https://doi.org/10.7840/kics.2017.42.1.257

포그 컴퓨팅 환경에서의 보안 및 프라이버시 이슈에 대한 연구

남 현 재•, 최 호 열*, 신 형 준*, 권 현 수**, 정 종 민*, 한 창 희**, 허 준 범°

Security and Privacy Issues of Fog Computing

Hyun-Jae Nam•, Ho-Yeol Choi*, Hyung-June Shin*, Hyun-Soo Kwon**,
Jong-Min Jeong*, Chang-Hee Hahn**, Jun-Beom Hur°

요 약

IoT(사물인터넷) 기술이 발전하여 적용 분야가 다양해지고 이에 따라 서비스를 이용하는 사용자 수도 크게 증가하였다. 수많은 IoT 디바이스들에 의해 발생되는 실시간 대용량 데이터를 클라우드 컴퓨팅 환경에서 처리하는 것은 더 이상 적합하지 않다. 이러한 문제를 해결하기 위해서 응답시간을 최소화 하고 실시간 처리가 적합하도록 하는 포그 컴퓨팅이 제안되었다. 하지만 포그 컴퓨팅이라는 새로운 패러다임에 대한 보안 요구사항이 아직 정립되지 않았다. 이 논문에서는 포그 컴퓨팅에 대한 모델 정의와 정의된 모델에 대한 보안 요구사항을 정리하였다.

Key Words : IoT, cloud computing, fog computing, security requirement, privacy

ABSTRACT

With the development of IoT (Internet of Things) technology, the application area has been diversified and the number of users using this service also has increased greatly. Real time big data generated by many IoT devices is no longer suitable for processing in a cloud computing environment. To solve this issue, fog computing is suggested which minimizes response time and makes real time processing suitable. However, security requirement for new paradigm called fog computing is not established until now. In this paper, we define models for fog computing, and the security requirements for the defined model.

Ⅰ. 서 론

사물 인터넷(Internet of Things, 약어 IoT)[1,35]는 통신 기능이 탑재된 주변 사물들이 인터넷으로 연결되어 데이터를 수집하고 분석하여 사용자에게 서비스를 제공하는 기술을 의미한다. 현재는 스마트홈, 스마트 그리드 등의 서비스를 제공하고 있으며 기술이 발전함에 따라 좀 더 다양한 형태의 서비스에 대한 수요가 늘어날 것으로 보인다. 이러한 IoT기기들은 각종 센서와 통신 장비를 통해 다양한 종류의 데이터를 대량으로 수집 및 생성한다. 이렇게 수집된 데이터들은 중앙 처리 장치로 보내지고 중앙 처리 장치는 서비스의 형태에 따라 데이터를 분석하고 처리한다.

하지만 중앙집중식 클라우드 컴퓨팅(cloud

※ 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단과 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. 2016R1A2A2A05005402), (No.R0190-16-2011,IoT 소프트웨어 보안 취약점 자동 분석 기술 개발).
• First Author : Korera University Department of Computer Science and Enginerring, niceotor13@korea.ac.kr, 학생회원
° Corresponding Author : Korera University Department of Computer Science and Enginerring, jbhur@korea.ac.kr, 정회원
* 고려대학교 컴퓨터학과 석사과정
** 고려대학교 컴퓨터공학과 박사 과정
논문번호 : KICS2016-11-352, Received November 14, 2016; Revised December 2, 2016; Accepted Jaunary 12, 2017

computing)[2,34] 시스템은 구조적 경직성으로 인하여 지리적 고밀도 분포 특성을 지니는 분산 IoT환경에서의 특징을 효과적으로 활용하기 어려운 문제점이 발생하였다. 그래서 클라우드 서비스를 사용자가 위치한 네트워크 엣지(edge)까지 확장하여 서비스 지연(latency) 최소화, 사용자 상황 인지 (context awareness), 이동성(mobility)등의 기능을 효율적으로 제공할 수 있는 포그 컴퓨팅(Fog computing)[3] 시스템이 새롭게 제안 되었다. 포그 컴퓨팅 시스템의 도입으로 인해 사용자는 실시간 서비스나 대용량의 데이터를 필요로 하는 서비스를 빠른 속도로 제공받을 수 있게 되었지만 포그 컴퓨팅 서비스의 수요 발생의 증가에 비하여 포그 컴퓨팅 운용 환경 및 발생 가능한 보안 취약점에 대한 분석이 미흡한 실정이다. 인터넷의 선 배포 후 보안 취약점 보안의 예에서와 같이 보안에 대한 사전 안전성 검토 및 보안 모델 설계에 따른 시스템 구성으로 안전한 포그 컴퓨팅 환경 설계에 대한 가이드라인이 필요하다. 따라서 차세대 혁신 플랫폼으로 기대되는 포그 컴퓨팅과 기존 클라우드-IoT와의 구조적 차이점으로 발생 가능한 취약점을 분석하여 정리 할 필요가 있다.

하지만 현재까지 명확한 포그 컴퓨팅 환경에서 시스템 모델 정의와 신뢰모델이 정의되지 않았기 때문에 보안상의 이슈를 정리하고 분류하는데 어려움이 있다. 따라서 본 논문 2장에서는 먼저 포그 컴퓨팅 환경에 대한 시스템 모델을 정의하고 신뢰모델, 서비스 제공 모델, 배포 모델을 정의한다. 그 후 3장에서는 발생할 수 있는 보안 이슈들에 대해 항목 별로 분류하여 포그 컴퓨팅 환경에서 만족되어야 하는 보안 요구사항을 정리한다.

Ⅱ. 포그 컴퓨팅

포그 컴퓨팅은 클라우드 컴퓨팅(Cloud Computing)과 지리적 고밀도 분포 특성을 지닌 IoT기기간의 연산, 데이터저장, 네트워크서비스를 보다 효율적으로 제공하기 위해 새롭게 제안된 플랫폼이다. 기존의 클라우드가 부담하고 있었던 일정 부분의 역할을 포그 노드가 담당하게끔 디자인 된 모델로써 이러한 포그 컴퓨팅을 클라우드 컴퓨팅의 확장으로 정의 할 수 있지만, 기존의 클라우드 컴퓨팅과는 분명한 차이점이 존재한다.

