양자내성 암호 전환 프로토콜 확인

유튜브 주소: https://youtu.be/5hvLpNu796o

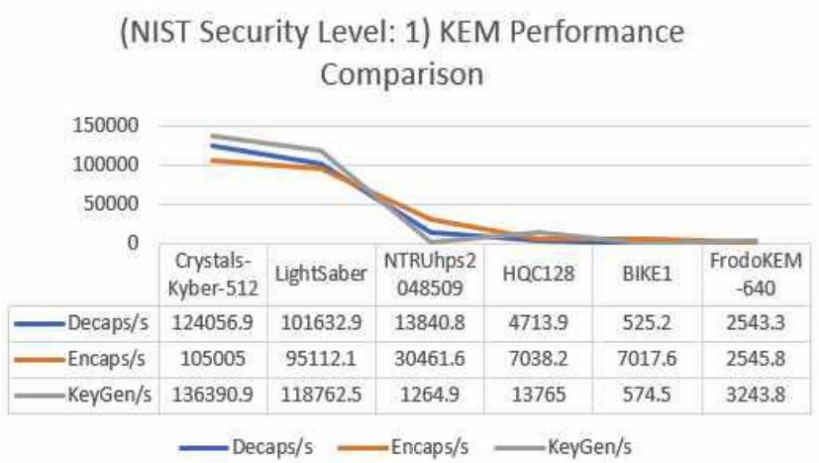
TLS 1.3에서 양자내성암호 적용 가능성 분석

TLS 1.3 프로토콜의 PQC 적용 가능성 및 성능 평가

- NIST 선정 PQC 알고리즘의 성능 비교 및 최적 알고리즘 제안
- **TLS 1.3 핸드셰이크 및 인증서 처리에서의 PQC 적용 가능성 분석**
  - 핸드 셰이크 과정 분석: 암호화, 복호화 속도 및 키 생성 속도 비교
    - PQC 알고리즘 적용 시 TLS 1.3의 성능 변화 측정 및 분석
- 핵심 이슈: **PQC의 공개키 및 서명 크기가 기존 TLS 1.3에 비해 큼**
  - Crystals-Dilithium5: 1,760bytes(RSA 대비 4배 이상)
  - Classic McEliece: 261,120bytes(키 크기 매우 큼)
- **X.509 인증서 수정**
  - “Subject Public Key Info” 섹션에 PQC 공개키 알고리즘 및 공개키를 추가
  - PQC를 통해 발급된 PQ(Pretty Quick) 서명으로 기존의 서명 항목을 대체

