Intel SGX 동향 및 병원 정보 시스템 제안

김경호* 최승주* 김현준* 서화정*
*한성대학교 IT융합공학부

Intel SGX Survey & Proposal of Hospital Information System

Kyung-Ho Kim* Seung-Ju Choi* Hyun-Jun Kim* Hwa-Jeong Seo*
*Division of IT convergence Engineering, Hansung University.

요 약

최근 개인 정보에 대한 해킹이 해마다 증가하면서 개인 정보 보호에 대한 중요도가 높아지고 있다. 특히 의료 정보와 같은 민감한 정보는 다른 개인 정보에 비하여 희소성과 활용성이 높기 때문에 해커의 주요 타겟이 된다. 현재 대부분의 병원 정보 시스템(HIS; Hospital Information System)은 사용자 부주의로 인한 정보 유출과 병원 네트워크 내부의 해킹에 취약하다는 문제점을 가진다. 본 논문에서는 사용자의 부주의로 인한 정보 유출 및 해커의 공격을 방어하기 위한 보안 플랫폼으로 TEE(Trusted Execution Environment) 기술을 소개한다. 특히 TEE 플랫폼 중 Intel에서 개발한 SGX(Software Guard eXtension)의 동작 메커니즘 설명과 이를 이용한 안전한 병원 정보 시스템 구축을 제안한다.

I. 서론

인터넷 기술의 발전으로 사용자 인증이나 사용자에 맞는 서비스를 제공할 때 개인 정보가 사용되면서 개인 정보 보호에 대한 중요도가 높아지고 있다. 특히 환자의 건강과 관련된 의료 정보는 다른 개인 정보에 비하여 매우 민감한 정보이고 한번 유출되면 원상회복이 어려운 특징이 있다.

하지만 국내 다수의 의료기관에서 환자의 의료 기록이 유출되는 사례가 해마다 발생하고 있다. 이러한 유출 사례의 유형으로는 데이터를 관리하고 있는 중앙 서버 및 관리자 PC가 해커에 의해 공격 받아서 환자의 개인 정보가 유출되거나 병원 내부 관계자가 환자 의료 기록 유출에 대한 심각성을 인지하지 못하고 개인 메신저나 메일로 의료 기록을 유출하는 사례가 대표적이다.[1]

본 논문에서는 이러한 환자 의료 정보 유출을 막기 위해서 기존의 시스템에 새로운 보안 기술인 TEE 기술을 접목한 새로운 병원 정보 시스템 구축을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 TEE 기술에 대해서 설명하고 3장에서는 TEE 기술 중 하나인 인텔 SGX의 메커니즘을 설명한다. 4장에서는 인텔 SGX를 이용하여 안전한 병원 정보 시스템 구축을 제안한다. 마지막으로 5장에서는 본 논문의 결론을 내린다.

II. TEE

TEE[2]는 기존의 SE(Secure Elements)의 한정적인 저장 공간 및 느린 처리 속도를 보완하고 TPM(Trusted Platform Module)[3], TXT(Trusted eXecution Technology)[4]의 인증 과정에서 존재하는 단점을 보완한 보안 기술이다.

TEE는 Secure World와 Non-Secure World

혹은 Trust Zone과 Untrust Zone으로 실행 환경을 분리해서 관리한다. Secure World에는 프로그램 실행 중에 민감한 데이터들을 저장하고 나머지 데이터는 Non-Secure World에 저장하여 정해진 API 함수를 이용해야만 분리된 실행 환경 사이의 통신이 허용된다.

또한 Secure World는 하드웨어적으로 분리되어 운영체제를 포함한 모든 권한의 소프트웨어가 직접 접근이 불가능하기 때문에 시스템을 관리하는 시스템 소프트웨어가 해킹에 의해 시스템 제어권을 뺏기더라도 Secure World에 저장한 민감한 데이터의 무결성과 기밀성을 보장할 수 있다.

TEE 기술은 삼성의 삼성페이 같은 다양한 결제 시스템에서 이미 사용하고 있고 생체 데이터를 이용하는 FIDO[5]와 결합하거나 사용자의 중요한 데이터를 안전하게 보호하는 시스템 설계 등 민감한 데이터가 사용되는 다양한 부분에서 사용될 수 있다.

