Journal of Convergence for Information Technology
Vol. 12. No. 2, pp. 23-29, 2022
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2022.12.02.023

에지 컴퓨팅 환경을 위한 IoT와 에지 장치 간 키 동의 프로토콜

최정희
목원대학교 스톡스대학 SW교양학부 교수

Key-Agreement Protocol between IoT and Edge Devices for Edge Computing Environments

Jeong-Hee Choi
Professor, Division of Software Liberal Arts, Stokes College, Mokwon University

요 약 최근 사물인터넷(Internet of Things, IoT) 기기 사용 증가로 인해 클라우드 컴퓨팅 서버로 전송해 처리하는 데이터양이 급증하고, 그 결과 네트워크 관련 문제점(지연, 서버의 과부하 및 보안 위협)들이 크게 대두되고 있다. 특히, 연산 능력이 클라우드 컴퓨팅보다 낮은 에지 컴퓨팅은 수많은 IoT 기기들을 손쉽게 인증할 수 있는 경량화된 인증 알고리즘이 필요하다. 본 논문에서는 IoT와 에지 장치 간 익명성과 순방향·역방향의 비밀성을 보장하고 중간자 공격과 재전송 공격에 안정적이며, 에지 장치와 IoT 기기 특성에 적합한 경량화 알고리즘의 키 동의 프로토콜을 제안하였고, 제안한 키 동의 프로토콜을 기존 연구와 비교·분석한 결과 IoT 기기와 에지 장치에서 효율적으로 사용 가능한 경량화 프로토콜임을 보였다.

주제어 : 클라우드 컴퓨팅, 에지 컴퓨팅, 사물인터넷, 인증, 클라우드 보안, 경량화

Abstract Recently, due to the increase in the use of Internet of Things (IoT) devices, the amount of data transmitted and processed to cloud computing servers has increased rapidly. As a result, network problems (delay, server overload and security threats) are emerging. In particular, edge computing with lower computational capabilities than cloud computing requires a lightweight authentication algorithm that can easily authenticate numerous IoT devices.In this paper, we proposed a key-agreement protocol of a lightweight algorithm that guarantees anonymity and forward and backward secrecy between IoT and edge devices. and the proposed algorithm is stable in MITM and replay attacks for edge device and IoT. As a result of comparing and analyzing the proposed key-agreement protocol with previous studies, it was shown that a lightweight protocol that can be efficiently used in IoT and edge devices.

Key Words : Cloud Computing, Edge Computing, Internet of Things, Authentication, Cloud Security, Lightweight

1. 서론

IoT 시대 수많은 기기가 생산하는 데이터를 중앙 집중 방식의 클라우드 컴퓨팅 기반에서 처리하기에는 네트워크 트래픽 증가로 인한 지연, 클라우드 서버에 집중된 처리로 서버의 부하 증가, 다양한 공격 경로로 인한 심각한 보안 위협등 해결해야 할 많은 문제가 존재한다[1,2]. 기존의 중앙집중식 클라우드 컴퓨팅 환경이 IoT 시대에서는 더 이상 이상적 환경이 될 수 없다. 최근에는 네트워크 지연을 최소화하고 데이터 처리 속도를 최대화하기 위한 에지 컴퓨팅(Edge Computing) 환경에서의 IoT의 활용이 활발히 연구되고 있다. 에지 컴퓨팅을 사용하면서 클라우드 컴퓨팅의 단점을 보완했지만, 여전히 에지 컴퓨팅 기반에서도 일부 데이터만 처리·분

*Corresponding Author : Jeong-Hee Choi(heebest@daum.net)
Received December 15, 2021
Revised January 20, 2022
Accepted February 20, 2022
Published February 28, 2022

