Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.2, Apr. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.2.335

HW 보안 모듈을 활용한 탬퍼링 대응 기술의 검증 및 평가 방안 조사*

이 동 호,[1†] 반 영 훈,[1] 임 재 덕,[3] 조 해 현[2‡]
[1,2]숭실대학교 (대학원생, 교수), [3]한국전자통신연구원 (연구원)

Investigation of Verification and Evaluation Methods for Tampering Response Techniques Using HW Security Modules*

Dongho Lee,[1†] Younghoon Ban,[1] Jae-Deok Lim,[3] Haehyun Cho[2‡]
[1,2]Soongsil University (Graduate student, Professor),
[3]Electronics and Telecommunications Research Institute (Researcher)

요 약

디지털 시대의 발전 속에서 데이터의 안전성은 어느 때보다 중요한 이슈로 주목받고 있다. 특히, 무분별한 해킹 및 무단 접근으로부터 정보를 보호하기 위한 안티탬퍼링 기술은 핵심 대응책으로 주목받고 있다. 본 논문은 TPM (Trusted Platform Module)과 SW(Software) 안티탬퍼링 기술의 발전 추세와 현대 디지털 환경에서 이 기술이 어떻게 적용되는지에 대한 사례를 다룬다. 기존에 존재하는 다양한 보안 가이드 및 지침들을 분석하여, 가이드 및 지침들에 포함된 모호한 부분들을 찾아내었으며, 최신 국/내외의 SW 안티탬퍼링 연구에 대한 동향을 알아본다. 결과적으로 안티탬퍼링 기법을 적용하기 위한 지침은 존재하지만 안티탬퍼링 기법을 평가하기 위한 방안이 존재하지 않는 것을 확인하였다. 따라서 본 논문에서는 기존에 제안된 다양한 SW 안티탬퍼링 기법들을 포함하여 앞으로 제안될 SW 안티탬퍼링 기법들을 평가하기 위한 포괄적이고 체계적인 평가 방안을 제시한다. 본 논문은 포괄적인 평가 방안을 제시하기 위해 최신 연구들이 사용하는 다양한 검증 방안을 이용한다. 본 논문이 제시하는 포괄적이고 체계적인 평가 방안은 각 연구에서 사용된 검증 방안들을 종합하여 3가지(기능, 구현, 성능)로 구분함으로써 SW 안티탬퍼링 기술을 전반적으로 평가할 수 있는 종합적이고 체계적인 상세한 평가 방안에 대해 제시한다.

ABSTRACT

In the digital era, data security has become an increasingly critical issue, drawing significant attention. Particularly, anti-tampering technology has emerged as a key defense mechanism against indiscriminate hacking and unauthorized access. This paper explores case studies that exemplify the trends in the development and application of TPM (Trusted Platform Module) and software anti-tampering technology in today's digital ecosystem. By analyzing various existing security guides and guidelines, this paper identifies ambiguous areas within them and investigates recent trends in domestic and international research on software anti-tampering. Consequently, while guidelines exist for applying anti-tampering techniques, it was found

Received(01. 25. 2024), Modified(03. 05. 2024), Accepted(03. 20. 2024)
* 이 논문은 2023년 정부(방위사업청)의 재원으로 국방기술진흥연구소의 지원을 받아 수행된 연구임(KRIT-CT-22-051) This work was supported by Korea Research Institute for defense Technology planning and advancement(KRIT) - Grant funded by Defense Acquisition Program Administration (DAPA) (KRIT-CT-22-051)
* 이 자료에 제시된 모든 의견, 발견 결과, 결론 또는 권고는 저자의 것이며, 방위사업청의 견해를 반영한 것이 아님.
† 주저자, tndtlfehdgh@soongsil.ac.kr
‡ 교신저자, haehyun@ssu.ac.kr(Corresponding author)

that there is a lack of methods for evaluating them. Therefore, this paper aims to propose a comprehensive and systematic evaluation framework for assessing both existing and future software anti-tampering techniques. To achieve this, it using various verification methods employed in recent research. The proposed evaluation framework synthesizes these methods, categorizing them into three aspects (functionality, implementation, performance), thereby providing a comprehensive and systematic evaluation approach for assessing software anti-tampering technology in detail.

Keywords: Hardware Security, Software Security, Trusted Platform Module, Anti Tampering

I. 서 론

현대 사회는 디지털화의 흐름 속에서 빠르게 변화하고 있으며 개인정보부터 기업의 민감한 데이터에 이르기까지 많은 정보가 디지털 형태로 저장 및 관리되는 환경을 만들어냈다. 이에 따라 디지털 보안이 중요한 이슈로 떠오르고 있어 하나의 작은 보안 사고만으로도 개인의 프라이버시와 기업의 경쟁력이 큰 위협을 받게 된다. 2019년부터 2022년까지의 사이버 보안 침해 사고 신고 건수는 꾸준히 증가하였으며, 2022년에는 1,045건에 달하는 것으로 나타났다[1]. 2023년 10월에 게시된 보안 뉴스에 따르면 증권사와 대부 중계 플랫폼 등 9개의 업체에서 해킹으로 인해 106만 건의 개인정보가 유출되었다는 보고가 있었다[2].

탬퍼링과 같은 보안 위협은 정보나 시스템의 무결성을 손상하기 위한 불법적인 조작으로서, 현대 디지털 시대에는 더욱 다양한 형태로 나타나고 있다. 탬퍼링 공격은 시스템의 기능을 손상시키거나 중요한 데이터를 유출할 위험이 있기 때문에 금전적 손해나 사용자의 개인정보 노출은 물론 기업의 경쟁력 저하와 같은 심각한 문제를 초래할 수 있기 때문에 디지털 보안 위협에 대응하기 위해서 HW(Hardware) 보안 모듈의 필요성이 대두되고 있다[3]. HW 수준에서의 보안은 SW(Software)만을 이용한 방법보다 훨씬 강력하며 이를 통해 물리적인 보안 장벽을 구축할 수 있다. 특히, TPM(Trusted Platform Module) 같은 보안 모듈은 안전한 키 관리와 데이터의 무결성 및 기밀성을 보장하는 기능을 제공하며, 디지털 보안 환경에서 그 중요성이 점점 더 커지고 있다[4].

