한국산학기술학회논문지
Vol. 12, No. 4 pp. 1876-1883, 2011

TPM에서 명령어 인가에 대한 오프라인 사전 공격과 대응책

오두환[1], 최두식[1], 김기현[2], 하재철[1*]
[1]호서대학교 정보보호학과, [2]충북대학교 컴퓨터공학과

An Off-line Dictionary Attack on Command Authorization in TPM and its Countermeasure

Doo-Hwan Oh[1], Doo-Sik Choi[1], Ki-Hyun Kim[2] and Jae-Cheol Ha[1*]
[1]**Dept. of Information Security, Hoseo University,**
[2]**Dept. of Computer Eng., Chungbuk National University**

요 약 TPM(Trusted Platform Module)은 신뢰 컴퓨팅 환경을 구성하기 위해 근간이 되는 하드웨어 칩으로서 이 칩의 주요 기능들을 사용하기 위해서는 명령어에 대한 사전 인가 과정이 필요하다. TPM 명령어 인가 과정은 명령어 사용자가 정당한 비밀 정보(usage secret)를 알고 있는 사용자인지를 TPM에게 인지시키는 과정이다. 명령어 인가에 사용되는 비밀 정보는 패스워드 형식이어서 공격자가 사용자와 TPM간의 전송 메시지를 도청할 수 있다면 오프 라인 사전 공격을 통해 유추될 가능성이 있다. 본 논문에서는 이론적으로 제시된 오프라인 사전 공격(off-line dictionary attack)이 가능한지 실제 TPM 칩을 장착한 PC 환경에서 그 가능성을 검증하고 이 공격을 방어하기 위한 새로운 대응책을 제안하였다. 제안하는 대응책은 현재 사용 중인 TPM 칩의 명령어 구조를 바꾸지 않은 상태에서 바로 응용할 수 있어 이전의 대응책보다 효과적으로 적용할 수 있다.

Abstract The TPM is a hardware chip for making a trusted environment on computing system. We previously need a command authorization process to use principal TPM commands. The command authorization is used to verify an user who knows a usage secret to TPM chip. Since the user uses a simple password to compute usage secret, an attacker can retrieve the password by evasdropping messages between user and TPM chip and applying off-line dictionary attack. In this paper, we simulate the off-line dictionary attack in real PC environment adopted a TPM chip and propose a novel countermeasure to defeat this attack. Our proposed method is very efficient due to its simplicity and adaptability without any modification of TPM command structures.

Key Words : TPM, OIAP, Usage Secret, Salt

1. 서 론

컴퓨팅 환경의 발전과 함께 IT 기기들은 다양해지고 소프트웨어도 복잡해짐으로써 이로 인해 발생하는 취약점들이 증가하고 있다. 이러한 문제를 해결하기 위하여 안전하고 신뢰할 수 있는 컴퓨팅 환경(TC : Trusted Computing)을 구축하기 위해 하드웨어적으로 플랫폼에 신뢰성을 부여하는 TPM이 개발되었다. TPM은 PC, 노트북, 스토리지, 스마트폰 등의 다양한 플랫폼에 장착될 수 있는 신뢰 컴퓨팅 전용 칩이다[1,2].

신뢰된 컴퓨팅을 위하여 1999년에 TCPA(Trusted Computing Platform Alliance)가 조직되어 신뢰 컴퓨팅이 연구되었고, 2003년 TCG(Trusted Computing Group)로 확대되었다. TCG는 다양한 워킹 그룹을 통하여 신뢰 컴퓨팅 환경을 위한 기반 기술 및 표준에 기여하고 있다[3]. TPM에 관한 국제 표준은 ISO/IEC 11889-1~4에서 규정하고 있는데 여기에는 TPM의 기능 및 구조 그리고 사용하는 명령어에 관한 표준을 제시하고 있다[4-7].

*교신저자 : 하재철(jcha@hoseo.edu)
접수일 10년 12월 30일 수정일 11년 02월 10일 게재 확정일 11년 04월 07일

TPM은 다음 3가지의 신뢰 컴퓨팅 기능을 제공하기 위한 근간이 된다[5]. 첫째, RTM(Root of Trust for Measurement)은 플랫폼을 구성하는 각각의 컴포넌트들에 대한 동작 무결성을 측정하는 기능에 대한 근간이 된다. 둘째, RTS(Root of Trust for Storage)로서 플랫폼에 저장되는 데이터를 보호하기 위한 키 관리 및 암•복호화를 제공한다. 마지막으로 RTR(Root of Trust for Reporting)은 다른 플랫폼에게 자신의 플랫폼에 대한 신뢰 상태를 검증하는 메커니즘을 제공한다. 이러한 TPM은 현재 Infineon, Atmel, Broadcom 등의 반도체 업체들에 의해 제조되어 PC나 노트북에 장착되어 사용되고 있다.

