 난수 다항식 계수를 $\{-1, 0, 1\}$ 에서 균등하게 선택되어야 한

다는 제약이 있다. 이를 이진수 기반의 컴퓨터에서 구현하려면 기각 샘플링(rejection sampling)을 수행해야 하나, 기각 횟수가 난수 값에 따라 달라져 상수 시간(constant-time) 구현이 어려워 부채널 공격에 취약하다는 근본적 한계를 지닌다.

NTRU+는 앞서 살펴본 NTRU 계열 기법들이 겪었던 NTT 적용의 어려움, 복호화 실패 확률에 따른 CCA 안전성 문제, 그리고 부채널 공격 취약성 등의 문제를 종합적으로 해결하기 위해 고안되었다. 구체적으로 NTTRU[4]와 마찬가지로 변형 NTRU 구조에 NTT를 도입하였고, ACWC[6]변환을 개선하고자 SOTP(Semi-Generalized One-Time Pad)라는 사전 인코딩을 사용하는 ACWC₂ 변환을 도입하여 최악의 메시지에 대한 복호화 실패율과 CCA 안전성, 부채널 공격 저항성을 동시에 확보한다. 추가로 메시지 m 과 암호문 $c = hr + m$ 로부터 난수 다항식 r 를 복원할 수 있는 NTRU의 특징을 이용하여 CCA에 안전하기 위해 사용되는 FO 변환[7]에 필요한 재암호화 과정을 제거함으로써 구현 효율성과 부채널 공격 저항성을 동시에 향상시켰다.

본 논문은 NTRU+의 핵심 구성 요소인 SOTP 인코딩, ACWC₂ 변환, 그리고 재암호화 과정을 제거한 FO 변환을 차례로 살펴본다. 이어서 이들 요소를 통합한 NTRU+KEM과 NTRU+PKE의 전체 구조, 주요 성능, 보안 특성을 살펴본다.

II. 배경 지식

2.1. CCA 안전성

CCA는 KEM/PKE에 대한 가장 강력한 공격 모델로, 공격자는 목표 암호문을 제외한 모든 암호문에 대해 복호화 권한을 갖는다. 만약 공격자가 목표 암호문 c 를 변조하여 c' 와 연관된 유효한 암호문 c' 를 생성하고, c' 에 대한 복호화 결과로 c' 에 대응하는 키/평문 정보를 얻을 수 있다면, c 에 대한 키/평문 정보를 얻을 수 있게 된다. 따라서 CCA 안전성을 만족하려면 공격자가 임의의 방식으로 암호문을 변조해도 복호화가 성공하지 않도록 해야 한다.

2.2. FO 변환

1999년 Fujisaki와 Okamoto가 제안한 FO 변환[7]

은 공개키 암호의 가장 기본적 공격인 선택 평문 공격 (Chosen-Plaintext Attack, CPA)에 안전한 PKE를 CCA에 안전한 KEM으로 변환하기 위해 널리 사용되는 기법이다. FO 변환은 CCA 공격자가 암호문을 변조해도 재암호화 검증을 통과하지 못하도록 하여 암호문의 유효성을 검증한다. FO 변환의 구체적인 절차는 다음과 같다. 송신자는 난수 메시지 m 을 선택하고 $(r, K) = H(m)$ 와 같이 해시함수 H 로 난수 r 와 키 K 를 생성하여 암호문 $c = \text{Enc}(pk, m; r)$ 를 생성한다. 수신자는 암호문 c 를 복호화해 메시지 m' 을 얻고, $(r', K') = H(m')$ 로 난수 r' 를 생성한 후, 재암호화를 통해 $c' = \text{Enc}(pk, m'; r')$ 을 계산한다. 이후 c' 와 c 가 같으면 유효한 암호문으로 판단하여 키 K' 를 복호화 결과로 출력한다.

