2011 대한임베디드공학회 추계학술대회(2011년 11월 11~12일) 제주 샤인빌리조트

(사) 대 한 임 베 디 드 공 학 회
2011년도 추계학술대회 학술발표 논문집

2011 대한임베디드공학 추계학술대회

▣ 일 시 : 2011. 11. 11 ~ 12
▣ 장 소 : 제주 샤인빌리조트
▣ 주 최 : (사)대한임베디드공학회
▣ 주 관 : 대구경북과학기술원,대구디지털산업진흥원,
(재)경북IT융합산업기술원, (재) 경북차량용임베디드기술연구원
정보통신연구소, Wi-Media 지역혁신센터(RIC)
▣ 후 원 : 한국과학기술단체총연합회, 삼성전자, 하이버스

(사) 대한임베디드공학회
Institure of Embedded Engineering of Korea

Presentation Session

2011. 11. 12 (토)

Track C-1 임베디드시스템 응용 - 바이올렛(신관2층)

좌 장 : 변태영 교수 (대구가톨릭대)

09:00~09:20	제목 : 근거리 무선 통신 시스템을 위한 2.4GHz RF Wake-up 시스템 구현 저자 : 정태현, 신종욱, 박준홍, 김성률, 서대화(경북대학교 임베디드소프트웨어연구센터)
09:20~09:40	제목 : IEEE 802.15.4 기반 전자 가격표시 시스템의 구현 저자 : 배정규, 이우영, 양은주, 송정훈, 김성률, 서대화(경북대학교 임베디드 소프트웨어연구센터)
09:40~10:00	제목 : 무선 센서 네트워크 기반의 무기 자산관리 시스템 저자 : 이승일, 윤경효, 이진영(경북대학교 임베디드소프트웨어연구센터), 유창석(국방과학연구소), 하성기(퍼스텍), 김성률, 서대화(경북대학교 임베디드소프트웨어연구센터)
10:00~10:20	제목 : 보행자를 위한 GPS 위치보정 시스템 설계 및 구현 저자 : 송일선, 강보영, 서대화(경북대학교 임베디드소프트웨어연구센터)

Track C-2 Embedded Software - 빈카(신관2층)

좌 장 : 조두산 교수 (순천대)

09:00~09:20	제목 : 실시간 처리를 위한 리프팅 기반의 고속 정수 웨이블릿 변환의 FPGA 구현 저자 : 김석준, 장영조(한국기술교육대학교)
09:20~09:40	제목 : 호출 함수의 특성을 고려한 테스트 케이스 생성 기법 저자 : 신영술, 후세인 무하메드 입팔, 이우진(경북대학교)
09:40~10:00	제목 : WSN에서 클러스터 사이즈를 고려한 동적 클러스터링 방안 저자 : 오종하(대구가톨릭대학교), 김덕춘(대구과학고등학교), 변태영(대구가톨릭대학교)
10:00~10:20	제목 : PRAM을 내장한 플래시 변환 계층에서 효율적인 가비지 컬렉션 기법 저자 : 박지훈, 정승완, 서대화(경북대학교)

Track C-3 Embedded Hardware - 아잘리아(신관2층)

좌 장 : 임채덕 박사 (한국전자통신연구원)

09:00~09:20	제목 : TCG 기반 모바일 환경에서 TPM 에뮬레이터를 활용한 성능향상 기법 저자 : 신재복, 박우람, 박찬익(포항공과대학교)
09:20~09:40	제목 : u-Transportation 서비스를 위한 통신 시스템 구현 저자 : 이재정, 송정훈, 안태식, 박진홍, 김성률, 서대화(경북대학교 임베디드소프트웨어연구센터), 한동석(경북대학교)
09:40~10:00	제목 : TRIAC을 이용한 AC 모터의 PI 제어기 설계 저자 : 신동협, 권순태, 주문갑(부경대학교)
10:00~10:20	제목 : CPS를 위한 지식베이스 및 자율컴퓨팅 엔진의 설계 저자 : 이민영(과학기술연합대학원대학교, 한국전자통신연구원), 전인걸(한국전자통신연구원), 김동관(과학기술연합대학원대학교, 한국전자통신연구원), 김원태(한국전자통신연구원)

TCG 기반 모바일 환경에서 TPM 에뮬레이터를 활용한 성능향상 기법

Performance Enhancement Method using TPM Emulator in TCG-based Mobile Device

신 재 복*, 박 우 람, 박 찬 익
포항공과대학교 컴퓨터공학과
(Jae-Bok Shin, Woo-Ram Park, Chan-Ik Park)
(Department of Computer Science and Engineering, POSTECH)

Abstract : As mobile computing technologies are advanced and become popular nowadays, many computing tasks which are only processed by desktops or laptops can be processed by handheld mobile devices. Accordingly, it is necessary to provide mobile security environment for some jobs that require high security. That is, processing and storing important data in mobile device must be done in a safe environment.

