

하드웨어 보안 소자 결속 기반 디지털 콘텐츠 복제 방지 방법

이 효 승¹ · 홍 진 우² · 정 지 우² · 이 준 호³ · 한 기 태^{4*}

¹㈜네오와인 대표이사 ²㈜네오와인 연구원 ³University of South California 학부생 ⁴㈜네오와인 전무이사

Digital Content Copy Protection Method Using Hardware Security Element Binding

Hyo Seung Lee¹ · Jin Woo Hong² · Ji Woo Jeong² · Jun Ho Lee³ · Gi-Tae Han^{4*}

¹CEO, Neowine Co.,Ltd, Seongnam 13595, Korea

²Researcher, Neowine Co.,Ltd, Seongnam 13595, Korea

³Undergraduate, University of Southern California, Los Angeles CA 90033, USA

⁴Executive Director, Neowine Co.,Ltd, Seongnam 13595, Korea

[요 약]

디지털 콘텐츠는 복제의 한계비용이 영에 가까워 원본 콘텐츠가 한 번 노출되면 무제한 재배포가 가능하다. 이에 대한 복제 방지 방안으로 다양한 소프트웨어 기반 방식과 하드웨어 기반 방식들이 연구되었다. 기존 소프트웨어 기반 방식은 일반적으로 실행 메모리 덤프에 의하여 키 추출의 위험성이 존재하고, 하드웨어 기반 방식은 보안 구현이 SoC 설계나 플랫폼에 종속적이라는 한계가 있다. 본 논문은 외부로 반출되지 않는 마스터키를 전용 하드웨어 보안 소자의 비휘발성 메모리에 두고, 소자 내부에서만 디바이스 고유키를 유도하여 콘텐츠를 정품 장치에 결속시키는 복제 방지 방법을 제안한다. 제안한 방법은 서버와 단말이 각자의 보안 소자에서 동일한 키를 독립적으로 유도하여 별도의 키 전송 없이 상호 정합으로 정품 여부를 입증하고, 기존 방법들과 비교하여 키 추출의 저항성과 오프라인에서의 동작, 키 갱신이나 폐기 및 기존 장치에의 적용성 측면에서 우수성이 있음을 보인다.

[Abstract]

Digital content has a near-zero marginal cost of reproduction, enabling unlimited redistribution once the original content is exposed. To prevent unauthorized copying, various software-based and hardware-based protection methods have been studied. Conventional software-based approaches are generally vulnerable to key extraction through runtime memory dumps, whereas hardware-based approaches are often limited by their dependence on specific SoC designs or platform-specific security architectures.

This paper proposes a content copy protection scheme in which a master key that never leaves a dedicated hardware security module is securely stored in its non-volatile memory, while a device-specific key is derived exclusively within the secure hardware to bind digital content to authorized devices. In the proposed scheme, both the server and the client independently derive the same cryptographic key within their respective secure hardware modules, thereby verifying device authenticity through key agreement without transmitting cryptographic keys between them. Comparative analysis shows that the proposed scheme provides stronger resistance to key extraction than conventional methods while supporting offline operation, efficient key renewal and revocation, and compatibility with existing devices.

색인어 : 디지털 콘텐츠 복제 방지, 하드웨어 보안소자, 장치 결속, 디지털 권리 관리, 신뢰 점

Keyword : Digital Content Copy Protection, Hardware Security Element, Device Binding, Digital Rights Management, Root of Trust

<http://dx.doi.org/10.9728/dcs.2026.27.9.2633>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 30 July 2026; **Revised** 21 August 2026

Accepted 01 September 2026

***Corresponding Author, Gi-Tae Han**

Tel: ※ 개인정보 표시제한

E-mail: gthan@neowine.com

I. 서론

스트리밍과 다운로드를 통한 디지털 콘텐츠 유통이 보편화되면서, 정당한 이용 권한이 없는 사용자에 의한 무단 복제와 재배포가 콘텐츠 산업의 지속적인 위협으로 남아 있다. 물리적 매체와 달리 디지털 콘텐츠는 완전한 복제본을 생성하는데 드는 비용이 사실상 존재하지 않으므로, 보호되지 않은 평문이 한 번이라도 노출되면 그 콘텐츠의 상업적 가치는 급격히 훼손된다.

이러한 위협에 대응하기 위하여 디지털 권리 관리(DRM: Digital Right Management)를 비롯한 다양한 복제 방지 기술이 그동안 사용되어왔다. 이들 기술은 ROT(Root of trust), 즉 전체 보안 체계가 의존하는 최상위 비밀키를 어디에 두고 어떻게 보호하는가에 따라 크게 소프트웨어 기반 방식과 하드웨어 기반 방식으로 구분된다. 소프트웨어 기반 방식은 별도의 하드웨어 없이 기존 장치에 즉시 적용할 수 있어 배포성이 높은 반면, 보호용 키와 복호 로직이 공격자가 완전히 관찰하고 조작할 수 있는 호스트 환경 내부에 존재한다는 구조적 취약점을 가진다.

본 논문은 전용 하드웨어 보안 소자(ALPU-K)를 표준 직렬 인터페이스로 서버와 단말 양측에 결합하여, 외부 라이선스 서버에 대한 상시 의존이나 별도의 인증 절차 없이 콘텐츠를 정품 장치에 결속시키는 복제 방지 방법을 제안한다. 본 논문의 기여는 다음과 같다. 첫째, 마스터키가 소자 외부로 반출되지 않고 디바이스 고유키가 소자 내부에서만 유도되는 키 관리 구조와, 난수 블라인딩을 통한 버스 노출 차단 및 상호 정합 기반의 암묵적 정품 인증 절차를 제시한다. 둘째, 대표적 소프트웨어 기반 방식인 화이트 박스 암호 기반 방식과 소프트웨어 전용 DRM 및 하드웨어 신뢰점 기반의 TEE(Trusted Execution Environment)와 TPM(Trusted Platform Module)을 분석하고, 제안 방법과의 보안성 및 환경 적응성 등을 비교하여 제안한 방법의 우수성을 살펴본다.

II. 관련연구

제안하는 하드웨어 보안 소자 기반 복제 방지 방법과의 비교를 위하여 본 장에서는 기존의 소프트웨어 기반 콘텐츠 복제 방지 방식과 소프트웨어 공격에 대한 저항성이 높게 평가되는 하드웨어 신뢰점 기반의 TEE와 TPM 방식 및 유선방송에서의 콘텐츠 보호 방법에 대하여 살펴본다.

2-1 소프트웨어 기반 복제 방지 방식

본 절에서는 비교 대상으로 삼을 두 가지 대표적 소프트웨어 기반 방식을 살펴보고, 각 방식에 대한 취약점을 정리한다.

두 방식은 모두 암호·복호화 연산이 호스트의 소프트웨어와 메모리 안에서 이루어진다는 공통점을 가진다.

1) 화이트 박스 암호(White-Box Cryptography) 방식

화이트 박스 암호는 공격자가 실행 환경 전체를 통제하고 모든 중간 연산을 관찰할 수 있는 이른바 화이트 박스 공격 모델을 가정하고도 소프트웨어에 내장된 비밀키가 추출되지 않도록 설계하는 기법이다. Chow 등은 2002년 AES와 DES에 대한 최초의 화이트 박스 구현을 제안하였으며[1],[2], 라운드 키를 무작위 선형과 비선형 인코딩으로 감춘 록-업 테이블 네트워크로 암호 연산을 대체하여 키 추출을 어렵게 하였다. 이 방식은 별도 하드웨어 없이 키를 소프트웨어에 은닉할 수 있어 DRM 클라이언트 등에 널리 응용되었다.

그러나 화이트 박스 구현의 안전성은 이후 정공법 분석에 의하여 크게 약화되었다. Billet 등[3]은 2004년 이른바 BGE(Billet, Gilbert, and Ech-Chatbi) 공격을 제시하여, Chow 등이 제안한 화이트 박스 AES 한 개에 대하여 약 230의 작업량만으로 AES 키를 복원할 수 있음을 보였다. 이는 록-업 테이블에 내장된 키가 수학적 구조 분석을 통하여 사실상 추출될 수 있음을 의미한다. 또한, Bos 등[4]은 구현의 내부 구조를 알지 못하더라도 실행 중 중간값의 상관을 분석하는 차분 계산 분석(DCA: Differential Computation Analysis)으로 다수의 화이트 박스 구현에서 키를 자동 복원할 수 있음을 보였다. 요컨대 화이트 박스 암호는 키를 소프트웨어 안에 두는 한 논리적으로 안전성이 결여됨을 알게 하였다.

2) 온라인 라이선스 서버 기반 소프트웨어 전용 DRM 방식

서버 기반 소프트웨어 DRM 방식은 콘텐츠를 공통 키로 암호화하여 배포하고, 재생 시점에 온라인 라이선스 서버로부터 복호화 키를 발급받아 소프트웨어 복호화 모듈에서 복호화하는 구조이다. 대표적으로 널리 쓰이는 상용 DRM의 소프트웨어 전용 보안 등급(예: Widevine L3)이 여기에 해당한다. 이 등급에서는 모든 암호 연산 및 키 저장과 콘텐츠 복호가 신뢰 실행 환경(TEE) 없이 호스트 프로세스의 메모리 안에서 수행된다[5],[6].