2.3. 복호화 실패

복호화 실패(decryption failure)란 송신자가 메시지와 난수를 이용해 생성한 암호문을 수신자가 복호화했을 때 원래 메시지를 정확히 복원하지 못하는 것을 말한다. FO 변환을 기반으로 설계된 KEM은 암호화를 위한 난수를 메시지를 해시하여 생성하므로 공격자가 난수를 임의로 조작하기 어렵다. 따라서 복호화 실패 확률은 메시지 분포 관점에서 정의하며, 분포에 따라 샘플링된 메시지의 실패 확률을 평균 복호화 실패 확률(average-case correctness error)로, 가장 실패율이 높은 메시지의 실패 확률을 최악 복호화 실패 확률(worst-case correctness error)로 구분한다.

복호화 실패는 사용자 편의성 저하뿐 아니라, 공격자가 오류가 발생한 (메시지, 난수, 암호문) 목록으로부터 비밀키 정보를 얻을 수 있다는 점에서 치명적이다. NTRU의 경우, 공격자가 복호화 에러가 발생하는 메시지 m 및 난수 r 를 알고 있다면, 복호화 실패가 발생했기 때문에 $\|pgr + fm\|_\infty > q/2$ 를 만족함을 알 수 있다. 이때, m 및 r 에 대한 정보를 알고 있으므로 비밀키 역할을 할 수 있는 g 와 f 에 대한 부등식 정보를 얻을 수 있다. 공격자가 이러한 정보를 누적하면 g 와 f 를 복원할 수 있게 된다. CRYSTALS-Kyber와 같이 복호화 실패 확률이 메시지에 의존적이지 않은 암호는 평균 복호화 실패 확률만으로 FO 변환을 통해 CCA 안전성을 만족할 수 있지만, NTRU 계열은 복호화 실패 확률이 메시지에 의해 결정되므로 CCA에 안

전한 기법을 설계하기 위해서는 최악 복호화 실패 확률 또한 고려해야 한다.

2.4. ACWC 변환

ACWC 변환[6]은 PKE가 평균 복호화 실패 확률만 낮은 것이 보장될 때, 그 구조적 한계를 극복하기 위해 고안된 변환으로, 변환 후 기법의 최악 복호화 실패 확률이 변환 전 평균 실패 확률과 거의 동일한 수준으로 유지되도록 보장한다. 여기서 쓰이는 핵심 도구는 One-Time Pad를 일반화한 GOTP다. GOTP는 메시지 x 와 난수 u 를 입력받아 $y = \text{GOTP}(x, u)$ 를 생성하며, 이때 y 는 x 와 통계적으로 독립인 균등 분포를 따른다. ACWC 변환의 암호화는 인코딩용 난수 M_1 을 랜덤하게 선택하고, 해시함수와 메시지 m 을 이용해 $M_2 = \text{GOTP}(m, H(M_1))$ 를 생성한다. 이후, 인코딩된 메시지 $M = M_1 \| M_2$ 에 대한 암호문을 생성한다. 랜덤 오라클로 모델링된 해시함수 H 의 출력은 랜덤하게 선택되기에 GOTP의 성질에 따라 M_2 또한 랜덤하게 되어 어떤 메시지 m 을 선택해도 복호화 실패 확률을 인위적으로 높이기 어렵게 한다. 복호화는 $M = M_1 \| M_2$ 를 복원한 뒤 GOTP에 대응하는 Inv 함수를 통해 $m = \text{Inv}(M_2, H(M_1))$ 을 복원한다.

이와 같은 ACWC 변환에도 불구하고 ACWC를 적용하여 설계된 NTRU-B($p=3$ 를 사용)는 메시지 다항식 계수가 $\{-1, 0, 1\}$ 로 제한되는 문제가 있다. 암호화 시 인코딩용 난수 M_1 은 $\{-1, 0, 1\}$ 에서 균등하게 선택해야 하나, 대부분의 컴퓨터는 비트 단위로 난수를 생성해 2^n 중 하나를 균등한 확률로 제공하기 때문에 $\{-1, 0, 1\}$ 에 벗어난 값을 선택하면 범위에 맞는 값이 나올 때까지 다시 뽑기를 반복하는 기각 샘플링을 해야한다. 따라서 샘플링마다 실행 시간이 달라지고, 결과적으로 공격자는 시간 차를 이용해 부채널 공격으로 난수 정보를 유출할 수 있게 된다. 따라서 NTRU-B에서는 메시지에 무관한 최악 복호화 실패 확률 보장을 안전하고 효율적으로 구현하기 어렵다.

