The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 19, No. 5, pp.1-8, Oct. 31, 2019. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2019.19.5.1
JIIBC 2019-5-1

타원곡선암호 알고리즘을 이용한 종단간 MQTT 보안 프로토콜

End-to-end MQTT security protocol using elliptic curve cryptography algorithm

민정환*, 김영곤**

Jung-Hwan Min*, Young-Gon Kim**

요 약 다양한 인터넷 장치들을 상호 연결하여 보다 지능적인 서비스를 제공하기 위한 사물인터넷(IoT)이 확산되고 있으며, 사물인터넷의 대표적 통신 프로토콜로 TCP 기반의 MQTT가 사용되고 있다. MQTT 기반의 IoT 장치 간 데이터 암호화 기술로 TCP의 경우 TLS/SSL 보안 프로토콜 사용을 권고하고 있지만, 저사양/저용량 IoT 장치에 적용할 경우 암복호화로 인한 성능 저하가 초래된다. 본 논문에서는 MQTT 프로토콜로 연결된 IoT 장치 간에 경량화 암호 알고리즘인 타원곡선암호를 적용한 종단간 메시지 보안 프로토콜을 제안하며, TLS/SSL과 제안 프로토콜에 대한 시뮬레이션을 통해 제안 프로토콜이 클라이언트와 서버 양측에서 성능이 향상됨을 검증하였다.

Abstract Internet of Things (IoT) is proliferating to provide more intelligent services by interconnecting various Internet devices, and TCP based MQTT is being used as a standard communication protocol of the IoT. Although it is recommended to use TLS/SSL security protocol for TCP with MQTT-based IoT devices, encryption and decryption performance degenerates when applied to low-specification / low-capacity IoT devices. In this paper, we propose an end-to-end message security protocol using elliptic curve cryptosystem, a lightweight encryption algorithm, which improves performance on both sides of the client and server, based on the simulation of TLS/SSL and the proposed protocol.

Key Words : IoT, MQTT, broker, publisher, subscriber, TLS/SSL, Elliptic Curve Cryptosystem, ECDH

Ⅰ. 서 론

사물인터넷(IoT)은 사람과 사물 그리고 사물과 사물을 인터넷을 통해 상호 연결하여 초연결 네트워크를 구축하고, 구축된 네트워크 기반으로 사람에게 보다 편리하고 친숙한 지능형 서비스를 제공하는 확장 및 개방형 기술이다[1]. 초연결 네트워크를 근간으로하는 사물인터넷은 스마트 홈, 스마트 카, 스마트 의료, 스마트팜과 스마트 팩토리 분야까지 다양한 분야로 확대되고 있으며[2][3], Gartner에 의하면 2017년 전세계에 84억 개의 사물이 인터넷에 연결되고, 2020년에는 204억 개까지 확장될 것으로 예측되고 있다[4].

*정회원, 한국산업기술대학교 컴퓨터공학과
**정회원, 한국산업기술대학교 컴퓨터공학과
접수일자 2019년 7월 28일, 수정완료 2019년 8월 28일
게재확정일자 2019년 10월 4일

Received: 28 July, 2019 / Revised: 28 August, 2019 / Accepted: 4 October, 2019
**Corresponding Author: ykkim@kpu.ac.kr
Dept. of Computer Engineering, Korea Polytechnic University, Korea

일반적인 인터넷 연결 장치 대비 낮은 사양과 저전력으로 운영되는 사물인터넷 환경은 연결 장치에 내재된 취약점과 장치 간 암호화되지 않은 데이터 전송으로 인하여, 데이터의 유출 및 변조 등 위협에 쉽게 노출될 수 있는 상황이며[5], CCTV를 통한 프라이버시 침해, 다리미와 전기주전자에 대한 해킹 사고 등 사물인터넷 장치와 관련된 개인정보침해 사고가 다수 보고되고 있다[6]. 이러한 사물인터넷 환경 취약성에 대한 대응 방안으로, 하드웨어 중심의 데이터 암호화, 출시 전 보안 진단 그리고 국가적 차원의 정책 수립 등이 제시되고 있다[7][8].

사물인터넷 환경의 데이터 암호화 방안으로, 저사양/저용량의 장치 특성을 고려하여 UDP 기반의 CoAP (Constrained Application Protocol), TCP 기반의 MQTT(Message Queuing Telemetry Transport) 등의 경량화 프로토콜이 사용되고 있으며, 데이터 암호 기술로는 CoAP의 경우 DTLS(Datagram Transport Layer Security), MQTT의 경우 TLS(Transport Layer Security)/SSL(Security Socket Layer) 적용을 권고하고 있다[9]. TCP 기반 데이터 암호화 방안인 TLS/SSL는 일반인 인터넷 연결 장치의 데이터 암호화에 적합하지만[10], 사물인터넷 장치에 적용할 경우 암호 데이터 처리로 인한 성능 저하가 필연적이므로, 이에 대한 개선 방안으로 경량 암호 알고리즘을 이용한 메시지 암호화, 메시지 전체를 암호화하는 TLS/SSL과 달리 메시지 중 페이로드만을 암호화하는 종단간 암호화 방식 등이 제시되고 있다[11][12][13][14].

