TLS 구현에서 양자내성암호 적용 사례

: 라이브러리 및 연구에 관하여

https://youtu.be/0vTbOL57owQ

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB

서론

관련연구

사례

결론

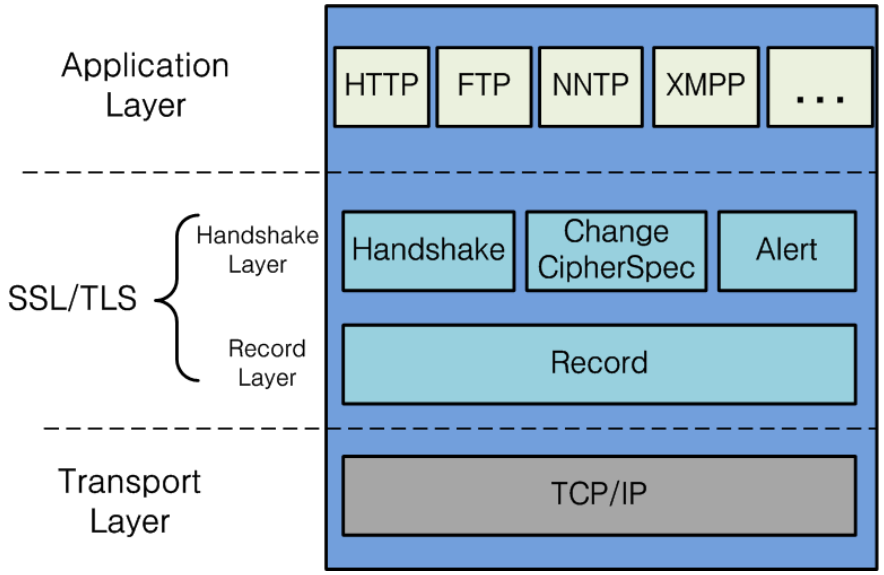
서론

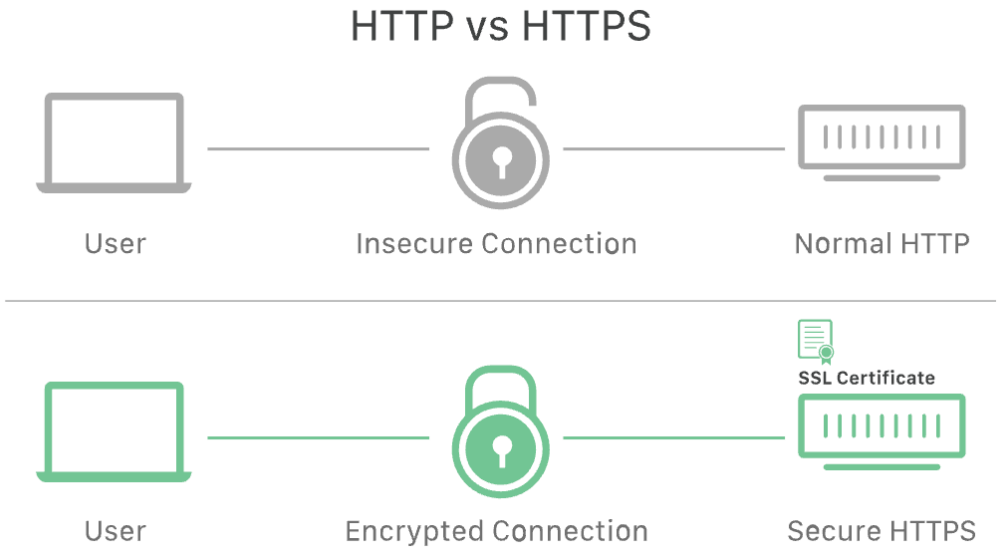
- 최근 양자컴퓨터가 등장함에 따라, 쇼어 알고리즘 및 그루버 알고리즘을 활용하여 현대 암호에 대한 공격이 가능
  → 현대 암호를 양자컴퓨터 환경에서의 공격에도 안전한 **양자내성암호로 교체 필요**
- 현재 네트워크 상에서는 일반적으로 **SSL**과 **TLS 프로토콜**이 사용
- 해당 프로토콜에서는 주로 RSA 키 교환 알고리즘이 사용
  → **프로토콜의 안정성이 위협**
- 최근 TLS 구현에서 양자내성암호를 적용한 사례들이 등장