이에 따라 TPM의 장점을 활용하는 SW 안티탬퍼링 기술 역시 많이 발전하고, 다양한 연구들이 수행되고 있다[4,7,9,16,17].

하지만 이 연구들에서 사용된 평가 방법/지표 역시 표준화된 방법은 아니며, SW 안티탬퍼링 기법을 평가하기에 범용적으로 사용하기 어렵다.

따라서 본 논문에서는 TPM 기반의 SW 안티탬퍼링 기법들과 SW 안티탬퍼링 기법들을 알아본다. 이후 이 연구들이 사용하는 평가 및 검증 기법을 이용하여 다양한 SW 안티탬퍼링 기법들을 종합적으로 평가할 수 있도록 체계적이고 상세한 평가 방안을 제시한다.

II. 배경지식

현대의 디지털 환경에서 컴퓨터와 서버의 보안은 핵심적인 이슈로 부상하고 있다. 기업의 중요한 정보나 개인의 민감한 데이터를 보호하는 것 외에도, 시스템의 안정적인 작동을 위해 다양한 보안 기술과 메커니즘들이 연구되고 있다. 이러한 보안 기술 중에서 시스템의 무결성과 보안 강화를 목표로 하는 TPM과 Secure Boot는 매우 중요한 역할을 담당하고 있다[5].

이 장에서는 TPM의 다양한 형태와 기능, 그리고 Secure Boot의 중요성 및 작동 원리에 대해 알아본다.

2.1 TPM (Trusted Platform Module)

TPM은 컴퓨터나 서버에 부착되는 전용 보안 칩으로, 시스템의 무결성과 보안을 강화하기 위해 사용된다[6]. TPM은 안전한 방법으로 암호키를 생성하고 저장하기 때문에 시스템의 위변조 여부와 데이터가 올바르게 암호화되어 보호되고 있는지를 확인할 수 있다. TPM의 구성은 아래와 같다.

2.1.1 암호화 알고리즘 엔진과 암호화키 생성기

암호화 알고리즘 엔진은 데이터의 보안을 위해 필수적으로 사용되며 이를 통해 생성된 암호화키는 데이터를 안전하게 암호화 및 복호화하는 데 사용한다. TPM은 암호화 과정에서 키 생성을 보조하며 다양한 암호화 알고리즘을 지원하여 보안성을 높인다.

2.1.2 해시엔진

해시 알고리즘은 데이터의 무결성을 확인하는 데 사용된다. 데이터의 작은 변화도 해시 값의 큰 변화로 나타나기 때문에 데이터의 무결성을 확인하는 데 효과적이다. TPM에서는 주로 SHA-1과 SHA-256과 같은 해시 알고리즘을 활용하여 데이터 무결성 검증을 지원한다.

2.1.3 난수 생성기

난수 생성기는 보안 프로세스 중 중요한 요소인 난수 생성을 지원하며 다양한 보안 요소(OTP, 암호화키, 토큰 등)의 생성에 활용될 수 있다.

2.1.4 비휘발성 메모리 (Non-Volatile Memory)

TPM에는 비휘발성 메모리가 포함되어 있어, 전원이 종료되어도 데이터를 유지할 수 있다. 이 메모리는 암호화된 키, 인증서, 그리고 다른 보안 관련 정보를 저장하는 데 사용된다.

2.1.5 TPM 장점 및 사용사례

TPM은 시스템의 보안을 강화하기 위해 설계되었으며, 물리적인 형태로 존재하여 공격자가 정보를 추출하기 어렵게 만든다 [7].

또한, TPM은 시스템의 신뢰할 수 있는 부팅 과정을 지원하며, 부팅 중에 HW와 SW의 무결성을 검사한다. 해당 기능을 통해 악성코드나 루트킷과 같은 위협으로부터 시스템을 보호할 수 있으며, 원격 인증도 지원한다.

TPM의 장점 및 기능은 다음과 같다.

- 물리적 강도: TPM은 물리적인 칩 형태로 제공되어 HW 수준에서의 보안이 강화된다.
- 보안 강도: 공격자가 직접적으로 칩에 접근하지 않는 한, 정보를 추출하기는 어렵다.
- 시스템 무결성 검증: 부팅 시 또는 애플리케이션 실행 과정에서 시스템의 무결성을 검사한다.
- 암호화 및 인증: 데이터 암호화 및 시스템 인증 과정을 담당한다.
- 키 및 비밀번호 관리: 안전하게 키를 생성하고 관리할 수 있는 기능을 제공하며, 비밀번호 관리 기능도 포함되어 있다.
- 제한된 SW 실행: 미인증 SW의 실행을 제한한다.
- HW 기반 난수 생성: 보안 프로세스에 필요한 난수를 생성한다.

TPM을 사용하는 사례는 다음과 같다.

- 노트북 및 서버: 대부분의 최신 노트북과 서버의 메인보드에는 TPM이 내장되어 있다. 이를 통해 안전한 부팅 과정을 보장하며, 디지털 인증서의 개인 키와 같은 중요한 정보를 안전하게 보관한다.
- 클라우드 및 가상화 환경: 클라우드 및 가상화 환경에서는 물리적인 HW에 직접 접근하거나 이를 통제하기 어렵다. 따라서 TPM을 지원하는 클라우드 서비스 또는 가상화 TPM(virtual TPM)을 사용하여 안정적인 보안 기능을 제공한다.
- 개발 및 테스트 환경: 실제 환경을 반영하는 개발 및 테스트 단계에서 TPM은 보안 메커니즘의 검증 및 실험에 사용된다.
- 키 관리: TPM을 활용하여 키의 생성, 저장 및 관리에 대한 유연한 접근이 가능하다.

