정 보 보 호 학 회 지
제24권 제6호, 2014. 12

# 사물인터넷과 MQTT 기술

심 승 현*, 김 학 범**

요 약

최근 정보통신기술 분야의 최대 관심사는 모든 사물간의 통신이 이루어지는 Internet Of Things, 더 나아가 Internet Of Everythings 분야이다. 이러한 흐름에 발맞추어 사물과 관련된 다양한 기술이 점진적으로 발달하였고, 누구나 쉽게 사물간의 인터넷 결합을 이용하여 Ubiquitous 시대를 맞이해 인간의 편리성을 위한 다양한 서비스를 이용할 수 있을 것으로 예상된다. 앞으로의 우리 생활에 점차 밀접한 관계를 맺음에 따라 다양한 기술들이 개발 및 접목될 것으로 예상이 된다. 본 논문에서는 Internet Of Things의 시대가 발전함에 따라 이에 관련된 현재 활용되는 기술 중 MQTT(Message Queue Telemetry Transport)에 대해 기술 할 것이며 이 기술과 관련된 다양한 정보 침해에 대한 향후대책 및 대응방안에 대해 기술한다.

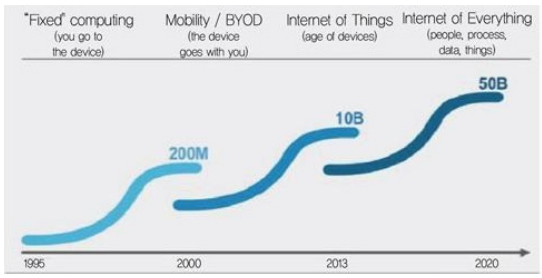
## Ⅰ. 서 론

ICT의 기술진보에 따라 인터넷의 질적 변화도 급속히 이루어지고 있다. 1990년대는 PC를 연결하는 유선 인터넷시대, 2000년대에는 모바일과 연결하는 이동통신의 무선인터넷시대, 2010년대에는 ICT가 지능을 갖고 사물과 사람을 연결하는 초고속인터넷시대로진화하고 있다. 최근 스마트기기 및 SNS 등의 등장과 활용으로 인간간 연결은 더욱 활발하게 진행되고 있을 뿐만 아니라, 사물통신(M2M : Machine to Machine), 사물인터넷(IoT : Internet of Things), 만물인터넷(IoE : Internet of Everything) 등 IT의 기술적 발전에 따라 인간과 사물, 사물과 사물 등으로 연결 범위가 확대되고 있는 추세이다.[1]

[그림 1][2]은 전 세계적으로 연결된 디바이스들을 나타내고 있다. 2020년까지 다른 디바이스들에 비해 사물통신을 위한 Machine to Machine 디바이스가20억개에서 120억개로 눈에 띄게 늘어날 것임을 보여주고 있다.

현재 전 세계 1.5조개의 단말기 중에 인터넷에 연결된 것은 100억대수준이며 99% 이상이 아직 연결되어 있지 않기 때문에 [그림 2][3]에 따라 시스코는 향후 2020년 사물인터넷의 경제적 가치를 14조 4천만불이라는 엄청난 규모로 보고 있다. 클라우드, 모바일, 소셜네트워크, 빅데이터등의 복합적인 발전이 이 시장을 계속 견인할 것으로 예상이 되는데 재물관리에 2조5000만불, 고용관리에 2조5000만불, 공급망관리 2조7000만불, 고객서비스에 3조7000만불, 서비스혁신에 3조불 규모


〔그림 1〕 M2M 시장현황과 국내외 성장 전망




〔그림 2〕 인터넷에 연결된 단말수 현황 및 전망

* 동국대학교 국제정보대학원 (2014125543@dongguk.edu)
** 동국대학교 국제정보대학원 / ㈜ 이너버스 (khb0305@dongguk.edu)

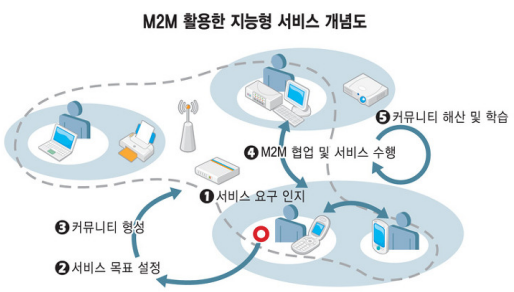
| | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 연평균 변화(%) |
|---|---|---|---|---|---|---|---|---|
| Public Safety & Urban Security | 7.9 | 13.8 | 21.6 | 28.7 | 34.9 | 41.8 | 48.7 | 28.7 |
| Retail | 8.8 | 15.3 | 34.0 | 32.0 | 38.9 | 46.5 | 54.2 | 28.8 |
| Healthcare | 4.0 | 6.9 | 10.6 | 14.0 | 16.7 | 19.5 | 22.4 | 26.6 |
| Energy & Power | 2.6 | 4.8 | 7.8 | 10.7 | 13.5 | 16.8 | 20.6 | 33.8 |
| Transportation | 2.0 | 3.9 | 6.7 | 9.8 | 12.9 | 16.6 | 20.5 | 39.5 |
| Telecom & IT | 5.3 | 9.2 | 14.3 | 18.9 | 22.6 | 26.5 | 30.3 | 27.0 |
| Consumer & Residential | 6.2 | 10.5 | 16.1 | 21.2 | 25.4 | 30.0 | 34.5 | 26.9 |
| Industrial & Commercial Buildings | 3.3 | 6.0 | 9.6 | 13.2 | 16.5 | 20.5 | 24.9 | 33.0 |
| Manufacturing | 2.4 | 4.5 | 7.2 | 10.1 | 12.6 | 15.8 | 19.6 | 34.4 |
| Others | 1.5 | 2.9 | 4.8 | 6.8 | 8.9 | 11.3 | 14.2 | 37.6 |
| Total | 44.0 | 77.7 | 122.7 | 165.3 | 202.8 | 245.2 | 290.0 | 30.1 |

〔그림 3〕 IoT/M2M 버티컬 서비스 시장규모(단위:억 달러)

로 응용이 될 것으로 긍정적인 전망을 하고 있다. 국내 시장만해도 2013년 2조 2800억규모에서 2022년 22조 8200억원 규모로 10배이상 성장 할 잠재성이 있다고 보고 있다. 특히 디바이스시장이 45%, 서비스와 응용앱 시장이 33%로 다양한 분야의 창조와 융합을 견인하게 될 중요한 시장이 형성 될 것으로 예상이 된다.[4]

