외부 가속기를 이용한 Kyber 역캡슐화에 관한 연구

류지은, 강주성, 염용진*
국민대학교
{ofryuji, jskang, *salt}@kookmin.ac.kr

A Study on Kyber Decapsulation Using an External Accelerator

Jieun Ryu, Ju-sung Kang, Yongjin Yeom*
Kookmin Univ.

요 약

본 논문은 저사양 단말의 인증에 양자내성암호 Kyber을 사용할 때 외부 가속기로써 외부 단말을 활용하는 모델을 제시한다. 또한, Kyber을 이용한 인증 절차에서 저사양 단말이 수행하는 복호화(decapsulation) 연산 중 단말 외부로 누출되면 안되는 키가 입력으로 사용되는 함수인 NTT^{-1} 연산을 외부 단말에 요청하기 앞서 키를 인코딩하는 방식을 제안하고 실효성을 분석한다.

Ⅰ. 서론

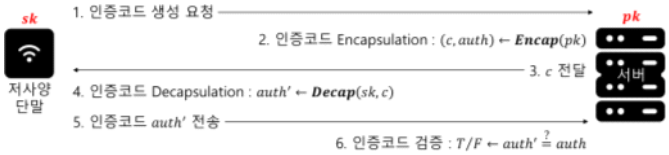
최근 양자컴퓨팅 기술이 발전함에 따라 인증 체계에 사용되는 기존 암호 알고리즘에 대한 양자내성암호(post-quantum cryptography, PQC)로의 전환이 이뤄지고 있다[1]. 이는 일반적인 PC 환경에 국한된 것이 아닌 IoT, 클라우드 등 다양한 환경을 고려하며, 특히 저사양 환경에 PQC를 적용하기 위한 최적화 구현 및 가속기 구현에 관한 연구가 다수 수행되었다. 하지만, 가속기를 적용할 수 없는 저사양 환경에서 PQC를 구동하는 것은 여전히 어렵다.

본 논문에서는 미국의 PQC 표준 알고리즘 중 저사양 단말 환경에 가장 적합한 Kyber를 대상으로 인증 연산 모델을 설정하고, 성능 문제를 해결하기 위해 외부 가속기를 활용하는 방식을 제안한다. 특히, Kyber 연산 중 가장 많은 연산량을 요구하는 NTT(Number Theoretic Transform)를 외부 가속기로 연산하는 방식을 제시하고 PC 환경에서의 구현을 통해 제안 방식의 실효성을 예측한다.

Ⅱ. 저사양 환경을 위한 NTT 연산의 분할

ⅰ. Kyber를 이용한 인증 방식

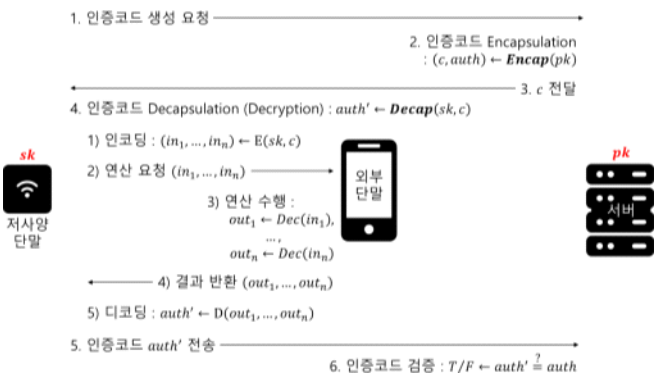
Kyber는 2022년 PQC 표준인 ML-KEM으로 선정된 키 설정(key establishment) 알고리즘이다[2]. 이는 키를 공유하기 위한 목적으로 개발되었으나, <그림 1>과 같은 방식으로 변형하여 단말 및 단말 소유자에 대한 사용자 인증에도 활용할 수 있다. 이때, Kyber의 캡슐화(encapsulation) 함수의 출력값 $auth$는 $Encap(pk)$의 내부 연산 중 랜덤하게 생성한 데이터에 대한 해시값(hash value)이며, 역캡슐화(decapsulation) 함수에 입력되는 sk가 pk에 대응되는 올바른 키인 경우 $Decap(sk,c)$의 출력값 $auth'$은 $auth$와 동일하다. sk와 pk는 사전 생성 및 공유되는 키로 반복 사용 가능하다.