1) 네트워크 끝단에 위치(Edge location) : 포그 컴퓨팅은 IoT기기와 지리적으로 가까운 위치에 존재하기 때문에 기존의 클라우드 환경에서는 제공하기 힘들었던 대용량의 데이터를 실시간으로 처리해야하는 서비스(ex 증강현실, 실시간 비디오 스트리밍 등)를 보다 빠르고 효율적으로 제공할 수 있다.

2) 위치 인지(Location awareness) : 포그 컴퓨팅은 IoT기기의 위치정보를 대략적으로 알 수 있기 때문에 위치기반서비스(Location Based Service)에서 중요한 요소인 기기의 이동성 지원하는데 이점을 가진다.

3) 사용자 상태 인지(Context awareness) : 포그 노드는 IoT기기와 통신을 할 때 로컬네트워크 상태, 기기의 상태 정보 등의 상황 정보를 활용 할 수 있으므로 보다 최적화된 서비스를 제공 할 수 있다.

이 절에서는 포그 컴퓨팅 모델의 전체적인 구조와 각 개체의 역할 즉 시스템 모델(System Model), 신뢰 모델(Trust Model), 그리고 서비스 전달 모델(Service Delivery Model)과 배포 모델(Deployment model)을 정의한다.

2.1 시스템 모델

2.1.1 클라우드

클라우드는 다양한 서비스를 제공하기 위해 만들어진 데이터 센터로 유저에게 편의성을 제공하기 위해 대용량의 데이터를 관리, 저장, 연산을 하며 필요한 서비스를 제공한다. 대표적으로 외국에서는 Amazon(AWS), Google(Google Drive), DropBox등이 있으며 국내에서는 네이버(n드라이브), KT(유클라우드) 등의 기업이 클라우드 서비스를 제공하고 있다. 포그 컴퓨팅 환경에서 클라우드는 포그 노드에게 필요한 서비스를 위탁한다. 따라서 실시간 데이터 처리나 대용량 데이터 처리에 대한 서비스를 제외한 전체적인 데이터에 대한 연산이나 분석, 포그 노드의 관리, 기계 학습 등의 많은 양의 연산 필요로 하는 업무를 수행한다.

2.1.2 포그 노드

앞서 언급한 바와 같이 사물인터넷이 빠르게 발전함에 따라 클라우드에게 새로운 특성을 요구하게 되었다. 유선환경 뿐만 아니라 무선 환경에서의 대용량 데이터 처리, 실시간 서비스, 기기의 이동성 등 다양한 요구사항이 생겨나면서 포그 노드라는 새로운 개체가 제안되었다. 이러한 포그 노드는 서비스의 종류에 따라 다양한 형태로 존재 하고 정의 될 수 있다. 예를 들어, 포그 노드의 자원을 많이 필요로 하는 경우에는 단순히 제2의 클라우드(데이터센터)가 포그 노드의 역할을 수행할 수 있으며 많은 양의 연산을 필요로

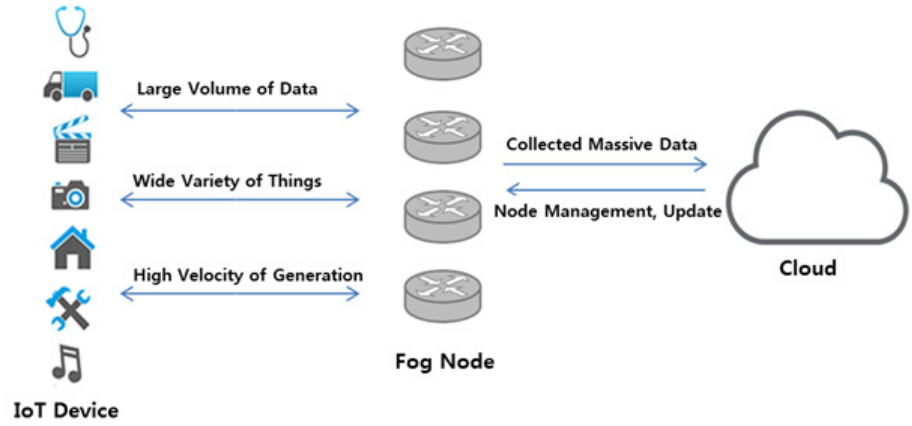


그림 1. 포그 컴퓨팅 시스템 모델
Fig. 1. system model of fog computing

하지 않는 경우에는 WIFI AP나 가정의 셋탑박스 등이 포그 노드가 될 수 있다. 하지만 분명한 점은 IoT 기기와 가까운 위치에서 실시간 서비스, 신속한 의사결정을 위한 재난 방지/대응 시스템 등 다양한 서비스를 효율적으로 제공해야 한다는 것이다.

2.1.3 IoT기기

지리적 고밀도 분포 특성을 가진 IoT기기는 다양한 종류, 거대한 양의 데이터를 빠르게 생성하고 전송한다. 또한 그에 따른 빠른 응답시간을 요구한다. 인터넷으로 연결되어 다양한 형태의 데이터를 수 집, 생성하고 이를 포그 노드에게 전송하거나 클라우드와 포그 노드로부터 필요한 데이터를 제공받아 유저에게 특정 서비스를 수행하는 모든 기기들로 정의한다. 단순히 데이터의 수집을 목적으로 하는 센서도 IoT기기로 분류할 수 있다.

2.2 신뢰 모델

신뢰 모델은 제공하고자 하는 서비스의 형태, 포그 노드 또는 IoT기기의 특성에 따라 크게 달라질 수 있다. 본 절에선 이러한 다양한 신뢰모델을 수용하고 모든 공격시나리오를 고려한 신뢰모델을 정의한다.

2.2.1 클라우드

클라우드는 기본적으로 honest but curious(정직하지만 호기심이 많은)모델을 따른다. 이는 서비스 제공, 데이터 관리, 연산 등의 업무는 정직하게 수행하지만 필요한 경우 암호화된 데이터를 복호화하려는 시도가 있을 수 있는 Semi-trust(반-신뢰)모델이다. 하지만 서비스 제공자가 제3의 기관에 클라우드 서비스를 위탁했을 시 클라우드를 Untrust(비신뢰)모델로 정의 할 수 있다. 이러한 경우 클라우드에게 데이터에 대한 암호화, 연산, 저장 등에 대한 증명이나 안전한 키 관리를 필수적으로 요구할 수 있어야 한다.