TPM 사용자는 120여개의 명령어를 통하여 TPM의 다양한 기능들을 수행할 수 있다. 그리고 TPM의 중요한 기능들을 사용하기 위해서는 한 번 또는 두 번의 명령어에 대한 인가(command authorization)가 필요하다. 명령어 인가과정에서 사용자는 자신이 알고 있는 비밀 정보를 이용하여 인가 데이터(authorization data)를 구성하고 명령어와 함께 TPM으로 전송함으로써 자신이 정당한 사용자임을 TPM에게 증명하게 된다. 그런데 명령어 인가 과정에서 사용되는 사용자 비밀 정보는 사용자가 입력하는 패스워드를 사용하는데 이 값을 SHA-1[8]으로 해쉬 연산을 수행한 후 인가 데이터를 생성하기 위한 HMAC의 비밀 키로 사용된다[9]. 즉, TPM에서 명령어 인가 비밀 정보는 사용자가 입력한 패스워드라고 볼 수 있다.

하지만, 공격자가 정당한 사용자와 TPM 사이에서 명령어 전송 과정을 도청한 후에 오프라인 사전 공격(off-line dictionary attack) 방법을 이용하면 사용자 비밀 정보가 복구될 수 있음이 Chen과 Ryan에 의해 제기되었다[10]. 그러나 저자들은 어느 부분에서 어떤 방식으로 패스워드를 공격하는지에 대한 구체적인 내용은 제시하지 않았다. 다만, 오프라인 사전 공격의 위험성을 제시하면서 이에 대응하기 위하여 SPEKE(Simple Password Exponential Key Exchange) 방법을 제안하였다[11,12]. 그렇지만 이 대응 방법을 사용하기 위해서는 TPM 칩의 명령어 구조 및 설계 자체를 변경해야 하므로 실제로 사용하기는 어렵다.

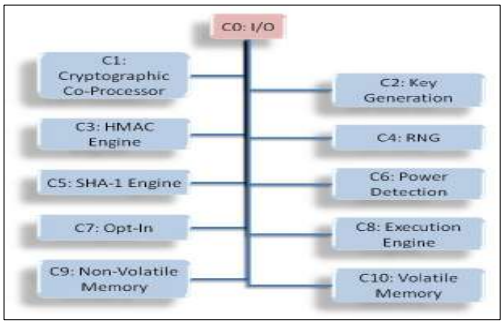
본 논문에서는 TPM 명령어 인가에 대한 오프라인 사전 공격을 분석하고 시뮬레이션을 통하여 실제 공격 가능성을 검증한다. 또한, 이전의 대응 프로토콜을 분석한 후에 현재 TPM에서 바로 적용될 수 있는 새로운 오프라인 사전 공격 대응 방법을 제안하고 이를 실제 TPM에 적용해 본다.

본 논문의 구성은 다음과 같다. 2장에서는 TPM 칩 구성 요소, 키 구조, TSS(TCG Software stack)에 대한 기본 개념을 소개하고, 3장에서는 TPM 명령어에 대한 인가, 오프라인 사전 공격과 이전의 대응 방법을 분석한다. 4장에서는 제안하는 대응 방법을 소개하고 이를 실제 TPM을 이용한 응용 프로그램에 적용시켜 그 실효성을 검증해 본다. 마지막으로 5장에서 결론을 맺는다.

2. TPM 개요

2.1 TPM 칩 구성요소

TPM 칩 내부에는 그림 1과 같이 다양한 보안 서비스를 제공하기 위한 기능적 요소가 설계되어 있다[5]. TPM 칩의 각 컴포넌트들에 대한 내용은 표 1에 정리하였다.



[그림 1] TPM의 기본구조

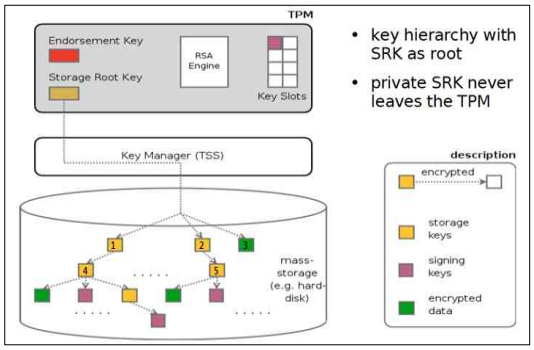
[표 1] TPM 주요 컴포넌트 기능

컴포넌트 명	내 용
Cryptographic Co-Processor	다양한 암호학적 연산을 위한 프로세서
Key Generation	RSA 키쌍을 생성
HMAC Engine	인가 값에 대한 검증 및 응답 명령어 구성 시 사용
RNG	nonce값 또는 키 생성
SHA-1 Engine	해쉬 연산을 수행
Execution Engine	칩 운영체제와 명령어 처리
Non-Volatile Memory	SRK와 EK가 저장
Volatile Memory	PCR 레지스터와 AIK가 저장