For x86 system, there are many research to provide safe computing environment by using Trusted Platform Module(TPM) which is hardware security chip proposed by Trusted computing Group(TCG). For mobile device, same framework can be applied to provide key management and integrity verification of working device by using TPM. However using TPM generates additional overhead, so this causes delay of jobs and fast consumption of battery.

In this paper, we introduce our framework providing secure computing environment for mobile device by using TPM and describe out method to mitigate TPM overhead.

Keywords : Security, TPM,

I. 서 론

모바일 기기의 고성능화로 인하여 데스크탑 등 고정된 컴퓨팅 환경에서만 수행 가능하였던 작업들을 모바일 환경에서도 수행할 수 있게 되었다. 스마트 폰과 태블릿 등 고성능, 고 휴대성을 갖춘 모바일 기기가 대중화 됨에 따라 언제 어디서든 인터넷 검색, 문서 작업, 멀티미디어 감상 등이 가능하게 되었다. 모바일 기기와 PC의 기능, 성능 상 경계가 허물어지면서 모바일 기기가 PC를 점점 대체해 나가고 있고 다양한 사용자 응용프로그램에서 사용자 개인정보를 포함한 많은 양의 데이터가 저장, 처리되고 있다.

엔터프라이즈 환경 등 보안이 요구되는 컴퓨팅 환경에서도 이러한 변화에 따라 시간과 장소에 제약 없는 작업 환경 제공의 필요성이 나타나고 있다. 하지만 다수의 모바일 기기는 보안이 필요한 작업 환경을 제공해주기에는 다양한 보안 취약성을 가지고 있다[2][3]. 따라서 모바일 기기에서 보안성이 요구되는 작업을 수행하거나 기밀 데이터에 접근을 할 때 추가적인 보안 방법이 필요하다.

기존 x86 컴퓨팅 환경에서는 안전한 보안 환경을 제공하기 위해서 TCG (Trusted Computing Group)에서 제안한 보안 하드웨어 칩인 TPM

* 교신저자(Corresponding Author)
신재복, 박우람, 박찬익 : 포항공과대학교 컴퓨터공학과
※ "본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (NIPA-2011-C1090-1131-0009)

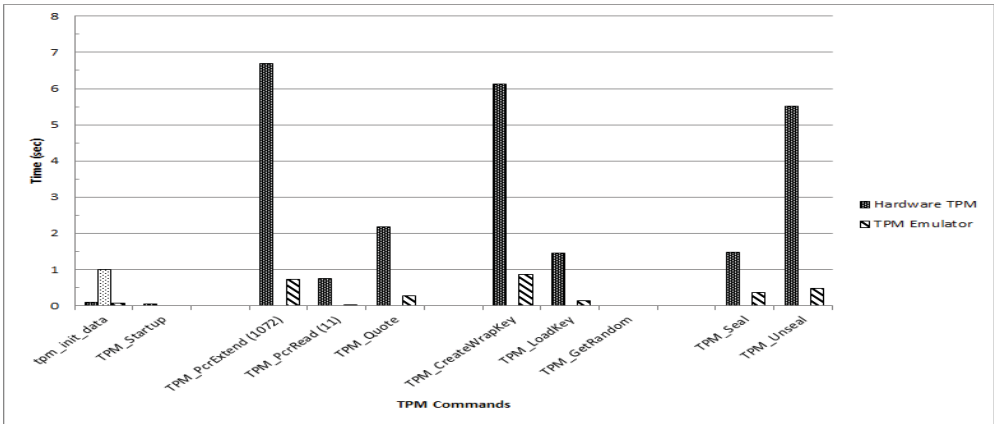

그림 1. 하드웨어 TPM과 에뮬레이터 커맨드 성능비교
Fig 1. Performance of H/W TPM and TPM Emulator Commands

(Trusted Platform Module)을 활용하고 있다. TPM을 기반으로 하여 작업을 수행하고자 하는 시스템의 무결성을 검증함으로써 초기에 구축된 안전한 실행환경을 제공하고, TPM의 하드웨어 키 관리 기능을 통한 데이터의 암호화/복호화를 이용하여 보안이 요구되는 데이터의 안전한 보관 및 접근을 보장하고 있다. 이 프레임워크를 모바일 환경에 적용하게 되면 모바일 환경에서도 기존과 동일하게 강력한 보안 환경을 제공할 수 있다.

