Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.1, Feb. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.1.75

NTRU를 결합한 하이브리드 세션 보호 프로토콜을 이용한 금융 오픈 API 환경의 거래 세션 안전성 강화*

권 수 진,[1†] 김 덕 상,[2] 박 영 재,[1] 류 지 은,[1] 강 주 성,[3] 염 용 진[3‡]
[1,3]국민대학교 (대학원생, 교수), [2]에잇바이트 (대표)

Enhancing Security of Transaction Session in Financial Open API Environment Using Hybrid Session Protection Protocol Combined with NTRU*

Sujin Kwon,[1†] Deoksang Kim,[2] Yeongjae Park,[1]
Jieun Ryu,[1] Ju-Sung Kang,[3] Yongjin Yeom[3‡]
[1,3]Kookmin University (Graduate student, Professor), [2]8byte (CEO)

요 약

현재 금융거래 서비스에서 보편적으로 사용하는 RSA와 ECC 같은 공개키 암호 알고리즘은 양자 컴퓨터가 실현되면 더 이상 안전성을 보장할 수 없으므로 기존 레거시 알고리즘을 양자내성암호로 전환해야 한다. 하지만 다양한 서비스에 사용 중인 알고리즘을 교체하는 데에는 상당한 시간이 소요될 것으로 예상된다. 다가올 전환기를 대비하기 위하여 두 알고리즘을 결합하는 하이브리드 방식에 관한 연구가 필요하다. 본 논문에서는 레거시 알고리즘인 ECDH 알고리즘과 양자내성암호 알고리즘인 NTRU 알고리즘을 결합하여 세션키를 생성하는 하이브리드 세션키 교환 프로토콜을 제안한다. TLS 1.3 기반 하이브리드 키 교환을 위해 IETF에서 제안한 방식들을 적용해본 결과 기존 금융거래 세션 보호 솔루션에 우리가 제안한 프로토콜을 사용하면 안전성을 강화할 수 있을 것으로 기대된다.

ABSTRACT

Public key cryptography algorithm such as RSA and ECC, which are commonly used in current financial transaction services, can no longer guarantee security when quantum computers are realized. Therefore it is necessary to convert existing legacy algorithms to Post-Quantum Cryptography, but it is expected that will take a considerable amount of time to replace them. Hence, it is necessary to study a hybrid method combining the two algorithms in order to prepare the forthcoming transition period. In this paper we propose a hybrid session key exchange protocol that generates a session key by combining the legacy algorithm ECDH and the Post-Quantum Cryptographic algorithm NTRU. We tried the methods that proposed by the IETF for TLS 1.3 based hybrid key exchange, and as a result, it is expected that the security can be enhanced by applying the protocol proposed in this paper to the existing financial transaction session protection solution.

Keywords: electronic financial transaction, PQC, hybrid key combination, HKDF

I. 서 론

금융거래 서비스는 무결성(integrity), 인증(authentication), 부인방지(non-repudiation)를 제공하기 위해 공개키 기반 구조(public key infrastructure, PKI)를 구축한다. 현재 PKI에

Received(12. 06. 2022), Modified(01. 17. 2023), Accepted(01. 17. 2023)
* 본 연구는 한국연구재단-기후변화대응기술개발 사업(No.2021M1A2A2043893)의 지원으로 수행했습니다.
† 주저자, tnwls1595@kookmin.ac.kr
‡ 교신저자, salt@kookmin.ac.kr(Corresponding author)

널리 사용되는 공개키 암호로는 RSA와 ECC가 있으며, 이들은 수학적 문제에 기반하여 안전성을 보장한다. 그러나 1994년 발표된 쇼어(Shor) 알고리즘에 의하면 양자 컴퓨터(quantum computer)가 상용화될 경우 RSA나 ECC 같은 현재 주로 사용되는 공개키 암호 시스템의 안전성이 더 이상 보장되지 않는다[1].

미국 국립표준기술연구소(NIST)는 2016년부터 양자 컴퓨터에도 안전한 양자내성암호(post-quantum cryptography, PQC) 공모사업을 진행하고 있다[2]. 하지만 아직까지 양자내성암호의 이론 및 실증적 안전성 검증에 어려움이 있어, 현재 보편적으로 사용 중인 공개키 암호를 양자내성암호로 전환하는 것은 간단하지 않다[3]. 따라서 현 상황에서 FIPS 인증과 양자 컴퓨터에 대한 안전성을 보장하기 위해 기존에 사용되고 있는 레거시(legacy) 알고리즘과 양자내성암호 알고리즘을 결합한 하이브리드(hybrid) 방식을 도입할 필요가 있다.

TLS(Transport Layer Security)와 SSL(Secure Sockets Layer)상에도 하이브리드 키 교환 방식을 통해 양자내성암호를 적용하는 연구가 현재 활발히 진행 중이다[4-9]. 하지만, 기존 연구들은 하이브리드 키 교환 성능을 레거시 알고리즘만을 사용했을 때와 비교 분석하는 방식을 중심으로 진행되고 있어 하이브리드 결합 방식에 따른 성능 분석 연구가 필요하다.

본 논문에서는 금융거래 서비스에서 E2E(End-to-End) 암호화 안전성을 강화하기 위한 하이브리드 키 교환 방식으로 타원 곡선 기반 키 교환 방식 ECDH와 양자내성암호의 KEM(Key Encapsulation Mechanism)의 결합을 제안한다. KEM 알고리즘 중 암·복호화 속도가 빠르고 키와 암호문의 크기가 작은 격자 기반 암호는 효율성이 중요한 금융 서비스에 적합하다. NTRU는 격자 기반 암호로 NIST PQC 공모사업 3라운드 후보 알고리즘이었다. 다른 격자 기반 암호에 비해 속도는 느리지만 가장 오랜 기간 안전성을 검증받은 알고리즘이므로, 본 논문에서 타겟 알고리즘으로 선정한다.