2.2.2 포그 노드

포그 노드는 악의적인 공격자에 의해 통제 될 수 있다. 공격자가 포그 노드와 타협하여 악의적인 행동을 할 경우 데이터의 안전성과 사용자의 프라이버시가 크게 위협이 된다. 포그 노드는 특정 IoT기기를 고립시키거나 IoT기기의 대략적인 위치정보를 활용하여 다양한 공격을 시도할 수 있다. 안전한 인증 기법으로 공격자가 악의적인 포그 노드를 사용 수 없게 하는 방법이 최선이지만 만약 인증에 성공하거나 포그 노드와 공격자가 공모한다 하더라도 합법적인 연산 이외에 다른 악의적인 행동은 할 수 없게 설계 되어야 한다.

2.2.3 IoT기기

IoT기기는 악의적인 공격자에 의해 통제 될 수 있다. 무의미한 데이터를 전송하거나 필요한 서비스를 제공하지 않을 수 있다. 공격자가 IoT기기와 타협하여 이러한 악의적인 행동을 하더라도 이를 탐지 가능해야 하며 권한 밖의 연산을 수행하거나 접근권한이 없는 데이터를 얻는 것이 불가능해야 한다.

2.3 서비스 제공 모델

기존의 클라우드 컴퓨팅의 서비스 제공 모델을 적용 할 수 있으며 크게 3가지로 분류 할 수 있다: Software as a Service(SaaS), Platform as a Serivce(PaaS), Infrastructure as a service(IaaS)[2]. SaaS모델은 서비스 제공자가 제공하고자 하는 어플리

케이션을 클라우드 환경에 업로드하여 유저에게 어플리케이션 서비스를 제공하는 모델이다. PaaS 모델은 서비스 제공자가 프로그래밍언어, 라이브러리, 툴 등을 제공하여 소비자가 원하는 어플리케이션을 만들 수 있는 플랫폼 환경을 서비스하는 모델이다. IaaS모델은 사용자가 원하는 어플리케이션이나 운영체제를 업로드할 때 필요한 처리장치, 저장 공간, 네트워크 등의 기초적인 컴퓨팅 자원을 제공하는 모델이다.

2.4 배포 모델

기존의 클라우드 컴퓨팅의 배포 모델을 적용 할 수 있으며 크게 4가지로 분류 할 수 있다: Private fog, Community fog, Public fog, Hybrid fog[2]. Private fog는 여러 소비자로 구성된 하나의 독립적인 기관에서 독점적인 사용을 위해 제공된다. Community fog는 공통된 관심사를 공유하는 특정 공동체의 독점적인 사용을 위해 제공된다. Public fog는 일반 대중들을 대상으로 공개된 사용을 위해 제공된다. Hybrid fog는 2개 또는 그 이상의 구별되는 클라우드가 데이터와 어플리케이션의 이동성을 가능하게 하는 기술로 같이 묶인다.

Ⅲ. 포그 컴퓨팅 보안 및 프라이버시 이슈

본 절에서는 앞서 정의한 시스템모델과 신뢰모델을 기반으로 전반적인 포그 컴퓨팅 환경에서 고려되어야 하는 보안상의 이슈에 관하여 항목별로 소개한다. [그림 2]는 포그 컴퓨팅 환경에서의 보안 및 프라이버시 이슈를 트리 형태로 나타낸 것이다.

3.1 인증

인증이란 어떤 개체가 실제로 이전에 미리 확인된 개체인지 판단하는 절차이다. 이는 대부분의 시스템에서 선행되는 가장 중요한 요소이다. 포그 컴퓨팅 환경에서 클라우드와 포그 노드 간의 인증 절차 또는 포그 노드에 연결된 다양한 IoT 장치에 대한 인증 절차가 이루어지지 않을 경우 위조 노드(fake node)와 같은 보안 위협에 취약해질 수 있다. 위조 노드의 경우 공격자가 포그 컴퓨팅 시스템 내부에 가짜 노드를 심어 두고 위조된 코드나 데이터를 시스템 내부에 전송하는 방법이다. 변조된 데이터는 유휴 상태의 IoT 장치를 의도적으로 작동시켜 전력을 소모시킨다. 전력을 모두 소모한 IoT 장치는 시스템에서 제외되고 이러한 공격의 연쇄 작용은 결국 전체 시스템의 성능 저하를 일으킨다. 시스템에서 인증 절차를 거친다 하더라도 공격자는 다양한 방법을 통하여 인증 절차를 회피할 수 있다. 다음의 두 가지 공격은 인증절차를 피하기 위해 공격자가 취할 수 있는 공격 방법이다.

3.1.1 재전송 공격(Replay Attack)

인증 받으려는 개체가 인증기관에 보내는 인증정보를 공격자가 가로챈 후 재전송하여 인증 받는 방법의 공격이다. 이 공격에 대한 대응방안으로는 세션 토큰(session token)을 사용하는 것이다. 인증기관이 인증하려고 하는 개체에게 세션 토큰을 보내고 인증 개체는 세션 토큰의 해시값을 인증정보에 덧붙여 보낸다.