하지만 모바일 기기에서 TPM을 사용하기 위해서는 해결해야 하는 문제점이 있다. 하드웨어 TPM은 수행 성능 및 전력소모 측면에서 단점을 가지고 있다[4]. <그림1>에서 볼 수 있듯이, TPM의 인증 및 키관리를 위해 사용되는 주요 커맨드들에 대한 수행시간은 긴 시간을 요구하고 있는 것을 알 수 있다. 이는 작업의 효율성과 사용자 편의성을 떨어뜨리고, 커맨드별로 요구되는 많은 전력소모양은 모바일 기기의 중요한 요소인 배터리 지속시간을 줄이는 결과가 된다.

따라서 본 논문에서는 TPM을 모바일 기기에서 사용 할 때 성능 상 오버헤드를 줄이면서 보안 작업을 안전하게 수행할 수 있는 프레임워크를 디자인하고 그 결과에 대해 논의한다.

II. 본 론

본 논문에서는 모바일 기기에 하드웨어 TPM을 사용하는 환경을 가정하였으며, 부트 과정에서 모바일 기기의 무결성을 검증한 뒤, 보안 스토리지를 사용하는 프레임워크를 디자인하였다. 하드웨어 TPM에서 발생하는 성능 오버헤드를 줄이기 위하여, 빈번하게 사용되는 커맨드에 대해 TPM 에뮬레이터[5]를 이용하였다. <그림 1>에서 볼 수 있듯이, TPM 에뮬레이터는 하드웨어 TPM에 비하여 커맨드별로 보다 좋은 성능을 보여주고 있음을 알 수 있다. 하지만 TPM 에뮬레이터는 하드웨어 TPM에 비하여 보안 상 취약점을 가지고 있기 때문에 TPM 키와 관련된 작업은 하드웨어 TPM을 이용하여 상호 보완하는 프레임워크를 구축하고자 한다.

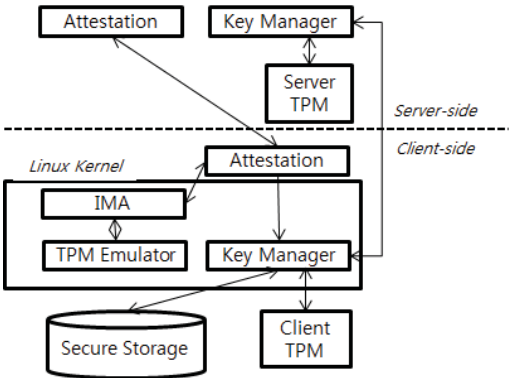

그림 2. 프레임워크 디자인
Fig 2. Framework Design

<그림 2>는 본 논문에서 제안하는 프레임워크의 전체적인 디자인을 나타낸다. 서버와 모바일 클라이언트 각각에 하드웨어 TPM이 장착되어 있고 키 마이그레션 및 보안 스토리지를 사용하기 위하여 키 관리 모듈이 커널에서 동작하고 있다. 키 관리 모듈

은 해당 스토리지에서 데이터 암호화/복호화를 위하여 사용하는 키를 서버로부터 전달받아 모바일 클라이언트의 하드웨어 TPM에 저장하고 실제로 보안 스토리지를 사용할 때, 저장된 키를 가져와서 사용한다.