[그림 1]에서 예측한 바와 같이 추후 사물통신을 위한 디바이스는 점차 증가하게 되고 지금의 IoT/M2M 산업은 [그림 3][5] 을 통해 알 수 있듯이 국가 주도의 공공 분야 및 통신사 그리고 인력이 많이 드는 분야의 성장이 크게 이뤄지고 있으며 시장 규모 가치가 높은 분야인 Public Safety & Urban Security와 Telecom & IT의 시장 규모가 점점 높아짐을 알 수 있다. 또한 주로 공공 안전 분야와 외진 지방의 보안을 위한 산업군과 다양한 물품 및 배송 분야의 많은 수작업 인력이 필요한 Retail 부분 그리고 통신사들의 전략적 서비스 구축 부분과 전자회사들의 스마트 전략에 기인하고 있음을 나타낸다.[5]

사물인터넷은 [그림 4]와 같은 서비스 개념도를 갖으며 사물인터넷이 현재의 이동통신 음성시장의 포화상태




〔그림 4〕 M2M 활용한 지능형 서비스 개념도

를 해결할 수 있는 중요한 융합서비스로 자리 잡을 것이며, 스마트 홈, 스마트 그리드, 헬스케어, 지능형 교통 서비스 등을 중심으로 서비스가 다각화될 것으로 전망되며 우리는 점점 더 편리하고 다양한 향유를 누리게된다.[6] Machine To Machine의 기술에 이어서 사물인터넷의 시대로 도래함에 따라 다양한 보안 위협이 발생되고 이러한 사물인터넷 시대를 열기 위해 다양한 기술들이 발전하고 있다. 앞으로 빛을 보게될 사물인터넷과 만물인터넷은 새로운 기술과 이전에 개발된 기술들의 혼합된 기술들로 그 시대를 열어갈 것이다.

본 논문에서는 MQTT 기술을 이용한 사물인터넷 기술에 대해서 소개하고 MQTT 기술에 적용할 수 있는 보안기술에 대해 기술한다.

## Ⅱ. IoT 주요 핵심 기술

### 2.1. IoT의 원리

#### 2.1.1. 통신원리[7]

사물인터넷(IoT)은 임베디드형 단말기에서 인터넷을 활용하는 모든 기술을 포괄하는 의미이다. 사물인터넷(IoT)이 이슈가 된 것은 최근의 일이지만 다른 관점에서 보면 완전히 새로운 기술은 아니라고 할 수 있다. 모든 기술이 그러하듯 사물인터넷(IoT)의 기반이 되는 이전 기술이 있는데 그것이 바로 기계 간의 통신을 의미하는 사물통신(Machine to Machine, M2M)이다. 현재 출시되고 있는 퓨어 밴드, 스마트 워치, 구글 글래스와 같은 사물인터넷(IoT) 제품도 사물통신(M2M) 방식이라 볼 수 있다. 사물통신(M2M)과 사물인터넷(IoT)는


〔그림 5〕 Internet Of Things(사물인터넷) 통신과정

통신 방식으로 둘을 구분할 수 있다. 사물과 사물간의 1:1 방식인 사물통신(M2M)과 달리 사물인터넷(IoT)은 인터넷을 기반으로 사람과 사물, 사물과 사물간의 정보를 상호 소통하는 방식이다.

우리가 알고 있는 통신은 개인이 정보의 수신자 또는 송신자가 되어 데이터를 주고 받는 것을 의미한다. 이처럼 데이터를 주고 받는 통신은 사실 세 가지 단계를 통해 이루어진다. 송신자와 수신자는 각자 통신을

위한 단말기가 필요하고 이러한 단말기를 통해 송신자와 수신자는 서로를 인식하는 접속 과정을 거치고, 이후 서로의 존재를 확인하고 사용자를 직접 연결하는 인증 과정이 필요하다. 우리가 익명의 상태로 접속한다고 인지하더라도 실제로는 내부적인 인증 단계를 다 거치고 있는 것이다. 이를 통해 우리가 흔히 통신이라고 인지하는 데이터 전송을 할 수 있는 것이다. [그림 5]는 이 과정을 기본적인 그림으로 나타내고 있다. 이처럼 세 단계를 거치는 통신은 그 매개체가 PC, 스마트폰, 태블릿 PC와 같은 스마트 디바이스든, 스마트 디바이스의 인터넷과 연결하여 통신하는 단말기든, 사물인터넷(IoT)이든 동일하게 진행된다. 이런 기본 통신을 가능하게 해주는 핵심기술 중 MQTT에 대해 2.2절에서부터 알아보도록 한다.

## 2.2. MQTT

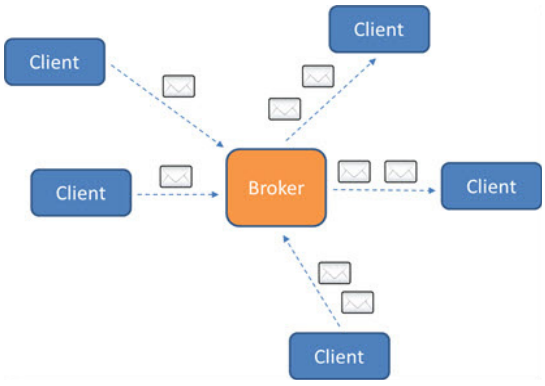
### 2.2.1. 정의

MQTT(Message Queue Telemetry Transport)는 메시지 큐 원격 측정 전송의 약자이며 제한된 자원 (CPU, RAM, 배터리 등)을 가진 임베디드 디바이스를 제한된 네트워크에서 비동기 통신을 가능하게 해주는 경량 메시징 프로토콜이다.[8] 이름에서 알 수 있듯이, 센서 데이터와 같은 원격 데이터 전송에 적합하지만 스마트폰의 페이스북 메신저와 같은 어플리케이션에서 클라이언트간 메시지 교환을 위해 MQTT 프로토콜을 사용한다. MQTT는 IBM 및 EUROTECH 의해 개발 되었으며 개방형 표준 프로토콜로 배포되었으며 2013년 3월 26 일에 개최 된 첫 번째 기술위원회 회의에서 OASIS에 의해 표준화 되었다. OASIS는 "구조적 정보 표준의 발전을위한 조직"의 줄임말이며 개발, 통합 및 채택을 구동하는 글로벌 컨소시엄 개방형 표준이다.

