

영지식 비밀 공유 기법을 이용한 프라이버시 보존형 음원 정산 무결성 검증 시스템

최창준*, 장성일*, 조인희*, 신동명*†

A Privacy-Preserving Integrity Verification System for Music Royalty Settlements Using a Zero-Knowledge Secret Sharing Scheme

Chang-Jun Choi*, Sung-Il Jang*, In-Hee Jo*, Dong-Myung Shin*†

요 약

최근 스트리밍 중심의 음원 유통이 보편화되면서 합의된 지분율에 따라 다수의 권리자가 음원 수익을 배분받는 구조가 일반화되었으나, 플랫폼의 폐쇄적인 데이터 처리로 인해 권리자가 정산의 정확성을 검증하기 어려운 실정이다. 기존 블록체인 기록 방식은 정산 데이터의 투명성 확보가 가능하나, 상세 이용 로그와 지분율 등 계약 조건이 노출되어 프라이버시 침해 및 영업 비밀이 유출되는 한계점이 존재한다. 이에 따라, 본 논문에서는 영지식 증명과 비밀 공유 기법을 이용하여 원본 데이터를 공개하지 않고도 정산 결과의 무결성을 보장할 수 있는 프라이버시 보존형 검증 시스템을 제안한다. 제안 시스템은 계약 조건을 (t, n) 비밀 공유로 다수의 독립 노드에 분산 보관하고 임계치 승인 시에만 검증 연산을 수행하며, 대용량 원천 로그를 머클 트리로 압축·가명화한 상태에서 정산 규칙 적용의 정합성을 단일 증명으로 입증함으로써 기밀성·결탁 저항성·검증 가능성을 동시에 제공한다. Hyperledger Fabric 기반 프로토타입 실험 결과, 원천 이용 로그 100만 건 규모에서도 온체인 검증 약 0.87 ms, 전송량 1 KB 미만으로 로그 규모와 무관하게 일정한 성능을 확인하였다.

Abstract

As streaming-centric music distribution grows, revenues are shared among multiple rights holders based on agreed equity ratios. However, opaque platform operations make verifying settlement accuracy difficult for these rights holders. While blockchain solutions can offer transparency, they risk exposing detailed usage logs and sensitive contract terms, leading to privacy and trade secret leaks. To resolve this, this paper proposes a privacy-preserving verification system utilizing zero-knowledge proofs (ZKP) and secret sharing schemes (SSS). By distributing contract terms across independent nodes via (t, n) secret sharing, verification operations are executed only upon threshold approval. Large-scale source logs are compressed and pseudonymized using a Merkle tree, allowing the system to prove the consistency of settlement rules through a single proof without revealing raw data. This approach successfully ensures confidentiality, collusion resistance, and verifiability. A Hyperledger Fabric-based prototype demonstrated constant performance regardless of data size, achieving approximately 0.87 ms on-chain verification time and under 1 KB transmission size even for 1 million logs.

한글키워드 : 영지식 증명, 비밀 공유 기법, 음원 정산, 블록체인, 프라이버시 보존 검증

keywords : Zero-Knowledge Proof (ZKP), Secret Sharing Scheme (SSS), Music Royalty Settlement, Blockchain, Privacy-Preserving Verification

* 엘에스웨어(주) 소프트웨어연구소 연구개발본부

접수일자: 2026.08.24. 심사완료: 2026.09.02.

† 교신저자: 신동명(email: roland@lsware.co.kr)

게재확정: 2026.09.20.

1. 서론

스트리밍 중심으로 재편된 디지털 음원 시장은 작사·작곡가, 실연자, 음반제작자, 유통사, 신탁관리단체 등 다수의 이해관계자가 하나의 음원 수익을 사전 합의된 지분율에 따라 배분받는 다중 권리자 구조를 기본으로 한다[1]. 음원 소비마다 발생하는 저작권료의 경우, 플랫폼이 수집한 이용 로그(재생 횟수, 상품 유형, 지역·기간별 단가 등)에 정산 규칙을 적용하여 산출되지만, 원천 데이터와 배분 로직이 플랫폼 내부에 폐쇄적으로 관리되므로 데이터 보유 측(플랫폼)과 검증 요구 측(권리자) 사이에 심각한 정보 비대칭이 발생한다. 권리자는 지급액이 규칙대로 산정되었는지 확인할 수단이 사실상 없으며, 이러한 구조는 정산 투명성에 대한 분쟁의 원인이 되어왔다[2].