III. NTRU+

본 장에서는 NTRU+의 설계 원리를 중심으로 그 구조를 살펴본다. 우선, 최악 복호화 실패 확률을 평균 복호화 실패 확률 수준으로 낮추면서 부채널 공격에

대응하기 위해 도입된 $ACWC_2$ 변환을 살펴보고, 이어 이를 바탕으로 재암호화 과정 없이 CCA 안전성을 만족하는 변형된 FO 변환을 살펴본다. 마지막으로 NTRU+ 기법과 그 성능을 살펴본다.

3.1. $ACWC_2$ 변환

3.1.1. SOTP

SOTP는 최악 복호화 실패 확률을 평균 복호화 실패 확률로 변환하기 위해 사용되는 인코딩으로, 인코딩 전과 후의 분포를 독립적으로 만들어 주는 것이 목표이다. 즉, 어떤 메시지를 인코딩하더라도 그 결과가 특정 분포를 따르도록 강제함으로써, 복호화 실패율이 입력 메시지에 의존하지 않게 하려고 사용된다.

SOTP는 입력 메시지 x 을 난수 u 를 이용하여 코드 y 로 인코딩하는 SOTP와 코드 y 을 난수 u 를 이용하여 디코딩하는 Inv 알고리즘으로 이루어져 있다. SOTP는 인코딩한 메시지를 디코딩했을 경우 메시지를 정확히 복구되어야 한다는 기본 성질 이외에 두 가지 추가적인 성질이 필요하다.

- **Message-hiding:** 난수 u 가 정상적인 분포에서 선택 되었으면 코드 y 결과의 분포가 입력 메시지 x 와 독립적인 분포를 따라야 한다.
- **Rigidity:** 코드 y 를 난수 u 로 디코딩하였을 때, 그 결과가 유효한 경우 이를 다시 인코딩하면 원래의 코드 y 와 동일한 결과가 출력되어야 한다.

GOTP에서는 인코딩 과정에 사용되는 난수 정보를 노출되면 안 된다는 Randomness-hiding 성질이 추가로 필요해 메시지가 특정 분포를 따라야 하는 제약이 있었지만, SOTP는 Message-hiding 성질만 요구하므로 메시지 m 분포에 제약이 없다. SOTP는 GOTP보다 요구 사항이 완화되어, 보다 다양한 분포를 사용할 수 있다. NTRU+에서는 상수 시간 구현이 가능한 중심 이항 분포(Centered Binomial Distribution, CBD)를 지원하는 SOTP를 사용하며, 구체적인 인코딩 방법은 다음과 같다.

- SOTP(x, u)
 - 1) $u = (u_1, u_2) \in \{0,1\}^n \times \{0,1\}^n$ 를 파싱한다.

- 2) $y = (x \oplus u_1) - u_2 \in \{-1, 0, 1\}^n$ 를 계산한다.
- 3) y 를 출력한다.

- Inv(y, u)
 - 1) $u = (u_1, u_2) \in \{0,1\}^n \times \{0,1\}^n$ 를 파싱한다.
 - 2) 만약 $y + u_2 \notin \{0,1\}^n$ 이면 \perp 을 출력한다.
 - 3) $x = (y + u_2) \oplus u_1 \in \{0,1\}^n$ 를 출력한다.

난수 u_1 을 이용하여 $x \oplus u_1$ 를 계산하면 One-Time Pad 성질에 의해 메시지 x 의 정보가 은폐되어 균일한 랜덤 분포를 따르게 된다. 이후, $y = (m \oplus u_1) - u_2$ 와 같이 두 난수 값의 차를 계산하면 최종적으로 생성된 코드 y 가 CBD 분포를 따르게 된다.