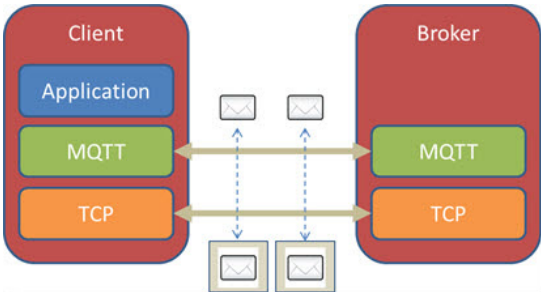
### 2.2.2. Architecture : Client and Broker

MQTT 아키텍처는 [그림 6]과 같이 구성되어 있다. 일반적으로 MQTT 프로토콜을 기반으로 네트워크는 많은 클라이언트(최대 10K 디바이스)와 브리지에 연결된 브로커를 추가하여 네트워크 크기를 증가시킬 수 있는 브로커라는 서버로 구성된다.[9]

각 클라이언트는 브로커에게 자신의 고유 식별자인 클라이언트 ID를 제공하기 위해 연결되며 브로커는 클라이언트의 연결을 관리하고, 그들 사이에 메시지를 전송한다. 또한, 브로커는 전형적으로 데이터베이스에 저장하여 모든 메시지의 영속성을 관리하는 책임이 있다. 그래서 브로커는 일시적으로 연결이 끊어진 모든 클라이언트에게 온라인 반환을 통해서 메시지 전송이 가능하다. 이러한 특징을 "메시지를 유지“ 라고하고 연결이 끊어지기 쉬운 신뢰할 수 없는 네트워크에 중요한 특징이다. 클라이언트가 어떠한 메시지도 보낼 필요가 없거나 오랜 시간동안 메시지를 수신하지 않을 때, 클라이언트는 주기적으로 "keep-alive" 메시지를 브로커에게 보




〔그림 6〕 MQTT 아키텍처, 클라이언트와 브로커

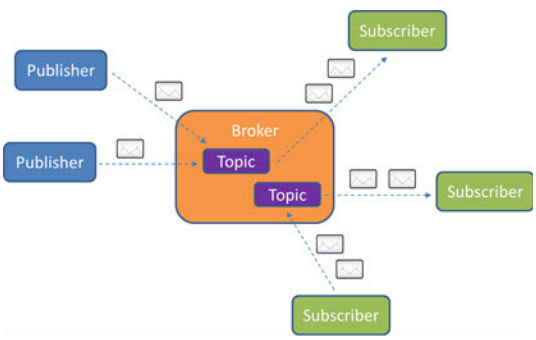



〔그림 7〕 TCP/IP 기반 MQTT. TCP 패킷에 싸여진 메시지

내서 지속적인 연결을 유지해야 한다. 그렇지 않으면 브로커는 서버에 연결할 때 클라이언트에 의해 지정되는 연결 유지 시간이 지나면 연결을 끊는다. 전체적인 아키텍처는 TCP/IP를 기반으로 하며 그렇기 때문에 클라이언트 간 각 메시지 교환은 [그림 7]과 같이 TCP 패킷 안에 싸여서 진행된다.

### 2.2.3. Message : Publish, Subscribe and Topic

MQTT 프로토콜은 [그림 8]과 같은 구조를 가지는 "Publish/Subscribe" 패턴과 "Topic"개념을 기반으로 한다.[10] 클라이언트는 발행자, 구독자 또는 둘 모두가 될 수 있다. 발행자인 경우, topic에 메시지를 발행하고, 구독자인 경우, 특정 topic에 등록되어 topic에 게시된 모든 메시지를 수신할 수 있다. topic은 클라이언트가 메시지를 교환하는 메커니즘이며, 기술적으로는 메시지 큐이고 브로커는 모든 등록을 큐로부터 관리해야한다. 그리고 관리는 클라이언트의 누군가 그 topic을 등록한 확실한 topic의 발행 메시지로 부터 메시지를 릴레이 해야한다. 클라이언트가 발행 된 메시지를 수신하는데 관심 있는 항목을 관찰할 수 있기 때문에 이 패턴은 "감시




〔그림 8〕 TCP/IP 기반 MQTT. TCP 패킷에 싸여진 메시지

자"로 알려져 있다. 이 모델은 일대일과 일대다 분배를 허용 한다. 그리고 게시자는 가입자에 대해 아무것도 알 필요가 없다. 이러한 모든 특징은 클라이언트 비동조화와 비동기 통신을 보장한다.

구독은 영속성 또는 비영속성이 될 수 있다. 영속성인 경우 브로커가 연결되어 있다면 즉시 등록하고 있는 클라이언트의 메시지를 릴레이하고 그렇지 않으면 다음 번 구독자 연결이 될 때까지 메시지를 저장한다. 비영속성의 경우 브로커는 오직 가입자가 연결되어 있을 때 메시지를 릴레이하며 그렇지 않은 경우에는 메시지를 잃는다.

이러한 선택은 클라이언트가 접속된 메시지에 "clean sessiong flag"를 사용하여 브로커에 연결할 때 이루어진다. 플래그가 true로 설정되어있는 경우 클라이언트와 연결이 되어있지 않을 때 모든 클라이언트의 등록은 제거된다(비영속성 구독). 플래그가 false로 설정되어있는 경우 클라이언트와 연결이 끊어 경우에도 모든 등록은 활성 상태로 유지된다(영구 구독). 더욱이 발행자는 “retained”의 메시지를 표시할 수 있다. 그래서 브로커는 새로운 구독자에게 마지막으로 성공한 메시지를 보낼 수 있다. 그렇기 때문에 브로커와의 연결 후 발행자의 첫 번째 메시지인 발행 메시지를 받기위해 기다릴 필요가 없다. 마지막으로, 클라이언트가 서버로 접속할 때 예기치 않게 연결이 끊어지는 경우에 브로커가 특정 topic을 발행해야하는 “will” 메시지를 구체화할 수 있다.

