Master's Thesis

Securing the Integrity of Open mHealth-Compliant Data by using TPM 2.0

Mirae Lim (임 미 래)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2019

TPM 2.0 을 이용한 Open mHealth 데이터 무결성 검증 방법

Securing the Integrity of Open mHealth-Compliant Data by using TPM 2.0

Securing the Integrity of Open mHealth-Compliant Data by using TPM 2.0

by

Mirae Lim
Department of Computer Science and Engineering
Pohang University of Science and Technology

A thesis submitted to the faculty of the Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Computer Science and Engineering

Pohang, Korea
12. 20. 2018
Approved by
Chanik Park (Signature)
Academic Advisor

Securing the Integrity of Open mHealth-Compliant Data by using TPM 2.0

Mirae Lim

The undersigned have examined this thesis and hereby certify that it is worthy of acceptance for a master's degree from POSTECH

12. 20. 2018

Committee Chair	Chanik Park	(Seal)
Member	Jong Kim	(Seal)
Member	Gwangsun Kim	(Seal)

MCSE 임 미 래, Mirae Lim
20172826
Securing the Integrity of Open mHealth-Compliant Data by using TPM 2.0,
TPM 2.0 을 이용한 Open mHealth 데이터 무결성 검증 방법
Department of Computer Science and Engineering, 2019, 40p, Advisor : Chanik Park

ABSTRACT

In this thesis, we propose a protocol to secure the integrity of Open mHealth-Compliant data by using Trusted Platform Module (TPM) 2.0. This allows the Data Consumer to verify the integrity of the Data points generated on the open mhealth platform. To implement the protocol, we use raspberry pi 3 board based on Ubuntu 16.04 as IoT gateway and another raspberry pi 3 board based on Raspbian as Data Consumer. The proposed protocol generates the integrity information to guarantee the acquisition provenance and sequence of data by using PCR, QUOTE in TPM and hash chain. When the protocol is applied, the integrity information and key certificate for this is included in the header part of original Data point, and the ID of Data point is set to hash value of the previous Data point. We call this newly defined Data point as Extended Data point. Extended Data points are saved in Data Storage Unit (DSU). Later, Data Consumer can obtain Extended Data points from DSU. We also use Hardware TPM2.0 provided by IBM to protect Data point from malicious attacker in DSU and to verify in Data Consumer. We implemented a prototype in raspberry pi and performed the performance evaluation. To generate Extended Data point, it needs additional time to generate key certificate and integrity information. As a result, we show that it takes 2~20 times more than to generate original Data point.

Contents

I. 서론

센서에서부터 얻어온 데이터들은 분석 및 수정 후에 저장장치에 저장이 된다. 그러나 클라우드 서버까지 데이터들이 전송되고 데이터들이 처리되기에는 꽤 오랜 시간이 소요된다. 따라서 센서 데이터 정보를 클라우드 서버로 보내기 전 중간 처리하기 위해 중간 사물 인터넷 디바이스(예: smart, gateway, router, controller)들이 사용된다. 이 경우, 중간 사물 인터넷 디바이스에서 센서 데이터들이 분석 및 처리 과정을 거쳐서 클라우드 서버에 저장이 된다. 특히, 건강에 대해 관심을 갖는 사람들이 늘어나고 있는 요즘 건강 데이터 처리 방법에 대한 연구가 많이 이루어지고 있다. 모바일 건강 데이터의 경우 Open mHealth[1]라는 플랫폼을 이용하여 센서에서 얻어온 모바일 건강 데이터를 표준화된 데이터 포맷인 Data point 로 변경하여 저장 장치인 Data Storage Unit(DSU)에 저장한다. Data point 는 모바일 건강 데이터를 관리하는데 용이하여 기업이나, 기관 혹은 개인이 데이터를 쉽게 이용하고 처리하는데 도움을 준다. 이렇게 얻어온 건강 데이터는 추후 사용 시 사용자가 정확한 데이터를 사용하고, 정확한 진단에 데이터가 이용될 수 있도록 무결성을 검증하는 과정이 필요하다. 게다가 점점 더 많은 사람들이 모바일 건강 데이터를 사용하고 있는 현대 사회에서 2016 년에 IT 전문가들은 모바일 건강 데이터에 대한 위험을 제기하였다[2]. 이렇게 모바일 건강 데이터에 대한 무결성 검증의 필요성이 증가됨에 따라, 모바일 건강 데이터 무결성 검증의 중요성이 커지고 있는 추세이다.

데이터 무결성을 검증하는 요소는 3 가지가 있다. 첫 번째로 데이터 내용에 대한 무결성 검증이다. 이는 실제로 얻어온 데이터의 변경 유무를 검증하는

것이다. 두 번째로 데이터 출처 정보에 대한 무결성 검증이다. 이는 실제 데이터가 어디서 생성되었는지, 언제 생성되었는지, 어떻게 생성 되었는지에 대한 정보의 변경 유무를 검증하는 것이다. 마지막으로 데이터 순서에 대한 무결성 검증이다. 데이터는 저장장치에 저장된 후 악의적인 사용자에 의하여 저장된 순서가 변경될 수 있다. 따라서 데이터가 전송되는 순서의 변경 유무를 검증하여 데이터의 순서를 보장해야 한다.

본 논문에서는 Trusted Platform Module (TPM) 2.0 을 이용한 건강 데이터의 무결성 검증 프로토콜을 제안한다. 저장 장치에 저장되어 있는 데이터가 공격을 받을 수 있다고 가정하며 IoT gateway 에서 무결성 검증 프로토콜을 수행한다. 모바일 건강 데이터의 무결성 검증 프로토콜 수행하는 IoT gateway 에서는 보안 모듈인 TPM 을 사용하며, 2 가지 요소인 데이터 생성 정보 무결성 검증, 데이터 순서 무결성 검증에 대해 고려한다. 이를 통해 데이터 소비자는 전송 받은 데이터의 무결성을 검증할 수 있다.

이어서 chapter 2 는 건강 데이터의 무결성 검증 프로토콜에 필요한 배경 지식을 다루며, chapter 3 관련 연구를 다룬다. 또한 chapter 4 에서는 제안한 프로토콜을 설계한다. chapter 5 에서는 제안한 프로토콜에 대한 보안 분석을 서술하며, chapter 6 와 chapter 7 에서는 각각 실험과 그에 대한 한계점에 대해 서술한다. 마지막으로 chapter 8 에서 결론을 다룬다.

II. 배경 지식

2.1 TPM 2.0 배경 지식

TPM (Trusted Platform Module)[6]은 통합 암호 키를 통해 하드웨어를 보호하도록 설계된 전용 마이크로 칩이다. 주로 암호화 키를 포함하며, 디바이스 확인, 입증, 암호화, 무결성 검증에 이용된다. TPM 은 일반적으로 컴퓨터의 마더 보드에 설치되며, 하드웨어 I/O 를 통하여 주변 기기와 통신한다. 기존 TPM 1.2 를 사용하던 중, 2005 년에 암호 사용자는 SHA-1 hash 알고리즘 사용하는데 있어 공격을 받았다. Trusted Computing Group(TCG)에서는 SHA-1 hash 알고리즘을 사용하는 TPM 1.2 가 더 이상 안전하지 않다는 것을 인지하고, TPM 2.0 을 개발하기 시작했다. 이 결과로 2014 년 10 월 TCG 는 TPM 2.0 library Specification[6,7,8]을 배포하였다. 이는 TPM 1.2 버전과 비교하여 암호화 알고리즘과 enhanced authorization 이 추가되었다. TPM 1.2 에서는 RSA 2048bit 와 SHA-1 hash 알고리즘을 제공하였다면, TPM 2.0 에서는 이와 더불어 ECC, SHA-2 AES, HMAC 알고리즘이 확장되었다. 또한 Authorization 부분에서도 TPM 1.2 버전에서는 SHA-1 hash 나 Platform Configuration Register(PCR), 바인드 된 인증 메커니즘을 패스워드로 이용하였다면, 2.0 버전에서는 policy command 를 통하여 더 추상적인 인증을 제공한다[6].