이러한 소프트웨어 전용 구조의 근본적 취약점은, 코드 난독화가 적용되더라도 개인키와 복호 키가 실행 중 어느 시점에는 반드시 프로세스 메모리에 존재하고 복호 알고리즘이 CPU에서 실행된다는 점이다[6]. 공개된 개념 증명들은 브라우저의 암호화 미디어 확장(EME: Encrypted Media Extensions) API 호출을 가로채 라이선스 교환을 포착하고, 역공학으로 확보한 키 소재로 콘텐츠 키를 평문으로 복원하여 사실상 무보호(clearkey) 상태로 전락시킬 수 있음을 보였다[5],[6]. 나아가 일부 연구는 장치 키박스가 메모리에서 그대로 덤프될 수 있는 안전하지 않은 저장 문제를 지적하였고, 관련 취약점은 공개 취약점 식별 체계인 CVE(Common Vulnerabilities and Exposures)에서 CVE-2021-0639로

할당되었다[7]. 또한, 이 방식은 키 발급을 위하여 라이선스 서버에 상시 접속하여야 하므로, 망 분리와 차단 환경에서는 동작 자체가 불가능하다.

2-2 하드웨어 신뢰점 기반 방식

1) TEE 방식

TEE는 메인 프로세서 내부를 일반 실행 영역(REE: Rich Execution Environment)과 보안 실행 영역(Secure World)으로 하드웨어적으로 분리하고, 키 관리 및 복호와 같은 민감 연산을 보안 실행 영역에서 수행하도록 하는 기술이다. 두 영역의 격리는 프로세서의 특권 상태, 메모리 컨트롤러 및 버스 수준에서 강제되므로, 일반 영역의 운영체제가 권한 상승이나 루팅에 의하여 완전히 장악되더라도 보안 영역의 메모리에 접근할 수 없으며, 응용 프로그램은 연산 결과만을 전달받을 뿐 키 자체를 관측할 수 없다. 대표적 구현으로 ARM TrustZone, Intel SGX, Apple Secure Enclave 등이 있고, Widevine DRM의 L1 등급은 콘텐츠 키를 TEE 내부에서 처리할 것을 요구한다[8],[9]. 이처럼 TEE는 키를 사용자 접근 가능한 메모리에 노출을 시키지 않으므로 소프트웨어 공격에 대한 저항성이 높으나, 보안 영역이 메인 프로세서와 동일한 실리콘 및 캐시와 메모리 버스를 공유하는 구조적 특성으로 인하여 신뢰 컴퓨팅 기반(TCB: Trusted Computing Base)의 규모가 크고 부채널 공격 및 보안 영역 내부 구현 취약점의 여지가 남는다. 또한, TEE의 지원 여부는 SoC 설계 단계에서 결정되므로, 해당 기능을 갖추지 못한 기존 출하 장치에 이를 소급 적용하는 것이 불가능하다는 제약이 있다.

2) TPM 방식

TPM은 메인보드에 내장되거나 별도 칩으로 장착되는 하드웨어 보안 모듈로서, 암호 키의 생성과 저장 및 봉인(sealing), 원격 증명(remote attestation) 등을 하드웨어 수준에서 제공하며, 부팅 과정의 각 구성 요소를 순차적으로 측정하여 그 결과를 플랫폼 구성 레지스터(PCR: Platform Configuration Register)에 누적 저장하고, 이를 통해 부팅 무결성을 검증한다.

TPM 기반 DRM은 콘텐츠 복호 키를 특정 플랫폼의 부팅 무결성이나 TPM 내부 키에 결속시켜, 인가된 상태의 정품 장치에서만 복호가 이루어지도록 한다. 즉, 신뢰점이 모듈 내부의 비휘발성 메모리에 존재하여 순수 소프트웨어 공격만으로는 원칙적으로 반출되지 않기 때문에, 이러한 면에서는 신뢰점을 실리콘 내부에 두는 TEE나 제안한 방법과 방향을 같이 한다고 볼 수 있다. 다만 TPM은 PCR, 봉인, 원격 증명 및 키 계층에 이르는 방대한 명령 집합과 이를 구현하는 펌웨어 스택에 안전성이 종속되고, 호스트와 TPM을 잇는 직렬 버스(LPC/SPI 등)가 별도 보호 없이 사용될 경우 버스 도청으로 봉인 해제된 키가 노출되는 공격과 펌웨어 취약점을 통한 우

회가 보고된 바 있다[10],[11]. 또한, TPM은 플랫폼 제조 단계에서 통합되어야 하므로 이미 보급된 장치에 사후적으로 도입하기가 어렵다는 배포성의 제약이 있다.

2-3 유선방송의 콘텐츠 보호방법

유선방송의 콘텐츠 보호방법은 CAS(Conditional Access System, 제한수신시스템) 체계 안에서 특정 조건을 만족하는 수신기에서만 시청을 허용하는 형태로, 암호화된 데이터를 보내되 암호화를 풀 수 있는 열쇠를 선별적으로 주는 방식으로 운영된다. CAS는 고유키(Unique Address: UA key)가 서버 장치와 사용자 단말 장치 양측에 존재하며, CAS 헤드의 SAS(Subscriber Authorization System)가 UA(Unique Address, 카드 일련번호)와 고유키의 대응 관계를 보관하고, 실제 키 값은 HSM(하드웨어 보안모듈) 안에 격리된다.

서버장치에서 콘텐츠를 암호화하기 위하여는 CW(Control Word)가 요구되며, CW와 사용자 권한 등을 암호화하는데 계층적 키 관리 체계(UK:고유키, SK:서비스키, CW:제어단어 등)를 갖는다. 이 CW는 SK(서비스키)로 암호화되어 ECM(Entitlement Control Message)에 실려 나가며, SK 자체와 가입자의 시청 권한 정보는 수신기별 고유키(UK)로 암호화되어 EMM(Entitlement Management Message)에 실려 나간다. 사용자 단말(수신기) 측에서는 복호화가 역방향으로 진행되어, 스마트카드가 자신의 고유키로 EMM을 풀어 가입 권한과 SK를 확보한 다음, 그 권한이 유효한 경우에만 ECM을 복호화해 CW를 꺼내 복호화기에 넘겨준다. 즉, 서버에서는 CW->SK->UK 순으로 감싸고(wrapping), 클라이언트에서는 UK->SK->CW 순으로 벗긴다(unwrapping). 이때, CW는 암호화 주기(보통 5~10초) 마다 난수 발생기로 생성하여 암호화 엔진에 공급한다. 그러나, 이 방법은 콘텐츠를 암호화하는데 필요한 CW를 계층적 암호화 체계를 통하여 서버 측에서 클라이언트 측으로 전달되는 형태로, CW를 아무리 자주 변경하더라도 통신망 상에서 암호화에 사용된 키가 전달되므로 그 키의 노출 우려는 여전히 존재하게 된다.

III. 제안한 하드웨어 보안 소자 기반 복제 방지 방법

3-1 제안한 방법의 시스템 구성

제안 시스템은 그림 1과 같으며 콘텐츠 제공 측(서버)과 사용자 단말 양측에 동일한 하드웨어 보안 소자(ALPU-K: Advanced Licensed Protection Unit-Korea)를 두고, 이를 I2C 또는 USB 등 표준 직렬 인터페이스로 각 호스트에 결합한다.

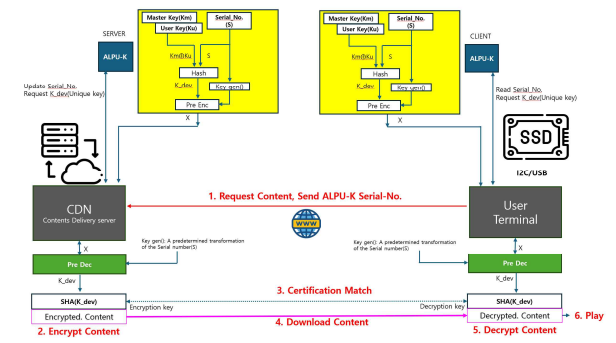


그림 1. 제안하는 하드웨어 보안 소자 기반 콘텐츠 복제 방지 시스템의 개념도

Fig. 1. Concept diagram of hardware security element-based content anti-piracy system

각 소자는 (i) 외부로 반출되지 않는 마스터키(Master Key: Km), (ii) 인증된 명령으로 갱신 가능한 세컨드키(User key: Ku), (iii) 외부 판독이 허용되는 고유 시리얼 S(Serial No.)를 보유한다. 소자는 콘텐츠 자체를 직접 암호화나 복호화하지 않고, 콘텐츠의 암호·복호화에 필요한 디바이스의 고유키(Key of device: K_dev)만을 생성하여 호스트 소프트웨어에 제공하는 키 볼트 겸 신뢰점으로 동작한다. 전체 개념도는 그림 1과 같다.