석하면서 원시정보와 불완전한 정보들을 버림으로 인해 정보손실이 존재하며[3], IoT 환경에서 이기종의 다양한 기기들의 네트워크 연결장치 및 내장 컴퓨터가 추가됨에 따라 악의적 공격자에 대한 공격 경로가 증가하여 MITM 공격, Replay 공격 그리고 Impersonating등 보안 위협이 존재한다[1,2]. 또한, 악의적인 공격자의 비정상적 인증으로 기기 간 통신이 이루어진다면, 에지 컴퓨팅 더 나아가 전체 클라우드 컴퓨팅 시스템이 큰 위험에 노출된다. 기존 클라우드 컴퓨팅 환경과는 다르게 IoT 환경의 특성은 저전력의 효율성을 위한 경량화 알고리즘이 절대적으로 필요하다. 결국 기존의 많은 연산이 필요한 암호 알고리즘은 IoT 환경에서는 사용할 수가 없고, 새로운 경량화 알고리즘이 필요하다.

본 논문에서는 정상적으로 접근하는 IoT 기기와 에지 장치 간 안전하고 효율적 인증을 위한 키 동의 프로토콜을 제안한다. 경량화와 보안(MITM과 재전송 공격, 익명화와 양방향 비밀성보장)에 안정적인 인증 방법을 위해서 제안 프로토콜에서는 IoT와 에지 장치는 에지 서버로부터 부여받은 가상 ID와 무작위 수를 해시 함수와 XOR으로 인증 요청하고, 에지 서버로부터 short term key를 부여받아 세션키로 사용한다. 본 논문의 구성은 다음과 같다. 2장에서는 에지 컴퓨팅 특징과 기존 연구에 대하여 알아본다. 3장에서는 에지 컴퓨팅 환경에서 IoT 기기와 에지 장치 간 안전하고 효율적인 키 동의 프로토콜을 제안하고, 4장에서는 제안 프로토콜의 보안 평가 및 효율성을 기존 연구와 비교 평가하고 마지막으로 결론을 맺는다.

2. 관련 연구

2.1 에지 컴퓨팅

에지 컴퓨팅(Edge Computing)은 클라우드 컴퓨팅 환경의 가장자리에서 클라우드 대신 데이터를 다운스트림하고 IoT 대신 데이터를 업스트림 연산을 수행하는 기술을 의미한다[4]. IoT 기기는 인터넷상에서 다른 장치 및 기기들과 직접 연결하고 통신할 수 있다[1,4]. IoT 활용을 위한 에지 컴퓨팅 특징들을 정리하면 Table 1과 같다. 에지 컴퓨팅의 특징을 살펴보면, 네트워크 지연 문제 감소와 분산처리를 통한 효율성은 있지만 다양한 장치의 연결로 인한 보안의 문제가 발생한다. 신뢰가능한 건강한 IoT 생태계가 만들어지지 않는다면 에지 컴퓨팅구조는 더 이상 확장될 수 없다[5]. 결국, 에지 컴퓨팅 기반의 IoT에서는 기존의 보안 문제와 더불어 IoT 기기의 정보보호, 인증, 관리 문제 그리고 데이터 손실 문제 등 앞으로 해결해야 할 문제가 있다[1].

Table 1. Edge Computing Characteristics

Edge Computing	Description	
Structure	Layer 1	- Smart devices, smart vehicles, smart home appliances, and various smart sensors are located in the IoT device layer[4,15]
	Layer 2	- Middleware layer between IoT devices and Cloud Computing as Edge Computing layer[1] - Distributed data processing - Processing high-level operations that IoT devices cannot process[15]
	Layer 3	- Cloud computing layer. - Processing high-level operations that edge computing cannot process - Storage and processing of data collected from edge computing
Advantages	Reduce Delay	- Achieve response time with reduced network delay[15,16]
	Reduce Cost	- Network cost reduction due to data transmission[15] - Reduce storage and processing costs for Cloud Systems[15]
	Scalability	- Edge computing devices that allow storage and analysis functions to be placed closer to end users than configuring a data center[16,17]
	Reliability	- Even when the data center is interrupted, the Edge computing device basically performs an important processing function[17]
	Security	- A small amount of data moving through the network[14,17] - Distributed storage and processing
Disadvantages	Data Loss Problem	- Edge computing analyzes and processes only the necessary data created by processing[11] - The source data and incomplete information are discarded
	Data Hacking Problem	- With the addition of the Internet of Things (IoT), network connection devices, and built-in computers, malicious attacks through multiple devices increase[4,16] - Hackers infiltrate and increase access to important data[16]
	High-Performance Hardware	- Large edge computing requires more local hardware[4] - Various equipment is needed for data processing - Need hardware to process computing processes - Then the cost increases.