2.1.1 Platform Configuration Register (PCR)

PCR[6]은 측정된 로그의 내용을 검증하는 데 사용되는 TPM 내부의 영역이다. PCR 은 휘발성 메모리이며 여러 개로 구성되어 있다. PCR 값은 *Event()* 나 *Extend()* 명령어를 통해서만 수정(Figure 2.1)되며, 아래 식과 같이 기존 PCR 값에 새로운 데이터의 hash 값이나 실제 hash 값을 병합한 값으로만 변경할 수 있다. PCR 만의 특수한 값 업데이트 방식으로 인하여, PCR 은 디바이스가 처음 시작할 때부터 hash chain 을 이용하여 소프트웨어나 펌웨어의 상태를 저장 및 유지하는 역할로 사용되고 있다. 또한 PCR 은 Remote Attestation 시 사용되며, 이때 선택적으로 고른 PCR 값을 서명하여 제 3 자에게 보고한다.

tpm2_pcrevent -i N *datafile*	……. (1)
tpm2_pcrextend N:alg=*hashdata*	……. (2)

Figure 2.1. TPM PCR 수정을 위한 명령어

2.1.2 QUOTE 명령어

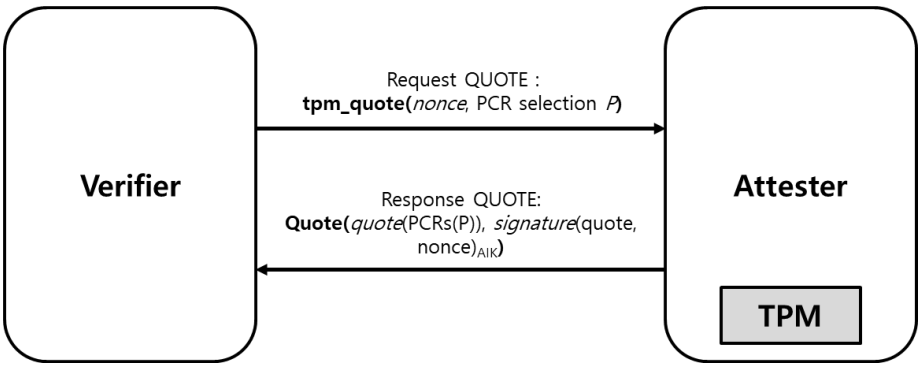

Figure 2.2. QUOTE 를 이용한 TPM Attestation.

QTUOE[7, 8]는 TPM 에서는 3 자가 기기에 대한 증명을 확인하기 위해 주로 사용하는 방법이다. QUOTE (Figure 2.2)로 증명하기 위해 TPM 에서는 주로 PCR 값을 사용하며, 이 PCR 값은 TPM 에서만 확인될 수 있는 값이다. QUOTE 로 PCR 을 증명하기 위하여 제 3 자는 증명을 위해 선택적으로 고른 PCR 들과 nonce 를 포함하여 같이 요청을 한다. QUOTE 요청을 받은 TPM 에서는 QUOTE 증명을 생성해내며, 이는 QUOTE 메시지와 이에 대한 서명으로 구성되어있다. QUTOE 증명 생성 시 먼저 선택된 PCR 값들로 hash 연산을 하여 QUOTE 메시지를 생성한다. 또한 QUOTE 메시지와 nonce 값을 같이 서명한다. 서명 생성 시 TPM 에서 생성된 Attestation Identify Key(AIK)를 사용한다.

이렇게 생성된 QUOTE 증명을 검증하기 위해서는 PCR 에 저장된 정보들로 다시 QUOTE 메시지를 생성한다. 새로 생성한 QUOTE 메시지와 nonce 값을 이용하여 성명 증명한다[8, 9].

2.1.3 TPM monotonic counter

TPM 2.0 에서는 TPM NV counter 라고 불리는 TPM monotonic counter 가 존재한다[6]. NV counter 는 64bit 로 구성되어 있으며 이 값은 오직 증가만 할 수 있다. NV counter 는 현재 정의된 counter 의 index 와 과거에 정의되었지만 TPM 에서 아직 사용하고 있지 않은 counter 들로 구성되어 있다. 따라서 index 를 지우거나 재 생성해도 사용할 수 없다.

TPM 2.0 에서 사용되고 있는 NV counter 는 TPM 1.2 에서 monotonic counter 와 역할은 같지만, 사용자가 TPM 메모리 영역에 따라 counter 를 정의하고 삭제할 수 있다는 점에서 다른 특징을 갖고 있다.

2.2 Open mHealth 배경 지식

Open mHealth[1]는 모바일 건강 데이터를 관리하는데 적합한 플랫폼으로써 회사나 기관, 개인들이 데이터를 교환하거나 재사용할 시 도움을 준다. Open mHealth 를 통해서 모바일 건강 데이터는 더 쉽게 접근 가능 하도록 만들어 준다. 이를 위해 Open mHealth 는 schema[10, 11]를 정의하며, 이는 소프트웨어 프로그램이 해당 건강 데이터를 처리하는 방법에 영향을 준다. 정의된 schema 를 이용하여 Open mHealth 는 모바일 건강 데이터의 첫 번째, 표준화 된 형태인 Data point 를 생성 해낸다.

2.2.1 Data point

Data point[10]는 센서에서 측정된 데이터를 Open mHealth 에서 정의한 schema[10, 11]를 이용하여 변경시킨 모바일 건강 데이터의 표준화된 데이터이다. Data point 는 데이터가 어디서 생성되었는지, 어디서 전송되었는지 상관 없이 좀 더 처리하고 이해하기 쉽도록 해준다. Data point (Figure 2.3)는 header 와 body 로 구분할 수 있다. Data point 의 header 에는 Data point 의 ID, 생성 시간, 데이터가 생성된 곳과 시간, 생성 방법이 포함되어 있는 데이터의 출처 정보, 데이터 body 에서 사용되고 있는 schema 가 저장된 위치 schema 이름, schema 버전이 포함되어 있는 schema 정보가 포함되어 있다. Data point 의 body 에는 Data point 의 header 에 포함되어 있는 schema 정보를 이용하여 실제 건강 데이터의 값을 저장한다.

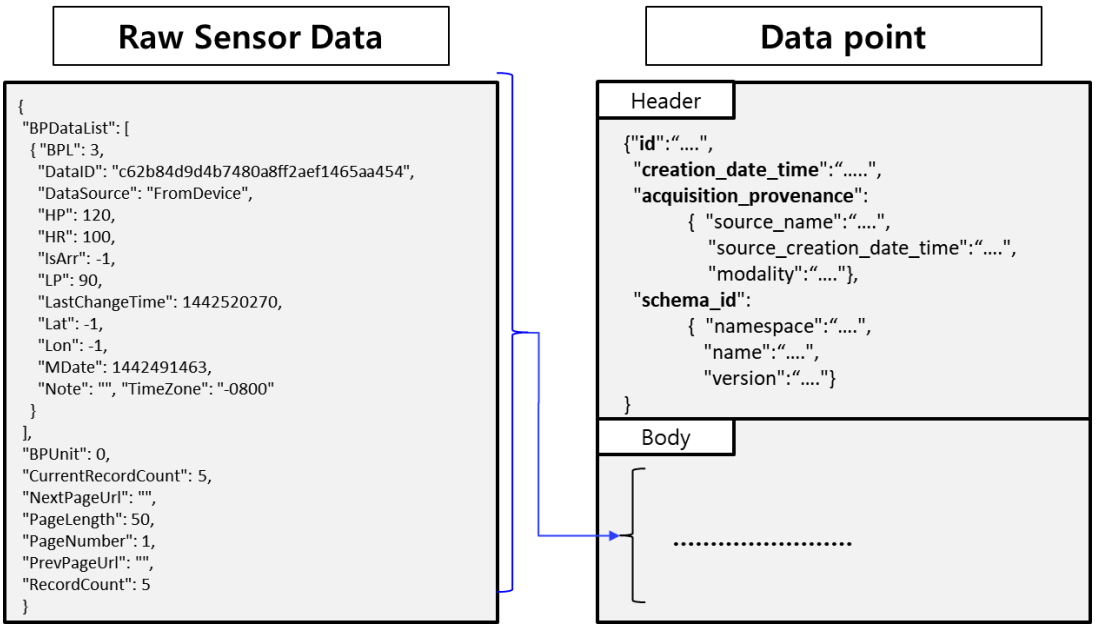

Figure 2.3. ihealth 의 혈압 데이터와 이에 대한 Data point 예시.