3.1.2. $ACWC_2$

$ACWC_2$ 변환은 SOTP 인코딩을 기반으로 하여, 최악의 경우 복호화 실패율을 평균의 경우 복호화 실패율로 전환하도록 설계된 변환이다. $ACWC_2$ 변환을 통해 생성된 PKE' 기법은 기반하는 PKE 가 CPA에 안전하다면 PKE' 역시 CPA에 안전하다. 기존 GOTP 기반 $ACWC$ 변환은 KEM 설계에 적용 가능하지만, 메시지 분포에 제약이 있어 PKE 로의 확장은 어렵다는 한계가 있다. 반면, $ACWC_2$ 변환은 SOTP를 통해 메시지 분포 제약 문제를 효과적으로 극복하여 PKE 설계도 가능하다.

$PKE = (\text{Gen}, \text{Enc}, \text{Dec})$ 에 $ACWC_2$ 변환을 적용하기 위해서는 여러 성질이 필요하나, 설명을 단순화하기 위해 난수 복구 성질만 설명한다.

• 난수 복원(Randomness Recovery, RR)

해당 성질을 단순화해 설명하면 $c = \text{Enc}(pk, m; r)$ 와 같이 생성된 암호문에 대해 암호문과 c 와 메시지 m 이 주어졌을 때, 암호문을 생성했을 때 사용한 난수 r 을 복원할 수 있는 성질이다. NTRU는 암호문 $c = hr + m$ 에서 $r = (c - m)h^{-1}$ 과정을 통해 난수를 복원할 수 있다. 난수 복원 알고리즘은 $r = \text{RRec}(pk, m, c)$ 와 같이 표기한다.

해시함수를 G 를 이용하여 $ACWC_2$ 변환을 적용한 기법을 $PKE' = ACWC_2[PKE, SOTP, G]$ 라 할 때,

구체적인 변환은 다음과 같다.

- $\text{Gen}'(1^\lambda)$
 - 1) $(pk, sk) = \text{Gen}(1^\lambda)$
- $\text{Enc}'(pk, m, R)$
 - 1) R 로 암호화에 사용할 난수 r 을 생성한다.
 - 2) $M = \text{SOTP}(m, G(r))$ 를 생성한다.
 - 3) 암호문 $c = \text{Enc}(pk, M; r)$ 을 출력한다.
- $\text{Dec}'(sk, c)$
 - 1) $M' = \text{Dec}(sk, c)$ 을 계산한다.
 - 2) $r' = \text{RRec}(pk, M, c)$ 을 계산한다.
 - 3) $m' = \text{Inv}(M, G(r'))$ 을 계산한다.
 - 4) r' 혹은 m' 이 \perp 인 경우 \perp 을 출력한다.
 - 5) m' 을 출력한다.

3.2. 재암호화 없는 FO 변환

일반적으로 CPA에 안전한 기법은 FO 변환을 적용하여 CCA에 안전한 기법을 설계할 수 있다. 하지만, FO 변환은 복호화 과정에서 암호문이 올바르게 생성되었는지 확인하기 위해 암호문을 다시 암호화하는 비효율적인 재암호화 과정을 수행한다. NTRU+에서는 비효율적인 재암호화 과정을 피하고자 기반하는 암호 기법이 가지는 난수 복원 성질을 활용하여 재암호화 과정이 없는 변환인 $\overline{\text{FO}}_{\text{KEM}}^\perp$ 와 $\overline{\text{FO}}_{\text{PKE}}^\perp$ 을 제안하였다. 해당 변환은 난수 복원 알고리즘 RRec을 통해 복원한 난수 r 과 복호화된 메시지 m 를 해시하여 생성한 난수 r' 이 일치하는지 확인한다. 이러한 단순한 조건 검사만으로도 암호문 정당성을 충분히 검증할 수 있으며, 재암호화 과정을 제거하면서도 이전의 FO 변환을 통한 CCA의 안전성과 동일한 수준의 안전성을 유지한다. 이 방법은 RRec로 인해 연산이 추가되지만, 재암호화 과정을 생략함으로써 복호화 속도를 향상한다.