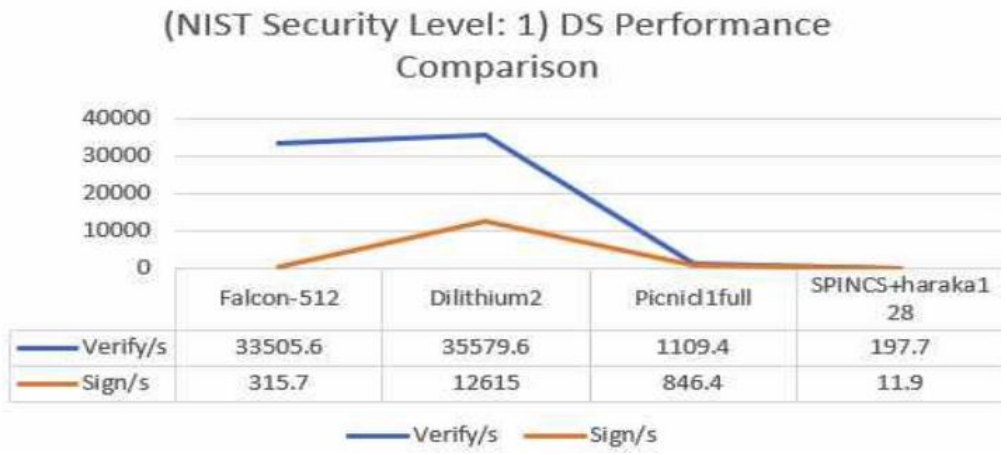
TLS 1.3에서 양자내성암호 적용 가능성 분석

암호화 알고리즘 성능 측정 결과 **Crystals-Kyber-512가 가장 높은 성능**을 보임

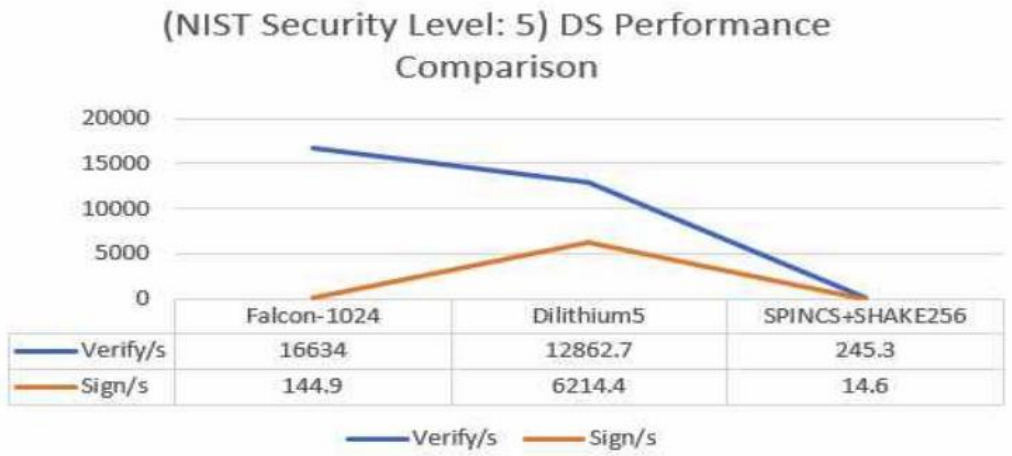


	Crystals-Kyber-512	LightSaber	NTRUhps2048509	HQC128	BIKE1	FrodoKEM-640
Decaps/s	124056.9	101632.9	13840.8	4713.9	525.2	2543.3
Encaps/s	105005	95112.1	30461.6	7038.2	7017.6	2545.8
KeyGen/s	136390.9	118762.5	1264.9	13765	574.5	3243.8

전자서명 알고리즘 성능 측정 결과 **Dilithium2, Falcon-1024가 가장 높은 성능**을 보임



	Falcon-512	Dilithium2	Picnic1full	SPINCS+haraka128
Verify/s	33505.6	35579.6	1109.4	197.7
Sign/s	315.7	12615	846.4	11.9

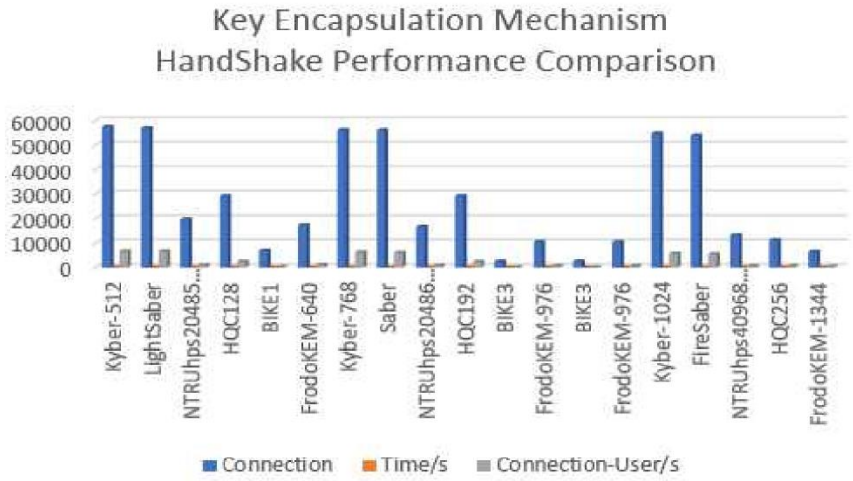


	Falcon-1024	Dilithium5	SPINCS+SHAKE256
Verify/s	16634	12862.7	245.3
Sign/s	144.9	6214.4	14.6

