JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 4, pp. 815-825, Apr. 2023

Check for updates

엣지 클라우드 환경에서 IoT 기기들과 엣지 서버 간 블록체인 기반 상호 인증 방법

최 정 희
목원대학교 SW교양학부 교수

A Blockchain-based Mutual Authentication Scheme between IoT Devices and Edge Servers in Edge Cloud Environments

Jeong-Hee Choi

Professor, Stokes College, Division of Software Liberal ArtsCourse, Mokwon University, Daejeon 35349, Korea

[요 약]

사물인터넷 사용 증가로 클라우드 컴퓨팅이 엣지 클라우드 환경으로 진화되고 이에 따라 IoT 기기 간 인증방법의 경량화, 탈중앙화 그리고 악의적 사용자에 의한 잠재적 보안 위협이 중요한 문제점으로 대두되었다. 본 논문에서는 엣지 클라우드에 적합한 블록체인 기반의 경량화된 분산 상호인증방법을 제안한다. 제안한 상호 인증과정에서 사용자의 가상아이디를 이용한 익명성 보장, 무작위 수와 타임스탬프를 사용한 재전송 공격과 중간자 공격에 대한 안전성 그리고 완전한 순방향 비밀성을 보장하였다. 또한, 전자서명으로 부인봉쇄가 가능하며 위장 공격으로부터 안전함을 보였다. 제안 인증 기법은 ECC 암호화 알고리즘 기반의 ECDH 알고리즘과 ECDSA 알고리즘을 활용한 경량화된 분산 상호인증 기법으로 인증 서버로 집중되는 계산을 IoT 기기로 분산하여 중앙 인증 서버의 계산 과정을 최소화하였다. 알고리즘 복잡도 계산 결과 ECDH 알고리즘의 동일 크기의 키 생성시간이 ECC와 RSA보다 적음을 보였다.

[Abstract]

Cloud computing has evolved into an edge cloud environment due owing to the increased use of the Internet of Things (IoT). As a result, lightweight authenticationlightweight of authentication methods between IoT devices, decentralization, and potential security threats by malicious users have emerged as seriousimportant problems. This paper proposes a lightweight distributed mutual authentication method based on a blockchain suitable for edge clouds. The proposed mutual authentication process guarantees anonymity using users' virtual IDs, safety against replay and MITM attacks using random numbers and timestamps, and as well as complete forward secrecy. Additionally, an electronic signature allowed for non-repudiation.In addition, it was possible to Non-repudiation with an electronic signature. The proposed authentication method is a lightweight distributed mutual authentication method that utilizes the ECCECC encryption algorithm-based ECDH algorithm and ECDSA algorithm to distribute the calculation concentrated on authentication servers to IoT devices to minimize the calculation process of the central central authentication server. As a result of algorithm complexity calculation, it was shown that the key generation time of the same size of the ECDH algorithm was less than that of ECC and RSA.

색인어 : 클라우드 컴퓨팅, 엣지 클라우드 컴퓨팅, 사물인터넷, 상호인증, 보안

Keyword : Cloud computing, Edge cloud computing, IoT, Mutual authentication, Security

http://dx.doi.org/10.9728/dcs.2023.24.4.815

Received 14 January 2023; **Revised** 27 February 2023
Accepted 15 March 2023

***Corresponding Author; Jeong-Hee Choi**

Tel: +82-42-829-8235
E-mail: jhchoi@mokwon.ac.kr

Ⅰ. 서 론

최근 클라우드 컴퓨팅 환경은 IoT 기기의 증가로 인하여 기존의 정적 사용자 위주의 클라우드 컴퓨팅 시스템 환경에서 동적 사용자 위주의 엣지 클라우드 컴퓨팅 환경으로 진화하고 있다[1],[2]. 엣지 클라우드 환경을 기반으로 하는 대다수의 IoT 기기들은 높은 이동성과 에너지 사용에 대한 경량화된 시스템이다. 사용자 가까이 있는 IoT 장치들의 수 증가 및 잦은 이동성으로 엣지 클라우드 서버와 IoT 기기 간 데이터 전송 및 인증 요청이 빈번하게 일어나고 있다. 클라우드 컴퓨팅을 이용하는 IoT 기기들의 이용 형태로 기인하여 중앙집중식 인증센터에서 모든 인증을 처리하는 기존 클라우드 컴퓨팅 환경에서는 서버의 과부화 문제, 데이터 전송 지연 문제 등이 발생하게 된다[3]-[5].

기존 클라우드 컴퓨팅 환경은 중앙 인증 서버에서 기기들을 인증하는 중앙집중식 방법을 사용한다. 이때, 인증 서버에 IoT 기기의 악의적 사용자로 인한 위협이 발생하면 전체 클라우드 환경에 큰 영향을 끼치게 된다[6]. 이동성이 높고 경량화된 IoT 기기들이 엣지 클라우드 컴퓨팅 환경에서 기존 중앙집중식 클라우드 컴퓨팅 환경에서 적용되었던 보안(개인정보 보안과 데이터 보안 등) 기법을 그대로 사용하는 것은 적합하지 않다[7].

전통적인 클라우드 컴퓨팅 환경에서 사용하던 인증방법을 엣지 클라우드 환경을 사용하는 IoT 기기들의 인증 시에 사용하게 된다면, 다음과 같은 문제점이 발생한다. 첫째, 기존 클라우드 컴퓨팅 환경에 적용했던 복잡하고 어려운 인증 알고리즘은 경량화된 IoT 기기들에 적합하지 않다. 둘째, 기존에 제안된 인증 알고리즘 대다수가 중앙집중식의 인증센터를 통한 인증확인 방법을 사용하기 때문에 폭발적인 수 증가와 이동성이 잦은 IoT 기기들은 현재 클라우드 환경의 중앙집중식 인증방법은 서버의 과부화, 데이터 전송 지연 그리고 보안의 취약점을 야기시킨다. 셋째, 수많은 IoT 기기들은 각기 다른 키 생성 알고리즘을 수행하기 때문에 성능과 보안에 있어서 많은 차이를 갖는다. 따라서 엣지 클라우드 컴퓨팅 환경에서 IoT 기기와 엣지 서버 간 통신할 때 개인정보보호와 데이터 보호에 효율적인 경량화된 새로운 형태의 분산 인증과 익명성을 보장하는 상호인증 기법이 필요하다[8],[9].

Ibrahim(2016)은 탈중앙화의 상호인증 기법을 제안[10]하였으나 인증과정에서 사용자 ID를 공개된 채널로 주고받기 때문에 익명성이 보장되지 않는다. Viet et al.(2005)이 제안한 PIR(Private Information Retreval:개인정보검색) 기법은 전체 데이터베이스를 전달하여 사용자를 탐지하는 방법이다[11]. 제안 기법은 익명화된 비밀번호를 사용하지만, 중앙서버에 사용자 데이터가 저장되고 저장된 전체 데이터베이스를 전달하여 사용자를 탐지하는 중앙집중식 방법을 사용하기 때문에 IoT 기기인증에는 적합하지 않다. Faraz et al.(2014)의 CUA(Client User Authentication)과 MDHA(Modified Diffie Hellmann)을 사용한 기법으로 인증 프로세스가 클라우드 서버와 분리되어 있을 뿐 인증 서버에서 사용자 인증을 수행한다[12]. Sandeep et al.(2014)은 ID 기반 공개키 암호화를 이용해 생성된 공유키로 상호인증하는 기법을 제안하였다[13]. 그러나 이 기법은 익명성을 보장하지 않으며 통신 전 인증서버에 사용자 ID를 등록하여 사용하기 때문에 중앙집중식 인증방법을 사용하고 있다. Seyed et al.(2016)은 공개키 기반의 상호인증 기법을 제안[14]하였으나, 서버에 사용자 등록시 사용자 이름을 ID로 사용하고 있어서 익명성을 보장하지 않는다. Hammi et al.(2018)이 제안한 인증 기법은 인증센터(CA)를 통한 중앙집중식 인증방법을 제안[15]하였으나 이 인증 기법은 IoT 기기 환경의 상호인증에는 비효율적이다.