따라서 본 논문에서는 사물인터넷 장치의 성능적 제한을 고려하여, 타원곡선 암호 알고리즘을 이용한 인증서를 생성하고, 메시지 통신에 참여하는 클라이언트 간에 ECDH 키 교환 절차를 통해 생성된 비밀키 기반으로 MQTT 메시지 중 페이로드에 한하여 암복호화하므로써 전송 데이터의 암호 처리 성능이 향상된 종단간 메시지 보안 프로토콜을 제안한다.

제안하는 타원곡선암호 알고리즘을 이용한 종단간 보안 프로토콜과 TLS/SSL 프로토콜 간의 성능 측정 및 비교를 위하여, 평문 전송의 프로토콜, 암호문 전송의 TLS/SSL 프로토콜 그리고 제안한 보안 프로토콜을 포함하는 3종의 절차 모델을 제안하고, 제안된 모델의 성능 검증을 위한 시뮬레이션 프로그램을 구현한 후, 모델별 성능 측정 결과를 비교하여 제안한 보안 프로토콜의 사용 타당성을 제시한다.

본 논문은 제1장의 서론을 포함하여, 총 5개의 장으로 구성되었다. 제2장에서는 사물인터넷 프로토콜인 MQTT, 경량의 타원곡선 암호 기술, MQTT 지원 소프트웨어에 대해 소개하고, 제3장에서는 종단간 보안 프로토콜과 시뮬레이션 모델을 각각 제안하고, 제4장에서 시뮬레이션 환경과 측정 결과에 대하여 정리하고 , 제5장은 본 논문의 결론으로 논문의 전반적인 내용 정리와 향후 연구 방향에 대하여 기술한다.

II. 관련연구

1. 사물인터넷 프로토콜 MQTT

MQTT는 OASIS(Open standards. Open source.)에 의해 2013년에 사물인터넷 표준 메시지 전송 프로토콜로 지정되었고 2016년도에 ISO 표준(ISO/IEC PRF 20922)으로 채택된 TCP 기반의 경량 메시지 전송 프로토콜이다[15].

MQTT는 publisher/subscriber 기능을 수행하는 클라이언트와 publisher와 subscriber 사이에서 메시지를 중개해주는 브로커/서버로 구성되며, publisher와 subscriber 그리고 브로커 간의 MQTT 통신 절차는 그림 1과 같다.

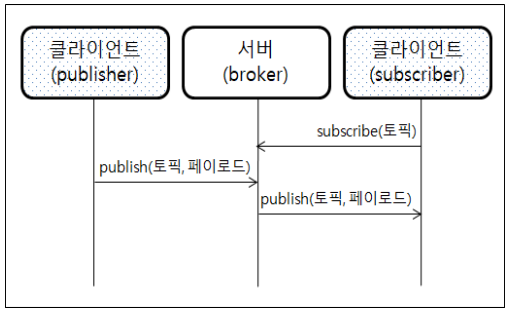

그림 1. MQTT 통신 절차
Fig. 1. MQTT communication procedure

MQTT 메시지는 publisher/subscriber 간에 사전 공유된 토픽을 이용하여 브로커를 경유하여 publisher/subscriber 간에 송수신되며, publisher는 subscriber에 전송하는 메시지의 신뢰수준을 QoS(Quality of Servce) 레벨로 설정할 수 있으며, 0 레벨은 At most once, 1 레벨은 At least once, 2 레벨은 Exactly once를 의미한다. publisher로부터 publish 메시지를 수신한 브로커는 설정된 레벨을 확인한 후 subscriber에 적절한 절차에 따라 메시지를 전달하며, subscriber 역

시 필요한 QoS 레벨을 브로커에 등록할 수 있다.

MQTT 메시지 중에서 publish 메시지는 메시지의 형식을 지정하는 고정 헤더, 토픽의 이름과 길이를 지정하는 가변 헤더 그리고 페이로드로 구성되고, subscribe 메시지는 형식 지정 고정 헤더, 토픽의 이름과 길이 지정 헤더 그리고 필요한 QoS 레벨을 포함하는 페이로드로 구성된다. MQTT publish와 subscribe 메시지 구조는 그림 2와 같다.