특히 TPM은 클라우드 컴퓨팅에서 인증, 신뢰할 수 있는 부팅, HW root of trust, 인증, 데이터 보호, 키 생성 및 저장 등 다양한 기능이 사용되며, Fig 1은 [13]에서 분석한 TPM이 수행하는 다양한 기능의 비율을 보여준다. 또한 Table. 1. 은 TPM을 사용함으로 인해서 완화할 수 있는 위협을 보여준다.

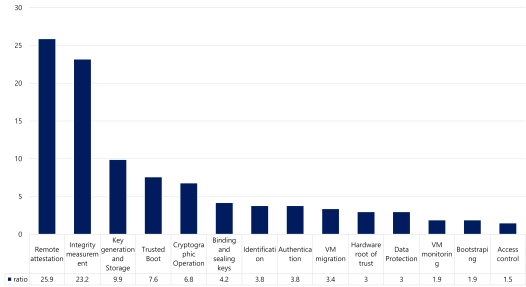

Fig. 1. The TPM performs a variety of work.

Table. 1. Threats that can be mitigated by using a TPM.

Category	Threats
Network	Man in The Middle
	Replay Attack
	Denial Of Service and distributed denial of service
	Attack to steal cryptographic keys and secrets
	Unauthorized access
	Identification problem / identity spoofing
Application	Malware
	Run time attacks to software stack
	Reversing
	Tampering
Data Tampering	Audit log records
	Measurement data
	VM and its data at the time of transmission
Data Leakage	Geolocation data

2.2 Secure Boot

Secure Boot는 컴퓨터나 기타 장치가 부팅 과정에서 신뢰할 수 있는 SW와 운영체제만 로드할 수 있도록 보장하는 보안 표준이다. Secure Boot는 부팅 과정에서 악성 코드가 실행되는 것을 방지하기 위해 설계되었고 안전한 실행 환경을 구축하기 위해 많이 사용되는 SW 안티 탬퍼링 기법이다 [4].

Secure Boot는 시스템 부팅 중에 로드되는 각 SW 이미지의 무결성을 검증하여 시스템이 변조되거나 악의적인 코드에 의한 피해 방지를 보장한다 [8].

Secure Boot의 중요성 및 기능은 다음과 같다.

- 무결성 확인: Secure Boot를 사용하면, Boot loader가 로드하는 SW 이미지가 변조되었는지 확인할 수 있다.
- 신뢰할 수 있는 부팅 과정: OEM(Original Equipment Manufacturer)은 신뢰할 수 있는 이미지만을 사용하여 기기를 부팅할 수 있다.
- 악성 코드로부터의 보호: 악의적인 코드나 RootKit과 같은 APT(Advanced Persistent Threat)의 실행을 방지한다.
- 플랫폼 보안 강화: Secure Boot는 전반적인 플랫폼 보안을 강화하여, 기기나 시스템이 공격자로부터 보호받을 수 있게 도와준다.

III. SW 안티탬퍼링 기술 개발 현황

SW 안티탬퍼링 기술은 정보나 시스템의 무결성을 보장하기 위한 디지털 환경에서 주목받는 보안 기술이다. 탬퍼링 공격이 다양한 형태로 나타나면서 이를 방어하기 위한 SW 안티탬퍼링 기술의 중요성은 더욱 높아졌다. 특히, SW 안티탬퍼링 기술은 빠르게 발전하며 다양한 연구와 적용 사례가 나타나고 있다

이 장에서는 최근 제안된 SW 안티탬퍼링 기술 개발 현황을 살펴본다.

3.1 소스코드 난독화 기반의 국내 SW 안티탬퍼링 기술

이규호 외는 무기체계 안티탬퍼링을 위해 소스코드 난독화 도구를 구현했다 [16]. 이 난독화 도구는 무기 체계에 사용되는 핵심 알고리즘 및 중요 데이터 등을 보호하기 위해 제어흐름 난독화, 데이터 난독화 기법을 활용하여 실행흐름을 복잡하게 만들어 분석을 어렵게 하며 중요한 데이터를 숨기거나 변조하였다.

무기체계 SW는 이식성이 우수한 C/C++ 언어를 주로 사용하고, 실시간 OS인 VsWorks를 주로 사용한다. 그렇기 때문에 [16]은 LLVM과 gcc 라이브러리를 활용하여 바이너리를 생성하는 프로세스를 모듈화한 후 난독화를 진행한다. Fig. 2. 는 [16]이 제안하는 난독화 기법의 모듈 및 데이터 흐름도를 보여준다.

난독화의 순서는 C/C++ 소스코드를 컴파일하여 바이너리 코드로 변환한 후 제어흐름, 데이터 난독화를 적용한다.

제어 흐름 난독화 알고리즘은 두 가지의 방법으로 난독화를 진행하며, 첫 번째로는 기본 블록에 의미 없는 블록과 조건을 삽입하는 방법을 통해 난독화를 진행한다. 두 번째로는 switch 명령문에 의미 없는

case 문을 삽입하는 방법으로 난독화를 진행한다.

데이터 난독화 기법으로는 문자열을 문자 단위로 구성하여 문자열을 버퍼에 하나씩 대입하는 명령문으로 구성되며, 문자열을 비트 단위로 나누어 구성 후 합치는 방식을 사용한다.

저자들은 제안된 난독화 기법을 평가하기위해 난독화 미적용 SW와 Low, Medium, High 3가지 수준으로 난독화가 적용된 SW 총 4가지의 Cyclomatic Complexity값을 측정하여 복잡도를 비교했다. 평가 결과 난독화 수준이 High로 갈수록 if-else와 같은 조건 분기 문의 증가로 인해 복잡도가 증가함을 보여주었다.