모바일 클라이언트가 부팅되는 과정에 클라이언트 커널에서 IMA (Integrity Measurement Architecture)[6] 모듈이 동작하여 클라이언트의 무결성 검증 작업을 수행한다. 무결성 검증 데이터는 하드웨어 TPM의 오버헤드를 줄이기 위하여 TPM 에뮬레이터의 PCR (Platform Configuration Register)에 PCR extend 명령을 통해 기록된다. 보안 스토리지를 사용하기 위하여, 서버로부터 인증을 받고 보안 스토리지에 접근하기 위한 키를 받는 과정을 거쳐야 한다. 클라이언트의 인증 모듈은 TPM 에뮬레이터로부터 TPM Quote 명령을 통해 무결성 검증 데이터를 수집한 후, 서버의 검증 모듈로 데이터를 전송한다. 서버의 검증 모듈에 의해 클라이언트 무결성이 확인되면, 서버의 키관리 모듈을 이용하여 서버의 하드웨어 TPM에 보관되어 있는 보안 스토리지 접근 키를 클라이언트의 하드웨어 TPM으로 migration을 수행한다.

보안 스토리지 접근 키 관리를 TPM 에뮬레이터가 아닌, 하드웨어 TPM을 이용하는 이유는 메모리 접근을 통하여 키를 유출하거나 변경하는 등의 보안상 위협이 있기 때문이다. 하지만 <그림 1>에서 보듯이, TPM key migration에 사용되는 TPM 커맨드(TPM_CreateWrapKey, TPM_LoadKey, TPM_GetRandom 등)는 매우 큰 성능 상의 오버헤드를 가지고 있다. 이 오버헤드를 줄이기 위하여 본 프레임워크에서는 TPM Key Prefetching 기법을 사용한다.

클라이언트가 부팅되고 IMA가 동작하여 무결성 검증 결과를 서버로 보내기 전에 클라이언트는 자신의 스토리지 정보와 key migration 중 보안 스토리지 접근 키 암호화에 사용할 public/private 랜덤 키를 생성하여 서버로 public 키를 미리 전송한다. 서버는 이것을 전송받아 클라이언트가 보안 스토리지 접근에 사용할 키를 서버의 하드웨어 TPM으로부터 가지고 와서 클라이언트의 public 키로 암호화한다. 이 작업은 클라이언트의 무결성 검증 과정과 동시에 진행이 된다. 클라이언트에 대한 무결성 검증이 완료된 후, 보안 스토리지 접근 키에 대한 migration 요청이 발생하게 되고, 서버는 prefetching 된 키를 클라이언트의 하드웨어 TPM으로 migration 하게 된다. 그 후, 클라이언트의 하드웨어 TPM은 자신의 private 랜덤 키를 이용하여 해당 키를 복호화 하여 사용하게 된다. Key Prefetching을 수행함으로써 기존의 방법, 즉 인증 과정이 끝난 후에 키를 생성하는 방법보다 시간을 단축할 수 있다. 이에 대한 실험 결과는 다음장에 설명한다.

Ⅲ. 실험결과

1. 실험 환경

실험에 사용된 환경은 다음과 같다.

○ Lenovo T400
- Intel Core 2 Duo P8600 2.4 GHz
- 3GB Memory
- Intel TPM 1.2

○ Fedora 14
- Kernel Version 2.6.35.12

○ TSS (TrouSers 0.3.7)

○ TPM Emulator 0.7.2

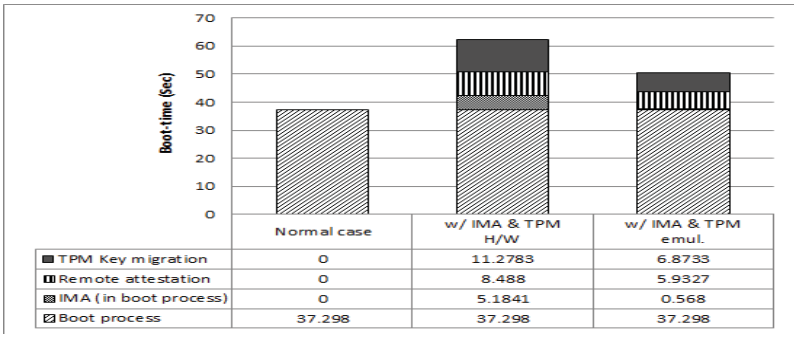

그림 3. 실험 결과

Fig 3. Experimental Result

2. 결과 및 분석

<그림 3>은 본 프레임워크의 부팅 시 걸리는 시간을 측정한 그래프이다. Normal case는 일반적인 리눅스의 부팅시간을 말한다. 두 번째와 세 번째 막대는 IMA를 통하여 클라이언트 무결성을 검증하고 하드웨어 TPM을 사용한 경우, IMA와 TPM 에뮬레이터를 사용한 경우의 부팅 시간을 각각 나타낸다.

