KIPS Trans. Comp. and Comm. Sys.
Vol.9, No.8 pp.165~170 pISSN: 2287-5891


https://doi.org/10.3745/KTCCS.2020.9.8.165

# Design of a Lightweight Security Protocol Using Post Quantum Cryptography

Kyung Bae Jang† · Min Joo Sim†† · Hwa Jeong Seo†††

## ABSTRACT

As the IoT (Internet of Things) era is activated, a lot of information including personal information is being transmitted through IoT devices. For information protection, it is important to perform cryptography communication, and it is required to use a lightweight security protocol due to performance limitations. Currently, most of the encryption methods used in the security protocol use RSA and ECC (Elliptic Curve Cryptography). However, if a high performance quantum computer is developed and the Shor algorithm is used, it can no longer be used because it can easily solve the stability problems based on the previous RSA and ECC. Therefore, in this paper, we designed a security protocol that is resistant to the computational power of quantum computers. The code-based crypto ROLLO, which is undergoing the NIST (National Institute of Standards and Technology) post quantum cryptography standardization, was used, and a hash and XOR computation with low computational consumption were used for mutual communication between IoT devices. Finally, a comparative analysis and safety analysis of the proposed protocol and the existing protocol were performed.

Keywords : IoT, Information Protection, Lightweight Security Protocol, RSA, ECC, Quantum Computer, NIST Post Quantum Cryptography Standardization, Code-Based Cryptography, ROLLO

# 양자내성암호를 활용한 경량 보안 프로토콜 설계

장 경 배† · 심 민 주†† · 서 화 정†††

## 요 약

IoT (Internet of Things) 시대가 활성화되면서 개인정보를 포함한 많은 정보들이 IoT 디바이스들을 통해 전달되고 있다. 정보보호를 위해 디바이스끼리 상호 암호화하여 통신하는 것이 중요하며 IoT 디바이스 특성상, 성능의 제한으로 인해 경량 보안 프로토콜 사용이 요구된다. 현재 보안 프로토콜에서 사용하는 암호 기법들은 대부분 RSA, ECC (Elliptic Curve Cryptography)를 사용하고 있다. 하지만 고사양의 양자 컴퓨터가 개발되고 쇼어 알고리즘을 활용한다면 앞선 RSA와 ECC가 근거하는 안정성의 문제를 쉽게 해결할 수 있기 때문에 더 이상 사용할 수 없다. 이에 본 논문에서는 양자 컴퓨터의 계산능력에 내성을 가지는 보안 프로토콜을 설계하였다. 미국 NIST (National Institute of Standards and Technology) 양자내성암호 표준화 공모전을 진행중인 코드기반암호 ROLLO를 사용하였으며, IoT 디바이스끼리의 상호 통신을 위해 연산 소모가 적은 해시, XOR연산을 활용하였다. 마지막으로 제안하는 프로토콜과 기존 프로토콜의 비교 분석 및 안전성 분석을 실시하였다.

키워드 : IoT, 정보보호, 경량 보안 프로토콜, RSA, ECC, 양자 컴퓨터, NIST 양자내성암호 표준화 공모전, 코드기반암호, ROLLO

## 1. 서 론

현대에 있어 IoT[1] 기술의 중요성은 점차 부각되지만 사용자의 개인정보를 보장하고 민감 데이터를 보호하는데 있어 많은 문제와 위험도 따른다. 예를 들어 악의적인 공격자가 IoT 시스템인 헬스케어를 사용하는 이용자에 대한 거짓 의료정보를 의사에게 전송한다면 의사는 잘못된 처방을 내릴 수 있다.

이러한 보안 위협에 대처하기 위해서는 서로의 신원을 올바르게 확인하고 통신할 수 있는 보안 프로토콜이 필요하다. 이때 IoT 디바이스의 제한된 성능 탓에 경량 설계가 요구되므로 사용하는 암호화 방식도 경량 암호화 방식을 사용해야 한다.