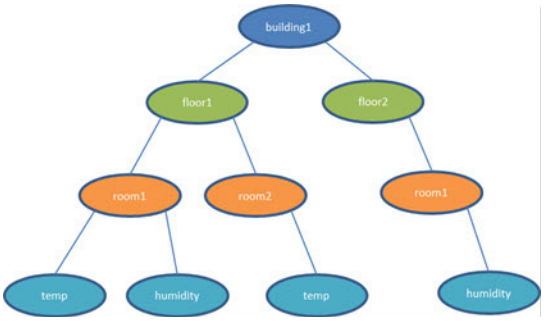
### 2.2.4. Topic Format

Topic은 문자열로 표현하고 하위 항목의 각 계층은 ‘/’문자로 구분되는 계층 구조를 가지고 있다.[11] 예를 들어, 회사가 각 층과 방을 위한 온도, 습도, 등의 많은 센서를 가진 건물 내부의 물리적 데이터를 모니터를 원한다고 가정하였을 때 센서 데이터를 발행하고 수신하기 위한 항목 구조는 다음과 같다.

- building1/floor1/room1/temperature
- building1/floor1/room1/humidity
- building2/floor3/room4/temperature

계층 구조는 [그림 9]와 같이 건물로 시작하며 특정 방과 바닥을 따라서 특정 센서로 끝난다.

발행자는 절대 topic으로만 발행할 수 있지만 구독자




〔그림 9〕 Topic의 계층구조

는 절대 topic을 구독하거나 또는 그 이상의 topic에 발행된 메시지를 수신할 수 있도록 와일드카드를 사용할 수 있다. 와일드카드는 다음의 두 가지 유형이 있다.

- 단일 레벨 와일드카드 "+"는 계층 구조의 특정 레벨에 대한 하위 항목을 나타낸다.
- 멀티 레벨 와일드카드 "#"은 문자열의 끝 부분만이 표시되고, 이 계층의 하나 이상의 레벨을 나타낸다.

위의 예를 고려하는 경우에 building1안 floor1에 모든 객실의 모든 온도 값에 관심을 가질 수 있다. 이 경우 다음과 같은 방법으로 주제 계층 구조 내부의 객실 수준을 축소하기 위해 와일드카드 "+"를 사용할 수 있다

- building1/floor1/+/temperature

같은 방법으로 building2내 floor2에 room3의 온도, 습도 등 모든 데이터 센서에 관심이 있다면 우리는 다음과 같은 방법으로 주제 계층 구조의 마지막 수준을 축소하려면 와일드카드 "#"을 사용할 수 있다.

- building2/floor2/room3/#

### 2.2.5. Quality of Service

TCP/IP 기반의 MQTT는 데이터 전달을 보장하지만 TCP 연결이 끊어졌을 경우에는 메시지를 잃을 수 있다. 모든 통신 프로토콜은 Quality of Service를 보장하고 MQTT는 메시지 전달을 TCP의 상위 3계층을 정의한다.[12] Level0에서 Level2까지 서버는 등록자에게 큰 대역폭 소비와 대기시간이 존재하지만 높은 QoS를 전달해주는 메시지를 보장하기 위해 노력을 기울인다. 서비스 레벨 품질은 다음과 같다.

- QoS Level0(거의 한 번) : 이 경우 MQTT 어떤 기능도 TCP에 추가하지 않는다. 그래서 이 레벨은 TCP의 “best-effort”와 같다. 메시지는 목적지에 도달하거나 도달하지 못한다.
- QoS Level1(적어도 한 번) : 이 수준은 메시지가 등록자에게 도착하는 것을 보장하지만 중복이 있을 수 있다. 클라이언트는 한 번 또는 그 이상 메시지를 받을 수 있다.
- QoS Level2(정확히 한 번) : 이 경우 등록자에게 정확히 한 번 전달된다. 이 경우에 최대 오버헤드가 발생하고 브로커는 메시지를 로컬로 저장한다.

### 2.2.6. 인증 및 보안

MQTT 프로토콜은 클라이언트 인증에 대해 두 가지 즉, 클라이언트 ID와 사용자의 심플한 메커니즘을 제공한다.[13] 클라이언트가 브로커에 접속했을 때 유일한 식별자를 서버에게 제공한다. 이 식별자는 연결된 모든 클라이언트에서 고유해야 하고 존재하는 식별자이면 브로커는 연결을 거절한다. 게다가 연결 메시지는 클라이언트의 사용자임을 식별할 수 있는 사용자 이름과 패스워드를 포함해야 한다. 보안에 있어 MQTT 프로토콜은 자체 보안 레벨을 구체화하지 않는다. TCP 기반이기 때문에 SSL/TLS을 이용하여 데이터를 암호화할 수 있다. 이러한 방법으로 클라이언트 ID와 사용자 이름, 패스워드 보다 더 나은 방법인 클라이언트와 서버의 인증서를 통해 인증할 수 있다. MQTT는 페이로드에 얽매이지 않기 때문에 SSL/TLS 사용하지 않고 응용계층에서 메시지 페이로드를 암호화하는 보안계층을 추가할 수 있다

### 2.2.7. Message Flow

MQTT 프로토콜의 더 나은 이해를 위해서는 클라이언트와 서버 사이의 메시지 흐름을 연구하는 것이 유용할 수 있다.[14]다음은 흐름의 단계를 나타낸 것이다.

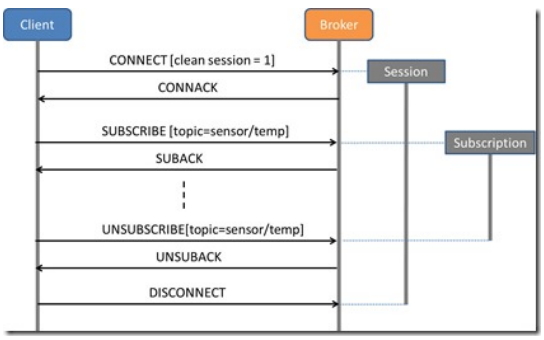
- 브로커에 클라이언트를 연결

- Topic에 클라이언트 등록과 등록으로부터 클라이언트는 메시지 수신.
- 클라이언트가 모든 가능성이 있는 QoS 레벨을 위한 topic에 메시지를 발행하는 방법
- 클라이언트와 서버간의 연결 유지
- 연결의 종료
- 예기치 않은 클라이언트의 연결이 끊길 경우 will 메시지의 수신