2.2 TPM 키 구조

TPM을 사용하는 플랫폼에서는 다양한 키가 사용될 수 있다. 이 키들은 TPM의 공간이 한정적이기 때문에 TPM 내부가 아닌 외부에 계층적으로 안전하게 저장되어 관리된다. TPM 칩 제조사가 생성하며 다른 TPM과 구별

하기 위한 유일한 키인 EK(Endorsement Key)와 TPM 소유권 획득 과정을 통하여 TPM 키 계층 최상위에 위치하게 되는 SRK(Storage Root Key)만이 TPM 내부에 안전하게 저장된다. 이 키들은 RSA 키 쌍으로 이루어져 있으며 비밀 키는 절대 TPM 외부로 노출되지 않는다.

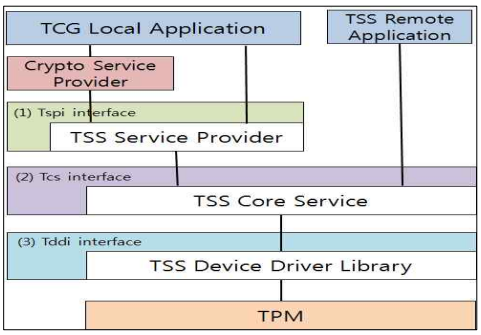


[그림 2] TPM 키 계층 구조

그림 2와 같이 플랫폼의 모든 키들은 SRK 키를 최상위로 하는 트리 구조 노드들로 구성된다. 계층 구조를 이루고 있는 키들의 공개 키는 암호화되어 저장되지 않지만, 비밀 키는 부모 키의 공개 키로 암호화되어 저장된다. 예를 들어 그림의 1, 2번 비밀 키 또는 암호화된 데이터는 부모 키인 SRK 키의 공개 키로 암호화되어 TPM 외부에 저장되며, 4번의 비밀 키는 1번 키의 공개 키로 암호화되어 저장된다.

2.3 TSS(TCG Software Stack)

TSS란 TPM에 대한 사용자 인터페이스를 제공하고, 사용자가 응용 프로그램을 통하여 TPM의 기능을 사용할 수 있는 계층적 구조를 나타낸 것이다[13]. 그림 3은 TSS의 기본 구조를 나타내고 있다.



[그림 3] TSS 기본 구조

그림 3과 같이 사용자 프로그램과 TPM 사이에는 4개의 계층이 존재하게 되고 TSP(TSS Service Provider), TCS(TSS Core Service), TDDL(TSS Device Driver Library) 단계를 이용하여 사용자는 TPM 칩을 이용할 수 있다.

일반적으로 TPM 칩은 여러 사용자가 하나의 칩을 이용할 수 있도록 설계되어 있어 TCS는 서버처럼 백그라운드로 운영되며 TSP는 클라이언트처럼 다수의 사용자에 의해 이용된다.

3. TPM 명령어 인가 및 사전 공격

3.1 OIAP 프로토콜

TPM에서 사용하는 명령어 인가 프로토콜 중 OIAP(Object-Independent Authorization Protocol)는 개체 독립적인 인가 프로토콜로서 사용자의 TPM 요청 명령어 이전에 TPM_OIAP라는 명령어를 통해 수행된다. 그림 4은 OIAP를 이용한 명령어 인가 과정을 도식화한 것이다. 여기서 사용된 파라미터는 아래와 같다.

- sH : 현재 인가 세션의 핸들 값
- CM : ordinal + 요청 명령어의 파라미터들
- RV : return code + ordinal + 결과 값
- ParamD : SHA-1(CM) 또는 SHA-1(RV)
- nO : 사용자가 생성한 랜덤 값
- nE_1, nE_2 : TPM이 생성한 랜덤 값
- continueAuthS : 인가 세션 플래그
- kH : 키의 핸들 값
- KH(usageSecret) : 키 사용 비밀정보
- AuthD : 인가 데이터

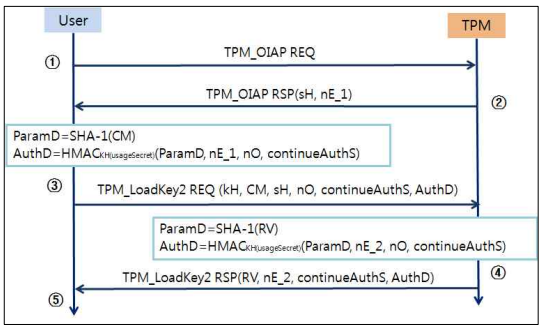
TPM의 OIAP 프로토콜 및 실제 명령어 요청 및 응답은 다음과 같은 절차에 의해 수행된다. 여기에서는 TPM 칩에 키를 로드하는 명령어인 TPM_LoadKey2를 사용하는 절차를 보이고 있다.