현재까지 제시된 인증 관련 연구는 다음과 같은 문제점이 존재한다.

첫째, IoT 기기 및 사용자의 익명성을 완벽하게 보장하지 않아 개인정보 노출과 관련된 문제점이 존재한다.

둘째, 중앙집중식 인증 처리에 따른 서버의 과부화, 전송지연, 악의적 사용자에 의한 잠재적 보안 위협 등의 문제점이 존재한다.

따라서, 이러한 문제점들을 해결하기 위한 추가적인 방안이 필요하다.

본 논문은 중앙집중식 인증방법의 단점을 개선한 블록체인 기반의 분산 인증방법으로 ECC 암호화 알고리즘 기반의 ECDH 알고리즘을 응용한 키 생성 및 교환과 ECDSA 알고리즘을 응용한 전자 서명 및 증명 알고리즘을 사용하여 엣지 컴퓨팅 환경에 적합한 경량화된 상호인증 기법을 제안하였다. 또한, 제안한 상호인증 기법은 등록단계에서 사용자는 가상 아이디를 사용하기 때문에 사용자의 익명성을 보장하였다.

본 논문의 구성은 2장에서는 기존에 제안된 IoT 기기를 위한 다양한 인증방법들을 분석하고 3장에서는 본 논문에서 제안하는 블록체인 기반의 IoT 기기와 엣지서버 간 경량화된 인증 기법을 기술한다. 4장에서는 제안 기법의 보안 안전성에 대한 평가와 성능분석을 기술하고 5장에서는 향후 연구 방향 및 결론을 기술한다.

Ⅱ. 본 론

이 장에서는 타원곡선 암호화 알고리즘을 기반으로 하는 타원곡선 디피-헬만 알고리즘과 타원곡선 디지털 서명 알고리즘을 살펴보고, 기존에 제안된 IoT 기기 간 상호인증 기법들을 분석한다.

2-1 엣지 클라우드 환경에서 IoT 기기

엣지 클라우드란 주변 기기에서 생성된 데이터를 클라우드 중앙에서 처리하지 않고, 사용자 또는 데이터 소스의 물리적

위치나 그 근처에서 컴퓨팅을 수행할 수 있는 새로운 데이터 처리 패러다임이다[16]. 엣지 컴퓨팅은 데이터 발생 근처에서 처리하기 때문에 데이터 전송 시에 발생하는 지연시간을 감소, 클라우드 서버의 데이터 처리 비용 절감의 효과 그리고 클라우드 서버 보안 위협이 감소된다.

엣지 클라우드에서 사용되는 IoT 기기는 데이터를 수집 및 처리하는 IoT 기기와 IoT 기기가 처리한 데이터를 분석하여 서버로 전송하는 엣지 기기로 구분된다.

엣지 기기는 네트워크 엣지의 원격 위치에 있는 물리적 하드웨어로 실시간으로 데이터를 수집, 처리, 실행할 수 있을 만큼 충분한 메모리, 프로세싱 성능, 컴퓨팅 리소스를 갖춘 장치다.

IoT 기기는 인터넷에 연결되어 데이터 소스인 물리적 오브젝트이며, 데이터 수집 및 처리되는 위치가 바로 엣지 기기다. 짧은 대기 시간(millisecond:ms) 안에 의사 결정을 내리고 데이터를 처리할 수 있을 정도로 충분한 스토리지와 컴퓨팅 성능을 갖추고 있는 엣지 기기는 IoT의 일부로 간주된다.

엣지 클라우드에서 IoT 기기와 엣지 컴퓨팅의 결합은 IoT 장치와 중앙 IT 네트워크 간의 통신 대기 시간 감소, 응답 시간 단축으로 운영 효율성 향상, 네트워크 대역폭 향상, 네트워크 연결이 끊어져도 시스템이 오프라인에서 계속 작동 가능 마지막으로, 분석 알고리즘 및 머신 러닝을 통한 로컬 데이터 처리, 집계 및 신속한 의사 결정 등의 장점을 갖는다.

2-2 암호화 알고리즘

1) 타원곡선 암호화 알고리즘(ECC Algorithm)

타원곡선 암호화(ECC:Elliptic Curve Cryptography)는 타원 곡선 계산에서 소수(prime)로 나눈 나머지를 곱하는 것은 간단하지만 나눗셈은 사실상 불가능한 이산 로그 문제를 이용한 암호화 방법이다[17]. 즉, 타원곡선에서 두 점 P와 Q가 주어졌을 때, $Q = kP$의 계산에서 k를 만족하는 수를 구하기 어려운 문제를 이용한 암호화 방법이 이산대수 문제를 이용한 공개키 암호화 방식인 타원곡선 암호화(ECC) 알고리즘이다.

2) 타원곡선 디피-헬만 알고리즘(ECDH Algorithm)

타원곡선 디피-헬만 알고리즘(ECDH:Elliptic Curve Diffie-Hellnam Algorithm)은 타원곡선 암호화 알고리즘을 활용한 키 교환 기법으로 메시지 송·수신 시 사용되는 세션키를 생성할 수 있는 대칭키 암호화 방법이다[18]. 두 노드가 세션키를 만들 때, 하나의 노드 E_1이 알려진 곡선의 한 점 G와 임의로 선택한 소수 d_1를 곱하여 e_1을 생성하여 상대 노드 E_2에 전송하고, 노드 E_2는 알려진 곡선의 한 점G와 임의로 선택한 소수 d_2를 곱하여 e_2을 생성하여 상대 노드 E_1에 전송한다. 노드 E_1은 자신이 받은 e_2에 d_1를 곱하여 세션키 $SK=(d_2 \cdot G \cdot d_1)$를 만들고, 노드 E_2은 자신이 받은 e_1에 d_2를 곱하여 세션키 $SK=(d_1 \cdot G \cdot d_2)$를 만든다. 즉, 두 노드는 상대가 선택한 임의의 소수(prime number)를 알지 못해도 동일 세션키를 생성할 수 있다는 개념의 키 교환 기법 알고리즘이다.

3) 타원곡선 디지털 서명 알고리즘(ECDSA Algorithm)

타원곡선 디지털 서명 알고리즘(ECDSA:Elliptic Curve Digital Signature Algorithm)은 서명 증명 알고리즘과 검증 알고리즘으로 구성되어 있다[19]. 타원곡선 디지털 서명 알고리즘에서 서명자와 증명자는 타원곡선 방정식 $E(Z_p)$과 타원 곡선상의 한 점 G, 곡선 순서(차수) P 그리고 서명자의 공개키 $PK=dG$(d는 서명자의 개인키)를 공유하고 있다. 서명 증명의 알고리즘은 서명자는 메시지(m)를 해시값 $h(m)$으로 만든다. 해시값은 0부터 $P-1$까지 하나의 정수이다. 그리고 1에서 $P-1$ 사이 수에서 난수 k를 선택해서 점 kG를 계산하여 한 점을 얻는다. 이 점(x,y)의 x좌표를 이용하여 $r = x \bmod P$과 $s=(h(m)+rd)/k \bmod P$를 계산하여 서명 증명의 키 쌍 $signature\ (r,s)$를 구한다. 서명 검증 알고리즘은 서명자의 공개키를 이용하여 서명의 유효성을 검증한다. 가장 먼저 $signature\ (r,s)$와 메시지 해시값$h(m)$의 유효성을 검증하기 위해 s의 역원($w=s^{-1}$)을 계산하여 $u=w \cdot h(m)$을 구한 후 $v=wr$을 계산한다. 여기서 u와 v를 이용하여 Q를 ($Q=uG+vPK$)과 같이 계산할 수 있다. 공개키 PK는 서명자의 개인키 d와 생성자 G의 곱이므로 ($uG+vdG=(u+vd)G$)와 같이 나타낼 수 있다. 결국 u와 v를 실제 값을 대입하여 정리하면 서명자가 서명 증명할 때 생성한 난수 k와 같고, $uG+vdG$는 kG와 같다. 여기서 kG의 x'좌표값이 서명에서 사용된 x좌표값과 일치한다면 검증 유효성이 증명된다.

