Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.5, Oct. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.5.787

IoT 자동화 시스템의 지연 공격 탐지*

김 영 덕,[1†] 최 원 석,[2] 이 동 훈[2‡]
[1,2]고려대학교 (대학원생, 교수)

Detection of Delay Attack in IoT Automation System*

Youngduk Kim,[1†] Wonsuk Choi,[2] Dong hoon Lee[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

가정에서 IoT 장비가 많이 활용되면서, IoT 장비를 통합하여 사용자의 편의에 맞게 사용하는 IoT automation system을 향한 관심도 많아졌다. IoT automation system에서는 사전 정의된 자동화규칙에 따라 IoT 장비의 정보를 수집하고, IoT 장비가 작동한다. 하지만 공격자는 패킷을 지연시켜 실제 상태와 시스템에서 인식한 상태의 불일치가 되는 시간을 만든다. 이 시간 동안 시스템은 사전 정의한 자동화 규칙대로 동작하지 않는다. 제안된 지연 공격 탐지방법이 일부 있지만, 트래픽량이나 배터리에 민감한 IoT system에 적용하기에 제한사항이 있다. 본 논문에서는 IoT 시스템에 적용할 수 있는 실용적인 패킷 지연 공격 탐지 방법을 제시한다. 제안 방법은 메세지를 전송할 때, 메세지의 전송을 알리는 패킷을 브로드캐스트로 발송함으로써 서버가 이벤트의 발생을 인지할 수 있도록 한다. 평가를 위하여 Raberry pi로 구현된 IoT system을 구성하고, 패킷 지연 공격에 대하여 평균 2.2초 이내로 탐지할 수 있음을 보였다. 실험 결과 소모 전류 Overhead는 초당 평균 2.5 mA, 트래픽 Overhead는 15% 발생하였고, 기존 제안된 탐지방법보다 효율적으로 Delay attack을 탐지할 수 있음을 밝혔다.

ABSTRACT

As IoT devices are widely used at home, IoT automation system that is integrate IoT devices for users' demand are gaining popularity. There is automation rule in IoT automation system that is collecting event and command action. But attacker delay the packet and make time that real state is inconsistent with state recongnized by the system. During the time, the system does not work correctly by predefined automation rule. There is proposed some detection method for delay attack, they have limitations for application to IoT systems that are sensitive to traffic volume and battery consumption. This paper proposes a practical packet delay attack detection technique that can be applied to IoT systems. The proposal scheme in this paper can recognize that, for example, when a sensor transmits an message, an broadcast packet notifying the transmission of a message is sent to the Server recognized that event has occurred. For evaluation purposes, an IoT system implemented using Raspberry Pi was configured, and it was demonstrated that the system can detect packet delay attacks within an average of 2.2 sec. The experimental results showed a power consumption Overhead of an average of 2.5 mA per second and a traffic Overhead of 15%. We demonstrate that our method can detect delay attack efficiently compared to preciously proposed method.

Keywords: IoT, Automation system, Delay attack

Received(08. 08. 2023), Modified(09. 25. 2023), Accepted(09. 25. 2023)

* 이 성과는 2023년도 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2021R1A2C2014428).

† 주저자, 2022572@korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr(Corresponding author)

I. 서 론

최근 많은 가정과 산업 현장에서 IoT(Internet of thing) 장비를 도입하여 사용하고 있으며, 2022년 기준 43.8%의 집에서 IoT 장비를 사용 중이다[20]. 기존 IoT 시스템은 IoT 장비별로 독립적으로 사용되었으나, 다양한 IoT 장비를 사용자의 편의에 맞게 통합하여 사용하고 싶은 요구가 많아지면서 IoT automation system이 발전되었다. IoT automation system에서는 이기종 IoT 장비에서 발생한 Event를 서버로 발송하고, IoT 서버는 사용자가 정해놓은 규칙에 따라 자동적으로 명령을 IoT 장비로 발송한다.

IoT automation system에서 발생할 수 있는 취약점에 대한 연구가 다양한 분야(e.g., 장비, 프로토콜, TAP(Triger-Action Programing) 모델, Cross-platform..)에서 진행되었다[12,20]. 이중 자동화 규칙의 취약점에 대한 연구도 활발히 진행되어 왔다[3,13-19]. 기존 자동화 취약점에 대한 연구는 특정한 조건에서 시스템을 불안정하게 만드는 규칙을 찾고, 이를 방지하는 수정된 규칙을 사용자에게 제시하였다. 하지만 공격자가 의도적으로 지연시킨 패킷에 대한 고려를 하지 않았다. Fu et al.의 연구에서는 공격자가 IoT 시스템의 패킷을 지연시켜 자동화 규칙의 정상적인 동작을 방해하는 공격이 제시되었다[4]. 이에 대한 탐지방법으로 Keep-alive Timeout의 시간을 줄이는 방법과 Application계층의 표준프로토콜에 Timestamp를 필드로 추가하는 것을 제안하였다. 하지만 전자의 방법은 과도한 트래픽을 유발하며, 후자의 방법은 메시지를 수신한 이후 타임스탬프를 비교하기 때문에 공격자의 목표달성을 막을 수 없다.

따라서 본 논문에서는 과도한 트래픽을 유발하지 않으면서, 일정시간 이내로 공격을 탐지하는 방법을 제안한다. 제안 방법은 Event / Command 메세지가 전송될 때, 패킷을 지연시키는 공격자를 우회하여 정당한 수신자에게 패킷이 전송되었음을 알리는 패킷을 전달한다. 본 논문에서 전송을 알리는 추가적인 패킷을 알람패킷이라고 한다. 알람패킷이 도착하였음에도 Event / Command 메세지가 도착하지 않으면 지연 공격이 발생하였음을 탐지할 수 있음을 보인다.

본 논문의 기여도는 다음과 같다.

1. 특정 프로토콜(e.g. ARP, TCP, MQTT..) 취약점에 대한 공격에 한정되지 않은 패킷 지연 공격 탐지 방법을 제안한다.
2. Event-trigger방식으로 기존 탐지방법 대비 공격 탐지 비용(i.e., 배터리 소모, 트래픽량)을 크게 줄인다.
3. 공격자가 충분한 목표를 달성하기 전에 탐지할 수 있음을 보인다.
4. 사용 중인 시스템을 수정하지 않고, 적용할 수 있는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구을 설명하고, 3장에서는 배경지식에 대하여 설명하였으며, 4장에서는 위협 모델 및 제안 방법을 설명하였으며, 5장에서는 제안 방법을 실험하고, 6장에서는 제안 방법을 평가하고, 7장에서는 제약사항 및 향후 연구를 제시하고, 마지막으로 8장에서는 결론을 맺는다.