본 논문은 2장에서 금융거래 서비스, 양자내성암호 및 하이브리드 키 교환에 관한 개요를 설명하고, 3장에서 현재 저축은행중앙회에 사용되는 E2E 암호화 솔루션과 하이브리드 방식을 사용하여 양자내성암호를 적용하는 새로운 프로토콜을 제안한다. 4장에서는 구현 결과를 통해 하이브리드 키 교환 방식의 성능을 평가한다. 마지막으로 5장에서 결론을 정리한다.

II. 연구배경

2.1 금융거래 서비스

전자금융거래란 금융회사 또는 전자금융업자가 전자적 장치를 통하여 금융상품 및 서비스를 제공하고, 이용자가 금융회사 또는 전자금융업자와 직접 대면하거나 의사소통하지 않고 자동화된 방식으로 이를 이용하는 거래를 의미한다[10]. 전자금융거래 시장 규모는 꾸준히 확대되고 있다. 2021년 전자금융거래 이체 기준 총액은 약 40,929조 원으로 2020년 대비 약 4398조 원 증가했고, 2017년 대비 약 15,007조 원 증가했다[11].

전자금융 서비스의 안전한 이용을 위해 데이터 보안 서비스 제공은 필수적이다. 금융위원회는 핀테크 인프라 구축을 위해 지속적인 API 개방 정책(오픈 API)을 추진하여 금융회사가 핀테크 기업과 다양한 핀테크 서비스를 출시할 수 있도록 통로를 제시했다[12]. 디지털 금융으로의 전환과 핀테크 산업 고도화같은 환경변화에 따라 오픈 API의 중요성이 증가하고 있으며, 현재 16개 은행의 일부 지급결제망과 데이터가 오픈 API로 제공된다[13]. 따라서, 오픈 API에 사용자를 인증하고 금융거래정보를 보호하기 위한 E2E 암호화가 필요하며, 금융 보안원은 이에 대한 주요 보안 요구사항을 제시했다[14].

E2E 암호화는 일반적으로 TLS 기반으로 이뤄지며 최소한 BEAST(Browser Exploit Against SSL/TLS) 공격에 방어할 수 있는 버전 1.2 이상 또는 POODLE, SLOTH, DROWN 공격을 방어할 수 있는 버전 1.3을 사용하도록 권장된다[15]. Fig. 1.은 TLS 1.3 프로토콜의 통신 과정을 나타낸 것이다[16]. E2E 암호화는 클라이언트(client)와 서버(server)가 주고받는 데이터를 암호화하는 방법으로 대화 당사자만 암호화된 데이터를 복호화할 수 있다.

현재 저축은행중앙회는 통합 회원사 67개를 대상으로 오픈 API 서비스를 제공하며, 저축은행중앙회에서 제공하는 오픈 API 서비스의 구성도는 Fig. 2.와 같다[17]. 회원사는 오픈 API 게이트웨이(gateway)를 통해 저축은행중앙회 업무 시스템에 연계하여 로그인, 계좌조회, 거래명세조회, 예·적금

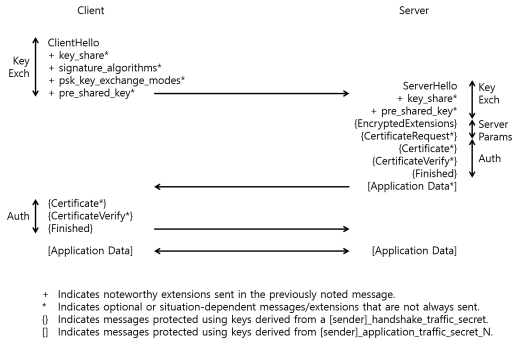

Fig. 1. TLS 1.3 Protocol [16]

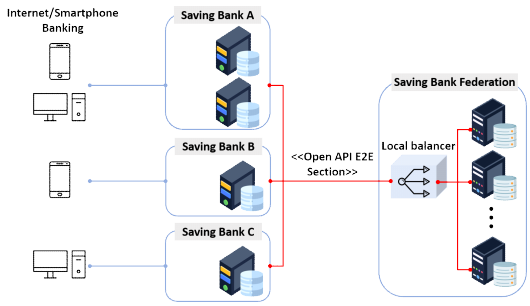

Fig. 2. Open API system configuration diagram of the Korea Savings Bank Federation [17]

가입/해지 등 200여 개 업무를 처리하고 있다. 해당 오픈 API 서비스는 기본적으로 TLS와 SSL 프로토콜 상에서 이뤄지고 있으며, 이외에도 오픈 API 구간 보안성 강화를 위해 E2E 암호화 적용과 이중 암호화 기능을 적용하여 서비스를 운영한다. 본 논문에서는 양자내성암호를 적용하여 저축은행중앙회에서 사용되는 오픈 API 서비스의 보안성을 강화하는 방향으로 연구를 진행한다.

2.2 양자내성암호

기존 컴퓨터(conventional computer)에서 다항식 시간 내에 풀 수 없는 문제를 해결하기 위해 양자 현상을 이용하는 양자 컴퓨터에 관한 연구가 활발히 이뤄지고 있다[18,19]. 1994년 이상적인 양자 컴퓨터를 사용하면 다항식 시간 내에 소인수분해와 이산 로그 문제가 해결 가능함이 증명됐다[1]. 따라서, 각 문제의 어려움에 기반을 둔 RSA와 ECC 및 현재 널리 사용되고 있는 공개키 암호는 양자 컴퓨터 상의 공격에 안전하지 않다.