<그림 1> Kyber를 활용한 저사양 단말의 사용자 인증 절차

저사양 단말에서 수행되는 역캡슐화 연산은 암호문 c를 복호화하는 Decryption과 검산을 위한 Encryption으로 구성된다. 이 중 Encryption은 생략하더라도 $auth'$을 구하는 데 영향을 미치지 않는다. 본 논문에서는 저사양 단말이 Encryption을 수행하기에 열악한 환경이라는 점을 고려하여 이를 생략한다.

한편, IC chip과 같은 저사양 단말은 외부 가속기로써 외부 단말의 연산 능력을 활용하여 계산량이 많은 연산을 가속 구현할 수 있다[3]. 이때 외부 가속기는 암호 경계(cryptographic boundary)에 속하지 않는 신뢰할 수 없는 영역으로 간주한다. 따라서 외부 단말이 저사양 단말 내부에 저장된 sk 및 기타 비밀 정보를 탈취할 수 없도록 막아야 하며, 이를 위해 저사양 단말은 외부 단말에 연산을 요청하기에 앞서 요청 데이터에 인코딩(encoding)을 적용해야 한다. <그림 2>는 외부 단말이 추가된 저사양 단말의 사용자 인증 절차이다.



<그림 2> 외부 단말이 추가된 저사양 단말의 사용자 인증 절차

ⅱ. Kyber 복호화 중 외부 가속기 활용 가능성 분석

Kyber 역캡슐화 절차는 <알고리즘 1>과 같다.

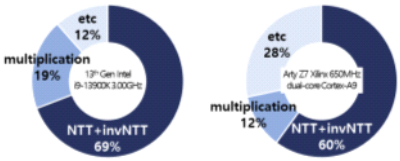
<알고리즘 1> 역캡슐화 $Decap(sk,c)$

1. $\hat{s}, h \leftarrow sk$ // $sk = \hat{s} \| h$
2. $u', v' \leftarrow c$ // $c = u' \| v'$
3. $\hat{u} \leftarrow NTT(u')$
4. $w' \leftarrow NTT^{-1}(\hat{s}^T \circ \hat{u})$
5. $w \leftarrow v' - w'$
6. $m' \leftarrow Encode(Compress(w))$
7. $auth' \leftarrow Hash(m' \| h)$
8. return $auth'$

우선 단계 1은 누출되면 안되는 sk에 관한 연산이므로 외부 단말에 연산을 요청하면 안 되는 단계이다. 단, 저사양 단말에 sk를 저장해둔 후 반복 사용할 수 있으므로 단계 1은 연산 속도에 영향이 없다. 단계 2와 3은 c가 공개된 값이므로 외부 단말에서 저사양 단말과 관계없이 수행 가능하다.

단계 4에서 NTT^{-1}의 입력은 단계 1에서 저사양 단말에 저장된 키 sk로부터 유도된 $\hat{s}$를 포함한다. 이는 외부로 유출되면 안 되는 비밀 값이므로 외부 단말에 연산을 요청하기 앞서 반드시 인코딩되어야 한다. 또한, 단계 4 이후 수행되는 연산의 값이 누출되면 공개된 값인 u'과 단계 4~6 함수의 선형성에 의해 $\hat{s}$이 유도되므로 해당 단계의 값에 대한 연산을 외부 단말에 요청할 경우, 모든 변수에 대한 인코딩을 적용해야 한다.

<그림 3>과 같이, 저사양 단말이 수행하는 역캡슐화 연산에서 가장 연산량이 많은 함수는 NTT와 NTT^{-1}이다. 따라서 해당 연산을 외부 단말에 요청하여 처리할 수 있다면 연산 속도가 크게 향상될 것을 기대할 수 있다.