III. 관련 연구

3.1 Temper Detection in Audit Logs

감사 로그는 비즈니스 시스템의 우수 사례로 여겨지며 보안 시스템, 의료 정보 유출 등에 대한 규정에서 필요로 한다. 감사 로그의 중심 역할을 감안했을 때, 감사 로그 그 자체의 무결성을 보장하는 것 또한 중요하다. Temper Detection in Audit Logs[3] 에서는 Data Base Management System(DBMS) 메커니즘을 제안하였다. 이는 암호로 강력한 단 방향 hash 함수를 기반으로 하였으며 감사자, 직원, DBMS 침입자로부터 감사 로그가 충돌이 일어나는 것을 막는다. 감사 로그를 보호하기 위하여 감사 로그를 공증할 때마다 감사 로그와 시간 값을 hash 함수로 계산한다. 또한, 매번 이렇게 hash 함수를 계산할 때 생기는 오버 헤드를 줄이기 위하여 공증 횟수를 줄이거나 기록 테이블을 만드는 방법을 사용하였다. 감사 로그의 무결성을 증명할 시 hash 함수를 사용하며, 로그 출처 정보나 순서의 무결성을 고려하지 않았다는 점에서 본 논문과 다른 점을 지닌다.

3.2 Pasture

Pasture[4]는 신뢰할 수 있는 서버가 제공한 데이터에 대하여 오프라인 상태에서 사용자의 접근 여부를 판단할 수 있게 하는 시스템을 제안하였다. TPM 에 있는 PCR 을 사용하였고, 이를 증명하기 위하여 QUOTE 를 이용하였다는 점은 같지만, Pasture 는 사용자의 네트워크 상태가 항상 온라인일 수 없기 때문에 오프라인 상태에서 접근 여부를 판단하는 것에 초점을 둔다. 반면 본 논문은 전송 받은 데이터 출처와 데이터 순서가 신뢰할 수 있으며,

이의 변경 유무에 초점을 둠으로써 데이터 출처와 순서의 무결성을 좀 더 고려한다.

3.3 Temper-proof Logging

Temper-proof Logging[5]은 시스템에서 생성된 로그를 TPM 2.0 기반으로 검증하는 logger 프로그램을 제안하였다. 시스템에서 생성된 로그를 검증하기 위한 방법으로 HMAC 을 사용하였으며, HMAC 계산 시 사용되는 key 를 보호하기 위해 이 key 를 TPM 을 이용하여 암호화를 하였다. 네트워크가 오프라인 상태일 때를 고려하여, TPM monotonic counter 를 사용하여 로그에 대한 검증을 하였다. 또한 Temper-proof logging 은 본 논문과 달리 로그의 업데이트 주기, 로그에 대한 용량은 어떻게 유지할 것인지, 로그의 효과적인 검증 방법에 더 초점을 맞추어 프로그램을 제안하였다.

IV. 제안한 프로토콜

본 논문은 TPM 2.0 을 이용하여 모바일 건강 데이터 출처 정보와 순서 정보 무결성 검증 방안을 제안한다. 이 검증 방안을 통해 생성된 데이터는 나중에 데이터 사용자가 실제로 모바일 건강 데이터의 출처 정보가 정확한지, 모바일 건강 데이터의 순서가 올바른지 판별할 수 있어야 한다. 이 검증 방안을 통해 생성된 검증 정보들과 Open mHealth 플랫폼에 정의 된 schema[10, 11]를 이용하여 새로운 Data point 를 정의하였다. 이러한 Data point 를 Extended Data point 라고 부른다. 이때, 데이터 소비자는 검증이 완료된 Extended Data point 만이 무결성이 검증 된 데이터라고 판단하게 된다.

4.1 시스템 구조

시스템 (Figure 4.1)은 건강 데이터를 얻어오는 센서 디바이스, IoT gateway, 데이터가 저장되는 Data Storage Unit(DSU), 데이터를 검증하는 데이터 소비자(Data Consumer)로 이루어져 있다. IoT gateway 는 데이터 무결성 검증 정보를 포함하기 위해 TPM 을 갖고 있다. TPM 내부에서 PCR 11 은 생성된 Extended Data point 의 hash 값들을 유지하며, PCR 12 는 데이터 출처 정보의 hash 값을 유지한다. PCR 11 은 매번 Extended Data point 가 생성될 때마다 업데이트가 되며, PCR 12 는 처음 데이터 출처 정보를 업데이트 한 후 그대로 유지된다. 또한 TPM 내부에 2 개의 NV memory 영역을 정의해 둔다. 첫 번째 NV_INDEX_ID 영역은 Extended Data point 의 hash 값이 저장되어 있어 이를 다음 Extended Data point 생성 시 ID 로 사용하고 있으며, 두 번째 NV_INDEX_PCR 영역은 PCR 11 의 값을 저장하고 있어 PCR 이 초기화 될 경우에 대비하여 초기화가 일어날 경우 업데이트 해야 할 값들을 저장하고

있다. 센서 디바이스에서 데이터가 생성 되면, TPM 이 설치되어 있는 IoT gateway 에서 무결성 정보를 포함한 Extended Data point 를 생성해내며, 이를 Data Storage Unit 에 저장한다. 생성된 Extended Data point 는 추후 데이터 소비자가 사용 시 검증을 진행하여 사용하게 된다.

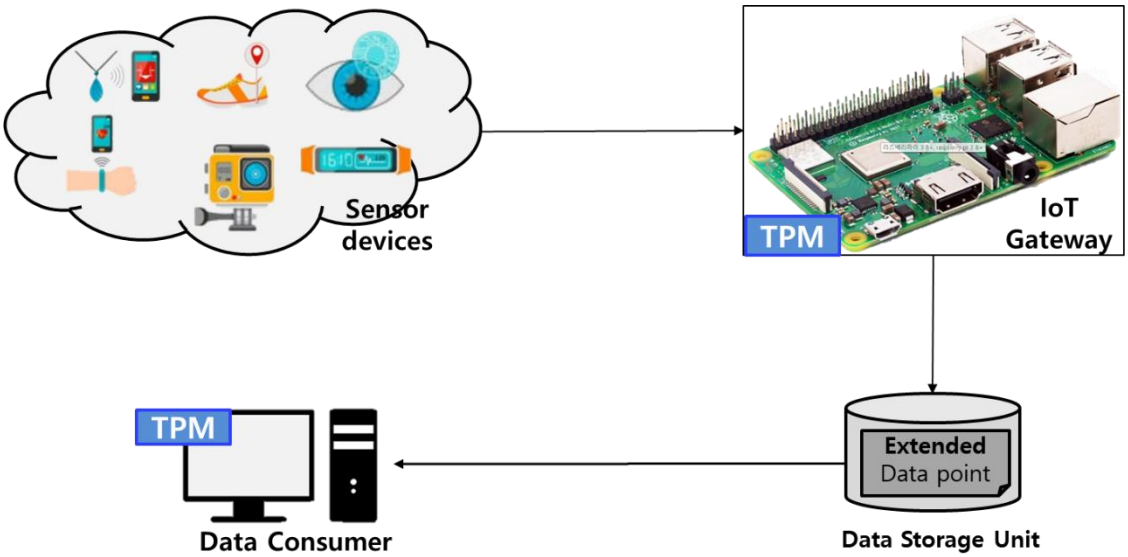

Figure 4.1. 무결성 검증 시스템 구조. **Sensor Device**: 센서 데이터 생성. **IoT gateway**: Extended Data point 생성. **Data Storage Unit(DSU)**: Extended Data point 저장. **데이터 소비자(Data Consumer)**: Extended Data point 사용

4.2 위협 모델

본 연구에서 공격자는 Data point 가 저장되는 DSU 에 대한 공격이 가능하다고 가정한다. 즉, TPM 을 활용하여 Data point를 생성하는 IoT gateway에 대한 공격은 없다고 가정한다. 공격자가 DSU 에 대한 해킹 공격이 성공한다면, DSU 에 저장되어 있는 데이터의 출처 정보 및 순서 정보 등 Data point 의 모든 정보들이 가 변경될 수 있다.

4.3 가정

센서 디바이스(e.g. 혈압 측정기, 혈당 측정기)에서 측정된 데이터는 안전하며, 이를 IoT gateway 로 전송하는 과정도 안전하다고 가정한다. IoT gateway 또한 안전하여 전송 받은 데이터가 변경되지 않고 유지되며, 이를 DSU 에 저장하는 과정도 안전하게 보장된다고 가정한다. IoT gateway 는 전원이 항상 유지 되어 모바일 건강 데이터 전송이 중단되는 일이 없다고 가정한다(Figure 4.1). TPM 은 PCR 11 통해 지속적으로 생성된 Extended Data point 를 저장할 수 있으며, PCR 12 를 통해 데이터 출처 정보를 저장할 수 있다. 또한 PCR 11 과 12 를 이용한 QUOTE 증명을 통해 요청한 Extended Data point 가 변경되지 않고 안전한 상태인지 데이터 소비자가 판별할 수 있다.