대안으로 분산원장에 정산 데이터를 투명하게 기록·공개하는 방식이 논의되어 왔으나, 상세 이용 로그·매출 정보는 영업 비밀이고 지분율·정산 단가 등 계약 조건은 비밀 유지 의무의 대상이자 민감 정보기 때문에, 이를 공유 원장에 그대로 기록하면 영업 비밀 유출과 프라이버시 침해가 발생한다. 즉 '검증을 위한 투명성'과 '보호되어야 할 비밀성'이 충돌하는 구조적 딜레마가 존재한다. 검증을 단일 기관이 전담하는 신뢰 제3자(Trusted Third Party, TTP) 방식 역시 데이터 단독 열람·수정에 따른 위·변조 위험과 신뢰의 단일 주체 종속이라는 한계를 갖는다[3].

본 논문에서는 이러한 딜레마를 해소하기 위하여 영지식 증명(Zero-Knowledge Proof, ZKP)과 임계치 기반 비밀 공유 기법(Secret Sharing Scheme, SSS)을 결합한 영지식 비밀 공유(이하 zk-SSS) 기반의 프라이버시 보존형 음원 정산 무결성 검증 시스템을 제안한다. 권리자별 지분율 등 민감 계약 조건은 Shamir (t, n) 비밀 공유로 n 개 조각으로 분할되어 신탁관리단체·감사

기관·권리자 대표 등 상호 독립적인 검증 노드에 분산 보관되며, 임계치(Threshold) t 개 이상의 승인이 모인 경우에만 영지식 연산 구조의 비공개 입력으로 활성화된다. 대용량 원천 로그는 머클 트리로 압축되어 단일 루트로 약정됨으로써 개별 레코드가 식별 불가능한 가명화 형태로만 유통되고, 증명자는 은닉된 로그와 활성화된 조건에 정산 규칙이 정확히 적용되었음을 입증하는 증명을 생성하며, 허가형 블록체인의 다수 피어가 원본을 열람하지 않고 결과값·증명·머클 루트만으로 정산 결과를 교차 검증한다.

2. 관련 연구

2.1 음원 저작권료 정산 시스템의 현황

국내 음원 저작권료는 관련 정수규정에 따라 산정되어 다수의 이해관계자(저작권자, 실연자, 유통사 등)에게 분배되며, 이들 사이의 지분율과 정산 단가는 비공개 계약에 의해 결정된다[4]. 문제는 정산 근거인 이용 로그가 플랫폼 사업자에 의해 독점 수집되고, 정산 연산 과정이 외부에서 재현 및 검증 불가능한 형태로 운영된다는 점이다. 이로 인한 취약점은 세 가지로 요약된다.

첫째, 연산 과정의 블랙박스화로 인해 권리자는 통보된 정산 명세를 수용할 뿐, 오적용이나 누락 등의 오류를 탐지할 독립적 수단이 없다. 둘째, 검증을 위해 원천 로그를 공개할 경우 영업 비밀과 타인의 계약 정보가 침해되므로 투명성 요구와 비밀 보호 의무가 충돌한다. 셋째, 정산 권한이 단일 주체에 집중되어 해당 기관의 내부 오류나 시스템 장애가 곧바로 전체 정산 신뢰의 붕괴로 이어진다. 최근 블록체인 기반 플랫폼이 제안되었으나, 대부분 결과의 기록과 지급 자동화에 집중할 뿐 원천 데이터와 계약 조건을 은닉한 상태에서 연산 과정 자체를 암호학적으로 검증하는 문제는 해결하지 못하고 있다[5].