<그림 3> 역캡슐화의 Decryption 구성 함수별 연산 소요 시간 비율

Ⅲ. 인코딩된 NTT^{-1} 연산 방식에 대한 분석

저사양 단말의 경우, 외부 단말에 함수 연산을 요청할 때 감소하는 인증 연산 시간이 통신량 증가로 느려지는 인증 시간보다 짧아 역캡슐화 성능을 저해하는 문제가 발생할 수 있다. 덧셈, 뺄셈 및 비트 연산은 위와 같은 문제를 촉발하므로 외부 단말에 연산 요청을 하지 않고 저사양 단말 내부에서 연산해야 한다. 따라서 <알고리즘 1>에서 외부 단말에 요청할 연산은 단계 2~4이다.

ⅰ. 안전성 분석

본 절에서는 NTT^{-1}에 대한 인코딩 방식을 제안하고, 전수조사 공격 측면에서 대략적인 안전성을 제시한다. 앞서 언급한 것과 같이 단계 4는 외부 단말에 연산을 요청하기 앞서 입력에 대한 인코딩이 요구된다. $\hat{u}$는 공개된 값을 사용하여 결정론적(deterministic) 연산을 수행한 결과이므로 인코딩하지 않는다.

NTT^{-1}는 선형 연산이므로 입력 $x = x_1 + \cdots + x_n$와 상수 $a_1, \ldots, a_n$에 대하여 다음 식이 성립한다.

$$NTT^{-1}(x) = \frac{1}{a_1} NTT^{-1}(a_1 x_1) + \cdots + \frac{1}{a_n} NTT^{-1}(a_n x_n)$$

따라서 $\hat{s}$을 다음과 같이 $(x_1, \ldots, x_{n+l})$로 인코딩한다. 이때, n과 l은 주어진 범위 내에서 랜덤(random)하게 선택한다.

$$(\hat{s}_1, \ldots, \hat{s}_{n-1}) \xleftarrow{\$} R_q, \quad \hat{s}_n = \hat{s} - \sum_{i=1}^{n-1} \hat{s}_i,$$

$$(a_1, \ldots, a_n) \xleftarrow{\$} Z, \quad (\hat{d}_1, \ldots, \hat{d}_l) \xleftarrow{\$} R_q$$

$$(x_1, \ldots, x_{n+l}) \leftarrow Shuffle(a_1 \hat{s}_1, \ldots, a_n \hat{s}_n, \hat{d}_1, \ldots, \hat{d}_l)$$

이때, R_q는 입력의 Kyber가 정의되는 다항식 환(polynomial ring)이며, $\xleftarrow{\$}$는 랜덤하게 선택함을 의미한다. $d_i\,(i = 1, \ldots, l)$는 공격자의 연산량을 증가시키기 위하여 추가된 더미(dummy) 데이터이다.

위 인코딩 기법에 대하여 $\hat{s}$를 복구하길 바라는 공격자의 전수조사 공격 연산량은 다음과 같다. 상수 $a_1, \ldots, a_n$는 공격자가 시도해야 하는 NTT^{-1} 연산량을 증가시킨다. 공격자는 $\{a_i \hat{s}_i\}_{i=1,\ldots,n}$으로부터 $\sum_{i=1}^{n} \hat{s}_i$를 구해야 하는데, $a_i \in Z$가 $\hat{s}_i \in R_q$와 무관하게 선택되므로 공격자는 모든 a_i에 대하여 $\frac{1}{a_i} NTT^{-1}(a_i \hat{s}_i)$를 시도해야 한다. 가능한 a_i의 수가 j라고 할 때, 공격자가 $\{a_i \hat{s}_i\}_{i=1,\ldots,n}$에 대하여 시도해야 하는 NTT^{-1} 연산량은 $O(j^n)$이다.

또한, 공격자는 n과 l을 알 수 없기 때문에 $\hat{s}$를 구성하는 x_i를 구별할 수 없고 $n+l$개의 x_i에 대한 모든 경우를 시도해야 한다. 따라서 제안한 인코딩에 대한 NTT^{-1} 전수조사 연산량은 다음과 같다.

$$O(j^{n+l})$$

128-bit 안전성을 만족하는 파라미터 쌍 (j, n, l)의 예시로 통신량을 줄여야 할 경우 관련 파라미터 n, j의 크기를 줄인 $(256, 14, 2)$와 같은 파라미터를 선택할 수 있다. 또한, 인코딩 및 디코딩을 테이블 참조로 사용하는데 테이블을 저장할 메모리가 부족한 경우 $(2, 96, 32)$와 같은 파라미터를 선택할 수도 있다.