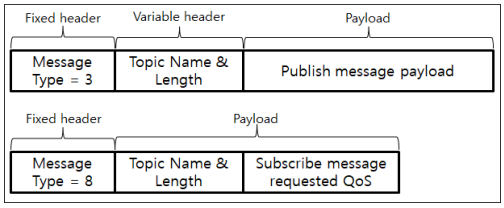

그림 2. MQTT publish와 subscribe 메시지 구조
Fig. 2. MQTT publish and subscribe message format

MQTT에서 장치간 평문 통신의 경우 메시지는 포트번호 1883번을 사용하면서 mqtt 프로토콜을 지원하는 브로커를 경유해서 publisher/subscriber 간에 송수신되고, TLS/SSL을 사용하는 암호 통신의 경우 포트번호 8883번을 사용하면서 secure-mqtt 프로토콜을 지원하는 브로커로 전송된다. 암호 통신의 경우 전송할 평문 메시지는 secure-mqtt 내부에서 암호화 후 송신되며, 암호화된 메시지를 수신한 secure-mqtt 프로토콜은 암호화된 메시지를 복호화한 후 평문 메시지를 전송한다.

메시지 암복호화에 따라 클라이언트는 연결할 브로커의 호스트 주소 값과 연결 포트 번호를 적절하게 선택하여야 한다. 평문 통신과 암호화 통신 모두 가능한 MQTT 네트워크 기반 시스템 구성도는 그림 3과 같다.

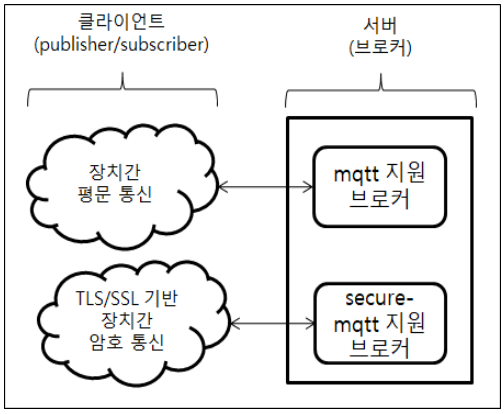

그림 3. MQTT 네트워크 기반 시스템 구성도
Fig. 3. MQTT network-based system configuration

2. 경량의 타원곡선 암호 기술

타원곡선암호 알고리즘은 이산대수에서 사용하는 유한체의 곱셈군을 타원곡선군으로 대치한 암호체계로써, RSA 암호체계에 비하여 1/10 수준의 키 길이로 대등한 보안 수준을 제공한다. 타원곡선 알고리즘을 이용한 암호 시스템에서, 통신에 참여할 클라이언트는 먼저 공개키를 생성하고, 통신에 참여할 클라이언트 간에 공유한 후 ECDH(Elliptic Curve Diffie-Hellman) 알고리즘을 이용하여 각각 비밀키를 생성하고, 생성한 비밀키를 이용하여 메시지를 암호화하여 송신하고, 수신 후 복호화하는 과정으로 진행된다[16]. MQTT 메시지 전체를 암호화하는 TLS/SSL 암호 대신 경량화된 타원곡선암호 알고리즘을 메시지 중 페이로드에 한하여 적용할 경우 클라이언트 및 서버의 처리 속도도 향상시킬 수 있을 뿐 만 아니라 자원 사용량도 경감시킬 수 있다.

3. MQTT 지원 소프트웨어

MQTT 서버와 클라이언트 프로그램은 공개 소프트웨어 형태로 입수 가능하며, 대표적인 브로커/서버 프로그램으로 Eclipse mosquitto, HiveMQ 등이 있고, 클라이언트(publisher/subscriber) 프로그램 개발용으로 자바, 파이썬 등 다수 언어를 지원하는 Eclipse Paho 클라이언트 라이브러리[17]와 HiveMQ Client 라이브러리[18]가 있다.

Ⅲ. 제안하는 종단간 보안 프로토콜

1. 종단간 MQTT 보안 프로토콜 제안

제안하는 종단간 MQTT 보안 프로토콜은, 메시지 통신에 참여하는 클라이언트 간에 적용되어 서버에 의한 암복호화 과정이 없어, 클라이언트/서버 간 암복호화가 필요한 TLS/SSL와 달리 서버에 의한 부하를 경감시킬 수 있다. 또한 통신에 참여하는 클라이언트는, 타원곡선 암호 알고리즘을 이용한 인증서를 생성하고, 통신에 참여하는 클라이언트 간에 공개키를 상호 교환한 후, ECDH 키 교환 절차를 통해 생성된 비밀키 기반으로 MQTT 메시지 중 토픽을 제외한 페이로드에 한하여 암복호화 과정을 진행하므로, 메시지 전체를 암복호화하는 TLS/SSL 대비 부하가 감소한다.