2.2 기존 연구

Erroutbi et al.에서는 IoT 기기와 Fog 노드의 안전한 통신을 위한 경량화 인증 방법을 위해 HMAC (Hash-based Message Authentication Code)를 이용한 인증 프로토콜을 제안하였다[6]. IoT기기와 Fog 노드가 비밀키를 공유하여 인증하는 과정에서 무작위수(C_A), 자신의 아이디(ID_A), syn를 상대 노드(Part B)에 전송하여 인증 요청을 시작한다. 이때 인증 요청이 시작될 때, Part A의 아이디가 그대로 노출되어 전달되고 이후에도 아이디(ID_A)가 그대로 사용되고 있다. 결국 Part A는 익명성을 보장받지 못한다. Loffi et al.에서는 통신하는 상대의 공개키를 이용해서 메시지를 암호화하여 전송하는 방법 사용한다[7]. 이때. 통신 상대의 아이디와 같은 개인 정보를 사용하지 않기 때문에 익명성은 보장하겠지만, 비대칭키를 이용하기 때문에 대칭키를 사용하는 방법에 비해 노드에서 더 많은 연산 시간을 요구한다. 결국 어느 하나의 IoT 노드에 다수의 다른 IoT 기기가 인증을 요구하면 효율성이 낮아진다. Aman et al.에서는 복제 불가능한 생체정보(PUF)를 포함하여 인증하는 기법을 제안했다[8]. 지문과 같이 물리적 복제가 불가능한 기능(PUF)을 사용하기 위해서는 하드웨어에 장치 설치가 필요하다. 또한, 제안된 기법에서는 인증을 요청할 때 인증 요청하는 측의 ID를 그대로 사용하기 때문에 익명성이 보장되지 않는다. Shahidinejad et al이 제안한 기법은 서버와 모든 기기는 Trust Center(TC)에 등록하고 고유한 ID ($PUID_i$)를 발급받아 사용한다. 두 기기 사이에 요청과 응답 시간 파라미터 값의 임계치 값으로 인증 유효성 검사를 하는 방법을 제안하였다[9]. 제안된 인증 방법에서는 TC에서 발급받은 고유 ID($PUID_i$)를 사용하기 때문에 익명성은 보장되지만, 기기 간 인증을 위해 사용되는 상호 인증키는 long-term key를 사용하기 때문에 forward secrecy와 backward secrecy는 보장하지 못한다. Ibrahim은 대칭키를 이용해서 경량 인증과 키 동의 프로토콜을 제안하였다[10]. 제안된 프로토콜은 상호인증은 가능하나, 인증과 키 동의를 위한 세션키와 암호화를 long term key를 사용하기 때문에 forward and backward secrecy를 보장하지 못한다.

3. 제안 프로토콜

이 절에서는 IoT와 에지 장치 간 안전하고 효율적인 인증을 위해 경량화 알고리즘을 이용한 키 동의 프로토콜을 제안하고 있다.