또한 바이너리 파일이 생성된 이후 데이터 난독화가 제대로 수행되었는지 확인하기 위해 ms-string v2.53 도구를 사용하였으며, 그 결과 난독화 대상 데이터인 2,590개 데이터를 찾을 수 없었다. 임베디드 시스템에선 SW의 크기와 시간이 성능에 영향을 주기 때문에 난독화 설정 수준에 따른 무기체계 SW의 성능 변화도 측정하였다. 성능의 변화는 난독화 수준을 높일수록 바이너리의 크기가 증가 하지만, 실제 코드가 동작하는 시간은 난독화를 수행함에 따라 큰 변화를 주지 않음을 확인하였다.

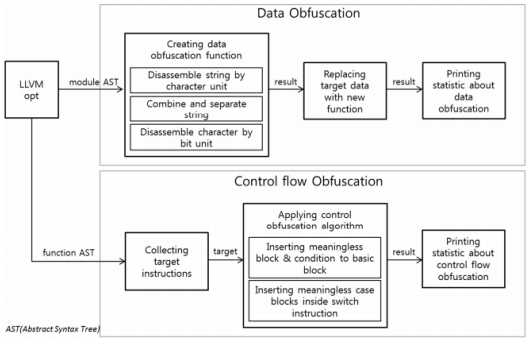

Fig. 2. Structure of obfuscation modules and data-flow

3.2 국내 SW 안티탬퍼링 기술

김주현 외는 IoT 장치에 저장되어 있는 SW와 SW에 포함된 개인정보 및 특허 기술을 보호하기 위해 안티탬퍼링 기술을 개발자가 쉽게 적용할 수 있도록 Jenkins 기반 통합 개발 환경을 구축했다 [17]. 김주현 외 의 제안기법 구현은 크게 세 가지로 구분된다.

첫 번째로는 안티탬퍼링 대상 파일 식별 기능을 추가하여 안티탬퍼링이 가능한 파일 확장자인지 식별하기 위해 파일을 식별하는 기능을 구현하였다.

두 번째로 안티탬퍼링 기능을 추가하기 위해 C/C++과 Java Script, Perl 같은 스크립트 언어를 지원하는 난독화 도구인 Stunnix Obfuscator를 호출하여 소스코드 난독화를 진행한다. 또한 동시에 컴파일된 실행파일의 리버싱 방지를 위해 UPX를 사용하여 Packing을 적용 하였다. 세 번째로 압축한 실행파일을 타겟 장치에 다운로드 하는 기능을 구현하기 위해 ftp를 이용하였다.

김주현 외[17]는 임베디드 리눅스를 사용하는 IoT 장치에서 SW 리버싱을 방지하고, 개발자가 안티탬퍼링 기법을 쉽게 적용할 수 있는 통합 개발 환경을 구축하였다. 이런 통합 개발 환경은 개발자로 하여금 SW의 개발 및 실행과 다운로드 단계까지의 SW 개발 및 사용주기 전 단계에서 안티탬퍼링을 적용할 수 있도록 하였다.

3.3 ARM TrustZone 기반 Secure boot scheme

Jiang et al. 는 ARM TrustZone 기반의 Secure boot scheme은 Xilinx zynq-zc702 보드에서의 구현을 통해 주목받고 있다 [4]. 해당 구현의 핵심은 FSBL(First Stage Boot Loader)의 로드 과정에서 시작된다. 이 과정에서 FSBL의 무결성과 탬퍼링 여부를 확인하기 위해 BootROM은 RAS, AES, HMAC (Keyed-Hash Message Authentication Code) 인증을 차례대로 평가한다.

Fig. 3. 는 Jiang et al. 의 제안기법 구조를 보여주며, 제안된 기법은 Normal world와 Secure world로 2가지로 나뉜다. Normal world에서는 Linux와 Android 같은 비 보안 운영체제가 실행되며 Secure world에서는 op-tee, sirreTEE 같은 보안 커널과 Trustlets 같은 신뢰 서비스가 실행된다. 프로세서는 주로 Normal world에서 실행되며 보안 관련 데이터와 서비스는 Secure world에 저장된다. 프로세서는 필요에 따라 두 월드 간 전환을 할 수 있으며 해당 전환은 모니터 모드에서 관리되며 특정 조건에서만 가능하다.

로드된 FSBL은 이후 U-Boot로의 제어 전달 이전에, U-Boot 이미지의 안전성을 평가하기 위해

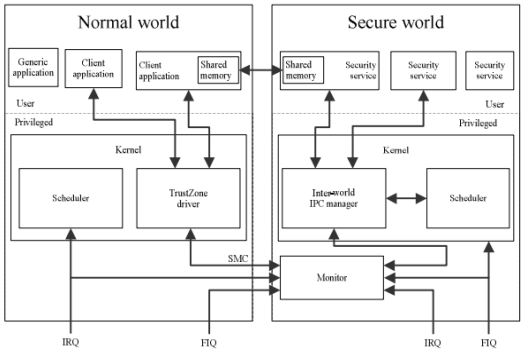

Fig. 3. The architecture of the isolated execution environment

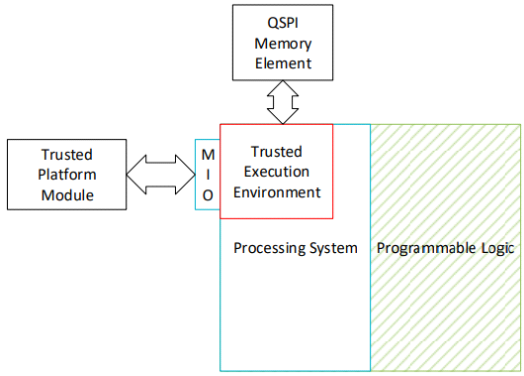

Fig. 4. Proposed System Architecture.