Mosca는 양자 컴퓨팅 공격에 대한 대비 시간에

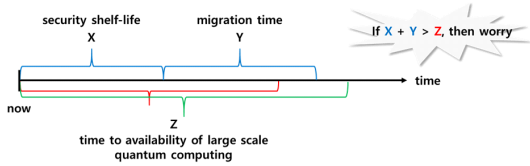

Fig. 3. Mosca's Theory: Time to prepare for quantum computing attacks [20]

관한 이론을 제시했다[20]. 이는 Fig. 3.과 같이 데이터를 보호해야 하는 시간과 레거시 알고리즘을 양자내성암호로 전환하기 위해 걸리는 시간의 합보다 양자 컴퓨터 개발 속도가 더 빠른 경우, 양자 컴퓨터 공격으로부터 안전하지 않음을 의미한다.

이에 따라 NIST는 양자 컴퓨팅 공격에도 안전한 공개키 암호체계 및 전자서명 알고리즘의 표준화를 위하여 2016년부터 양자내성암호 공사업을 진행하고 있다[2]. 공모를 통해 제안된 양자내성암호의 수학적 기반문제는 격자 기반(lattice-based), 부호 기반(code-based), 다변수 기반(multivariate-based), 아이소제니 기반(isogeny-based) 등으로, NP-hard 문제에 속한다. 2022년 7월 기준으로 4개의 표준 알고리즘이 선정되었고, 현재 4라운드가 진행 중이다[21]. 본 절에서는 양자내성암호 NTRU가 선정된 3라운드를 중심으로 확인한다. Table 1.은 양자내성암호 공모사업의 3라운드 후보 알고리즘으로 격자 기반 암호가 다수를 차지한다[22]. 안전성 수준(security category)이 동일하도록 파라미터 값을 설정한 각 후보 알고리즘의 키생성, 암·복호화 속도 합과 암호

Table 1. Third-round candidate algorithm for NIST PQC Standardization Conference

	Mathematical Problem	Algorithm	Feature
PKE/KEM	Lattice-based Cryptography	NTRU	NTRU
		CRYSTALS KYBER	LWE
		SABER	LWR
	Code-based Cryptography	Classic McEliece	Goppa
Digital Signature	Lattice-based Cryptography	CRYSTALS DILITHIUM	LWE
		FALCON	NTRU
	Multivariate-based Cryptography	Rainbow	UOV

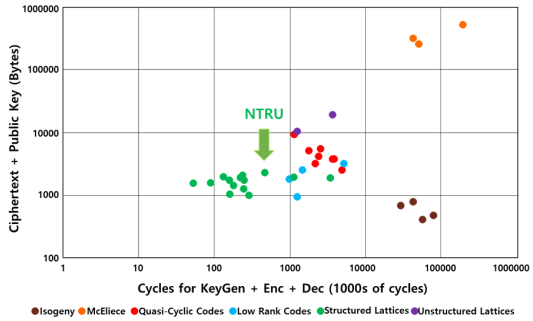

Fig. 4. The speed and size of the PQC public offering candidate Algorithm [23]

문 크기, 공개키 크기의 합은 Fig. 4.와 같다[23]. 동일한 안전성 수준일 때, 격자 기반 암호는 다른 후보군에 비해 속도가 빠르고 크기가 작으므로 높은 성능을 보인다.

3라운드 PKE/KEM 후보 알고리즘 중 격자 기반 암호인 NTRU는 다른 격자 알고리즘과 비교할 때 속도는 느리지만, 가장 오랜 기간 안전성을 검증받았다[24]. 또한, NTRU를 사용할 수 있는 환경에서는 다른 격자 기반 알고리즘도 적용할 수 있으므로, NTRU의 적용 가능성에 대한 분석은 매우 유용하다. 따라서 본 논문은 NTRU를 타겟 암호 알고리즘으로 설정한다.

Fig. 5.는 NTRU-HPS KEM 동작 과정을 나타낸다. 먼저, Alice가 개인키와 공개키 쌍을 생성한다. 그 뒤, Bob은 해시함수(hash function)를 이용하여 공유 비밀(shared secret) k를 생성하고, Alice의 공개키를 받아 해시함수에 입력한 값을 암호화하여 암호문 ct를 생성한다. ct를 Alice에게 전달하면 개인키를 가진 Alice는 ct를 복호화하여 나온 값에 해시함수를 적용하여 k를 생성한다. 이를 통해 Alice와 Bob은 동일한 공유 비밀 k를 가지게

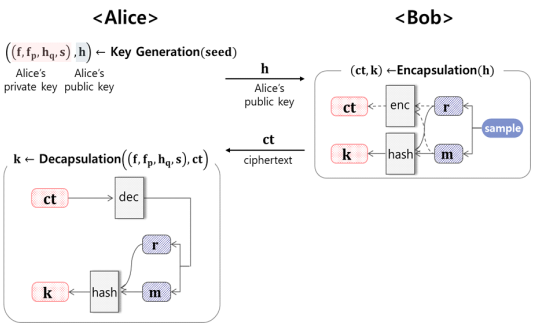

Fig. 5. Operation process of NTRU-HPS KEM

된다[25, 26].

2.3 하이브리드 키 교환

양자 컴퓨팅 상에서의 공격에도 안전성을 보장하기 위해 양자내성암호로의 전환은 불가피하다. 그러나, 현재 다양한 영역에 사용되는 공개키 암호 알고리즘을 바로 양자내성암호로 교체하여 사용하기는 어렵다[3]. 따라서, 레거시 알고리즘과 양자내성암호 알고리즘을 결합한 하이브리드 키 교환 방식을 도입하면 FIPS 인증과 양자 컴퓨터에 대한 안전성을 보장할 수 있다.