4.4 요구 사항

이 무결성 검증 기법은 센서 디바이스에서 측정된 데이터의 무결성을 검증하기 위하여 2 가지 (1)데이터 출처 무결성, (2) 데이터 순서 무결성을 보장해야 한다.

(1) **데이터 출처 무결성** 모바일 건강 데이터를 Data point 형태로 바꾸어 전송할 때, 데이터가 어디서 생성 되었는지, 언제 생성 되었는지, 어떻게 생성되었는지에 대한 데이터 출처 정보가 Data point header 에 포함되어 있다. 데이터 사용자는 Data point 를 사용할 때 해당 정보가 맞는지 판단할 수 있어야 한다.

(2) **데이터 순서의 무결성** Data point 는 DSU 에서 저장될 때 순서대로 저장된다. 하지만 저장된 후 DSU 에서 악의적인 사용자에 의하여 순서가 바뀔 수 있다. 따라서 데이터 사용자는 Data point 를 사용할 시 이에 대한 순서를 검증할 수 있어야 한다.

4.5 검증 프로토콜

무결성 검증 프로토콜은 (1) 초기화 단계 (2) 무결성 검증 정보 생성 단계 (3) Extended Data point 생성 및 저장 단계 (4) 무결성 정보 검증 단계로 이루어진다(Figure 4.2). 초기화 단계에서는 센서 디바이스에서 데이터를 생성해낸다. IoT Gateway 에서는 데이터 무결성 검증 정보 생성을 위해 TPM 에서 Endorsement Key(EK)를 얻고 이를 이용하여 Attestation Identity Key(AIK)를 생성하며, EK 와 AIK 에 대한 인증을 각각 얻는다. 무결성 검증 정보 생성 단계에서는 데이터 출처에 대한 무결성 검증과 데이터 순서에 대한 무결성 검증 정보를 생성한다. 데이터 출처에 대한 무결성 검증을 위하여 데이터의 데이터 출처 정보를 TPM PCR 12 에 extend 해놓고, PCR 12 에 대한 QUOTE 증명을 실행한다. 이때 QUOTE 를 하기 위하여 초기화 단계에서 생성하였던 TPM AIK 를 이용한다. 데이터 순서의 무결성 검증을 위해서는 Data point 새로 생성 시 ID 를 이전 Data point 의 hash 값으로 설정해 준다. 또한 Data point 생성

후 PCR 11 에 hash 값을 항상 extend 하며, PCR 11 에 대한 QUOTE 증명을 한다. 이때, 데이터 출처 정보 검증 시 사용했던 key 와 동일한 key 를 사용하며, QUOTE 의 순서를 보장하기 위해 nonce 값으로 TPM monotonic counter 를 사용한다. Extended Data point 생성 및 저장 단계에서는 이렇게 생성된 AIK 에 대한 증명과 QUOTE 증명 결과, 새로 생성된 ID 를 이용하여 새로 정의한 Extended Data point 를 생성해내며 이를 DSU 에 저장한다. 무결성 정보 검증 단계에서는 Extended Data point 를 사용하고자 하는 데이터 소비자가 Extended Data point 에 포함된 무결성 검증 정보를 보고 데이터 출처 정보와 데이터 순서에 대한 무결성을 검증하여 데이터의 신뢰성을 판별할 수 있다.

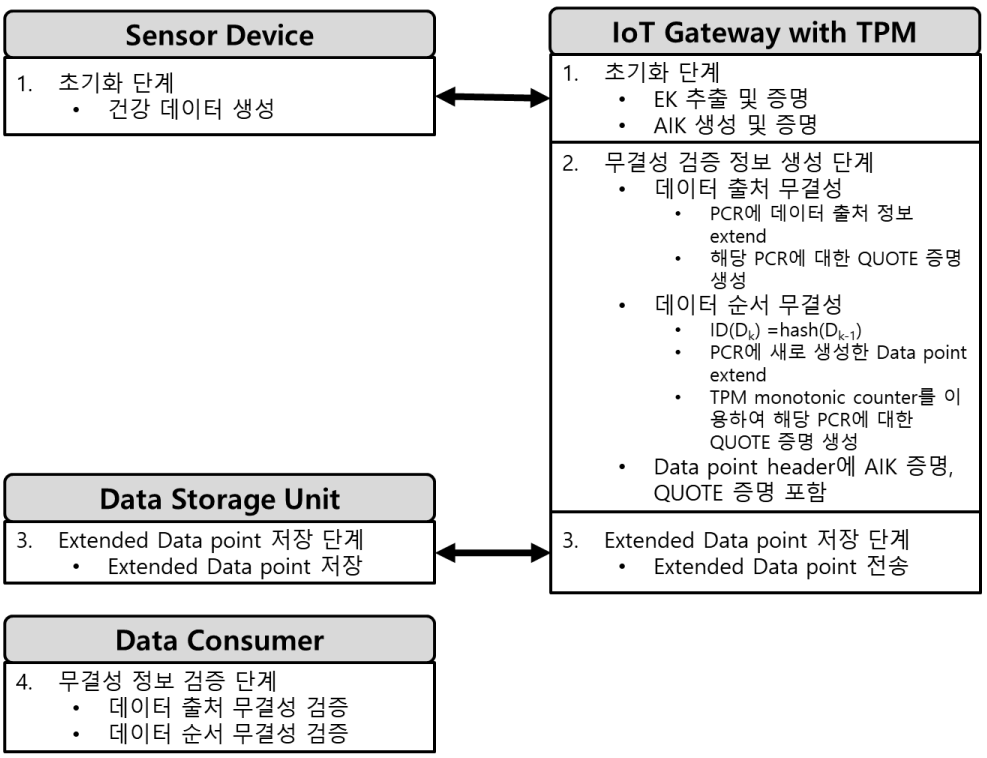

Figure 4.2. 데이터 무결성 검증 프로토콜 시나리오. 4 단계로 구성: 초기화 단계, 무결성 검증 정보 생성 단계, Extended Data point 생성 및 저장 단계, 무결성 정보 검증 단계

4.5.1 초기화 단계

초기화 단계 (Figure 4.3)는 센서 데이터를 생성해내며, 무결성 검증 기법에서 무결성 검증 정보 생성 시 필요한 TPM Endorsement key(EK), Attestation Identity Key(AIK)를 생성하며, 이에 대한 인증을 얻는다. 먼저 센서 디바이스에서는 건강 데이터를 측정하며, 이를 TPM 이 설치되어 있는 IoT gateway 로 전송한다. TPM 이 설치되어 있는 IoT gateway 에서는 TPM 에서 이미 설정 되어 있는 EK 의 public part 를 얻으며 이를 TPM 의 지정된 영역에 영구적으로 저장한다. EK 얻어온 후 EK 가 정확하게 생성되었는지 TPM 제조사의 endorsement certificate hosting server 에서 EK 에 대한 인증을 얻어온다. 이렇게 인증된 EK 를 이용하여 AIK 를 생성하며, 이 또한 TPM 의 지정된 영역에 영구적으로 저장한다. AIK 생성 후 생성된 AIK 가 TPM 에서 생성 되었는지에 대한 인증을 얻어 오기 위하여 PCA 에서 AIK 에 대한 인증[12]을 얻어온다. 초기화 단계에서는 위 과정을 통해 실제로 TPM 에서 EK 를 얻고 AIK 를 생성해내며, 이들을 각각 인증할 수 있다. AIK 에 대한 인증은 Data point header 에 포함시킨다.

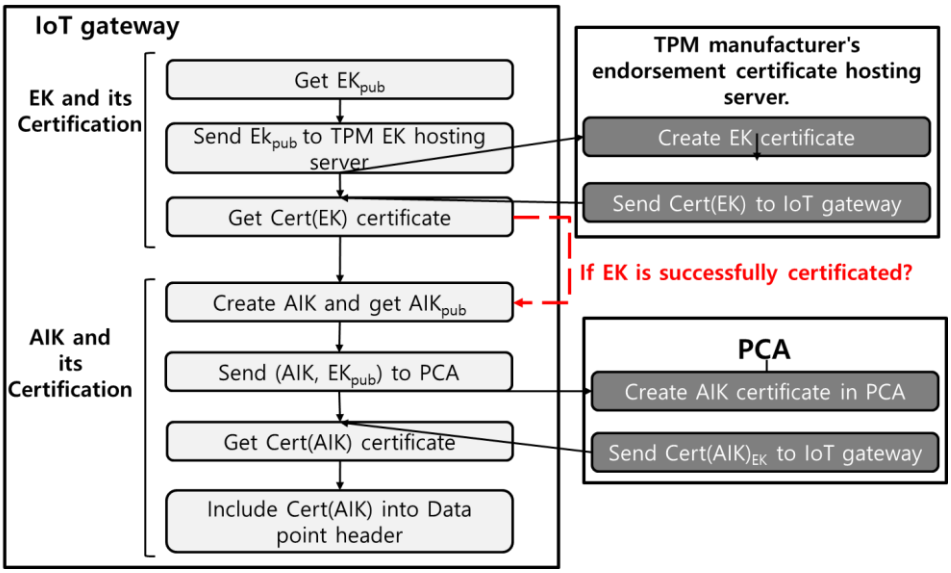

Figure 4.3. 초기화 단계: EK 와 그에 대한 인증을 얻고, AIK 생성과 그에 대한 인증을 얻는다.