3.3. NTRU+ 기법

본 절에서는 NTRU+KEM와 NTRU+PKE 기법을 살펴본다. 이들 기법은 SOTP 인코딩(3.1.1절), ACWC₂ 변환(3.1.2절), 그리고 재암호화 없는 FO 변환(3.2절)을 바탕으로 설계된다. 두 기법은 다양한 파라미터를

위해 공통으로 순환 삼항식(cyclotomic trinomial)에 기반한 $R_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1 \rangle$ 상에서 정의되며 구체적으로 모듈러스 $q = 3457$ 와 다항식 차수 $n = \{576, 768, 864, 1152\}$ 를 사용한다.

3.3.1. NTRU+KEM 기법

NTRU+KEM의 세부 알고리즘은 다음과 같다.

- $\text{Gen}(1^\lambda)$
 - 1) ψ_1 에서 계수를 샘플링한 다항식 f', g 를 생성한다. ψ_1 는 중심 이항 분포를 의미하며, $\{-1, 0, 1\}$ 에서 각 $\{1/4, 1/2, 1/4\}$ 의 확률로 샘플링한다.
 - 2) 비밀키 $f = 3f' + 1$ 을 계산한다. 이때 R_q 상에 f, g 의 역원이 존재해야 한다. 역원이 존재하지 않는 경우 1)로 돌아가 다시 생성한다.
 - 3) 공개키 $h = 3 \times g \times f^{-1}$ 를 계산한다.
 - 4) 공개키 $pk = h$ 와 비밀키 $sk = (f, h^{-1}, F(pk))$ 를 출력한다. F 는 공개키 압축을 위한 해시함수다.
- $\text{Encap}(pk)$
 - 1) $m \leftarrow \{0, 1\}^n$ 을 선택한다.
 - 2) $(R, K) = H_{\text{KEM}}(m, F(pk))$ 을 계산한다.
 H_{KEM} 은 난수 및 키 생성을 위한 해시함수이다.
 - 3) 난수 R 를 기반으로 $r \leftarrow \psi_1^n$ 을 선택한다.
 - 4) $m = \text{SOTP}(m, G(r))$ 를 계산한다.
여기서 SOTP는 m 과 $G(r)$ 을 이용해 중심 이항 분포를 따르는 m 을 출력한다. (3.1.1절 참조)
 - 5) 암호문 $c = hr + m$ 을 계산한다.
 - 6) 암호문과 비밀키를 (c, K) 를 출력한다.
- $\text{Decap}(sk, c)$
 - 1) $m' = (cf \bmod q) \bmod 3$ 를 계산한다.
 - 2) $r' = (c - m')h^{-1}$ 를 계산한다.
 - 3) $m' = \text{Inv}(m', G(r'))$ 를 계산한다.
 Inv 는 SOTP의 역과정으로 m' 와 $G(r')$ 을 입력 받아 m' 을 출력한다. (3.1.1절 참조)
 - 4) $(R', K') = H_{\text{KEM}}(m', F(pk))$ 을 계산한다.
 - 5) 난수 R' 을 기반으로 $r'' \leftarrow \psi_1^n$ 을 선택한다.
 - 6) $m' = \perp$ 이거나 $r' \neq r''$ 이면 실패를 나타내는 \perp 를 출력하고, 그 외의 경우 K' 를 출력한다.

3.3.2. NTRU+PKE 기법

NTRU+PKE의 세부 알고리즘은 다음과 같다. $\text{Gen}(1^\lambda)$ 는 NTRU+KEM에서 정의된 함수와 동일하여 기술을 생략한다.