하이브리드 키 교환의 목표는 구성(component) 알고리즘 N개로부터 생성된 암호키 N개 중 하나 이상의 암호키가 안전하다면, 이들을 조합하여 만든 하이브리드 키 역시 안전함을 보장하는 것이다. 하이브리드 키 교환을 위해 고려할 사항으로는 각 알고리즘이 생성한 공개키의 전송 방식과 세션키 결합 방식 등이 있다[27]. 레거시 알고리즘과 양자내성암호의 공개키는 수용할 수 있는 패킷(packet) 크기에 따라 각각 전송하거나 연접(concatenate)하여 한 번에 전송할 수 있다.

Fig. 6.은 연접을 통한 하이브리드 모델의 두 가지 방식을 나타낸 것이다[28]. 첫 번째 방식은 각 알고리즘으로 생성한 값을 키유도함수(key derivation function, KDF)에 입력하여 그 출력을 공유 비밀로 사용하는 방식이다. 또 다른 방식은 레거시 알고리즘으로 생성한 값을 양자내성암호 알고리즘에 입력하여 공유 비밀을 생성하는 방식이다.

NIST와 국제 인터넷 표준화 기구(IETF)는 첫 번째 방식을 제안하고 있다[27, 29]. IETF는 가장 널리 쓰이는 암호화 프로토콜인 TLS 프로토콜 상의 하이브리드 키 교환에 관하여 인터넷 표준화 작업을 진행 중이다. 여기서 공유 비밀인 세션키를 결합하는 방식으로는 연접, XOR(exclusive or) 연산, KDF를 이용하는 방식이 제안되었으며, 각 연산은 Table 2.와 같다[27].

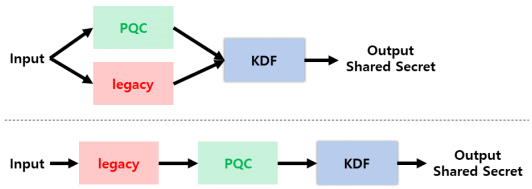

Fig. 6. Hybrid model[27]

Table. 2 Hybrid key exchange method proposed by IETF [27]

Method	Summary
Concatenation	Concatenate the two session keys ss_1 and ss_2 to generate the final session key ss $ss = ss_1 \| ss_2$
XOR	Do XOR operation of the two session keys ss_1 and ss_2 to generate the final session key ss $ss = ss_1 \oplus ss_2$
KDF-1	Put two session keys ss_1 and ss_2 into the KDF to generate the final session key ss $ss = KDF(ss_1, ss_2)$
KDF-2	Concatenate two session keys ss_1, ss_2 and input them into the KDF to generate the final session key ss $ss = KDF(0, ss_1 \| ss_2)$
KDF-3	Put two session keys ss_1 and ss_2 sequentially into the KDF to generate the final session keys ss $ss = KDF(KDF(prk, ss_1), ss_2)$

KDF는 암호키가 높은 엔트로피를 갖도록 만드는데 사용되며, 본 논문에서는 TLS 1.3 키 스케줄(key schedule)에 사용되는 HMAC 기반 KDF인 HKDF(HMAC-based KDF)를 중점적으로 분석한다. HKDF는 Fig. 7.과 같이 HKDF-Extract 단계에서 엔트로피가 높은 의사난수 키를 생성하고 HKDF-Expand 단계에서 원하는 길이로 키를 확장한다[30].

HKDF-Extract 단계는 선택적 입력인 솔트(salt)와 IKM(Input Keying Material)를 입력하여 의사난수 키 PRK(Pseudo-Random Key)를 출력한다. 이를 식으로 표현하면 다음과 같다.

$$PRK \leftarrow HKDF-Extract(salt, IKM) \quad (1)$$

HKDF-Expand 단계는 엔트로피가 높은 PRK와 선택적 입력인 텍스트 info를 입력하여 원하는 길이(L)의 암호키 OKM(Output Keying Material)을 출력한다. 이를 식으로 표현하면 다음과 같다.

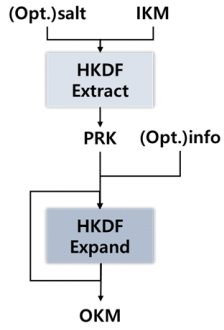

Fig. 7. HMAC-based Key Derivation Function [30]

$$OKM \leftarrow HKDF-Expand(PRK, info, L) \quad (2)$$

III. 제안하는 프로토콜

3.1 기존 프로토콜[31]

오픈 API 서비스는 다수의 회원사가 저축은행중앙회에 오픈 API 게이트웨이로 접속을 요청함으로써 업무를 처리한다. 따라서, 요청에 대한 시스템 부하를 줄이고 가용성을 향상할 필요가 있다. 또한, 이는 보안을 위해 안전성이 훼손되지 않는 범위 내에서 허용한다.

Fig. 8.은 현재 저축은행중앙회 오픈 API에서 사용하는 상용화된 E2E 프로토콜이다. 해당 그림에서 교환하는 평문(pt_1, pt_2)은 이체, 회원정보, OTP, 인증, 고객정보관리 등 총 200여개가 넘는 데이터이다. 교환하는 데이터의 오버헤드(overhead)를 줄이기 위해 secp256k1을 이용한 타원곡선 암호를 사용하며, 공개키 전달 시 타원 곡선 점의 X 좌표만 전달함으로써 오버헤드가 32바이트(byte)만 발생하

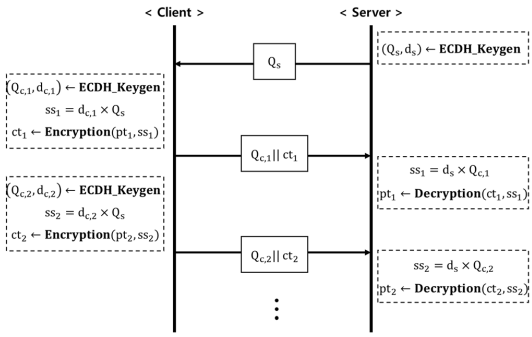

Fig. 8. E2E Encryption protocol commercialized by the Korea Savings Bank Federation [31]

도록 한다.