AES 및 HMAC 인증을 수행한다. 이와 같은 연속적인 검증 단계를 통해, Trust chain을 구축하며 실행 환경이 신뢰할 수 있는지 확인한다.

Trust chain의 검증은 Demper-Shafer theory를 활용하여 수행되며, 이를 통해 TCG (Trusted Computing Group)가 정의한 Trust chain보다 더 높은 신뢰도를 보장할 수 있음을 입증하였다.

3.4 TPM 기반의 Root of Trust 설정 및 Secure First Stage Boot Loader

Siddiqui et al. 은 TPM을 활용하여 Hardware level에서 Root of Trust를 구축하고 air updates를 가능하게 하는 architecture와 Secure FSBL를 제안했다 [7]. Siddiqui et al. 의 제안기법은 reconfigurable computing architectures의 발전과 자동차와 같은 안전에 직결된 인프라에 중요하다.

Target architectures의 발전에 따라, target platform에서도 remote update 기능이 가능해졌지만, 이러한 업데이트 과정은 공격자에 의한 remote hijacking에 노출되어 있는 위험이 있다. 따라서, 업데이트 및 인증 과정에서 사용되는 각 요소의 보안성을 평가하는 것이 중요하다.

Fig. 4. 는 Siddiqui et al. 이 제안한 시스템 구조로 TPM과 암호화를 사용하여 HW 수준에서의 Root of Trust를 설정하고 HW 재구성을 위한 안전한 OTA (Over-the-Air) 업데이트를 가능하게 한다.

Siddiqui et al. 는 Digital certificates, key, encrypted configuration 및 Bitstream symmetric encryption key와 같은 요소들의 안전성과 무결성은 업데이트 및 인증 과정의 신뢰성을 결정짓기 때문에 해당 요소들을 중점적으로 평가하였다.

3.5 Hybrid Boot Scheme

Ling et al. 는 IoT 시스템의 무결성을 강화하기 위해 ARM TrustZone을 기반으로 Hybrid Secure boot를 제안했다 [9].

Fig. 5.은 Ling et al. 이 제안한 Hybrid booting sequence이다. IoT 시스템에 전원이 켜지면 FSBL이 먼저 실행되고, FSBL이 SSBL(Second stage Boot Loader)를 로드하고 무결성을 확인한 뒤 서명이 확인되면 컨트롤을 SSBL로 넘긴다. 이후 SSBL은 나머지 펌웨어 이미지인 secure OS 커널, rich OS 커널, 파일 시스템 이미지를 메모리에 로드한 뒤, 세 종류의 이미지들을 검증하여 Secure world의 로드 타임 무결성을 강화한다. 이 Secure boot을 통해 secure OS 커널 이미지의 무결성이 검증되기 때문에 Normal world의 Trusted boot을 위한 trusted based로 사용될 수 있다. Trusted boot 중에 secure OS 커널은 rich OS 커널과 파일 시스템 이미지를 모두 검증하고, 이후 rich OS 커널로 control을 넘긴다. rich OS가 시작되면 검증 결과가 원격 인증서버로 전송되어 Normal world 로드 시 무결성을 검증하게 된다.

그렇기 때문에 Trusted Secure world가 Normal world의 프로세스 무결성을 측정하고 검

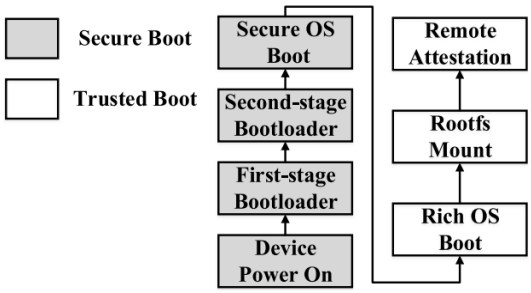

Fig. 5. Proposed Hybrid Booting Sequence

증하면서 시스템의 런타임 프로세스 무결성을 강제하게 된다.

Ling et al. 은 제안기법의 검증을 위해 Secure boot 모듈과 Trusted boot 모듈에 소요되는 시간과 SSBL과 Secure OS 의 총 부팅 시간을 측정했다. 추가로 Trusted Boot 과정에서 발생하는 무결성 측정 및 증명 방법에 대한 성능 오버헤드를 측정했다.

IV. SW 안티탬퍼링 평가 방안에 대한 고찰

이 장에서는 TPM을 활용한 SW 안티탬퍼링 기술에 대한 체계적인 검증 방법 및 평가 기준의 필요성을 강조한다. 최근 국내에서 탬퍼링 공격에 대한 문제가 대두되었다 [1,2,3]. 특히 높은 수준의 보안이 요구되는 HW 및 SW 시스템들의 보호를 위해서는 탬퍼링 대응 기술의 개발이 필수적이라고 할 수 있기 때문에, 국/내외에서 다양한 SW 안티탬퍼링 연구들이 제안되었다 [4,7,9,16,17]. 이와 관련하여 새롭게 제안된 SW 안티탬퍼링 기술들의 성능과 보안성을 평가하기 위한 시험 및 검증 방법론에 대한 필요성도 함께 증대되고 있다. 그렇기 때문에 철저한 검증 체계 및 평가 기준 수립에 대한 연구가 필요한 상황이다.

하지만 현 상황에서는 새롭게 제안될 SW 안티탬퍼링 기법들을 체계적으로 평가할 수 있는 검증 방법이나 평가 기준이 부재한 상황이다. 실제로 미 국방성의 SW 안티탬퍼링 기술 지침이 존재하지만, 해당 지침에서는 SW 안티탬퍼링 기술 적용을 위한 절차만 명시하고 있다. 또한 현재 배포된 다양한 보안 가이드 및 지침들에 서술된 일부 기준들은 너무 단순하거나 모호한 부분이 존재한다 [11,12,14,18]. 이런 모호한 보안 가이드 및 지침은 급변하는 실상황에 빠르게 대응하기 어렵다. 따라서 국제 표준화 기구와 국내/외 기관들이 SW 안티탬퍼링 기술을 평가 및 검증하기 위한 명확한 기준과 평가 방안을 설립하여 안전한 디지털 환경을 위해 보안 가이드라인과 지침 및 평가 기준을 제공해야 한다.