또한, 같은 종류의 보안 소자라 하더라도 제조사에서 출하될 때부터 시리얼 번호(S)는 보안 소자마다 유일한 고유 값으로 세팅되어 출하되며, 사용자 단말에 접속된 보안 소자의 시리얼 번호(S)는 임의로 사용자가 갱신할 수 없다. 그러나, 서버 장치에 접속된 보안 소자의 시리얼 번호(S)는 갱신이 허용되며, 이는 사용자 단말 장치에 대응하는 디바이스 고유키(K_dev)를 생성하기 위하여 전송받은 사용자 단말장치의 시리얼 번호를 사용하기 위함이다.

위협 모델로는, 공격자가 호스트에서 실행되는 소프트웨어와 그 메모리를 완전히 관찰하거나 조작할 수 있고 직렬 버스 구간의 신호를 도청할 수도 있다고 가정한다. 다만 전용 보안 소자 내부의 비휘발성 메모리와 연산은 정당한 인터페이스를 통해서만 접근 가능하며, 소자에 대한 침습적 물리 공격(디캡 솔레이션 등)은 본 논문의 범위에서 제외한다.

3-2 디바이스 고유키 유도과 정품 결속

각 소자는 자신의 마스터키와 세컨드키 및 고유 시리얼을 입력으로 하는 소자 내부의 일방향 해시함수(예: SHA256)로 외부 추출이 불가능한 K_dev를 유도한다. 그림 1에 나타낸 바와 같이 K_dev는 마스터키와 세컨드키의 배타적 논리합 및 고유 시리얼을 입력으로 하는 해시로 산출되며 수식 (1)로 나타낼 수 있다.

$$K_{dev} = Hash(Km \oplus Ku, S) \quad (1)$$

콘텐츠 구독의 절차는 그림 1과 같이 사용자 단말에 위치한 ALPU-K의 고유 시리얼을 서버에 전송하면 콘텐츠 서버에서는 자신의 ALPU-K 소자내 고유 시리얼을 구독을 요청한 사용자 단말의 고유 시리얼로 갱신 등록한 뒤, ALPU-K내에서 식(1)을 통하여 K_dev를 생성하고, 이를 서버의 호스트 소프트웨어에서 다시 한번 더 해시(SHA(K_dev))하여 암호화 키 EK(Encrypt Key)를 유도하여 그 EK로 구독 신청된 콘텐츠를 암호화한다. 암호화된 콘텐츠를 단말로 다운을 받아 복호화하기 위하여는 단말에 부착된 ALPU-K에서도 서버에서의 ALPU-K와 동일한 절차로 K_dev를 생성하고 단말 호스트 소프트웨어에서 K_dev를 다시 한번 더 해시(SHA(K_dev))하여 복호화 키 DK(Decrypt Key)를 유도한다.

서버와 단말의 소자가 동일한 마스터키 군을 공유하고 동일한 고유 시리얼 번호와 세컨드키를 입력으로 사용하므로, 정품 단말에서 유도된 DK는 서버에서 유도된 EK와 일치한다. 따라서 별도의 명시적 키 전송이나 네트워크상의 키 비교 없이, 단말이 자신의 소자로 복호에 성공한다는 사실 자체가 정품 결속을 암묵적으로 입증한다.

즉, EK는 서버장치에 접속된 보안 소자로부터 생성된 디바이스 고유키(K_dev)를 가지고 유도한 암호화키이고, DK는 사용자 단말 장치에 접속된 보안 소자로부터 생성된 디바이스 고유키(K_dev)를 가지고 유도한 복호화키로 서버장치와 사용자 단말장치 간에 키의 전송 없이 EK는 서버 장치에서 콘텐츠를 암호화하기 위하여 사용하고, DK는 사용자 단말장치에서 암호화된 콘텐츠를 복호화하기 위한 키로 사용한다. 구독 콘텐츠를 서버 장치에서 EK로 암호화하여 사용자 단말 장치에 보내면 사용자 단말장치에서는 DK가 암호화된 해당 콘텐츠를 복호화하기 위하여 사용되기 때문에, 장치 인증을 포함하여 구독 콘텐츠가 하드웨어 보안소자로 부터 유도된 특정 키와 결속하고 있음을 의미한다. 이 EK와 DK의 상호 정합을 본 논문에서는 정품 인증 정합(Certification Match)이라 한다.

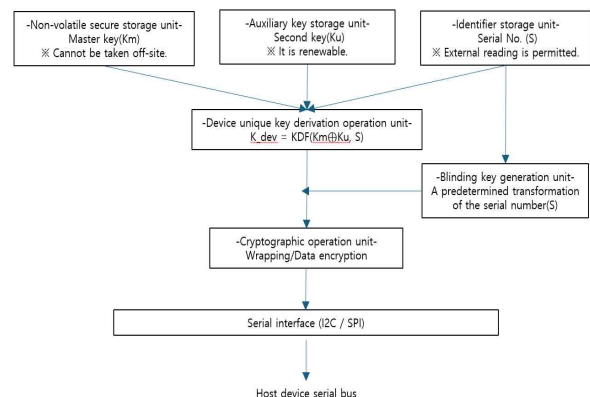


그림 2. ALPU-K로부터 디바이스 고유키(K_dev) 생성
Fig. 2. Device-unique key generation in the ALPU-K

제안한 방식은 마스터키가 소자 밖으로 나오지 않으므로, 고유 시리얼(Serial-No.)이 공개되더라도 마스터키 없이는 K_{dev} 와 EK 및 DK를 재현할 수 없어, 고유 시리얼을 장치 식별 및 구독 식별의 공개 주소로 활용할 수 있다. 보안 소자 내부 구성과 키 유도 흐름은 그림 2와 같다.

3-3 블라인딩에 의한 버스 노출 차단

직렬 버스 구간의 도청으로부터 키를 보호하기 위하여, 보안 소자 ALPU-K는 그림 1과 그림 2에서 살펴보는 바와 같이 디바이스 고유키(K_{dev})를 블라인딩하기 위하여 보안 소자 내에서 $Key_gen()$ 함수를 통하여 블라인딩 키(BK)를 생성하며, 생성된 블라인딩 키(BK)로 K_{dev} 를 AES256(K_{dev} , BK)로 암호화(Pre Enc)하여 블라인딩한 결과 X(암호화된 고유키)만을 버스로 출력한다. $Key_gen()$ 함수는 호스트 생성 난수(NH: Nonce of Host)와 보안 소자 생성 난수(ND: Nonce of Device)를 세션 단위로 주고받으며, 이 난수들의 결합과 시리얼 번호(S)를 가지고 블라인딩 키(BK)를 생성한다. 하나의 세션 안에서 보안 소자가 디바이스 고유키(K_{dev})를 블라인딩 하기 전에 블라인딩 키(BK)를 생성하는데 필요한 NH와 ND를 호스트와 보안 소자간 상호 전달하여 BK를 생성할 준비를 한다.

호스트 소프트웨어는 버스로 수신한 결과 X에 대하여 호스트의 $Key_gen()$ 함수로 보안 소자 내에서 BK를 생성했던 방법과 동일하게 디블라인딩 키(BK)를 생성하여 중간값 X를 디바이스 고유키(K_{dev})로 복원한 뒤(Pre Dec)에 이를 다시 해시(SHA(K_{dev})))하여 최종 암호화 키(EK)를 얻는다.

그러므로 암호화 키의 신뢰점인 마스터키가 외부로 절대 노출되지 않고, 장치에 결속된 보안 소자별로 암호화키가 될 디바이스 고유키(K_{dev})가 블라인딩되어 전달된다. 이때, 블라인딩 키인 BK는 외부에 노출되지 않으며 보안소자와 호스트에서 자체적으로 하나의 세션 단위로 다르게 생성된다. 여기서 BK의 인자가 되는 난수인 NH와 ND가 하나의 세션 단위로 다르게 생성되어 전달되기 때문에 도청 등에 대한 난독의 수위를 높일 수 있게 된다.

제안한 방법은 원천 비밀인 마스터키와 유도 결과인 K_{dev} 가 평문 형태로 버스에 실리지 않는다는 점이 소프트웨어 방식과의 근본적인 핵심적 차이이다.