기존 TLS 방식과 같이 암호키를 미리 교환하여 세션(session) 당 하나의 암호키를 사용하지 않고, 거래가 발생할 때마다 키 쌍(공개키, 개인키)을 생성해 세션 암호키(S)를 생성한다. 그 뒤 클라이언트의 공개키를 전달하여 순방향 비밀성(forward secrecy)을 구현한다. 서버의 키 쌍(공개키, 개인키)은 24시간 주기로 생성함으로써 키의 라이프 타임(life time)을 최소화한다.

3.2 양자내성암호를 적용한 하이브리드 방식 제안

본 논문은 현재 금융거래 시스템의 안전성 강화를 위해 오픈 API 서비스에 Fig. 9.와 같이 하이브리드 키 교환 방식을 적용한다[32]. 서버가 키 쌍을 생성하여 클라이언트에게 전송하는 과정은 기존 프로토콜과 동일하다. 서버는 ECDH와 추가로 NTRU에 대한 키 쌍(공개키, 개인키)을 24시간 주기로 생성하고 두 공개키를 연접하여 클라이언트에게 한 번만 전송한다. 클라이언트는 거래가 발생할 때마다 ECDH 키 쌍을 생성하고 서버로부터 받은 NTRU 키 쌍에 대한 세션키를 생성한다. 이때 하이브리드 방식의 구성 알고리즘으로 ECDH는 기존 프로토콜의 알고리즘을 사용하고, NTRU는 보안수준 1을 만족하는 NTRU-HPS-2048509를 사용한다.

한편, 양자내성암호 공모전 3라운드 종료 이후 NIST는 NTRU 사용을 권장하지 않는다. 본 논문은 하이브리드 결합 방식에 양자내성암호의 적용 가능성을 보여주기 위한 예시로 NTRU를 타겟 암호 알고리즘으로 선정하였으며, NIST 표준으로 선정된 KYBER512 또한 동일한 방식으로 적용 가능하다.

생성한 ECDH 키와 NTRU 세션키를 하이브리드 방식으로 결합한 뒤 데이터를 하이브리드 키로 암호화하여 ECDH 공개키와 NTRU 암호문에 연접하여 서버에 전송한다. 기존 서비스는 8K 정도의 패킷 크기를 제공하므로 한 패킷에 ECDH와 NTRU의 공개키와 암호문을 수용할 수 있다. 따라서, 공개키 전송방식으로 이들을 연접하여 한 번에 전송하는 방식을 채택한다.

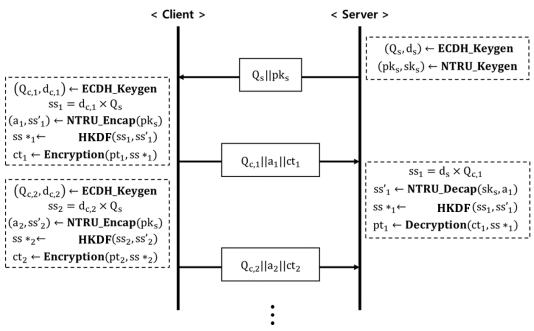

Fig. 9. Hybrid method combined with PQC [32]

본 논문에서는 [27]에서 제안한 하이브리드 키 결합 방식을 구현하여 분석한다. 그중 연접 방식은 생성한 최종 세션키 크기가 데이터 암호화를 위한 암호키 크기와 맞지 않는다. 또한, XOR 연산으로 유도한 세션키는 랜덤함수족과 구별 가능하므로 안전하지 않을 수 있다[33]. 이러한 이유로 연접 방식과 XOR 방식을 제외한 3가지 방식을 선택한다.

3.2.1 KDF - 1

ECDH로 생성한 공유 비밀 ss1을 HKDF의 IKM으로 사용하고, NTRU로 생성된 공유 비밀 ss2를 HKDF의 솔트로 사용한다. 이는 Fig. 10.과 같다.

Fig. 10. Using generated shared secret value as IKM and salt of HKDF

3.2.2 KDF - 2

ECDH로 생성된 공유 비밀 ss1와 NTRU로 생성된 공유 비밀 ss2를 연접하여 HKDF의 IKM으로 사용한다. 이는 Fig. 11.과 같다.

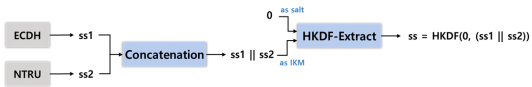

Fig. 11. Concatenate the generated shared secret value and use it as IKM

3.2.3 KDF - 3

먼저, Qs를 HKDF에 입력하여 PRK를 생성한다. 이를 HKDF의 솔트로 사용하고 ECDH로 생성된 공유 비밀 ss1을 HKDF의 IKM으로 사용하여

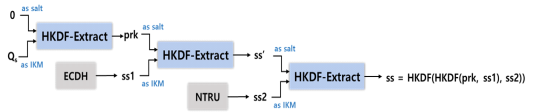

Fig. 12. Use the generated shared secret values as IKM of HKDF in a chain

ss′을 생성한다. 생성한 ss′를 다음 HKDF의 솔트로 사용하고, NTRU로 생성된 공유 비밀 ss2를 HKDF의 IKM으로 사용하여 최종 세션키 ss를 계산한다. 이는 Fig. 12.와 같다.

IV. 구현 및 결과

4.1 하이브리드 키 교환 실험 환경

하이브리드 키 교환 실험은 Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz에서 진행하였다. 사용한 키 교환 알고리즘은 타원 곡선 암호 기반 공개키 암호 ECDH와 격자 기반 양자내성 암호 NTRU-hps2048509이다. ECDH는 OpenSSL에서 제공되는 소스 코드를 사용하였으며[34], NTRU는 NIST PQC 공모사업 3라운드에 제출된 최적화된(optimized) 구현 코드를 사용하였다[25]. KDF는 RFC 5869[30]에 기반하여 HKDF를 직접 C언어로 구현하였으며, Visual Studio 2019에서 실험을 진행하였다. 실험에 사용한 키 교환 알고리즘과 파라미터를 요약하면 Table 3.과 같다.