- $\text{Enc}(pk, m \in \{0,1\}^{l_m})$
 - 1) $r \leftarrow \{0,1\}^l$ 을 선택한다.
 - 2) $\tilde{m} = m \| r \in \{0,1\}^{l_m + l_r}$ 을 생성한다.
 - 3) $R = H_{\text{PKE}}(\tilde{m}, F(pk))$ 를 계산한다.
 H_{PKE} 은 난수 생성을 위한 해시함수이다.
 - 4) 난수 R 를 기반으로 $\mathbf{r} \leftarrow \psi_1^n$ 을 선택한다.
 - 5) $\mathbf{m} = \text{SOTP}(\tilde{m}, G(\mathbf{r}))$ 를 계산한다.
 - 6) 암호문 $\mathbf{c} = \mathbf{h}\mathbf{r} + \mathbf{m}$ 을 계산하여 출력한다.
- $\text{Dec}(sk, \mathbf{c})$
 - 1) $\mathbf{m}' = (\mathbf{c}\mathbf{f} \bmod q) \bmod 3$ 를 계산한다.
 - 2) $\mathbf{r}' = (\mathbf{c} - \mathbf{m}')\mathbf{h}^{-1}$ 를 계산한다.
 - 3) $\tilde{m}' = \text{Inv}(\mathbf{m}', G(\mathbf{r}'))$ 를 계산한다.
 - 4) $R' = H_{\text{PKE}}(\tilde{m}', F(pk))$ 를 계산한다.
 - 5) 난수 R' 을 기반으로 $\mathbf{r}'' \leftarrow \psi_1^n$ 을 선택한다.
 - 6) $\tilde{m}' = \perp$ 이거나 $\mathbf{r}' \neq \mathbf{r}''$ 이면 실패를 나타내는 \perp 를 출력하고, 그 외의 경우 $[\tilde{m}]_{l_m}$ 를 출력한다.

3.4. NTRU+ 안전성 및 성능 비교

본 절에서는 NTRU+KEM의 보안 강도와 성능을 측정하고, 이를 NIST PQC 표준화 공모전에서 최종 선정된 CRYSTALS-Kyber[8] 및 KpqC 공모전에서 최종 선정된 또 다른 기법인 SMAUG-T[9]와 비교한다. [표 1]에는 세 가지 기법에 대해 LWE 문제의 보안 강도, 공개키·비밀키 및 암호문 크기, 그리고 C 언어 구현과 AVX2 최적화 구현에서의 키 생성(Gen), 캡슐화(Encap), 복호화(Decap)에 소요된 CPU 사이클을 정리하였다. LWE 문제의 보안 강도는 Lattice Estimator[10]를 사용해 분석했으며, 성능 측정은 Intel Core i7-8700K(3.70 GHz, 싱글 코어, 하이퍼스레딩 비활성화), 64 GB RAM, Ubuntu 22.04 LTS 환경에서 GCC 11.4를 -O3 옵션과 함께 적용하여 수행하였다. 특히 CPU 사이클 값은 각 알고리즘을 10만 회 반복 실행하여 얻은 평균값이다.

[표 1]을 살펴보면 동일 보안 수준에서 오히려 NTRU+KEM의 공개키와 암호문의 크기가 CRYSTALS-Kyber와 SMAUG-T보다 더 크다는 사실을 확인할 수 있다. 일반적으로 NTRU+의 공개키와 암호문은 단일 다항식으로만 구성되므로 크기가 작을 것으로 기대되지만, 실제로는 그렇지 않다. CRYSTALS-Kyber와 SMAUG-T처럼 격자 기반 암호가 Diffie-Hellman 키 교환 형태로 설계된 경우에는 암호문 상위 비트에 메시지를 먼저 인코딩한 뒤 하위 비트를 잘라내어(truncation) 전송량을 효과적으로 줄

[표 1] NTRU+KEM, Kyber, 그리고 SMAUG-T의 성능 비교

Scheme	Sec (c)	Sec (q)	n	q	pk (byte)	ct (byte)	sk (byte)	$\log_2 \delta'$	C (K cycles)			AVX2 (K cycles)		
									Gen	Encap	Decap	Gen	Encap	Decap
NTRU+KEM	111	99	576	3457	864	864	1760	-487	102	60	58	26	26	16
	156	139	768	3457	1152	1152	2336	-379	124	80	75	28	32	19
	176	160	864	3457	1296	1296	2624	-340	147	91	87	30	36	22
	248	222	1152	3457	1728	1728	3488	-260	226	118	119	46	47	29
CRYSTALS-Kyber	115	103	256×2	3329	800	768	1632	-139	132	157	199	27	29	30
	174	155	256×3	3329	1184	1088	2400	-164	231	256	316	45	44	46
	241	215	256×4	3329	1568	1568	3168	-174	337	387	464	62	63	67
SMAUG-T	112	100	256×2	1024	672	608	832	-161	117	102	137	42	25	35
	112	100	256×2	1024	672	672	832	-118	116	103	136	42	25	34
	174	156	256×3	2048	1088	992	1312	-179	222	205	254	60	47	61
	236	211	256×4	2048	1440	1374	1792	-194	357	337	401	82	66	85