2. MQTT 통신 방식과 통신 절차 제안

표 1. MQTT에서 지원하는 표준, 권고 및 제안 방식 비교
Table 1. Comparison of standard, recommendation, and proposed communication methods supported by MQTT

통신방식	장치 간 평문 통신	TLS/SSL 기반 장치간 암호 통신	ECC 기반 종단간 암호 통신
암복호화 실행 장치	해당없음	클라이언트, 서버	클라이언트
메시지 내 암호화 대상	해당없음	토픽과 페이로드	페이로드
기반 인증서	해당없음	RSA	ECC
프로토콜 (포트 번호)	mqtt (1883)	secure-mqtt (8883)	mqtt (1883)
비고	표준	권고	제안

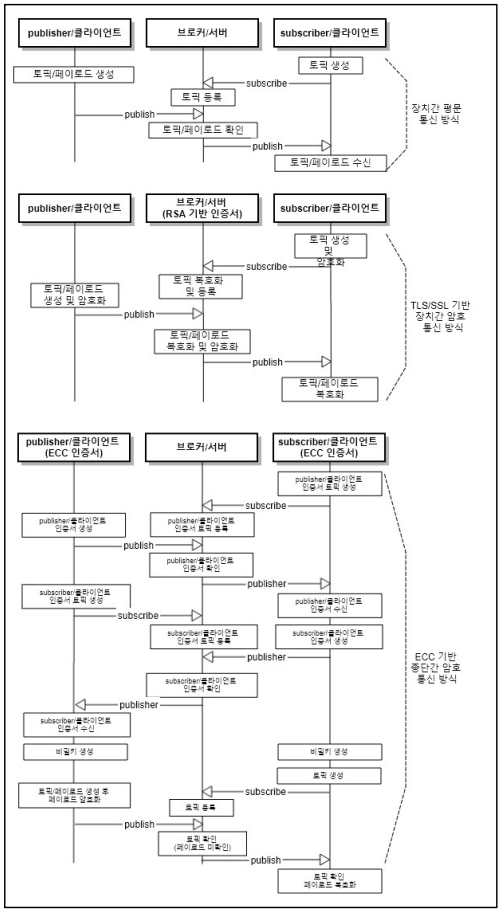

그림 4. MQTT 지원 3종 통신 절차
Fig. 4. 3 Communication procedures supported by MQTT

MQTT에서 제공하는 표준인 장치간 평문 통신 방식, 권고인 TLS/SSL 기반 장치간 암호 통신 방식 그리고 제안하는 암호 통신 방식 등이 있다. MQTT에서 지원하는 표준, 권고 및 제안 방식 비교는 표1과 같고, MQTT 지원 3종 통신 절차는 그림 4와 같다.

3. 처리 성능 측정을 위한 통신 모델 제안

MQTT에서 제공하는 표준, 권고 통신 방식과 제안하는 통신 방식의 시뮬레이션을 위한 절차 모델을 각각 M0, M1, M2라 할 때 MQTT에서 지원하는 3종 통신 절차 모델은 그림 5와 같다.

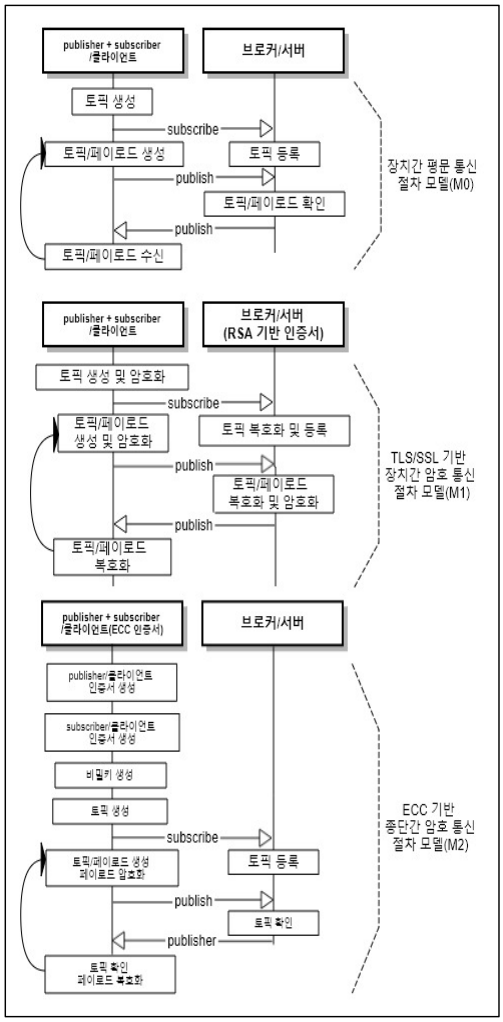

그림 5. MQTT에서 지원하는 3종 통신 절차 모델
Fig. 5. 3 operation models supported by MQTT

M0 모델은 장치 간의 평문통신 방식을 모델링한 것으로, publisher와 subscriber 기능을 모두 지원하는 클라이언트가 mqtt 프로토콜을 이용하여 특정 토픽으로 subscribe 메시지를 발송하여 브로커/서버에 토픽을 등록한 후, 동일한 토픽으로 publish 메시지를 발송하면 브로커는 수신된 publish 메시지를 클라이언트로 재발송하게 되고, 클라이언트는 당초 발송한 publish 메시지를 수신하게 되므로, 당초 발송한 메시지의 내용과 비교와 검증이 가능하고, 필요 횟수 만큼 과정을 반복 수행하면서 메시지 송수신 소요 시간을 측정할 수 있다. 모델 간 정확한 소요 시간 측정을 위해 QoS는 0으로 설정한다.