Table 3. Key exchange algorithm and parameters used in the experiment

Algorithm	Mathematical Problem	(data unit: byte)			
		pk	sk	ct	ss
ECDH NIST P-256	Discrete Logarithm	65	32	-	32
NTRU-hps 2048509	NTRU	699	935	699	32

4.2 하이브리드 키 교환 실험 결과 및 금융 API로의 적용 가능성 분석

진행한 3가지의 키 결합 방식(KDF-1, KDF-2, KDF-3)에 대한 속도 측정 결과는 Fig. 13.과 같다. 측정한 결과는 키 결합 과정을 10,000번 실행하였을 때의 결과이며, 실험 결과의 편향을 줄이기 위해 동일한 실험을 10번 진행하여 평균을 계산하였다.

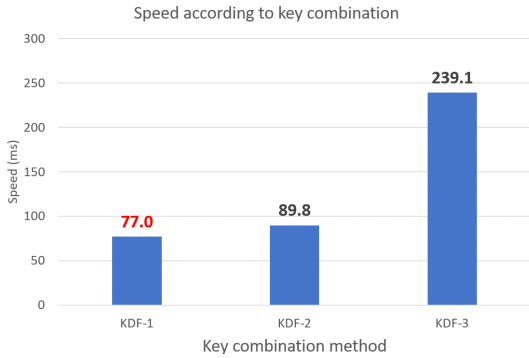

Fig. 13. 10,000 execution speed according to hybrid key combination method(㎳)

KDF-1과 KDF-2 실험에서 해시함수 SHA2-256는 각각 2번 호출되며, KDF-3 실험은 6번 호출된다. 동일한 횟수의 해시함수 호출임에도 불구하고 KDF-2가 KDF-1보다 느린 이유는 해시함수 내부 동작에서 다이제스트(digest) 연산이 한 번 더 진행되기 때문이다.

제안하는 프로토콜에 대해 저축은행중앙회 오픈 API 서비스에서의 적용 가능성을 분석하고자, 진행한 실험에 대하여 저축은행중앙회에서 지원하는 언어인 Java 프로그래밍 환경에서 JNI(Java Native Interface)를 이용한 실제 서비스에서의 성능 추정치를 분석한다. 추정한 성능은 Table 4.와 같다. Table 4.에서 n은 데이터 블록(block)의 개수를 의미한다.

하이브리드 방식을 사용하였을 때 데이터 오버헤

Table 4. Estimation of performance when applied to real environment(Java)

	ECDH (existing)	NTRU	ECDH session key + NTRU session key		
			KDF-1	KDF-2	KDF-3
Transmission data (byte)	32+128*n	699	-	-	-
Time	16.8 (㎳)	6.95 (㎳)	7.7 (㎲)	9.0 (㎲)	23.9 (㎲)

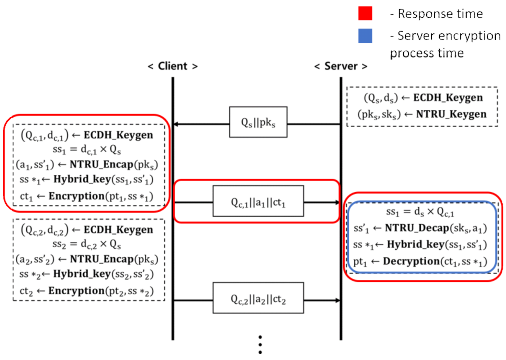

Fig. 14. Define response time and server encryption process time for SafeE2e product experiment

드는 699바이트가 증가하지만, 이는 실제 서비스에서 큰 영향을 주지 않을 것으로 예상한다. 기존 금융 거래 서비스에서 사용되고 있는 ECDH는 Java에서 구현이 되어있으며, 추가한 기능인 NTRU와 KDF는 JNI를 이용하여 C를 호출한다. 따라서 기존 ECDH 속도에 NTRU 알고리즘과 KDF의 속도가 추가로 소요된다. 이는 기존 ECDH만을 사용했을 때의 속도 대비 거래 당 약 6.96㎳ 정도의 시간이 더 소요될 것으로 추정된다.

4.3 하이브리드 키 교환 방식이 적용된 양자내성암호 제품 실험 결과

4.2절에서 추정한 실험 결과로 하이브리드 키 결합 방식 중 KDF-1이 가장 효율적임을 확인하였다. 추정 결과를 바탕으로 가장 효율적인 키 결합 방식을 실제 제품에 적용하기 위해 실험을 진행하였다.

제품은 하이브리드 방식의 데이터 암호화 기술을 적용한 주식회사 에잇바이트의 SafeE2e이다. SafeE2e는 데이터 암/복호화에 대칭키 알고리즘 SEED-128를 사용했으며, 대칭키는 키 교환 과정을 통해 생성한 16바이트 세션키를 사용한다. 본 제품의 키 교환 알고리즘은 하이브리드 KDF-1 모델을 따르며, NIST의 타원 곡선 기반 공개키 암호 ECDH Curve25519와 양자내성암호 NTRU-hps2048509를 사용한다.