II. 관련연구

2.1 Cross-Rule-Interference

Smart Home system에서는 자동화 규칙을 통하여 이기종의 IoT 장비를 통합하여 운용하기 때문에 많은 위협이 발생한다. 자동화 규칙간 간섭으로 발생하는 위협을 Cross-Rule-Interference라 하고,이에 대한 연구가 활발히 이루어지고 있다[3,14-19].

Chi et al.은 복수의 휴대폰 Application간의 발생할 수 있는 위협을 정적으로 분석하고, 이를 CAI(Cross-App Interference)라 하였다[14]. CAI는 IoT automation system에서 특정 조건에서 상충되는 동작이 유발되거나, 특정 조건을 통하여 의도되지 않은 동작이 발생할 수 있음을 보였다. 예를 들어, “TV가 켜졌을 때, 온도가 30도 이상이면 창문을 열어라.” 와 “TV가 켜졌을 때, 비가 내리면 창문을 닫아라”는 비가 오는날 온도가 30도 이상인 조건에서는 의도되지 않은 동작이 발생할 수 있다.

Celik et al.은 IoT 장비간의 동적 분석을 통하여 사용자가 안정적인 상태를 정의하고, 이를 벗어나는 행동을 차단하는 방법을 제안하였다[3]. Alhanahnah et al.은 IoT 장비가 동작할 때 생기는 물리적인 상호작용을 분석하여 발생할 수 있는 위협을 보이고, 자동화 규칙을 분석하여 이를 탐지 할 수 있는 방법(IoTCoM)을 제안하였다[13]. 하지만

위 연구들은 사용자가 설정한 규칙의 구조적인 취약점으로 인하여 발생할 수 있는 위협을 고려하였지만, 공격자에 의하여 발생한 IoT 메세지 전송 중단이나 메세지 지연은 고려하지 않는다.

2.2 IoT 메세지 지연 공격

Fu et al.은 IoT automation system에서 패킷을 지연시키는 공격자를 가정하고, 이로부터 발생할 수 있는 위협을 제안하였다[4]. 패킷이 지연되면 IoT 시스템은 Condition-Trigger-Action을 사용자가 의도한 상황과 다르게 판단한다. 공격자는 이를 통해 의도되지 않은 동작이 발생하거나 의도된 동작이 발생하지 않도록 유발한다. 이와 같은 시나리오는 기존 메세지를 폐기하는 공격과 다르게 Time-out 등 공격 탐지에 필요한 특징를 발생시키지 않아 탐지가 어렵다.

Fu et al. 연구팀은 패킷 지연 공격의 탐지방법으로 Event / Command 메세지에 대한 응답을 요구하고, Keep-alive 메세지의 간격을 줄이는 방법과 Time stamp를 확인하는 방법을 제안했다. 하지만 이 연구에서 제안한 탐지방법은 한계점이 명확하다. 전자는 불필요한 패킷 전송을 지속적으로 발생시켜, 네트워크와 배터리 소모에 큰 악영향을 미친다. 또한 단순히 Keep-alive 시간 간격을 줄이면 네트워크 또는 장비의 문제로 지연이 발생하게 되었을 때, 문제가 발생할 수 있다. 후자는 기존 프로토콜을 수정하여 Timestamp를 메세지 필드로 추가하고, 이를 수신자 측에서 확인한다. 하지만 이 방법은 수신자측에서 확인한 이후 공격이 탐지되기 때문에, 공격자가 원하는 만큼 지연을 발생시킬 수 있다.

2.3 지연 공격 탐지

Ranganathan et al.는 이벤트 시간과 센서 데이터를 정확하게 맞출 수 있는 시간 동기화 프로토콜들을 조사 분석하였다[17]. 하지만 사용중인 IoT system에 적용하기에 프로토콜을 수정해야하는 소요가 있다. Jiang et al.는 IoT automation system에서 머신러닝 기반으로 시간간격을 학습할 수 있는 새로운 학습모델을 통하여 지연공격을 탐지하는 방법을 제안하였다[16]. 하지만 서버로그를 기반으로 한 머신러닝 방법은 공격자가 메시지를 충분히 지연시킨 이후 서버에 로깅이 될 때 이상을 탐지할 수 있다. 이러한 방법은 공격자의 목표가 달성된 이후에 공격을 탐지할 수 있으며, 사용자가 규칙을 새롭게 추가하고 삭제할 때마다 새롭게 학습해야 하는 어려움이 있다.

III. 배경 지식

3.1 Smart Home환경의 IoT automation system 네트워크

Fig. 1.과 같이 Smart Home환경에서 IoT automation system의 네트워크 구성은 서버가 설치된 위치와 IoT 장비가 지원하는 네트워크 방식에 따라 나눌 수 있다.

서버가 설치된 위치에 따라 구분하면 첫 번째는 사용자의 IoT 장비와 같은 네트워크 대역에 서버를 설치하는 로컬 서버가 있다. 대표적으로 Apple homekit나 open source로 제공되는 Home Assistant가 있다[22,23]. 두 번째, 사용자의 네트워크 대역과 다른 네트워크에 설치된 Cloud 서버가 있다. Samsung에서 지원하는 Smartthings나 Amazon에서 제공하는 AWS IoT가 있다[24,25].

IoT 장비가 지원하는 네트워크 방식에 따라 나누면 WiFi를 지원하는 장비와 아닌 장비로 나눌 수 있다.WiFi를 사용하는 장비는 Home WiFi Router에 WiFi를 사용하여 연결되고, Zigbee / Zwave를 사용하는 장비는 IoT hub / Bridge와 Zigbee / Zwave를 사용하여 연결된다. 연결된 IoT hub / Bridge는 WiFi를 사용하여 Home WiFi Router에 연결된다. Router는 로컬 서버는 WiFi를 사용하여 연결되고, 클라우드 환경에서 동작하는 IoT 서버는 라우팅 규칙에 따라 WAN을 거쳐 IoT 서버로 전송된다.