M1 모델은 TLS/SSL 기반 장치간 암호통신 방식을 모델링한 것으로, M0 모델과 달리 브로커/서버에 설치된 RSA 인증서를 클라이언트에서 공유하고, 공유된 인증서를 활용하여 TLS/SSL 기반의 secure-mqtt 프로토콜을 통해 브로커/서버 인증, 비밀키 생성 및 공유 등의 절차를 거쳐, 비밀키를 이용하여 토픽과 페이로드 등 메시지 전체를 암복호화한다.

끝으로 M2 모델은 ECC 기반 종단간 암호통신 방식을 모델링한 것으로, 클라이언트에서 publisher와 subscriber 인증서를 각각 생성한 후, 각각의 공개키를 ECDH 키 교환을 거쳐 비밀키를 생성하고, 생성된 비밀키를 이용하여 토픽을 제외한 페이로드를 암호화하여 mqtt 프로토콜을 이용하여 브로커/서버에 publish 메시지로 전송하고, 브로커/서버로부터 수신된 publish 수신된 메시지 중 페이로드만을 복호화하여 당초 발송한 메시지의 내용과 동일한 지의 여부를 검증한다.

MQTT 상에 3종의 모델을 활용하여, 통신 방식 간 성능 측정과 비교를 위한 시뮬레이션 프로그램을 구현하였다. 시뮬레이션 프로그램은, Eclipse Paho 자바 클라이언트 라이브러리를 이용하여 publish/subscribe 통신 기능을, SSLSocket 클래스를 이용하여 TLS/SSL 기반 암호통신 기능을, 그리고 타원곡선 암호 클래스를 이용하여 ECC 인증서 기반 암호통신 기능을 모두 지원할 수 있도록 자바 언어로 구현하였다. 또한 시뮬레이션 프로그램은, M1 모델을 지원하기 위해 브로커/서버에 2048 비트 키 길이의 RSA 기반 인증서를 사전 설치하였고, M2 모델을 지원하기 위해 각 클라이언트에 RSA 기반 인증서 키 길이 수준의 SECG SEC2에 속하는 256 비트 길이 타원곡선암호인 secp256r1 기반 인증서를 생성 후 사용하였으며, M2 모델은 M1 모델 TLS/SSL의 암호화 스위트(Cipher Suite)에서 지원하는 프로토콜, 키 교환방식, 인증서 검증, 대칭키를 이용한 블록 암호화 방식, 블록 암호 운영방식, 메시지 인증 기능 중에서 대칭키를 이용한 블록 암호화 방식과 블록 암호 운영방식 기능 만을 지원하도록 구현하였다.

시뮬레이션 프로그램은 publish 메시지 페이로드 길이를 128 바이트로 고정하고, 메시지 송수신 과정를 필요한 횟수만큼 반복하며 모델 별 소요 시간을 측정 후 상호 비교할 수 있도록 구현하였다.

IV. 시뮬레이션 환경과 측정 결과

1. 클라이언트 성능 측정 결과

클라이언트 성능 측정을 위한 제안 시스템 구성은 그림 6과 같고, 사용한 서버와 클라이언트의 사양은 표 2와 같다.

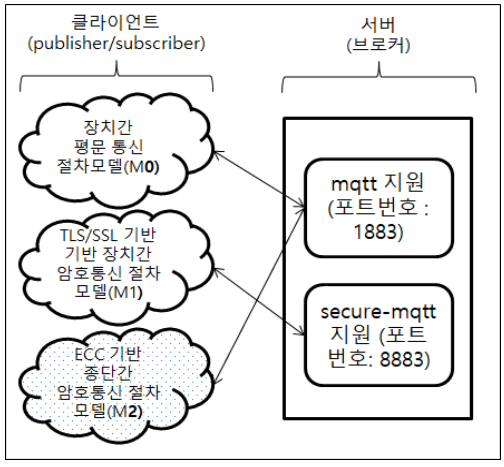

그림 6. 제안 시스템 구성
Fig. 6. Proposed system configuration

표 2. 서버와 클라이언트의 사양
Table 2. Server and client specifications

통신방식	장치 간 평문 통신	TLS/SSL 기반 장치간 암호 통신	ECC 기반 종단간 암호 통신
암복호화 실행 장치	해당없음	클라이언트, 서버	클라이언트
메시지 내 암호화 대상	해당없음	토픽과 페이로드	페이로드
기반 인증서	해당없음	RSA	ECC
프로토콜 (포트 번호)	mqtt (1883)	secure-mqtt (8883)	mqtt (1883)
비고	표준	권고	제안

