코드기반암호를 활용한 IoT 환경 보안 프로토콜 설계

장경배*, 심민주*, 서화정* †
*한성대학교 IT 융합공학과
starj1023@gmail.com, minjoos9797@gmail.com, hwajeong84@gmail.com

Design of IoT Environment Secure Protocol Using Code-Based Cryptography

Kyung-Bae Jang*, Min-Joo Sim*, Hwa-Jeong Seo*†
*Dept. of IT Convergence Engineering, Hansung University

요 약

IoT(Internet of Things) 시대가 활성화되면서 개인정보를 포함한 많은 정보들이 IoT 디바이스들을 통해 전달되고 있다. 정보보호를 위해 암호화하여 통신하는 것이 중요하며 성능의 제한으로 인해 경량 보안 프로토콜 사용이 요구된다. 현재 많은 암호 시스템들은 인수분해 그리고 이산대수의 어려움에 기반하고 있다. 하지만 양자 알고리즘이 실현 가능한 양자 컴퓨터가 개발된다면 앞선 문제들을 쉽게 해결할 수 있다. 이에 본 논문에서는 양자내성암호 중 코드기반암호를 사용한 경량 보안 프로토콜을 제안한다. 기존 프로토콜과 비교 분석해보고 안전성 분석 또한 실시하였다.

1. 서론

현대에 있어 IoT[1] 기술의 중요성은 점차 부각되지만 사용자의 개인정보를 보장하고 민감 데이터를 보호하는데 있어 많은 문제와 위험도 따른다. 만약 헬스케어 시스템 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.

현재 ECC(Elliptic Curve Cryptography)를 많은 곳에서 경량 공개키 암호로 사용하고 있으며 대표적으로 2017 년 Wang 의 프로토콜[2] 또한 그렇다. 하지만 양자 컴퓨터가 개발된다면 더이상 사용할 수 없다는 문제점이 있다.

이에 기존 암호시스템들을 무너뜨릴 수 있는 양자 컴퓨터의 계산 능력에 내성을 가진 양자내성암호 연구가 이루어지고 있다. 미국 NIST(National Institute of Standards and Technology)에서는 2016 년 양자내성암호 표준화 공모전을 주최 하였고 세계 여러 각국에서 양자내성암호 알고리즘을 제출하였다. 현재 26 개 후보들이 살아남아 Round2 에 대한 평가를 진행 중이며 코드, 격자, 다변수다항식, 아이소제니 기반암호들로 구성되어 있다.

이에 본 논문에서는 NIST 양자내성암호 공모전을 진행중인 코드기반암호 중 ROLLO 를 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 실시하였다.

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. McEliece 는 1978 년, 최초의 코드기반암호 McEliece[3]를 제안하였다.

McEliece 에서는 Goppa 코드라는 오류수정코드를 사용하는데 현재 NIST 양자내성암호 공모전 Round 2, 7 개의 코드기반암호 중 Classic McEliece 와 NTS-KEM 이 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 5 개의 암호가 Round 2 를 진행중

이다.

2.2 ROLLO

ROLLO[4]는 양자내성암호 표준화 공모전 Round2 를 진행중인 코드기반암호이다. Goppa 코드가 아닌 효율성을 중시한 Rank Metric 코드(1991)를 기반으로 하여 키 사이즈와 계산 복잡도 측면에서 매우 효율적이다. Goppa 코드를 사용하는 Classic McEliece 와 NTS-KEM 과의 성능 비교를 위해 저전력 모바일 프로세서인 ARM 에서 속도를 측정하였다. 실험 환경과 연산 속도 비교 결과는 표 1, 표 2 와 같다.