TEE는 구현 방식에 따라서 사용 방법이 차이가 나는데 대표적인 TEE 플랫폼으로는 인텔의 SGX와 ARM의 TrustZone, AMD의 PSP(Platform Security Processor)가 있다.

III. Intel SGX

Intel SGX는 인텔에서 2015년에 소개된 6세대 인텔 코어 마이크로프로세서인 스카이레이크 모델에서 처음 등장한 TEE 기반 플랫폼이다.

SGX는 Trust Zone과 Untrust Zone으로 실행 환경을 분리해서 관리하고 Enclave로 불리는 분리된 환경(Secure Container)을 제공하여 시스템 소프트웨어를 포함한 다른 응용 프로그램의 접근을 제어한다. 또한 키 유도 방식을 이용해 상황에 맞는 키를 유도하고 암호화 및 복호화함으로써 보안성을 유지한다. 그리고 증명 메커니즘을 사용하여 다른 서버와 민감한 데이터를 안전하게 전달할 수 있다.

본 논문에서 기술한 SGX 관련 지식은 인텔 개발자 매뉴얼[6], SGX 관련 논문[7]에서 나오는 정보를 종합하여 소개한다.

3.1. Memory Architecture

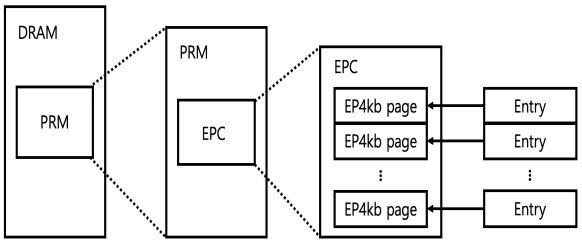

그림 1. Intel SGX 메모리 구조

SGX는 PRM(Processor Reserved Memory)이라고 불리는 DRAM의 특정 부분을 분리하여 관리하며 물리적 주소 직접 접근 공격을 막기 위해서 해당 부분을 AES 알고리즘으로 암호화하여 기밀성, 무결성을 보장한다. PRM 내부에는 4KB 페이지 단위로 저장되는 페이지의 집합인 EPC(Enclave Page Cache)가 존재한다.

SGX는 기존의 시스템 소프트웨어가 사용하는 주소 변환 방식을 그대로 사용하여 4KB의 Page 단위로 EPC 페이지를 관리한다. 하지만 시스템 소프트웨어는 신뢰하는 소프트웨어가 아니기 때문에 EPC 페이지 당 1개의 EPCM(Enclave Page Cache Map)이라는 공간을 두어 시스템 소프트웨어가 제대로 된 페이지 할당 및 해제를 할 수 있도록 보장한다.

SGX는 위에서 설명한 다양한 구조들 내부의 값을 이용해서 분리된 환경에 대해 다른 소프트웨어의 물리적, 논리적 접근을 모두 차단한다. 그리고 기존 시스템과 동일한 주소 변환 방식을 사용함으로써 데이터 이동에 부가적인 오버헤드가 발생하지 않는다. 전체적인 SGX 메모리 구조는 그림 1과 같다.

3.2 Enclave 동작 과정

Enclave는 그림 2에 있는 명령어를 이용하여 생성 및 해제 그리고 코드를 실행한다. Enclave는 ECREATE 명령어를 이용하여 Enclave의 SECS 페이지 할당이 끝난 후 바로 Enclave 코드를 실행 할 수 있는게 아니라 EADD 명령어로 초기 데이터와 코드를 로드하고 EINIT 명령어를

사용해 인텔의 정해진 보안 메커니즘에 따른 초기화 과정을 거쳐야 Enclave 코드를 실행할 수 있다.

Enclave 코드는 시스템 권한이 아닌 응용 프로그램 권한의 Enclave 모드에서만 동작하도록 설계 되었다. 따라서 EENTER 명령어를 사용하여 응용프로그램을 Enclave 모드로 변환하고 프로세서가 안전하게 Enclave 코드로 분기할 수 있도록 하여 보안성을 유지한다. 코드 실행이 끝나면 EEXIT 명령어를 사용하여 Enclave 모드를 종료한다.