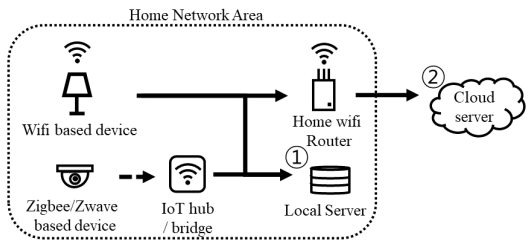

Fig. 1. Network of IoT Systems According to Server Location

3.2 자동화 규칙

Fig. 2.는 Apple Homekit에서 자동화 규칙을 설정하는 어플리케이션 화면이다. 이와 같이 IoT automation system에서는 자동화 규칙을 활용하여 IoT 장비를 운용한다. 자동화 규칙은 플랫폼 별로 차이는 있으나 크게 Trigger - Condition - Action 으로 이루어져 있다. 어떤 사건이 발생하여 IoT 장비에서 메시지가 IoT 서버로 발송되면 서버는 이를 Trigger로 활용한다. 이때 발송된 메세지를 Event 메세지라고 한다. Trigger가 수신되면 서버에 저장된 Event 메세지를 Condition으로 활용하여 사전에 정의된 자동화 규칙의 참, 거짓을 판단한다. 참으로 판단되면 Action을 수행하고 동작이 필요한 IoT 장비로 패킷을 전송한다. 이때, IoT 장비로 발송되는 패킷을 Command 메세지라고 한다. 만약 거짓으로 판단되면 Action을 수행하지 않는다.

3.3 중간자 공격(man in the middle attack)

패킷 지연 공격을 위하여 중간자 공격이 선행되어야 한다. 중간자 공격은 통신하고 있는 두 당사자 사이에 들키지 않게 끼어들어 당사자들이 교환하는 통신내용을 바꾸거나 도청하는 공격 기법이다. 공격자는 가로챈 패킷의 내용을 확인하여 기밀성을 침해할 수 있고, 패킷을 조작하여 무결성을 침해할 수 있다. 마지막으로 패킷을 누락시켜 정상적인 사용자의 서비스 사용을 방해하여 가용성을 침해할 수 있다. 중간자 공격의 기법으로 ARP Spoofing, DNS Spoofing 등 여러 기법이 널리 알려져 있고, 각 기법의 탐지방법에 대한 연구가 활발히 진행되고 있다[1,2]. 하지만 최근 상용 IoT 서버와 장비를 조사한 연구에서는 여전히 ARP spoofing 등 공격기법을 활용한 중간자 공격에 취약하다는 것이 밝혀졌다[11].

기존 중간자 공격은 주로 패킷의 내용을 확인하거나, 패킷의 내용을 위조하였고, 이는 TLS 등 암호화를 통하여 막을 수 있었다. 하지만 최근 연구에서 메시지 지연공격은 TLS를 적용하고 있는 IoT 시스템에서도 비정상적인 동작을 유발할 수 있음이 밝혀졌다[4].

IV. 시스템 모델

4.1 시스템 모델

Smart Home 환경의 IoT automation 시스템에 대한 모델은 사용하는 IoT 서버와 IoT 장비에 따

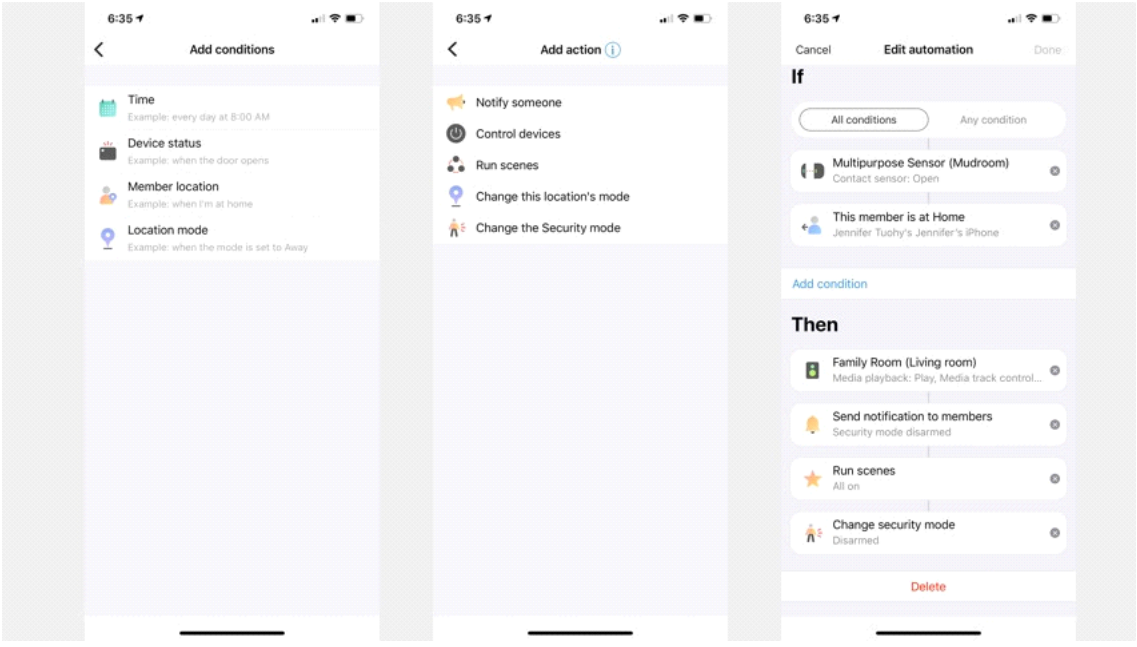

Fig. 2. Structure of Automation Rule in SamSung SmartThings Application[26]

라 2가지 유형으로 분류된다. 먼저 IoT 장비와 IoT 서버 모두 동일한 네트워크 대역에 위치한 경우와 IoT 장비와 IoT 서버가 다른 네트워크 대역에 위치한 경우로 구분할 수 있다. 본 논문에서는 위 2가지 경우 모두를 고려한다.

IoT 장비와 IoT 서버가 모두 동일한 네트워크 대역에 위치한 경우는 Zigbee / Zwave 통신 기반 장비는 IoT hub / bridge와 연결되고, WiFi 통신 기반 장비와 IoT hub / bridge는 WiFi AP와 연결되어 IoT 서버와 패킷을 교환한다. IoT 장비와 IoT 서버가 다른 네트워크 대역에 위치한 경우는 라우터 또는 라우터의 역할을 하는 WiFi AP가 다른 네트워크 대역에 위치한 IoT 서버로 패킷을 전달한다.

IoT 장비는 이벤트가 발생하면, Event message를 생성하여 IoT 서버로 전달한다. IoT 서버는 전달받은 Event message를 Trigger로 하여 자동화 규칙을 판단한다. 자동화 규칙 중 참인 규칙이 있다면 Command 메시지를 해당하는 IoT 장비로 전달한다.

4.2 공격자 모델

공격자의 목표는 Timeout 또는 연결 끊김을 발생시키지 않고, 공격목표인 IoT 장비 / 서버의 Event / Command 메세지를 지연시키는 것이다. 공격 대상은 TCP / IP를 활용하여 메세지를 전달하는 IoT automation system이다.