3.1 구성요소

제안 프로토콜의 구성요소는 Fig.1과 같이 ES(에지 서버:Edge Server), ED(에지 장치:Edge Device) 그리고 IoT(사물인터넷:Internet of Things) 기기로 구성된다. ES는 에지 서버로 주변에 접속된 기기들의 정보를 저장하고 ED와 IoT 기기 인증에 필요한 가상ID ($EPID_j$, $UPID_i$)와 Secure Key(AK_j^e, AK_i^u)를 사전에 안전한 채널을 통해 사용자 등록과 함께 발급한다.

Fig. 1. Edge Computing

3.2 IoT 기기 등록

IoT 기기 등록은 IoT와 ED 그리고 ES 간 동작하는 세부적인 등록 과정은 Fig.2, Fig.3과 같다.

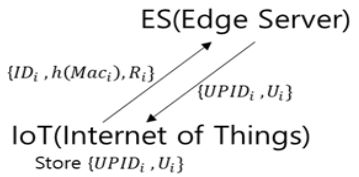

Fig. 2. Registration process IoT

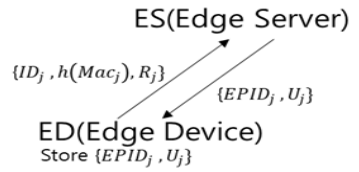

Fig. 3. Registration process ED

IoT와 ED는 ES에 각각 자신의ID(Identify)(ID_i, ID_j), 기기 Mac 주소의 해시값($h(Mac_i)$와 $h(Mac_j)$)

그리고 무작위 수(R_i, R_j)를 전송하여 등록을 요청한다. ES는 IoT와 ED를 등록한 후, 가상 ID ($UPID_i$, $EPID_j$)와 ES가 식(1), 식(2)와 같이 생성한 (U_i , U_j)를 전송한다.

$$AK_i^u = h(ID_i \| SK_u) \quad (1)$$
$$U_i = (AK_i^u \oplus R_i), (AK_i^u \oplus N_{es}^u)$$

$$AK_j^e = h(ID_j \| SK_e) \quad (2)$$
$$U_j = (AK_j^e \oplus R_j), (AK_j^e \oplus N_{es}^e)$$

여기서 SK = {SK_u , SK_e}는 ES가 소유하고 있는 키 그룹이고, N_{es}^u와 N_{es}^e은 ES가 IoT와 ED 인증키를 생성할 때 만든 Nonce값이다.

3.3 IoT와 에지 장치 간 키 동의 프로토콜

IoT 기기와 에지 장치 간 인증을 위한 키 동의 절차는 Fig. 4와 같다. IoT 기기는 자신의 가상 ID($UPID_i$)와 식(3)과 같이 생성된 (C_i , O_i)를 ED에 전송한다.

$$C_i = h(AK_i^u \| UPID_i \| N_{es}^u) \quad (3)$$
$$O_i = (AK_i^u \oplus R_i^{new} \| N_{es}^u)$$

ED는 자신의 가상 ID($EPID_j$)와 식(4)와 같이 생성된 (E_j, O_j)를 IoT에서 전송된 인증 요청 메시지와 함께 ES에 전송한다.

$$E_j = h(AK_j^e \| UPID_j \| N_{es}^e) \quad (4)$$
$$O_j = (AK_j^e \oplus R_j^{new} \| N_{es}^e)$$