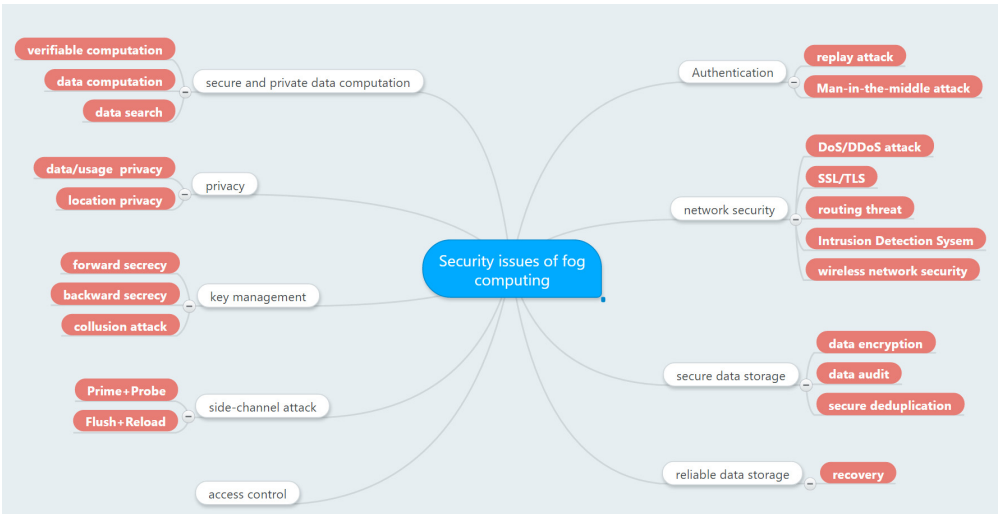


그림 2. 포그 컴퓨팅 보안 및 프라이버시 이슈
Fig. 2. security and privacy issues of fog computing

공격자가 인증정보를 가로채어 다른 세션에서 재전송 공격을 시도한다고 할 때, 인증기관은 새로운 세션에 대한 세션 토큰을 발급했기 때문에 인증이 승인되지 않는다.

3.1.2 중간자 공격(Man-in-the-middle-attack, MITM)

인증기관과 개체 간의 인증 절차 사이에서 데이터를 위변조 하여 인증 절차를 회피하는 방법이다. 두 개체는 서로 인증되어 연결되었다고 생각하지만 실제로는 중간자에 의해 연결되어 있고 두 개체간의 데이터는 모두 중간자를 통하여 전송된다. 이를 감지하고 방어하기 위한 다양한 방법들이 제안되었다[4-6]. 하지만 이러한 방법도 중간자 공격을 완전히 방지 할 수는 없기 때문에 IoT 장치에 대한 인증 절차에서 중간자 공격에 대한 감지와 방어가 가능한 방어 체계를 구축해야 한다.

포그 컴퓨팅 환경에서는 말단에 수많은 IoT 장치들이 연결되어 있다. 기존의 PKI 기반 인증 시스템은 확장성과 효율성이 떨어져서 포그 컴퓨팅 환경에 적용하기 힘들다. ad-hoc 무선 네트워크에서 사용될 수 있는 효율적인 인증 알고리즘이 제안되었지만 포그 컴퓨팅 환경에서는 무선 네트워크 환경 외에도 다양한 네트워크 환경을 고려해야 하는 만큼 각 환경에 맞는 경량화된 인증 알고리즘이 필요하다[7].

3.2 네트워크 보안

클라우드-포그 노드간의 네트워크 통신, 포그 노드-IoT간의 네트워크 통신에서 발생 할 수 있는 보안 이슈들에 대해 다룬다. 공격자는 연결된 네트워크에 대한 공격을 통해 여러 가지 형태의 공격을 할 수 있으며 이러한 다양한 공격들에 대해 방어하기 위한 기법들을 효율적으로 설계해야 한다.

3.2.1 DoS/DDoS 공격

Denial of Service(DoS)/Distributed Denial of Serivce(DDoS)공격은 공격자가 서버가 처리할 수 있는 능력 이상의 것을 요구하여 다른 서비스를 정지시키거나 시스템을 다운시키는 공격이다. 네트워크 기능을 마비시키는 공격으로써 빠른 응답/처리를 요구하는 포그 환경에서는 네트워크 속도가 느려지는 것만으로도 심각한 문제가 초래 될 수 있다. 따라서 네트워크 환경을 구축할 때 DoS/DDoS 공격에 대한 방어책을 반드시 고려되어야 한다.

3.2.2 SSL/TLS

Secure Socket Layer(SSL)/Transport Layer Security(TLS)[8]는 가장 많이 사용되고 안전하다고 알려진 표준화된 네트워크 보안 프로토콜이다. 기본적으로 일대일 통신에서 상호간의 인증, 데이터기밀성과 무결성을 제공한다. 하지만 높은 수준의 안전성을 제공하는 만큼 프로토콜이 무겁다는 단점을 지닌다. 클라우드-포그 노드 간의 통신에서는 활용될 수 있지만 무선 환경, 연산 능력이 낮은 포그 노드-IoT기기 간의 통신에서는 부담이 될 수 있다. 따라서 이러한 환경에서 쓰일 수 있는 경량화 된 네트워크 보안 프로토콜을 제시해야 한다. 또한 현재까지도 여러 가지 취약점이 발견되고 패치가 이루어지고 있으므로 만약 SSL/TLS를 제공하고 있는 서비스 제공자라면 해당 네트워크 관리자는 주기적으로 최신버전으로 업데이트하여 보안 패치를 진행해야 한다.

3.2.3 라우팅 위협(Routing Threat)

안전한 네트워크를 구성하려면 라우팅 공격에 대해 안전해야 한다. 공격자는 라우팅 정보를 변조, 간섭을 통하여 라우팅 루프를 생성하거나 네트워크 전송을 방해 할 수 있다. 또한 에러 메시지를 생성할 수 있고 임의로 지연시간을 발생 시킬 수 있다. 안전한 라우팅 프로토콜을 사용하여 이러한 공격에 대해 방어할 수 있어야 한다.

3.2.4 침입 탐지 시스템(Intrusion Detection System)

침임 탐지 시스템은 전통적인 방화벽이 탐지할 수 없는 모든 종류의 악의적인 네트워크 트래픽과 행동, 정책위반등을 감시한다[9]. 포그 컴퓨팅 환경에서 침입 탐지 시스템은 포그 노드를 감시하며 악의적인 행동을 감지 할 수 있다. 하지만 이러한 환경에서는 한 개의 클라우드가 관리하는 포그 노드 수 또는 한 개의 포그 노드가 관리하는 IoT기기의 수가 상당히 많을 수 있다. 또 IoT기기는 수시로 움직이기도 한다. 이는 분명 침입 탐지 시스템을 설계할 때 추가적으로 고려해야 하는 부분이다. 침임 탐지 시스템은 이러한 포그 컴퓨팅 환경에서 효율적인 방향으로 구현되어야 한다.