제안 시스템 중 서버에 브로커인 Eclipse mosquitto를 각각 설치한 후, M0와 M2 모델의 경우 mqtt로, M1의 경우 secure-mqtt로 설정하였다. 다음으로 구현된 시뮬레이션 프로그램을 클라이언트에 설치하고, 3종 모델에 대해, 단일 프로세스로, 128 바이트 길이의 고정된 페이로드를 갖는 publish 메시지를 1개부터 4001개까지 200개 단위로 증가시켜 가면서 우분투에서 제공하는 time 명령어를 이용하여 실행 프로세스의 real/user/sys time을 각각 측정하였다. 3 종 통신 모델의 클라이언트 성능 측정 결과는 그림 7과 같고, 3 종 통신 모델의 클라이언트 성능 측정 결과 비교는 표 3과 같다.

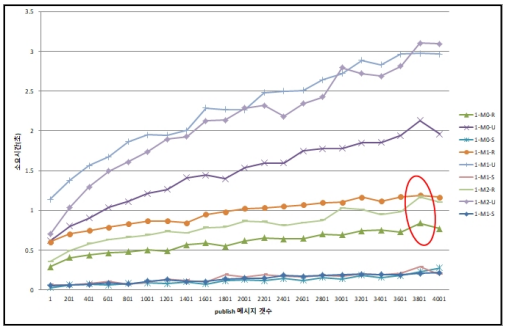

그림 7. 3 종 통신 모델의 클라이언트 성능 측정 결과
Fig. 7. Performance measurement of 3 communication models in client

표 3. 3 종 통신 모델의 클라이언트 성능 측정 결과 비교
Table 3. Performance measurement of 3 communication models in client

(단위 : 초, %)

메시지 개수	1-M0-R	①1-M1-R	②1-M2-R	(②-①)/①
1	0.29	0.60	0.35	-0.42
201	0.41	0.71	0.49	-0.30
401	0.44	0.75	0.58	-0.22
601	0.47	0.79	0.63	-0.20
801	0.48	0.83	0.66	-0.20
1001	0.50	0.87	0.69	-0.20
1201	0.49	0.87	0.74	-0.15
1401	0.57	0.85	0.72	-0.15
1601	0.59	0.95	0.78	-0.18
1801	0.55	0.98	0.79	-0.19
2001	0.62	1.03	0.87	-0.16
2201	0.65	1.04	0.86	-0.17
2401	0.64	1.05	0.81	-0.23
2601	0.65	1.07	0.84	-0.21
2801	0.70	1.10	0.88	-0.20
3001	0.69	1.10	1.03	-0.07
3201	0.74	1.17	1.01	-0.14
3401	0.75	1.12	0.95	-0.15
3601	0.73	1.17	0.99	-0.16
3801	0.84	1.19	1.17	-0.01
4001	0.77	1.17	1.10	-0.05
평균				-0.18

3종 모델에 대한 성능 측정 결과, 단일 프로세스 소요 시간은 publish 메시지 개수가 증가함에 따라 sys 〈 real 〈 user time 순서로 증가하였고, 3종 모델의 실제 소요 시간인 real time은 publish 메시지 개수가 4001개(송수신 메시지는 총 1.02 M바이트, 2회 왕복 * 128 바이트 * 4001개)까지 M0 〈 M2 〈 M1 순서가 유지되었으며, M1 모델 대비 M2는 평균 18% 감소하였다.

2. 서버 성능 측정 결과

서버 성능 측정을 위한 제안 시스템 구성은 클라이언트의 경우와 동일하다. 추가로 클라이언트 측에서 브로커/서버에 부하를 가중시킬 수 있도록, 3종 모델 별로, 메시지 송수신 프로세스를 11개부터 71개까지 20개 단위로 증가시켜 가면서 병렬 실행을 통해, 병렬 프로세스 시작부터 종료까지 소요된 시간을 동일하게 측정하였다. M0 모델을 제외한 M1과 M2 모델에 대한 2 종 통신 모델의 서버 성능 측정 결과는 그림 8과 같다.