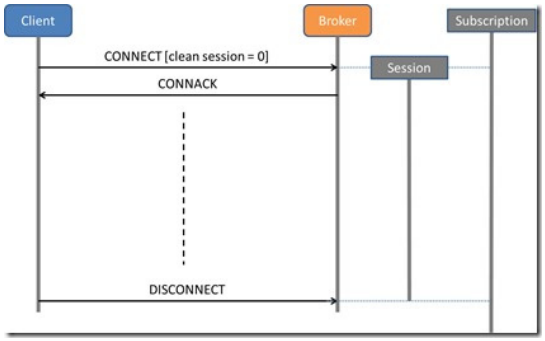
클라이언트는 CONNECT 메시지를 브로커에게 보내 연결을 하고 브로커는 CONNACK 메시지로 응답을 한다. 클라이언트가 연결되었을 때 세션이 시작되고 메시지를 받기 위한 하나 또는 그 이상의 topic을 구독할 수 있다. CONNECT 메시지는 클라이언트가 설정할 수 있거나, 브로커로부터 연결이 끊어졌을 때 모든 구독은 잃게 되는 “clean session” 플래그를 갖는다. 만약 “clean session”이 true라면 구독은 오직 세션 안에서만 유효하며 만약 클라이언트가 브로커로부터 “clean session” 없이 연결이 끊긴다면 모든 topic으로부터 구독이 해제된다. 브로커는 clean session을 통해 세션이 종료될 것이고 클라이언트는 다음 연결 시에 이 topic에 대한 새로운 세션을 맺어야 한다.

만약 “clean session”이 false인 경우 브로커는 이전 세션의 모든 topics에 대해 자동으로 등록할 수 있도록 클라이언트를 세션을 유지한다. [그림 11]

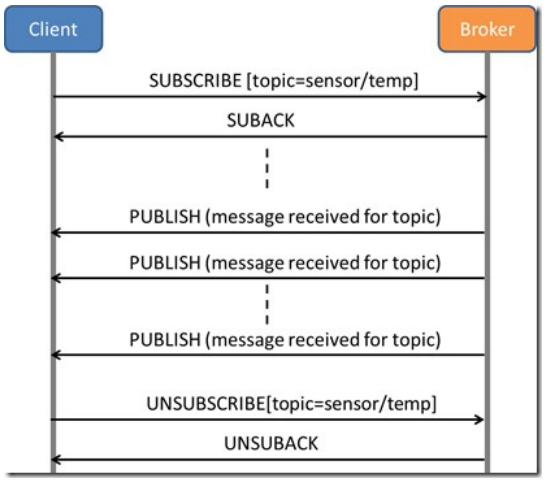
클라이언트 등록은 매우 간단하다. 클라이언트가 등록을 원한다는 topic을 가진 SUBSCRIBE 메시지를 보내고 브로커로부터 SUBACK 메시지를 수신한다. 이 순간부터 클라이언트는 topic에 대해 발행한 모든 메시지를 클라이언트가 UNSUBSCRIBE 메시지를 브로커에게 보내고 UNSUBACK 메시지를 받을 때까지 수신




〔그림 10〕 “clean session” 설정과 CONNECT 메시지




〔그림 11〕 “clean session” 미설정과 CONNECT 메시지

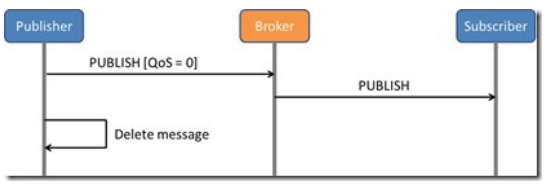



〔그림 12〕 주제에 관한 SUBSCRIBE, UNSUBSCRIBE 메시지

할 수 있다. [그림 12] 참조

클라이언트는 메시지를 3개의 다른 QoS 레벨 중 하나를 사용하여 발행할 수 있다. 레벨0(많아야 한 번)의 경우 메시지는 전송되고 그 메시지가 브로커에게 도달했다면 브로커는 구독자에게 이 메시지를 전달할 것이다. 클라이언트는 브로커로부터 acknowledge를 기다리지 않고 즉시 메시지를 지운다. 이것은 메시지가 목적지에 도달한 것을 보장하지 않는다. [그림 13] 참조

QoS 레벨1(적어도 한 번)을 사용하면 클라이언트는



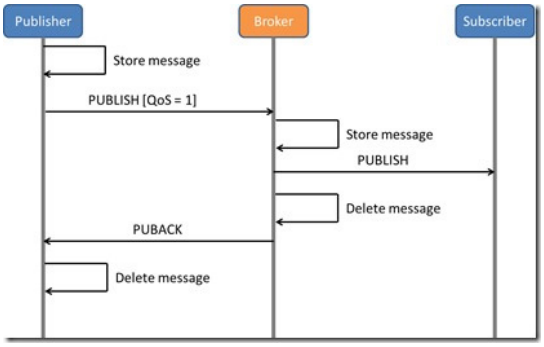

〔그림 13〕 QoS Level0와 PUBLISH 메시지

구독자에게 메시지를 보낸 브로커로부터 acknowledge를 받을 때까지 메시지를 로컬로 저장한다. 만약 클라이언트가 acknowledge 메시지를 받지 못했을 경우 클라이언트는 메시지를 재전송한다. 메시지는 목적지에 도달할 수 있기 때문에 “at least once”라고 한다. [그림 14] 참조

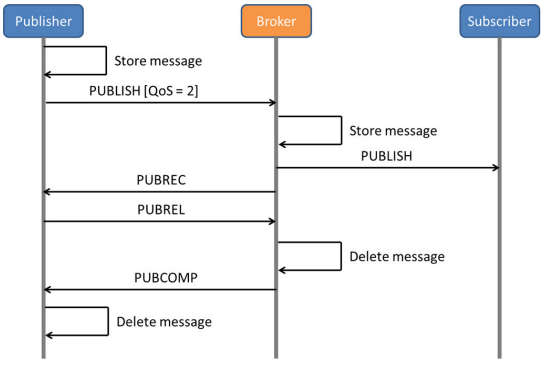
마지막 가능성은 QoS 레벨2(정확히 한 번)이다. 이 경우 메시지의 손실과 목적지를 향한 메시지의 중복을 피하기 위해 발행자와 브로커 사이에 4-Way Handshake를 통해 많은 트래픽이 발생한다. [그림 15] 참조