<표 1> 실험 환경

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

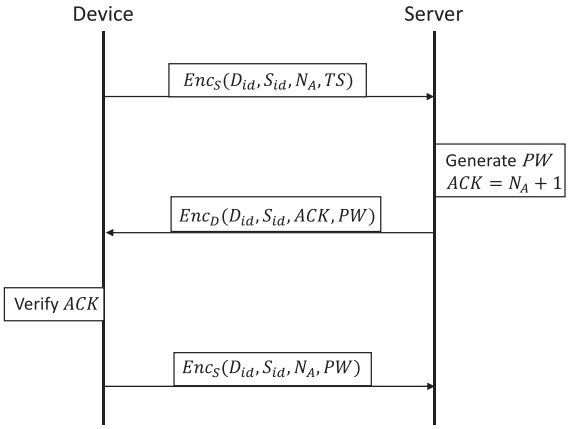
<표 2> 연산 속도(ms) 비교

	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
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

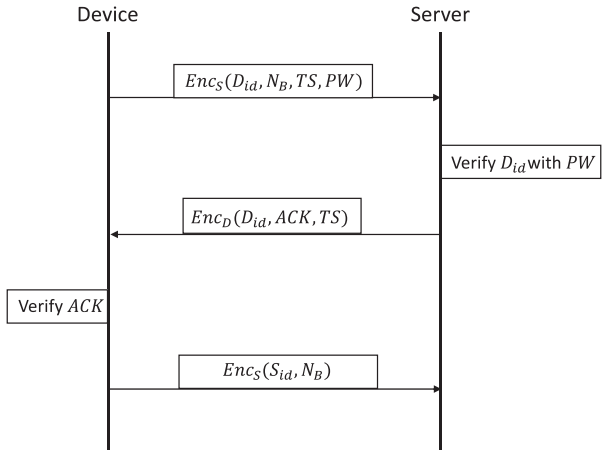
40 년의 역사를 가지고 있는 Goppa 코드에 비하여 Rank Metric 코드는 상대적으로 짧은 검증 기간을 가지고 있지만 높은 성능을 보여준다. 128-bit 보안 레벨 기준으로 Classic McEliece 의 공개키는 261120 바이트, 개인키가 6452 바이트인 반면, ROLLO 의 공개키는 634 바이트, 개인키는 40 바이트로 훨씬 작은 키 사이즈를 제공한다. 본 논문에서 제안하는 프로토콜의 암호화 기법은 ROLLO-I, II, III 중 II 을 선택하였다. ROLLO-I, ROLLO-III 는 KEM(Key Encapsulation Mechanism) 방식으로 자체적으로 세션 키를 설립하지만 ROLLO-II 는 메세지 암호화에 적합한 PKE(Public Key Encryption) 방식을 제공하기 때문이다.

2.3 Kumar S. Roy's Protocol

2019 년 Kumar S. Roy 는 코드기반암호 McEliece 를 사용하여 경량 보안 프로토콜[5]을 제안하였다. 등록과 인증 2 가지 절차로 구성되며 그림 1, 그림 2 와 같다. 암호화는 등록 과정에서 3 번, 인증 과정에서 3 번 수행된다. Kumar S. Roy 의 프로토콜은 5 장에서 보다 세부적으로 살펴보며 제안하는 프로토콜과 성능 및 안정성 측면에서 비교 분석하고자 한다.



(그림 1) 등록 절차



(그림 2) 상호 인증 절차

3. 제안 프로토콜

제안하는 보안 프로토콜은 다음 두가지로 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 II 을 사용하였다.

ROLLO-I, ROLLO-III 는 KEM(Key Encapsulation Mechanism) 방식으로 자체적으로 세션 키를 설립하지만 ROLLO-II 는 PKE(Public Key Encryption) 방식이기 때문이다. 또한 경량 설계를 위해 암호화 횟수를 최소로 수행하고 나머지 연산들은 해시 함수와 XOR 연산 만을 사용하였다. 프로토콜 설명을 위한 표기법은 표 3 과 같다.

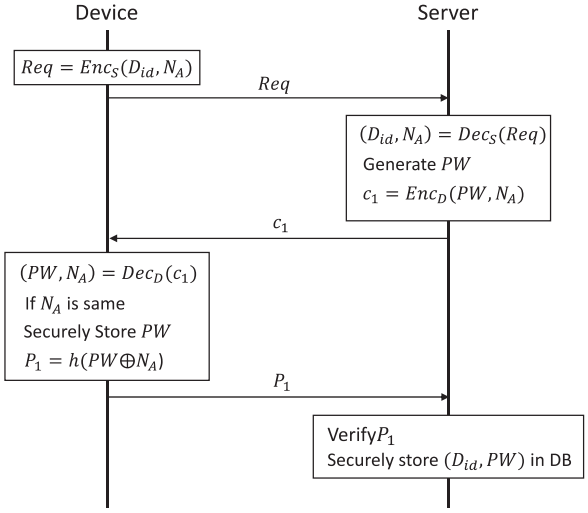
<표 3> 표기법

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 4 단계로 구성되며 그림 3 과 같다.