2.2 영지식 증명(ZKP)

영지식 증명은 증명자(Prover)가 원본 데이터나 비밀 정보 자체의 노출 없이 특정 명제가 참임을 검증자(Verifier)에게 입증하는 암호 프로토콜이다[6]. 영지식 증명이 수학적으로 안전하게 성립하기 위해서는 세 가지 필수 속성이 충족되어야 한다. 첫째, 참인 명제에 대한 정확한 증명자가 생성한 증명은 검증자에게 수용되어야 하는 완전성(Completeness)이다. 둘째, 거짓 증명에 대해 악의적인 증명자가 검증자를 속일 가능성이 무시할 수 있을 정도로 작아야 하는 건전성(Soundness)이다. 셋째, 검증자는 명제의 참·거짓 여부 외에는 원본 데이터나 비밀 정보에 관한 어떠한 추가 정보도 획득할 수 없어야 하는 영지식성(Zero-knowledgeness)이다. 이러한 속성을 만족하는 비대화형 기법 중 zk-SNARK는 증명 크기가 상수 수준으로 매우 작고 연산 복잡도와 무관하게 빠른 검증을 지원하여, 블록체인 검증 환경에 가장 최적화된 기술로 평가받는다[7]. 허가형 블록체인인 하이퍼레저 패브릭(Hyperledger Fabric)에서도 영지식 증명을 체인코드에 결합하여 온체인 검증 비용을 상수로 유지하는 기술의 실용성이 입증되었다[8].

2.3 비밀 공유 기법(SSS)

비밀 공유 기법은 단일 장애점과 중앙집중형 저장에 따른 데이터 유출 위험을 줄이기 위해, 비밀 정보 S 를 여러 조각으로 분할하여 다수의 참여자에게 분산 보관하는 암호학적 기법이다[9]. 이 기법의 핵심은 사전에 설정된 임계치 이상의 조각이 모인 경우에만 원본 비밀 정보를 복원할 수 있고, 임계치 미만의 조각으로는 비밀 정보에 관한 어떠한 유의미한 정보도 얻을 수 없도록 하는 임계치 구조에 있다. 가장 대표적인 Shamir의 (t, n) 비밀 공유 기법은 비밀을 총 n 개의 조각으로 분할하고, 그중 최소 t 개의 조각이 모였을

때만 복원을 허용한다[10]. 이는 비밀 S 를 상수항으로 갖는 차수 $t-1$ 의 다항식의 평가값을 비밀 조각으로 배포하고, 라그랑주 보간법을 통해 t 개의 조각을 복원하는 수학적 원리에 기반한다.

3. 제안 기법

본 장에서는 제안 시스템의 전체 아키텍처와 위협 모델을 정의하고, 계약 조건의 임계치 분산 및 활성화로부터 영지식 증명 생성과 온체인 교차 검증에 이르는 프로토콜을 기술한다. 본 제안 시스템의 주요 시스템 파라미터는 표 1과 같다.

표 1. 주요 기호 및 시스템 파라미터
Table 1. Notation and System Parameters

기 호	설 명
S	- 외부에 공개되지 않는 민감 계약 조건 (e.g., 권리자별 지분율 등)
(t, n)	- 전체 n 개 비밀 조각 중 최소 t 개가 모일 때, 계약 조건을 활성화할 수 있는 임계치 설정
$s_i = f(x_i)$	- 공개 좌표 x_i 에서 계산되어 검증 노드 N_i 에 배포되는 i 번째 비밀 조각
N_i	- 비밀 조각 s_i 를 보관하고, 승인 서명을 제출하는 i 번째 분산 검증 노드
id	- 정산 회차별 검증 요청을 구분하고, 승인 서명 및 검증 결과를 동일 요청에 마인딩하기 위한 고유 식별자
A	- 유효한 승인 서명과 비밀 조각을 제출한 검증 노드의 집합
σ_i	- 검증 노드 N_i 가 생성한 승인 서명
L_{raw}	- 음원 플랫폼이 수집한 전처리 이전의 원천 이용 로그
L_{norm}	- 표준 스키마로 정규화된 이용 로그
$h_i = H(l_k)$	- 충돌 저항 해시 함수 H 를 이용하여 계산한 정규화 로그 레코드 l_k 의 해시값
rt	- 정규화 로그의 해시값으로 구성된 머클 트리 MT 의 루트이자 로그 집합의 공개 약정값
F	- 로그와 계약 조건에 적용되는 사전 합의된 결정적 정산 규칙
y^{plat}	- 음원 플랫폼이 제출한 검증 대상 정산 결과값
y^{calc}	- 정규화된 이용 로그와 재구성된 계약 조건에 정산 규칙 F 를 적용하여 영지식 연산 구조 내부에서 재계산한 정산 결과값
π	- 로그 무결성, 임계치 충족 및 정산 연산의 정합성을 입증하는 영지식 증명
pk	- 영지식 증명 π 를 생성할 때 사용하는 증명키
vk	- 온체인에서 영지식 증명의 유효성을 판정할 때 사용하는 검증키