이러한 목표를 위해 공격자는 Smart Home 환경에서 하나의 WiFi기반 장비를 제어하고, 이를 사용하여 WiFi 트래픽을 탐지한다. 공격자는 암호화된 트래픽의 메타데이터(e.g. 패킷 헤더, 길이, 빈도..)를 분석하여 Event / Command 메세지 및 자동화 규칙 그리고 Time-out 시간을 추론할 수 있다[4-6]. 공격자는 공격 대상 IoT 장비의 TCP 세션을 가로채고, 일정시간 지연 후 패킷이 수신된 순서대로 재전송하여 Timeout 메세지가 발생하지 않는 범위 내에서 패킷을 지연시킬 수 있다.

4.3 공격 시나리오

Fig. 3.는 4가지 타입의 Delay attack(i.e., State-update Delay attack, Action Delay attack, Suprious Execution, Disabled Execution)을 나타낸다.

- State-update Delay attack : 공격자는 Event 메시지를 지연시켜 서버에서 Event의발생을 늦게 인지하도록 한다.
- Action Delay attack : 공격자는 Command 메시지를 발생시키는 Event 메시지를 지연하거나, 발생된 Command 메시지를 지연하여 IoT 장비의 동작을 지연한다.
- Spurious Excution : Condition 되는 Event 메시지를 지연한다. 이를 통해 Tigger가 발생했을 때, Condition의 불일치가 발생하여 IoT 서버에서 Command메시지를 발송한다. Fig. 3.의 예시를 들면, '사용자가 집에

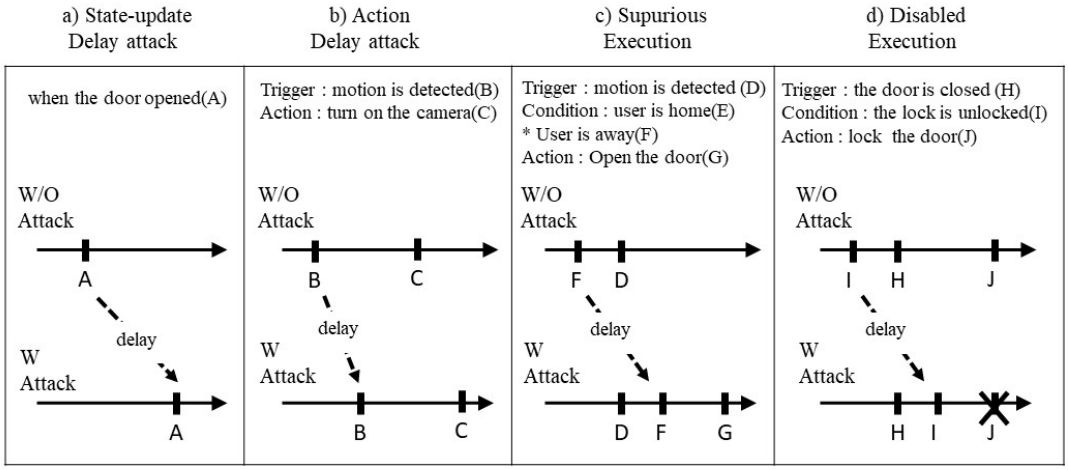

Fig. 3. Delay Attack Types

서 벗어났다.'는 Event 메시지를 지연시켜, 움직임이 발생했을 때 '사용자가 집에 존재한다.'라는 Condtion으로 인식하도록 한다. IoT 서버는 잘못 인식된 자동화 규칙에 따라 문을 여는 Command 메시지를 발생시킨다.

· Disabled Excution : Spurious Excution과 비슷하지만, 반대로 Action을 발생시키지 않도록 Condtion을 지연한다. Fig. 3.의 예시를 보면 공격자는 잠금장치가 풀린 Event 메시지를 지연시킨다. IoT 서버는 문이 닫힌 Trigger가 발생했을 때 잠금장비가 이미 잠겨 있다고 인식한다. 따라서 문이 잠긴 Action을 발생시키지 않는다.

본 논문에서는 Fig. 4.의 시나리오에 따라서 실험을 진행한다. 4가지 공격 유형에 대하여 공격이 없는 상황과 공격이 있는 상황 모두 발생하며, 공격자는 공격 유형을 발생시킬 수 있는 패킷을 선택적으로 지연한다.

V. 제안 방법

공격자가 충분한 지연을 만들기 전에 지연 공격을 탐지하기 어려운 이유는 IoT 시스템에서 패킷이 언제 발송되었는지 알 수 없기 때문이다. 일반적인 패킷은 공격자가 중간에서 가로채어 충분한 시간 동안 지연을 할 수 있다. 하지만 브로드캐스트 패킷은 중간자 공격자가 있더라도 정보전송을 알리는 패킷이 수신자에게 도착하는 것을 보장한다.

제안 방법은 Event 메세지 또는 Command 메세지가 발송될 때 메세지 송신되었음을 알리는 알람 패킷을 발송한다. 별도의 패킷은 공격자가 가로챌 수 없도록 데이터링크 계층에서 브로드캐스트되며, 브로드캐스트된 패킷은 Event 및 Command 메세지와 유사한 네트워크에서 전송되어 패킷과 유사한 네트워크 지연을 겪는다. 이를 통해 패킷이 의도적으로 지연되었는지, 네트워크 지연으로 늦게 도착하였는지 구분할 수 있다.

알람패킷의 브로드캐스트를 위하여 본 논문에서는 ICMP 프로토콜을 사용하여 구현하였다. ICMP 프로토콜은 네트워크 통신 문제를 진단하는데 사용한다. 하지만 본 논문에서는 ICMP 프로토콜을 단순히 브로드캐스트를 위한 도구로 사용한다. 일반적인 ICMP 프로토콜의 동작에 영향을 주지 않기 위하여 Type을 Reserved 값인(44)으로 사용하였다.

5.1 Event 메세지 발송시

Fig. 5.는 Event 메시지 발송시 Sequence diagram이다. IoT 장비는 Event 메시지를 발송할 때, Fig. 5.처럼 Event 메시지 내용과 Time stamp를 Hash 함수에 넣어 획득한 값과 Timestamp를 Broadcast 패킷에 넣어 발송한다. IoT 서버는 수신한 패킷을 비교하여 검증한다. 이때, 공격자가 Event 메시지를 지연시켰다면, Broadcast 패킷만 도착하게 되어 IoT 서버는 Delay attack을 탐지할 수 있다.