다만, 상용 제품을 만드는 과정에서 4.2절에서 사용한 P-256을 Curve25519로 변경하였으며, Fig. 10.의 KDF-1에 내부의 HKDF 함수 대신 해시함수(SHA2-256)를 적용하였다. 먼저 타원곡선은 제품 제작 과정에서 Curve25519가 속도가 더 빨라 교체하였다. 다음으로 HKDF에는 salt 값이 사용되는데, 따라서 이 값을 인가된 사용자들 간에 미리 공유할 계획이었다. 다만, 사전에 salt를 공유하는 것은 67개의 개별저축은행 클라이언트와 1개의 중앙회 서버간에 해당 salt를 저장, 관리하는 이슈가 발생하게 되어 고객사 측에서는 수용하기 어려운 구성을 가질 수밖에 없었다. 이에 KDF-1 방식에 HKDF의 기본 알고리즘인 SHA2-256을 이용하여 일방향 암호화하게 되었다. 추가로 KDF-1 방식이 KDF-2, KDF-3보다 함수 API 호출이나 입출력 빈도수가 가장 적으므로 SHA2-256을 적용한 KDF-1 방식도 가장 빠르다고 추정 후 적용하였다.

서버의 실험 환경은 Linux 5.15 Intel(R) Pentium(R) CPU G3260 @ 3.30GHz 2CPU, 8GB MEM에서 OpenJDK 11 spring boot web application을 사용하고, 클라이언트의 실험 환경은 Windows 10 Intel(R) Core i3-9100 CPU @ 3.60GHz, 8GB MEM에서 OpenJDK 11 apache jmeter (1-thread, 10-loops)을 사

Table 5. Key exchange algorithm and parameters of SafeE2e product

Algorithm	Mathematical Problem	(data unit: byte)			
		pk	sk	ct	ss
ECDH NIST Curve25519	Discrete Logarithm	32	32	-	32
NTRU-hps 2048509	NTRU	699	935	699	32

Table 6. SafeE2e performance measure according to encrypted data size

	Encrypted data size			
	1MB	5MB	10MB	100MB
Response time (sec)	1.115	2.376	3.776	35.373
Server encrypt process time (sec)	0.036	0.158	0.311	3.093
Latency	3.18%	6.63%	8.24%	8.74%

용하여 진행하였다. 실험에 사용한 키 교환 알고리즘과 파라미터를 요약하면 Table 5.와 같다.

암호화하는 데이터 크기를 기준으로 제품의 응답 시간 및 서버 암호 처리 시간을 측정한 결과는 Table 6.과 같다. 응답시간과 서버 암호 처리 시간은 Fig. 14와 같이 정의하고 평균을 계산하였다. 측정은 데이터 크기마다 하이브리드 키 교환을 포함한 데이터 암호화를 10번씩 시행하였다. PQC의 서비스 지연율은 응답 시간에서 서버 암호 처리 시간이 차지하는 비율로, 데이터 크기가 10MB 이상이면 데이터 크기 변화와 비교하여 지연율이 크게 증가하지 않는 것을 확인하였다.

V. 결 론

본 논문에서는 금융거래 서비스의 프로토콜 안전성 강화를 위한 양자내성암호가 적용된 하이브리드 키 교환 프로토콜을 제안했다. 실험을 통해 현재 서비스에는 3가지 하이브리드 키 결합 방식 중 KDF-1 방식이 가장 적합함을 확인했다. 또한, NIST PQC 공모사업의 3라운드 격자기반 후보 알고리즘 중 가장 느린 NTRU도 실제 서비스에 적용할 수 있음을 확인했다. 따라서 KDF-1 방식에 다른 격자 기반 알고리즘을 사용하더라도 모두 실제 서비스에 적은 지연으로 적용할 수 있을 것으로 예상한다. 향후 연구에는 하이브리드 키 교환 프로토콜에 사용한 양자내성암호를 표준으로 선정된 격자 기반 암호 CRYSTALS KYBER[35]로 적용하는 연구와 함께, Java 구현 및 최적화를 진행할 예정이다.

References

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", SIAM review, vol 41, no 2, pp. 303-332, 1999.

[2] NIST, "Post-Quantum Cryptography" https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization, 2022.10.23.

[3] NIST, "Post-Quantum Cryptography: call for proposals" https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/call-for-proposals, 2022.10.23.

[4] Imperialviolet, "CECPQ1 results" https://www.imperialviolet.org/2016/11/28/cecpq1.html, 2022.10.23.

[5] Cloudflare, "Towards Post-Quantum Cryptography in TLS" https://blog.cloudflare.com/towards-post-quantum-cryptography-in-tls/, 2022.10.22.

[6] Cloudfare, "The TLS Post-Quantum Experiment" https://blog.cloudflare.com/the-tls-post-quantum-experiment/, 2022.10.24.

[7] Imperialviolet, "CECPQ2" https://www.imperialviolet.org/2018/12/12/cecpq2.html, 2022.10.23.

[8] Saarinen, O. M. Juhani, "Mobile energy requirements of the upcoming NIST post-quantum cryptography standards", 2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 23-30, Aug. 2020.

[9] D. Sikeridis, P. Kampanakis, M. Devetsikiotis, "Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH", Proceedings of the 16th International Conference on emerging Networking Experiments and Technologies, pp. 149-156, Nov. 2020.

[10] Y. Yeong-rok, C. Ji-sun, K. Green et. al., "Current status and prospects of security technology along with digital finance", Financial Security Institute, Yongin, CRR-VIII-2020-②-196, 2020.

[11] KFTC, "Payment and settlement statistics" https://www.kftc.or.kr/kftc/data/EgovkftcCount.do, 2022.10.25.

