MIRACL을 활용한 인증 및 키 합의 프로토콜의 효율성 분석

조영재, 오지현, 손승환, 김명현, 이준영, 박영호

경북대학교

sunsky1111@knu.ac.kr, chldlstnr071@knu.ac.kr, saqywer@naver.com,

kimmyeong123@knu.ac.kr, harry250@knu.ac.kr, parkyh@knu.ac.kr

Efficiency Analysis of Authentication and Key Agreement Protocol using MIRACL

Cho Yeong Jae, Oh Ji Hyeon, Son Seung Hwan, Kim Myeong Hyun,

Lee Joon Young, Park Young Ho

Kyungpook National Univ.

요 약

최근 정보를 저장하고 통신하는 기기의 범위가 확대됨에 따라 제한된 연산자원을 가진 기기에도 적용할 수 있는 효율적인 인증 및 키 합의 프로토콜에 관한 연구가 활발히 진행되고 있다. 본 논문에서는 공개소스 암호화 라이브러리인 MIRACL (Multi-precision Integer and Rational Arithmetic Cryptographic Library)을 사용해 2020년에 제안된 Yu 등과 Lee 등의 인증 및 키 합의 프로토콜의 실행 시간을 측정하여 각각 유사한 환경의 논문들과 비교분석한다.

Ⅰ. 서 론

최근 임베디드 기술 및 정보통신 기술의 발전으로 다양한 무선 환경에서 데이터를 주고받는 기기의 수가 증가하고 있다. 그러나 무선 환경에서 통신하는 기기들은 공개 채널을 통해 데이터를 교환하므로 보안공격으로부터 안전하지 않다. 또한 연산자원이 제한된 통신기기들로 인해 보안 알고리즘 실행 과정 중 최적화 문제도 발생할 수 있다. 이러한 문제들을 해결하기 위해 안전하고 효율적인 인증 및 키 합의 프로토콜 연구가 국내·외에서 활발히 이루어지고 있다[1,2].

본 논문에서는 2020년 Yu 등이[3] 제안한 무선 센서 네트워크 환경에서의 인증 및 키 합의 프로토콜과 Lee 등이[4] 제안한 차량 클라우드 컴퓨팅 환경에서의 인증 및 키 합의 프로토콜의 실행 시간을 MIRACL[5]로 측정한다. 측정된 결과를 바탕으로 Yu 등과 Lee 등의 논문과 유사한 환경을 지닌 논문들을 통해 각 프로토콜 효율성을 비교분석한다.

Ⅱ. MIRACL

MIRACL은 제한된 연산자원을 지닌 임베디드 및 모바일 환경 등에서 보안 프로토콜을 구축할 수 있도록 설계된 소프트웨어 개발 키트(SDK)로 대칭 키 암호, 단방향 해시 함수 및 타원곡선 암호 등의 암호 연산 소스코드를 포함한 다양한 암호 알고리즘을 제공하는 오픈소스 C/C++ 라이브러리이다. 이러한 특성으로 최근 효율적인 보안 프로토콜 구축을 위한 다양한 연구에서 성능 분석을 위해 MIRACL이 사용되고 있다.

Ⅲ. 암호 연산 실행 및 시간 측정

본 논문에서는 Yu 등과 Lee 등이 제안한 인증 및 키 합의 프로토콜의 효율성 분석을 위해 퍼지 추출, 타원 곡선에서의 덧셈, 타원 곡선에서의 곱셈, AES 암호 및 단방향 해시 함수 연산에 대한 시간을 MIRACL을 활용하여 측정한다. 실행 시간 측정 시 각각의 암호화 연산을 100회 실행하여 평균 실행 시간을 측정한다. 그림 1은 Yu 등과 Lee 등이 제안한 프로토콜에서 사용한 단방향 해시 함수의 실행 시간을 측정하는 소스코드이다. 본 논문에서는 $Ubuntu 18.04\,LTS, 8GB\ memory, Intel\ Core\ i5-10400\ (2.9GHz, 6Core), 64bits$ 환경에서 표준 라이브러리인 $<time.h>$ 헤더 파일의 $Clock()$ 함수를 사용하여 MIRACL 소스코드의 실행 시간을 측정하였다.

```
clock_t start, end;
double time[100];
clock_t start, end;
double sum, average;
for (int x = 0; x < runtime; x++){
    char *test = new char(length);
    for (int z = 0; z < length; z++){
        test[z]=GetRandomCharacter();
    }start = clock();
    shs_init(&sh);
    for (int i=0;test[i]!=0;i++) shs_process(&sh,test[i]);
    shs_hash(&sh,hash);
    end = clock();
    time[x]=(double)(end-start);
    delete[] test;
}sum=0;
for (int j = 0; j < runtime; j++){
    sum+=time[j];
}average=sum/runtime;
```

그림 1. 단방향 해시 함수의 평균 실행 시간을 측정하는 소스코드.