3.2.5 무선 네트워크 보안

포그 노드와 IoT기기간의 네트워크 환경은 무선 네트워크 환경일 가능성이 높다. 무선이라는 특수한 환경 때문에 여러 가지 보안상의 위협이 존재한다. 재밍 공격(Jamming Attack)은 공격자가 무선 네트워크 환경에 라디오주파수를 전송하여 원래의 메시지를 훼손

하거나, 수신자에게 메시지가 도달하지 못하게 방해하여 네트워크 성능을 저하시키는 공격이다. 스니퍼 공격(Sniffer Attack)은 공개된 채널을 사용하는 무선 환경에서 더욱 효과적인 공격이 될 수 있다. IoT기기와 포그 노드는 무선 네트워크라는 도메인에서 발생할 수 있는 취약점에 대해 안전하게 설계 되어야 한다.

3.3 안전한 데이터 저장

3.3.1 데이터 암호화

사용자가 데이터 기밀성을 요구한다면 반드시 해당 데이터는 암호화 되어야 한다. 기밀성을 만족하기 위해서는 기존에 존재하는 여러 가지 암호화 기법을 사용하되 반드시 안전하다고 증명된 암호화 기법을 사용해야 한다. 특별히 포그 환경에서 고려되어야 할 점은 암호화의 효율성이다. IoT기기는 상대적으로 연산량이 적기 때문에 경량화된 암호화 기법[10]을 사용해야 하며 저장 공간도 적기 때문에 키의 크기를 최소화해야 한다. 즉, 암호화에 대한 효율성을 극대화하되 반드시 기밀성을 보장할 수 있어야 한다.

3.3.2 데이터 감사

사용자 또는 IoT기기는 클라우드와 포그 노드에게 데이터와 데이터에 대한 통제를 위탁한다. 그렇기 때문에 요청한 일을 제대로 처리하였는지에 대한 증명이 필요하다. 클라우드와 포그 노드는 데이터에 대한 소유권(PoW)[11], 데이터에 대한 사용가능성(PoR)[12,13], 데이터의 저장(PoS)[14], 데이터의 삭제(PoD)[15]등에 대한 증명을 사용자에게 줄 수 있어야 한다. 기존의 클라우드 환경에서는 이미 해결 된 부분이지만 악의적인 포그 노드가 존재하는 환경에서 어떻게 데이터를 감사할 것인지에 대한 해결책이 필요하다.

3.3.3 안전한 중복제거

중복제거 기술은 클라우드가 저장 공간의 효율을 높이기 위해 서로 다른 유저에게 동일한 데이터를 받을 경우 중복된 데이터를 감지하여 제거하는 기술이다. 수렴 암호화 기법(Convergent Encryption)을 통한 안전한 중복제거[16], 키 서버를 통한 안전한 중복제거[17,18], PAKE 프로토콜(Password Authenticated Key Exchange Protocol)을 활용한 기법[19]등 여러 가지 기법이 제안되었고 안전성의 정도와 효율성에 따라 각기 다른 기법들을 사용 할 수 있다. IoT기기와 포그 노드간의 통신, 포그 노드와 클라우드간의 통신에서 어떤 기법을 사용하며 어느 정도의 안전성을 제공할 것인지 대해서 충분하게 고려되고 설계 되어야 한다.

3.4 신뢰성 있는 데이터 저장

클라우드나 포그 노드는 신뢰성 있는 데이터 저장을 제공하기 위해 시스템 오류로 인해 발생 할 수 있는 데이터의 손실, 손상에 대처하기 위한 기법들을 필요로 한다. 기본적으로 데이터를 중복으로 저장하여 예기치 못한 오류로 인해 손상된 데이터를 복구하는 방법이 사용되지만 이는 분명 효율성 측면에선 좋은 방법은 아니다. 클라우드에 비해 저장 용량이 작은 포그 노드나 IoT기기에서 어떠한 방식으로 신뢰할 수 있는 데이터 저장을 효율적으로 제공할 것인지 고려해야 한다.

3.5 안전한 데이터 연산

3.5.1 검증 가능한 연산

신뢰할 수 없는 기관에 어떠한 연산을 맡기는 것은 위험하다. 신뢰 할 수 없는 클라우드, 포그 노드에게 어떠한 연산을 위탁하는 경우 연산을 올바르게 수행했는지 검증할 필요가 있다. 검증 가능한 연산[20] 기법을 통해 이러한 문제를 해결 할 수 있다. 검증 가능한 연산이란 신뢰할 수 없는 개체에게 어떠한 연산을 위탁할 경우, 생성된 증명을 통하여 계산된 결과 값이 올바른지 판단할 수 있는 기법을 말한다. 신뢰할 수 없는 포그 노드가 존재하는 환경에서 보안 측면으로 볼 때 이러한 기법은 큰 이점을 지닌다. 하지만 포그 컴퓨팅 환경의 특성을 고려해 볼 때 계산자원이 적은 기기에서 활용하기 위해선 이러한 증명을 생성하거나 검증하는 과정에서 발생하는 추가적인 비용이 문제가 될 수 있으므로 가능한 최소화해야 한다.

3.5.2 암호 데이터 연산

클라우드 환경에서 암호데이터 연산은 암호화되어 위탁된 자료에 대해 복호화 과정 없이 특정 연산을 처리 가능하도록 함으로써 암호화 및 복호화에 소용되는 시간을 줄여 연산의 효율성을 향상시킨다. 또한 연산을 위한 데이터 복호화 시 발생할 수 있는 데이터유출 피해를 막을 수 있다. 하지만 포그 환경에서 데이터 아웃소싱에 주체가 되는 IoT기기의 경우 제한된 자원으로 인해 기존의 암호데이터 연산을 가능하게 해주는 동형암호[21], 함수암호[22]의 암/복호 연산조차 어렵기 때문에 단순 적용이 어려운 상황이다. 이를 반영하여 포그 컴퓨팅 환경에 적합한 경량화된 암호데이터 연산 기법이 필요하다.