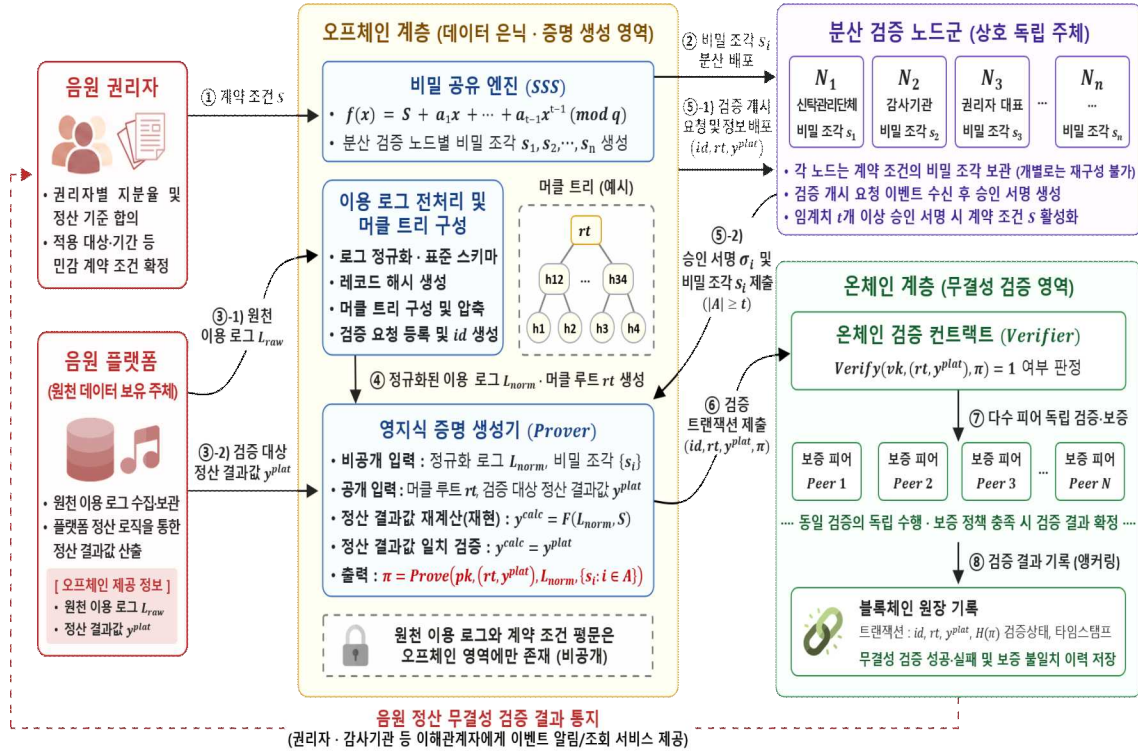


그림 1. 제안 시스템의 전체 아키텍처 및 동작 흐름
Fig 1. Overall Architecture and Operational Flow of the Proposed System

3.1 시스템 아키텍처

본 논문의 제안 시스템은 민감한 원천 데이터와 계약 조건을 외부에 공개하지 않은 상태에서 음원 정산 결과의 무결성을 검증할 수 있도록, 오프체인 데이터 은닉 계층과 온체인 분산 검증 계층을 결합한 하이브리드 구조로 설계하였다. 그림 1과 같이 오프체인 계층에서는 계약 조건의 비밀 공유, 원천 이용 로그의 정규화와 머클 트리 기반 약정[11], 영지식 증명 생성이 수행된다. 반면, 온체인 계층에서는 허가형 블록체인의 다수 보증 피어가 검증 요청 식별자 id , 머클 루트 rt , 검증 대상 정산 결과값 y^{plat} , 영지식 증명 π 만을 이용하여 정산 결과의 정당성을 독립적으로 검증하고, 보증 정책을 충족한 검증 결과를 분산 원장에 기록한 후 감사·추적할 수 있도록 한다.

