한국시뮬레이션학회 논문지
Vol. 28, No. 4, pp. 67-74 (2019. 12)
http://doi.org/10.9709/JKSS.2019.28.4.067
ISSN 1225-5904

SEED 암호 라이브러리를 활용한 안전한 Android Things 통신 환경연구

박화현 · 윤미경 · 이현주 · 이해영 · 김형종 †

A Study on the Secure Communication at Android Things Environment using the SEED Library

Hwa Hyeon Park · Mi Kyung Yoon · Hyeon Ju Lee · Hae Young Lee · Hyung-Jong Kim†

ABSTRACT

As the market for Internet of Things (IoT) service grows, the security issue of the data from IoT devices becomes more important. In this paper, we implemented a cryptographic library for confidentiality of sensor data from Android Things based IoT services. The library made use of the SEED algorithm for encryption/decryption of data and we verified the library by implementing a service environment. With the library, the data is securely encrypted and stored in the database and the service environment is able to represent the current sensing status with the decrypted sensor data. The contribution of this work is in verifying the usability of SEED based encryption library by implementation in IoT sensor based service environment.

Key words : Internet of Things, Android Things, SEED Algorithm, Sensor Security

요 약

사물인터넷(IoT)의 시장 확대로 IoT 기기가 받아오는 정보에 대한 보안성이 중요해 지고 있다. 본 논문에서는 IoT 센싱 데이터의 비밀성을 보장하기 위한 암호 라이브러리를 구현하고, Android Things 기반 서비스 환경 개발을 통해 이를 검증하였다. 본 연구의 라이브러리는 SEED 암호를 이용하여 데이터에 대한 암복호화 기능을 구현하였고, 센서 정보를 라이브러리에 넣으면 데이터가 데이터베이스에 안전하게 암호화되어 저장될 뿐만 아니라 웹 환경에서도 정상적인 복호화가 되도록 하였다. 본 연구의 기여점은 SEED와 같은 암호기술을 IoT 센서 기반 서비스 환경에서 라이브러리 형태로 구현하여 이의 활용성을 검증하는 데에 있다.

주요어 : 사물인터넷, Android Things, SEED 알고리즘, 센서 보안

1. 서론

IoT 시대에는 인터넷에 연결되는 제품이 기하급수적으로 늘어나므로 인터넷에 연결되는 냉장고, 청소 로봇 등 모든 홈·가전 IoT 제품이 해킹 대상이 될 수 있다. Gartner(2018)의 보고서에 따르면 2018년 기업 IoT 보안 지출 규모가 2017년 12억 달러에서 28% 증가한 15억 달러에 이를 것으로 전망함과[3] 동시에 세계 약 20%의 기업이 3년 이내 최소 한 차례 이상의 IoT 기반 공격을 경험한 것으로 보고했다. 또한, IoT 제품은 일반 ICT 시스템과 달리 보안기술을 적용하기 어려워 상대적으로 보안에 취약하다는 문제점을 다룬 Kim(2013)의 연구도 있다[5]. IoT 센서 데이터는 단일정보로는 프라이버시의 침해가 없을 수 있으나, 이것이 개인식별정보와 연동될 때 문제가 될 수 있다.

본 연구에서는 SEED 암호 알고리즘에 기반하여 센서 데이터의 암호 라이브러리를 개발하고 서비스 환경에 적용해 봤다. 이를 통해 암호기술이 어떤 형태로 IoT 서비스에 활용될 수 있는지를 제시했다. 본 연구는 Android

* 이 연구는 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2017R1D1A1B03034644).

Received: 29 May 2019, **Revised**: 31 December 2019, **Accepted**: 31 December 2019

† **Corresponding Author**: Hyung Jong Kim
E-mail: hkim@swu.ac.kr
Dept. of Information Security, Seoul Women's University

Things 환경에 이를 적용해 본 데에 그 의의가 있다고 할 수 있다. Android Things 환경은 IoT 서비스를 제공할 수 있도록 하기 위한 Android 기반 플랫폼으로 해당 영역에서 암호 라이브러리를 실험해 보고 이의 활용성을 검증하였다.