클라이언트가 네트워크 상에 어떤 메시지도 보내지 않을 때 브로커가 스스로 연결을 끊는 것을 피하기 위해 ping 메시지를 사용하여 연결을 유지한다.[그림 16] 참조

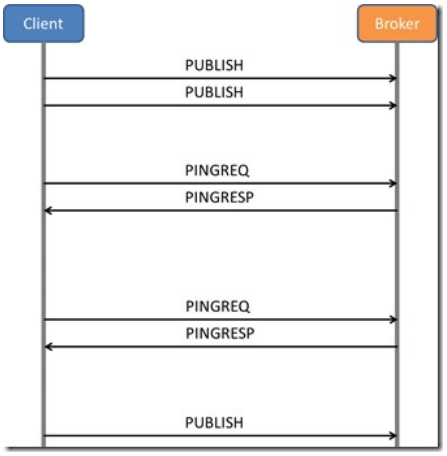
MQTT 프로토콜 구조의 단순한 메시지는 연결을 종료하고 싶을 때 클라이언트가 브로커에게 보내는 연결해제 메시지이다. 이 경우 브로커는 클라이언트에게 응답 메시지를 보내지 않는다. 만약 메시지를 받지 못한다면 시간초과로 인해 연결을 종료할 것이다.[그림 17] 참조



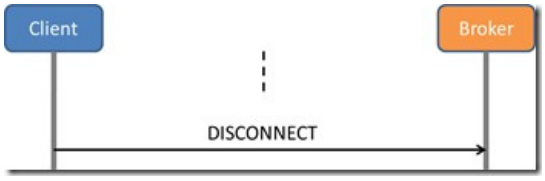

〔그림 14〕 QoS Level1와 PUBLISH 메시지




〔그림 15〕 QoS Level2와 PUBLISH 메시지

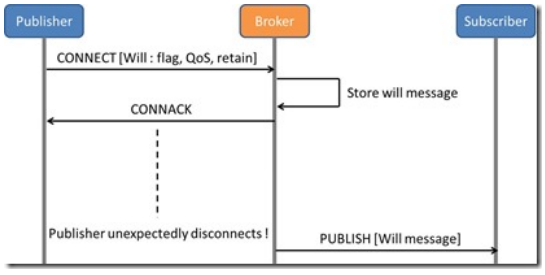



〔그림 16〕 지속적 연결을 위한 ping 메시지




〔그림 17〕 클라이언트의 연결종료

MQTT 프로토콜 구조에서 제공하는 흥미로운 특징은 바로 “will” 메시지이다. 클라이언트가 브로커에 접속했을 때 클라이언트는 “will” 메시지를 보낼 수 있고 연결 메시지의 페이로드로 연관된 topic이다. 브로커는 발행자의 연결해제 메시지 전송 없이 예상치 못한 연결해제가 탐지되었을 때 이 메시지를 모든 관심 구독자에게 보낼 것이다.[그림 18] 참조




〔그림 18〕 “Will” 메시지 특징

## Ⅲ. 보안 기술

### 3.1. MQTT의 보안기술

#### 3.1.1. 보안의 근본 개념

MQTT의 보안의 근본이 되는 3가지 개념이 있는데 다른 프로토콜이 기술과 같이 근본이 되는 개념은 사용자 또는 사물이 권한을 부여받고 권한이 주어지는 클라이언트 이름으로 본인임을 식별할 수 있는 식별(ID)과 패스워드를 이용해 클라이언트의 식별을 입증하는 인증(Authentication), 그리고 인증을 통한 클라이언트에게 주어진 권한부여 및 관리를 하는 권한부여(Authorization)가 있다.[15] 아래에 식별, 인증 권한부여를 이용한 MQTT의 보안기술에 대해 설명할 것이다.

### 3.2. Identification

MQTT 클라이언트는 사용자 ID 또는 공공 디지털 증명서로 식별한다. MQTT서버는 SSL 프로토콜을 통해 클라이언트에게 증명서를 보내거나 클라이언트가 식별자와 패스워드를 설정함으로써 인증한다. 서버는 클라이언트 식별자에 기초한 클라이언트가 접근할 수 있는 자원들을 제어할 수 있다. MQTT 서버는 스스로 IP 주소와 디지털 증명서로 클라이언트를 식별할 수 있다. MQTT 클라이언트는 서버에게 증명서를 입증받기 위해 SSL 프로토콜을 통해 전송한다. 어떤 경우에 클라이언트는 서버를 입증하기 위해 서버의 DNS 이름을 이용하는데 DNS 이름으로 보내진 증명서는 증명서 보유자에 등록되어있다. 다음과 같은 방법 중의 하나로 클라이언트 식별을 설정한다.

#### 3.2.1. Client Identifier

MQTT 클라이언트 클래스는(MQTT 클라이언트 _create 또는 MQTT 비동기_create in C) 클라이언트 식별자를 설정한다. 클라이언트 식별자를 파라미터로 설정하기 위한 클래스 생성자를 호출하거나 클라이언트 식별자를 랜덤하게 생성하여 부여한다. 서버에 접속하는 모든 클라이언트 사이에서 클라이언트 식별자는 유일해야 하고 서버의 큐 매니저 이름과 달라야 한다. 비록 식별 확인에 사용하지 않더라도 모든 클라이언트는 클라이언트 식별자를 가져야한다.

#### 3.2.2. User ID

MQTT 클라이언트 클래스는(MQTT클 라이언트 _create 또는 MQTT비동기_create in C) MQTTConnectOptions (MQTTC 클라이언트 _ConnectOptions in C)의 속성으로 클라이언트 ID를 설정한다. 사용자 ID는 클라이언트에 유일 할 필요는 없다.

#### 3.2.3. Client Digital Certificate

클라이언트 디지털 증명서는 클라이언트 키 스토어에 저장되어있다. 키 저장소 위치는 클라이언트에 의존한다. 다음은 언어별 설정법이다.

JAVA
- MqttConnectOptions의 메소드인 setSSLProperties의 호출에 의해 클라이언트 키 저장소 등록과 위치를 설정하고 키 저장소 등록을 통과한다.

C
- MQTTClient_create 또는 MQTTAsync_create는 MQTTClient_SSLOptions의 ssl_opts 속성으로 키 저장소 등록을 설정한다. openSSL 툴은 C의 MQTT 클라이언트에 접근되는 키와 키 저장소를 생성하고 관리한다.