3-4 동작 절차

전체 동작은 그림 1에 표시된 흐름으로 이루어지며, ALPU-K 기반 콘텐츠의 암호화 및 배포 동작 절차는 그림 3과 같다. 동작의 흐름을 따라가 보면 ① 단말이 콘텐츠의 구독을 요청하며 단말 자신의 보안 소자인 ALPU-K의 고유 시리얼을 읽어서 서버에 전송한다. ② 서버는 요청한 콘텐츠의 구독료 지불과 같은 수행 과정이 끝나면 서버 자신의 ALPU-K 소자 내 고유 시리얼 번호를 수신한 단말의 고유

시리얼 번호로 갱신한다. ③ 서버의 ALPU-K 소자 내에서 마스터 키와 세컨드 키 및 시리얼 번호를 조합하고 해시를 취하여 디바이스 고유키(K_{dev})를 유도한다. ④ 서버 호스트는 ALPU-K로부터 유도된 디바이스 고유키(K_{dev})에 SHA 함수를 취하여 암호화 키 EK를 유도한다. ⑤ EK로 구독한 콘텐츠를 암호화한 후 요청 단말에 배포한다. ⑥ 구독 요청한 단말은 배포된 콘텐츠를 복호화하기 위한 절차로 단말의 ALPU-K 칩으로부터 마스터키와 세컨드 키 및 시리얼 번호를 조합하고 해시를 적용하여 단말의 디바이스 고유키(K_{dev})를 유도한다. ⑦ 단말의 호스트는 단말의 ALPU-K로부터 생성한 디바이스 고유키(K_{dev})에 SHA 함수를 취하여 복호화 키 DK를 유도한다. ⑧ 단말의 호스트는 자신의 보안 소자에서 유도한 복호화 키 DK로 배포된 콘텐츠를 복호화면서 콘텐츠를 구동한다.

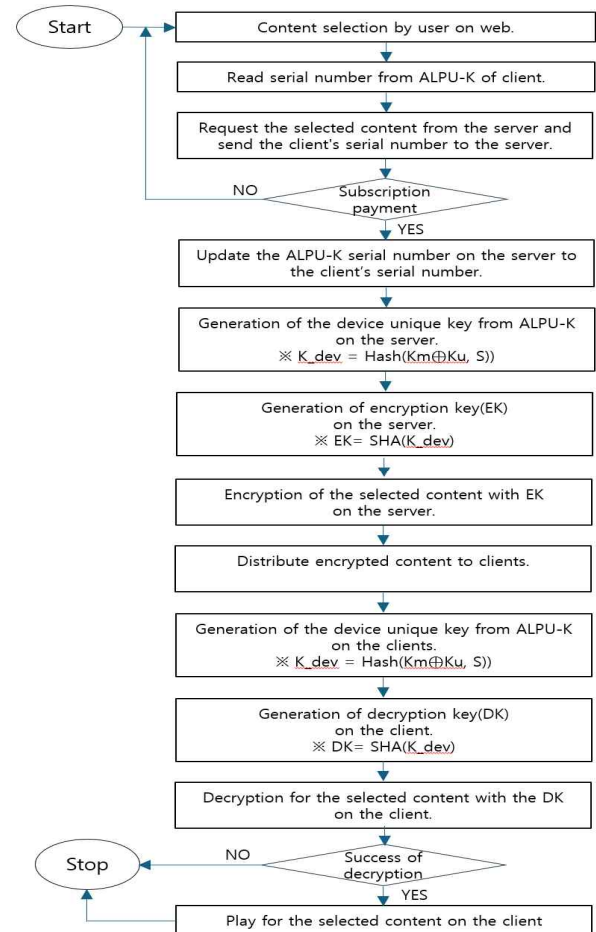


그림 3. 제안한 콘텐츠 복제 방지 시스템 동작 흐름도
Fig. 3. Operational flowchart of the proposed content copy protection system

단말의 ALPU-K로부터 유도된 DK가 서버에서 콘텐츠 암호화에 사용된 EK와 동일해야만 암호화된 콘텐츠가 정상적으로 복호화된다. 만약, 복호화 과정에서 오류 발생으로 구동

이 안 된다는 것은 해당 콘텐츠를 인증된 사용자 단말에 배포된 콘텐츠가 아닌 복제된 콘텐츠로 판단할 수 있다. 그러므로 ALPU-K에서 유도된 단말의 고유키(K_dev)는 사용자의 식별로 활용이 가능하고, 콘텐츠 암호화와 복호화 키의 핵심이 된다. 이 과정에서 암호화에 사용할 디바이스 고유키 발급이 ALPU-K 소자 내부에서 완결되므로 사용자 단말의 콘텐츠 재생 시점에서 외부 라이선스 서버로의 키를 요청할 절차가 필요 없게 된다.

3-5 보안 소자 구현 및 실험

보안 소자 ALPU-K는 그림 1의 상단부에 표시된 요소들이 모두 포함되도록 USB 형태로 설계하고, 보안 소자 내부에 위치하는 키들(마스터, 세컨드, 시리얼번호 등)로부터 디바이스 고유키(K_dev)를 유도하고 블라인딩 하는 과정에 필요한 기능들을 프로토타입으로 구현한다. 보안 소자 ALPU-K 안에서 필요한 기능이 정상적으로 수행되는지의 여부를 테스트하기 위하여 서버 장치에 접속된 ALPU-K 상에서 그림 2에서 수행하는 단위별 기능을 수행하고 정상적 수행에 소요되는 시간을 측정한다. 표 1은 ALPU-K 상에서 디바이스 고유키(K_dev)를 유도하는데, 필요한 각 기능과 기능별 결과 도출에 요구되는 시간을 나타내고 있다. 보안 소자 ALPU-K 내에서 디바이스 고유키(K_dev)를 유도하고 블라인딩하여 그림 1에서 X 값을 인터페이스를 통하여 출력하는데 소요되는 시간이 204.5ms로 약 0.2초 정도로 나타났다.

표 1. 보안 소자 내부 기능 및 테스트 결과

Table 1. Internal functions and test results of the secure element

Function	Name	Time(ms)
Read serial No.	getSerialNo()	58.23
Write serial No.	putSerialNo()	149.11
Write second key	putSecond()	163.59
Derivation of K_dev	genK_dev()	75.93
Generation of blinding key	genBK()	76.39
Blinding K_dev	bldK_dev()	204.59

3-6 가변키 도메인 관리

세컨드키(Ku)는 인증된 API 명령으로 갱신할 수 있다. K_dev가 마스터키와 세컨드키 및 고유 시리얼의 함수이므로, 세컨드키를 갱신하면 K_dev가 변경되어 갱신 이전 키에 기초해 발급된 기존 콘텐츠 결속은 더 이상 유효하지 않게 된다.

이 같은 세컨드 키 갱신은 신규 콘텐츠 사업자의 등록이나 변경 혹은 특정 사용자 단말에 결속된 보안 소자의 도난(예: USB의 경우) 등 불법 사용 차단 등의 특별한 조치를 위하여 마련한 관리자 기능으로 정상적 사용자 단말에 결속된 보안 소자의 경우는 적용되지 않는다. 만약, 특정 사용자 단말에 결

속된 보안 소자 도난 등의 불법 사용 차단 등의 문제로 세컨드 키 갱신이 발생한다면 해당 사용자 단말을 통하여 기존에 정상적으로 구독한 콘텐츠에 대한 재구독 절차(기존 구독 콘텐츠의 재암호화 및 재배포)를 자동으로 수행한다.

따라서 세컨드키 갱신은 기존 키 도메인의 폐기와 신규 도메인으로의 전환을 동시에 달성하며, 키 노출 시 소자 자체를 폐기하지 않고도 신속한 무효화가 가능하다. 나아가 세컨드키를 사업자 세컨드키와 사용자 세컨드키의 계층으로 구성하면, 단일 소자 위에서 복수 사업자와 사용자가 상호 격리된 키 도메인을 형성하는 멀티테넌트 격리의 구현이 가능하다.

IV. 보안성 분석 및 비교

제안한 방법에 대한 보안성 비교 분석은 II 장 2-1에서 살펴본 화이트 박스 암호 기반 방식과 소프트웨어 전용 DRM, 그리고 2-2에서 언급한 TEE(예: Widevine L1)와 TPM 방식과의 비교를 대상으로 진행한다. 비교는 (i) 신뢰점 추출에 대한 저항성, (ii) 공격면과 노출 지점, (iii) 배포성과 운영 특성의 세 관점에서 수행하며, 각 관점에서 방식별 취약성의 구조적 원인과 잔존 문제점을 함께 도출한다. 분석의 기준 위협 모델은 2-1과 동일하게 공격자가 호스트에서 실행되는 소프트웨어와 그 메모리를 완전히 관찰하고 조작할 수 있는 화이트 박스 수준의 소프트웨어 가시성을 가진다고 가정한다. 또한, 유선방송의 콘텐츠 보호 방법과 제안한 방법에서의 콘텐츠 보호 방법에 대한 비교는 권한 획득 등의 서비스 환경과 시스템 구성 형식이 상이하여 항목별 비교는 배제하고, 방법 간 핵심적인 기능 중심의 차이점에 대하여만 간단히 살펴본다.