2. 선행연구

IoT 기술의 사용 비중이 증가함에 따라 편리함이 높아지지만, 그에 대한 위험도도 비례한다. Shin(2019)은 IoT와 같은 초연결적인 네트워크에서는 데이터 위조 및 변조와 클라우드 과정에서 발생할 수 있는 보안 문제가 위협적이라고 경고한다[11]. 사물 인터넷 기술을 도입하여 사용하는 사례 중에 의료 분야가 있다. 의료 영역은 전 세계 병원들을 중심으로 사물인터넷 기술을 도입해 스마트 병원 시스템을 구축하여 스마트폰, 웨어러블 단말 등을 이용해 홈케어, 개인적인 의료서비스를 제공함으로써 의료 품질 향상과 의료비 절감의 효과를 나타내고 있다. 하지만 개인 정보 보호 협회(2013)에 따르면 헬스케어 분야의 정보 유출 사고 시 가장 많은 사고 대응 비용 시 소요된다는 발표와 더불어[12] 개인 의료 정보의 침해 위험성을 높인다는 의견들이 적지 않다. 또한 Woo(2015)의 연구는 정보침해 사례와 위험을 제시하고 있으며 보호 방안으로 개인정보를 수집 후 보관 및 관리 단계에서 DB 암호화, 사용자 인증, 정보보안을 제시하며 정보보안이 전제조건이 되어야 할 것이라고 말한다[13]. 이러한 필요에 따라 사물인터넷을 위한 암호화 라이브러리 연구로, Cho(2018)의 아두이노에 가속도 센서를 연결하여 사용자의 운동 정보를 파악한 후 맞춤형 운동 정보와 상태 모니터링이 가능하도록 한 암호 라이브러리 연구가 존재한다[2]. 또한, 보안성이라는 같은 이유로 Lee(2015)의 연구에서는 IoT의 데이터를 저장하는 과정에서 LEA 알고리즘을 사용하여 데이터의 기밀성을 강화하는 과정을 거쳤다[8]. 라이브러리에 사용되는 암호 기법은 다양하지만, 그중에서 본 프로젝트에서는 SEED 암호화를 사용하였는데 그 이유는 SEED 암호를 사용하여 키가 일치하지 않으면 정상적인 데이터를 볼 수 없다는 점을 이용하였다. 이는 키가 다르면 비정상적인 결과를 출력하여 도어락을 제어하지 못하게 하는 Lee(2017) 연구의 목적과 같다고 할 수 있다[9]. 또한, Kim (2008)에서 공개키 암호가 복호화 과정에서 오랜 시간이 소요된다는 단점을 최소화하기 위해 RFID 보안에 SEED 알고리즘을 적용했듯이 Android Things가 경량화된 운영체제를 사용한다는 것을 고려했을 때 SEED 암호 사용은 적합하다고 할 수 있다[4]. 이러한 기존 연구를 기반으로 볼 때 Android Things 환경에 SEED 암호 라이브러리는 적용은 IoT 서비스의 안정성 확보를 위해 그 의의가 크다고 본다.

3. Android Things 환경에서의 SEED 암호 라이브러리의 구현 및 적용

Android Things는 Google에서 개발한 IoT 운영체제로 사용 중심이던 운영체제에서 장치 개발도 가능하게 만든 플랫폼이다. 이 플랫폼을 기반으로 SEED 기반 암호 라이브러리와 전체적인 기술을 구현하였고 이 기술은 크게 센서, 정책, 데이터베이스, 웹으로 나눌 수 있다.

3.1 SEED 기반 암호 라이브러리의 설계

본 연구의 암호 라이브러리는 Android Things뿐만 아니라 다양한 IoT 분야에서도 활용 가능한 범용성을 갖는다. 개발자는 센서를 기기에 연결하고 라이브러리를 통해 정책을 기반으로 하여 데이터를 쉽게 구분하여 암호화할 수 있고 사용자는 데이터가 분류되고 암호화된 상태를 웹에서 볼 수 있다. 본 라이브러리를 통해 암호에 대해서 비전문가라 하더라도 데이터를 쉽고 안전하게 다룰 수 있을 것이다.

본 논문의 안드로이드 프로젝트는 총 8개의 자바 클래스로 구성되어 있으며 다음 과정은 역할에 따라 총 4가지의 과정(Encryption, PPCL, Sensor, Send)으로 구분한 시스템 구조도로 각 과정 당 1~3개 정도의 클래스로 이루어져 있다. Encryption 과정은 암호화 과정으로 사용자의 키를 대칭 키로 받아 저장하고, 받아온 센서 데이터를 SEED CBC로 암호화하는 역할을 한다. PPCL 과정은 센서 데이터에 정책을 설정하고 그것에 맞게 저장을 한다. Sensor 과정은 보드와 센서를 초기화하고 센서를 작동시키는 일을 한다. Send 과정은 데이터를 직접적으로 사용하기 위한 과정으로 암호화된 정보를 Android Things 화면에 보여주고, 그 데이터를 웹 서비스 환경을 위해 전송하는 역할을 한다. Figure 1은 각 요소의 상호작용을 보여주고 있다.