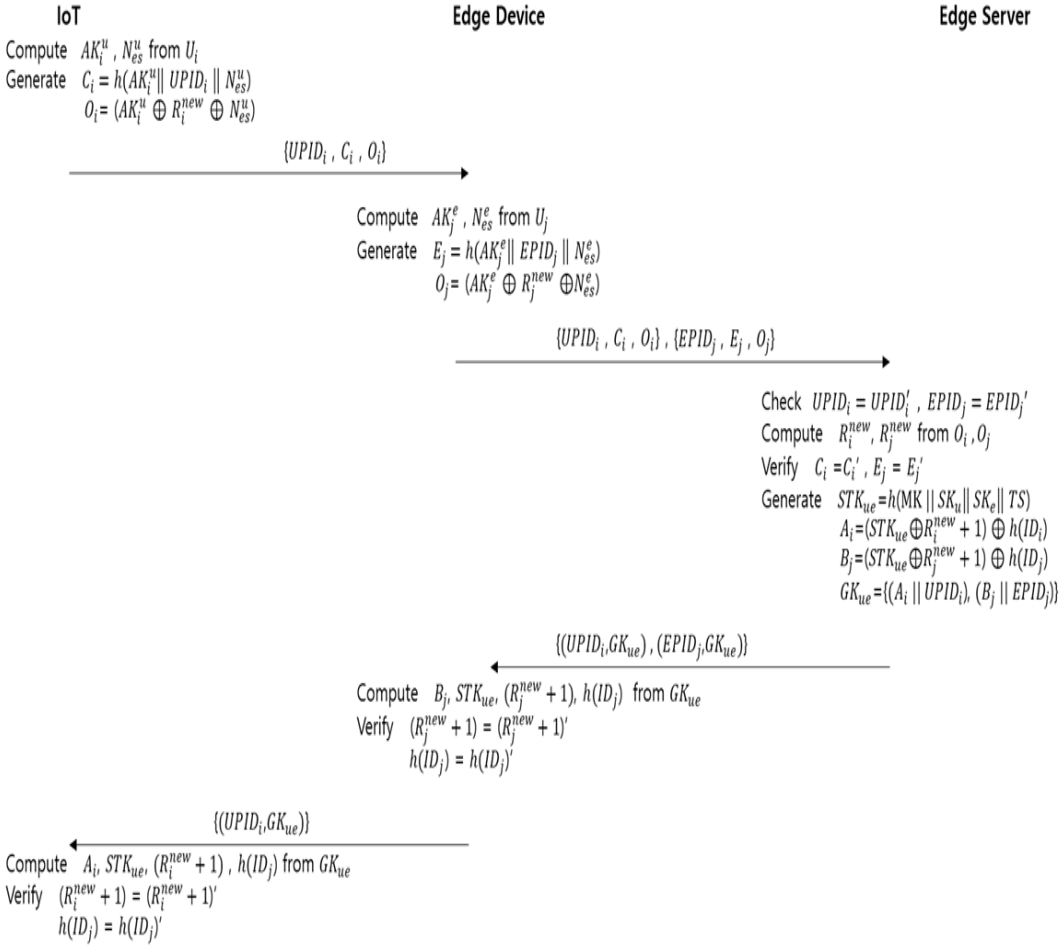

Fig. 4. IoT and ED's key agreement procedure

ED로부터 인증 확인 요청을 받은 ES는 가상 아이디($UPID_i$, $EPID_j$)로 등록된 기기인지 확인한다. 만약, 등록되어 있지 않았다면 다음 절차로 진행하지 않는다. ES는 전달받은 O_i, O_j를 XOR 연산을 통해 R_i^{new}와 R_j^{new}을 추출한다. ES는 식(5)와 같이 IoT와 ED가 정당한 사용자인지 검증한다.

$$C_i = C_i', E_j = E_j' \quad (5)$$

검증이 완료되면, 해당 세션에서만 사용될 TS(Time Stamp)를 생성하여 식(6)과 같이 short term key(STK_{ue})를 생성한다. 여기서 MK는 ES의 master key를 의미한다.

$$STK_{ue} = h(MK \| SK_u \| SK_e \| TS) \quad (6)$$

식(6)에서 생성된 short term key(STK_{ue})를 IoT와 ED에 안전하게 전달하기 위해 식(7)과 같이 A_i, B_j, GK_{ue}를 생성한다.

$$A_i = (STK_{ue} \oplus R_i^{new} + 1) \oplus h(ID_i) \quad (7)$$
$$B_j = (STK_{ue} \oplus R_j^{new} + 1) \oplus h(ID_j)$$
$$GK_{ue} = \{(A_i \| UPID_i), (B_j \| EPID_j)\}$$