타원 곡선 암호화(ECC) 기반의 공개키 암호화 방식은 두 개의 키 쌍(공개키, 비밀키)으로 이루어진 방식으로 디지털 서명에 사용된다. 비밀키를 이용하여 디지털 서명하고, 공개키로 서명 증명하기 때문에 비밀키를 모르면 디지털 서명을 생성할 수 없다.

2-3 기존 연구

공개키 암호화 방식은 두 개의 키 쌍(공개키, 비밀키)으로 이루어진 방식이다. 기존 클라우드 컴퓨팅에서 사용된 암호화 및 인증은 공개키 방식(PKI)의 제3자의 신뢰성 있는 인증센터로부터 부여받은 키를 사용하였다. 이러한 중앙집중식 암호화 방법은 하나의 악의적인 IoT 기기의 접근이 전체 클라우드 시스템에 치명적 위협이 될 수 있다. 표 1은 기존 다양한 인증 기법에 대한 분석을 표로 나타낸 것이다.

Yao et al.(2019)에서 제안한 기법은 이동수단에 적용되는 인증 기법을 제안했다[20]. 그러나 이동수단 간 상호인증이 보장되지 않는 중앙집중식의 인증방법을 사용하기 때문에 이동성이 잦은 이동수단의 인증 시에는 인증 서버 과부화 및 인증 서버에 인증 요청이 집중되면서 발행하는 보안의 위협

이 증가한다. 또한, 제안 기법은 인증과정에서 악의적인 제3자의 정상적인 사용자 아이디 탈취 및 도청 후 재전송 공격을 시도할 수 있어서 IoT 기기뿐 아니라 인증 서버의 보안에도 큰 위협이 될 수 있다.

Amor et al.(2017)에서는 엣지 포그 클라우드 환경에서의 타원곡선 암호화 알고리즘 기반의 인증 기법을 제안하였다[21]. 제안된 인증 기법은 상호인증, 도청, 중간자 공격 등의 보안에 취약한 부분을 개선하였으나 상호인증 과정에서 부인봉쇄가 보장되지 않기 때문에, 탈중앙화된 엣지 클라우드 컴퓨팅 환경에서 IoT 기기 간 상호인증에 필요한 부인 방지를 보장하지 못한다.

Ibrahim(2016)에서는 상호인증을 위한 기법을 제안하였다[10]. 제안된 상호인증 기법은 기기 혹은 사용자의 ID가 공개 채널을 통해 빈번히 사용되어 익명성이 보장되지 않는다. 따라서 개인정보보호에 민감한 개인 사용자들에게는 치명적인 문제를 일으킬 수 있으며, 더 나아가서는 서버에도 위협이 될 수 있다.

Hammi et al.(2018)에서 제안한 인증 기법은 인증센터(CA)를 통한 중앙집중식 인증방법이다[15]. 그러나 제안된 인증 기법은 기기 간 상호인증이 이루어지지 않으며, 인증센터(CA)로 아이디를 직접 등록하기 때문에 익명성이 보장되지 않는다. 전통적인 클라우드 컴퓨팅 시스템에 적합한 인증방법으로 엣지 클라우드 컴퓨팅 환경의 IoT 기기 간 상호인증에는 적합하지 않은 인증방법이다.

Zhang et al.(2021)에서는 등록과정에서 사용자의 ID를 관리 시스템에 전송하고 관리자는 사용자에 익명성이 보장되는 GID(Global unique Identifier)를 부여한다[22]. 사용자는 부여받은 GID를 이용하여 인증 및 세션키 생성 시에 사용된다. 그러나 관리자로부터 GID를 부여받는 과정에서 도청 또는 중간자 공격 등으로 인한 사용자 아이디(GID)의 보호가 보장되지 않는다. 또한, 관리자로부터 사용자 익명성을 위한 아이디(GID)를 부여받기 때문에 IoT 기기의 수가 증가한다면 아이디를 부여하는 관리 시스템의 과부하 발생으로 성능의 안정성이 보장되지 않는다.

Li et al.(2019)에서 제안한 인증 기법은 제3자에 암호화 키를 위탁(key escrow)하는 과정에서 발생하는 문제를 해결하기 위해 신원 기반 암호화(IBE:Identity-based Encryption)의 변형된 무인증암호화 방법을 제안했다[23]. 제안 인증 기법에서는 키 위탁 문제는 해결하였지만, 여전히 사용자의 익명성 보장과 사용자 간 상호인증은 보장되지 않는 문제가 있다.

표 1. 기존 연구 분석
Table 1. Analysis existing research

	Mutual Authentication	Anonymous	Non-Repudiation	Decentralization	MITM
Yao et al.	X	○	X	X	X
Amor et al.	○	○	X	○	○
Ibrahim	○	X	X	○	X
Hammi et al.	X	X	X	X	○
Zhang et al.	○	○	○	○	X
Li et al.	X	X	X	○	○

○:Satisfaction, X:Non-Satisfaction

Ⅲ. IoT 기기들과 엣지 기기들 간 상호인증

이 장에서는 본 논문이 제안한 엣지 클라우드 환경에서 IoT 기기들과 엣지서버들 간 동일 네트워크에서 상호인증 방법과 서로 다른 네트워크에서 상호인증 방법을 기술한다.

3-1 엣지 클라우드 환경

1) 구성 요소

등록서버(Registration Server:*RS*)는 엣지서버(Edge Server:*ES*)들과 단말노드(End Node:*EN*)들의 아이디(*ID*) 또는 가상 아이디(*VID*)를 블록체인 시스템에 트랜잭션을 등록하고 주변 엣지서버(*ES*)들에 새로 등록된 단말노드(*EN*)의 가상 아이디(VID_j)를 broadcast한다. 또한, 등록서버(*RS*)는 처음 등록한 *ES*들과 *EN*들에 타원곡선 알고리즘 $E(Z_p)$과 G(타원곡선의 임의의 한 점)와 P(비밀키 크기 값)를 튜플 $T(G, P)$의 형태로 공유한다.

엣지서버(Edge Server:*ES*)는 자신의 ID_i를 이용하여 *RS*에 등록하고 동일 네트워크에 있는 최종 단말 IoT 기기들인 *EN*들의 가상 아이디(VID_j), 네트워크 그룹 아이디(GID_k) 그리고 $T(G, P)$를 받아 블록체인 시스템에 트랜잭션을 기록한다.

단말노드(End Node:*EN*)가 자신의 VID_j를 이용하여 *RS*에 등록하면 *RS*는 *EN*이 연결 요청한 엣지서버 아이디(ID_i)와 동일 네트워크 엣지서버들의 그룹 아이디(GID_k)를 *EN*에

전송한다. EN은 전송받은 엣지서버 아이디(ID_i)와 엣지서버 그룹 아이디(GID_k)를 자신의 기기에 저장한다.

블록체인 시스템(Blockchain System)은 등록서버와 엣지서버들의 등록 및 거래 관련 트랜잭션을 기록하는 시스템으로 등록 서버에 등록된 단말노드들의 가상 아이디를 분산 저장한다. 등록서버와 엣지서버들은 각자의 블록체인 시스템 환경을 갖추고 있다. 표 2는 본 논문에서 사용되는 표기법이다.