Android
- 세팅으로부터 안드로이드 디바이스 키 저장소 관리 > 보안 메뉴. SD 카드로부터 새로운 증명서 로드

서버 키 저장소에 저장된 서버의 사설 키 의한 서버의 식별 설정 : WebShpere MQ

#### 3.2.4. WebShpere MQ

MQTT 서버 키 저장소는 클라이언트가 연결된 원격 측정 채널의 속성이다. WebSphere MQ Explorer 또는 DEFINE CHANNEL를 통한 키 저장소 위치와 속성 설정. 다중 채널은 키 저장소를 공유할 수 있다.

### 3.3. Authentication

MQTT 클라이언트는 서버에 접속함으로써 MQTT 서버를 인증할 수 있고 서버는 클라이언트를 연결함으로써 인증할 수 있다. 클라이언트는 SSL 프로토콜로 서버를 인증한다. MQTT 서버는 SSL 프로토콜 또는 패스워드 또는 둘 다를 통해 클라이언트를 인증한다. 만약 서버가 클라이언트를 인증하지 않은 상태에서 클라이언트가 서버를 인증한다면 클라이언트는 종종 익명의 클라이언트로 알려져 있다. 이 SSL을 통해 익명의 클라이언트 연결을 설정한 다음 SSL 세션에 의해 암호화 된 패스워드를 사용하여 클라이언트를 인증하는 것이 일반적이다. 증명서의 분배와 관리의 문제로 클라이언트 증명서보다 클라이언트 패스워드로 인증하는 것이 좀 더 일반적이다. 스마트 전기계량기와 같은 커스텀 디바이스 그리고 ATM과 Chip & Pin Machines과 같은 고부가가치 디바이스들에 사용되는 클라이언트 증명서를 찾으려 할 것이다.

#### 3.3.1. Server Authentication by a client

MQTT 클라이언트는 SSL 프로토콜을 통해 서버 증명서 인증하여 올바른 서버에 접속했는지 확인한다. HTTPS 프로토콜을 통해 웹 사이트를 검색할 때처럼 이러한 형태의 입증은 친숙하다. 서버는 인증기관에서 서명한 서버 공인인증서를 클라이언트에게 보낸다. 클라이언트는 서버 인증서에서 인증기관의 서명을 입증하기 위해 인증기관의 공개키를 사용하고 인증서가 통용되는지를 확인한다. 이러한 확인은 인증서의 유효성을 설정한다. 종종 root 인증서로 불리는 인증기관의 인증서는 클라이언트 truststore에 저장한다. 다음은 언어별 설정법이다.

JAVA
- MqttConnectOptions의 setSSLProperties 메서드 호출과 위치를 설정하기 위한 truststore 속성, 클라이언트 truststore의 속성의 전달

C
- MQTTClient_create 또는 MQTTAsync_create는 MQTTClient_SSLOptions의 ssl_opt 속성으로 truststore 속성 설정한다. openSSL 툴을 통한 truststores와 인증서 관리

Android
- 세팅을 통한 안드로이드 디바이스 truststore 관리 > 보안 메뉴. SD 카드로부터 root 인증서 로드

#### 3.3.2. Client authentication by a server

MQTT 서버는 SSL 프로토콜을 통한 클라이언트 인증서 인증 또는 클라이언트 ID와 패스워드 인증을 통해 서버에 연결된 올바른 클라이언트 연결을 입증한다. 클라이언트는 MQTT 서버의 패스워드를 이용한 클라이언트 인증을 MQTT 프로토콜 헤더 서버로 보낸다. 서버는 Client 식별자 또는 패스워드 인증으로 인증을 선택할 수 있다. 보통 서버는 사용자 ID를 인증한다. SSL 연결을 통한 패스워드를 확인은 패스워드가 평문으로 보내지는 것을 막기 위해 서버의 확인으로 보안되어 왔다.

#### 3.3.3. WebSphere MQ

WebSphere MQ는 SSL 프로토콜로 클라이언트 인증서를 인증한다. WebSphere MQ 원격 측정 키 저장소에 root 인증서를 저장. 상호 SSL인증의 일부로 클라이언트 인증서를 인증할 수 있다. 즉 클라이언트에게 서버의 공인인증서 제공해야 할뿐만 아니라 서버에게 클라이언트의 공인인증서를 제공해야한다.

- WebSphere MQ 원격 측정은 자신의 개인, 공인인증서에 동일한 저장소를 사용한다. 그리고 root 인증서와 같은 다른 공인인증서는 인증기 관에 제공한다.
- WebSphere MQ Explorer 또는 DEFINE CHANNEL 명령으로 키 저장소 위치와 속성 설정, 다중 채널은 키 저장소를 공유할 수 있다.
- WebSphere MQ는 Java 인증 권한부여 서비스(JAAS)를 호출함으로써 클라이언트 사용자 ID 인증한다.
- MQXRConfig 구성 stanza에서 JAAS 구성은 jaas.config 파일에 저장된다. 파일은 WebSphere MQ 데이터 경로의 qmqrs\QmgrName\mqxr 디렉토리 내에 저장된다.
- JAASLoginModule의 로그인 메소드를 작성하여 클

라이언트의 인증을 확인한다.

- WebShpere MQ 원격 측정은 JAASLoginModule을 통한다. 로그인 메소드는 다음 파라미터를 사용한다.
  · User ID
  · Password
  · Client Identifier
  · Network Identifier
  · Channel Name
  · ValidPrompts

### 3.4. Authorization

권한부여는 MQTT 프로토콜의 한 부분이 아니며 MQTT 서버에 의해 제공된다. 서버가 무엇을 하는지에 따라 어떤 권한이 부여되는지가 달라진다.

MQTT 서버는 publish/subscribe brokers이며 클라이언트가 서버에 접속할 수 있고 클라이언트가 발행하거나 구독할 수 있는 주제인 유용한 MQTT 권한 규칙 제어를 한다. 만약 MQTT 클라이언트가 서버를 관리할 수 있다면 클라이언트가 서버를 다른 양상으로 관리할 수 있는 더 많은 권한 규칙 제어를 한다.

가능한 클라이언트의 수는 거대하나 개별적으로 각 클라이언트에게 권한을 부여하기 위한 가능성이 아니다. MQTT 서버는 프로필 또는 그룹에 의한 클라이언트를 그룹화하기 위한 수단을 가질 것이다.