3.1. Yu 등이 제안한 프로토콜

2020년 Yu 등이 제안한 인증 및 키 합의 프로토콜은 그림 2와 같다.

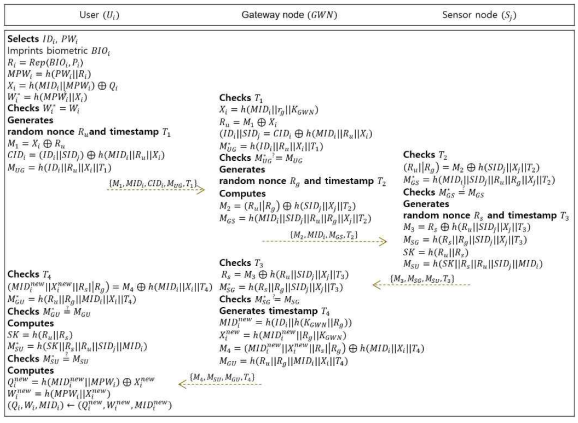

그림 2. Yu 등이 제안한 인증 및 키 합의 프로토콜.

Yu 등은 단방향 해시 함수와 퍼지 추출을 기반으로 한 인증 및 키 합의 프로토콜을 제안하였다. 본 논문에서는 Yu 등의 논문과 유사한 환경에서 에서 제안된 Mo 등의[6] 인증 및 키 합의 프로토콜을 분석하여 효율성을 비교하였다. Mo 등은 Yu 등의 논문과 유사한 환경에서 단방향 해시 함수, 퍼지 추출, 대칭 키 암호화 및 타원곡선에서의 곱셈 연산을 사용하여 인증 및 키 합의 프로토콜을 제안하였다. 표 1은 본 논문에서 MIRACL 소스코드를 사용하여 각 암호화 연산의 실행 시간을 측정한 것이다. 표 1에서 T_h는 단방향 해시 함수, T_R은 퍼지 추출, T_m은 타원 곡선에서의 곱셈 및 T_s는 대칭 키 암호화 연산의 실행 시간을 의미한다.

표 1. 측정 시간

Entity	T_s	T_m	T_R (= T_m)	T_h
Computation cost (ms)	0.0005	0.496	0.496	0.0003

표 1에서 측정된 값을 사용하여 Yu 등과 Mo 등의 프로토콜 실행 시간을 계산한 결과는 표 2와 같다. 실행 시간 비교를 통해 Yu 등이 분석한 결과와 동일하게 Yu 등의 프로토콜이 Mo 등의 프로토콜에 비하여 더 효율적임을 확인하였다.

표 2. 시간 비교

Schemes	Total Computation Cost	
Yu 등의 프로토콜[3] (ms)	$28T_h + T_R$	0.504
Mo 등의 프로토콜[6] (ms)	$27T_h + T_R + 4T_m + 2T_s$	2.489

3.2. Lee 등이 제안한 프로토콜

2020년 Lee 등이 제안한 인증 및 키 합의 프로토콜은 그림 3과 같다.

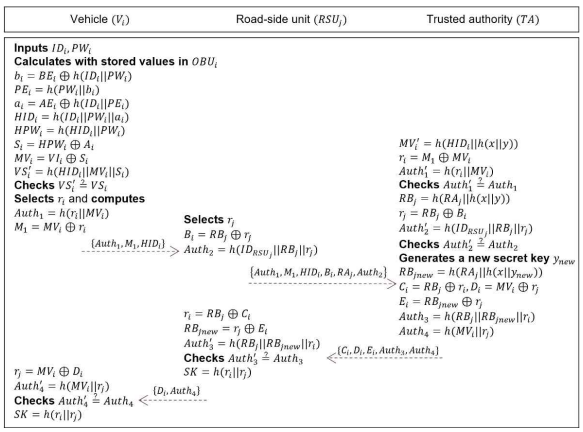

그림 3. Lee 등이 제안한 인증 및 키 합의 프로토콜.