표 2. 표기법
Table 2. Notations

Notation	Meaning
$E(Z_P)$	Elliptic Curve Equation
$T(\ )$	Tuple
G	An arbitrary point on an elliptic curve
P	Order of G
PuK	Public key
PrK	Private key
$PuK\{\}$ $PrK\{\}$	Encryption with Public key or Private key
EN	End Node
ES	Edge Server
RS	Registration Server
$N,\ R$	RandomNumber
TS	Timestamp

2) ECDH기반 키 생성

엣지기기들과 단말노드들의 공개키와 비밀키는 타원 곡선 암호 알고리즘(ECC Algorithm: Elliptic Curve Cryptography)을 기반으로 하는 타원 곡선 디피-헬만(ECDH algorithm: Elliptic Curve Diffie-Hellman) 키 합의 암호화 알고리즘을 이용하여 생성된다. 엣지기기와 단말노드가 암호화된 메시지를 주고받을 때 암·복호화에 사용되는 공개키 PuK_1와 개인키 PrK_1는 식 (1)과 같이 생성된다.

$$K_1 = d_1 G \quad (1)$$
$$d_1 \in \{1, 2, 3, \ldots, n-1\}$$

$$PrK_1 \leftarrow d_1$$
$$PuK_1 \leftarrow K_1$$

이때, 사용되는 G는 타원 곡선 방정식 $E(Z_p)$의 한 점으로 송·수신자 모두 공유하고 있는 한 점의 값이다. K_1은 G에 d_1을 곱한 수로 사용자의 공개키(PuK_1)가 된다. d_1은 사용자가 선택한 임의의 소수(prime)로 곡선 점의 수(차수) n보다 작은 수로 사용자의 비밀키(PrK_1)가 된다.

3-2 단말노드와 엣지서버 등록

엣지서버(ES_a)와 IoT 단말 노드(EN_j)는 통신을 위해 등록서버(RS)에 자신의 아이디 또는 가상 아이디를 그림 1. 과 같이 등록한다.

EN_j의 공개키와 비밀키는 타원곡선 암호화 알고리즘을 기반으로 생성된다. 타원 곡선상 임의의 한 점P와 난수생성기로 생성된 소수(Prime) d의 곱으로 만들어진 Q에서 d는 비밀키(PrK_j), Q는 공개키(PuK_j)로 사용된다. 이때 d는 모듈러 $(\mathrm{mod}\, p)$ p보다 작은 소수(Prime)로 유한체 $GF(p)$상에 있다. 공개키와 비밀키는 식 (2)와 같이 생성된다.

$$Q(x,y) = dP(x_0, y_0)\ (\mathrm{mod}\, p) \quad (2)$$

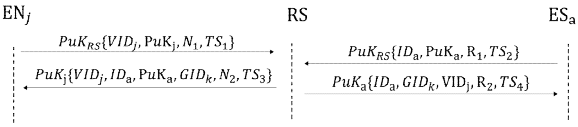

그림 1. RS 등록단계
Fig. 1. Registration phase to RS

IoT 단말노드 EN_j의 가상 아이디 VID_j 생성은 식 (2)의 개인키(d)와 랜덤한 수(r_j)의 곱의 해시값으로 식 (3)과 같다.

$$VID_j = H(d \cdot r_j) \quad (3)$$

EN_j는 등록서버(RS)에 자신의 가상 아이디(VID_j), 자신의 공개키(PuK_j), 무작위 수(N_1) 그리고 타임스탬프(Timestamp: T_1)을 RS의 공개키(PuK_{RS})로 암호화하여 등록 요청한다.

ES_a의 공개키(PuK_a)와 비밀키(PrK_a)는 식(1)과 동일한 방법으로 생성한다. 엣지서버(ES)는 자신의 아이디(ID_a), 공개키(PuK_a), 무작위 수(R_1) 그리고 타임스탬프(T_2)를 등록서버(RS)의 공개키(PuK_{RS})로 암호화하여 등록 요청한다.

새로운 EN과 ES의 등록을 완료한 RS는 EN_j과 ES_a에 타원 곡선 방정식 $E(Z_p)$과 타원 곡선상의 한 점 G 그리고 곡선 순서(차수) P를 튜플 $T(G,P)$ 값으로 전달한다. 엣지서버(ES_a)는 자신의 블록체인 시스템에 $E(Z_p)$와 $T(G,P)$의 트랜잭션 그리고 EN_j의 가상 아이디(VID_j)의 트랜잭션을 저장한다. 단말노드(EN_j)는 $E(Z_p)$와 $T(G,P)$ 값을 자신의 단말기에 저장한다. 등록서버(RS)는 새로운 단말노드(EN)의 등록이 완료되면 주변 엣지서버(ES)들에 단말노드(EN)의 가상 아이디(VID)를 broadcast하고, 엣지서버(ES)들은 새로 등록된 단말노드(EN)의 가상 아이디(VID) 등록과 저장 트랜잭션을 자신의 블록체인에 기록한다.

3-3 블록체인 기반 단말노드와 엣지서버 간 상호인증

1) 단말 노드와 엣지 서버 간 상호인증

등록서버(RS)에 등록이 정상적으로 완료되었다면 단말노드(EN_j)와 엣지서버(ES_a)는 상호인증 단계를 거쳐 메시지 암호화에 사용될 세션키SK_{ja}을 생성하고 생성된 세션키(SK_{ja})로 암호화된 메시지 $SK_{ja}\{M\}$를 전송한다. 상호인증과 세션키 생성은 그림 2와 같은 절차로 진행된다.

단말노드 EN_j는 등록 후 등록서버(RS)로 부터 받은 엣지서버 아이디(ID_a), 엣지 클라우드 그룹 아이디(GID_k)와 자신의 가상 아이디(VID_j), 자신의 공개키(PuK_j), 무작위 수(N_1) 그리고 타임스탬프(TS_1)를 엣지서버(ES_a)의 공개키(PuK_a)로 암호화하여 메시지(M_1)를 생성한다. 생성된 메시지(M_1)는 엣지서버(ES_a)로 전송되고 상호인증 요청이 시작된다.

에지서버 ES_a는 단말노드(EN_j)로부터 받은 메시지(M_1)을 자신의 개인키(PrK_a)로 복호화하여 단말노드(EN_j)의 가상 아이디(VID_j)가 자신의 블록체인 시스템에 기록된 아이디인지 확인한다. 확인이 정상적으로 이루어지면 송수신 측의 아이디, 무작위 수(R_1), 자신의 전자 서명 $sign(R_1, PrK_a, P)$ 그리고 타임스탬프(TS_2)를 단말노드(EN_j)의 공개키(PuK_j)로 암호화하여 메시지 M_2를 생성한 후 단말노드(EN_j)에 전송한다.

단말노드(EN_j)는 수신 메시지 M_2를 자신의 개인키(PrK_j)로 복호화하여 송신측 아이디(ID_a)와 엣지 서버의 전자 서명 $sign(R_1, PrK_a, P)$을 확인한다. 엣지서버(ES_a)의 전자 서명이 정상적인 것으로 확인되면 무작위 수(N_1)와 자신의 전자 서명 $sign(N_1, PrK_j, P)$을 엣지서버의 공개키(PuK_a)로 암호화하여 메시지 M_3를 생성한 후, 엣지서버(ES_a)로 전송한다.

엣지서버(ES_a)는 자신의 개인키(PrK_a)로 단말노드(EN_j)로 부터 수신한 메시지 M_3을 복호화하여 단말노드(EN_j)의 전자 서명 $sign(N_1, PrK_j, P)$이 정상적인 전자 서명임을 확인한다.

전자서명을 통한 상호인증이 완료되면 단말노드 EN_j와 엣지 서버 ES_a 간 생성된 세션키SK_{ja}로 메시지를 암호화하여 통신한다.

상호인증 단계에서 사용되는 전자 서명은 ECDSA(Elliptic Curve Digital Signature Algorithm)을 기반으로 식 (4)와 같이 전자 서명하고, 식 (5)와 같은 절차로 서명증거의 유효성 검사를 진행한다.