접근과 권한부여의 관점에서의 클라이언트의 ID는 MQTT 클라이언트에 고유한 것이 아니다. 클라이언트 식별자와 클라이언트 ID를 동일시하지 않는다. 둘은 같은 것처럼 보이나 일반적으로 다르다. 예를들어 공통적인 서비스의 수에 걸친 사용자 이름을 가지고 있다면 이러한 서비스 중 일부는 Single Sign-On과 협조한다. MQTT 서버의 기업적인 규모는 다른 어플리케이션에 공통적인 ID와 권한을 제공하는 권한부여 서비스를 호출하는 것과 같다.

#### 3.4.1. WebSphere MQ

WebSphere MQ는 접속 가능한 권한부여 서비스를 가진다. 기본 권한부여 서비스는 Windows와 Linux 객체 권한 매니저로 제공된다.(OAM) OAM은 topic과 큐와 같은 OS 사용자 ID와 WebSphere MQ 객체에 작동하는 그룹에 연관되어 있다.

고정된 사용자 ID로 WebSphere MQ에 접근할 수 있는 원격측정 채널을 구성할 수 있다. 이것은 샘플 채널을 설정하는 방법이다. 또는 MQTT 클라이언트로 사용자 ID를 설정하는 WebSphere MQ에 접근할 수 있다. WebSphere MQ 객체에 접근할 수 있는 MQTT 클라이언트의 권한부여는 낮은, 중간, 세분화된 클라이언트 액세스제어를 달성하기 위해 WebSphere MQ 원격 측정을 설정한다.

## Ⅴ. 결 론

사물인터넷 기술 이전에 M2M 기술이 이전에 기대되는 기술로 꼽혔던 기술 중 하나이다. 그러나 기술 발전에 의해 M2M 통신에서 사물인터넷으로 빠른 발전을 보였다. 사물인터넷으로 통합되므로 인해 새로운 기술들이 접목되고 각 기술들에 의해 다양한 기기가 서로 연결되어동작하는 M2M 및 IoT 시대가 도래하고 MQTT와 같은 표준화된 프로토콜을 사용할 수 있게 됨에 따라, 사물 간 통신이 이루어질 수 있게 되었다. 사물인터넷 시대를 위해 MQTT 뿐만 아니라 XMPP, CoAP 신 기술의 등장으로 서비스는, 점차 폭넓고 깊게 확장될 것이다. 앞서 살펴본 MQTT 보안과 같이 변하는 시대의 흐름에 맞춰 개별의 사물들이 융합되어 통신이 이뤄짐에 따라 개인정보 및 하나의 종단시스템으로써 일어날 수 있는 보안 사고를 대비하기 위해 다각도로 연구해야할 것이다.

## 참 고 문 헌

[1] 주대영, 김종기 “초연결시대 사물인터넷(IoT)의 창조적 융합 활성화 방안” 산업연구원 January. 2014

[2] GSMA London Office, “Experience a world where everything intelligently connects : The Connected Life”, February 2012.

[3] J.Bradley, J.Barbier, D.Handler “Embracing the Internet of Everything To Capture Your Share of $14.4 Trillion”, February 2013.

[4] Tuesday, September 10, 2013 “세상의 모든것들이 연결되는 날 Internet of Things all connected” http://huchoi.blogspot.kr/2013/09/internet-of-thin

gs-all-connected.html
[5] January 2014. 선웅균 “IoT/M2M 관련 산업별 시장동향” http://mktmaster.wordpress.com/2014/01/07/iotm2m-%EA%B4%80%EB%A0%A8-%EC%82%B0%EC%97%85%EB%B3%84-%EC%8B%9C%EC%9E%A5%EB%8F%99%ED%96%A5/ 선웅균 회계사의 Social Media
[6] January 2014. “국내외 사물인터넷 정책 및 시장 동향과 주요 서비스 사례” http://fishpoint.tistory.com/1140
[7] May 2014. “사물인터넷 활용과 보안-사물인터넷 활용을 위한 우리의 보안 수준은” http://blog.lgcns.com/494
[8] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/217/3-1-MQTT-Introduction.aspx
[9] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/219/3-2-Architecture-Clients-and-Broker.aspx
[10] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/220/3-3-Message-Publish-Subscribe-and-Topic.aspx
[11] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/221/3-4-Topic-Format.aspx
[12] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/222/3-5-Quality-of-Service.aspx
[13] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/223/3-6-Authentication-and-Security.aspx
[14] http://www.embedded101.com/DevelopM2MIoTDevicesEbook/DevelopM2MIoTDevicesContent/tabid/155/ArticleId/224/3-7-Message-Flow.aspx
[15]https://www.ibm.com/developerworks/community/blogs/c565c720-fe84-4f63-873f-607d87787327/entry/mqtt_security?lang=en

## 〈저 자 소 개〉

**심 승 현 (Seung-Hyun Shim)**
학생회원

2011년 2월 : 한남대학교 정보통신공학과 졸업
2014년 3월~현재 : 동국대학교 정보보호학과 석사과정
관심분야 : 디지털 포렌식, 리버스 엔지니어링

**김 학 범 (Hak-Beom KIM)**
정회원

1990년 8월 : 중앙대학교 대학원 전자계산학과 졸업(공학석사)
2001년 2월 : 아주대학교 대학원 컴퓨터공학과 졸업(공학박사)
1991년 10월~1996년 6월 : 한국전산원 주임연구원
1996년 7월~2001년 8월 : 한국정보보호진흥원(KISA) 기술표준팀장
2001년 9월~2003년 1월 : (주)드림시큐리티 상무이사
2003년 2월~2005년 3월 : (주)장미디어인터렉티브 상무이사
2008년 4월~2009년 6월 : 인포섹 (주) 수석컨설턴트
2009년 7월~2010년 12월 : 에스지에이(주) 연구소장
2011년 9월~2013년 3월 : (주)지엔에스인증원 ISMS본부장
2001년 3월~2009년 2월 : 순천향대학교 정보보호학과 겸임교수
2005년 9월~현재 : 동국대학교 국제정보대학원 겸임교수
2011년 7월~현재 : 한국정보보호학회 이사
2013년 4월~현재 : ㈜이너버스 연구소장
관심분야 : 통합로그 시스템, 빅데이터 보안, 클라우드 컴퓨팅 보안, 개인정보보호, PIMS