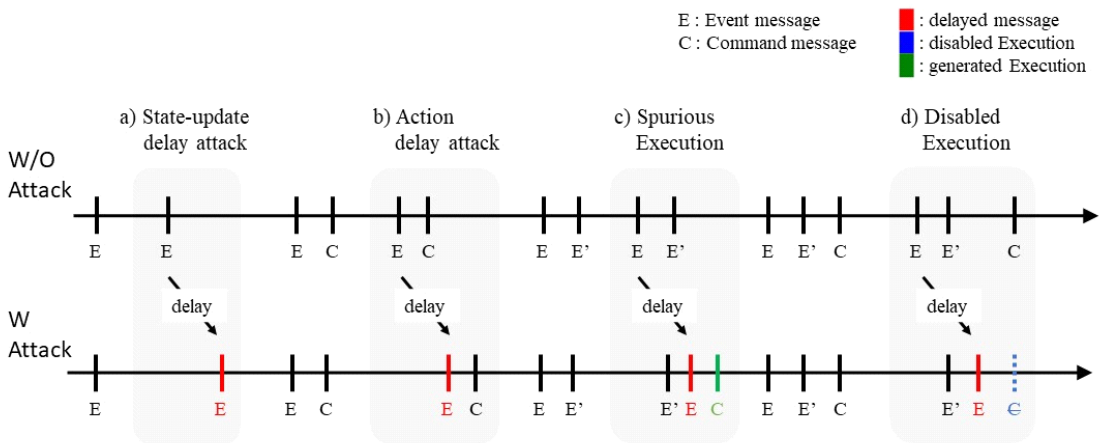

Fig. 4. Attack scenario

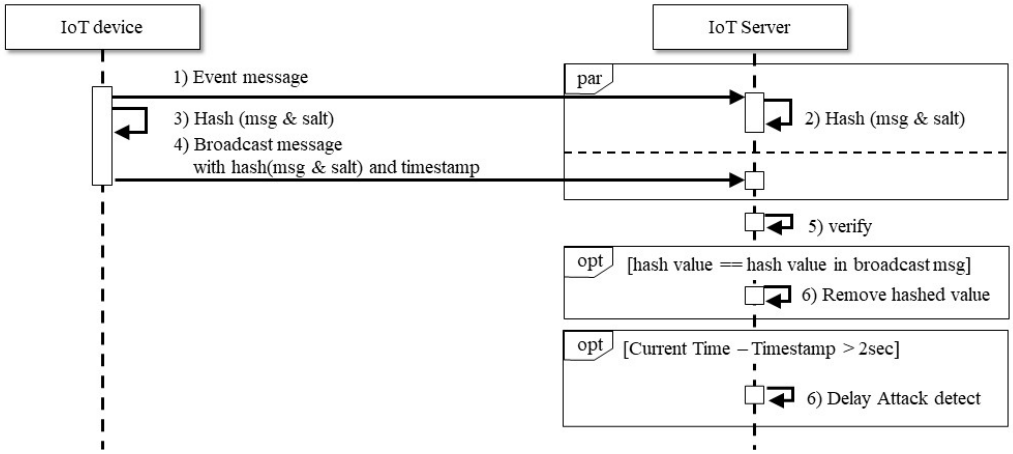

Fig. 5. Sequence diagram when sending Event message

5.2 Command 메세지 발송시

Fig. 6.는 Command 메시지 발송시 Sequence diagram이다. Command 메시지는 IoT 서버가 IoT 장비에 발송한다. Command 메시지를 수신한 IoT 장비는 메시지 내용과 Timestamp값을 Hash 함수에 넣어 획득한 값을 Broadcast 패킷에 넣어 발송한다. 이때, 공격자가 Command 메시지를 지연시켰다면, Broadcast 패킷을 수신하지 못한 IoT 서버가 Delay attack을 탐지 할 수 있다.

VI. 평 가

6.1 실험 환경

제안 방법의 평가를 위하여 IoT 시스템의 IoT 장비와 라우터, IoT 서버를 라즈베리파이로 구현하였다. Table 2은 평가간 사용한 장비와 해당 장비가 실험간 시뮬레이션하는 역할을 나타내였다. 전력 소모 Overhead 측정을 위한 장비는 Monsoon Power Monitor를 사용하였다[21]. Monsoon Power Monitor는 리튬배터리로 동작하는 장비에 대한 소모 전력을 측정하는 용도로 사용되고 있다. Fig. 7.처럼 IoT 장비 역할을 하는 라즈베리파이를 Monsoon Power Monitor 전원 공급 케이블로 연결하여 4.54 V를 인가하고, 측정된 데이터는

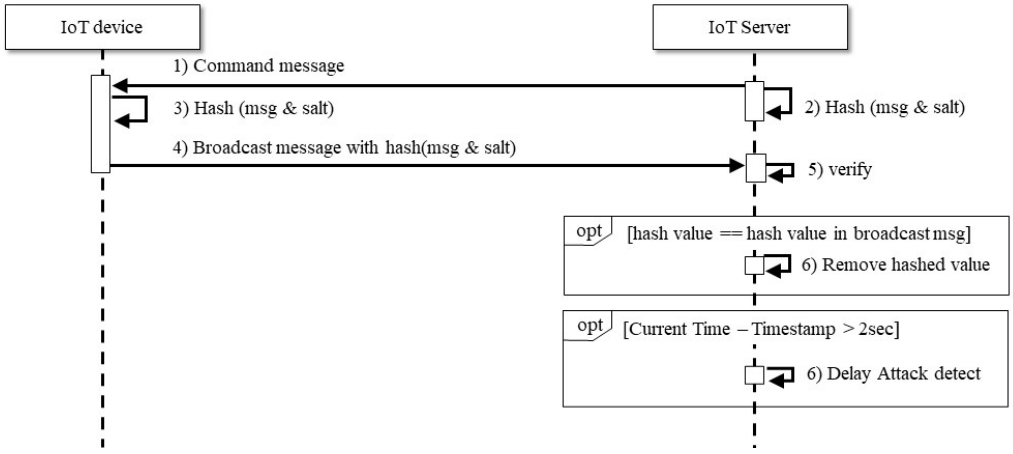

Fig. 6. Sequence diagram when sending Command message

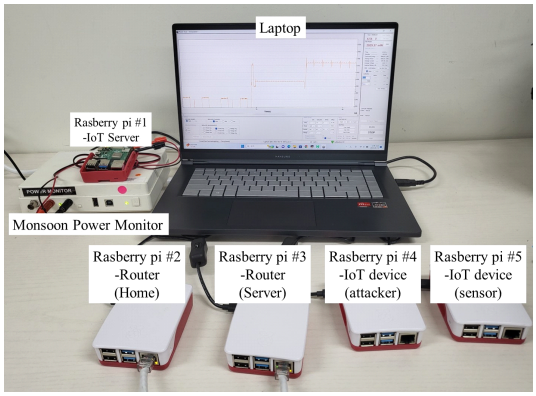

Fig. 7. Experimental setup

Monsoon Power Monitor와 USB로 연결된 노트북에서 수집된다. 구성된 실험 환경에서 Fig. 3. 시나리오에 따라 Event 메시지와 Command 메시지를 발송하고, 공격자는 선택적으로 메시지를 지연한다.