※ 이 성과는 부분적으로 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1C1B5075 742) 그리고 이 논문은 부분적으로 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구) 그리고 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의지원을 받아 수행된 연구임(〈Q|Crypton〉, No.2019-0-00033, 미래컴퓨팅 환경에 대비한 계산 복잡도 기반 암호 안전성 검증 기술개발).
※ 이 논문은 2020년도 한국정보처리학회 춘계학술발표대회에서 '코드기반암호를 활용한 IoT 환경 보안 프로토콜 설계'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 한성대학교 IT융합공학부 석사과정
†† 준 회 원 : 한성대학교 IT융합공학부 학사과정
††† 종신회원 : 한성대학교 IT융합공학부 조교수
Manuscript Received : June 29, 2020
Accepted : July 8, 2020
* Corresponding Author : Hwa Jeong Seo(hwajeong84@gmail.com)

현재 ECC(Elliptic Curve Cryptography)를 많은 곳에서 경량 공개키 암호로 사용하고 있으며 대표적으로 2017년 Wang의 프로토콜[2] 또한 그렇다. ECC는 이산대수 문제의 어려움에 기반하고 있지만 이는 양자 컴퓨터가 개발된다면 쉽게 해결되기 때문에 더 이상 사용할 수 없게 된다.

이에 기존 암호시스템들을 무너뜨릴 수 있는 양자컴퓨터의 계산 능력에 내성을 가진 양자내성암호 연구가 이루어지고 있다. 미국 NIST에서는 2016년 양자내성암호 표준화 공모전을 주최 하였고 세계 여러 각국에서 양자내성암호 알고리즘을 제출하였다. 현재 26개 후보들이 살아남아 Round2에 대한 평가를 진행 중이며 코드, 격자, 다변수다항식, 아이소제니 기반암호들로 구성되어 있다. 이에 본 논문에서는 NIST 양자내성암호 공모전을 진행 중인 코드기반암호 7가지에 대해 비교해보고 그 중 하나인 ROLLO를 활용하여 경량 보안 프로토콜을 설계하였다. IoT 환경을 대상으로 하기 때문에 암호화 횟수를 최소로 수행하고, 상대적으로 연산이 적은 해시 연산과 XOR 연산을 사용하여 설계하였다. 또한 통신과정에서 가능한 다양한 공격들을 가정하여 안전성 분석을 실시하였으며 기존 프로토콜과의 성능비교를 진행하였다.

## 2. 관련 연구

### 2.1 코드기반암호

코드기반암호의 원리는 송신자가 메시지에 고의로 수정 가능한 오류를 첨부한다. 그리고 올바른 수신자는 오류수정코드를 알고 있어 첨부된 오류를 손쉽게 수정할 수 있다. Robert J.McEliece는 1978년, 최초의 코드기반암호 McEliece[3]를 제안하였다. McEliece에서는 Goppa 코드라는 오류수정코드를 사용하는데 현재 NIST 양자내성암호 공모전 Round 2, 7개의 코드기반암호 중 Classic McEliece와 NTS-KEM이 Goppa 코드를 그대로 사용하고 있다. Goppa 코드는 역사가 길어 뛰어난 보안성을 자랑하지만 키 사이즈가 매우 크다는 단점이 있다. 하지만 한계점인 키 사이즈를 줄이기 위해 Goppa 코드가 아닌 새로운 코드를 사용하는 연구가 진행중이며 Quasi Cyclic, Rank metric 코드에 기반한 5개의 암호가 Round 2를 진행중이다.

### 2.2 NIST 양자내성암호 공모전 코드기반암호와 ROLLO

ROLLO[4]는 양자내성암호 표준화 공모전 Round2를 진행중인 코드기반암호 중 하나이다. Goppa 코드가 아닌 효율성을 중시한 Rank Metric 코드(1991)를 기반으로 하여 키 사이즈와 계산 복잡도 측면에서 매우 효율적이다. NIST 양자내성암호 공모전 Round2를 진행중인 Goppa 코드를 사용하는 Classic McEliece, NTS-KEM 등 다른 코드기반암호와의 성능 비교를 위해 7개의 후보군 모두를 저전력 모바일 프로세서인 ARM에서 속도를 측정하였다. 실험 환경과 연산 속도 비교 결과는 Table 1, Table 2와 같다.