Enclave 코드 및 데이터에는 민감한 비밀이 저장되기 때문에 코드 실행 중에 하드웨어 예외처리가 발생할 경우 기존의 데이터를 메모리에 저장하고 예외 처리기로 분기하는 방식을 그대로 사용할 수 없다. 따라서 AEX(Asynchronous Enclave Exit) 명령어를 사용하여 수행중인 Enclave의 비밀 데이터들을 EPC 내부의 SSA(The State Save Area) 공간에 저장한 후 Enclave 모드를 빠져나와서 기존의 예외 처리기를 호출하는 방식을 이용한다.

예외 처리기가 끝난 후에는 ERESUME 명령어를 실행하도록 설정하여 AEX 명령어 실행 시 SSA에 저장한 데이터를 복원함으로써 기존의 코드 수행 흐름으로 돌아오도록 하여 보안성을 유지한다.

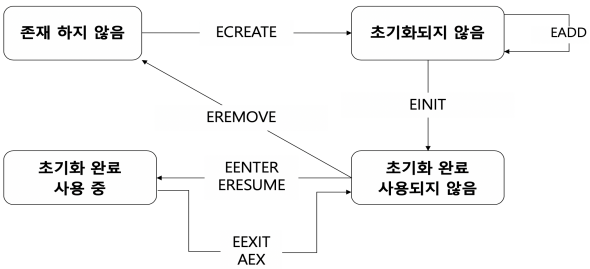

그림 2. Intel SGX Life Cycle

3.3 증명

Enclave는 같은 플랫폼 내부의 다른 Enclave의 접근도 제한하기 때문에 다른 Enclave와 데이터 교환을 하기 위해서 지역 증명을 통해 자신을 증명한다. 또한 같은 플랫폼이 아닌 다른 외부의 플랫폼과 안전한 통신을 위해 본인을 인증하는 과정을 원격 증명이라고 한다.

우선 지역 증명은 Enclave가 자기 자신을 증명하기위해 Enclave가 생성되는 명령어 실행 과정에서 Enclave의 특정한 값들을 이용하여 SHA-2 해쉬 알고리즘을 사용해 생성한 측정값(Measurement)을 만드는데 해당 값은 Enclave의 메타 데이터를 저장하는 EPC Page인 SECS(SGX Enclave Control Structure)의 MRENCLAVE 필드에 저장된다.

지역 증명의 구조는 그림 3과 같다. 우선 요청할 Enclave의 MRENCLAVE 값을 받아온 뒤 현재 Enclave의 SECS 구조를 상대 MRENCLAVE 값을 인자로 한 인텔 SGX의 키 유도 방식을 통해 유도된 키로 암호화 한 뒤 상대 Enclave로 전송해서 동일한 방법으로 키를 유도하여 복호화 후 증명을 성공한다.

원격 증명의 경우 키 유도 방식을 통해 Provisioning Key를 유도한 뒤 해당 키로 인텔의 Provisioning 서비스에 자신을 인증한다. Provisioning 서비스가 신뢰 할 수 있는 환경이라고 인증을 하게 되면 안전한 인증키를 전달 받고 해당 키를 이용해 지역 증명에서 사용한 REPORT의 MAC을 증명키로 서명을 한 뒤 원격 서버에게 전달한다. 원격 서버는 해당 정보를 공개키로 복호화하여 서명을 확인함으로써 원격 증명을 성공한다.

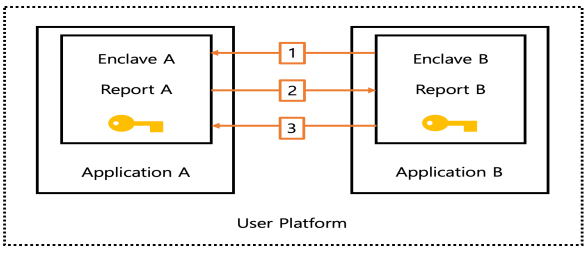

그림 3. Intel SGX Local Attestation

IV. 병원 정보 시스템 제안

병원 정보 시스템은 환자의 개인 정보뿐만 아니라 병원 전체의 데이터를 관리하는 통합 서비스 환경이다. 기존의 병원 정보 시스템은 병원의

규모 및 환경에 따라 주요 기능을 모듈로 나눠서 개방형 클라우드 플랫폼인 PaaS(Platform as a Service) 환경을 사용한다.