Figure 2는 데이터 흐름도이며 여기서 주로 전송되는 데이터는 센서 데이터로, 시스템 구조도에서 설명했듯이 Sensor 과정에서 설정한 센서 정보는 PPCL 과정의 정책에 따라 저장된다. 이후 Encryption 과정을 통해서 암호화된 센서 데이터는 데이터베이스에 저장이 되고, Send 과정을 통해 Android Things와 웹 서비스 환경으로 보내

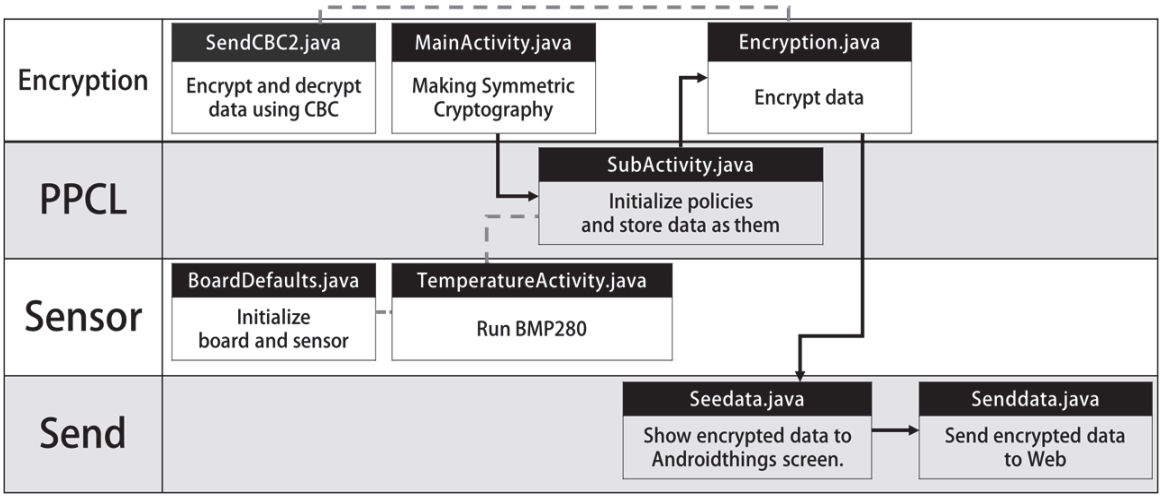

Fig. 1. Work Flow of IoT Service Environment

진다. 암복호화를 진행할 때는 대칭 키 방식 SEED CBC를 사용하였다. Encryption 과정에서 사용자가 설정한 키는 PPCL 과정에서 정책에 의해 선택된 센서 데이터의 암복호화 키로 사용된다. Park(2016)에서 IoT 기반 환경에서 데이터의 기밀성, 무결성을 지킬 수 있던 것 같은 기능을 Android Things 환경에서 설계하였다.

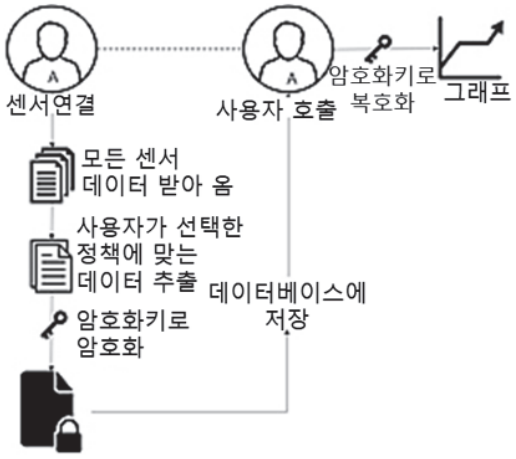

Fig. 2. Data flow

3.2 센서의 연결

센서는 안드로이드 스튜디오에서 드라이브를 제공하고 서로 통신이 겹치지 않는 HC-SR501 센서와 BMP 280을 선택하였다.

HC-SR501 센서는 PIR 센서로 GPIO 통신을 사용한 센서로 설치된 공간에서 움직임이 발견되면 HIGH(1), LOW(0)으로 알려주는 센서이다. 이 센서에는 움직임을 탐지하는 주기가 있는데 그걸 delay time이라 한다. 구현할 때는 사용자의 주기에 맞게 정보가 와야 하므로 센서의 delay time은 최솟값으로 설정하였다.

그리고 BMP 280 센서는 온도, 압력 값을 I2C(Inter-Intergrated Circuit) 통신으로 전달하는 센서로 구현할 때에는 온도, 기압 기능 중에서 온도 부분만 사용하였다. Android Things가 아닌 다른 운영체제를 사용할 때는 라즈베리파이에서 I2C 통신을 사용하기 위해서는 따로 I2C 통신 활성화 설정을 해주어야 하지만 Android Things에서는 모든 통신이 기본적으로 활성화가 되어있어 라이브러리 임포트를 제외하고는 따로 설정해주는 단계는 필요없다. BMP280 센서 역시 사용자가 센서와 주기를 선택하면 해당 주기마다 값을 받아오도록 만들었다.

개발기기는 공식적으로 Google에서 지정한 Android Things를 운영체제로 사용할 수 있는 기기인 NXP I.MX7D와 Raspberry Pi 두 가지 중에서 개발하기에 접근성이 높았던 Raspberry Pi를 사용하였다.