Table 1. Experiment Environment

| | Raspberry Pi B+ |
|---|---|
| CPU | ARM Cortex-A53@1.4 GHz |
| Memory | 1GB LPDDR2 SDRåAM |
| OS | Raspbian |

Table 2. Comparison of Timings (in ms)

| | Key gen | Enc | Dec |
|---|---|---|---|
| mceliece348864 | 1780.94 | 0.84 | 247.94 |
| mceliece460896 | 3864.43 | 2.52 | 630.32 |
| nts_kem_12_64 | 291.66 | 1.28 | 9.8 |
| bike-1 CCA | 3.40 | 3.46 | 19.14 |
| bike-2 CCA | 37.86 | 1.86 | 16.11 |
| bike-3 CCA | 1.88 | 3.81 | 18.26 |
| hqc-128-1 | 50.65 | 9.02 | 15.30 |
| rqc-128 | 2.64 | 4.65 | 27.03 |
| LEDAcrypt PKE 1-2 | 4123.01 | 43.21 | 44.81 |
| rollo-I-128 | 8.19 | 1.21 | 4.27 |
| rollo-II-128 | 73.94 | 6.89 | 19.84 |
| rollo-III-128 | 1.67 | 2.62 | 3.98 |

7가지 코드기반 암호 후보들의 특성과 성능을 비교 분석해보면 기존 Goppa 코드 사용을 고수하는 Classic McEliece, NTS-KEM 과 ROLLO의 Rank Metric 코드와 같이 새로운 코드로 대체하는 암호들로 나뉜다.

40년의 역사를 가지고 있는 Goppa 코드는 보안성을 강점으로 내세우지만 효율성이 떨어진다. 새롭게 연구된 코드들은 상대적으로 짧은 검증 기간을 가지고 있지만 높은 성능을 보여준다. 128-bit 보안 레벨 기준으로 Classic McEliece의 암호문은 128 바이트로 작은 크기지만 공개키는 261120 바이트, 개인키가 6452 바이트이며 이러한 키 크기는 저성능 8-bit AVR 프로세서의 경우에는 저장조차 할 수 없다. 반면, 본 논문에서 사용하는 ROLLOO-II의 공개키는 1941 바이트, 개인키는 40 바이트로 훨씬 작은 키 사이즈를 제공하며 암호문은 2089 바이트의 크기를 가진다.

7가지 코드기반암호들은 초기에 IND-CPA의 보안레벨을 제공하였다. 하지만 공모전이 Round2로 진행되면서 대부분의 암호들이 기존 PKE(Public Key Encryption) 구조를 KEM(Key Encapsulation Mechanism) 구조로 변환함으로써 IND-CCA의 보안레벨로 증가시켰다. KEM은 실제로 메시지는 대칭키로 암호화 하고 이때 사용되는 키를 상대방과 공개키 암호를 사용하여 공유하는 하이브리드 구조이다. 현재 Round2의 코드기반암호 중 LEDAcrypt, ROLLO만이 PKE 버전을 제공하고 있다.

본 논문에서 제안하는 프로토콜의 암호화 기법은 ROLLO-I, II, III 중 II를 선택하였다. ROLLO-I, ROLLO-III는 KEM 방식으로 자체적으로 공개키와 개인키를 활용하여 임의의 대