그림 2에서 엣지서버 ES_a는 무작위 수 R_1, 개인키 PrK_a 그리고 곡선 순서 P를 이용하여 식 (4)와 같이 생성된 전자서명 $signature\{x,s\}$이 단말노드 EN_j에 전달한다. 여기서 x는 $X(x,y)$의 x좌표값이고, s는 서명증거가 된다. 이때, 중요한 것은 전자 서명은 비밀키를 이용하여 생성하기 때문에 비밀키를 모르는 경우는 전자 서명을 만들 수 없다. 만약, 잘못된 비밀키로 만들었다면 식 (5)의 유효성 검사에 유효하지 않은 결과를 얻게된다.

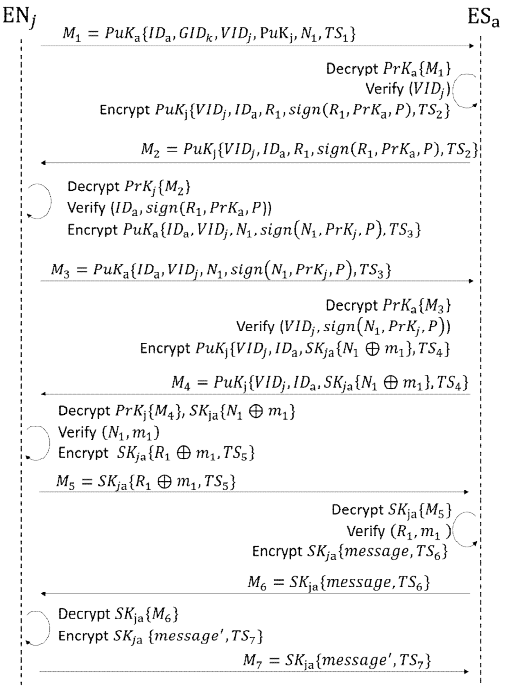

그림 2. EN_j와 ES_a간 상호 인증

Fig. 2. Mutual Authentication between EN_j and ES_a

$$sign\ (R_1, PrK_a, P) \tag{4}$$

$$K = h(R_1) + PrK_a$$
$$x \leftarrow \underline{X}(x,y) = K \cdot G$$
$$s = \overline{K} \cdot (h(R_1) + x \cdot PrK)(mod\ P)$$
$$signature\{x, s\}$$

엣지서버 ES_a의 전자 서명 유효성을 확인하기 위해서 EN_j는 엣지서버 ES_a로부터 받은 서명증거 s를 엣지서버(ES_a)의 공개키 PuK_a를 이용하여 다음 식 (5)와 같은 방법으로 전자 서명의 유효성 검사를 한다.

$$w = s^{-1}\ mod\ p \tag{5}$$
$$u_1 = w \cdot h(R_1) mod\ p$$
$$u_2 = w \cdot x\ mod\ p$$
$$X' = u_1 \cdot G + u_2 \cdot PuK_a$$
$$x' \leftarrow X'(x', y')$$

$$x' = x \rightarrow valid$$
$$x' \neq x \rightarrow invalid$$

엣지서버 ES_a가 전송한 전자 서명 x와 단말노드 EN_j가 복구 후 얻은 x'를 비교하여 전자 서명의 유효성을 검사한다.

2) 동일 네트워크에서 상호인증

IoT 환경에서 수많은 단말노드는 이동과 함께 서로 통신을 요청하고 요청받는다. 단말노드 EN_j가 동일 네트워크 환경에 있는 새로운 엣지서버인 ES_d와 통신하기 위한 상호인증 과정은 그림 3과 같다.

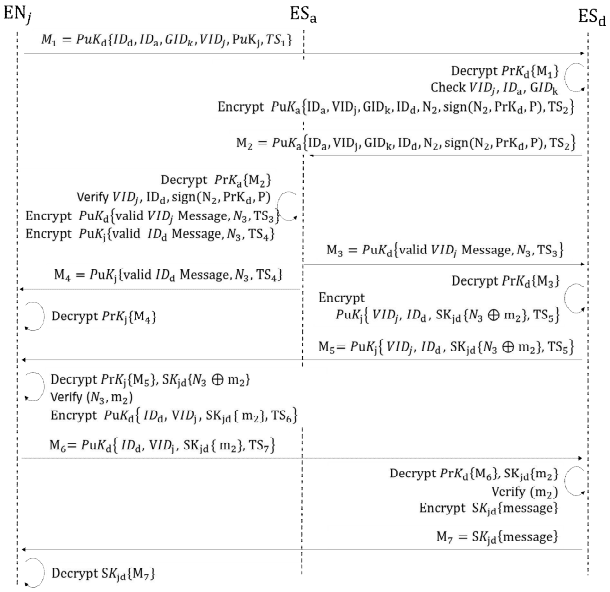

그림 3. 인터 클라우드 내에서 상호인증
Fig. 3. Mutual authentication in intra-cloud

단말노드 EN_j는 엣지서버 ES_d에 자신의 가상 아이디 VID_j와 등록서버(RS)에 등록과 동시에 부여받은 엣지 클라우드 그룹 서버 아이디 GID_k, 엣지서버 아이디 ID_a와 자신의 공개키 PuK_j를 엣지서버 ES_d의 공개키 PuK_d로 암호화하여 전송한다.

엣지서버 ES_d는 자신의 블록체인 시스템에 기록된 단말노드들의 아이디를 확인하고 EN_j와 일치된 기록이 없다면 엣지서버 ES_a에 단말노드 EN_j의 인증을 요청한다. 이때, 인증 요청 메시지 M_2는 엣지서버 ES_d 자신의 전자 서명 $sign(N_2, PrK_d, P)$과 함께 EN_j의 가상 아이디 VID_j를 엣지서버 ES_a의 공개키로 암호화하여 생성한 후, 엣지서버(ES_a)에 전송한다.

엣지서버 ES_a는 엣지서버 ES_d의 전자 서명$sign(N_2, PrK_d, P)$과 단말노드 EN_j의 가상 아이디 VID_j의 유효함을 확인한 후, 메시지 M_3=$PuK_d\{valid\ VID_j message, N_3, TS_3\}$와 메시지 M_4=$PuK_j\{valid\ ID_d message, N_3, TS_4\}$를 엣지서버 ES_d와 단말노드 EN_j에 각각 전송한다.

단말노드 EN_j와 엣지서버 ES_d는 세션키 교환 방식에 따라 세션키 SK_{jd}를 생성한다. 그리고 엣지서버 ES_d는 엣지서버 ES_a로부터 받은 무작위 수 N_3와 자신이 선택한 무작위 수 m_2의 XOR연산 결과를 세션키 SK_{jd}로 암호화하여 메시지 $SK_{jd}\{N_3 \oplus m_2\}$를 단말노드 EN_j로 전송한다.

단말노드 EN_j는 엣지서버 ES_d로부터 받은 메시지 $SK_{jd}\{N_3 \oplus m_2\}$를 세션키 SK_{jd}로 복호화한다. 그리고 $(N_3 \oplus m_2 \oplus N_3)$와 같이 ES_a로부터 받은 무작위 수 N_3와 XOR연산을 통해 얻어낸 m_2를 세션키로 $SK_{jd}\{m_2\}$ 암호화하여 엣지 서버 ES_d로 전송한다.

엣지서버 ES_d는 단말노드 EN_j으로부터 받은 암호문 $SK_{jd}\{m_2\}$을 세션키로 복호화하여 자신이 보낸 메시지 m_2와 일치함을 확인하고 이후 통신은 세션키 SK_{jd}로 암호화하여 전송한다.

3) 다른 네트워크에서 상호인증

이동성이 높은 IoT 단말노드가 새로운 네트워크 환경에 있는 엣지서버에 통신을 요청(단말노드 EN_j가 다른 네트워크 환경에 있는 새로운 엣지서버인 ES_g와 상호인증 요청)하는 과정은 그림 4와 같다.