4-1 신뢰점 추출에 대한 저항성

앞에서 언급한 기존 방식인 화이트 박스 암호 기반 방식과 소프트웨어 전용 DRM 방식 및 TEE(예: Widevine L1)와 TPM 방식이 본 논문에서 제안한 보안 소자(Security Element) 기반 ALPU-K 방식과의 본질적 차이는 신뢰점, 즉 전체 보안 체계가 의존하는 최상위 비밀키가 어느 계층에 존재하며, 이를 추출하는데 요구되는 공격 비용과 추출 성공시의 파급 범위가 어떠한가에 있다. 각각의 비교 대상별로 신뢰점의 위치, 대표적 키 추출 경로, 추출 결과의 파급 등을 다룬다.

1) 화이트 박스 암호(White-Box Cryptography) 방식

화이트 박스 암호 방식의 신뢰점은 실행 파일에 포함된 록-업 테이블 안에 수학적으로 내장된다. 테이블은 배포된 모든 사본에 동일하게 존재하는 정적 자산이므로, 공격자는 프로그램을 실행할 필요조차 없이 바이너리 확보만으로 오프라

인 분석을 시작할 수 있다. Billet 등의 BGE 공격[3]은 Chow 방식 화이트 박스 AES의 내장 키를 약 230의 작업량, 즉, 범용 PC로 단시간에 수행 가능한 수준으로 복원할 수 있음을 보였고, 후속 연구들은 이 작업량을 더욱 낮추었다. 나아가 Bos 등의 차분 계산 분석(DCA)[4]은 구현 내부 구조에 대한 지식 없이 실행 추적의 통계 분석만으로 키를 자동 복원하므로, 설계를 비공개로 유지하는 방어까지 무력화한다. 특히 심각한 문제는 추출의 과급 범위이다. 동일 테이블이 전 사본에 복제되는 구조상 한 사본에 대한 1회의 분석 성공이 배포본 전체의 보호를 동시에 무효화하며(한 번 파훼, 전체 무력화), 설령 키 복원에 실패하더라도 테이블과 주변 코드를 통째로 적출하여 복호 오라클로 재사용하는 코드 리프팅(code lifting)이 가능하므로 키 은닉만으로는 보호가 완결되지 않는다.

2) 소프트웨어 전용 DRM 방식

소프트웨어 전용 DRM의 신뢰점은 단말에 프로비저닝된 장치 키(키박스)와 이를 통하여 발급받는 콘텐츠 키로서, 난독화된 소프트웨어 저장소와 실행 중 프로세스 메모리에 존재한다. 난독화는 분석에 드는 시간을 지연시킬 뿐 추출 가능성 자체를 제거하지 못하며, 복호 시점에는 키가 반드시 메모리에 평문으로 전개되어야 하므로 메모리 덤프, 디버거 부착, EME API 가로채기에 의한 추출이 반복적으로 시연될 수 있다[5], [6]. 이와 같은 키박스가 메모리에서 그대로 덤프되는 안전하지 않은 저장 문제는 CVE-2021-0639로 보고되었다[7]. 과급 측면의 문제는 더 크다. 하나의 장치 키가 추출되면 공격자는 해당 단말을 소프트웨어로 모사하여 라이선스 서버로부터 임의 콘텐츠의 키를 지속적으로 발급받을 수 있으므로[7], 단일 키 추출이 특정 콘텐츠의 유출에 그치지 않고 대량의 키 수확 수단으로 확장된다.

3) TEE 기반 방식

TEE 기반 방식의 신뢰점은 SoC 제조 시 퓨징되는 장치 루트키와 TEE 내부에서 처리되는 파생 키로서, REE의 순수 소프트웨어 공격만으로는 원칙적으로 추출되지 않는다. 그러나 TEE의 격리는 호스트와 동일한 프로세서 위에서 동작하는 신뢰 OS와 신뢰 애플리케이션(TA: Trust Application)이라는 또 하나의 소프트웨어 계층으로 구현되므로, 신뢰점의 안전성이 결국 이 소프트웨어 스택의 구현 품질에 종속된다는 문제를 가진다. 실제로 상용 TrustZone 기반 TEE에서 신뢰 OS와 TA의 구현 취약점을 통한 권한 상승과 키 유출 사례가 체계적으로 보고되었으며[12], 전력 관리 인터페이스를 악용한 결합 주입 공격은 소프트웨어 조작만으로 TrustZone 내부의 비밀키를 추출할 수 있음을 실증하였다[13]. 요컨대 TEE는 신뢰점을 소프트웨어 공격면으로부터 크게 이격시키지만, 방대한 TCB와 REE를 공유하는 실리콘이라는 구조적 요인으로 인하여 추출 불가능성을 보장하지는 못한다.

4) TPM 기반 방식

TPM의 신뢰점은 모듈 내부의 비휘발성 메모리(NVM)에 저장되는 계층적 키 구조의 최상위 키들, 대표적으로 제조 시에 각인되는 승인 키(Endorsement Key: EK)와 소비자 설정 시 생성되는 저장 루트키(Storage Root Key: SRK)이다. 이 키들은 TPM 칩 경계 밖으로 평문 상태로 반출되지 않도록 설계되어, 호스트 CPU에서 실행되는 운영체제나 응용이 아무리 완전히 장악되더라도 순수 소프트웨어 공격만으로는 루트키 자체를 직접 읽어낼 수 없으며, 하위 키들은 SRK로 암호화(wrapping)되어 호스트 저장소에 보관되고, 사용 시점에만 TPM 내부에서 복호되어 연산에 사용된다. 이 구조가 TPM이 화이트박스 암호나 소프트웨어 전용 DRM보다 신뢰점 추출 저항성에서 근본적으로 우위에 있는 이유이다.

TPM 방식은 PCR 측정, 봉인(sealing), 원격 증명(attestation), 키 계층 관리에 이르는 방대한 명령 집합을 제공하지만, 그 안전성은 이를 구현한 펌웨어 스택에 종속되며, TPM이 별도 칩(dTPM)으로 구현될 경우에는 호스트와 LPC 또는 SPI 같은 직렬 버스로 연결이 되어야 하는데, 이 버스가 암호화되지 않은 채 운용되면 봉인이 해제되어 전달되는 키 재료나 인증값이 버스 신호 스니핑으로 포착될 수 있다. 또한, 펌웨어 TPM(fTPM)은 SoC의 TEE 안에서 구현되어 별도 버스가 없다는 장점이 있으나, 그 대신 저항성이 호스트 SoC의 TEE 격리 강도에 종속되게 되는 한계가 존재한다.

5) 제한한 보안 소자(ALPU-K) 기반 방식

제한 방법인 보안 소자 기반 방식의 신뢰점인 마스터키는 호스트와 물리적으로 분리된 전용 소자의 비휘발성 메모리에만 존재하고, 소자의 명령 집합에는 마스터키를 외부로 반출하는 인터페이스 자체가 정의되어 있지 않다. 디바이스 고유 키 유도는 식 (1)에 따라 소자 내부에서 완결되고 일방향 해시를 경유하므로, 공격자가 호스트의 소프트웨어와 메모리 및 버스 로그를 모두 확보하더라도 얻는 것은 특정 시리얼에 대한 유도 결과에 한정되며 이로부터 마스터키를 역산할 수 없다. 또한, 소자는 키 유도와 세컨드키 갱신이라는 최소 기능만 수행하므로, 범용 신뢰 OS와 다수의 TA를 수용하는 TEE에 비하여 TCB가 현저히 작아 구현 취약점이 개입할 여지도 구조적으로 축소된다. 결국, 동일한 화이트 박스 수준의 소프트웨어 가시성 하에서, 화이트 박스 암호와 소프트웨어 전용 DRM은 신뢰점이 공격면 안에 있어 추출이 실제로 시연되고, TEE나 TPM은 신뢰점이 공격면 밖에 있어서 이러한 소프트웨어 기반 방식의 문제점은 없으나 TEE의 경우는 대형 소프트웨어 스택과 공유 실리콘을 경유한 우회 추출이 보고되었고, TPM은 구현한 보안의 안정성이 펌웨어에 종속되거나 별도 칩으로 구현될 경우 버스의 암호에 대한 문제가 여전히 존재한다. 그러나, 제한한 ALPU-K 기반 방법은 신뢰점이 공격면 밖에 존재하기 때문에 소프트웨어 기반 복제 방지 방식에서의 문제점이 없고 하드웨어 신뢰점 기반 방식인 TEE와

TPM의 펌웨어 스택 종속 같은 문제점을 해결할 수 있다.

다만 제안한 방법에도 신뢰점 계층의 잔존 문제가 있을 수 있다. 서버와 단말의 소자가 동일한 마스터키 군을 공유하는 구조이므로, 위협 모델에서 제외한 침습적 물리 공격(디캡슐레이션, 마이크로프로빙 등)으로 단일 소자의 마스터키가 노출될 경우 그 파급이 해당 소자를 넘어 동일 키 군 전체에 미칠 수 있다. 따라서 제조 로트나 사업자 단위로 마스터키 군을 분할하고 세컨드키 도메인으로 파급 범위를 격리하는 키 군 관리가 전제되어야 하며, 제조 단계의 키 주입과 관리 공정이 새로운 신뢰 경계로 편입된다는 점이 고려될 필요가 있다.