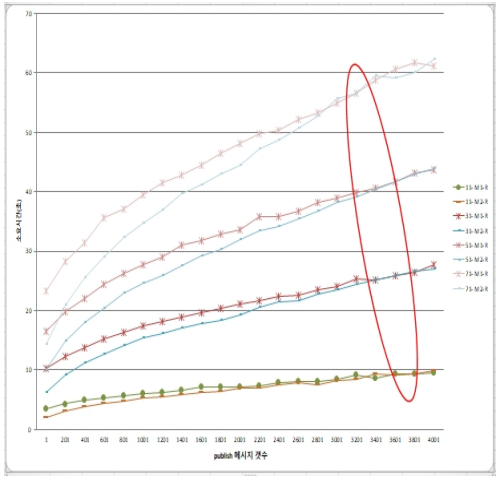

그림 8. 2 종 통신 모델의 서버 성능 측정 결과
Fig. 8. Performance measurement of 2 communication models in server

3종 모델의 서버 성능 측정 결과, publish 메시지 전송 프로세스의 수가 증가함에 따라 병렬 프로세스들의 실제 소요 시간은 real time 기준으로 M2 〈 M1의 순서를 유지하며, publish 메시지 개수가 약 3000 ~ 3500여개 될 때까지 지속적으로 유지되었다. 특히 71개 프로세스 병렬 실행 시 M1 모델 대비 M2는 평균 9% 감소하였다.

71개 병렬 프로세스 실행 시 3 종 통신 모델의 서버 성능 측정 결과와 비교는 표 4와 같다.

표 4. 71개 병렬 프로세스 실행 시 3 종 통신 모델의 서버 성능 측정 결과와 비교

Table 4. Comparison of server performance measurement of 3 communication models with 71 parallel process execution

(단위 : 초, %)

메시지 개수	71-M0-R	①71-M1-R	②71-M2-R	((②-①)/①)
1	12,41	23,28	14,21	-0,39
201	17,14	28,20	20,87	-0,26
401	19,02	31,37	25,42	-0,19
601	21,37	35,53	29,02	-0,18
801	23,30	37,07	32,31	-0,13
1001	25,03	39,47	34,61	-0,12
1201	26,77	41,46	36,85	-0,11
1401	27,94	42,79	39,68	-0,07
1601	29,30	44,50	41,15	-0,08
1801	30,81	46,35	42,98	-0,07
2001	31,60	48,00	44,35	-0,08
2201	32,98	49,79	47,13	-0,05
2401	34,13	50,22	48,69	-0,03
2601	36,82	52,16	50,67	-0,03
2801	36,98	53,17	52,74	-0,01
3001	37,29	54,94	55,58	0,01
3201	38,14	56,45	56,45	0,00
3401	42,38	58,79	59,40	0,01
3601	40,55	60,57	59,07	-0,02
3801	42,53	61,75	59,95	-0,03
4001	43,74	61,12	62,13	0,02
평균				-0,09

V. 결 론

본 논문에서는 사물인터넷 장치 간에 타원곡선암호 기반 경량화 암호 알고리즘을 적용한 종단간 MQTT 보안 프로토콜을 제안하였으며, TLS/SSL과 제안 프로토콜에 대한 시뮬레이션을 통해 제안 프로토콜이 클라이언트와 서버 양측에서 성능이 향상됨을 검증하였다.

제안한 종단간 MQTT 보안 프로토콜을 포함하여 MQTT에서 지원하는 3종 모델에 대한 시뮬레이션 프로그램을 이용한 클라이언트 성능 측정 결과, 모델별 프로세스 소요 시간인 real time은 M1 모델 대비 M2는 평균 18% 감소하였다. 이는 제안한 M2 모델은 1.02 M바이트 이하의 데이터를 송수신할 경우 M1 보다 처리 성능이 향상됨을 의미한다.

또한 3종 모델의 서버 성능 측정 결과, publish 메시지 전송 프로세스의 수가 증가함에 따라 병렬 프로세스들의 실제 소요 시간은 real time 기준으로 M2 〈 M1의 상태를 유지하며 71개 프로세스 병렬 실행 시 M1 대비 M2는 평균 9% 감소하였으며, 이는 제한된 송수신 데이터 범위 내에서 M2 모델 경우 M1 보다 처리 성능이 우수하고 많은 클라이언트 접속이 가능함을 뜻한다.

본 논문에서 제안한 타원곡선암호 알고리즘을 이용한 종단간 MQTT 보안 프로토콜을 사용하므로써, MQTT 표준인 장치간 평문 통신 방식 대비 보안 통신이 가능하고, 권고인 TLS/SSL 기반 암호 통신방식 대비 클라이언트와 서버 공히 성능 향상이 가능하며 또한 저사양의 사물인터넷 장치에도 적용 가능하다.

향후 연구에서는 제안한 프로토콜에 대한 보안 기능 추가와 안전성 점검 그리고 MQTT 네트워크에서 운영되는 서버와 클라이언트의 관리와 모니터링 그리고 사물인터넷 장치별 인증서 배포와 인증 방안에 대한 연구를 진행할 계획이다.