① 사용자는 TPM에게 TPM_OIAP 명령어를 전송한다.
② TPM은 세션의 핸들 값(sH)와 랜덤 값(nE_1)를 사용자에게 전송한다.
③ 사용자는 ordinal과 요청 명령어에 필요한 파라미터를 SHA-1 해쉬하여 ParamD를 생성한 후 랜덤한 값인 nO를 생성하고 사용하려고 하는 키(kH)의 usageSecret, ParamD, nE_1, nO의 정보로 HMAC을 계산한다. 그 결과 AuthD이 생성되고 이를 명령어

(TPM_Loadkey2)에 포함하여 TPM에게 전송한다.

④ TPM은 요청 명령어를 수신한 후에 사용자처럼 자신이 알고 있는 키(kH)의 usageSecret를 이용하여 사용자와 동일한 방법으로 AuthD를 계산하고 사용자가 보낸 AuthD와 비교하여 같으면 명령어를 처리하고, 다르면 사용자에게 에러 코드를 전송한다. 이 후에 TPM은 새로운 랜덤 값(nE_2)을 생성하여 키(kH)의 usageSecret를 이용하여 응답 명령어에 대한 AuthD를 계산한 후에 사용자에게 응답 명령어를 전송한다.

⑤ 사용자는 응답 명령어를 TPM으로부터 수신한 후에 자신이 알고 있는 키(kH)의 usageSecret을 이용하여 AuthD를 계산한 후에 TPM으로부터 수신한 AuthD과 비교를 통해 응답 명령어를 검증한다.



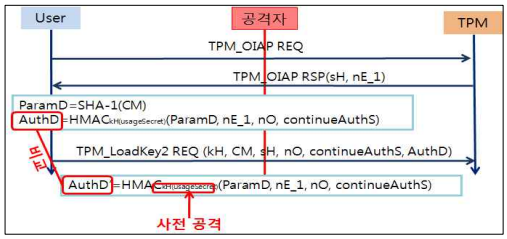
[그림 4] OIAP를 이용한 명령어 인가

3.2 OIAP에 대한 사전 공격

처음 TPM에서의 오프라인 사전 공격을 제시한 Chen과 Ryan는 공격의 위험성만 제시하고 구체적인 공격 방법은 언급하지 않았다. 따라서 본 논문에서는 실제로 오프라인 사전 공격이 가능한 모델을 제시하고자 한다.

TPM에서 인가가 필요한 명령어들은 명령어에 포함되서 전달되는 인가 데이터를 통하여 검증되며 명령어에 대한 인가 절차는 그림 4와 같은 과정을 수행한다. TPM_OIAP 명령어를 통해 인가를 받은 사용자는 실제로 TPM 구동 명령어를 보내게 된다.

이 경우 공격자가 사용자와 TPM 사이에 위치하여 메시지 도청 프로그램을 구동한다고 가정해 보자. 메시지 도청 프로그램을 사용하여 송·수신되는 정보를 도청한 공격자는 도청 정보를 가지고 사용자 비밀 정보인 usage secret을 공격하게 된다. 사전 공격이 성공하면 공격자는 정당한 사용자로 위장할 수 있고 TPM의 다양한 기능을 수행할 수 있다. 그림 5는 OIAP에 대한 사전 공격 과정을 나타낸 것이다.



[그림 5] OIAP에 대한 사전 공격

그림 5와 같이 공격자는 사용자와 TPM 사이에 위치하여 명령어 송·수신 메시지를 도청하면 ParamD, nE_1, nO, continueAuthS 그리고 AuthD를 알 수 있다. 이 경우 각 파라미터간 관계식은 아래와 같다. 이때 $HMAC_s(M)$는 메시지 M을 대한 비밀 정보 s로 해쉬 기반 메시지 인증코드를 계산함을 의미한다.