4-2 공격면과 노출 지점 비교

공격면 비교는 키 재료와 콘텐츠 평문이 시스템의 어느 지점에, 어느 시간 동안, 어떤 형태로 존재하는가를 기준으로 한다. 노출이 상시적일수록, 그리고 노출된 재료의 재사용 가치가 클수록 실효 공격면은 넓어진다.

1) 화이트 박스 암호(White-Box Cryptography) 방식

화이트 박스 방식에서 키를 내장한 록-업 테이블은 실행 파일과 프로세스 메모리에 상시 상주하는 정적 노출 지점이다. 공격자는 공격 시점을 선택할 필요 없이 언제든 바이너리를 확보하여 오프라인에서 분석할 수 있고, 실행 추적을 수집하면 DCA[4]와 같은 자동화 공격이 적용된다. 여기에 테이블과 주변 코드를 통째로 적출하는 코드 리프팅으로 키 복원 없이 복호 기능 자체가 탈취될 수 있으며, 이를 완화하기 위한 외부 인코딩 역시 인코딩과 디코딩 루틴이 동일 호스트에 존재하므로 근본적 차단이 되지 못한다. 노출이 상시적이고 정적이면서 전체 사본에 공통 적으로 적용된다는 점에서 비교 대상 중 공격면이 가장 넓다.

2) 소프트웨어 전용 DRM 방식

소프트웨어 전용 DRM의 노출은 실행 시점에 집중된다. 라이선스 수신 직후부터 복호 종료까지 콘텐츠 키가, 복호 이후에는 콘텐츠 평문이 프로세스 메모리에 전개되며, 이 구간은 디버거 부착과 메모리 스캔 및 함수 후킹 등으로 관찰된다[5, 6]. 나아가 네트워크 구간의 라이선스 교환 트래픽과 브라우저의 EME 인터페이스가 프로토콜 계층의 추가 노출 지점을 형성하며, 공개된 개념 증명들은 바로 이 계층에서 키 소재를 포착하였다[5]. 즉, 공격면이 메모리 계층과 프로토콜 계층의 두 층위에 걸쳐 있고, 난독화와 무결성 검사 등 자체 방어 수단이 모두 공격자가 장악한 동일 호스트 위에서 동작한다는 점에서 자기방어의 한계가 존재한다고 볼 수 있다.

3) TEE 기반 방식

TEE 기반 방식은 키 재료가 REE 메모리에 나타나지 않으며, 보안 미디어 경로를 갖춘 구현에서는 복호된 콘텐츠까지

보호 버퍼 내에서 처리되어 일반 영역에 노출되지 않으므로, 소프트웨어 기반 방식과 비교하여 노출 지점이 상대적으로 좁다. 그러나, 공격면이 소멸하는 것이 아닌 이동하는 것으로 보아야 한다. REE와 TEE를 오가는 전환 인터페이스와 신뢰 OS 및 TA가 노출하는 호출 표면이 새로운 공격 면이 되고 [12], REE와 캐시 및 파이프라인을 공유하는 마이크로아키텍처 부채널과 전압 및 클럭 조작에 의한 결함 주입[13]으로 격리 경계를 우회한 사례가 보고되었다. 즉, TEE의 공격 면은 좁지만 깊은 형태이며, 그 실효성과 안전성은 SoC 제조사별 구현 품질과 패치 상태에 크게 의존한다고 볼 수 있다.

4) TPM 기반 방식

TPM 기반 방식의 공격면은 TEE와 유사하게 키 재료가 일반 호스트 메모리에 상시 노출되지는 않으나, 공격면이 소멸하는 것이 아니라 이동한다는 점에 유의하여야 한다. 호스트와 TPM을 연결하는 직렬 버스(예: LPC/SPI)가 암호화되지 않은 채로 운용되는 구성에서는 버스 도청을 통하여 전달되는 키 재료가 포착될 수 있으며, TPM 펌웨어와 명령 인터페이스 자체도 새로운 경계 공격면이 형성될 수 있다.

5) 제안한 보안 소자(ALPU-K) 기반 방식

제안 방법에서 원천 비밀인 마스터키와 키 유도 로직은 물리적으로 분리된 소자 내부에 있고, 호스트에 드러나는 공격면은 표준 직렬 버스 위의 소수 명령과 응답 인터페이스로 한정된다. 버스 구간에는 3.3절에서 제시하는 바와 같이 디바이스 고유키(K_dev)를 블라인딩하여 출력하므로, 어떠한 경우에도 마스터키가 평문 형태로 버스에 실리지 않으므로 버스 관측만으로 신뢰점을 복원할 수 없다. 소자의 명령 집합이 키 유도와 세컨드키 갱신 등 소수 기능으로 제한되어, 경계 인터페이스의 복잡도 역시 TEE나 TPM의 월드 전환 표면에 비하여 현저히 작다.

다만, 제안한 방법의 잔존 노출 지점도 명확히 제시할 필요가 있다. 잔존 노출 지점을 살펴보면 첫째, 최종 콘텐츠 복호를 호스트 소프트웨어가 수행하므로, 호스트가 산출한 복호 키 DK와 복호된 콘텐츠 평문은 재생 시점의 호스트 메모리에 일시적으로 존재한다. 그러나 이 노출의 파급은 소프트웨어 방식과 질적으로 다르다. DK는 특정 소자의 고유 시리얼과 현재 세컨드키 도메인에 결속된 파생 키이므로, 이를 탈취하더라도 공격자가 얻는 것은 해당 단말에서 해당 도메인의 콘텐츠를 복호하는 능력, 즉 호스트를 이미 장악한 공격자가 재생 화면 포착 등으로 달성할 수 있는 수준을 넘지 않는다. 다른 단말이나 다른 도메인의 키 유도, 신뢰점 접근에는 어떠한 이득도 제공하지 않으며, 세컨드키 갱신으로 해당 도메인을 폐기하면 탈취된 키의 시간적 유효성마저 차단된다. 이는 테이블 하나의 분석이 전 사본을 무력화하는 화이트 박스 방식이나 장치 키 하나의 추출이 대량 라이선스 수확으로 확장되는 소프트웨어 전용 DRM과 대비되는, 노출 파급 반경(blast radius)의 구조적 축소이다. 둘째, 장악된 호스트의 악성 코

드는 정당한 소프트웨어와 동일한 자격으로 소자에 키 유도를 요청할 수 있으므로, 소자가 해당 단말에 결속된 범위 안에서 복호 올라클로 오용될 수 있다. 이 역시 결속 범위를 넘는 복제와 재배포로 확장되지는 않으나, 명령 인증과 요청 빈도 제한 같은 소자 측 접근 통제에 보장할 수 있는 영역이다.

마지막으로, 재생 출력 단계의 평문 노출은 정도의 차이는 있으나 위에서 살펴본 네 방식에 공통된 한계이다. 따라서 출력 단계의 보호는 어느 방식에서든 화면 출력 보호(예: HDCP)와 같은 직교적 수단으로 다루어야 하며, 제안 방법의 기여는 그 이전 단계인 키와 신뢰점 계층과 장치 결속 계층에서 공격면을 근본적으로 축소하는 데 있다.

4-3 배포성과 운영 특성

보안 강도와 별개로, 복제 방지 방식의 실용성은 이미 보급된 장치에 대한 적용 용이성(배포성)과 키 수명주기 관리, 오프라인 동작, 권리 정책 통제와 같은 운영 특성에 의하여 좌우된다. 이러한 관점에서 제안한 방법은 강한 보호를 제공하는 TEE 혹은 TPM 기반 방식과도 뚜렷한 차이를 볼 수 있다.

1) 화이트 박스 암호(White-Box Cryptography) 방식

화이트 박스 암호 방식은 순수 소프트웨어이므로 초기 배포성이 가장 높고 키가 코드에 내장되어 오프라인 동작도 가능하다. 그러나 운영 단계의 문제가 크다. 키가 록-업 테이블에 고정되므로 키 갱신·폐기가 곧 소프트웨어 재빌드와 전체 단말 재배포를 의미하고, 이미 유출된 구버전 사본을 회수할 수단이 없어 파훼 이후의 복구가 사실상 불가능하다. 단말별 차등 키를 부여하려면 단말 수만큼의 개별 빌드가 필요하여 확장성이 낮고, 테이블 저장 공간과 테이블 조회 중심 연산으로 인한 성능 저하도 수반된다.

2) 소프트웨어 전용 DRM 방식

소프트웨어 전용 DRM은 클라이언트 모듈의 설치만으로 적용되어 배포성이 높고, 라이선스 서버를 통한 중앙집중 운영이 가능하다. 재생 시마다 서버 정책을 강제할 수 있으므로 대역 기간, 동시 시청 수 제한과 같은 세밀한 권리 통제와 계정 단위의 즉시 차단은 비교 대상 중 가장 유연하고 볼 수 있다. 그러나 이 장점은 운영상 제약이 될 수 있는데, 재생이 라이선스 서버의 가용성에 상시 종속되어 서버 장애나 통신 두절 시 정당한 사용자도 재생할 수 없고, 망 분리와 차단 환경에서는 도입 자체가 불가능하다. 모든 재생 요청이 서버로 집중되므로 시청 이력이 축적되는 프라이버시 문제와 라이선스 인프라의 구축과 운영 비용이 발생하고, 라이선스 서버 침해가 전체 가입자 키 발급 체계의 침해로 직결되는 문제가 될 수 있다.