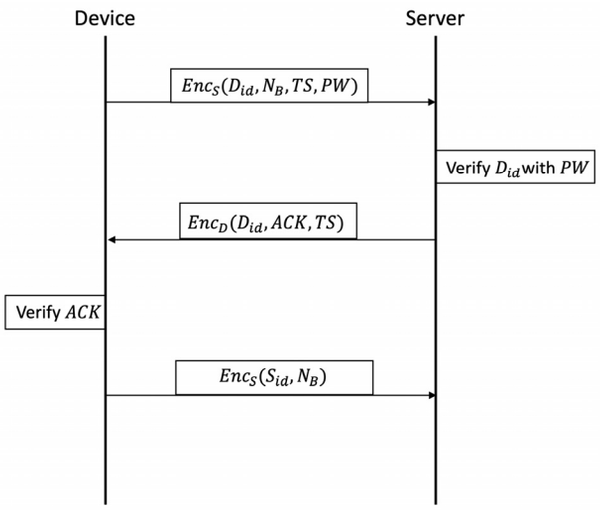


Fig. 1. Kumar S. Roy's Registration Phase

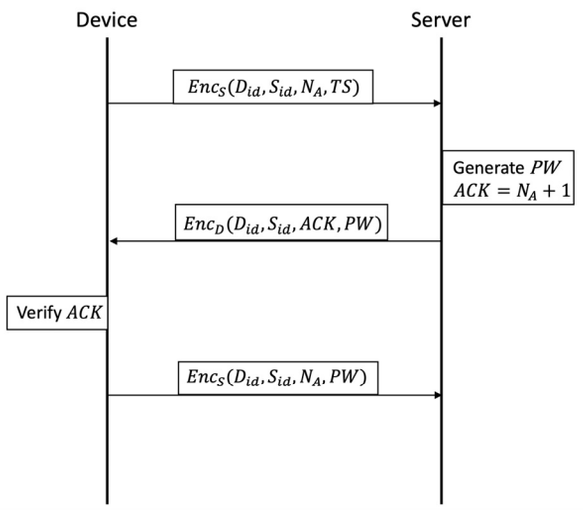


Fig. 2. Kumar S. Roy's Authentication Phase

칭키를 양측에서 비밀스럽게 설립한 후, 대칭키를 활용하여 서로의 메시지를 암호화 및 복호화한다. 하지만 ROLLO-II는 공개키로 전달하고 싶은 메시지를 바로 암호문으로 생성하고 개인키로 복호화 하는 PKE 방식을 제공한다. KEM 구조의 ROLLO-I, ROLLO-III를 사용하여도 무방하지만 본 논문에서는 디바이스 ID, 패스워드, nonce 값을 바로 암호화하여 전송하는 PKE 구조의 ROLLO-II를 선택하였다.

### 2.3 Kumar S. Roy's Protocol

2019년 Kumar S. Roy는 코드기반암호 McEliece의 변형버전인 Neiderrieter를 사용하여 경량 보안 프로토콜[5]을 제안하였다. 등록과 인증 2가지 절차로 구성되며 Fig. 1, Fig. 2와 같다. 표기법에 대한 설명은 Table 3을 참고하면 된다. 프로토콜을 살펴보면, 보안성의 측면에서는 통신과정에서 발생할 수 있는 재전송 공격을 방지하기 위해 타임스탬프와 nonce 값이 메시지에 포함된다. 디바이스의 익명성은 등록 및 통신과정에서 디바이스의 id를 코드기반암호 Nederrieter로 암호화한다. 그리고 모든 디바이스와 서버끼리 주고받는 메시지는 모두 암호화 되어 등록 과정에서 3번, 인증 과정에서는 3번의 암, 복호화가 수행된다.

본 논문에서 제안하는 프로토콜에서는 등록과정에서 맨 처음 신원을 확인하고 패스워드를 부여받는 과정에서는 기존 기법과 동일하게 총 2번의 암호화를 수행한다. 하지만 이후 통신과정에서는 신원 확인 시 사용한 nonce 값과 해시 및 XOR 연산만을 활용한다. 이는 암호기법에 비해 매우 단순한 연산으로써 디바이스와 서버에서 수행하는 연산을 대폭 줄일 수 있다. 인증 과정에서는 맨 처음 디바이스의 익명성을 지키기 위해 한 번의 암호화만 수행하고, 이후부터는 마찬가지로 nonce 값과 해시 및 XOR 연산을 활용하여 안전한 통신을 수행한다. 5장에서 다시 제안하는 프로토콜과 기존 프로토콜과의 성능 및 안정성 측면에서 비교 분석하고자 한다.