먼저 센서를 사용하기 위해서는 안드로이드 프로젝트에서 개발 장치와 센서를 연결한 핀 번호를 설정해야 한다. 개발 프로젝트에서는 BoardDefaults라는 자바 파일을 별도로 만들어 설정해주었다. Figure 3처럼 각 센서는 해당 통신 핀에 맞춰서 연결하여 HC-SR501 센서는 GPIO 통신 핀인 BCM 21에, BMP280 센서는 I2C 통신 핀인 BCM 2(I2C1), BCM 3(I2C2) 핀에 연결하였다.

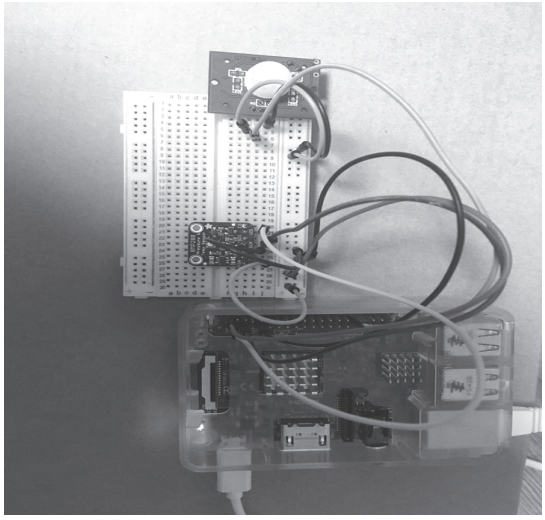

Fig 3. Raspberry Pi connected HC-SR501 and BMP 280

3.3 정책

정책은 센서 연결을 한 이후 사용자가 필요로 하는 센서 정보만 수집할 수 있도록 만든 개념으로 환경, 위치, 주기, 주체 총 4종류로 구성되어 있다. 환경은 측정되고 있는 시간을 뜻하고, 위치는 어디서 측정되고 있는지를 말하며, 주기는 센서값을 받아올 기간에 대한 것이며, 주체는 측정되는 데이터의 대상을 의미한다. 그리고 이 모든 정책들은 Cho(2017)의 센서에 따라 IoT 데이터를 다르게 처리하는 것처럼 사용되는 상황이나 센서 종류에 따라 변경이 가능하다.

기본적으로 모든 센서에 들어가는 정책은 환경과 위치이며 주기나 주체의 경우는 사용자가 선택하거나 입력하는 정책이다. 정책은 각 센서에 맞게 구성이 되지만 모든 센서에 기본적으로 들어가는 환경, 위치, 주기 정보 정책은 Default 클래스에서 정책의 구조체 형태로 정보를 제공해준다. 그 이외 주체나 센서값은 센서의 클래스를 따로 만들어 맞춤 정보를 저장하도록 하였다. 본 논문 구현 시에서 사용한 BMP280 센서를 예로 들어 설명하자면, Android Things에서 센서를 선택하는 액티비티가 시작되는 순간부터 Default 클래스가 작동한다. 여기서 온도 센서를 선택하면 Temperature sensor 클래스가 불리면서 센서의 주체를 입력하도록 하고, 정확한 값을 받아 올 수 있도록 값에 맞는 데이터 타입이 설정 된 정책을 준비해준다. 설정이 다 끝나면 선택한 주기대로 Default 클래스의 정책들과 Temperature 클래스 안의 주체와 센서 값을 암호화 단계로 정보를 보내준다.

3.4 Android Things에서의 데이터베이스 연결

데이터베이스는 MySQL을 사용하였다. Android studio의 특성상 MySQL과 바로 연동되지 않기에 데이터 전송 시 웹을 거쳐 데이터베이스로 전송한다. MySQL을 웹상에서 관리하고 쉽게 보기 위하여 phpMyAdmin을 사용하였다.

많은 양의 데이터를 다루지 않고 원활한 복호화를 위해서는 대칭 키를 사용한다는 점을 고려하여 KISA의 SEEDCBC 128비트를 사용하였다. 데이터 암호화의 세부 과정은 KISA의 소스 코드 개발 매뉴얼을 따라 다음과 같이 진행하였다. SubActivity.java 파일에서 받아온 센서 데이터를 Encryption.java 파일에서 암호화 한다. Encryption.java 파일 속 seedcbc2.java와 seedcbc.c, seedcbc.h, Utils.java 파일을 이용하여 암호화를 진행한다. 암호화의 키는 MainActivity.java에서 사용자에게 입력받은 키를 사용한다. 데이터 암호화 과정을 좀 더 세부적으로 설명하면, Encryption.java의 onReceive() 함수를 이용하여 SubActivity. java에서 데이터를 받아온 후 Encryption.java의 encrypt()함수를 사용하여 데이터타입을 byte[](byte array) 형으로 바꾸어준다. 이 후 byte[]형으로 변환된 데이터를 SEncryption()과 키 값으로 암호화한 후 데이터 타입을 다시 원래의 형태인 String형으로 변환시켜 데이터베이스로 보내준다. 여기서 사용하는 SEncryption()은 객체를 생성하여 진행하는 함수이며, Seedcbc2.java의 seed.init()함수의 인자로 데이터와 키, 초기화 벡터값을 주어 진행한다.