ⅱ. 구현 및 성능 분석

제안한 방식은 $\hat{s}$를 분할하고, 반환받은 NTT^{-1} 연산 결과를 디코딩 하는 추가 연산이 발생한다. 단, 이는 직접 NTT^{-1}를 수행하는 것보다 훨씬 적은 연산 시간을 소요할 것으로 예상 가능하다.

또한 Kyber 구현에 필요한 다양한 내부 함수는 반환하는 값의 범위에 조금씩 차이가 있으므로, 제안한 인코딩 기법을 구현에 적용할 때 오버플로우(overflow)를 방지하려면 인코딩과 디코딩의 매 단계마다 범위 조정을 위한 reduction 함수를 적용해 주어야 한다. 이때, Kyber의 reduction은 덧셈, 뺄셈 및 비트 연산만으로 수행 가능한 함수이므로, 인증 연산에 부하가 크지 않을 것으로 예상한다.

<표 1>은 제안한 인코딩 및 디코딩을 수행함에 따라 NTT^{-1} 연산에 대하여 저사양 단말과 외부 가속기 각각에 추가되는 연산 별 소요 시간을 정리한 것이다. 속도 측정은 127차 다항식 $\hat{s}$, $\hat{u}$를 기준으로 $n = 2, l = 0$이라고 설정하고 진행했다.

저사양 단말				외부 가속기	
$\hat{s}$ 분할	a_i 곱셈	a_i^{-1} 곱셈	$\sum_{i=1}^{n} NTT^{-1}$	$\hat{s}_i \cdot \hat{u}$	NTT^{-1}
35.98	8.33	3.61	4.19	58.23	97.79
52.11				156.02	

<표 1> 인코딩 적용에 따라 추가된 연산의 소요 시간 (ns)
@13th Gen Intel i9-13900K 3.00GHz

NTT^{-1}를 외부 가속기를 활용하여 연산함에 따라 저사양 단말에 추가되는 연산의 소요 시간은 외부 가속기에서 수행하는 연산 소요 시간의 약 1/3로, 실제 표준 Kyber 파라미터를 적용하면 그 차이가 증가할 것으로 기대된다.

실제 구현 환경에서는 통신량 증가에 따른 성능 저하를 고려하여 n과 l을 조정할 수 있으며, 이는 후속 연구를 통해 구체화할 예정이다.

Ⅳ. 결론

본 논문에서는 저사양 단말 환경에서 Kyber로 인증 연산을 수행하기 위해 외부 가속기를 활용하는 모델을 제시하고, 저사양 환경에서 수행되는 역캡슐화 연산의 구현을 위해 NTT를 외부 가속기에서 sk 노출 없이 실행하는 방식을 제안하였다. 외부 가속기를 사용할 경우, 역캡슐화의 NTT^{-1} 연산 소요 시간이 1/3 이상 줄어들 것으로 예상되므로 해당 모델을 활용하면 IC chip이나 IoT 장비 등의 저사양 단말에서 PQC 인증을 빠르게 수행할 수 있을 것으로 기대된다. 향후 연구에서는 제안한 모델에 대한 구체적인 안전성을 분석하고 인코딩 파라미터 최적화를 진행하고자 한다.

ACKNOWLEDGMENT

이 논문은 서울시 산학연 협력사업 2025년도 양자 기술개발 지원사업 (QR250002, 양자내성암호 Kyber를 이용한 스마트카드 인증기술 개발)의 지원을 받아 수행된 연구임

참 고 문 헌

[1] Barker W., Polk W. and Souppaya M., “Getting Ready for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic Algorithms,” NIST Cybersecurity White Paper(CSWP) 15, 2021, (https://doi.org/10.6028/NIST.CSWP.15)

[2] National Institute of Standards and Technology (2024) Module-Lattice-Based KeyEncapsulation Mechanism Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 203. https://doi.org/10.6028/NIST.FIPS.203

[3] 류지은, 김덕상, 강주성, 염용진, “저사양 스마트카드의 PQC 인증 방안에 관한 연구”. 한국통신학회 학술대회논문집, 경북, 2025-11-19