TLS 1.3에서 양자내성암호 적용 가능성 분석

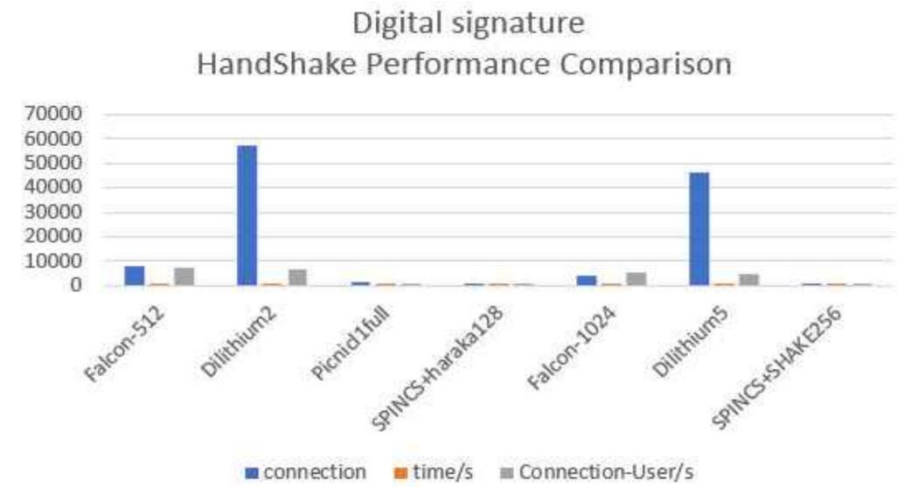
암호화 알고리즘 핸드셰이크 성능 측정 결과

Crystals-Kyber-512가 가장 높은 효율성을 보임



전자서명 알고리즘 핸드셰이크 성능 측정 결과

Dilithium이 가장 높은 효율성을 보임



Post-Quantum Cryptography (PQC) Network Instrument: Measuring PQC Adoption Rates and Identifying Migration Pathways

- PQC 네트워크 인스트루먼트를 사용한 **PQC 알고리즘 적용 현황 분석**
  - PQC 인스트루먼트: 네트워크 프로토콜에서 PQC 채택률을 실시간으로 측정
  - 트래픽에서 암호화 알고리즘 사용 현황 자동 수집 후 정규화
    - 정규화된 데이터를 통해 각 프로토콜 별 PQC 채택 상태 비교 가능
  - **SSH 및 TLS 등의 프로토콜에서 사용된 암호화 알고리즘 분석 수행**

- **응용 프로그램 및 프로토콜의 PQC 적용 현황**
  - **대부분의 프로토콜 및 응용프로그램은 PQC 준비 상태 미비**
    - DHCP, DNS, SMTP 등은 현재 PQC 구현이 진행되지 않음
  - SSH는 sntrup761x25519-sha512 키 교환 방식을 통해 PQC 구현
  - SSL은 KEM(BIKE, Kyber), DSA(Dilithium)을 통해 PQC 구현
  - HTTP 및 FTP는 SSL/TLS를 통해 PQC 구현 가능

Post-Quantum Cryptography (PQC) Network Instrument: Measuring PQC Adoption Rates and Identifying Migration Pathways