일 수 있다. 그러나 NTRU+는 메시지를 암호문의 하위 비트 영역에 직접 인코딩하므로, Kyber나 SMAUG-T 방식과 같은 트렁케이션 기법을 적용할 수 없다. 이 때문에 NTRU+의 암호문 크기를 추가로 줄이기가 어려워 결과적으로 동일 보안 수준에서 Kyber와 SMAUG-T보다 더 큰 암호문 크기를 갖게 되는 것이다.

그러나 알고리즘 속도 측면에서는 NTRU+KEM이 CRYSTALS-Kyber와 SMAUG-T보다 빠른 것을 볼 수 있다. CRYSTALS-Kyber와 SMAUG-T는 공통적으로 Module-LWE 문제를 기반으로 Diffie-Hellman 키 교환 구조 형태로 기법이 설계되어 다항식 곱셈 시 상대적으로 복잡한 연산을 수행해야 한다. 특히 CRYSTALS-Kyber는 메시지를 상위 비트에 인코딩한 뒤 하위 비트를 잘라내는 트렁케이션을 다항식환 공간에서 해야 하므로, 암호문 생성 및 복호화 과정에서 NTT와 역NTT 연산이 반복 호출된다. 따라서 곱셈 연산 횟수가 증가하고 전체 실행 속도가 느려진다. SMAUG-T의 경우, NTT를 직접 지원하지 않아, 실제 모듈러스보다 큰 값을 이용한 NTT를 사용해야 하기 때문에 이 과정에서 성능이 저하된다[11].

반면, NTRU+는 암호문이 단일 다항식으로만 구성되고 하위 비트를 잘라내지 않기 때문에 처음부터 암호문을 NTT 도메인에 보관할 수 있다. 따라서 암호문 생성·복호화 시 불필요한 NTT 및 역NTT 호출을 최소화할 수 있으며, 곱셈 연산 횟수 자체도 CRYSTALS-Kyber나 SMAUG-T보다 적다. 또한 NTRU+는 기본적으로 NTT를 지원하므로 다항식 곱셈 시 별도 변환 없이 바로 NTT 기반 연산을 수행하여 실행 속도가 크게 향상된다.

IV. 결 론

본 논문에서는 KpqC 공모전에 최종 선정된 NTRU+에 대해 살펴보았다. NTRU+는 기존 NTRU 계열 암호의 한계를 극복하기 위해 세 가지 핵심 기법을 결합하여 설계된 격자 기반 공개키 암호 기법이다. 첫째, SOTP 인코딩을 도입함으로써 메시지 분포를 완전히 은닉하면서도 비트 연산만으로 복호화 실패율을 평균 수준으로 전환할 수 있도록 하였다. 둘째, 중심 이항 분포를 지원하는 SOTP 기반의 ACWC₂ 변환을 적용하여 최악 복호화 실패 확률을 평균 복호화 실패 확률 수준으로 낮게 유지하면서 기존 ACWC 변환이

부채널 공격에 안전하지 않은 분포를 사용하는 문제를 해결하였다. 셋째, 재암호화 과정을 사용하지 않는 변형된 FO 변환을 통해 암호문 검증 절차를 간소화하고 부채널 공격에 대한 내성을 확보하였다.

NTRU+가 KpqC 공모전에서 최종 선정된 만큼, 향후 다양한 응용 환경에서의 확장 및 경량화 구현, 부채널 공격 분석, 그리고 인증·키 관리 구조로의 확대 등 후속 연구가 기대된다.