기존의 시스템은 중앙 서버에 모든 환자의 의료 정보가 저장되기 때문에 해커에 의해 서버의 제어권을 탈취당하는 경우 의료 정보에 대한 무결성 및 기밀성을 유지할 수 없다. 그리고 의료 정보를 다룰 수 있는 의료 관계자가 환자의 개인정보 파일을 메신저나 메일로 불특정 다수에게 보내는 것에 대한 기술적 방어 체계가 없다. 기존의 병원 정보 시스템에 Intel SGX를 적용함으로서 다음과 같은 취약점을 보완할 수 있다.

SGX는 Enclave를 활용하여 중요한 데이터를 저장하고 데이터 처리 과정을 안전하게 보호한다, 이는 공격자가 시스템의 취약점을 이용하여 내부 메모리의 값을 확인하거나 변수에 저장된 중요한 데이터를 확인할 수 없다. 또한 씰링을 이용하여 Enclave에서 처리된 이후 정해진 키 유도 방식에 의해 유도된 Seal Key를 이용하여 데이터를 암호화한 후 디스크에 저장한다. 이에 공격자가 운영체제와 같은 높은 권한의 시스템 소프트웨어의 시스템 제어권을 탈취해서 서버에 저장된 환자 정보에 접근을 하더라도 암호화된 데이터 밖에 읽을 수 없다. 이 방법을 이용하면 Enclave에서 처리가 끝나고 디스크에 저장된 환자 의료 정보 파일을 의료 관계자가 불특정 다수에게 전송하더라도 암호화된 정보이기 때문에 기밀성을 유지할 수 있다.

더 나아가 SGX는 증명을 통해 서버와 병원 내의 PC 및 동일한 PC에서 실행되는 응용 프로그램 내 환자 정보를 이용하는 다양한 서비스 간에 안전한 신뢰 관계를 구축할 수 있다. 이는 공격자가 중앙 서버와 클라이언트 PC 사이의 환자 개인 정보를 전달하는 과정에서 프로토콜 조작과 같은 올바르지 않은 방법으로 통신 과정에 접근할 수 없도록 유도한다. 또한 같은 PC에서 다수의 Enclave를 생성하여 응용프로그램 내에서 안전한 신뢰 관계 구축 후 서로 통신하기 때문에 PC 내부의 취약점을 이용한 통신 중인 정보 해킹을 막을 수 있다.

V.결론

본 논문에서는 인텔 SGX에 대한 동향과 이를 활용하여 병원 정보 시스템을 보완한다. 기존의 시스템에 존재하던 취약점들을 인텔 SGX의 분리된 실행환경을 사용하여 보완하였다. 앞으로의 연구 방향은 논문에서 제안한 병원 정보 시스템을 구체화 시켜 안전한 실행 환경 구현을 제시하는데 있다.

[참고문헌]

[1] “서울대병원, 환자 정보 무단 열람에 유출까지…처벌 수위는?”, KBS news, 2018.11.09. “https://news.kbs.co.kr/news/view.do?ncd=4070088“

[2] M. Sabt, M. Achemlal, and A. Bouabdallah,“Trusted execution environment: what it is,and what it is not,” 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, Aug. 2015.

[3] Trusted Computing Group. Tpm main specification. http://www.trustedcomputinggroup.org/resources/tpm_main_specification, 2003.

[4] David Grawrock. “Dynamics of a Trusted Platform: A building block approach. Intel Press“, 2009.

[5] Rob Philpott, Sampath Srinivas, John Kemp,UAF Architectural Overview. Version1.0-rd-20140209, FIDO Alliance, February2014

[6] “Intel Corporation. Intel 64 and IA-32 Architectures Software Developer's Manua“l, Reference no. 325462-056US.Sep 2015

[7] Victor Costan, Srinivas Devadas. “Intel SGX Explained“, IACR Cryptology e Print Archive 2016