$$AuthD = HMAC_{KH(UsageSecret)}(ParamD, nE\_1, nO, continueAuthS)$$

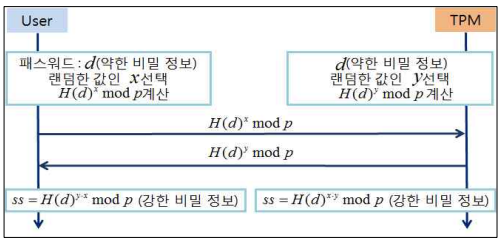
위 계산식에서 보면 다른 정보는 도청으로 모두 알 수 있지만 160비트의 UsageSecret은 알지 못한다. 그러나 UsageSecret=SHA-1(패스워드)의 관계식이 성립하므로 공격자는 사전에서 가능한 패스워드를 선택하여 UsageSecret을 계산하고 이를 이용하여 공격자의 AuthD'을 계산한다.

$$AuthD' = HMAC_{KH(UsageSecret)}(ParamD, nE\_1, nO, continueAuthS)$$

이제 공격자는 도청한 AuthD와 계산한 AuthD'이 동일한지를 검사하여 패스워드를 추출하게 된다.

3.3 Chen과 Ryan의 대응 방법

Chen과 Ryan이 제안한 사전 공격 대응 방법은 그림 6과 같다. 이 대응책에서는 HMAC 키를 입력한 패스워드의 SHA-1 해쉬 값을 이용하는 것이 아니라, 패스워드 기반의 키 동의 프로토콜인 SPEKE를 이용한다. 즉, 사용자가 입력한 패스워드의 SHA-1 해쉬 값을 약한 비밀 정보라 정의하고 사용자와 TPM과의 키 교환을 통하여 강한 비밀 정보를 생성한다.



[그림 6] Chen과 Ryan의 대응 방법

구체적으로, 그림 6과 같이 사용자는 패스워드를 입력하고 랜덤한 값 x를 선택하고 $H(d)^x \bmod p$를 계산한 후에 TPM으로 전송한다. TPM은 랜덤한 값 y를 선택하고 $H(d)^y \bmod p$를 계산한 후에 사용자에게 전송한다. 사용자와 TPM은 강한 비밀 정보인 ss를 계산한 후에 이 값을 UsageSecret으로 설정하고 HMAC 키로 사용한다.

하지만, 이 대응 방법은 현재 생산되고 있는 TPM 칩에서는 적용할 수 없다. 그 이유는 현재 사용중인 TPM 칩은 미리 정해진 명령어에 대해서만 응답하는 방식이므로 Chen과 Ryan이 제시한 명령어 구조를 사용할 수 없다. 또한, 이 대응방법을 사용하기 위해서는 TPM 내부에 $\bmod p$ 연산을 위한 엔진이 필요하지만 현재는 지원하지 않고 있다. 따라서 기존의 대응 방법은 UsageSecret을 결정하는 방법이 비현실적이며 사용에 제약적이다.

4. 오프라인 사전 공격 대응방법 제안

4.1 솔트를 이용한 비밀정보 생성 과정

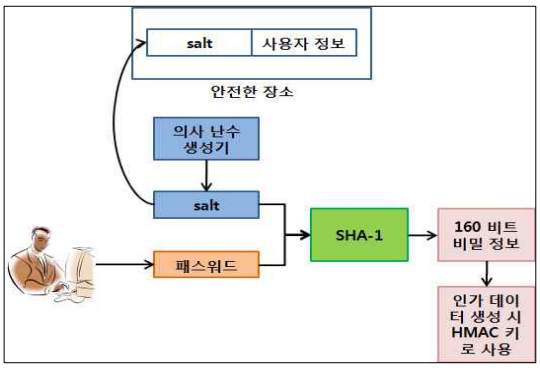
본 논문에서 TPM 명령어 인가 과정에서 사용하는 비밀 정보에 대한 사전 공격에 대응하는 효과적이고 실질적인 방법을 제안하고자 한다. 제안 방법에서는 UsageSecret을 생성할 때 패스워드와 솔트(salt)를 이용하고자 한다. 솔트란 의사난수 생성기로 만들어지는 랜덤한 값이며 사용자가 입력한 패스워드와 함께 해쉬 값의 입력으로 사용된다. 현재 OIAP를 이용한 명령어 인가에서는 사용자의 패스워드를 SHA-1 알고리듬으로 해쉬한 값을 인가 데이터를 생성하기 위한 HMAC 키로 사용하였다. 즉, UsageSecret=SHA-1(패스워드) 형식이었다.

그러나 제안하는 대응 방법에서는 패스워드를 해쉬 하기 전에 랜덤한 값인 솔트를 생성하여 패스워드와 함께 해쉬 입력으로 사용하고자 한다. 즉, UsageSecret=SHA-1(패스워드 ‖ salt)와 같이 생성 방법을 변경하였다. 또한 솔트의 크기는 공격자가 예측할 수 없을 정도의 큰 길이를 가져야 하며 160비트 정도로 가정할 수 있다. 그림 7은 솔트를 이용하여 사용자 비밀 정보를 생성하는 과정을 나타내고 있다.