참 고 문 헌

- [1] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th annual symposium on foundations of computer science, IEEE, pp. 124-134, Nov. 1994.
- [2] J. Kim and J.H. Park, "NTRU+ Algorithm Specifications And Supporting Documentation (version 2.0)," Korea Post-Quantum Cryptography Forum., pp. 1-59, Jan. 2025. available at https://kpqc.or.kr/competition_02.html
- [3] J. Hoffstein, J. Pipher, and J.H. Silverman, "NTRU: A ring-based public key cryptosystem," International algorithmic number theory symposium, pp. 267-288, Jan. 1998.
- [4] V. Lyubashevsky and G. Seiler, "NTTTRU: truly fast NTRU using NTT," Cryptology ePrint Archive, vol. 2019, no. 3, pp. 180-201, Feb. 2019.
- [5] N. Howgrave-Graham, P.Q. Nguyen, D. Pointcheval, J. Proos, J.H. Silverman, A. Singer, W. Whyte, "The impact of decryption failures on the security of NTRU encryption," In: Annual International Cryptology Conference. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.
- [6] J. Duman, K. Hövelmanns, E. Kiltz, V. Lyubashevsky, G. Seiler, and D. Unruh, "A thorough treatment of highly-efficient NTRU instantiations," In Alexandra Boldyreva and Vladimir Kolesnikov, editors, PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part I, volume 13940 of Lecture Notes in Computer Science, pa-

- ges 65 - 94, Atlanta, GA, USA, May 7 - 10, 2023. Springer, Cham, Switzerland.
- [7] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, CRYPTO'99, volume 1666 of LNCS, pages 537 - 554. Springer, Heidelberg, August 1999.
- [8] P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J.M. Schanck, G. Seiler, D. Stehle, and J. Ding, "CRYSTALS-KYBER," Technical report, National Institute of Standards and Technology, pp. 1-43, Oct. 2020. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>
- [9] J. H. Cheon, H. Choe, J. Choi, D. Hong, J. Hong, C.-G. Jung, H. Kang, J. Lee, S. Lim, A. Park, S. Park, H. Seong, and J. Shin, "SMAUG-T: Algorithm specifications and supporting documentation v4.0," Team SMAUG-T, Korean Post-Quantum Cryptogr. (KpqC) Res. Group, Seoul, South Korea, Tech. Rep., 2024. available at https://kpqc.or.kr/competition_02.html
- [10] M. R. Albrecht, R. Player, S. Scott, "On the concrete hardness of learning with errors." Journal of Mathematical Cryptology 9.3 (2015): 169-203.
- [11] C. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C. J. Shih, B. Y. Yang, "NTT multiplication for NTT-unfriendly rings: New speed records for Saber and NTRU on Cortex-M4 and AVX2." IACR Transactions on Cryptographic Hardware and Embedded Systems (2021): 159-188.

<저자 소개>



곽 현 지 (Hyun Ji Kwag)

2024년 8월: 성신여자대학교 수학과 졸업
2024년 9월~현재: 고려대학교 정보보호학과 석사과정
<관심분야> 암호학, 정보보호



이 미 라 (Mira Lee)

학생회원
2024년 2월: 덕성여자대학교 사이버보안전공 졸업
2024년 3월~현재: 고려대학교 정보보호학과 석사과정
<관심분야> 암호, 프로토콜, 양자 내성 암호



김 종 현 (Jonghyun Kim)

정회원
2014년 2월: 성균관대학교 수학과 졸업
2024년 8월: 고려대학교 정보보호학과 석박사 통합과정 졸업
2024년 9월~현재: 고려대학교 정보보호학과 박사후 연구원
<관심분야> 암호 프로토콜, 양자 내성 암호



박 종 환 (Jong Hwan Park)

증신회원
1999년 2월: 고려대학교 수학과 졸업
2005년 2월: 고려대학교 정보보호학과 석사
2008년 8월: 고려대학교 정보보호학과 박사
2013년 9월~2019년 8월: 상명대학교 컴퓨터과학과 조교수
2019년 9월~2024년 8월: 상명대학교 컴퓨터과학과 부교수
2024년 9월~현재: 상명대학교 컴퓨터과학과 정교수
<관심분야> 함수 암호, 브로드캐스트 암호, 암호 프로토콜, 양자 내성 암호