4.5.2 무결성 검증 정보 생성 단계

무결성 검증 정보 생성 단계는 초기화 단계에서 생성했던 AIK 를 이용하여 데이터 출처에 대한 무결성 정보, 데이터 순서에 대한 무결성 정보를 생성해낸다.

(1) **데이터 출처 무결성 검증 정보 생성** 데이터 출처에 대한 무결성 정보(Figure 4.4)를 생성 해내기 위해 먼저 데이터 출처 정보의 hash 값을 PCR 12 에 extend 한다. PCR 12 을 증명하기 위해 TPM QUOTE 를 수행한다. QUOTE 수행 시 초기화 단계에서 생성했던 AIK 를 이용하며, PCR 선택 인자로 PCR 12 를 포함시킨다. QUOTE 증명 결과는 Data point 의 header 에 포함시킨다. 또한 데이터 출처에 대한 정보는 데이터 검증 처음 시작 시에만 실행된다.

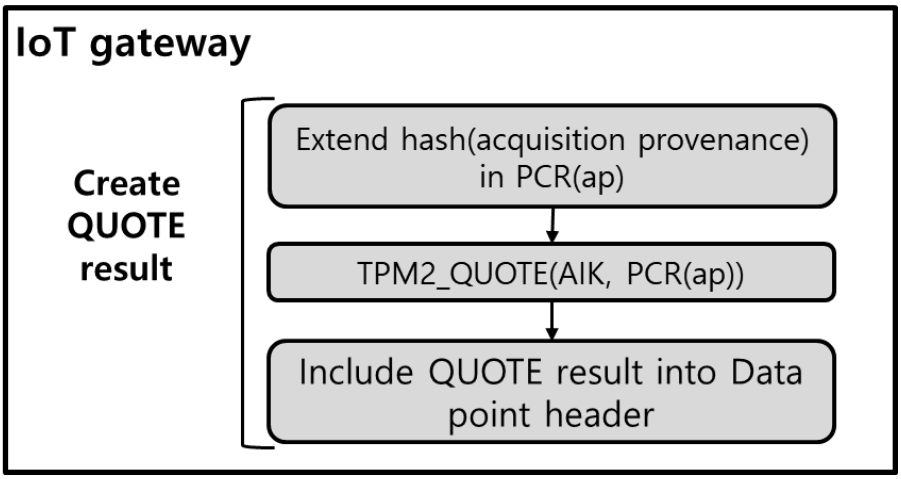

Figure 4.4. 데이터 출처 무결성 정보 생성 과정

(2) **데이터 순서 무결성 정보 생성** 데이터 순서에 대한 무결성 정보 단계(Algorithm 1)에서는 다음과 같은 정보를 유지해야 한다. 이전 Data point 의 hash 값을 TPM NV 메모리의 NV_INDEX_ID(0x1500000)에 유지하고 있어야 하며, TPM 의 PCR 12 에는 이전 모든 Data point 의 hash 값이 extend 되어 있어야 한다. 또한 TPM NV 메모리의 NV_INDEX_PCR(0x1500100)에는 현재 PCR 11 의 값이 유지되고 있어야 한다.

이렇게 기본적으로 유지되고 있는 정보를 이용하여 새로운 Data point 를 생성한다. NV_INDEX_ID 에 저장되어 있는 이전 Data point 의 hash 값을 읽어와 새로 생성되는 Data point 의 ID 로 사용한다. 이후 TPM QUOTE 를 이용해 PCR 11 에 대한 증명을 생성한다. 이때, 초기화 단계에서 생성했던 AIK 를 이용한다. QUOTE 증명 결과는 Data point header 에 저장된다. 이 경우 사용자가 Data point 생성 시 ID 를 악의적으로 바꾸어 새로운 sequence 를 만들어 낼 수 있다 (Figure 4.5). 따라서 QUOTE 결과 값의 순서 또한 보장되어야 한다. 이를 위해 QUOTE 증명 시 단조 증가하는 counter 값인 TPM monotonic counter 값을 읽어와 nonce 값으로 사용하며, QUOTE 증명 후에 이 counter 값을 증가 시킨다. 새로운 Data point 생성 후 이 Data point 의 hash 값을 PCR 11 에 extend 시키며 NV_INDEX_ID 에 저장시킨다. 업데이트 된 PCR 11 의 값은 NV_INDEX_PCR 에 저장하여 PCR 11 의 값이 초기화 되었을 시 복구 값으로 사용한다.

Algorithm 1 무결성 검증 정보 생성 단계

0: NV_INDEX_ID = hash(D_{k-1})

NV_INDEX_PCR = PCR(11)

PCR(11) = hash(PCR(11) || hash(D_{k-1}))

1: result = tpm2_nvread(NV_INDEX_ID);

2: ID(D_k) = result

3: nonce = tpm monotonic counter

4: quote_result() = tpm2_quote(PCR(11), nonce)

5: increase tpm monotonic counter

6: include quote_result() into Data point header

7: PCR(11) = hash(PCR(11) || hash(D_k))

8: tpm2_nvwrite(NV_INDEX_ID, hash(D_k))

9: tpm2_nvwrite(NV_INDEX_PCR, PCR(11))

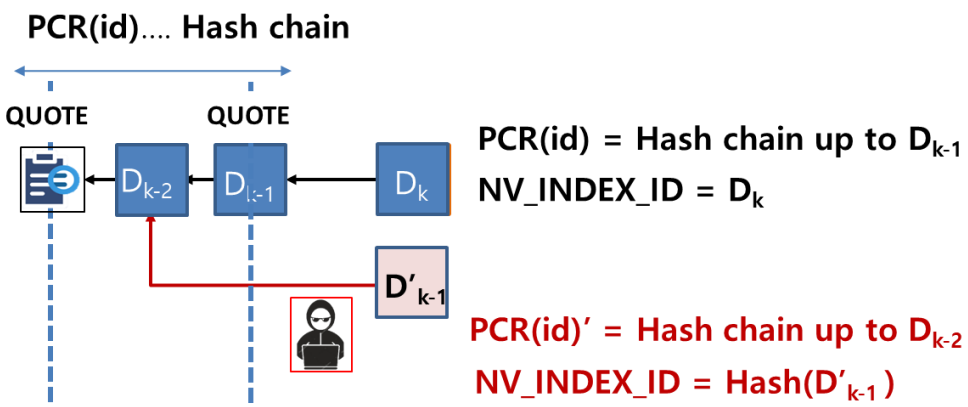

Figure 4.5. 악의적인 사용자에 의한 데이터 순서 변경 예시.

이렇게 무결성 정보를 포함한 Data point 를 Extended Data point (Figure 4.6)라고 정의한다. 이를 기존 Data point 와 비교하여 보면 ID 로는 이전 Data point 의 hash 값을 사용하며, QUOTE 증명 결과와 AIK 에 대한 증명이 포함되어 있다.

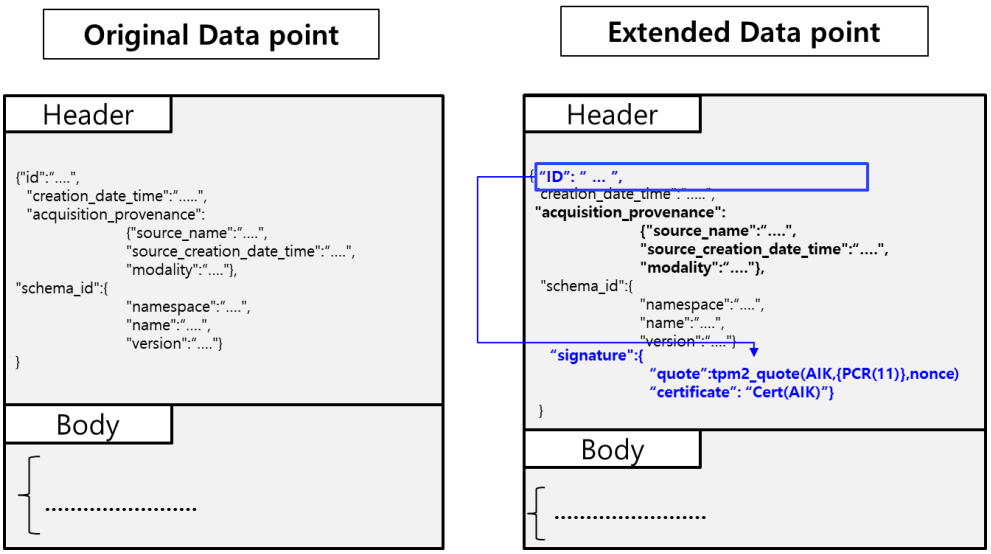

Figure 4.6. 기존 Data point 와 Extended Data point 비교.