4.1 SW 안티탬퍼링 평가 방안의 필요성

디지털 환경은 지속해서 발전하고 변화하면서, 보안에 대한 요구사항과 위협도 함께 다양해지고 있다 [10]. 하지만 일부 가이드라인이나 지침들에 서술된 기준들은 모호하거나 세부적인 평가 체계가 부족하기 때문에 급변하는 실상황에 빠르게 대응하기 어렵다.

IoT 공통 보안 가이드의 IoT 플랫폼 보안 요구사항에는 'oneM2M 시스템은 데이터의 기밀성/무결성을 보장할 수 있어야 한다.'라고 적혀있으며 [14], 어떤 방법을 통해 데이터의 기밀성과 무결성을 보장해야 하는지, 등 기타 세부적인 사항들은 언급하지 않고 있다.

SW 개발 보안 가이드 [11] 의 3장 2.8 중요 정보 전송의 설계 시 고려 사항에는 '안전한 암호 모듈로 암호화한 뒤 전송하거나 안전한 통신 채널을 사용하도록 설계한다.'라고 적혀있다. 하지만 이때 사용되는 안전한 암호 모듈에 대한 기준과 암호화 연산 및 방식에 대한 상세한 설명은 언급되지 않았다.

마지막으로 미 국방성의 안티탬퍼링 기술 지침 [15]에는 안티템퍼링 기술에 대한 체계적인 평가 기준이 부재한 것을 알 수 있다. 따라서 SW 안티탬퍼링 기법의 보안 수준을 향상하기 위해서는 보다 체계적인 평가 기준이 필요하다.

체계적인 평가 기준이 수립되면 각 모듈의 보안 수준을 더욱 정밀하게 비교하고 평가할 수 있게 된다. 또한, SW 안티탬퍼링은 다양한 시스템 및 SW와의 호환성을 가져야 하므로 호환성 체크의 기준 또한 명확해져야 한다. 마지막으로, SW의 품질을 보장하기 위해 그 품질을 측정하고, 설정된 표준에 부합하는지 확인하는 과정이 필요하다.

V. SW 안티탬퍼링 기술의 포괄적인 평가 방안

높은 수준의 보안이 요구되는 HW/SW 시스템들의 보호를 위해선 SW 안티탬퍼링 기술의 개발이 필수적이다.

하지만 SW 안티탬퍼링 기술을 평가하기 위한 평가 기준 및 검증 방법이 포함된 가이드라인이 존재하

지 않는 것을 확인하였다. 또한 기존에 존재하는 SW 안티탬퍼링 지침[15]에는 SW 안티탬퍼링 기술에 대한 시험 및 검증 방안이 존재하지 않고, SW 안티탬퍼링 기술 적용을 위한 절차만 언급되어 있다.

결국 SW 안티탬퍼링 기술을 평가하기 위한 체계화된 검증 방법 및 평가 기준이 존재하지 않은 것을 알 수 있으며, 그렇기 때문에 SW 안티탬퍼링 기술의 성능과 보안성을 평가하기 위한 체계적인 검증 및 평가 기준이 필요하다.

우리는 최신 SW 안티탬퍼링 연구를 통해 SW 안티탬퍼링 기법을 종합적으로 평가할 수 있는 포괄적이고 체계적인 평가 방안을 구성한다.

연구들이 사용하는 다양한 검증 방법들을 알아본 결과, Stress Test, SW 업데이트의 인증 과정, 서버의 신원 검증, 부팅 시간 측정, 자원 사용량 평가, 정/동적 분석, 무결성 검증 등 다양한 검증 방법을 사용하는 것을 알 수 있었다.

결과적으로 각 연구에서 사용한 검증 방법들은 각자의 제안된 SW 안티탬퍼링 기법을 검증하고 세부적으로 평가할 수 있지만 SW 안티탬퍼링 기술에 대한 전반적인 평가 기준으로 삼기엔 부족함이 있다.

본 논문은 최신 SW 안티탬퍼링 연구들이 검증 및 평가에 사용하는 다양한 방법들을 이용하여 각 평가 방안을 기능, 구현, 성능 세 분류로 구분 함으로써 SW 안티탬퍼링 기술에 대한 전반적인 평가를 할 수 있는 종합적이고 세부적인 SW 안티탬퍼링 평가 방안을 제시한다.

5.1 기능적 신뢰성

기능적 신뢰성은 SW 안티탬퍼링 기술이 제대로 작동하는지 확인하고 시스템 요구 사항을 정확히 만족하는지 검증한다.

- Stress Test: 시스템에 과도한 부하를 주어 안티탬퍼링 기술이 극한의 조건에서도 안정적으로 작동하는지 평가한다.
- 기능 체크: 안티탬퍼링 기술의 주요 기능을 모두 테스트하여 요구 사항이 제대로 충족되는지 확인한다. 예를 들어, SW 업데이트의 인증 과정, 서버의 신원 검증 등이 제대로 작동하는지 검증한다.
- 신뢰성 평가: 업데이트된 SW가 발행자로부터 제공된 것인지, 데이터 및 코드가 무단으로 변조되지 않았는지와 같은 무결성을 검증하여 신뢰성을 평가한다.