(그림 3) 등록 절차

첫번째, 서버에 등록을 원하는 디바이스는 자신의 디바이스 id 와 nonce 값을 서버의 공개키로 암호화하여 전송한다.

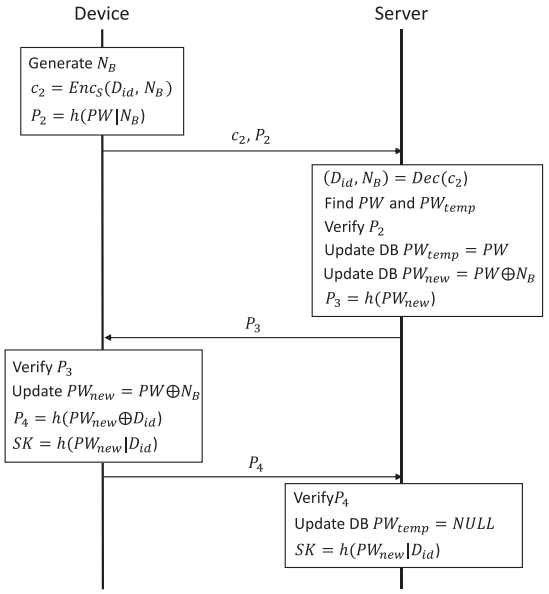
두번째, 서버는 수신한 메세지 Req를 자신의 개인키로 복호화 하여 디바이스 id 와 nonce 값을 확인한 뒤 해당 디바이스를 위한 패스워드를 생성한다. 그리고 서버의 신원확인이 가능한 nonce 값과 생성한 패스워드를 해당 디바이스의 공개키로 암호화하여 전송한다.

세번째, c_1 을 수신한 디바이스는 자신의 개인키로 복구한 nonce 값이 일치한다면 패스워드를 안전하게 저장한다.

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

3.2 디바이스 서버간 상호 인증

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



(그림 4) 상호 인증 절차

첫번째, 생성한 nonce 값과 자신의 디바이스 id 를 서버의 공개키로 암호화한 c_2 그리고 등록 시 부여받은 패스워드와 nonce 값으로 $P_2 = h(PW|N_B)$를 계산하여 서버에 전송한다.

수신한 서버는 자신의 개인키로 복호화 한 디바이스 id 와 데이터베이스를 대조하여 해당 디바이스의 패스워드와 임시 패스워드를 조회한다. nonce 값과 두 가지 경우의 패스워드로 해시 값 P_2가 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_3를 검증한다. 검증이 완료되면 서버와 같이 자신의 패스워드를 새로 갱신한 뒤, $P_4 = h(PW_{\text{new}} \oplus D_{\text{id}})$ 를 계산하여 전송하고 세션 키 $SK = h(PW_{\text{new}}|D_{\text{id}})$를 설립한다.

마지막으로 서버는 동일하게 해시의 입력 값을 구성하여 P_4가 검증되면, 임시 패스워드를 삭제하고 디바이스와 동일하게 세션 키를 설립한다.

4. 안전성 분석

4.1 디바이스 익명성 및 정보 기밀성

제안하는 프로토콜의 등록, 상호 인증 초기에 디바이스의 신원을 추측할 수 있는 디바이스 id 를 암호화하고 nonce 값으로 인해 암호문도 항상 변한다. 때문에 어떤 디바이스가 어느정도 통신하고 있는지 추적이 불가능하다.

또한 적합한 사용자만이 송수신되는 정보를 알 수

있어야 한다. 제안하는 프로토콜에서는 중요 정보들이 암호 기법, 해시 함수를 통해 암호화 되기 때문에 적합하지 않은 사용자는 디바이스 id, nonce 값, 패스워드와 같은 정보에 접근할 수 없다.