References

[1] https://en.wikipedia.org/wiki/Internet_of_things

[2] Hun Jung, “Study on the MQTT protocol design for the application of the real-time HVAC System”, International Journal of Internet, Broadcasting and Communication, Vol. 8 No. 1 pp. 19-26, 2016.
DOI: http://dx.doi.org/10.7236/IJIBC.2016.8.1.19

[3] Se-Chun Oh, Tae-Hyung Kim, Young-Gon Kim, “Implementation of factory monitoring system using MQTT and Node-RED”, The Journal of The Institute of Internet, Broadcasting and Communication(IIBC), Vol. 18, No. 4, pp. 211-218, Aug 2018.
DOI: https://doi.org/10.7236/JIIBC.2018.18.4.211

[4] https://www.gartner.com/newsroom/id/3598917

[5] Se-Hwan Park, Jong-Kyu Park, “IoT Industry & Security Technology Trends”, International Journal of Advanced Smart Convergence, Vol. 5, No. 3, pp 27-31, 2016.
DOI:http://dx.doi.org/10.7236/IJASC.2016.5.3.27

[6] Namseoul University, "In the 'Internet of Things', a research of actual cases and analysis of factors infringing privacy," Personal Information Protection Commission, Dec. 2015.

[7] Seon-Keun Lee, "A Study on Pseudo-random Number Generator with Fixed Length Tap unrelated to the variable sensing nodes for IoT Environments", Journal of the Korea Academia-Industrial cooperation Society(JKAIS), Vol. 19, No. 3, pp. 1-7, 2018.
https://doi.org/10.5762/KAIS.2018.19.2.676

[8] Sunghyuck Hong, Hyeon-Jun Sin, "Analysis of the vulnerability of the IoT by the Scenario", Journal of

the Korea Convergence Society Vol. 8. No. 9, pp. 1-7, 2017.
DOI:https://doi.org/10.15207/JKCS.2017.8.9.001

[9] Se-Ra Oh, Young-Gab Kim, "Security Analysis of MQTT and CoAP protocols in the IoT Environment", Spring Conference of the Korea Information Processing Society, Vol. 23, pp. 297-299, Apr. 29 ~ 30, 2016.

[10] Shailendra Singh Tanwar, Gamini Sharma, Dixit Soni, "Internet of Things (IoT) Reliability in Transport Encryption System with Cryptographic Solution", Journal of Basic and Applied Engineering Research, Vol. 1, No. 7, pp. 71-75, Oct 2014.

[11] Hee-jeong Kim, Jeong Nyeo Kim, "A Study of End-to-End Message Security Protocol Based on Lightweight Ciphers for Smart IoT Devices", Journal of The Korea Institute of Information Security & Cryptology, Vol. 28, No. 6, Dec 2018.
https://doi.org/10.13089/JKIISC.2018.28.6.1309

[12] Danish Bilal Ansari, Atteeq-Ur-Rehman, Rizwan Ali Mugha, "Internet of Things (IoT) Protocols: A Brief Exploration of MQTT and CoAP", International Journal of Computer Applications, Vol. 179, No .27, Mar 2018.
DOI: https://doi.org/10.5120/ijca2018916438

[13] Choi Bo-mi, KimSung-hyun, "Survey on Security Technology and Research Trend for IoT Environment", Proceedings of Korean Institute of Information Technology(KIIT) Conference, pp. 240-242, Dec 2017.

[14] HiveMQ, "MQTT Security Fundamentals", https://www.hivemq.com/mqtt-security-fundamentals

[15] OASIS, "MQTT Version 3.1.1 OASIS Standard", Oct 2014.

[16] V. Gayoso Martinez, L. Hernandez Encinas, "Implementing ECC with Java Standard Edition 7", International Journal of Computer Science and Artificial Intelligence, Vol. 3, Iss. 4, pp. 134-142, Dec 2013.
DOI:https:/doi.org/10.5963/IJCSAI0304002

[17] ECLIPSE, "Eclipse Mosquitto", https://mosquitto.org/

[18] HiveMQ, "MQTT Client Library Encyclopedia", https://www.hivemq.com/mqtt-client-library-encyclopedia/

저 자 소 개

민 정 환(정회원)

• Jung-Hwan Min received his BS and MS degree in Electronic Engineering at Yonsei University in 1985 and 1987, respectively. He has been a Ph.D. student in Department of Computer Science at Korea Polytechnic University since 2017. He is currently an associate professor of Department of Computer Science at Korea Polytechnic University.

• His fields of research are Security in Systems and Network, Information Security Systems, Internet of Things, etc.

김 영 곤(정회원)

• Young-Gon Kim received his BS degree in Electronic Engineering at Kyungpook National University in 1983 and MS degree in Electronic Engineering at Yonsei University in 1985, respectively. In 2000, he received his Doctor degree in Computer Science at KAIST. He is currently a professor of Department of Computer Science at Korea Polytechnic University.

• His fields of research are Software Engineering, Information and Communication Systems, Object-Oriented Design, etc.