Siddiqui etl al. 는 Secure Boot process와 펌웨어 업데이트를 수행하기 위해서 Boot level에서 TPM 기능과 First Secure Boot Loader의 Driver와 통합하여 bit stream을 확인함으로써 Anti tampering을 수행했다 [7]. Siddiqui etl al. 는 실험을 통해 인증과 업데이트 과정 중에 의도적으로 계산된 해시값과 매치되지 않는 올바르지 않은 해시값을 로드하여 부팅 과정을 중지하여 fallback process로 전개하는 결과를 보여주었다.

5.2 구현적 안정성

구현적 안정성은 SW 안티탬퍼링 기술이 안정적으로 구현되었는지, 코드 레벨에서의 오류나 취약점이 없는지 확인한다.

- Bug Check: 일반적인 버그 및 안티탬퍼링 기술과 관련된 특수한 오류를 찾아내어 수정한다.
- API Check: 사용된 API가 안정적으로 작동하는지 확인하며, 안티탬퍼링 기술과 충돌하는 API가 없는지 검증한다.
- 정적 분석: 소스 코드를 직접 분석하여 오류, 취약점, 불필요한 코드 등을 찾아낸다.
- 동적 분석: 실행 중인 프로그램의 동작을 모니터링하여 런타임 중 발생할 수 있는 문제를 탐지한다.

IoT 공통 보안 가이드의 6-1 시큐어 코딩에서는 의도하지 않은 방법으로 API를 사용하거나, 보안에 취약한 API를 사용할 수 있는 보안 약점을 API 오용으로 정의하였으며, 인증, 접근제어 등 보안 기능을 적절하지 않게 구현할 시 발생하는 보안 약점과 개발자가 범할 수 있는 코딩 오류로 인해 유발되는 보안 약점 등을 정의하고 있다 [14].

5.3 성능의 효율성

성능의 효율성은 SW 안티탬퍼링 기술이 제대로 작동하는지 확인하고 시스템 요구 사항을 정확히 만족하는지 검증한다.

- 부팅 시간 측정: 안티탬퍼링 기술이 적용된 SW나 시스템의 부팅 시간을 비교하여 성능 향상 또는 저하를 평가한다. 기존 시스템과 안티탬퍼링 기술이 적용된 시스템 간의 시간 차이를 분석함으로써, 기술 적용의 효율성을 판단할 수 있다.
- 응답 시간 측정: 프로세스의 요청에 대한 시스템

의 반응 시간을 측정하여, 성능의 변화를 파악한다. 안티탬퍼링 기술 적용이 시스템의 반응 속도에 어떠한 영향을 미치는지 평가한다.

- 자원 사용량 평가: CPU, 메모리, 디스크 등의 자원 사용량을 모니터링하여, 안티탬퍼링 기술 적용 전과 후의 자원 효율성을 비교한다. 과도한 자원 사용은 시스템의 전반적인 성능 저하를 초래할 수 있으므로, 이를 통해 기술의 적합성을 검토한다.
- 확장성 및 유연성: 안티탬퍼링 기술이 다양한 시스템 및 애플리케이션과의 통합이 용이한지 평가하고, 모듈화하여 확장 또는 수정이 쉬운지 평가한다.

이와 같은 평가 기준 및 방법을 통해, 안티탬퍼링 기술의 실제 적용 가능성과 효율성을 체계적으로 검토해야 한다.

Wang et al. 는 3가지 방식의 Hardware, Software, hybrid Secure boot에 대해 각각 사용되는 리소스의 양, Secure boot에 소요되는 시간, Trusted의 여부를 측정함으로써, Secure boot에 소비하는 리소스 양과 소요되는 시간이 반비례하는 것을 관찰하였다 [5]. Siddiqui etl al. 은 bit stream 누적 해시를 계산하기 위해 TPM2.0 locality4를 사용하였으며, 3.85MB 크기의 비트스트림 파일에 대한 해시 계산 시간은 40초가 소요되었다 [7].

VI. 결 론

디지털 환경이 급속도로 성장하며 변화하는 가운데, HW 및 SW 보안에 대한 중요성이 대두되고 있다. 탬퍼링은 HW 및 SW 시스템의 무결성을 손상시키고 데이터를 유출할 수 있는 불법적인 조작이다. TPM은 템퍼링 공격을 완화하고, HW 시스템의 무결성을 보장하며, 보안성을 강화하기 위한 다양한 기능을 제공한다. 다양한 SW 안티탬퍼링 기술들은 탬퍼링 공격을 방어하고, 완화하기 위해서 TPM을 활용하고, 소스코드를 난독화하는 등 다양한 방식으로 제안되었다. SW 안티탬퍼링 기술은 개인과 기업의 정보를 보호하는 핵심 기술로 부상하게 되었다.

2장에서는 TPM의 구성 요소, 장점 및 사용 사례와 함께 SW 안티탬퍼링 기술 중 가장 널리 사용되는 Secure Boot의 기본 개념과 중요한 기능들을 알아보았다.

3장에서는 최근 국내에서 제안된 SW 안티탬퍼링 연구들과 해외에서 제안된 TPM 기반의 SW 안티탬퍼링 연구를 조사했다. 최신 SW 안티탬퍼링 기술들의 동향을 살펴보고, 연구들에서 사용된 평가 방안 및 검증 기법을 조사했다.

조사 결과 SW 안티탬퍼링 연구들은 저자들의 제안기법을 평가 또는 검증하기 위해 제각각 다양하고 세부적인 검증 기법을 사용했다. 추가로 현존하는 안티탬퍼링 지침이 존재하지만, 해당 지침에는 안티탬퍼링 기술의 적용을 위한 결정 절차만 명시하고 있는 것을 확인하였다. 따라서 SW 안티탬퍼링 기법의 상세하고 포괄적인 검증 방법은 존재하지 않는 것을 확인했다.

그렇기 때문에 기존에 제안된 다양한 SW 안티탬퍼링 기법들뿐만 아니라 향후 새롭게 제안될 SW 안티탬퍼링 연구를 종합적으로 평가 및 검증할 수 있는 체계적이고 포괄적인 방법이 필요하다.