NIST
National Institute of
Standards and Technology

SSL (Secure Socket Layer) 및 TLS(Transport Layer Security)

- SSL은 1995년에 Netscape가 처음으로 개발한 프로토콜이며, TLS는 SSL 버전 3.0을 표준화한 것
- 클라이언트와 서버에 대한 **인증 및 데이터 암호화를 수행**함으로써 개인정보 보호, 인증 그리고 데이터 무결성을 보장하는 보안용 프로토콜
- 주로, 웹 브라우저와 웹 서버간의 **안전한 통신을 보장**하기 위해 사용
- TLS 기능을 제공하는 라이브러리 : GnuTLS, mbedTLS, OpenSSL 그리고 WolfSSL 등





양자컴퓨터

- 양자컴퓨터는 양자 중첩과 얽힘을 활용하여 데이터를 처리하는 컴퓨터[2]
  → 양자컴퓨터의 기본 연산 단위는 **큐비트**
- 큐비트는 양자 중첩을 통해 확률적으로 가능한 상태들을 모두 가질 수 있어 0과 1의 상태를 동시에 가지며 연산 가능
- 고전 컴퓨터보다 **월등한 속도로 연산을 수행**
  →기존 암호 알고리즘이 **양자 컴퓨터에 의해 해독될 위험**
- 쇼어 알고리즘: 다항 시간 내에 소인수 분해를 수행할 수 있는 양자 알고리즘[3]
  → **여러 공개키 암호 알고리즘 위협**
- 그루버 알고리즘 : 정렬되지 않은 데이터 집합 내에서 특정 데이터를 빠른 속도로 찾아내는 양자 알고리즘[4]
  → 그루버 알고리즘이 $n-\text{bit}$의 안전성을 갖는 대칭키 암호에 대해 적용될 경우 $n/2$-bit까지 낮출 수 있음

Open Quantum Safe(OQS)

- 양자내성암호를 통한 개발 및 프로토타입을 지원하는 것을 목표로 하는 오픈 소스 프로젝트[5]
- liboqs : 양자내성암호를 C언어로 구현한 라이브러리
- OpenSSL과 BoringSSL 라이브러리에 양자내성암호를 적용 (OQS-OpenSSL, OQS-BoringSSL)

Key Exchange	Authentication
Bike Classic McEliece FrodoKEM HQC Kyber NTRU NTRU-Prime SABER	CRYSTALS-Dilithium Falcon Picnic Rainbow SPHINCS-Haraka SPHINCS-SHA256 SPHINCS-SHAKE256

liboqs

liboqs is an open source C library for quantum-safe cryptographic algorithms.

OQS-OpenSSL

- OpenSSL 1.1.1로부터 분기된 프로젝트
- TLS 1.3 프로토콜에서 **양자 내성 키 교환** 및 **인증 알고리즘**을 liboqs를 통해 추가
- macOS 10.14 (with clang 10.0.0), Ubuntu 18.04.1 (with gcc-7)에서 정상적으로 작동
- Windows 10 상에서는 Visual Studio 2019에서 작동

OpenSSL
Cryptography and SSL/TLS Toolkit

Key Exchange	Authentication
Bike	CRYSTALS-Dilithium
CRYSTALS-Kyber	Falcon
FrodoKEM	Picnic
HQC	Rainbow
NTRU	SPHINCS-Haraka
NTRU-Prime	SPHINCS-SHA256
SABER	SPHINCS-SHAKE256