그림 7에서 보는 바와 같이 사용자가 패스워드를 입력하면 플랫폼은 의사 난수 생성기를 이용하여 솔트를 생성하고 생성된 솔트 값과 패스워드를 해쉬한 결과 값을 UsageSecret으로 사용한다. UsageSecret 생성 과정에서 UsageSecret 정보는 TPM으로 전송되고 TPM은 이를 저장하고 있다.

이때 솔트 정보가 매우 중요한데 그 이유는 공격자가 솔트 값이 공격자에게 노출되면 역시 동일한 방법으로 사전공격이 가능하기 때문이다. 따라서 솔트는 그에 대응되는 사용자 정보와 함께 사용자의

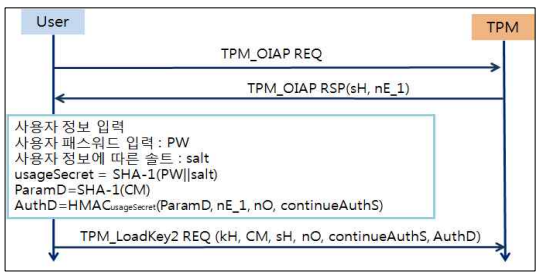
USB 혹은 보안 토큰과 같은 개인적인 저장소에 저장하여야 한다. 사용자 플랫폼의 하드디스크를 이용할 경우에는 일반 사용자의 접근이 어려운 데이터베이스를 활용하거나 공격자가 알 수 없는 파일명 형태로 저장하여 프로토콜 도청 공격외의 2차적인 솔트 탐색 공격에 대비해야 한다.



[그림 7] 솔트를 이용한 비밀 정보 생성 과정

4.2 비밀정보 사용 과정

위의 비밀정보 생성과정을 거친 사용자는 이후 TPM 명령어를 사용하게 된다. 이때에도 3장에서 언급한 바와 같이 OIAP를 이용하여 명령어 인증을 수행한다. 그림 8은 OIAP에 제안 방식을 적용한 명령어 인가 과정을 나타낸 것이다. 이전의 OIAP 프로토콜 과정에서 사용자는 사용자 정보와 패스워드를 입력하게 되고 salt의 위치를 지정하게 된다.



[그림 8] OIAP 프로토콜에서 대응방법 적용

그 후 사용자는 SHA-1(패스워드 ‖ salt) 연산을 거쳐 UsageSecret을 계산하고 인가 데이터 AuthD를 생성하여 TPM에게 전송하게 된다.

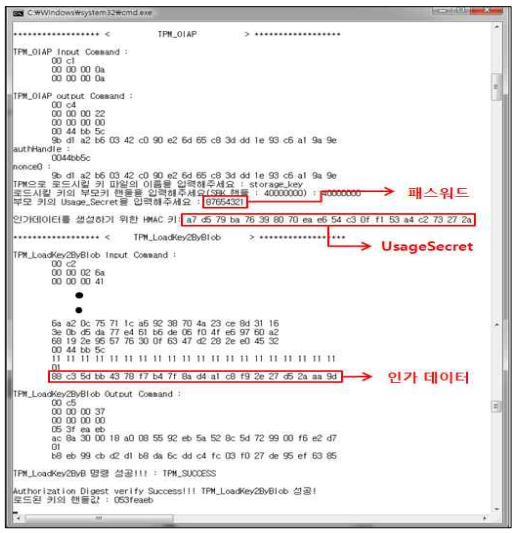
제안 방식을 사용할 경우, 공격자의 도청에 의한 오프라인 사전 공격을 가정해 보자. 그림 8에서 공격자는 위에서와 동일하게 ParamD, nE_1, nO, continueAuthS 그리고 AuthD를 알 수 있다. 이 경우에도 각 파라미터간 관계식은 아래와 같다.

$$AuthD = HMAC_{KH(UsageSecret)}(ParamD,\ nE\_1,\ nO,\ continueAuthS)$$

이제 공격자는 패스워드 사전 검사에 필요한 UsageSecret을 계산해야 한다. 그러나 제안 방식에서는 패스워드를 정확히 추측해야 할 뿐만 아니라 솔트 값을 알고 있어야만 UsageSecret을 계산할 수 있다. 만약 사용자에 의해 솔트 값이 물리적으로 안전한 위치에 있다고 가정하면 공격자는 오프라인 사전 공격을 성공할 수 없다.

4.3 제안 방법 시뮬레이션

논문에서는 TPM이 장착된 PC에서 오프라인 사전 공격을 실험하여 그 공격 가능성을 확인하고 제안 방식을 적용한 프로그램의 동작 과정을 시뮬레이션 하였다. 시뮬레이션은 Infineon사의 TPM 칩인 SLB 9635가 장착한 PC[14]에서 수행하였다. TPM 칩 구동을 위한 운영체제로 Window 7을 사용하였으며 Visual Studio 2008에서 개발하였다.