응용 프로그램 및 프로토콜의 PQC 적용 현황

프로토콜	프로토콜 설명	pqc 구현
BHR	데이터 차단 라우터	N/A
DHCP	동적 호스트 구성 프로토콜	N/A
DNS	도메인 이름 시스템	N/A
DPD	피어 검출	N/A
HTTP	하이퍼텍스트 전송 프로토콜	SSL/TLS를 통해 구현
FTP	파일 전송 프로토콜	SSL/TLS를 통해 구현
Kerberos	네트워크 인증 프로토콜	N/A
krb5	클라이언트/서버 데이터 통신 프로토콜	N/A
LDAP	관계형 데이터베이스 프로토콜	N/A
NTLM	LAN 관리 프로토콜	N/A
RADIUS	원격 인증 사용자 서비스	N/A
RDP	원격 데스크탑 프로토콜	N/A
SIP	세션 시작 프로토콜	N/A
SMB	서버 메시지 블록	N/A
SSH	보안 셸	sntrup761x25519-sha512@openssh.com 키 교환 방식을 통해 구현
SSL	보안 소켓 계층	KEM(BIKE, CRYSTALS-Kyber), DSA(CRYSTALS-Dilithium)를 통해 구현
SMTP	간이 메일 전송 프로토콜	N/A
Shibboleth	분산 컴퓨팅 인증 시스템	N/A

Post-Quantum Cryptography (PQC) Network Instrument: Measuring PQC Adoption Rates and Identifying Migration Pathways

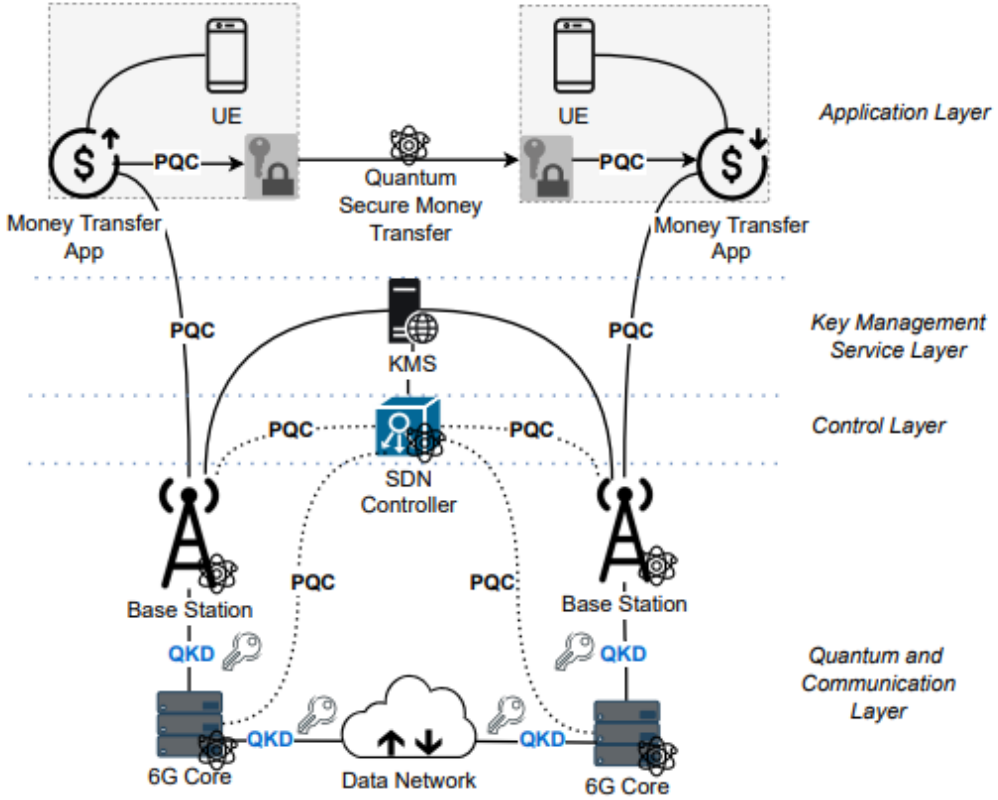
- 네트워크 프로토콜 PQC 채택 분석 방법론
  - 네트워크 트래픽을 분석하여 **주요 프로토콜에서 암호화 알고리즘 사용 현황 수집**
    - 수집한 데이터를 바탕으로 선형 회귀 분석 등의 통계적 기법을 활용해 PQC 채택률 측정