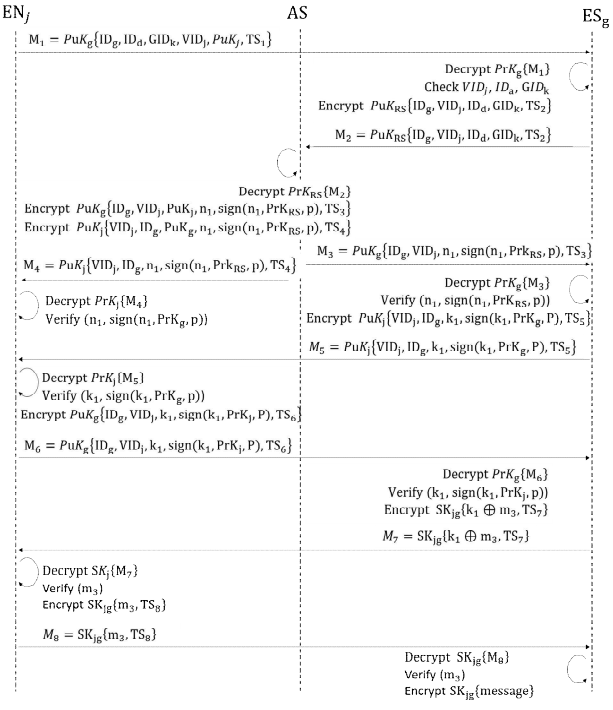

그림 4. 인트라 클라우드에서의 상호인증
Fig. 4. Mutual authentication in inter-cloud

단말노드 EN_j가 다른 네트워크 환경에 있는 엣지서버 ES_g에 M_1과 같은 메시지를 전송하여 통신을 요청하면 엣지

서버 ES_g는 자신의 블록체인 시스템의 트랜잭션 기록을 확인한다. 단말노드 EN_j의 가상 아이디 VID_j가 트랜잭션 기록에 없고, 엣지 그룹 아이디 GID_k가 자신의 그룹 아이디가 아니라면 엣지 서버 ES_g는 인증서버 RS 에 단말노드 VID_j 인증을 요청한다.

인증 서버 RS는 엣지 서버 ES_g의 메시지 M_2를 복호화하여 단말노드 EN_j의 가상 아이디 VID_j가 유효함을 확인한 후, 엣지서버 ES_g와 단말노드 EN_j에 인증 서버 자신의 전자 서명 $sign(n_1, PrK_{RS}, P)$과 무작위 수 n_1가 포함된 메시지를 엣지 서버 ES_g와 단말노드 EN_j의 공개키로 각각 암호화하여 전송한다.

인증 서버 RS로부터 받은 암호문을 엣지서버 ES_g와 단말노드 EN_j는 자신의 개인키(PrK_g, PrK_j)로 복호화하고, 인증 서버 RS의 공개키(PuK_{RS})를 이용하여 전자 서명의 유효성을 확인한다.

엣지서버 ES_g는 인증서버 RS로부터 단말노드 EN_j 아이디 VID_j의 유효성을 확인하고, 자신의 전자 서명 $sign(k_1, PrK_g, P)$을 첨부한 메시지를 단말노드 EN_j의 공개키 PuK_j로 암호화하여 단말노드 EN_j에 전송한다.

단말노드 EN_j는 엣지 서버 ES_g로부터 받은 암호문을 자신의 비밀키로 복호화하여 엣지서버 ES_g의 전자 서명의 유효성을 확인한 후, 자신의 전자 서명 $sign(k_1, PrK_j, P)$이 포함된 메시지를 엣지 서버 ES_g의 공개키로 암호화하여 전송한다.

엣지 서버 ES_g는 단말노드 EN_j가 전송한 암호문을 복호화하고, 단말노드 EN_j의 전자 서명 $sign(k_1, PrK_j, P)$의 유효성을 확인하고, 무작위 수 k_1과 임의의 정수 m_3의 XOR 연산 결과를 세션키 SK_{jg}로 $SK_{jg}\{k_1 \oplus m_3\}$와 같이 암호화하여 단말노드 EN_j에 전송한다.

단말노드 EN_j는 엣지 서버 ES_g로부터 받은 암호문을 세션키 SK_{jg}로 복호화하고 자신이 알고 있는 무작위 수 k_1을 이용하여 $(k_1 \oplus m_3 \oplus k_1)$와 같이 XOR 연산으로 m_3를 확인한다. 확인된 m_3을 세션키 SK_{jg}로 암호화하여 엣지 서버 ES_g로 전송한다.

상호인증 완료와 세션키 생성이 정상적으로 진행된 엣지 서버 ES_g와 단말노드 EN_j 는 생성된 세션키 SK_{jg}로 메시지를 암·복호화하여 통신한다.

4) 세션키 교환 방법

단말노드(EN)와 엣지 서버(ES) 간 상호인증이 완료되면 송·수신 메시지 암호를 위한 세션키를 교환한다.

세션키 교환 알고리즘으로 타원곡선을 변형한 타원곡선 디피-헬만(ECDH:Elliptic Curve Diffie Hellman) 키 교환 알고리즘을 이용한다. 단말노드(EN)와 엣지 서버(ES) 간 대칭키로 사용될 세션키(SK) 교환 과정은 그림 5와 같다.

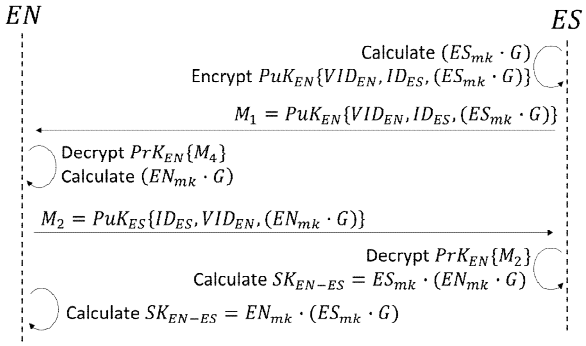

그림 5. EN과 ES간 세션키 교환
Fig. 5. Session Key Exchange between EN and ES

단말노드(EN)와 엣지 서버(ES)는 등록서버(RS)에 등록을 하면서 공유되는 타원 곡선상의 한 점 G값을 이용하여 식 (6)과 같이 키 합의 과정을 거쳐 세션키(SK_{EN-ES})를 생성한다.

$$\begin{aligned} &Calculate \quad S_{ES} = (ES_{mk} \cdot G) \\ &Calculate \quad S_{EN} = (EN_{mk} \cdot G) \end{aligned} \tag{6}$$

$$Exchange \quad S_{ES}, S_{EN}$$

$$\begin{aligned} &Calculate \quad SK_{EN-ES} = (S_{ES} \cdot EN_{mk}) \\ &Calculate \quad SK_{EN-ES} = (S_{EN} \cdot ES_{mk}) \end{aligned}$$

$$(ES_{mk} \cdot G) \cdot EN_{mk} = (EN_{mk} \cdot G) \cdot ES_{mk}$$

엣지 서버(EN)는 곡선상 한점 G과 임의의 수 ES_{mk}를 곱하여 S_{ES}를 만들어 단말노드(EN)에 전송한다. 단말노드도 엣지 서버가 만든 방식과 동일 방식으로 S_{EN}을 만들어 엣지 서버에 전송한다. 단말노드는 엣지 서버로부터 받은 S_{ES}과 자신이 임의로 선택한 EN_{mk}를 곱하여 세션키SK_{EN-ES}를 생성한다. 엣지 서버도 단말노드로부터 받은 S_{EN}과 자신이 임의로 선택한 ES_{mk}를 곱하여 세션키SK_{EN-ES}를 생성한다.

엣지 서버와 단말노드가 최종적으로 생성한 세션키는 곱셈 법칙의 교환법칙이 성립하여 동일 세션키 교환이 성공적으로 이루어진다.