본 논문에서는 단방향 해시 함수를 기반으로 한 Lee 등의 인증 및 키 합의 프로토콜 효율성을 분석하기 위해 Zhong 등의[7] 프로토콜과 실행 시간을 비교하였다. Zhong 등의 인증 및 키 합의 프로토콜은 단방향 해시 함수, 타원 곡선에서의 곱셈 연산 및 타원 곡선에서의 덧셈 연산을 사용하였다. 표 3은 각각의 암호 연산에 대해 본 논문의 환경에서 MIRACL을 활용하여 측정한 실행 시간을 정리한 것이다. 표 3에서 T_{sem}은 타원 곡선에서의 곱셈, T_{ea}는 타원 곡선에서의 덧셈 및 T_h는 단방향 해시 함수의 평균 실행 시간을 의미한다.

표 3. 측정 시간

Entity	T_{sem}	T_{ea}	T_h
Computation cost (ms)	0.001	0.0002	0.00008

표 4는 Lee 등과 Zhong 등이 제안한 인증 및 키 합의 프로토콜의 실행 시간을 본 논문의 환경에서 MIRACL 소스코드를 사용하여 측정한 것이다. 표 4를 통하여 Lee 등이 제시한 분석결과와 같이 Lee 등의 프로토콜이 Zhong 등의 프로토콜에 비하여 더 경량화된 프로토콜임을 분석하였다.

표 4. 시간 비교

Schemes	Total Computation Cost	
Lee 등의 프로토콜[4] (ms)	$22T_h$	0.0018
Zhong 등의 프로토콜[7] (ms)	$5T_{sem} + 3T_h + T_{ea}$	0.0054

Ⅳ. 결론

최근 무선 네트워크 환경에서 제한된 연산자원을 가진 통신기기의 수가 증가함에 따라 효율적인 인증 및 키 합의 프로토콜의 성능 분석을 위해 MIRACL을 활용한 연구가 활발히 이루어지고 있다. 본 논문에서는 MIRACL을 통해 Yu 등과 Lee 등이 제안한 인증 및 키 합의 프로토콜의 효율성을 분석하였다. 이러한 분석을 통하여 Yu 등과 Lee 등이 제안한 인증 및 키 합의 프로토콜이 각각의 논문과 유사한 환경에서 제안된 Mo 등과 Zhong 등의 인증 및 키 합의 프로토콜에 비하여 효율적임을 입증하였다. 본 논문에서는 실험을 통한 효율성 비교분석결과와 Yu 등과 Lee 등이 제시한 효율성 비교분석결과가 동일함을 통해 MIRACL의 신뢰성을 확인하였다. 따라서 MIRACL은 향후 인증 및 키 합의 프로토콜 연구에서 프로토콜의 효율성을 분석하는 과정에 사용하기 적합하다.

참 고 문 헌

[1] Oh, J., Yu. S., Lee, J., Son, S., Kim, M., and Park, Y. “A Secure and Lightweight Authentication Protocol for IoT-Based Smart Homes,” Sensors, vol. 21, no. 4, pp. 1-24, 2021.

[2] Kwon, D., Yu S., Lee, J., Son, S., and Park Y. “WSN-SLAP: Secure and Lightweight Mutual Authentication Protocol for Wireless Sensor Networks,” Sensors, vol. 21, no. 3, pp. 1-23, 2021.

[3] Yu, S., and Park, Y. “SLUA-WSN: Secure and Lightweight Three-Factor-Based User Authentication Protocol for Wireless Sensor Networks,” Sensors, vol. 20, no. 15, pp. 1-26, 2020.

[4] Lee, J., Yu, S., Kim, M., Park, Y., Lee, S., and Chung B. H. “Secure Key Agreement and Authentication Protocol for Message Confirmation in Vehicular Cloud Computing,” Applied Sciences, vol. 10, no. 18, pp. 1-20, 2020.

[5] MIRACL Cryptogtaphic SDK, (https://github.com/miracl/MIRACL).

[6] Mo, J., and Chen, H. “A Lightweight Secure User Authentication and Key Agreement Protocol for Wireless Sensor Networks,” Security and Communication Networks, vol. 2019, 2019.

[7] Zhong, H., Wen, J., Cui J., and Zhang, S. “Efficient Conditional Privacy-Preserving and Authentication Scheme for Secure Service Provision in VANET,” Tsinghua Science and technology, vol. 21, no. 6, pp. 620-629, 2016.