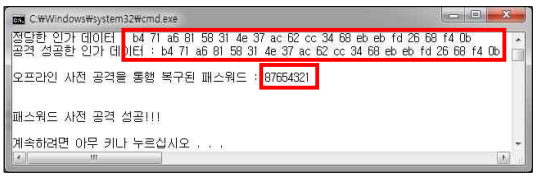


[그림 9] 일반적인 OIAP 프로토콜 수행 화면

먼저, 그림 9는 2장에서 기술한 일반적인 OIAP 인가 프로토콜을 사용한 키 로드 명령어(TPM_LoadKey2)를 수행하는 과정을 보인 것이다. 여기서 사용자의 패스워드는 문자열“87654321”이라고 가정하였다.

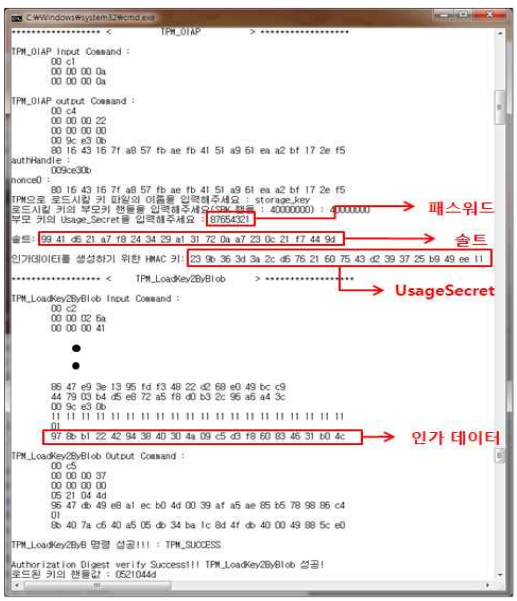
그림에서 보는 바와 같이 패스워드 “87654321”를 해쉬하면 그 결과는 “A7 D5 79 BA 76 39 80 70 EA E6 54 C3 0F F1 53 A4 C2 73 27 2A”이 되고 이것이 UsageSecret으로 사용됨을 볼 수 있다. 이러한 환경하에서 사용자와 TPM간의 전송 정보를 도청할 수 있는 프로그램을 개발하여 메시지 도청을 시도해 보았다.

시뮬레이션 결과, 공격자가 메시지 도청 프로그램을 이용하여 사전 공격에 필요한 파라미터인 ParamD, nE_1, nO, continueAuthS 그리고 AuthD를 절취할 수 있음을 확인하였다. 그림 10은 오프라인 사전 공격에 대한 실험 화면으로서 3장에서 소개하였던 것과 동일한 사전 공격 방법을 이용하여 사용자 패스워드를 찾아낼 수 있음을 보이고 있다.



[그림 10] 오프라인 사전 공격 실험 화면

논문에서는 위와 같은 오프라인 사전 공격을 방어할 수 있는 대응 방법을 제안하였고 이를 시뮬레이션해 보았다. 그림 11는 제안하는 대응 방법을 이용하여 그림 9와 동일한 키 로드 명령어(TPM_LoadKey2)를 수행하는 과정을 나타낸 것이다.



[그림 11] 제안된 OIAP 프로토콜 수행 화면

그림 11에서 보면, 제안 방식에서 솔트를 사용했기 때문에 패스워드 문자열 “87654321”과 솔트 값인 “99 41 d6 21 a7 f8 24 34 29 a1 31 72 0a a7 23 0c 21 f7 44 9d”를 입력으로 사용하였다. 그 결과 SHA-1 해쉬 값인 “23 9B 36 3D 3A 2C D6 76 21 60 75 43 D2 39 37 25 B9 49 EE 11”이 UsageSecret으로 사용된다. 결론적으로 제안 방식은 실제 TPM 장착된 환경에서도 잘 동작하며 메시지가 중간에 도청되더라도 오프라인 사전 공격이 불가능함을 확인하였다.

5. 결 론

본 논문에서는 TPM에서 명령어 인가 프로토콜인 OIAP 수행 시 사용자 비밀 정보에 대한 사전 공격의 가능성을 소개하였다. 대부분의 사용자들은 기억하기 쉬운 패스워드를 선호하기 때문에 사용자와 TPM간의 도청 정보를 이용하면 사전 공격에 매우 취약하다. 지금까지 제안된 대응 방법은 이론적으로는 사전 공격을 방어할 수 있지만 현재 TPM 칩 명령어 구조에 적용할 수 없고 TPM 내부의 연산 엔진을 이용할 수 없다는 단점을 가지고 있다.