## 3. 제안 프로토콜

제안하는 보안 프로토콜은 다음 두 가지로 구성된다. 적합 디바이스를 서버에 등록하는 절차와 등록된 디바이스가 서버와 상호 통신하기 위한 인증 절차로 이루어진다. 암호화 기법으로는 양자컴퓨터의 공격에 내성을 가질 수 있도록 코드기반암호 ROLLO-I, II, III 중 II을 사용하였다.

프로토콜에서 사용되는 연산 중에는 암, 복호화에 많은 비용이 소모된다. 따라서 제안하는 프로토콜에서는 경량 설계를 위해 암, 복호화 횟수를 최소로 수행하고 나머지 연산들은 해시 함수 SHA-512와 XOR 연산만을 사용하였다. 프로토콜 설명을 위한 표기법은 Table 3과 같다.

Table 3. Notation

| Notation | Meaning |
|---|---|
| $Req$ | Request message for registration |
| $Enc_s$ | Encrypt with server's public key |
| $Dec_s$ | Decrypt with server's private key |
| $D_{id}$ | Device id |
| $S_{id}$ | Server id |
| $N_A$, $N_B$ | Nonce value |
| $TS$ | Time stamp |
| $PW$ | Password |
| $PW_{temp}$ | Temporary password |
| $h$ | Hash function |
| $DB$ | Database |
| $SK$ | Session key |

### 3.1 디바이스 등록

디바이스가 서버에 등록되는 단계는 총 4단계로 구성되며 Fig. 3과 같다.

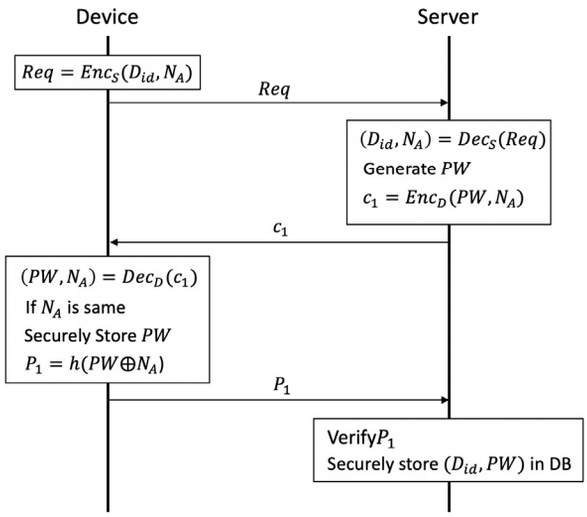


Fig. 3. Proposed Protocol's Registration Phase

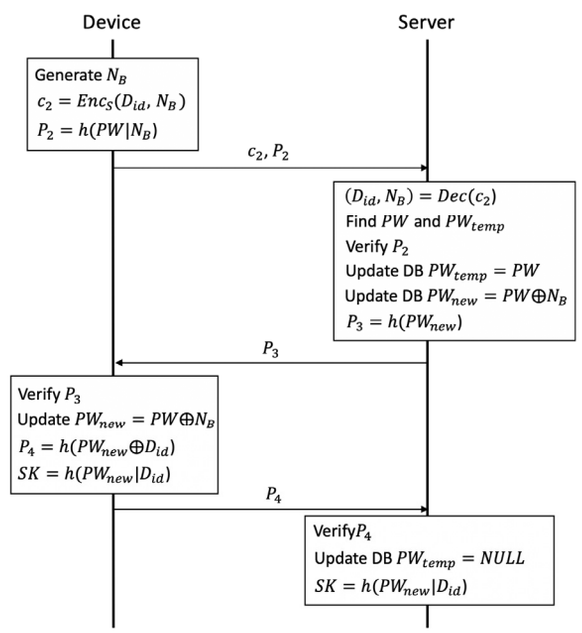


Fig. 4. Proposed Method's Authentication Phase

첫 번째, 서버에 등록을 원하는 디바이스는 자신의 디바이스 id와 nonce 값을 서버의 공개키로 암호화하여 전송한다.