Android Studio에서 데이터베이스 연결은 AsyncTask를 이용하여 진행한다. AsyncTask는 UI Thread를 쉽게 사용할 수 있도록 하는 클래스로 이 클래스를 사용하면 Thread 또는 Handler를 조작하지 않고도 UI Thread에서 Background 작업을 수행하고 결과를 게시할 수 있다. AsyncTask 클래스 속 doInBackground() 안에서 link 변수에 IP주소와 PHP 파일명을 넣고, 환경, 주체, 값이라는 세 가지 필드들과 그 안에 각각 암호화된 데이터가 담긴 PHP 파일을 Autoset의 public_html 폴더에 넣어준다. 이 후 Android Studio에서 전송되는 과정이 실행되며 설정한 IP주소에 맞는 PHP파일을 찾아서 데이터베이스와 연동이 된다.

3.5 웹 서비스 환경

웹 서비스 환경은 주로 암호화된 데이터를 복호화하여 데이터의 경향을 보여주기 위한 사용자 인터페이스 환경이다. 웹 환경을 동작시키기 위하여 PHP, Java script와

HTML을 활용하여 구현하였다. 웹 환경에서 복호화를 진행할 때도 암호화와 같이 SEED CBC_PHP를 사용하였으며 KISA에서 배포한 암호 알고리즘을 사용하였다.

웹 환경에서의 복호화를 위해서는 3가지 값이 필요하다. 첫 번째 복호화할 데이터이다. SEED CBC.php는 2바이트씩 콤마로 나누어서 복호화되도록 설정되어 있어 이 과정을 거쳤다. 두 번째 값은 키 값 이다. 키 값을 서버에서 POST 형식으로 받아 키 값이 일치하는가 하지 않는가를 확인하기 위해 복호화 키 값을 넣어준다. 만약 사용자가 설정하였던 키 값과 같은 값이 입력되면 정상 그래프를 출력하게 되고 아니면 그래프가 나올 수 없는 성질을 갖고 있다. 세 번째 값은 IV(초기화 벡터) 값으로 IV 값은 미리 자동으로 설정되어 들어가게 설정되었다.

Figure 4는 복호화가 숫자, 영어, 특수문자가 정상적으로 나오는가에 대한 테스트이다. 4가지 예는 키 값과 초기화 벡터(IV) 값이 모두 같은 조건에서 테스트하였다.

웹 환경의 그래프는 Google Developers에 Google Chart를 이용으로 그래프를 설정하였다. Figure 5는 사용자가 설정하였던 키 값과 동일한 값이 입력되었을 때 나타나는 그래프로 line 그래프에 값이 보이고 하단의 line 그래프에는 controls 구간으로 기간을 선택하여 그래프의

I. Setting

Key	02 00 01 08 02 00 01 07 02 00 01 09 02 00 01 09
IV	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

II. Case 1

암호문	A9 C2 31 56 A8 BC C5 63 CE 49 98 7A FB D6 CA DB
Hex	62 61 6D 6D 62 6F 6F
Plain	bamboo

III. Case 2

암호문	AC 4A 67 E9 6C DB 61 00 4F 7C BE OF 2D TA 4B OE
Hex	4E 43 49 53 20 31 36
Plain	NCIS 16

IV. Case 3

암호문	4A 48 59 21 8F CD 74 C8 68 96 C3 DD DB AF 28 41 F4 SE 56 84 8D 46 DE 48 00 B9 A4 D4 F2 DD 66 60
Hex	32 30 31 39 2D 30 31 20 31 37 20 30 39 3A 35 33
Plain	2019-01-17 09:53

V. Case 4

암호문	EA B2 12 5D 19 AE D9 SE B2 4F EB E7 11 B1 F7 EO C4 79 6C 16 C8 51 76 4F 55 55 8F 25 30 FO 01 08 1C 51 56 A3 84 40 DB CE DB OS AF 3F 84 SD CI B4 0948 81 E7 79 9A 68 C4 13 22 AB 33 EB 53 F3 OB
Hex	70 6 65 60 65 6E 74 69 E 67 20 61 6E 20 65 6E 63 72 79 70 74 69 6F 6E 20 6C 69 62 72 61 72 79 20 75 73 69 6E 67 20 41 6E 64 72 6 69 64 20 54 68 69 6E 67 73
Plain	Implementing an encryption library using Android Things

Fig. 4. Encoding Test

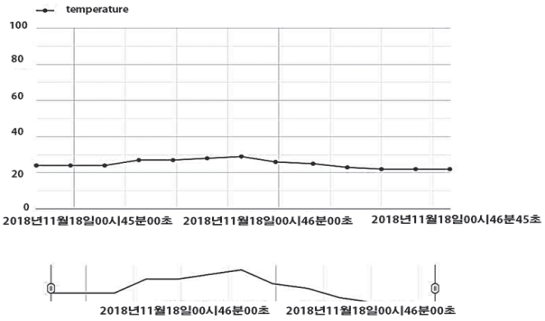

Fig. 5. Normal decoding graph

기간을 선택할 수 있도록 구현하였다. Line Chart와 Control bar를 생성할 영역을 지정한 후 복호화된 값을 각 열 마다 JSON 데이터로 바꾸어주었다.