하드웨어 TPM을 사용하는 경우 IMA 모듈이 동작하여 클라이언트 인증 작업을 수행하고 결과를 서버로 보내서 Remote attestation이 끝나면 키 전송 작업이 진행된다. IMA가 수행한 무결성 결과는 TPM의 PCR에 extend 명령을 통하여 저장되고 서버로 인증 결과를 보낼 때 quote 명령을 통하여 PCR에서 읽어 온다. quote와 extend 명령의 실행 회수를 측정한 결과 각각 1072회, 11회 수행됨을 알 수 있었다. 두 번째 막대의 IMA와 Remote attestation 부분은 하드웨어 TPM이 이들 커맨드를 수행하는 시간을 포함한다.

<그림 1>에서 볼 수 있듯이 quote와 extend 커맨드의 오버헤드가 TPM 에뮬레이터를 사용하여 수행했을 때보다 크다. 클라이언트가 부팅될 때 TPM 에뮬레이터를 사용한 경우에는 이들 커맨드를 소프트웨어적으로 처리하기 때문에 <그림 3>의 세 번째 막대에 나타나 있듯이 IMA와 Remote attestation에 걸리는 시간이 하드웨어 TPM의 경우보다 4.6초, 2.5초가 줄어들었음을 볼 수 있다.

Protected key migration의 시간은 key prefetching 기법을 적용하여 하드웨어 TPM의 경우 보다 본 프레임워크에서 적게 걸림을 알 수 있다. 인증 과정을 거친 후에 random 키를 클라이언트 TPM에서 생성하고 클라이언트 스토리지 정보를 보냄으로써 시작되는 key migration이 IMA의 검증 작업 전에 시작되도록 함으로써 서버의 인증 작업과 동시에 보안 스토리지에 사용되는 키가 생성 되도록 하였다.

IV. 결론 및 향후과제

본 논문에서는 모바일 기기에서 보안 작업을 수행하기 위해서 TPM의 기능을 사용한 보안 프레임워크를 설계하였다. 안전한 실행 환경에서 작업이 수행되는지를 서버로부터 인증 받고 보안 스토리지의 접근을 위해서 데이터를 암호화 하는데 필요한 키를 서버의 TPM으로부터 클라이언트 TPM에 전송 받음으로써 사용자 데이터의 악의적인 접근을 차단할 수 있다. 하지만 하드웨어 TPM을 모바일 단말에 완전히 적용하게 되면 TPM의 큰 성능 오버헤드로 인해 작업의 수행속도가 느려지고 배터리의 지속시간이 짧아진다는 문제점이 생기기 때문에 오버헤드를 줄이기 위한 방법을 제안하였다.

먼저 TPM의 주요 커맨드에 대한 하드웨어 TPM과 TPM 에뮬레이터의 성능차이를 측정하였다. 이를 바탕으로 암호화 키 관리는 소프트웨어의 보안 위험성 때문에 하드웨어 TPM이 수행하되 key prefetching을 적용하였으며, 모바일 기기의 무결성 검증에 관련된 커맨드는 소프트웨어 적으로 처리하도록 하여 TPM에 의한 오버헤드를 줄일 수 있었다.

하지만 TPM 에뮬레이터를 이용하게 될 경우, 모바일 기기의 메모리에 접근하여 인증 데이터를 수정 하거나 다른 데이터로 덮어쓰는 공격을 받을 수 있다. 이러한 문제점은 ARM core에서 제공하는 ARM TrustZone 기술을 사용하여 해결할 수 있을 것이다.

참 고 문 헌

[1] Enck W., Ongtang M., and McDaniel P., “Understanding Android Security”, IEEE Security and Privacy 7, 1, 2009

[2] Enck W., Octeau D., McDaniel P., and Chaudhuri S., “A Study of Android Application Security”, USENIX Security, 2011

[3] Trusted Computing Group (TCG) http://www.trustedcomputinggroup.org

[4] Trusted Platform Module (TPM) http://www.trustedcomputinggroup.org/developers/trusted_platform_module/

[5] Najwa Ararj, Anand Raghunathan, Srivaths Ravi, and Niraj K. Jha, “Energy and Execution Time Analysis of a Software-based Trusted Platform Module”, Proceedings of the conference on Design, automation and test in Europe, 2007.

[6] Software-based TPM Emulator http://tpm-emulator.berlios.de/