제안 시스템은 비밀 공유·데이터 전처리·영지식 증명 생성을 수행하는 오프체인 계층, 이해관계자별 상호 독립 주체인 분산 검증 노드군, 무결성 검증 및 감사 추적을 수행하는 온체인 계층으로 구성된다. 음원 관리자별 지분율과 같은 민감 계약 조건은 비밀 조각 형태로 분할되어 상호 독립적인 검증 노드에 분산 보관되며, 사전에 설정된 임계치 이상의 승인이 확보된 경우에만 영지식 증명 연산의 비공개 입력으로 활성화된다. 이에 따라, 원천 이용 로그와 계약 조건의 평문은 오프체인 은닉 영역을 벗어나지 않고, 검증에 필요한 최소 정보만 온체인 계층으로 전달된다. 이러한 계층 분리 구조는 데이터 기밀성과 결탁 저항성을 유지하면서도 정산 결과의 무결성, 독립적 검증 가능성 및 감사 추적성을 확보하도록 한다.

3.2 위협 모델 및 보안 요구사항

제안 시스템의 안전성을 분석하기 위해 공격자를 악의적 플랫폼, 결탁 검증 노드, 준정직 (Semi-honest) 검증자, 외부 공격자의 네 유형으로 구분한다. 먼저 악의적 플랫폼은 정산액을 축소하거나 부정확한 정산 결과를 생성하기 위해 원천 이용 로그를 누락·변조하거나, 사전에 합의되지 않은 정산 규칙을 적용할 수 있다. 결탁 검증 노드는 최대 $t-1$ 개의 노드가 공모하여 분산 보관된 계약 조건 S 를 복원하거나, 정당한 승인 절차를 우회하여 검증을 임의로 개시 또는 차단하려 할 수 있다. 준정직 검증자는 정해진 프로토콜을 정상적으로 수행하지만, 검증 과정에서 관찰되는 공개 입력과 증명 정보로부터 원천 로그 또는 권리자별 지분율과 같은 민감 정보를 추론하려 할 수 있다. 외부 공격자는 통신 구간의 도청, 비밀 조각의 탈취, 승인 서명이나 영지식 증명의 재사용 등을 통해 시스템의 기밀성과 무결성을 훼손하려 시도할 수 있다.

표 2 .위협 모델에 따른 보안 요구사항
Table 2. Security Requirements Based on Threat Model

구분	요구사항	설명
R1	기밀성	- 로그·계약 조건 평문이 온체인·외부에 노출되지 않고 검증 과정에서 추가 정보가 유출되지 않아야 함
R2	연산 무결성	- y^{plat} 가 머클 루트 rt 에 약정된 로그와 활성 조건 S 에 규칙 F 를 정확히 적용한 결과임이 보장되어야 함
R3	결탁 저항성	- $t-1$ 개 이하 노드 결탁으로 S 복원·검증 활성화가 불가능해야 함
R4	검증 가능성	- 원본 열람 없이 (id, y^{plat}, π, rt) 만으로 독립적인 검증이 가능해야 함
R5	재사용 방지	- 승인 서명·증명이 다른 정산 회차에 재사용될 수 없어야 함
R6	감사 추적성	- 검증 성공·실패가 위·변조 불가능한 형태로 기록되어야 함

본 연구에서는 공격자가 확률적 다항시간 범위의 계산 능력을 가지며, 사용되는 해시 함수의 충돌 저항성과 이산대수 문제의 계산적 난해성이

유지된다고 가정한다. 또한, 공격자가 결탁할 수 있는 검증 노드 수는 임계치 미만인 최대 $t-1$ 개로 제한한다. 이러한 위협 모델을 바탕으로 제안 시스템이 충족해야 할 핵심 보안 요구사항을 표 2와 같이 정의하였다.