4. 구현 결과

여기서는 개발된 결과에 대한 사용자 인터페이스 및 암복호화 결과를 기반으로 구현 결과에 대한 검증을 진행한다. 최종적으로는 암호화 키를 설정하고 암호화 라이브러리를 활용하는 'Android Things 앱'과 암호화된 정보를 담고 있는 '데이터베이스', 데이터베이스에서 정보를 가져와 복호화하여 그래프로 보여주는 '웹 환경'으로 구현이 진행된다.

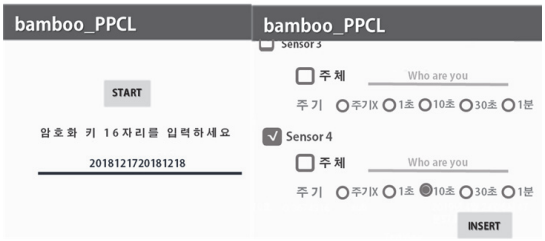

Fig. 6. PPCL Application setting screen

Figure 6은 사용자가 사용하는 앱 설정 화면이다. 가장 먼저, 앱을 실행할 시 암호화 키를 입력하는 창(Figure 6 왼쪽)이 뜬다. 암호화 키를 입력한 후 START 버튼을 누르면 다음 페이지인 센서 선택 페이지(Figure 6 오른쪽)로 넘어간다. 사용하려는 센서와 정책 정보를 모두 입력한 후 INSERT 버튼을 누르면 이 정보들이 암호화되어 데이터베이스로 전송되고 전송이 정상적으로 진행되면 'ACCESS'라는 알림이, 전송이 정상적으로 진행되지 않으면 'ERROR'라는 알림이 뜬다.

```
기존 환경 값 : 2018년 11월 17일 10시 11분15초
암호화 된 환경 값 : 1305746171654c5d22d01bd75071121010633687667063 42
                    69dfdbd2b004a51ac12d82201524bd5b10f92992df7311d76aa6241
기존 주체 값 : bamboo
암호화 된 주체 값 : 207f5a6267e80051 34e0e0b50eofd7b40000000000000
기존 센서 값 : 23.413548
 암호화 된 센서 값 : 47cfacagcceg3cd39e004b57eb8bd38f0000000000000000001
```

Fig. 7. Android Studio-log

Figure 7은 전송이 정상적으로 진행되어 'ACCESS'라는 알림이 떴을 때 Android Studio의 log 창이다. 전송이 정상적으로 진행되면 log와 같이 다음의 값이 나오게 된다. '기존 환경 값'은 측정된 시간이며 '기존 주체 값'은 사용자가 입력한 주체의 이름, '기존 센서값'은 측정된 센서 값이다. 각각의 기존 값에 대하여 암호화된 값이 '암호화된 환경 값,' '암호화된 주체 값,' '암호화된 센서 값'이다. Figure 8은 데이터베이스 화면이며 전송이 정상적으로 진행되었을 때 설정한 주기에 맞게 각 정책의 값들이 암호화되어 들어오는 것을 확인할 수 있다. (b), (c), (d) 는 Figure 8의 3개의 필드 데이터를 확대한 것이다.

enviro	subject ▼ 1	value
1305746171654c5d22d01bd75071	2075626728005134008065000	2775cc0a1f7b1e25b6ad5829462bd
1305746171654c5d22d01bd75071	2075626728005134008065000	22059171288212dd25b194de8084
1305746171654c5d22d01bd75071	2075626728005134008065000	140910db0141e8db310526692320
1305746171654c5d22d01bd75071	2075626728005134008065000	36550000258734752edce88b98392
1305746171654c5d22d01bd75071	2075626728005134008065000	399082ec6b0167237902411766000
1305746171654c5d22d01bd75071	2075626728005134008065000	37745414713146684 84.34eadslede
1305746171654c5d22d01bd75071	2075626728005134008065000	36550000258734752edce88b98392
1305746171654c5d22d01bd75071	2075626728005134008065000	271399082ec6b0167237902411766
1305746171654c5d22d01bd75071	2075626728005134008065000	37745414713146684 84.34eadslede
1305746171654c5d22d01bd75071	2075626728005134008065000	2071506267e80051340000650e0d7