따라서 본 논문에서는 TPM의 명령어 체계를 그대로 이용하면서 오프라인 사전 공격을 방어하는 실질적인 방법을 제안하였다. 제안하는 방법은 현재 사용 중인 TPM 칩에 바로 적용할 수 있을 뿐만 아니라 프로토콜이 매우 간단하다. 따라서 제안하는 OIAP 명령어 인가 프로토콜은 TPM 칩을 보다 안전하게 사용할 수 있는 기반 구조를 제시한다.

참고 문헌

[1] 김영수, 박영수, 박지만, 김무섭, 김영세, 주홍일, 김명은, 김학두, 최수길, 정성익, “신뢰 컴퓨팅과 TCG 동향”, 전자통신동향분석, 제22권, 제1호, pp. 83-96, 2007.

[2] 강동호, 한진희, 이윤경, 조영섭, 한승완, 김정녀, 조현숙, “스마트폰 보안 위협 및 대응 기술”, 전자통신동향분석, 제 25권 3호, 2010.

[3] Trusted Computing Group, "About TCG", Available at http://www.trustedcomputinggroup. org

[4] ISO/IEC 11889-1 : Information technology - Security techniques - Trusted Platform Module - Part 1: Overview, 2009.

[5] ISO/IEC 11889-2 : Information technology - Security techniques - Trusted Platform Module - Part 2: Design principles, 2009.

[6] ISO/IEC 11889-3 : Information technology - Security techniques - Trusted Platform Module - Part 3: Structures, 2009.

[7] ISO/IEC 11889-4 : Information technology - Security techniques - Trusted Platform Module - Part 4: Command, 2009.

[8] FIPS PUB 180-1, Secure Hash Standard (SHA-1), National Institute of Standards and Technology(NIST), 1995.

[9] FIPS PUB 180-1, The Keyed-Hash Message Authentication Code(HMAC), National Institute of Standards and Technology(NIST), 2002.

[10] L. Chen and M. D. Ryan, "Offline dictionary attack on TCG TPM weak authorization data, and solution", Future of Trust in Computing, Vieweg & Teubner, 2008.

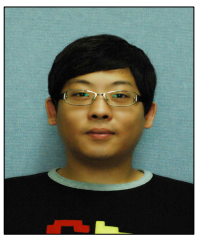
[11] D. Jablon, "Strong password-only authenticated key exchange", Computer Communication Review, 26(5):5-26, ACM SIGCOMM, Oct. 1996.

[12] D. Jablon, "Extended password key exchange protocols immune to dictionary attack", In Proceedings of WET-ICE'97, pp. 248-255, 1997.

[13] Trusted Computing Group, "TCG Software Stack(TSS) Specification Version 1.2 Level 1 Errata A", 2007.

[14] Infineon, "Trusted Platform Module TPM 1.2 SLB 9635TT1.2", Available at http://www.infineon.com/tpm

오 두 환(Doo-Hwan Oh) [정회원]



- 2010년 2월 : 호서대학교 정보보호학과 (공학사)
- 2010년 3월 ~ 현재 : 호서대학교 대학원 정보보호학과(석사과정)

<관심분야>
부채널 공격 네트워크 보안, 신뢰 컴퓨팅

최 두 식(Doo-Sik Choi) [정회원]

- 2010년 2월 : 호서대학교 정보보호학과 (공학사)
- 2010년 3월 ~ 현재 : 호서대학교 대학원 정보보호학과(석사과정)

<관심분야>
신뢰 컴퓨팅, 네트워크 보안, 부채널 공격

김 기 현(Ki-Hyun Kim) [정회원]

- 1993년 2월 : 경북대학교 전자공학과 (공학사)
- 1995년 2월 : 경북대학교 전자공학과 (공학석사)
- 2010년 2월 : 충북대학교 컴퓨터공학과 (박사 수료)
- 2009년 7월 ~ 현재 : 에스지에이㈜ R&D부문 총괄사장

<관심분야>
시스템 및 네트워크 보안, 보안관제, 전자문서보안

하 재 철(Jae-Cheol Ha) [종신회원]

- 1989년 2월 : 경북대학교 전자공학과 (공학사)
- 1993년 8월 : 경북대학교 전자공학과 (공학석사)
- 1998년 2월 : 경북대학교 전자공학과 (공학박사)
- 1998년 3월 ~ 2007년 2월 : 나사렛대학교 정보통신학과 부교수
- 2007년 3월 ~ 현재 : 호서대학교 정보보호학과 부교수

<관심분야>
정보보호, 네트워크 보안, 부채널 공격