3) TEE 기반 방식

TEE 기반 방식의 근본적 제약은 배포성에 있다. TEE는

SoC 설계 단계에서 통합되어야 하므로 이미 보급된 장치에 사후 적용할 수 없고, 신규 적용 역시 SoC 제조사와 단말 제조사의 협력 및 TA 서명과 탑재 절차를 전제로 하므로 자체 SoC 설계 권한이 없는 콘텐츠 사업자가 독자적으로 도입할 수 없다. 운영 측면에서도 제조사마다 상이한 TEE 구현이 혼재하는 파편화로 인하여 TA를 플랫폼별로 이식과 검증하여야 하고, 신뢰 OS와 TA의 보안 패치가 단말 제조사의 펌웨어 갱신주기에 종속되어 지원이 종료된 단말에는 알려진 취약점이 잔존한다[12]. 나아가 실제 서비스는 TEE 미탑재와 미인증 단말을 배제하기 어려워 소프트웨어 전용 등급으로의 하향 운용을 병행하게 되는데, 이 경우 체계 전체의 실효 보안 수준이 가장 취약한 등급으로 수렴하는 운영상의 역설이 발생할 수 있다.

4) TPM 기반 방식

TPM 기반 방식의 배포성 제약도 TEE와 본질적으로 동일하다. TPM은 플랫폼 제조 단계에서 메인보드에 통합되어야 하므로 엔드포인트 하드웨어에 대한 통제권이 없는 콘텐츠 사업자가 이미 보급된 장치에 사후적으로 도입하기 어렵다.

TPM 기반 방식은 키 보호의 근거를 소프트웨어의 난독화가 아니라 하드웨어의 물리적 격리에 두므로, 배포성과 운영 특성에서 TEE와 본질적으로 동일하고 소프트웨어 기반 방식과는 상반된 성질을 보인다. 배포성 측면에서는 단말에 전용 칩 또는 펌웨어 TPM이 사전에 탑재되어 있어야 하므로 초기 도입 장벽이 높고, 미탑재 단말은 소프트웨어 갱신만으로 대응할 수 없어 하드웨어 교체를 수반하며, 도입 가능 여부가 사실상 단말제조 시점에 확정되어 이미 현장에 배포된 레거시 단말군에는 소급 적용이 어렵다. 반면 운영 단계에서는 뚜렷한 이점을 갖는다. 키가 칩 내부에서 생성되고 평문 형태로 외부에 반출되지 않아 키가 코드에 종속되지 않으며, 그 결과 키 갱신과 폐기를 소프트웨어 재빌드나 전체 단말 재배포 없이 원격 명령만으로 수행할 수 있어 파훼 이후의 복구가 가능하다.

5) 제한한 보안 소자(ALPU-K) 기반 방식

제한 방법은 I2C 혹은 USB 등 표준 직렬 인터페이스로 결합되는 부가 소자이므로, SoC 교체나 제조사 협력 없이 기존 서버와 단말에 사후에도 장착할 수 있어 TEE 대비 배포 장벽이 현저히 낮다. 키 발급이 소자 내부에서 완결되므로 재생 시점의 서버 접속이 불필요하여 망 분리나 차단 환경과 통신 두절 상황에서도 동작하고, 재생 이력이 서버로 전송되지 않아 프라이버시 부담도 적다. 키 수명주기 관리 역시 소프트웨어 재배포 없이 세컨드키 갱신이라는 단일 인증 명령으로 도메인 전환과 폐기가 완결되며, 사업자와 사용자 계층의 세컨드키 구성으로 단일 소자 위의 멀티테넌트 격리가 가능하다.

제한한 방법에서의 운영상 고려사항이라면 다음과 같은 경우를 들 수 있을 것이다. 서버와 단말 양측에 물리 소자를 공급하여야 하므로 소자 단가와 장착 및 유통 물류라는 순수 소

소프트웨어 방식에는 없는 비용이 발생할 수 있다. 따라서 불특정 다수의 범용 단말을 대상으로 하는 대중 시장보다는 셋톱박스, 키오스크, 산업이나 교육용 단말과 같이 하드웨어 공급 경로가 통제되는 환경에서 우위가 있다고 할 것이다. 둘째, 결속의 단위가 호스트가 아니라 소자이므로, 탈착형(USB) 소자의 분실이나 도난 시 시청 권한이 소자와 함께 이동한다. 이에 대한 대응은 분실 신고 후 세컨드키 갱신으로 해당 도메인을 폐기하는 운영 절차로 뒷받침되어야 한다. 셋째, 오프라인 완결성은 사후 통제력과 상충한다. 재생 시점에 서버 검증이 개입하지 않으므로 이미 배포된 암호문과 결속 단말에 대해서는 대여 기간 만료와 같은 시간 기반 권리 제한이나 원격 즉시 차단을 강제하기 어렵고, 폐기의 효과는 신규 도메인으로 암호화되는 이후 콘텐츠부터 미친다. 즉, 오프라인 상태에서는 바로 세컨드 키 갱신에 따른 폐기의 효과를 볼 수 없으며, 어떤 형태(온라인 혹은 수동 갱신 등)로든 보안 소자 내 세컨드 키의 갱신이 진행된 후에 기존 콘텐츠의 폐기효과를 볼 수 있다. 그리고, 필요에 따라서는 정상 구독자의 기존 콘텐츠에 대한 재암호화 및 재배포를 자동으로 진행된다. 또한, 시간 제한형 사업 모델에는 세컨드키 도메인의 주기적 순환이나 온라인 검증의 선택적 병용과 같은 보안 설계가 필요하다. 넷째, 구독 등록 시 서버 측 소자가 가입자별 유도 연산을 수행하므로 대규모 동시 구독 처리에서는 소자 처리량이 병목이 될 수 있어, 다중 소자 병렬화 등 서버 측 확장 구조에 대한 정량 평가는 향후의 과제로 연구되어야 할 필요성이 있다.

종합하면 제안한 방법은 신뢰점 보호에서는 TEE나 TPM에 준하는 하드웨어 격리를 제공하면서 배포성에서는 소프트웨어 방식에 근접하는 사후 장악성을 확보하여, 콘텐츠 복제방지의 기존 방식들이 보안성과 배포성 사이에서 강요받던 양자택일을 완화한다. 반면 물리 소자의 공급 비용과 오프라인 구조에 따른 사후 통제력의 제약은 제안한 방법에서의 고유 운영 부담이 될 수 있으며, 적용 환경의 특성에 따라 상가보완 설계와 함께 검토되어야 할 부분이다.

4-4 종합 비교

4-1과 4-2에서 살펴본 내용을 종합하면 표 2와 같다. 제안한 방법은 신뢰점 위치와 키 추출 저항성에서 소프트웨어 방식을 능가하고, 배포성에서 TEE나 TPM 기반 방식을 능가한다.

4-5 유선방송 콘텐츠 보호 방법과의 비교

유선방송의 콘텐츠 보호 방법은 콘텐츠를 암호화하는데 필요한 CW(Control Word)를 계층적 암호화 체계(UK: Unique Key, SK: Service Key, CW: Control Word)를 통하여 서버 측에서 클라이언트 측으로 전달되는 형태로, CW를 아무리 자주 변경하면서 전달하더라도 통신망 상에서의 키 노출 우려는 여전히 존재한다고 볼 수 있다. 그러나, 제안

표 2. 콘텐츠 복제 방지 방식별 보안 특성 비교

Table 2. Comparison of security characteristics of content copy protection methods

Security Property	White-box cryptography-based	Software-only DRM	TEE based	TPM based	Proposed method (ALPU-K)
Root of Trust (Root Key) Location	Inside software (lookup table)	Process memory	On-chip TEE	TPM internal NVM	Dedicated secure element NVM
Software Extraction of Root Key	Possible (through mathematical analysis)	Possible (via memory dump)	Principally infeasible (bypass attacks reported [12],[13])	Principally infeasible (bus or firmware bypass attacks)	Impossible (key cannot be exported)
Representative Known Attacks	BGE attack ($\approx 2^{30}$ complexity)	Memory and EME interception	Trusted OS/TA vulnerabilities, fault injection [12],[13]	Bus sniffing, firmware vulnerabilities	None reported
Key Exposure on Bus / Memory	Constant exposure (lookup tables remain resident)	Plaintext key resides in memory during decryption	No exposure	Bus eavesdropping possible (if unprotected)	No exposure due to blinding
Offline Operation	Partially supported	Not supported (server-dependent)	Partially supported	Supported	Supported
Continuous License Server Dependency	Low	High	Medium	Low	None
Key Update / Revocation	Requires software redistribution	Server policy-based	Limited	Supported through key hierarchy	Immediate via secondary key update
Multi-tenant Isolation	Difficult	Managed on the server side	Limited	Supported through key hierarchy	Supported through key hierarchy
Applicability to Existing Devices	High	High	Low (requires new SoC)	Low (platform-integrated)	High (add-on secure element)

한 방법은 서버장치에서 암호화키를 생성하고 생성한 암호화키를 통신망 등 어떤 수단으로도 클라이언트에 전달하지 않으며, 암호화된 콘텐츠를 복호화하기 위하여 클라이언트 장치 자체에서 복호화키를 생성한다는 것이다.