3.3 영지식 비밀 공유 기반 가명화 프로토콜

제안 프로토콜은 계약 조건의 분산 구조화, 원천 로그의 가명화 및 증명 생성, 임계치 기반 계약 조건 활성화, 영지식 연산 구조의 결합 검증으로 구성된다. 각 단계는 계약 조건과 원천 이용 로그의 평문을 외부에 노출하지 않으면서도 정산에 사용된 데이터와 연산 과정의 정당성을 검증할 수 있도록 설계된다. 특히, 비밀 공유를 통한 접근 권한 분산, 머클 약정을 통한 로그 무결성 보장, 영지식 증명을 통한 연산 정합성 검증을 하나의 프로토콜로 결합한다.

3.3.1 계약 조건의 SSS 분산 구조화

음원 발매 시점에 권리자 간 합의로 계약 조건 S 가 확정되면, 비밀 공유 엔진이 이를 Shamir (t, n) 기법으로 분할한다. 이때 엔진은 유한체 $GF(q)$ 위에서 S 를 상수항으로 갖는 차수 $t-1$ 의 다항식 $f(x)$ 를 다음과 같이 생성한다.

$$f(x) = S + a_1x + a_2x^2 + \dots + a_{t-1}x^{t-1} \pmod{q} \quad (1)$$

이후 서로 다른 공개 좌표 x_i 에서 다항식을 평가하여 총 n 개의 비밀 조각 s_i 를 생성한다.

$$s_i = f(x_i), \quad i = 1, \dots, n \quad (2)$$

생성된 각 조각은 신탁관리단체, 감사기관, 권리자 대표 등 상호 독립적인 검증 노드에 하나씩 배포되며, $t-1$ 개 이하의 조각으로는 S 에 관한

유의미한 정보를 얻을 수 없다. 검증 요청은 오프체인 계층의 이용 로그 전처리 단계에서 등록되며, 이때 해당 회차의 정산 검증을 요청하는 고유 식별자 id 가 생성된다. 등록된 검증 요청 (id, rt, y^{plat}) 이 검증 노드군에 전달되면, 각 검증 노드 N_i 는 자신의 서명키 sk_i 로 해당 값들에 바인딩된 승인 서명 σ_i 를 생성한다.

$$\sigma_i = \text{Sign}_{sk_i}(id \parallel rt \parallel y^{plat}) \quad (3)$$

유효한 승인을 제출한 노드 집합을 A 라고 할 때, $|A| \geq t$ 를 충족할 때만 계약 조건이 검증 연산에 활성화된다. 각 승인 서명은 id 및 제출 데이터(rt, y^{plat})에 바인딩되므로 다른 검증 요청에 재사용될 수 없으며(R5), 승인 수가 임계치(t)에 미달하면 검증 절차는 개시되지 않는다.

3.3.2 오프체인 전처리 및 증명 생성

오프체인 전처리 계층에서는 음원 플랫폼에서 제공한 원천 이용 로그 L_{raw} 를 표준 스키마에 따라 정규화하고, 각 레코드의 해시값을 리프로 하는 머클 트리 MT 를 구성한다. 레코드 l_k 의 해시값과 전체 로그 집합의 머클 루트 rt 는 다음과 같이 계산된다.

$$h_i = H(l_k), \quad rt = MT(h_1, \dots, h_N) \quad (4)$$

머클 루트 rt 는 전체 로그에 대한 약정값으로 사용되며, 개별 레코드 l_k 를 변경할 경우 기존 머클 루트 rt 와 불일치하므로 로그의 위·변조를 탐지할 수 있다. 또한, 외부에는 원천 레코드 대신 단일 머클 루트만 공개되므로 이용 로그는 개별 데이터가 식별되지 않는 가명화 상태로 유지된다. 다만, 머클 약정은 머클 루트 rt 가 생성된 이후 발생한 로그의 변경·삭제·대체를 탐지할 수 있으나, 약정 이전에 플랫폼이 일부 원천 이용

로그를 누락한 경우까지 직접 검증하지는 못한다. 약정 이전의 로그 완전성을 보장하기 위해서는 로그 발생 시점의 주기적 머클 루트 기록, 외부 데이터 원천과의 대조 또는 서명된 이용 기록과 같은 별도의 보완 수단이 필요하다.