3.5.3 데이터 검색

신뢰할 수 없는 포그 노드로부터 민감한 데이터를 보호하기위해서는 전송하기 전 반드시 데이터를 암호화해야 한다. 복호화하지 않더라도 암호화 된 데이터에 대한 검색을 필요로 하는 경우 검색 가능 암호 기술(Searchable Encryption)[23]기법이 활용 될 수 있다. 하지만 제안된 많은 기법들에서 각 유저마다 데이터 검색을 위해 필요한 검색 토큰 전송을 필요로 한다. 많은 양의 데이터가 존재하는 경우 이러한 검색 토큰의 크기 및 개수가 문제가 될 수 있다. 또한 검색 가능 암호 기술에서 사용되는 키에 대한 안전한 관리와 크기에 대한 이슈도 고려해야 한다.

3.6 프라이버시

개인의 위치정보나 주소 등과 같은 개인 정보에 대한 보안은 클라우드 컴퓨팅 환경이나 무선 네트워크 환경에서 중요한 문제로 남아있다. IoT 장치에서는 실시간으로 개인의 민감한 정보에 대한 데이터가 수집되고 처리되기 때문에 개인정보보호 문제는 포그 컴퓨팅 환경에도 그대로 적용된다. 다양한 환경에서의 개인정보보호 기술은 지속적으로 발전하고 있다.

3.6.1 데이터 프라이버시

포크 컴퓨팅 환경에서도 기존의 클라우드 환경에서 사용되었던 프라이버시 보호(privacy-preserving)기법들이 적용될 수 있다. 사용자의 개인정보보호를 위해서 선행되는 과정은 데이터의 암호화이다. IoT 장치에서 수집된 데이터들은 암호화되어 포그 노드에서 처리되는데 이때 민감한 데이터에 대해서는 암호화 된 데이터를 복호화 하지 않고 처리하는 기술을 통하여 개인정보를 보호할 수 있다. 위에서 언급한 동형 암호화 기술은 이러한 성질을 만족시킨다[24]. 포그 서비스를 이용하는 사용자의 고유 사용 패턴(usage pattern) 또한 노출 되지 않아야 할 개인정보이다. 예를 들어, 스마트 그리드(smart grid) 서비스를 이용하는 사용자의 스마트 미터(smart meter)를 읽을 수 있는 공격자가 있다고 가정한다면, 이 공격자는 사용자의 전력 소모 패턴을 이용하여 사용자가 언제 외출 하는지 등의 정보를 도출 해낼 수도 있다.

3.6.2 위치 프라이버시

위치정보 프라이버시는 모바일 서비스 환경에서 중요한 보안 이슈이다. 이는 위치정보를 기반으로 포그 서비스를 제공받는 사용자의 위치정보를 노출 시키지 않도록 하는 것이다. 위치정보 프라이버시를 보장하기 위한 기술들은 크게 두 가지로 분류할 수 있다. 첫 번째는 익명성(anonymity)을 이용한 방법이다[25-27]. 이 방법의 경우 장치의 위치정보는 드러나지만 장치를 식별할 수 없도록 하는 기술이다. 장치의 신원정보(identification)을 공격자가 알 수 없기 때문에 사용자의 개인정보는 보호된다. 하지만 특정 위치를 방문하는 사용자가 특정되어 있다면 이러한 익명성을 이용해도 신원정보가 노출 된다는 문제점을 가지고 있다. 두 번째는 더미(Dummy)를 이용한 방법이다[28,29]. 이 방법은 위치정보를 여러 개를 만들어 공격자를 혼란시키는 방법이다. 가령 자신의 위치정보를 포그 시스템에 전송하여 서비스를 제공받는 사용자가 자신의 장치에서 자신의 위치정보와 함께 가짜 위치정보를 여러 개를 만들어서 포그 노드로 전송하게 되면 포그 노드는 가짜 위치정보를 포함하여 모든 위치정보에 대한 서비스를 제공하고 사용자는 그 중 실제 자신의 위치정보에 대해서만 확인하면 된다. 하지만 이 방법은 포그 노드가 가짜 위치정보에 대한 서비스 처리도 해야 한다는 점에서 오버헤드가 크고 위치정보들로부터 실제 위치정보를 추출해내는 추론 공격(inference attack)에 취약하다는 문제점이 있다.

3.7 접근 제어

안전한 포그 컴퓨팅 환경을 설계하기 위해서 접근 제어 기술을 응용할 수 있다. 클라우드로 데이터를 의탁하는 환경에서의 접근 제어는 보통 의탁된 데이터에 대한 암호화를 통하여 구현될 수 있다. 이러한 방식은 크게 대칭 키 기반의 접근 제어와 공개 키 기반의 접근 제어로 분류된다. 대칭 키 기반의 암호화는 처리 시간이 효율적이지만 일반적으로 복잡한 키 관리를 요구한다. 반면에 공개 키 기반의 암호화는 키 관리가 쉽지만 많은 처리 시간을 요구한다[30]. 기존의 대칭키 및 공개키 기반 암호 기법은 암호문과 비밀 키 간 일대일 대응만을 지원하기 때문에 다수의 IoT 기기 혹은 사용자 간 안전한 데이터 공유는 세분화된 접근 제어를 요구하기 때문에 이에 적합하지 않다. 이를 해결하기 위해서 속성 기반 암호 기법과 같은 함수 암호 기법을 이용할 수 있다.

포그 컴퓨팅 환경에서는 하나의 포그 노드가 여러 개의 IoT기기들을 관리하는 그룹 관리자가 될 수 있다. 이와 같은 환경에서 보통 IoT 기기들은 무선 통신을 하기 때문에 무선 환경에서의 그룹 통신 기법을 활용해야 한다. 또한 포그 컴퓨팅 환경에서 클라우드-포그 노드-IoT기기에 걸친 전체적인 접근 제어 프로토콜의 설계와 이와 동시에 IoT 장치와 같은 작은 기기

에서 연산 가능한 저비용 암호 기법의 설계가 필요하다.

3.8 키 관리

신뢰할 수 없는 포그 컴퓨팅 환경에서 IoT기기-포그 노드 간 안전한 그룹 통신 또는 포그 노드-클라우드 간 그룹 통신을 위해서는 안전한 키 관리가 선행되어야 한다. 키 관리란 안전한 통신, 즉 암호화 통신을 위해 필요한 키를 생성, 교환, 저장, 관리 등을 포함한다. 이러한 환경에서 전송되는 데이터의 안전성을 보장하기 위해서는 순방향 비밀성(Forward secrecy)/역방향 비밀성(Backward secrecy)[31]을 만족해야 하며 공모 공격에 대해 안전해야 한다.