(a) Database View

enviro
1305746171654c5d22d01bd75071121010633687667063 42

(b) Environment column

subject ▼ 1
2075626728005134008065000d7b4000000000000

(c) Subject column

value
2775cc0a1f7b1e25b6ad5829462bd000000000000000000

(d) Value column

Fig. 8. Database

Figure 9 형태의 웹의 첫 화면에서는 입력받은 키 값으로 복호화를 진행하여 그래프를 그리기 때문에 앱 실행 시 처음 설정했던 암호화 키와 같은 복호화 키를 입력해야 한다. 키를 제대로 입력했을 때는 Figure 10처럼 정확한 정보를 담은 그래프가 그려지지만 만일 암호화 키와 다른 키 값을 입력한다면 키가 아니더라도 그대로 복호화를 진행하여 그래프를 만들기 때문에 Figure 11과 같은 현상이 나타난다.

Fig. 9. Starting Page of Web Interface for IoT Data View

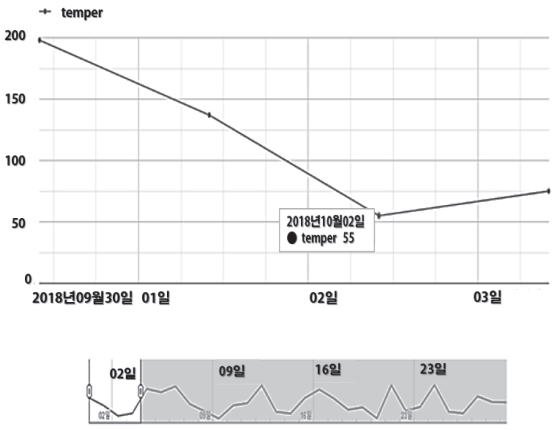

Fig. 10. Web Page View (When correct key is applied)

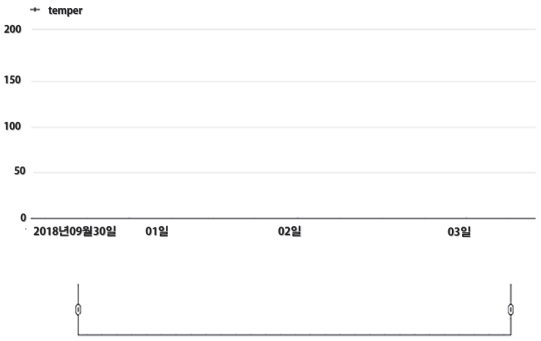

Fig. 11. Web page view (When incorrect key is applied)

5. 결론

본 논문에서는 SEED 기반 암호화 라이브러리를 사용하여 사용자의 센서 값을 암호화하고 복호화하는 방법을 제안하였다.

논문을 통해 기대한 첫 번째 효과는 Android Things를 사용함으로써 IoT 개발용이던 운영체제를 개발뿐만 아니라 보안에도 사용할 수 있다는 점이다. 두 번째는 IoT 해킹 차단으로 데이터를 암호화하여 송신 수신하여 만약에 공격자가 송수신 과정에서 데이터에 접근한다 해도 정확한 값은 보기 어렵다. 세 번째로는 이 논문에서 사용한 센서뿐만 아니라 암호화 라이브러리를 이용하면 다른 센서나 이미지나 음성과 같은 다양한 파일의 형식도 보안이 가능하여 새로운 보안 플랫폼을 제공 할 수 있다. 마지막으로 다양한 기기에 적용이 가능하여 본 연구의 암호화 과정을 이용하면 구글 어시스턴트와 같은 다양한 AI 기기의 데이터 송수신 시에도 보안을 적용할 수 있다.

이후로는 센서 정보가 라즈베리파이로 넘어오는 과정에도 보안을 적용하고 사용자가 센서뿐만 아니라 암호화 알고리즘도 선택할 수 있도록 개선하여 사용자의 편리함과 보안성을 더 높일 계획이다.

References

[1] Cho K.H.(2017) “IoT automatic management system based on sensor data identification algorithm,” *Graduate School of Paichai University*
(최강희 (2017) “센서 데이터 식별 알고리즘 기반 IoT 자동 관리 시스템,” 배재대학교 대학원)

[2] Cho Y.B,, S.B. Woo, S.B. Lee (2018) “A WPHR Service for Wellness in the Arduino Environment,” *Journal of the Korea Institute of Information and Communication Engineering*, 22 (1), 83-90
(조영복, 우성희, 이상호 (2018) “아두이노 환경에서 웰니슬 위한 WPHR 서비스,” 한국정보통신학회논문지, 22 (1), 83-90)

[3] Gartner, “2018 IoT Security Survey Report,” Available at https://www.gartner.com/en/documents/3855285 (Accessed May 8. 2019)