4.5.3 Data point 저장 단계

Data point 저장 단계에서는 무결성 검증 정보 생성 단계에서 생성된 Extended Data point 를 DSU 에 저장한다. 데이터 소비자는 데이터 필요 시 DSU 에서 데이터를 얻어와 저장된 데이터를 사용할 수 있다.

4.5.4 무결성 정보 검증 단계

무결성 정보 검증 단계는 데이터 소비자가 Data point 를 사용하기 위해 Data point 의 무결성을 검증하는 단계이다. DSU 로부터 Data point 를 얻어온 후 데이터가 안전한지 판단하기 위하여 Data point 에 포함되어 있는 데이터 출처 정보와 데이터 순서에 대한 증명을 검증한다.

(1) **데이터 출처 정보 검증** 데이터 출처 정보에 대한 검증(Algorithm 2)은 Data point 의 header 에서 PCR 12 에 대한 QUOTE 증명과 AIK 에 대한 증명을 얻어온다. AIK 에 대한 인증을 통해 증명에 사용되는 AIK_{pub} 가 TPM 에서 합법적으로 생성되었는지 확인한다. 인증된 AIK_{pub} 로 QUOTE 증명을 검증한다. QUOTE 증명을 검증하기 위해서 데이터 출처 정보를 이용하여 새로운 QUOTE 메시지를 생성해내며, 새로운 QUOTE 메시지와 signature 값을 AIK_{pub} 로 검증한다.

Algorithm 2 데이터 출처 무결성 검증

1: get QUOTE and cert(AIK) from Data point header

2: check cert(AIK) // to verify AIK_{pub} is legally issued from TPM

3: verify QUOTE // by using certificated AIK_{pub}

3-1: make new QUOTE message with acquisition provenance

3-2: verify signature of QUOTE message with new QUOTE message // by using AIK_{pub}

(2) **데이터 순서 검증** 데이터 순서를 검증(Algorithm 3)하기 위하여 우선 DSU 로부터 Data point set 을 얻어온다. 얻어온 Data point set 에서 이전 Data point 의 hash 값이 다음 Data point 의 ID 와 일치하는지 확인한다. Data point 의 ID 일치 여부를 확인 한 후, 연속해 있는 Data point 내 QUOTE 증명을 데이터 출처 정보 검증 시와 같은 방법으로 검증한다. 이때, 다른 점은 TPM monotonic counter 값을 nonce 값으로 사용하고 있다는 점이다. 증명 시 인접해 있는 QUOTE 증명의 nonce 값이 연속적을 증가하면 데이터 순서가 정확하게 이루어져 있다고 판단할 수 있다.

Algorithm 3 데이터 순서 무결성 검증

1: get Data point set from DSU
2: check ID(D_k) == hash(D_{k-1})
3: verify consecutive QUOTE // by using certificated AIK_{pub}
    3-1: make new QUOTE message with hash((D_k || D_{k-1} || .... || D_1)
    3-2: give nonce as TPM monotonic counter
    3-2: verify signature of QUOTE message with new QUOTE message // by using AIK_{pub}

V. 보안 분석

본 논문에서 제안한 무결성 검증 프로토콜을 통해 생성된 Extended Data point 는 데이터의 출처 정보를 신뢰할 수 있는지, 순서대로 데이터가 저장되었는지 데이터 소비자가 데이터를 사용하기 전 검증할 수 있어야 한다. 데이터 소비자가 각 요소를 어떻게 신뢰할 수 있는지 분석한다.

데이터 출처 정보는 데이터 소비자가 데이터 출처 정보를 신뢰할 수 있는지를 인증된 AIK 와 QUOTE 증명 검증을 통해 확인할 수 있다. IoT gateway 에서는 PCA 로부터 AIK 에 대한 인증을 받았다. 또한 데이터 출처 정보를 TPM 의 PCR 12 에 extend 하고 있고, QUOTE 명령어를 통해 PCR 12 에 대한 증명을 생성해 내기 때문에, 이를 통해서 데이터 출처 정보가 맞는지 확인 할 수 있다.

데이터 순서는 데이터 소비자가 Extended Data point 가 순차적으로 저장되었는지 확인한다. Extended Data point 생성 시 이전 Data point 의 hash 값을 ID 로 사용한다. 이전 모든 Extended Data point 의 hash 값은 PCR 11 에 extend 되어 있으며, PCR 11 대한 QUOTE 증명을 생성했다. 또한 QUOTE 증명 순서를 보장하기 위하여 TPM monotonic counter 값을 nonce 값으로 사용하였다. 따라서, Extended Data point 의 hash 체인을 재계산 하고 인증된 AIK 로 QUOTE 증명을 검증하여 연속된 QUOTE 증명의 nonce 값을 비교해봄으로써 데이터 저장 순서를 판별할 수 있다.

VI. 성능 평가

본 프로토콜은 각각 센서, TPM 이 설치되어 있는 IoT gateway, DSU, 데이터 소비자를 구성하여 진행되었다. 센서는 실제 센서를 사용하는 대신 Ubuntu 16.04 version 이 설치되어 있는 Desktop 에서 센서 데이터 생성기를 만들어 센서 데이터를 얻어오도록 하였다. IoT gateway 는 1.2GHz 쿼드 코어 ARM Cortex-A53 에 64bit CPU 를 사용하는 raspberry pi 3board 에 Ubuntu 16.04 version 을 탑재하였다. 또한 SanDisk Micro SD card 16GB 의 사양을 가진다. 이 board 에 OPTIGA TPM SLB 9670 TPM2.0 이 달려 있는 SPI interface 인 IRID9670TPM20LINUXTOBO1 를 설치하여 TPM 2.0 을 사용할 수 있도록 하였다. 또한 TCG 에서 제공하는 tpm2.0 library[13]와 tpm2.0 tools[14]를 설치하여 실제 TPM 명령어를 사용할 수 있도록 하였다. DSU 는 Ubuntu 16.04 Virtual box 에서 동작하며, 4GB RAM 을 갖는다. 데이터 소비자로는 동일 스펙의 또 다른 raspberry pi 3board 에 설치하여 실제 데이터를 저장되어 있는 곳에서 데이터 소비자가 검증할 수 있도록 검증해 두었다.

실험은 IoT gateway 에서의 초기화 단계, 무결성 검증 정보 생성 단계에서의 소요 시간과 기존 Data point 생성 시 소요 시간에 대해 측정 및 비교하였다. Data point 한 개 당 생성 시간은 1200 개의 데이터를 갖고 측정한 시간을 평균 낸 것이다. IoT gateway 에서의 초기화 단계에서 소요 시간(Table 6.1)은 모두 6.053 sec 이며, EK 생성 및 인증, AIK 생성 및 인증에 소요되는 시간을 포함한다. 여기서 EK 와 AIK 는 RSA 2048bit 기반이며, TCG profile 에 맞는 포맷으로 만들어진다. EK 는 TPM 제조사에서 제공하는 EK certificate hosting server 에서 인증을 얻어오며, AIK 는 PCA 에서 인증을 얻어온다. 또한 초기화 단계에서 EK 인증 얻어오는데 가장 많은 시간이 소요된다.

Get EK	Create EK certificate	Generate AIK	Create AIK certificate	Total
0.428	2.338	1.504	1.783	6.053

Table 6.1 초기화 단계에서 key 생성과 인증 생성 시간 (단위: sec): TPM 에서 EK 를 얻고 AIK 를 생성하며 이에 대한 인증을 얻는 동작임.