따라서 본 논문에서는 앞서 제안된 SW 안티탬퍼링 기법들이 검증에 사용한 기법들을 이용하고, 이 기법들을 종합하여 세 가지(기능적 신뢰성, 구현적 안정성, 성능의 효율성)로 세분화함으로써 SW 안티탬퍼링 기법에 대한 전반적인 평가를 할 수 있는 보다 체계적이고 포괄적인 SW 안티탬퍼링 평가 방안을 제시한다.

향후 연구로는 본 논문에서 제시한 평가 방안을 위한 세 가지 항목별 평가 지표와 상세한 측정 방법이 포함된 평가 및 검증 방안을 제시할 것이다.

References

[1] datasom.co.kr, "Data SOM" http://www.datasom.co.kr/news/articleView.html?idxno=125955, Accessed Sep. 2023.

[2] bboannews.com, "Security News" https://m.boannews.com/html/detail.html?mtype=1&idx=122264, Accessed Sep. 2023.

[3] Chakraborty, Dhiman, Lucjan Hanzlik, and Sven Bugiel, "{SimTPM}: User-centric {TPM} for Mobile Devices," USENIX Security Symposium (USENIX Security 19), pp.533-550, Aug. 2019.

[4] Jiang, H., Chang, R., Ren, L., and Dong, W, "Implementing an arm-based secure boot scheme for the isolated execution environment," International Conference on Computational Intelligence and Security (CIS), pp.336-340, China, Dec. 2017.

[5] Wang, R. and Yan, Y. "A Survey of secure boot schemes for embedded eevices," 24th International Conference on Advanced Communication Technology (ICACT), pp.224-227. Feb. 2022.

[6] Jyothi, T., and Shilpa Jain. "TPM based Secure Boot in Embedded Systems," International Conference on Secure Cyber Computing and Communication (ICSCCC), IEEE, pp.786-790. May. 2023.

[7] Siddiqui, Ali Shuja; Gui, Yutian; Saqib, Fareena. "Secure boot for reconfigurable architectures." Cryptography, vol. 4, no. 4, pp. 26, Dec. 2020

[8] Zimmo, S., Refaey, A., and Shami, A. "Trusted Boot for Embedded Systems Using Hypothesis Testing Benchmark," Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-2. Aug. 2020.

[9] Ling, Zhen, et al. "Secure boot, trusted boot and remote attestation for ARM TrustZone-based IoT Nodes," Journal of Systems Architecture vol. 119, pp. 102240, Oct. 2021

[10] Qualcomm.com, "Secure Boot and Image Authentication" https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/secure-boot-and-image-authentication-version_final.pdf, Accessed Sep. 2023.

[11] KISA.or.kr. "Software Development Security Guide" https://www.kisa.or.kr/2060204/form?postSeq=5&page=1, Accessed Aug. 2023.

[12] iec.ch, "IEC 60335-1" https://webstore.iec.ch/publication/67569, Accessed July. 2023.

[13] Hosseinzadeh, Shohreh, et al. "Recent trends in applying TPM to cloud computing," Security and Privacy, vol 3, no.1, pp. e93. Jan. 2020

[14] kisa.or.kr, "Common Security Guide" https://www.kisa.or.kr/2060205/form?postSeq=2&page=4#fnPostAttachDownload, Accessed Jan. 2024.

[15] Statement of Anti-Tamper (AT) Measures in the Letter of Offer and Acceptance(LOA), DSCA 00-07, 2000. Department of Defense DIRECTIVE : Anti-Tamper (AT), DoD Directive 5200.47E, 2015.

[16] Gyuho Lee, Jaegwan Yu, Insung Kim, and Taekyu Kim, "Implementation of Software Source Code Obfuscation Tool for Weapon System Anti-Tampering," Journal of KIISE, 46(5), pp. 448-456, Dec. 2019

[17] Juhyeon Kim, Shinho Lee, Ara Hur, and Yeonseung Ryu, "Development of an Anti-Tampering Tool for Embedded Linux Environment," Korea Computer Congress, (Online) pp. 1280-1282, July. 2020.

〈저자소개〉

이 동 호 (Dongho Lee) 학생회원
2016년 2월: 대덕대학교 정보보안학과 전문학사
2020년 2월: 공주대학교 컴퓨터공학과 학사
2022년 2월: 숭실대학교 융합소프트웨어학과 석사
2022월 3월~현재 : 숭실대학교 소프트웨어학과 박사과정
〈관심분야〉 모바일 보안, 컴퓨터 보안, 소프트웨어 보안 등

반 영 훈 (Younghoon Ban) 학생회원
2020년 2월: 청운대학교 컴퓨터공학과 학사
2022년 2월: 숭실대학교 융합소프트웨어학과 석사
2022월 3월~현재 : 숭실대학교 소프트웨어학과 박사과정
〈관심분야〉 AI 보안, 프로그램 분석, 소프트웨어 보안 등

임 재 덕 (Jae-Deok Lim) 종신회원
1999년 2월: 경북대학교 전자공학과 졸업
2001년 2월: 경북대학교 전자공학과 석사
2013년 8월: 충남대학교 컴퓨터공학과 박사
2000년 12월~현재: 한국전자통신연구원 정보보호연구본부 책임연구원
〈관심분야〉 IoT 보안, 운영체제 보안, 네트워크 보안, 접근제어 등

조 해 현 (Haehyun Cho) 종신회원
2013년 2월: 숭실대학교 컴퓨터학부 학사
2015년 2월: 숭실대학교 컴퓨터학과 석사
2021년 2월: Arizona State University, Computer Science 박사
2021년 3월~현재: 숭실대학교 소프트웨어학부 조교수
〈관심분야〉 시스템 보안, 취약점 탐지, 프로그램 분석, AI 보안