6.2 지연 공격 탐지

Fig. 4.의 공격 시나리오에 따라 4가지 공격 유형에 대하여 1,200번 실험을 진행하였다. 실험 진행간 IoT 장비는 계획된 시나리오에 따라 Event Message를 서버로 전송하였으며, 브로드캐스트 패킷을 함께 전송하였다. IoT 장비가 Command Message를 수신할 경우에도 브로드캐스트 메시지를 발송하였다. IoT 서버는 브로드캐스트 패킷을 받은 시간부터 패킷의 의도적인 지연 여부를 확인하였다. 실험 결과 미탐지된 공격은 없었으며 공격자가 의도적으로 지연한 패킷을 모두 탐지하였다.

Fig. 8.은 4가지 공격 유형에 대하여 탐지시간을 나타낸 그래프이다. 본 논문에서 제안한 방법이 각 패킷에 대한 지연 여부를 확인하는 방법이기 때문에, 공격 유형에 따라 탐지시간이 유의미하게 달라지지 않았으며, 평균 2.20초 내로 공격을 탐지하였다.

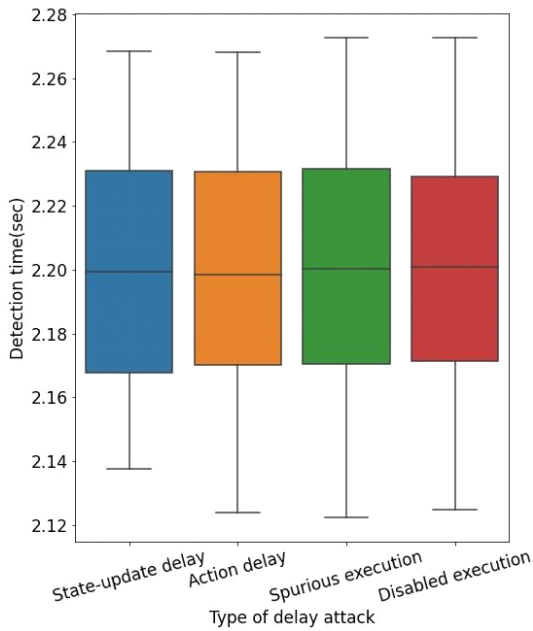

Fig. 8. Detection time of delay attack

6.3 기존 Delay Attack Detection과 비교

Table 1.는 Delay attack 공격 유형별 기존 탐지방법과 본 논문에서 제안 방법의 비교를 나타낸다.

ML 기반의 Delay attack 탐지는 이벤트 순서와 Time interval을 고려한다[16]. 이 방법은 이벤트 순서와 관련이 있는 Spurious Execution attack과 Disabled attack은 탐지가 가능하지만, 독립적으로 발생하는 State-update Delay attack과 Action Delay attack은 탐지가 불가능하다. Timestamp를 통한 탐지는 4가지 공격유형을 탐지할 수 있으나 State-update Delay attack과 Action Delay attack, Spurious Execution 공격 효과를 방지하기는 어렵다[4]. Fig. 3.의 Spurious Execution 공격을 예로 들면 공격자는 Event 메시지를 delay 시킨다. IoT 서버는

Table 1. Evaluation by detection method

	State-update Delay attack	Action Delay attack	Spurious Execution	Disabled Execution
ML based[17]	X	X	O	O
Timestamp checking[3]	△	△	△	O
Shortening timeout[3]	O	O	O	O
Proposed method	O	O	O	O

Table 2. Experimental environment

Device	Os	Model
Router(Home)	Raspberry pi OS	Raspberry Pi 4 B (RAM 2GB)
Router(Server)		
IoT Server		
IoT device (attacker)		
IoT device (sensor)		

Command 메시지를 실행한 이후에 Event 메시지를 수신하고 Timestamp를 확인하게 된다. Shortening timeout를 통한 Delay attack 탐지는 4가지 공격 유형 모두를 탐지할 수 있다[4]. 이 방법은 Application 계층의 Hearbeat메시지 전송 간격을 줄이는 방법으로 구현한다. 하지만 이 방법은 배터리 소모와 트래픽량에 민감한 IoT 장비에 적용하기에 어려움이 있다. 이에 대한 평가는 6.4에서 자세하게 평가한다. 본 논문에서 제안 방법은 4가지 공격 유형을 모두 2.2초 내 에 탐지했다.

6.4 배터리 소모 Overhead

Fig. 8.의 상단에 위치한 그래프는 본 논문에서 제안 방법을 적용하였을 경우와 적용하지 않았을 경우 시간에 따른 소모 전류값을 보여준다. 2가지 경우 모두 이벤트 발생시 가장 높은 소모 전류값이 측정되었으며, 이벤트가 발생하지 않는 시간동안 낮은 소모 전류가 측정되었다. 제안 방법을 적용했을 때 초당 평균 소모 전류는 411.4 mA로 측정되었고, 제안 방법을 미적용했을 때는 408.9 mA로 초당 평균 2.5 mA의 소모 전류 Overhead가 발생한다.

Fig. 9.의 하단에 위치한 그래프는 Delay attack의 4가지 유형을 모두 탐지할 수 있는 Shortening timeout 방법을 적용하였을 경우와 본 논문에서 제안 방법을 적용하였을 경우의 소모 전류를 비교한 그래프이다. 실험을 위하여 Keep-alive time을 5초로 설정하였다. Shortening timeout 방법을 적용하였을 때에도 이벤트가 발생했을 때 가장 높은 소모 전류가 측정되었다. 하지만 이벤트가 발생하지 않는 동안에 Keep-alive 메시지의 송수신으로 지속적으로 높은 소모 전류가 측정되었다.