무결성 검증 정보 생성 단계에서는 크게 NV memory 에 저장 된 값을 읽어오는 동작, 데이터 출처 정보를 PCR 12 에 extend 하는 동작, QUOTE 증명, TPM monotonic counter 읽고 증가, 데이터로 Extended Data point 작성, 생성된 Extended Data point 의 hash 값 계산, 이 hash 값을 PCR 11 로 extend 하는 동작, 생성된 Extended Data point 의 hash 값과 업데이트 된 PCR 11 값을 NV memory 에 저장하는 동작으로 나눌 수 있다. 데이터 출처 정보를 PCR 12 에 extend 하여 QUOTE 증명하는 동작은 맨 처음에만 포함되며, 이를 제외한 총 소요 시간은 1.866 sec 이다. 이때, QUOTE 증명에 가장 많은 시간이 소요된다(Table 6.2).

Create QUOTE for PCR(12)	Read from NV memory	Create QUOTE for PCR(11)	Read and Increase TPM monotonic counter	Write Extended Data point	Create hash of Extended Data point	Extend hash value to PCR(11)	Write in NV memory	Total
1.033	0.139	1.034	0.289	0.153	0.024	0.064	0.163	1.866

Table 6.2 Extended Data point 작성 시 각 동작 별 소요 시간 (단위: sec).

또한 모바일 건강 데이터를 이용하여 기존 Data point 생성 시 소요 시간에 대하여 측정해 보았다(Table 6.3). 소요 시간은 총 0.103sec 이며, 데이터를 읽는데 걸리는 시간과 읽은 데이터를 갖고 Data point 를 작성하는데 걸리는 시간을 포함한다. 이 때, 읽은 데이터로 Data point 작성하는데 대부분의 시간이 소요된다..

Read raw sensor Data	Write Data point	Total
0.008	0.095	0.103

Table 6.3 기존 Data point 작성 시 소요 시간 (단위: sec): 데이터를 읽고 Data point 작성하는 동작임.

기존 Data point 와 Extended Data point 의 한 개당 생성 시간을 비교해 보았다. 1 개의 기존 Data point 는 생성 시 0.103 sec 가 소요 되지만, Extended Data point 는 1 개 생성 시 1.866 sec 가 소요 된다 (Table 6.4).

Original Data point generation time	Extended Data point generation time
0.103	1.866

Table 6.4 기존 Data point 생성 시간과 Extended Data point 생성 시간 비교 (단위: sec)

성능을 개선하기 위하여 상대적으로 소요 시간이 길었던 QUOTE 증명과 이에 따른 TPM monotonic counter 증가, NV memory 에 값을 쓰고 읽는 동작을 일정 주기를 잡고 주기 별로 실행 해 보았다 (Figure 6.1). 각 주기는 데이터 개수로 나타나 있으며, 1200 개의 데이터를 가지고 데이터 개수 1 개, 2 개, 50 개, 100 개, 200 개, 300 개, 600 개 마다 위의 4 가지 동작을 수행하도록 하여 Extended Data point 를 생성하였다. 각 경우마다 1200 개 데이터를 이용하여 Extended Data point 1200 개 생성한 시간을 측정하여, 1 개의 Extended Data point 생성 시 평균 소요 시간을 계산하여 이를 기존 Data point 생성 시간과 비교해 보았다. 위 4 가지 동작을 매 번 실행하였을 때 Extended Data point 1 개당 생성 시간이 1.866 sec 였지만, 이 주기를 늘려가며 실험을 해 보았을 경우 약 0.243 sec 까지 줄어들며, 성능을 개선 시킬 수 있었음을 확인 해 볼 수 있었다.

또한 위 4 가지 동작을 주기를 늘려가며 실험해 보았을 경우 1 초당 Data point 의 생성 개수를 측정해 보았다(Figure 6.2). 기존 Data point 는 1 초에 약 10 개가 생성되며, Extended Data point 는 1 초에 최대 약 4 개까지 생성된다.

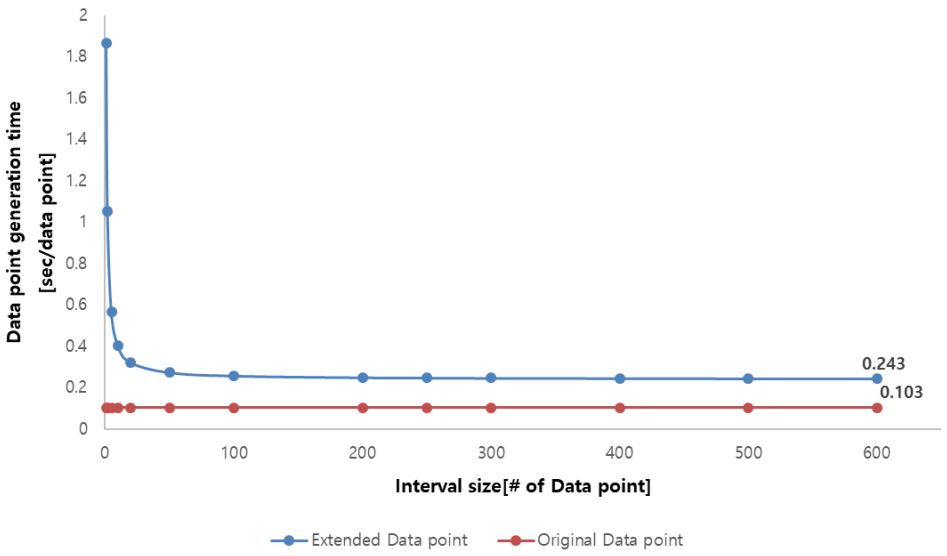

Figure 6.1 Data point 생성 시 QUOTE 증명, TPM monotonic counter, NV read & write 동작을 각 주기 별로 실행하였을 때 Data point 1 개당 생성 시간

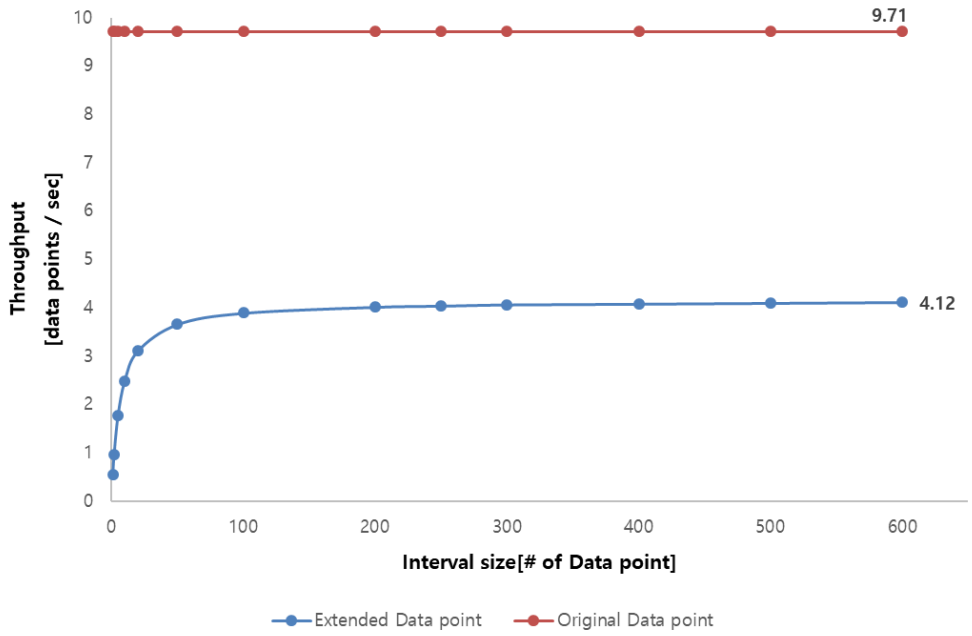

Figure 6.2 Data point 생성 시 QUOTE 증명, TPM monotonic counter, NV read & write 동작을 각 주기 별로 실행하였을 때 1 초당 Data point 생성 개수

VII. 한계점

본 논문에서 제안한 프로토콜은 3 가지 한계점이 있다.

높은 오버 헤드 제안한 프로토콜은 TPM 2.0 을 사용하여 TPM NV 메모리에 Extended Data point 의 hash 값과 현재 TPM 의 PCR 11 의 값을 저장하였다. 또한 저장된 값들을 NV 메모리에서 읽어와야 하며 PCR 11 에는 Data point 의 hash 정보, PCR12 에는 데이터 출처 정보가 저장되어 있다. 이 PCR 에 대한 QUOTE 증명도 생성해야 한다. 따라서 Extended Data point 생성 시 기존 Data point 생성 할 때와 비교하여 약 2~20 배 정도의 오버 헤드가 발생한다.