Ⅳ. 연구결과

이 장에서는 본 논문에서 제안한 기법의 안전성과 효율성을 기존 제안된 여러 인증 기법들과 비교 분석한다.

인증 단계에서 발생할 수 있는 보안 위협들에 대한 분석은 표 3과 같다.

표 3. 기존 연구와 보안 비교평가
Table 3. Security comparison of different schemes

Scheme	♈	%	ꝏ	♌	ℑ	𝔇
Yao et al.	⊙	⊙	×	⊙	⊙	×
Hammi et al.	⊙	⊙	⊙	×	⊙	×
Li et al.	⊙	⊙	⊙	×	⊙	×
Proposed Scheme	⊙	⊙	⊙	⊙	⊙	⊙

⊙: Satisfaction ×:Not Satisfaction

엣지 클라우드 환경에서 IoT기기와 엣지 서버의 등록, 두 기기의 상호 인증과정에서 발생할 수 있는 여러 위협들에 대한 기존 연구와 제안한 기법을 비교분석한다. 표 3.에서 중간자공격(♈ : MITM Attack : Man in the Middle Attack), 위장 공격(% : Impersonation Attack), 재전송 공격(ꝏ : Replay Attack), 익명성 보장(♌ : Anonymous), 부인봉쇄(ℑ : Non-reputation) 그리고 상호인증(𝔇 : Mutual Authentication)을 비교 분석하였다. Yao et al.(2019)에서 제안한 기법은 이동수단에 적용되는 인증기법을 제안했다. 이동수단 간 상호인증이 보장되지 않았으며, 인증과정에서 제3자로부터 재전송 공격에 취약하다. Hammi et al.(2018)에서 제안한 인증 기법은 인증센터(CA)를 통한 중앙집중식 인증방법으로 기기 간 상호인증이 이루어지지 않으며, 인증센터(CA)로 아이디를 직접 등록하기 때문에 익명성이 보장되지 않는다. Li et al.(2019)에서 제안한 인증 기법은 key escrow 문제를 해결하기 위한 IBE의 변형된 무인증암호화 방법을 제안했으나 여전히 익명성 보장과 상호인증은 보장되지 않는다. 그러나 본 논문에서 제안한 상호인증 기법은 가상아이디(VID)를 사용하여 익명성을 보장하고, 메시지 전송할 때 타임스탬프(TS)와 함께 전송하면서 재전송 공격으로부터 안전하다. 전자 서명은 송신자의 비밀키로 서명되기 때문에 비밀키를 알지 못하면 전자 서명을 생성할 수 없다. 전자 서명을 했다고 해도 비밀키와 쌍이 되는 공개키로 서명 증명하기 때문에 잘못된 비밀키로 서명된 전자 서명은 서명 증명 과정에서 올바른 서명이 아님을 알 수 있다. 제안한 기법의 상호 인증과정에서 송신자의 상대편에서 보내온 무작위 수(N 또는 R)를 IoT 기기 또는 엣지 서버의 비밀키(PrK)와 함께 전자 서명 $sign(R_1\, or\, N_1, PrK, P)$ 하여 상대 노드에 전송하기 때문에 중간자 공격, 위장 공격으로부터 안전하다. 또한, 전자 서명으로 부인봉쇄를 보장할 수 있다.

인증 단계에서의 성능평가는 표 4와 같으며, 제안한 기법과 Ibrahim(2016)이 제안한 기법의 초기단계, 등록단계 그리고 인증 단계에서 수행되는 암·복호화시 사용되는 암호화 종류와 회수를 비교한다. Ibrahim(2016)이 제안한 기법에서는 등록 서버는 새로운 서버가 등록할 때마다 해시(hash*)를 이용하여 동일 네트워크의 노드들과 사용할 비밀키를 계산한다.

표 5. 알고리즘별 보안 비교
Table 5. Security comparison for various algorithm

Key Size(No. of Bits)(L)	RSA	ECC	ECDH
8	26	26	20
16	76	76	32
32	287	287	64
64	819	819	128
128	2383	2383	256

결국, 새로운 서버의 수가 증가할수록 등록 서버의 부하 증가하여 중앙 집중적 인증방법에서 나타나는 서버 부하, 전송 지연 발생 가능성이 높다. 또한, 초기화 단계와 등록단계에서 등록 서버는 키 생성과 암호화 그리고 전자 서명도 계산한다. 만약, 새로운 서버의 수가 증가 되거나 연결이 해제되었다가 다시 연결되는 횟수가 증가한다면 등록 서버의 부하는 증가된다. 그러나 본 논문에서 제안한 분산 기반의 인증 기법에서는 IoT 기기의 수가 증가하거나 엣지서버가 증가하더라도 초기화, 등록 그리고 인증 단계에서 인증 서버의 계산 횟수를 최소화하여 인증 서버에 집중되는 과부하를 줄였다.

또한, 제안 기법에서는 인증 서버의 시간 복잡도를 RSA, ECC, ECDH 등의 알고리즘을 이용하여 평가한 결과가 표 5와 같다. 표 5에서 사용된 알고리즘별 복잡도를 분석한 결과, RSA 알고리즘과 ECC 알고리즘의의 복잡도는 O(n)이고, ECDH 알고리즘의 복잡도는 O($\sqrt{2^n}$)이다. n은 [24]에서 사용한 식 (7)을 이용한다.

$$n = \frac{1.923 * \sqrt[3]{L * \mathrm{In}(2)} * \sqrt[3]{\mathrm{In}(L * \mathrm{In}(2))^2} - 4.69}{\mathrm{In}(2)} \qquad (7)$$

표 4. 기존 연구와 성능 비교평가
Table 4. Computation comparison of different scheme

	Initialization phase		Registration Phase		Authentication Phase	
	Ibrahim	proposed	Ibrahim	proposed	Ibrahim	proposed
Node	-	1 asym gen	-	1 asym enc	1 hash 1 sym enc 1 sym dec	1 sign verify 1 asym enc 1 asym dec
Server	1 sign verify 1 asym gen	1 asym gen	1 asym dec 1 sign verify 1 asym enc	1 asym enc	1 sym enc 1 sym dec	1 sign verify 1 asym enc 1 asym dec
Registration Server	1 asym gen 1 sign 1 hash	1 asym gen	1 asym enc 1 hash* 1 sign	1 asym dec	-	-

표 5의 결과는 동일한 보안 강도를 위해 필요한 키의 크기를 나타낸다. 보안의 강도는 키의 크기에 따라 달라지기 때문에 알고리즘별 키 길이에 따른 RSA, ECC, ECDH를 분석한 결과, RSA와 ECC는 ECDH보다 동일한 강도의 키 생성에 필요한 시간이 크고, ECDH는 RSA와 ECC보다 적은 생성시간이 필요하다.

V. 결 론

다양한 형태의 IoT 기기들의 사용이 폭발적으로 증가하면서 클라우드 컴퓨팅 환경도 중앙집중식의 정적인 클라우드 환경에서 이동성이 높은 IoT 기기들에 적합한 엣지 클라우드 환경으로 진화하고 있다. 전통적 형태의 클라우드 컴퓨팅에서 사용되던 인증 기법들은 엣지 클라우드 컴퓨팅 환경에 적합하지 않은 몇 가지 단점이 있다. 첫째, 전통적인 클라우드 환경에 적용했던 복잡하고 어려운 인증 알고리즘은 경량화된 IoT 기기들에 적합하지 않다. 둘째, 기존에 제안된 인증 알고리즘의 대다수가 중앙집중식의 인증센터를 통한 인증확인 방법을 사용한다. 하지만, 높은 이동성을 갖는 IoT의 수가 폭발적으로 증가하는 현재 클라우드 환경에서는 중앙집중식의 인증방법은 사용할 수가 없다. 셋째, 수많은 IoT 기기들은 각기 다른 키 생성 알고리즘을 수행하기 때문에 성능과 보안에 있어서 많은 차이를 갖는다는 것이다.