두 번째, 서버는 수신한 메시지 $Req$를 자신의 개인키로 복호화 하여 디바이스 id와 nonce 값을 확인한 뒤 해당 디바이스를 위한 패스워드를 생성한다. 그리고 서버의 신원확인이 가능한 nonce 값과 생성한 패스워드를 해당 디바이스의 공개키로 암호화하여 전송한다.

세 번째, $c_1$을 수신한 디바이스는 자신의 개인키로 복구한 nonce값이 일치한다면 패스워드를 안전하게 저장한다.

마지막으로 $P_1 = h(PW \oplus N_A)$를 계산하여 서버에 전송한다. 해시의 입력 값을 알고 있는 서버는 $P_1$의 검증을 수행한 뒤, 데이터베이스에 디바이스 id와 패스워드를 저장한다.

### 3.2 디바이스 서버 간 상호 인증

등록절차를 거친 디바이스가 서버와 통신하기 위한 상호 인증은 총 4단계이며 Fig. 4와 같다.

첫 번째, 생성한 nonce 값과 자신의 디바이스 id를 서버의 공개키로 암호화한 $c_2$ 그리고 등록 시 부여 받은 패스워드와 nonce 값으로 $P_2 = h(PW \mid N_B)$를 계산하여 서버에 전송한다.

수신한 서버는 자신의 개인키로 복호화 한 디바이스 id와 데이터베이스를 대조하여 해당 디바이스의 패스워드와 임시 패스워드를 조회한다. nonce 값과 두 가지 경우의 패스워드로 해시 값 $P_2$가 검증된다면 기존 패스워드를 임시 패스워드로 바꾸고 새로운 패스워드를 $PW_{new} = PW \oplus N_B$ 로 업데이트한다. 그리고 새로운 패스워드를 해시 한 값을 전송한다.

디바이스는 자신의 패스워드와 자신이 생성했던 nonce 값으로 $P_3$를 검증한다. 검증이 완료되면 서버와 같이 자신의 패스워드를 새로 갱신한 뒤, $P_4 = h(PW_{new} \oplus D_{id})$를 계산하여 전송하고 세션 키 $SK = h(PW_{new} \mid D_{id})$를 설립한다. 마지막으로 서버는 동일하게 해시의 입력 값을 구성하여 $P_4$가 검증되면, 임시 패스워드를 삭제하고 디바이스와 동일하게 세션 키를 설립한다.

## 4. 안전성 분석

### 4.1 디바이스 익명성 및 정보 기밀성

제안하는 프로토콜에서는 등록, 상호 인증 초기에 디바이스의 신원을 추측할 수 있는 디바이스 id를 암호화하고 nonce값으로 인해 암호문도 항상 변한다. 때문에 어떤 디바이스가 어느 정도 통신하고 있는지 추적이 불가능하다. 또한 적합한 사용자만이 송수신되는 정보를 알 수 있어야 한다. 제안하는 프로토콜에서는 중요 정보들이 암호 알고리즘, 해시 함수, XOR 연산을 통해 암호화되기 때문에 적합하지 않은 사용자는 디바이스 id, nonce 값, 패스워드와 같은 정보에 접근할 수 없다.

### 4.2 중간자 공격, 재전송 공격

제안하는 프로토콜에선 디바이스를 등록하고 세션 키를 설립하는 과정 모두에서 인증 메세지 P를 통해 서로의 통신 사실을 확인하고 있기 때문에 중간자 공격에 대한 보안성을 확보할 수 있다. 재전송 공격에 대해서는 많은 프로토콜에서 nonce 값이나 타임스탬프를 활용하고 있다. 하지만 타임스탬프를 사용하는 경우에는 통신대상끼리의 동기화가 필요하기 때문에 제안 프로토콜에서는 nonce값을 활용하였다. 세션 초기에 생성한 nonce값이 암호화 되어 전송되거나, 전송되는 해시 입력 값에 영향을 주기 때문에 모든 메세지에 nonce값이 관여하게 된다. 따라서 이전 세션에서 사용된 메세지의 내용이 그대로 사용 된다면 재전송 공격이라 판단하여 방어할 수 있다.