ES는 생성된 short term key를 ED에 {($UPID_i$, GK_{ue}), ($EPID_j$, GK_{ue})}와 같이 전송한다. ED는 ES로부터 받은 GK_{ue}을 XOR 연산을 통해 B_j, STK_{ue}, $R_j^{new}+1$, $h(ID_j)$를 추출하여 자신이 요청한 인증 메시지에 대한 응답임을 확인하고, STK_{ue}를 IoT와 ED 간의 short term key로 사용한다. 그리고 ED는 IoT에 ES로부터 받은 {$EPID_j$, GK_{ue}}을 전달한다.

IoT는 ED로부터 전달받은 GK_{ue}를 XOR 연산을 이용하여 자신이 요청한 인증 요청 메시지에 대한 응답임을 확인하고 short term key(STK_{ue})를 추출한다.

IoT와 ED는 세션이 연결되는 동안 short term key(STK_{ue})를 이용하여 통신한다.

4. 평가

이 절에서는 제안 프로토콜을 보안평가와 성능평가로 구분하여 기존 기법들과 비교 평가를 수행하고 있다.

4.1 보안 평가

Table 2에서는 익명성(Anonymity:$\Pi\alpha 1$), 중간자 공격(Man in The Middle Attack:$\Pi\beta 2$), 순방향과 역방향 비밀성(Forward and Backward Secrecy:$\Pi\gamma 3$) 보장, 재전송 공격(Replay Attack:$\Pi\delta 4$) 그리고 사칭 공격(Impersonating Attack:$\Pi\varepsilon 5$)에 대하여 제안 프로토콜과 기존 연구된 프로토콜을 비교하였다.

제안 프로토콜은 IoT와 ED가 ES에 등록하면서 PSEUDO ID($EPID_j$, $UPID_i$)를 발급받아 사용하기 때문에 인증과정에서 사용자 정보가 노출되지 않아 익명성이 보장된다.

악의적인 공격자가 도청 또는 가로채기를 하여 IoT와 ED 간 통신에 사용될 GK_{ue}를 알아냈다고 하더라도 GK_{ue}에서 무작위 수(R_i^{new})와 IoT의 ID(ID_i)를 알 수 없어 STK_{ue}를 추출할 수 없어 중간자 공격은 성공할 수 없다.

IoT 기기와 ED가 ES로부터 받은 short term key인 STK_{ue}는 세션이 연결되는 동안 사용하고, 새로운 연결이 시작되면 새 short term key를 받는다. 따라서 새로운 세션이 연결되었을 때, 현재 세션이 연결되기 이전의 정보나 새로운 세션에서 현재의 정보를 알 수 없어서 forward and backward secrecy가 보장된다.

제안 프로토콜은 무작위 수(R_i^{new})의 증가 값($R_i^{new}+1$)을 통신 단계에서 사용한다. 이 무작위 수 R_i^{new}는 short term key STK_{ue}와 ID의 해시값($h(ID_i)$과 XOR 연산 되어 전달되기 때문에 도청 또는 가로채기가 성공을 해도 알 수 없다. 따라서 재전송 공격은 성공할 수 없다.

악의적인 공격자가 IoT 사칭하여 통신하고자 중간에 C_i와 O_i를 획득하여 서버로부터 GK_{ue}를 얻어 냈다고 하더라도, 공격자는 IoT가 등록할 때 사용했던 ID(ID_i), 인증키(AK_i^u) 그리고 무작위 수(R_i^{new})를 알 수 없어 GK_{ue}로부터 short term key STK_{ue}를 얻어 낼 수 없다. 결국 Impersonating 공격은 성공할 수 없다.