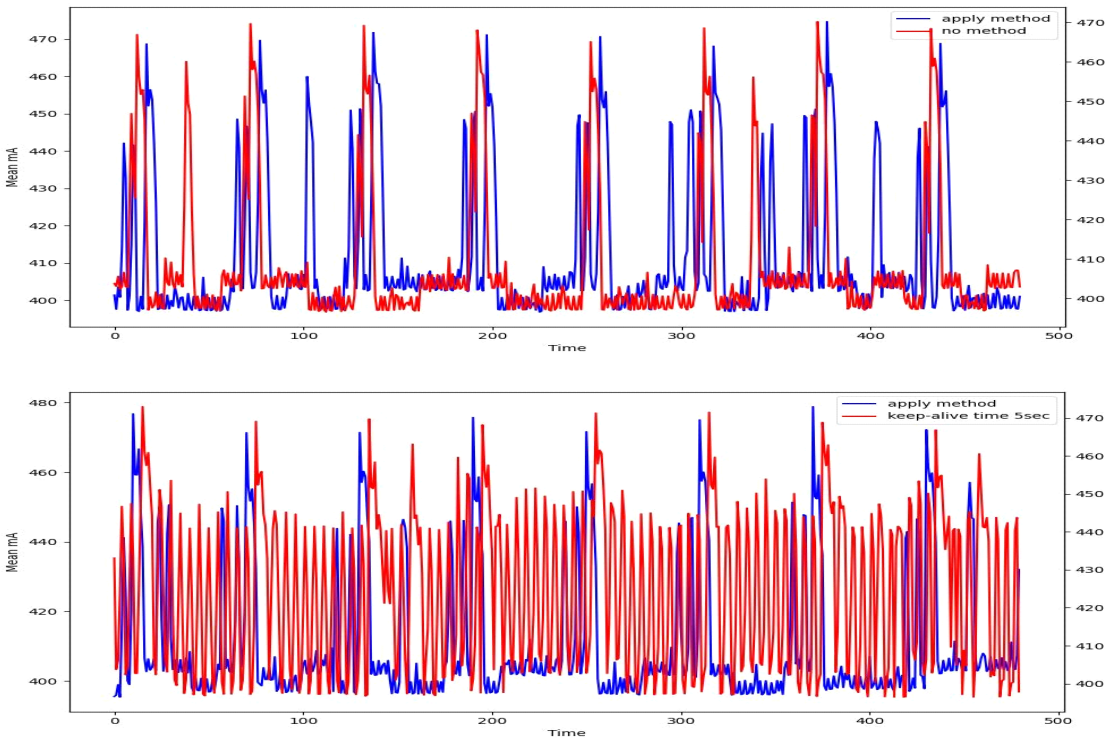

Fig. 9. Power Consumption Measurement

Keep-alive 5초 적용시 초당 평균 소모 전류는 424.5 mA로 본 논문에서 제안 방법(411.4 mA)보다 13.1 mA가 높게 측정되었다.

배터리에 민감한 IoT 장비는 배터리 소모량을 줄이기 위한 여러 가지 방법을 적용하는데, 대표적인 것이 절전모드이다. 절전 모드는 IoT 장치의 주변장치를 셧다운 하거나 정지시키는 방식으로 배터리 소모를 낮추는 방법이다. 하지만 Shortening timeout 방식은 지속적으로 통신이 필요하기 때문에 절전모드를 유지하는 시간이 짧아지고, 절전모드의 효과를 저해한다.

6.5 트래픽 Overhead

IoT 장비의 사용량에 따른 트래픽 Overhead를 측정하는 실험이다. 제안 방법은 Event나 Command 메시지 발생시 1개의 패킷이 추가로 발생한다. 본 논문에서 구현한 브로브캐스트 패킷은 94 byte로 패킷 헤더 34 byte와 데이터 60 byte로 이루어져 있다. 데이터는 해시 값 32 byte와 Timestamp 28 byte로 구성되어 있다.

Table 3.은 본 논문에서 제안 방법과 기존 연구된 Shortening timeout 방법에 대한 트래픽 Overhead를 측정한 표이다. Fig. 6.에 표시된 공격 시나리오를 1회 진행하는 20분 동안 Event 및 Command 메시지는 16번 생성되었고, 이에 따른 트래픽은 총 8,149 byte 발생하였다. 제안 방법을 적용할 경우 브로드캐스트 패킷이 1,504 byte 추가로 생성되어 전체 트래픽은 9,653 byte가 발생하였다. 본 논문에서 제안 방법을 적용하지 않았을 때(8,149 byte)에 비하여 18.46%의 Overhead를 발생시켰다. Shortening timeout 방법을 적용하였을 때는 5초마다 Keep-alive request (68 byte), Keep-alive response(68 byte) 메시지가 발생한다. 동일한 공격 시나리오를 20분 동안 진행한 경우 트래픽은 총 39,301 byte 발생하였고, Shortening timeout 방법을 적용하지 않았을 때의 트래픽량(8,149 byte)에 비하여 382.28% Overhead가 발생하였다.

Table 3. Traffic Overhead based on method

Method	Traffic voulme (byte)	Overhead (%)
Shortening timeout[4]	39,301	382.28
our method	9,653	18.46

Table 4. Traffic Overhead based on using interval

Using interval	Traffic voulme(byte)		Overhead (%)
	non-apply	apply our method	
120sec	5,352	6,261	16.98
60sec	7,882	9,666	22.63
30sec	12,363	16,209	31.11
20sec	16,896	22,557	33.50
10sec	30,545	41,399	35.53
5sec	57,126	78,965	38.23

Table 4.는 제안 방법의 IoT 장비 시간당 사용량에 따른 트래픽 Overhead를 측정한 결과이다. 120초마다 한번씩 동작하는 경우에 가장 낮은 트래픽 Overhead인 16.98%가 측정되었고, 5초마다 한번씩 동작하는 경우에 가장 높은 트래픽 Overhead인 38.23%가 측정되었다.

위 실험결과를 고려할 때, 본 논문에서 제안 방법은 기존 논문에서 제안 방법에 비하여 트래픽 Overhead를 적게 발생시킨다. Event-trigger방식으로 동작하기 때문에 IoT 장비의 사용량에 따라 트래픽 Overhead가 변화하는 점을 확인하였다.

VII. 제약사항 및 향후연구

본 연구는 시뮬레이션 환경에서 실험하였기 때문에 실제 환경에서 발생하는 영향에 대한 고려되지 않았다. 예를 들면, 클라우드 서버 사용시 클라우드 서버의 사용량 급증으로 인한 데이터 처리시간 지연이 과도하게 발생할 수 있다. 이렇게 실제 환경에서 발생한 의도하지 않은 지연은 false alarm을 유발할 수 있다.

향후 연구에서는 네트워크 혼잡이나, 서버의 사용량 과다로 인한 데이터처리 속도 저하 등 실제 환경에서 발생하는 영향을 고려한 연구가 이루어져야 한다.

VIII. 결 론

본 논문은 IoT automation System에서 패킷을 지연시켜 발생할 수 있는 공격을 탐지하는 방안을 제안하였다.