또한, 제안한 방법과 유선방송이 채택하는 방식인 CAS와의 환경적 차이는 CAS는 누구에게나 콘텐츠가 노출되며 구독 권한에 따라 "볼 수 있느냐"를 시청 이전에 통제하는 반면, 제안한 방법은 인증된 사용자에게만 콘텐츠가 노출되고 "시청 이후에도 암호화된 콘텐츠의 복사나 이동 및 재생"이 가능하지만 반드시 인증된 사용자 단말에서만 복호 및 재생이 가능하다는 것이다.

V. 결 론

본 논문은 마스터키를 전용 하드웨어 보안 소자의 비휘발성 메모리에 두고 소자 내부에서만 디바이스 고유키를 유도하여 콘텐츠를 정품 장치에 결속시키는 복제 방지 방법을 제안하고, 대표적 소프트웨어 기반 방식인 화이트 박스 암호화 방식과 소프트웨어 기반 DRM 방식 및 하드웨어 신뢰점 기반인 TEE와 TPM 방식과 비교 분석하였다. 화이트 박스 암호 기반 방식은 BGE 공격 등으로 키가 수학적으로 추출되며, 소프트웨어 전용 DRM은 메모리 덤프와 API 가로채기로 키가 평문 노출된다. 하드웨어 신뢰점 기반인 TEE나 TPM은 신뢰점의 위치가 칩내에 있어서 원칙적으로 소프트웨어적인 루트 키 추출이 불가능하고 버스나 메모리의 키 노출 방식이 가능하나 보안 기능이 SoC 설계 단계에서 통합되어야 하므로 이미 보급된 장치에는 사후에 적용할 수 없고, 신규 적용 역시 SoC 제조사와 단말 제조사의 협력 및 TA 서명과 탑재 절차를 전제로 하므로 자체 SoC 설계 권한이 없는 콘텐츠 사업자가 독자적으로 도입할 수 없다.

그러나 제안 방법은 원천 비밀을 소프트웨어 공격면 밖의 실리콘에 위치시키고 난수 블라인딩으로 버스 노출을 차단함으로써, 동일한 소프트웨어 가시성 하에서도 키 추출 저항성을 근본적으로 높일 수 있다. 또한, 오프라인 동작, 세컨드키 기반의 즉시 키 폐기와 멀티테넌트 격리, 부가 소자를 통한 기존 장치 적용성을 제공하여, 하드웨어 수준의 신뢰점 보호와 소프트웨어 수준의 배포성을 동시에 달성할 수 있는 장점이 있다.

다만, 모든 비교 대상 방식이 공통으로 가지는 재생 출력 단계의 평문 노출은 화면 출력 보호 등 직교적 수단으로 보완되어야 하며, 소자에 대한 침습적 물리 공격에 대한 정량적 평가는 향후 과제로 남는다. 향후 연구에서는 제안 방법의 프로토타입 구현을 통한 처리 지연 및 대역폭 측정과, 부채널 저항성에 대한 실증적 평가를 수행할 계획이다.

참고문헌

- [1] S. Chow, P. Eisen, H. Johnson, and P. C. Van Oorschot, "White-Box Cryptography and an AES Implementation," in *Proceedings of the 9th Annual International Workshop on Selected Areas in Cryptography (SAC 2002)*, Newfoundland, Canada, pp. 250-270, 2003.
- [2] S. Chow, P. Eisen, H. Johnson, and P. C. van Oorschot, "A White-Box DES Implementation for DRM Applications," in *Proceedings of the ACM CCS-9 Workshop on Digital Rights Management (DRM 2002)*, Washington, DC, pp. 1-15, 2003.
- [3] O. Billet, H. Gilbert, and C. Ech-Chatbi, "Cryptanalysis of a White Box AES Implementation," in *Proceedings of the 11th International Workshop on Selected Areas in Cryptography (SAC 2004)*, Waterloo, Canada, pp. 227-240, 2005.
- [4] J. W. Bos, C. Hubain, W. Michiels, and P. Teuwen, "Differential Computation Analysis: Hiding Your White-Box Designs Is Not Enough," in *Proceedings of the 18th International Conference on Cryptographic Hardware and Embedded Systems (CHES 2016)*, Santa Barbara: CA, pp. 215-236, 2016.
- [5] Widevine L3 Decryptor: A Proof-of-Concept Browser Extension Demonstrating Content-Key Extraction from Software-Only DRM via EME Interception (Public Repository), 2019.
- [6] Widevine DRM Security Levels (L1/L2/L3): Architectural Overview of TEE-Based versus Software-Only Implementations, Google Widevine Technical Documentation.
- [7] G. Patat, M. Sabt, and P.-A. Fouque, "Exploring Widevine for Fun and Profit," arXiv:2204.09298, 2022. (Reports Keybox Recovery from Memory; Related Flaw Assigned CVE-2021-0639.) <https://doi.org/10.48550/arXiv.2204.09298>
- [8] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted Execution Environment: What It Is, and What It Is Not," in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, pp. 57-64, August 2015. <https://doi.org/10.1109/TrustCom.2015.357>
- [9] G. Patat, M. Sabt, and P.-A. Fouque, "Exploring Widevine for Fun and Profit," in *Proceedings of the IEEE Security and Privacy Workshops (SPW)*, San Francisco: CA, pp. 277-288, May 2022. <https://doi.org/10.1109/SPW54247.2022.9833867>
- [10] J. Winter and K. Dietrich, "A Hijacker's Guide to Communication Interfaces of the Trusted Platform Module," *Computers & Mathematics with Applications*, Vol. 65, No. 5, pp. 748-761, March 2013. <https://doi.org/10.1016/j.camwa.2012.06.018>

- [11] D. Moghimi, B. Sunar, T. Eisenbarth, and N. Heninger, "TPM-FAIL: TPM Meets Timing and Lattice Attacks," in *Proceedings of the 29th USENIX Conference on Security Symposium*, Boston: MA, pp. 2057-2073, August 2020.
- [12] D. Cerdeira, N. Santos, P. Fonseca, and S. Pinto, "SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems," in *Proceedings of the IEEE Symposium on Security and Privacy*, San Francisco: CA, pp. 1416-1432, 2020.
- [13] A. Tang, S. Sethumadhavan, and S. Stolfo, "CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1057-1074, 2017.



이효승(Hyo Seung Lee)

1992년 : 건국대학교 전자공학과 졸업
2019년 : 성균관대 소프트웨어학과
공학석사

1992년~2000년: 삼성전자 비디오 연구실 연구원
2000년~2002년: 하이트론 ECT 반도체 개발 팀장
2002년~현 재: ㈜네오와인 대표이사
※관심분야 : AI 반도체, 정보보안, 디지털저작권(DRM) 등



홍진우(Jin Woo Hong)

2022년 : 가천대학교 컴퓨터공학과
졸업
2023년 : 가천대학교 컴퓨터공학과
대학원 공학석사

2023년~현 재: ㈜네오와인 연구원
※관심분야 : AI 모델개발, 컴퓨터비전, 디지털저작권(DRM) 등



정지우(Ji Woo Jeong)

2024년 : 대림대학교 사물인터넷공학과
졸업

2024년~현 재: ㈜네오와인 연구원
※관심분야 : 정보보호, SW개발, IOT, 디지털저작권(DRM) 등



이준호(Jun Ho Lee)

2026년 : University of Southern
California 재학

2026년~현 재: ㈜네오와인 연구원
※관심분야 : 정보보호, 알고리즘, 디지털저작권(DRM) 등



한기태(Gi-Tae Han)

1982년 : 충남대학교 계산통계학과
졸업
1990년 : 한양대학교 대학원
(공학석사)
2001년 : 한양대학교 대학원
(공학박사-컴퓨터비전)

1992년~2007년: 경원대학교 전자계산학과 교수
2010년~2011년: University of Texas at Austin, Research
Prof.

2007년~2025년: 가천대학교 컴퓨터공학과 교수
2025년~현 재: 가천대학교 컴퓨터공학과 명예교수
2025년~현 재: ㈜네오와인 연구소 전무이사
※관심분야 : 컴퓨터비전, AI 모델, 정보보안,
디지털저작권(DRM) 등