3.8.1 순방향 비밀성/역방향 비밀성

IoT기기 간 그룹 통신을 하고 있는 환경에서 새로운 기기가 추가되는 경우, 기존의 사용하고 있던 그룹키를 그대로 공유한다면 추가된 기기는 그룹에 참여하기전 암호화된 데이터에 대해서도 접근 가능하다. 이에 대한 안전성을 보장하는 특성을 순방향 비밀성이라고 한다. 반대로, 그룹에서 IoT기기가 그룹에서 탈퇴하는 경우, 그룹키를 갱신하지 않으면 탈퇴한 IoT기기는 계속해서 데이터에 대한 접근이 가능하다. 이에 대한 안전성을 보장하는 특성을 역방향 비밀성이라고 한다. 포그 컴퓨팅 환경에서 IoT기기들은 이동성을 갖기 때문에 그룹 구성원 변경이 잦다. 이러한 특성을 만족하면서 효율적인 키 관리를 할 수 있는 방안이 필요하다.

3.8.2 공모 공격(Collusion Attack)

그룹 통신 환경 내에서 각 개인의 경우 접근권한이 없지만 서로 공모하여 접근권한을 취득하는 공격을 공모 공격이라 한다. 한 그룹에서 탈퇴한 IoT기기와 새롭게 참여한 IoT기기가 서로의 정보를 공유하게 되면 모든 데이터에 대한 접근권한을 얻어 낼 수 있다. 따라서 이러한 공격에 대해 안전한 키 관리 기법이 설계되어야 한다.

3.9 부채널 공격

클라우드 또는 포그 노드 환경에서는 하나의 운영체제 위에 여러 개의 가상화 객체가 존재하게 된다. 이러한 특수한 환경에서 악의적인 공격자는 공유하고 있는 메모리나 캐시의 정보를 이용하여 같은 운영체제에 있는 다른 유저의 정보를 탈취할 수 있다. 이러한 공격은 부채널 공격의 한 종류이며 대표적인 공격 방법에는 준비 조사 공격(Prime+Probe)[36]방식과 초기화 재 적재 공격(Flush+Reload)[37]방식이 존재한다. 방어 방법으로는 공격을 탐지하여 공유된 부분을 잠시 해제 하는 기법[32]과 메모리 상태를 관리하여 방어하는 기법[33]이 존재한다. 이러한 공유상태가 악의적인 행동인지 아닌지를 판단해야 하며 공격자가 공격을 시도하였을 때 정보가 유출되었는지, 어떠한 정보가 유출되었는지 판단하는 기법이 필요하다.

Ⅳ. 결 론

기존 중앙집중식 클라우드 컴퓨팅 시스템은 구조적 경직성으로 인해 분산 IoT 환경에서 적합하지 않은 문제점을 가지고 있다. 서비스 지연(latency) 최소화, 사용자 상황 인지(context awareness), 이동성(mobility)등의 기능을 제공하기 위해 제안된 포그 컴퓨팅(Fog computing)은 현재까지 시스템모델이나 신뢰모델의 대한 정의가 명확하지 않았다. 이 논문에서는 이러한 모델을 정의하고 발생할 수 있는 보안 이슈에 대해 항목별로 정리 하였다. IoT기기가 실생활에 가까워짐에 따라 보안에 대한 중요성도 함께 대두되고 있다. 또한, IoT 기기는 어플리케이션의 형태에 따라 다양한 서비스 모델을 제공하고 그에 따른 다양한 보안 요구사항을 가진다. 시스템 설계자는 포그 환경에서 새롭게 발생할 수 있는 보안 이슈들과 함께 제공하고자 하는 포그 컴퓨팅 서비스에서 요구되는 보안 요구사항을 고려하여 시스템을 설계해야 한다.

References

[1] Postscapes, *Best Internet of Things Definition* (2015), http://postscapes.com/internet-of-things-definition.

[2] P. Mell and T. Grace, "The NIST Definition of Cloud Computing," *NIST Special Publication,* 800-145, 2011.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," *MCC Wksp. Mob. Cloud Comput.*, pp. 13-16, Helsinki, Finland, Aug. 2012.

[4] I. Dacosta, M. Ahamad, and P. Traynor, "Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties," *ESORICS 2012*, Pisa, Italy, Sept. 2012.

[5] N. Karapanos and S. Capkun, "On the

effective prevention of TLS man-in-the-middle attacks in web applications," *USENIX Security Symp. 2014*, San Diego, CA, Aug. 2014.

[6] P. Hoffman and J. Schlyter, *The dns-based authentication of named entities (DANE) transport layer security (TLS) Protocol: TLSA*, RFC 6698 (2012), https://www.rfc-editor.org/rfc/rfc6698.txt.

[7] D. K. Smetters, D. Balfanz, P. Stewart, and H. C. Wong, "Talking to strangers: authentication in Ad-Hoc wireless networks," *NDSS 2002*, San Diego, CA, Feb. 2002.

[8] T. Dierks, *The transport layer security (TLS) protocol version 1.2*, RFC 5246 (2008), https://www.ietf.org/rfc/rfc5246.txt.

[9] C. Modi, et al., "A survey of intrusion detection techniques in cloud," *J. Netw. and Comput. Appl.*, vol. 36, no. 1, pp. 42-57. Jan. 2013.

[10] S. Mun, M. Kim, and T. Kwon, "Lightweight cryptographic technology trends for IoT communication environment" *J. KICS*, vol. 33, no. 3, pp. 80-86, Mar. 2016.

[11] S. Halevi, et al., "Proofs of ownership in remote storage systems," *18th ACM Conf. Comput. and Commun. Security*, pp. 491-500, Oct. 2011.

[12] F. Armknecht, et al., "Mirror: Enabling proofs of data replication and retrievability in the cloud," *USENIX Security Symp. 2016*, Austin, Texas, Aug. 2016.