OQS-BoringSSL

- OpenSSL로부터 분기된 프로젝트로 구글에서 관리
  → 크롬과 안드로이드 환경에서 보다 최적화
- OQS-OpenSSL과 마찬가지로 liboqs를 사용하였으며 TLS 1.3 프로토콜에서 구현
- Ubuntu 18.08 이상 버전에서 정상적으로 작동함을 확인

google/**boringssl**

Mirror of BoringSSL

Key Exchange	Authentication
Bike	CRYSTALS-Dilithium
CRYSTALS-Kyber	Falcon
FrodoKEM	Picnic
HQC	Rainbow
NTRU	SPHINCS-Haraka
NTRU-Prime	SPHINCS-SHA256
SABER	SPHINCS-SHAKE256

WolfSSL

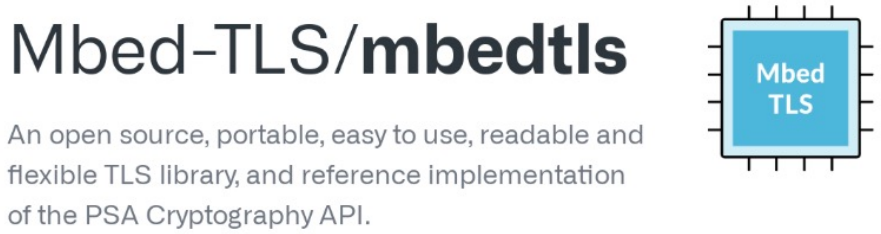
- WolfSSL은 임베디드 시스템 개발자의 사용을 타겟으로 개발된 라이브러리
- OpenSSL보다 경량화된 SSL/TLS 라이브러리
  → 리소스가 제한된 환경을 대상
- **C언어**, JAVA, C#, Python 지원
- 지원하는 운영체제 :Window, Linux, Mac OS 등

Key Exchange	Authentication
KYBER SABER NTRU	Dilithium FALCON

mbedTLS

- **mbedTLS**
  - C언어로 작성된 라이브러리로, SSL 프로토콜을 구현하고 다양한 유틸리티 기능을 제공
  - 소형 임베디드 장치에 적합하도록 경량화되었다는 점에서 WolfSSL과 유사
  - 다수의 운영체제 및 아키텍쳐에서 동작

- mbedTLS 상에서 격자기반 **양자내성 키 교환 알고리즘을 구현**[6]
  - Lizard : 노이즈를 추가하여 키를 숨기는 격자기반 양자내성 암호
    → 16KB 이상의 파라미터 데이터를 전송
  - mbedTLS의 경우 16KB 이상의 데이터를 TLS로 전송할 수 없음
    → 16KB 이상의 데이터 크기를 가지는 메시지를 나누어 보내는 기법 구현 (Handshake Message Fragmentation)



결론

- 양자내성암호를 TLS 상에 적용하기 위해 다양한 프로젝트가 진행되고 있음
- 기존에 구현되어 있는 양자내성암호 라이브러리를 통해 TLS에 적용한 사례
- mbedTLS 상에서 발생한 문제점을 해결하여 양자내성암호 Lizard를 적용한 사례
- TLS에서도 양자내성암호가 적용될 수 있음을 확인

참고문헌

[1] Hickman, Kipp, and Taher Elgamal. "The SSL protocol." (1995): 501.

[2] Steane, Andrew. "Quantum computing." Reports on Progress in Physics 61.2 (1998): 117.

[3] Shor, Peter W. "Algorithms for quantum computation: discrete logarithms and factoring." Proceedings 35th annual sympo sium on foundations of computer science. Ieee, 1994.

[4] Grover, Lov K. "A fast quantum mechanical algorithm for database search." Proceedings of the twenty-eighth annual AC M symposium on Theory of computing. 1996.

[5] Stebila, Douglas, and Michele Mosca. "Post-quantum key exchange for the internet and the open quantum safe project. " International Conference on Selected Areas in Cryptography. Springer, Cham, 2016.

[6] Park, Chanhui, et al. "Implementation of lattice-based quantum-resistant key exchange algorithm." Review of KIISC 30.3 (2020): 11-16.

Q & A