Table 2. Comparison of Security Goal

Scheme	$\Pi\alpha 1$	$\Pi\beta 2$	$\Pi\gamma 3$	$\Pi\delta 4$	$\Pi\varepsilon 5$
proposed protocol	✓	✓	✓	✓	✓
Loffi et al.[7]	×	✓	×	✓	×
Aman et al.[8]	×	✓	✓	✓	×
Shahidinejad et al.[9]	✓	✓	×	✓	✓
Kaur et al.[13]	✓	✓	✓	✓	×

✓: Satisfaction ×: Not Satisfaction

4.2 성능 평가

성능 평가는 인증과정에서 발생하는 통신 오버헤드를 메시지 수 계산으로 Kaur et al.[13]를 기반하여 Table 3과 같이 비교 평가한다. 전송되는 메시지 수 계산은 메시지의 순환 주기(G), 해시(H), 타임스탬프(T:Time Stamp) 그리고 신원확인 ID(ID)와 같이 구성된다. 제안 프로토콜은 2회의 메시지 순환주기(2|G|)를 갖지만 Loffi et al.[6]의 프로토콜은 메시지 순환주기가 4|G|의 순환 주기를 갖기 때문에 제안 프로토콜이 순환 주기 2|G|만큼 효율적이다. 또한 제안 프로토콜은 해시를 2|H| 만큼 사용하고 있지만 Shahidinejad et al.[9]의 프로토콜은 4|H|만큼 사용하고 있다. 따라서 제안 논문의 해시가 2|H| 만큼 적게 사용되고 있다.

Table 3. Comparison of Communication Overheads

Scheme	Cost			
proposed protocol	2|G|	2|H|	1|T|	|ID|
Loffi et al.[7]	4|G|	2|H|	-	|ID|
Aman et al.[8]	2|G|	3|H|	-	|ID|
Shahidinejad et al.[9]	3|G|	4|H|	-	|ID|
Kaur et al.[13]	2|G|	2|H|	2|T|	-

G:Cyclic additive group, H:hash, T:Time-Stamp, ID:identify

5. 결론

본 논문에서는 에지 컴퓨팅(Edge Computing)환경에서 IoT 기기와 에지 장치(Edge Device) 간 안전하고 효율적 통신을 위한 키 동의 프로토콜을 제안하였다. 제안한 프로토콜은 에지 컴퓨팅 환경에서 기기(사용자)의 익명성 보장, 중간자 공격, 순방향과 역방향 비밀성 보장, 재전송 공격 그리고 사칭 공격에 안전함을 확인하였다. 또한, IoT 기기와 에지 장치의 경량화 문제에 대해 일부 효율적인 성능을 보였다.

에지 컴퓨팅은 클라우드의 중앙집중식 방식의 네트워크 지연, 연산의 비효율성, 보안 취약점 등의 단점을 극복하기 위해 나온 클라우드의 또 다른 형태이다. 이러한 에지 컴퓨팅의 활용이 활성화되기 위해서는 확장 가능한 에지 컴퓨팅 연구가 필요하다. 다음 연구에서는 하나의 에지 서버를 둔 단일 에지 컴퓨팅환경이 아닌 다중 에지 서버가 존재하는 환경에서의 기기 간 효율적 인증 방법을 연구하고자 한다.

REFERENCES

[1] V. Hassija. V. Chamola. V. Saxena. D. Jain. P. Goyal & B. Sikdar. (2019). A Survey on IoT Security Application Areas Security Threats and Solution Architectures, *IEEE Access, .7,* 82721-82743.

[2] M. Endler. A. Silva & R. A.M.S. Cruz. (2017). An Approach for Secure Edge Computing in the Internet of Things, *1st Cyber Security in Networking Conference(CSNet)*, 1-8.

[3] S. H. Kim. D. H. Kim. H. S. Oh. H. S. Jeon & H. J. Park. (2016). The Data Collection Solution Based on MQTT for Stable IoT platforms, *Journal of the Korea Institute of Information and Communication Enginering, 20(4),* 728-738.