Data point 순서 공격 가능성 Extended Data point 생성 시 성능을 향상시키기 위하여 주기적으로 QUOTE 증명을 실행하였다. 매번 QUOTE 증명을 실행 할 시 연속된 Data point 의 QUOTE 증명 후 nonce 값을 비교함으로써 Data point 순서의 무결성을 검증할 수 있다. 하지만 매번 QUOTE 증명을 실행하지 않고 주기적으로 QUOTE 증명을 실행할 시 두 가지 경우의 공격 가능성이 존재한다. 첫 번째 QUOTE 증명 전에 Data point 순서 공격이 발생 할 경우, 이 공격을 예방하지 못하고 발생했다는 것을 알 수 없다. 두 번째, QUOTE 증명 후, QUOTE 증명 전 Data point 로 Data point ID 를 변경하는 공격이 발생하면, 이는 연속된 QUOTE 증명을 검증하고 nonce 값을 비교함으로써 공격을 감지할 수 있다.

Data point 삭제 공격 가능성 DSU 에서는 악의적인 사용자에 의해 저장된 Extended Data point 가 삭제될 수 있다. 하지만 본 논문에서 제안한 무결성 검증 프로토콜에서는 삭제된 이후의 Extended Data point 의 무결성을 검증할 수 없으며, 삭제된 이후의 Extended Data point 는 신뢰할 수 없는 Extended Data point 로 간주된다.

VIII. Conclusion

현대인들의 건강에 대한 관심이 늘고 있으며, 이에 따라 모바일 기기에 있는 센서에서 각 건강 정보를 측정하여 관리하는 경우가 늘어나고 있다. 이렇게 생성된 모바일 건강 데이터는 사용자가 관리하기 쉬워야 하며, 이에 대한 무결성을 판별할 수 있어야 한다.

특히, 악의적인 사용자가 강제적으로 건강 데이터가 저장되어 있는 곳을 공격하여 측정된 건강 데이터에 대한 정보가 변경될 가능성이 있다. 그러므로 모바일 건강 데이터를 신뢰하기 위하여 무결성 검증 프로토콜이 필요하다.

본 논문에서는 Trusted Platform Module(TPM) 2.0 을 이용하여 모바일 건강 데이터에 대한 무결성 검증 프로토콜을 제안하다. 이를 통해 데이터 사용자는 DSU 에 저장된 모바일 건강 데이터를 사용할 때, 무결성을 검증하여 저장된 모바일 건강데이터가 신뢰할 수 있는 데이터인지 검증할 수 있다. 모바일 건강 데이터 무결성 검증 프로토콜을 위해 보안 모듈인 TPM 을 사용하여 무결성 검증 프로토콜을 디자인 하였으며, 2 가지 요소인 데이터 출처 정보, 데이터 순서 정보에 대한 신뢰성을 검증할 수 있다.

프로토타입은 IoT gateway 를 Raspberry pi 3 board 에서 Ubuntu 16.04 기반으로 하여 구현하였으며, 센서 데이터는 Ubuntu 16.04 기반의 Desktop 에서 모바일 건강 데이터 생성기를 구성하여 얻었다. 또한 DSU 는 Ubuntu 16.04 Virtual Box 에서 구성하였으며, 데이터 소비자는 Raspberry pi 3 board 에서 Raspbian 기반으로 구성하였다. IoT gateway 와 데이터 소비자에는 TPM 2.0 을 설치하였으며, TCG 에서 제공하는 tpm 2.0 library[13]와 tools[14]를 사용했다. 초기화 단계, 무결성 검증 정보 생성 단계, Extended Data point 저장 단계, 무결성 정보 검증 단계로 구성하여 디자인을 하였으며 IoT gateway 에서 초기화 단계,

무결성 검증 정보 생성 단계에 대해 동작 과정별로 성능을 측정했다. 또한 성능을 향상시키기 위하여 QUOTE 증명, TPM monotonic counter 증가, NV 메모리 read & write 과정을 주기적으로 실행하였으며, 주기적에 따른 성능 평가를 진행하였다.

추후 모바일 건강 데이터뿐만 아니라, 병원에서 저장되는 건강 데이터에 대한 무결성 검증 프로토콜로 확장할 수 있으며, 데이터 순서 공격 및 삭제 공격에 대한 보완이 이루어 진다면 신뢰할 수 있는 건강 데이터를 사용할 수 있다.

참 고 문 헌

[1] Open mHealth. “http://www.openmhealth.org/features/features-overview/”

[2] Security in Smartphone health App. “http://www.itworld.co.kr/news/100485”.

[3] Richard T. Snodgrass Shilong Stanley Yao Christian Collberg, Tamper Detection in Audit Logs. In VLDB. 2004.

[4] Ramakrishna Kotla, Pasture: Secure Offline Data Access Using Commodity Trusted Hardware. In *presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation* (OSDI 12), pages 321-334, 2012.

[5] Jia L. England P. Lorch J. R. Sinha A, Continuous Tamper-proof Logging Using TPM 2.0. In *International Conference on Trust and Trustworthy Computing*, 2014.

[6] Trusted Computing Group. TPM platform module library part 1: Architecture. In f*amily 2.0, Level 00 Revision 01.38*, 2016.

[7] Trusted Computing Group. Trusted Platform Module Library Part 2: Structures. *In family 2.0, Level 00 Revision 01.38*, 2016.

[8] Trusted Computing Group. TPM platform module library part 3: command. In *family 2.0, Level 00 Revision 01.38*, 2016.

[9] Attestaion. “https://shazkhan.files.wordpress.com/2010/10/http__www-trust-rub-de_media_ei_lehrmaterialien_trusted-computing_exercise-attestation.pdf”.

[10] Open mHealth Data point. “http://www.openmhealth.org/documentation/#/schema-docs/schema-library/schemas

[11] Open mHealth Data point. “https://github.com/smalldatalab/omh-shims”.

[12] Trusted Computing Group. TCG Infrastructure Working Group A CMC Profile for AIK Certificate Enrollment. Version 1.0, Revision 7, 2011.

[13] tpm2-software. “https://github.com/tpm2-software/tpm2-tools”.

[14] tpm2-software. “https://github.com/tpm2-software/tpm2-tss”.

Acknowledgements

포항에서 짧으면서도 긴 2 년이라는 기간이 흘렀습니다. 석사 생활 2 년동안 정말 많은 일들이 있었지만, 주변에 계신 많은 분들의 응원과 도움 덕분에 별탈 없이 연구를 끝마칠 수 있었습니다.

우선 많이 부족한 저를 2 년동안 끝까지 열심히 지도해주신 박찬익 교수님께 감사의 말씀을 전하고 싶습니다. 교수님의 지도 덕분에 많은 것을 배우고, 연구를 진행 할 수 있었습니다. 또한 제 논문에 관심과 도움을 주신 김종 교수님과 김광선 교수님께 감사의 인사를 드립니다.

그동안 함께 생활했던 연구실 선후배 분들께도 감사 인사를 전하고 싶습니다. 우선, 제가 연구를 진행하는데 많이 알려주시고 도움을 주셨던 운성 오빠, Hieu 에게 감사하다고 말씀드리고 싶습니다. 짧은 시간이지만 같이 보냈던 백제 오빠, 유준 오빠, 영섭 오빠께도 감사인사 드립니다. 항상 멀리서 응원해주시고 많이 의지할 수 있었던 민경 언니께도 감사드립니다. 또, 많은 것을 배울 수 있었던 정현 오빠, 항상 친절하게 말씀해 주시는 윤봉 박사님, 부족한 저를 많이 도와주셨던 지은 언니, 동기 용래 오빠, 분위기 메이커 연규 오빠, 유일한 친구 동민이, 말동무가 되어 주던 융두 오빠께도 감사드립니다.

다른 랩 선배, 친구분들께도 감사인사 드리고 싶습니다. 제가 힘들 때 고민을 들어 주시던 동일 오빠, 포항 생활 하면서 같이 이런저런 얘기하고 서로 의지 할 수 있었던 정빈 언니와 채현이, 밤에 불빛과 키보드 소리로 힘들었을 룸메이트 선영이 에게도 감사 인사 드립니다.

마지막으로 항상 절 믿고 격려 해주고 응원해 주시는 가족, 부모님과 제 동생한테도 감사 말씀 전하고 싶습니다. 사랑합니다.

Curriculum Vitae

Name : Mirae Lim

Education

2013. 3. ~ 2017. 2.	School of Electronic Engineering, Soongsil University (B.S.)
2017. 2. ~ 2019. 2.	Department of Computer Science and Engineering, Pohang University of Science and Technology (M.S.)