4.2 중간자 공격, 재전송 공격

제안하는 프로토콜에선 디바이스를 등록하고 세션키를 설립하는 과정 모두에서 인증 메세지 P 를 통해 서로의 통신 사실을 확인하고 있기 때문에 중간자 공격에 대한 보안성을 확보할 수 있다.

재전송 공격에 대해서는 세션 초기에 생성한 nonce 값이 암호화 되어 전송되거나, 전송되는 해시 입력값에 영향을 주기 때문에 모든 메세지에 nonce 값이 관여하게 된다. 따라서 이전 세션에서 사용된 메세지의 내용이 그대로 사용 된다면 재전송 공격이라 판단하여 방어할 수 있다.

4.3 PFS(Perfect Forward Secrecy)

PFS 의 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 PFS 를 달성할 수 있다.

5. 비교 분석

5.1 프로토콜 성능

Kumar S. Roy 의 프로토콜은 암호화 기법으로 Goppa 코드 기반의 McEliece 를 사용한다. McEliece 의 연산 속도는 빠르지만 키 사이즈가 매우 크다. 대표적으로 Round2 를 진행중인 Classic McEliece 는 8-bit AVR 프로세서에는 저장할 수 없을 만큼 키 사이즈가 크다. 하지만 제안하는 프로토콜에서는 적합한 키 사이즈를 제공하며 연산 속도 또한 준수한 ROLLO 를 사용하였다. ARM 프로세서에서 프로토콜의 성능을 측정하였으며 표 4 와 같다.

<표 4> 성능 측정(ms)

	Registration	Authentication
Proposed Protocol	82	41

암호화에는 많은 연산이 요구되기 때문에 경량 프로토콜 설계를 위해서는 수행되는 암호화 횟수가 중요하다. Kumar S. Roy 의 프로토콜은 등록 과정에서 3 번, 인증 과정에서 3 번의 암호화가 수행되는 반면 제안 프로토콜에서는 등록 과정에서 2 번, 인증 과정에서는 1 번만 수행된다. 등록 과정은 초기에 각자의 신원을 확인해야 하기 때문에 디바이스에서 1 번, 서버에서 1 번, 총 2 번의 암호화를 수행하였다. 인증 과정에서는 패스워드와 익명성을 위한 디바이스 id 암호화 1 번만을 수행하고 이후로는 인증된 패스워드와 해시 함수를 활용하여 서로의 신원을 밝혔다.

5.2 프로토콜 안전성

PFS 달성의 측면에서 바라보았을 때, Kumar S. Roy 의 프로토콜은 통신과정에서 패스워드, 개인키를 정적으로 사용한다. 때문에 디바이스가 탈취되어 개인키와 패스워드가 노출되었을 때, 과거의 통신 기록이 해킹당할 수 있다. 하지만 제안하는 프로토콜은 통신 후 패스워드를 업데이트 하기 때문에 과거의 통신 기록 해킹에 필요한 패스워드를 알 수 없다.

6. 결론

본 논문에서는 다가오는 양자컴퓨터 시대를 대비하여 코드기반암호를 활용한 경량 보안 프로토콜을 설계하였다. 키 사이즈가 작고 연산 속도가 빠른 ROLLO 를 사용하였으며 암호화 수행 횟수를 줄이고 해시 함수와 XOR 연산을 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 경량 설계를 위한 개선을 진행할 예정이다.

참고문헌

[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] Wang KH, Chen CM, Fang W, Wu TY "A secure authentication scheme for internet of things" Pervasive Mobile Comput 42:15–26, 2017.

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

[4] C. A. Melchor, etc "ROLLO –Rank-Ouroboros, LAKE & LOCKER", Submission to the NIST post quantum standardization process, Round 2, 2019.

[5] K. S. Roy, 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, 06-Special Issue, 2019.

[6] Github: source code (Internet). Available: https://github.com/starj1023/Code-Based-Protocol-ROLLO-

[7] Armando A, etc, "The AVISPA tool for the automated validation of internet security protocols and applications. In: International Conference on Computer Aided Verification", Springer, pp 281–285, 2005.