[4] A. Al-Dulaimy. Y. Sharma. M. G. Khan & J. Taheri. (2020). Introduction to edge computing, *Institution of Engineering and Technology.* 1-24. DOI: 10.1049/PBPC033E_ch1

[5] M. Nakkar. R. AlTawy & A. Youseef. (2021). Lightweight Authentication and Key Agreement Protocol for Edge Computing Applica, *IEEE 7th World Forum on Internet of Things(WF-IoT),* 415-420.
DOI: 10.1109/WF-IoT51360.2021.9595939

[6] A. Erroutbi. A. E. Hanjri & A. Sekkaki. (2019). Secure and Lightweight HMAC Mutual Authentication Protocol for Communication between IoT Devices and Fog Nodes, *5th IEEE International Smart Cites Conference ISC2.* 251-257.

[7] L. Loffi. C. M. Westphall. L. D. Grudtner & C. B. Westphall. (2019). Mutual Authentication for IoT in the Context of Fog Computing. *11th International Conference on Communication System & Networks(COMSNETS).* 367-374.

[8] M. N. Aman. K. C. Chua & B. Sikdar. (2017). Mutual Authentication in IoT System Using Physical Unclonable Functions. *IEEE Internet of Things Journal, 4(5),* 1327-1340.

[9] A. Shahidinejad. M. G. Arani. A. Souri. M. Shojafar & S. Kumari. (2020). Light-Edge: A Lightweight Authentication Protocol for IoT Devices in an Edge-Cloud Environment, *IEEE Consumer Electronics Magazine,* 1-6.

[10] M. H. Ibrahim. (2016). Octopus: An Edge-Fog Mutual Authentication Scheme, International Journal of Network Security, *International Journal of Network Security, 18(6).* 1089-1101.

[11] C. Y. Weng. C. T. Li. C. L. Chen & C. C. Lee. (2021). Lightweight Anonymous Authentication and

Secure Communication Scheme for fog Computing Service, *IEEE Access, .9.* 145522-145537.

[12] J. Y. Choi. (2019). A study on the application of blickchain to the edge computing-based Internet of Things. *Journal of Digital Convergence. 17(12),* 219-228.

[13] K. Kaur. S. Garg. G. Kaddoum. M. Guizani & D. N. K. Jayakody. (2019). A Lightweight and Privacy-Preserving Authentication Protocol for Mobile Edge Computing. *IEEE Global Communications Conference(GLOBECOM),* 1-6. DOI : 10.1109/GLOBECOM38437.2019.9013856

[14] M. A. Rakeei & F. Moazami. (2020), An efficient and provably secure authenticated key agreement scheme for mbile edge computing, *IACR Criptol.,* 1-12

[15] Y. Li, Q.Cheng & X. Liu. (2021). A Secure Anonymous Identity-Based Scheme in New Authentication Architecture for Mobile Edge Computing, *IEEE Systems Journal, 15(1),* 935-946

[16] Y. Xiao. Y. Jia. C. Liu. X. Cheng. & J. Yu. (2019), Edge Computing Security: State of the Art and Challenges, *Preceedings of the IEEE, 107(8),* 1608-1631

[17] J. H. Hong. K. C. Lee & S. Y. Lee. (2020). Trends in Edge Computing Technology, *ETRI Electronics and Telecommunications Tends, 35(6).* 78-87.

최 정 희(Jeong-hee Choi) [정회원]

· 1999년 2월: 서원대학교 상업교육학과 졸업
· 2002년 8월: 충북대학교 컴퓨터과학과 졸업 이학석사
· 2019년 2월: 충북대학교 컴퓨터과학과 졸업 공학박사
· 2020년 3월 ~ 현재 : 목원대학교 스톡스 대학 SW교양학부 교수
· 관심분야 : 정보보호, 인증, 클라우드, 에지 컴퓨팅, IoT
· E-Mail : heebest@daum.net