### 4.3 PFS(Perfect Forward Secrecy)

PFS의 달성을 위해선 개인키가 노출되어도, 과거에 도청당한 통신 기록들의 보안이 지켜져야 한다. 제안하는 프로토콜에선 디바이스가 탈취되어 패스워드가 노출된다 해도 nonce 값을 알아내지 못하면 갱신되기 전의 패스워드를 추적할 수 없기 때문에 이전 통신 기록을 해킹할 수 없다. 따라서 최종적으로 PFS를 달성할 수 있다.

## 5. 비교 분석

### 5.1 프로토콜 성능

Kumar S. Roy의 프로토콜은 암호화 기법으로 Goppa코드 기반의 Niederreiter를 사용한다. Goppa 코드는 암,복호화 속도는 빠르지만 키 사이즈가 매우 크다. Niederreiter의 키 사이즈는 메모리가 적은 8-bit AVR 프로세서에는 저장할 수 없을 만큼 키 사이즈가 크다. 하지만 제안하는 프로토콜에서 사용하는 ROLLO는 적합한 키 사이즈를 제공하며 연산 속도 또한 준수하다.

Kumar S. Roy의 프로토콜에서 사용하는 Niederreiter 암호기법의 128-bit 보안레벨의 키 크기와 제안 프로토콜에서 사용하는 ROLLO-II의 128-bit 보안레벨의 키 크기는 Table 4와 같다.

Table 4. Parameters for ROLLO-II and Niederreiter

| | Security level | Public key | Secret key |
|---|---|---|---|
| ROLLO-IIl | 128-bit | 1941 bytes | 40 bytes |
| Niederreiter | 128-bit | 114,125 bytes | 22,750 bytes |

제안하는 프로토콜의 성능평가를 위해 많은 IoT 디바이스로 활용되고 있는 저전력 ARM 프로세서에서의 성능 측정을 진행 하였다. 또한 IoT 디바이스가 아닌 일반 컴퓨터에서의 통신 속도를 확인해보기 위해 고성능 Intel 프로세서에서도 성능을 측정하였다. ARM 프로세서 환경은 Table 1, Intel 프로세서 환경은 Table 5와 같으며, 이에 대한 결과는 각각 Table 6, 7과 같다.

Table 5. Experiment Environment

| | Intel NUC |
|---|---|
| CPU | Intel Core i5-8259U@3.80 Ghz |
| Memory | 32GB RAM |
| OS | Ubuntu 18.04.3 LTS |

Table 6. Performance in ARM Processor (in ms)

| | Registration | Authentication |
|---|---|---|
| Proposed Protocol | 82 | 41 |

Table 7. Performance in Intel Processor (in ms)

| | Registration | Authentication |
|---|---|---|
| Proposed Protocol | 4.5 | 2 |

암호화에는 많은 연산이 요구되기 때문에 경량 프로토콜 설계를 위해서는 수행되는 암호화 횟수가 중요하다. 암호화 및 복호화에 사용되는 비용이 제일 크기 때문이다. 암, 복호화 횟수에 대한 비교는 Table 8과 같다.

Table 8. Comparison of Encryption Count

| | Registration | Authentication |
|---|---|---|
| Kumar S.Roy's Protocol | 3 | 3 |
| Proposed Protocol | 2 | 1 |

Kumar S.Roy의 프로토콜은 등록 과정에서 3번, 인증 과정에서 3번의 암, 복호화가 수행되는 반면 제안 프로토콜에서는 해시 함수와 XOR연산으로 대체함으로써 등록 과정에서 2번, 인증 과정에서는 1번만 수행된다. 등록 과정은 초기에 각자의 신원을 확인해야 하기 때문에 디바이스에서 1번, 서버에서 1번, 총 2번의 암, 복호화를 수행하였다. 인증 과정에서는 패스워드와 익명성을 위한 디바이스 id 암호화 1번만을 수행하고 이후로는 인증된 패스워드와 해시 함수를 활용하여 서로의 신원을 밝혔다.