- 네트워크 프로토콜 PQC 채택 현황
  - **대부분의 프로토콜에서 레거시 암호화 알고리즘 주로 사용 중**
    - SSH: sntrup761x25519-sha512 비율 약 0.029%
    - TLS: Kyber를 지원하나, 실제 사용은 검출되지 않음

- 네트워크 프로토콜 PQC 전환 전략
  - **하이브리드 방식 도입 고려**
  - OpenSSH에서 sntrup761x25519-sha512와 같은 하이브리드 방식 채택

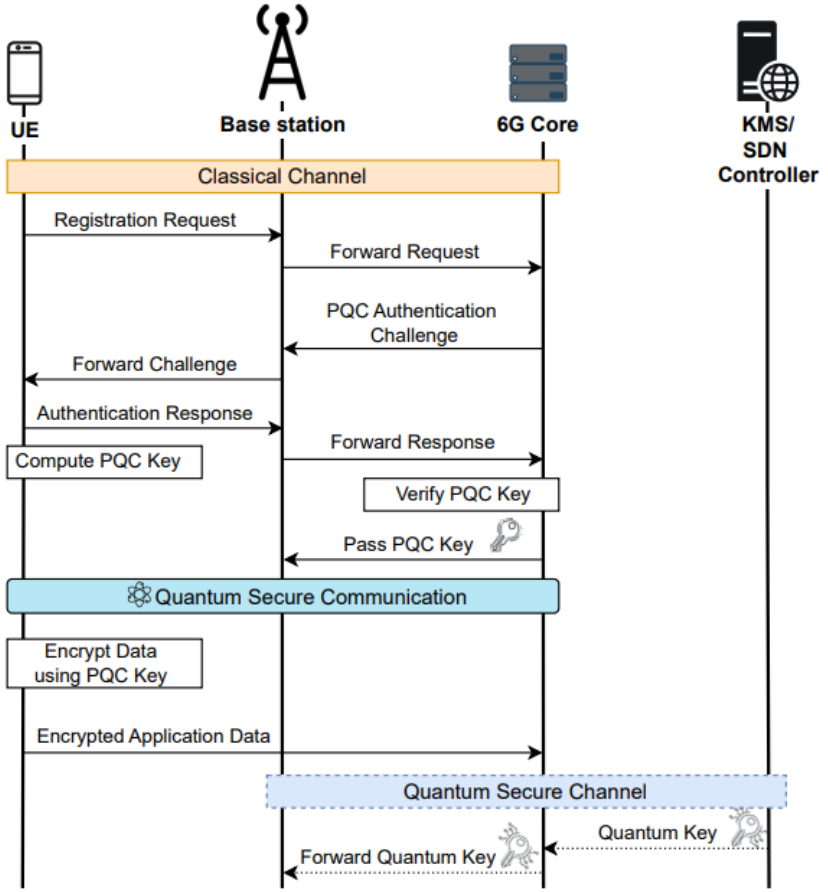
Exploring Post Quantum Cryptography with Quantum KeyDistribution for Sustainable Mobile Network Architecture Design

- **모바일 네트워크 보안을 위한 PQC 및 QKD 통합 아키텍처 제안**
  - Application Layer: 데이터 전송 시 PQC 암호화 수행
  - Key Management Service Layer: QKD를 통한 비밀키 관리
  - Control Layer & Communication Layer: 네트워크 제어 및 데이터 통신 수행



Exploring Post Quantum Cryptography with Quantum KeyDistribution for Sustainable Mobile Network Architecture Design

- **모바일 네트워크 환경에서의 PQC 및 QKD 구현 및 최적화**
- **구현 방안**
  1. 클래식 채널을 통한 초기 등록 및 인증
  2. 양자 보안 채널을 통한 키 전달
  3. 암호화 된 데이터 전송
  4. KMS, SDN을 통한 중앙 관리
- **최적화 방안**
  1. 하드웨어 가속기 활용
  2. 알고리즘 최적화
     - 키 관리 과정에서의 병목 현상 감소
  3. **프로토콜 오버헤드 최적화**
     - 실시간 통신이 필수적이기 때문



Q & A