제안 방법은 기존 탐지 방법들과 비교하여 IoT 장

비의 소모 전력을 크게 증가시키지 않으면서, 현재 사용 중인 IoT 시스템에도 쉽게 적용할 수 있다. 예를 들어, 스마트 CCTV 등의 설치 용이성을 가진 장비는 배터리를 사용하는데, 시간당 소모 전력과 트래픽 오버헤드는 장비의 배터리 교체 주기에 큰 영향을 미칩니다.

평과 결과에 따르면 지연 공격 유형 4가지 모두를 탐지 할 수 있었으며, 평균 2.2 초의 공격탐지 시간이 소요되었으며, 누락된 공격 탐지는 발생하지 않았다. 제안 방법으로 인한 소모 전류 Overhead는 평균 2.5 mA로 발생하였으며, Event 발생마다 94 byte의 트래픽 Overhead가 발생하였으며 기존 논문에 비하여 효율적으로 공격을 탐지할 수 있음을 밝혔다.

References

[1] Morsy, Sabah M., et al., "D-ARP: An Efficient Scheme to Detect and Prevent ARP Spoofing." IEEE Access, 10, pp. 49142-49153, May. 2022.

[2] Prasad, Arvind, et al., "Defending ARP Spoofing-based MitM Attack using Machine Learning and Device Profiling." 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). pp. 978-982, Nov. 2022.

[3] Celik, Z. Berkay, et al., "IoTGuard: Dynamic Enforcement of Security and Safety Policy in Commodity IoT." Network and Distributed Systems Security (NDSS) Symposium. pp. 24-27, Feb. 2019.

[4] Fu, Chenglong, et al., "Iot phantom-delay attacks: Demystifying and exploiting iot timeout behaviors." 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 428-440, Jun. 2022.

[5] Acar, Abbas, et al., "Peek-a-boo: I see your smart home activities, even encrypted!." Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 207-218, July. 2020.

[6] Luo, Yuan, et al., "Context-rich privacy leakage analysis through inferring apps in smart home iot." IEEE Internet of Things Journal vol. 8, no. 4, pp. 2736-2750, Aug. 2020.

[7] Wang, Qi, et al., "Fear and logging in the internet of things." Network and Distributed Systems (NDSS) Symposium. pp. 18-21, Feb. 2018.

[8] Celik, Z. Berkay, et al., "Soteria: Automated {IoT} safety and security analysis." 2018 USENIX Annual Technical Conference (USENIX ATC 18). pp. 147-158, July. 2018.

[9] Hariri, Ali, Nicolas Giannelos, et al., "Selective forwarding attack on iot home security kits." Computer Security: ESORICS 2019 International Workshops, CyberICPS, SECPRE, SPOSE, and ADIoT, Luxembourg City, Luxembourg, pp. 26-27, Sep. 2019.

[10] OConnor, T. J., William Enck, et al., "Blinded and confused: uncovering systemic flaws in device telemetry for smart-home internet of things." Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks. pp. 140-150, May. 2019.

[11] Huang, Danny Yuxing, et al., "Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 4, no. 2, pp. 1-21, Jun. 2020.

[12] Zhang, Pengfei, et al., "Secure location of things (slot): Mitigating localization spoofing attacks in the internet of things." IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2199-2206, Dec. 2017.

[13] Alhanahnah, Mohannad, et al., "Scala

ble analysis of interaction threats in iot systems." Proceedings of the 29th ACM SIGSOFT international symposium on software testing and analysis. pp. 272-285, Jul. 2020.

[14] Chi, Haotian, et al., "Cross-app interference threats in smart homes: Categorization, detection and handling." 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 411-423, Jul. 2020.

[15] Wang, Qi, et al., "Charting the attack surface of trigger-action IoT platforms." Proceedings of the 2019 ACM SIGSAC conference on computer and communications security. pp. 1439-1453, Nov. 2019.

[16] Jiang, Chenxu, et al., "Effective Anomaly Detection in Smart Home by Integrating Event Time Intervals." Procedia Computer Science. Vol. 210, pp. 53-60, Oct. 2020.

[17] Ranganathan, Prakash, and Kendall Nygard. "Time synchronization in wireless sensor networks: A survey." International Journal of Ubiquitous Computing. vo. 5, no. 1, pp. 92-102, Jun. 2019.

[18] Huang, Danny Yuxing, et al., "Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale." Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. vol. 4, no. 2, pp. 1-21. Jun. 2020.

[19] Kodali, Ravi Kishore, et al., "IoT based smart security and home automation system." 2016 international conference on computing, communication and automation (ICCCA). pp. 1286-1289, Apr. 2016.

[20] Statista, "Smart Home - Market Data & Forecast 2022", https://www.statista.com/study/42112/smart-home-report/ accessed, 2023.07.18

[21] Msoon, "Low Voltage Power Monitor Product Documentation", https://www.msoon.com/lvpm-product-documentation, accessed 2023.07.18

[22] Apple, "Create scenes and automations with the Home app", https://support.apple.com/en-us/HT208940, accessed 2023.07.18

[23] Home Assistant, "Automating Home Assistant", https://www.home-assistant.io/docs/automation/, accessed 2023.07.18

[24] SAMSUNG, "Setting up an Automation in Samsung SmartThings", https://www.samsung.com/sg/support/mobile-devices/setting-up-an-automation-in-samsung-smartthings-with-certain-time-and-device-condition/, accessed 2023.07.18

[25] AMAZON, "AWS IoT for the Connected Home", https://aws.amazon.com/ko/iot/solutions/connected-home/, accessed 2023.07.18

[26] The ambient, "Samsung SmartThings: Use the app, features and hubs for a better smart home", https://www.the-ambient.com/guides/samsung-smartthings-guide-smart-home-163, accessed 2023.07.18.

〈저자소개〉

김 영 덕 (Youngduk Kim) 정회원
2016년 2월: 육군사관학교 전자공학과 졸업
2022년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 IoT, 로봇, 무인이동체 보안, 네트워크

최 원 석 (Wonsuk Choi) 종신회원
2008년 2월: 서울시립대 수학과 졸업
2013년 2월: 고려대학교 정보보호대학원 석사 졸업
2018년 8월: 고려대학교 정보보호대학원 박사 졸업
2018년 9월~2020년 2월: 고려대학교 정보보호연구원 연구교수
2020년 3월~2023년 2월: 한성대학교 IT융합공학부 조교수
2023년 2월~현재: 고려대학교 정보보호대학원 조교수
〈관심분야〉 센서 보안, 자동차 보안, 암호 프로토콜

이 동 훈 (Dong hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학사 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월: 고려대학교 전산학과 조교수
1997년 3월~2001년 2월: 고려대학교 전산학과 부교수
2001년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호 프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET기술