본 논문에서 제안한 상호인증 기법은 블록체인 기반의 분산된 인증 기법으로 IoT 기기와 엣지서버 간 상호인증을 통해 엣지 컴퓨팅 환경에 적합한 인증 기법을 제안하였다. IoT 기기의 가상 아이디 사용하여 익명성 보장, 무작위 수와 타임스탬프를 이용하여 재전송, 중간자 공격 그리고 전자서명을 통해 부인봉쇄, 위장 공격으로부터 안전한 인증기법임을 확인할 수 있었다. 또한, 전통적인 인증과정에서 인증 서버(등록서버:RS)로 집중되는 계산을 본 논문의 제안 기법에서는 기기 간 상호인증을 통해 인증이 이루어지기 때문에 인증서버(등록서버:RS)의 계산 과정이 최소화되었다.

제안 기법을 인증 서버에 일반적으로 사용된 RSA, ECC, ECDH 알고리즘의 복잡도를 분석한 결과, RSA는 n이고, ECC는 $\sqrt{2^n}$ 으로 평가되었다. 인증 서버의 키 길이가 커짐에 따라 증가하는 시간의 비율을 최소화한 알고리즘은 ECDH가 적합한 것으로 확인되었다.

본 논문 연구의 한계점은 다음과 같다. 첫째, 클라우드 엣지 환경을 실제 환경으로 구현하지 않고 이론적 바탕으로 시뮬레이션하였다. 둘째, 제안 기법은 인증 서버의 시간 복잡도를 RSA, ECC, ECDH 알고리즘별로 이론적 개념을 바탕으로 평가하였으나, 실제 운영하는 클라우드 엣지 환경과 성능비교를 수행하지 못했다. 향후 이러한 한계점을 극복하기 위해서 제안 기법을 클라우드 엣지 환경을 구축하여 성능평가를 수행할 계획이다.

참고문헌

[1] A. D. Auday, S. Yogesh, G. K. Michel, and T. Javid, “Introduction to edge computing,” *Edge Computing: Models, Technologies and Applications,* September 2020. https://doi.org/10.1049/PBPC033E_ch1

[2] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, “A survey on the edge computing for the internet of things,” in *IEEE Access*, vol. 6, pp. 6900-6919, November 2018. https://doi.org/10.1109/ACCESS.2017.2778504

[3] Y. Xiao, Y. Jia, C. Liu, X. Cheng, and J. Yu, “Edge computing security: State of the art and challenges,” in *Proceedings of the IEEE*, Vol. 107, No. 8, pp. 1608-1631, August 2019. https://doi.org/10.1109/JPROC.2019.2918437

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp. 1125-1142, October 2017. https://doi.org/10.1109/JIOT.2017.2683200

[5] B. Zijian, W. Shi, D. He, and K. K. R. Choo, “IoTChain: A three-tier blockchain-based IoT security architecture,” *Computer Science ArXiv,*June 2018. arXiv:1806.02008

[6] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, “A blockchain-based mutual authentication scheme for collaborative edge computing,” in *IEEE Transactions on Computational Social Systems*, Vol. 9, No. 1, pp. 146-158, February 2022. https://doi.org/10.1109/TCSS.2021.3056540

[7] A. B. Amor, M. Abid, and A. Meddeb, “A privacy-preserving authentication scheme in an edge-fog environment,” in *Proceedings of 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA),* Hammamet, Tunisia, pp. 1225-1231, 2017. https://doi.org/10.1109/AICCSA.2017.57

[8] W. Choi, S. Kim, and K. Han, “Blockchain-based lightweight mutual authentication protocol for IoT systems,” *Journal of the Korea Society of Computer and Information*, Vol. 24, No. 1, pp. 87-92, November 2019. https://doi.org/10.9708/jksci.2020.25.01.087

[9] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, “A blockchain-based mutual authentication scheme for collaborative edge computing,” *IEEE Transactions on Computational Social Systems,* Vol. 9, No. 1, pp. 146-158, February 2022. https://doi.org/10.1109/TCSS.2021.3056540.

[10] M. H. Ibrahim, “Octopus: An edge-fog mutual authentication scheme,” *International Journal of Network Security,* Vol. 18, No. 6, pp. 1089-1101, November 2016.

[11] D. Q. Viet, A. Yamamura, and H. Tanaka, “Anonymous

password-based authenticated key exchange," in *Proceedings of International Conference on Cryptology*, India, Vol. 3797, pp. 233-257, 2005. https://doi.org/10.1007/11596219_20

[12] F. F. Moghaddam, S. G. Moghaddam, S. Rouzbeh, S. K. Araghi, N. M. Alibeig, and S. D. Varnosfaderani, "A scalable and efficient user authentication scheme for cloudcomputing Environments," in *Proceedings of IEEE Region 10 Symposium*, Malaysia, April 2014, pp. 508-513. https://doi.org/10.1109/TENCONSpring.2014.6863086.

[13] S. Saxena, G. Sanyal, and S. Srivastava, "Mutual authentication protocol using identity based shared secret key in cloud environments," in *Proceedings of IEEE International Conference on Recent Advances and Innovations in Engineering,* Jaipur, September 2014, pp. 1-6. https://doi.org/10.1109/ICRAIE.2014.6909267

[14] P. Reza and H. N. Farshad, "Mutual authentication protocol to share files in cloud storage," in *Proceedings of 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* July 2016. https://doi.org/10.1109/ICCICCT.2016.7987935.

[15] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: Adecentralized blockchain-based authentication system for IoT," *Computers & Security,* Vol. 78, pp. 126-142, June 2018. https://doi.org/10.1016/j.cose.2018.06.004

[16] J. H. Hong, K. C. Lee, and S. Y. Lee. "Trends in edge computing technology," *Electronics and Telecommunications Trends,* Vol. 35, No. 6, December 2020. https://doi.org/10.22648/ETRI.2020.J.350608

[17] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*, New York: Springer, 2006.

[18] R. Haakegaard and J. Lang, "The Elliptic Curve Diffie-Hellman (ECDH)," *Computer Science, Mathematics,* December 2015.

[19] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security,* Vol. 1, pp. 36-63, January 2001. https://doi.org/10.1007/s102070100002

[20] Y. Yao, X. Chang, J. Mi, V. B. Mi, and L. Li, "BLA:Blockchain assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet of Things Journal,* Vol. 6, No. 2, pp. 3775-3784, April 2019. https://doi.org/10.1109/JIOT.2019.2892009

[21] A. B. Amor, A. Mohamed, and A. Meddeb, "A privacy-preserving Authentication Scheme in an Edege-Fog Environment," in *Proceedings of 2017 IEEE/ACS 14th International Conference on Conference on Computer System and Applications,* pp. 1224-1231, October 2017. https://doi.org/10.1109/AICCSA.2017.57

[22] D. Zhang, S. Wang, Q. Zhang, Y. Deng, and J. Wang, "Blockchain-based mutual authentication protocol with privacy protection in telemedicine," *Journal of Physics : Conference Series*, Vol. 2026, July 2021. https://doi.org/10.1088/1742-6596/2026/1/012004

[23] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale Internet of Things data storage and protection," *IEEE Trans. Services Computer,* Vol. 12, No. 5, pp. 762-771, September 2019. https://doi.org/10.1109/TSC.2018.2853167

[24] T. K. Goyal and V. Sahula, "Lightweight security algorithm for low power IoT devices," *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* Jaipur, India, pp. 1725-1729, September 2016. https://doi.org/10.1109/ICACCI.2016.7732296

최정희(Jeong-Hee Choi)

2002년 : 충북대학교 대학원 (이학석사)
2019년 : 충북대학교 대학원 (공학박사)

2020년~현 재: 목원대학교 스톡스대학 SW교양학부 교수
※관심분야 : 정보보호, 인증, IoT, 클라우드 엣지 컴퓨팅