### 5.2 프로토콜 안전성

PFS 달성의 측면에서 바라보았을 때, Kumar S. Roy의 프로토콜은 통신과정에서 패스워드, 개인키를 정적으로 사용한다. 때문에 디바이스가 탈취되어 개인키와 패스워드가 노출되었을 때, 과거의 통신 기록이 해킹 당할 수 있다. 하지만 제안하는 프로토콜은 통신 후 nonce 값을 활용하여 패스워드를 업데이트한다. 디바이스의 개인키를 탈취해도 과거의 통신 기록 해킹에 필요한 패스워드를 알 수 없기 때문에 PFS를 달성할 수 있었다.

## 6. 결 론

본 논문에서는 다가오는 양자컴퓨터 시대를 대비하여 NIST 양자내성암호 공모전 Round2를 진행 중인 코드기반 암호들에 대해 비교해보고, 후보군 중 하나인 ROLLO를 활용하여 경량 보안 프로토콜을 설계하였다. 경량형 설계를 위하여 키 사이즈가 작고 연산 속도가 빠른 ROLLO를 사용하였으며 암호화 수행 횟수를 줄이고 해시 함수와 XOR연산을 사용하였다. 제안 프로토콜의 구현 코드는 Github[6]에 공개되어 있으며 향후 연구 방향으로는 프로토콜 안전성 분석에 범용적으로 사용되는 AVISPA 툴[7]을 활용한 자동화 분석과 프로토콜의 경량 설계를 위한 개선을 진행할 예정이다.

## References


[1] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Gen. Comput. Syst*, Vol.29, No.7, pp.1645-1660, 2013.

[2] K. H. Wang, C. M. Chen, W. Fang and T. Y. Wu, "A Secure Authentication Scheme for Internet of Things," *Pervasive and Mobile Computing*, Vol.42, pp.15-26, 2017.

[3] R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory," Technical Report, NASA, 1978.

[4] C. A. Melchor, N. Aragon, M. Bardet, S. Bettaieb, L. Bidoux, O. Blazy, J. C. Deneuville, P. Gaborit, A. Hauteville, A. Otmani, O. Ruatta, J. P. Tillich, and G. Zemor, "ROLLO-Rank-Ouroboros, LAKE& LOCKER," Submission to the NIST Post Quantum Standardization Process, Round 2, 2019.

[5] K. S. Roy and H. K. Kalita, "A Code based Light-weight Authentication Scheme for IoT in Fog Computing Environment," *Jour of Adv Research in Dynamical & Control Systems*, Vol.11, No.6, pp.97-107, 2019.

[6] Github: source code [internet], https://github.com/starj1023/Code-Based-Protocol-ROLLO-

[7] A. Armando, D. Basin, Y. Boichut,Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P. C. Heam, O. Kouchnarenko, J. Mantovani, S. Modersheim, D. V. Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Vigano, and L. Vigneron, "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, Vol.3576, pp.281-285, 2005.



**장 경 배**

https://orcid.org/0000-0001-5963-7127
e-mail : starj1023@gamil.com
2019년 한성대학교 IT응용시스템공학부 (학사)
2019년 ~ 현 재 한성대학교 IT융합공학부 석사과정
관심분야 : 정보보호, IoT, 양자컴퓨터



**심 민 주**

https://orcid.org/0000-0001-5242-214X
e-mail : minjoos9797@gmil.com
2019년 ~ 현 재 한성대학교 IT융합공학부 학사과정
관심분야 : 네트워크 보안, 시스템 보안



**서 화 정**

https://orcid.org/0000-0003-0069-9061
e-mail : hwajeong84@gmail.com
2010년 부산대학교 컴퓨터공학과(학사)
2012년 부산대학교 컴퓨터공학과(석사)
2012년 ~ 2016년 부산대학교 컴퓨터공학과 (박사)
2016년 ~ 2017년 싱가포르 과학기술청 연구원
2019년 ~ 현 재 한성대학교 IT융합공학부 조교수
관심분야 : 정보보호, 암호화 구현, IoT