[12] Financial Services Commission, "[Easy to understand fintech] Financial open API" https://www.korea.kr/news/pressReleaseView.do?newsId=156318431,

2022.10.25.
[13] KFTC, "Banking open API system status and performance" https://www.korea.kr/news/pressReleaseView.do?newsId=156318431, 2022.10.26.
[14] Financial Security Institute Convergence Security Department Fintech Security Team, "Independent security check guide for institutions using open API in the financial sector", Financial Security Institute, Yongin, AGR-VI-2018-②-81, 2018.
[15] Financial Security Institute Security Research Department Security Technology Research Team, "Guide to utilizing encryption technology in the financial sector", Financial Security Institute, Yongin, AGR-VII-2018-②-84, 2018.
[16] RFC 8446, The Transport Layer Security (TLS) Protocol Version 1.3, IETF, 2018.
[17] "Saving banks also open the era of 'open banking'... What are the characteristics of mobile app for each saving banks?", Korea Financial Newspaper, 2021.04.29, 1.
[18] F. Arute, K. Arya, R. Babbush, et. al., "Quantum supremacy using a programmable superconducting processor", Nature, vol 574, pp. 505-510, Oct. 2019.
[19] IBM, "IBM's roadmap for scaling quantum technology" https://research.ibm.com/blog/ibm-quantum-roadmap, 2022.10.26.
[20] M. Mosca, "Setting the scene for the ETSI quantum-safe cryptography workshop", e-proceedings of 1st Quantum-Safe-Crypto Workshop, Sophia Antipolis, 2013.
[21] NIST, "Post-Quantum Cryptography: Selected Algorithms 2022" https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022, 2022.10.28,
[22] NIST, "PQC Standardization Process: Third Round Candidate Announcement" https://www.nist.gov/news-events/news/2020/07/pqc-standardization-process-third-round-candidate-announcement, 2022.10.16.
[23] D. Moody, "The 2nd round of the NIST PQC standardization process", National Institute of Standards and Technology, Tech. Rep, 2019.
[24] G. Alagic, J. A. Sheriff, D. Apon, et. al., "Status report on the second round of the NIST post-quantum cryptography standardization process", US Department of Commerce, NIST, 2020.
[25] NIST, "NTRU-Algorithm specifications and supporting documentation" https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions, 2022.09.13.
[26] HyoEun Seong, Yewon Kim, Yongjin Yeom, Ju-Sung Kang, "Accelerated Implementation of NTRU on GPU for Efficient Key Exchange in Multi-Client Environment", JKIISC, 31(3), pp. 481-496, Jun. 2021.
[27] IETF, "Hybrid key exchange in TLS 1.3" https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/, 2022.09.16.
[28] R. Azarderakhsh, "Post-quantum cryptography in embedded IoT devices", ICMC 2020, Sep. 2020.
[29] NIST SP 800-56C Rev.2, Recommendation for Key-Derivation Methods in Key-Establishment Schemes, NIST, Gaithersburg, Md., 2020.
[30] RFC 5869, HMAC-based extract-and-expand key derivation function (HKDF), IETF, 2010.
[31] 8Byte, "8byte Protocol" https://www.8byte.co.kr/, 2022.10.17.
[32] B. Hess, "Hybrid key agreement/KEM

construction and integration to IPsec IKEv2 VPN", ICMC 2020, Sep. 2020.
〔33〕 Nayoung Kim, Ju-Sung Kang, Yongjin Yeom, "Provable Security of PRF-based Key Derivation Functions according to Input Types of Counters", JKIISC, 25(3), pp. 547-557, Jun. 2015.
〔34〕 OpenSSL Project, "OpenSSL" https://github.com/openssl/openssl, 2022.10.20.
〔35〕 Pq-crystals, "CRYSTALS-KYBER: Algorithm Specifications And Supporting Documentation (version 3.02)" https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf, 2022.10.13.

〈저 자 소 개〉

권 수 진 (Sujin Kwon) 정회원
2020년 2월: 국민대학교 정보보안암호수학과 학사
2022년 2월: 국민대학교 일반대학원 금융정보보안학과 석사
〈관심분야〉 암호구현 및 분석, 난수성 분석 및 평가

김 덕 상 (Deoksang Kim) 정회원
1995년 8월: 서울대학교 수학과 졸업
1996년 1월~1999년 10월: ㈜다우기술
1999년 11월~2013년 3월: ㈜아톤(구. 에이티솔루션즈) 상무이사
2014년 2월~현재: ㈜에잇바이트 대표이사
〈관심분야〉 암호/인증 프로토콜 설계 및 제품 개발

박 영 재 (Yeongjae Park) 정회원
2019년 2월: 국민대학교 수학과 학사
2022년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 석사과정
〈관심분야〉 화이트박스 암호, 난수성 분석 및 평가

류 지 은 (Jieun Ryu) 학생회원
2022년 2월: 국민대학교 수학과 학사
2022년 3월~현재: 국민대학교 일반대학원 금융정보보안학과 석사과정
〈관심분야〉 암호구현 및 분석, 양자내성암호

강 주 성 (Ju-Sung Kang) 종신회원
1989년 2월: 고려대학교 수학과 졸업
1991년 2월: 고려대학교 일반대학원 수학과 석사
1996년 2월: 고려대학교 일반대학원 수학과 박사
1997년~2004년: 한국전자통신연구원 선임연구원/팀장
2004년 3월~현재: 국민대학교 과학기술대학 정보보안암호수학과 정교수
2013년~현재: 국민대학교 BK21+ 안전한 초연결사회를 위한 문제해결형 정보보안 교육연구단 교수
〈관심분야〉 암호이론, 정보보안 프로토콜, 안전성 분석 및 평가

염 용 진 (Yongjin Yeom) 종신회원
1991년 2월: 서울대학교 수학과 졸업
1994년 2월: 서울대학교 수학과 석사
1999년 2월: 서울대학교 수학과 박사
2000년 4월~2012년 2월: ETRI 부설연구소 책임연구원/팀장
2012년 3월~현재: 국민대학교 과학기술대학 정보보안암호수학과 정교수
2013년~현재: 국민대학교 BK21+ 안전한 초연결사회를 위한 문제해결형 정보보안 교육연구단 교수
〈관심분야〉 암호구현 및 분석, 보안시스템 평가