[13] M. Etemad and A. Küpçü, "Generic efficient dynamic proofs of retrievability," in *Proc. 2016 ACM on Cloud Computing Security Workshop*, ACM, 2016.

[14] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in *Proc. 2nd ACM Conf. Data and Appl. Security and Privacy,* pp. 1-12, San Antonio, Texas, USA, Feb. 2012.

[15] K. M. Ramokapane, A. Rashid, J. M. Such, "Assured deletion in the cloud: requirements, challenges and future directions," in *Proc. 2016 ACM on Cloud Comput. Security Wksp.*, pp. 97-108, Vienna, Austria, Oct. 2016.

[16] JR. Douceur, et al., "Reclaiming space from duplicate files in a serverless distributed file system," *IEEE Distrib. Comput. Syst.,* pp 617-624, 2002.

[17] S. Keelveedhi, M. Bellare, and T. Ristenpart. "DupLESS: server-aided encryption for deduplicated storage," *USENIX Security 13*, 2013.

[18] P. Puzio, et al., "ClouDedup: secure deduplication with encrypted data for cloud storage," *IEEE CloudCom,* vol. 1, 2013.

[19] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," *1992 IEEE Comput. Soc. Symp. Security and Privacy,* May 1992.

[20] B. Parno, et al., "Pinocchio: Nearly practical verifiable computation", *2013 IEEE Symp. Security and Privacy,* 2013.

[21] C. Gentry, "A fully homomorphic encryption scheme," Ph.D dissertation, Stanford Univ., 2009.

[22] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," *Theory of Cryptography Conf.* vol. 6597, pp. 253-273, 2011.

[23] DX. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. 2000 IEEE Symp. Security and Privacy,* pp. 44-55, 2000.

[24] R. Lu, et al., "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parall. and Distrib. Syst.*, vol. 23, no. 9, pp. 1621-1631, 2012.

[25] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mob. Comput.* vol. 7, no. 1, pp. 1-18, 2008.

[26] X. Liu, et al., "Traffic-aware multiple mix zone placement for protecting location privacy," *IEEE INFOCOM,* pp. 972-980, 2012.

[27] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mob. Syst. Appl. and Serv.,* pp.

31-42, San Francisco, California, May 2003.
[28] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," *IEEE Symp. Security and Privacy*, May 2016.
[29] B. Niu, et al., "Protection of location privacy in continuous LBSs against adversaries with background information," *IEEE ICNC,* pp. 1-6, 2016.
[30] H. Wang, et al., "Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control," *IEEE ICDCS'08,* pp. 1-6, 2008.
[31] B. Jiang and X. Hu, "A survey of group key management," *IEEE Comput. Sci. and Softw. Eng. 2008*, vol. 3, pp. 994-1002, 2008.
[32] S. J. Moon, V. Sekar, and M. K. Reiter, "Nomad: Mitigating arbitrary cloud side channels via provider-assisted migration," in *Proc. 22nd ACM SIGSAC Conf. Comput. and Commun. Security*, pp. 1595-1606, Denver, United States, 2015.
[33] Z. Zhou, M. K. Reiter, and Y. Zhang, "A software approach to defeating side channels in last-level caches", *arXiv preprint arXiv: 1603.05615*, 2016.
[34] Y. Kim and S. Lee, "Analysis and comprehension of cloud computing," *J. KICS*, vol. 32, no. 4, pp. 87-92, Mar. 2015.
[35] D. Kim, S. Yun, and Y. Lee, "Security for IoT service," *J. KICS*, vol. 30, no. 8, pp. 53-59, Jul. 2013.
[36] G. Irazoqui, T. Eisenbarth, and B. Sunar, "S $ A: A shared cache attack that works across cores and defies VM sandboxing--and its application to AES," *2015 IEEE Symp. Security and Privacy*, pp 591-604, May 2015.
[37] Y. Yarom and K. Falkner, "Flush+ reload: a high resolution, low noise, L3 cache side-channel attack," *23rd USENIX Security Symp.*, pp. 719-732, San Diego, CA, Aug. 2014.

남 현 재 (Hyun-Jae Nam)

2016년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2016년 2월~현재 : 고려대학교 컴퓨터학과 석사과정
<관심분야> 네트워크 보안

최 호 열 (Ho-Yeol Choi)

2015년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2016년 2월~현재 : 고려대학교 컴퓨터학과 석사과정
<관심분야> 정보보호

신 형 준 (Hyung-June Shin)

2015년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2015년 2월~현재 : 고려대학교 컴퓨터학과 석사과정
<관심분야> 시스템 보안, 클라우드 보안

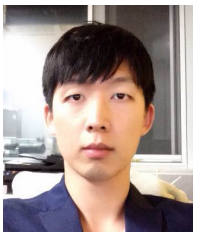
권 현 수 (Hyun-Soo Kwon)

2014년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2016년 8월 : 고려대학교 컴퓨터학과 석사과정 졸업
2016년 8월~현재 : 고려대학교 컴퓨터학과 박사 과정
<관심분야> 클라우드 보안, 네트워크 보안

정 종 민 (Jong-Min Jeong)

2016년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2016년 2월~현재 : 고려대학교 컴퓨터학과 석사과정
<관심분야> 시스템 보안, 네트워크 보안

한 창 희 (Chang-Hee Hahn)



2014년 2월 : 중앙대학교 컴퓨터공학과 학사 졸업
2016년 2월 : 고려대학교 컴퓨터학과 석사 졸업
2016년 2월~현재 : 고려대학교 컴퓨터공학과 박사 과정
<관심분야> 빅데이터 보안

허 준 범 (Jun-Beom Hur)

2001년 2월 : 고려대학교 컴퓨터공학 졸업
2005년 8월 : 한국과학기술원 전산학 석사
2009년 8월 : 한국과학기술원 전산학 박사
2009년 9월~2011년 8월 : University of Illinois at Urbana-Champaign 박사후연구원
2011년 9월~2015년 2월 : 중앙대학교 컴퓨터공학부 조교수
2015년 3월~2016년 8월 : 고려대학교 컴퓨터학과 조교수
2015년 9월~현재 : 고려대학교 컴퓨터학과 부교수
<관심분야> 클라우드 보안, 빅데이터 보안, 네트워크 보안, 응용 암호학