[4] Kim S.J., S.C. Park (2008) “RFID backward channel protection scheme by Partial Encryption scheme based on SEED.” *The Korea Institute of Information and Communication Engineering.* 12 (1). 215-222
(김성진, 박석천 (2008) “SEED 기반의 부분 암호화 기법을 이용한 RFID 백워드 채널 보호 기법.”한국정보통신학회논문지. 12 (1). 215-222)

[5] Kim D.H., S.W. Yoon, Y.P. Lee (2013) “Security and Telecommunications Association for IoT service,” *Information and Communication*, 30 (8), 53-59
(김동희, 윤석웅, 이용필 (2013) “IoT 서비스를 위한 보안,” 한국통신학회, 30(8), 53-59)

[6] Korea Internet & Security Agency[KISA], (2003) “Modes of Operation for the Block Cipher SEED,” 55

[7] Korea Internet & Security Agency[KISA], “Source code application manual for the SEED block cipher algorithm” Available at https://seed.kisa.or.kr/kisa/Board/17/detailView.do (Accessed May 8. 2019)

[8] Lee E.J. (2015) “Design and implementation of secure small-scale distributed data storage using IoT devices.” *The Graduate School Kyungpook National University*
(이응종(2015) “IoT 기기를 이용한 안전한 소규모 분산 데이터 저장소 설계 및 구현” 경북대학교 대학원)

[9] Lee S.W (2017) “Smart door lock systems using time synchronization based SEED algorithm.” *The Graduate School of Chung-Ang University*
(이성원(2017) “시간 동기화 방식 기반 SEED 알고리즘을 이용한 스마트 도어락 시스템” 중앙대학교 대학원)

[10] Park J.O. (2016) “A Study of Message Communication Method Using Attribute Based Encryption in IoT Environment,” *Journal of Digital Convergence*, 14 (10), 295-302
(박중오 (2016) “IoT 환경에서 속성기반 암호화 기술을 활용한 메시지 통신 기법에 관한 연구,”디지털융복합연구, 14 (10), 295-302)

[11] Shin M.J. (2019) “A Study on Secure Communication Frameworks for Various IoT Systems” *The Graduate School of Pukyong National University*
(신민정 (2019) “다양한 IoT 시스템을 위한 안전이 보장된 통신 프레임워크 연구” 부경대학교 대학원)

[12] Personal Information Protection Commission, “Analysis of social costs in the value of personal information and privacy,” Available at https://www.pipc.go.kr/cmt/not/ntc/selectBoardArticle.

do?nttId=4702&bbsId=BBSMSTR_000000000087 (Accessed May 8. 2019)

[13] Woo S.H.(2015) “Medical Information Security and Standard Technology On IoT Environment,” *Journal of the Korea Institute of Information and Communication Engineering*, 19 (11), 2683-2688 (우성희 (2015) “IoT 환경의 의료 정보보호와 표준 기술,” 한국정보통신학회논문지, 19(11), 2683-2688)

박 화 현 (ORCID : https://orcid.org/0000-0002-2199-7001 / 0425pipi@swu.ac.kr)

2016. 3~ 현재 서울여자대학교 정보보호학과 학부생 연구원

관심분야 : 정보보호, 사물인터넷, 센서 보안, 모델링 및 시뮬레이션

윤 미 경 (ORCID : https://orcid.org/0000-0002-6013-9341 / ehreh2698@swu.ac.kr)

2016. 3~ 현재 서울여자대학교 정보보호학과 학부생 연구원

관심분야 : 정보보호, 개인정보보호, IoT보안, 모델링 및 시뮬레이션

이 현 주 (ORCID : https://orcid.org/0000-0002-9589-6916 / hyeon27@swu.ac.kr)

2016. 3~ 현재 서울여자대학교 정보보호학과 학부생 연구원

관심분야 : 정보보호, 클라우드, 데이터베이스, 모델링 및 시뮬레이션

이 해 영 (ORCID : https://orcid.org/0000-0001-8918-3031 / haelee@cju.ac.kr)

2003 성균관대학교 정보통신공학부 공학사
2009 성균관대학교 컴퓨터공학과 공학박사
2009~2013 한국전자통신연구원 선임연구원
2013~2017 서울여자대학교 조교수
2017~2019 (주)두두아이티 연구소장
2019~현재 청주대학교 디지털보안전공 조교수
관심분야 : IoT 보안, 시뮬레이션 보안, 사이버 레인지

김 형 종 (ORCID : https://orcid.org/0000-0002-0608-5397 / hkim@swu.ac.kr)

1996 성균관대학교 정보공학과 공학사
1998 성균관대학교 정보공학과 공학석사
2001 성균관대학교 전기전자 및 컴퓨터공학과 공학박사
2001~2007 한국정보보호진흥원 수석연구원
2004~2006 미국 Carnegie Mellon University CyLab Visiting Scholar
2007~현재 서울여자대학교 정보보호학과 교수

관심분야 : IoT보안 및 시뮬레이션, 블록체인 서비스 성능평가, 이산사건 시뮬레이션 방법론