영지식 증명 생성기는 정규화된 로그 L_{norm} 과 활성화된 비밀 조각 $\{s_i\}$ 를 비공개 입력으로 사용하고, 머클 루트 rt 와 검증 대상 정산 결과값 y^{plat} 를 공개 입력으로 설정한다. y^{plat} 은 음원 플랫폼이 제출한 검증 대상 정산 결과값이며, 영지식 증명 생성기는 영지식 연산 구조 내부에서 정산 규칙 F 에 따른 결과 y^{calc} 를 다시 계산하여 $y^{calc} = y^{plat}$ 가 성립함을 증명한다(R2). 음원 권리자 j 의 정산 금액 y_j^{calc} 는 각 음원 레코드의 이용량 $cnt(l_k)$, 적용 단가 $price(l_k)$, 지분율 r_j 를 이용하여 다음과 같이 계산된다.

$$y_j^{calc} = F_j(L_{norm}, S) = \sum_{k=1}^N cnt(l_k) \cdot price(l_k) \cdot r_j \quad (5)$$

영지식 증명 생성기는 사전 합의된 정산 규칙 F 가 로그와 계약 조건에 정확하게 적용되었음을 증명하는 비대화형 영지식 증명 π 를 생성한다.

$$\pi = \text{Prove}(pk, (rt, y^{plat}), (L_{norm}, \{s_i : i \in A\})) \quad (6)$$

온체인 계층에는 검증 요청 식별자 id , 머클 루트 rt , 검증 대상 정산 결과값 y^{plat} , 영지식 증명 π 만 트랜잭션으로 제출되어 무결성 검증에 사용되고, 검증 후에는 id, rt, y^{plat} , 영지식 증명의 해시값 $H(\pi)$ 만 블록체인 원장에 기록된다. 원천 이용 로그 L_{raw} 와 계약 조건 S 의 평문은 오프체인 은닉 영역을 벗어나지 않아 외부에 공개되지 않으며(R1), 활성화된 비밀 조각 $\{s_i\}$ 과 중간 비공개 입력은 검증 종료 이후 영지식 증명 생성기가 안전하게 폐기한다고 가정한다.

3.3.3 계약 조건 활성화 메커니즘

임계치 승인이 충족되면 승인 노드가 제출한 비밀 조각을 영지식 연산 구조의 입력으로 활성화한다. 연산 구조 내부에서는 라그랑주 보간법을 통해 계약 조건 S 를 다음과 같이 재구성한다.

$$S = f(0) = \sum_{i \in A} s_i \cdot L_i(0) \quad (7)$$

재구성된 계약 조건 S 는 연산 구조 내부에서만 사용되며 공개 입력이나 출력으로 노출되지 않는다. 또한, 비밀 조각과 사전 공개된 커밋먼트의 일치성을 함께 검증하도록 설계함으로써 위변조를 방지할 수 있다. 승인 서명은 검증 요청에 바인딩되므로 재사용이 차단되며, $t-1$ 개 이하의 조각이나 승인만으로는 유효한 증명을 생성할 수 없다(R3).

3.3.4 영지식 연산 구조의 결합 제약 조건

제안하는 영지식 연산 구조는 로그 무결성, 임계치 충족 여부, 계약 조건의 올바른 재구성 및 정산 규칙의 적용 결과를 하나의 영지식 증명으로 통합한다. 이를 위해 표 3과 같은 조건을 단일 결합 제약으로 검증한다.

표 3. 영지식 연산 구조의 결합 제약
Table 3. Combined Constraints of the ZK Operation

구분	검증 항목	설명
C1	로그 무결성	- 사용된 레코드의 머클 포함 경로가 공개 머클 루트 rt 와 일치하는지 검증
C2	비밀 조각 유효성	- 사용된 비밀 조각이 사전 등록된 조각 커밋먼트와 일치하는지 검증
C3	임계치 충족	- 활성화된 t 개 이상의 비밀 조각으로 계약 조건을 재구성하고, 승인 서명의 유효성과 임계치 충족 여부를 확인
C4	정산 규칙 정합성	- 정산 규칙 F 가 로그와 재구성된 계약 조건에 정확히 적용되었는지 검증
C5	결과값 일치성	- F 로 재계산한 y^{calc} 가 음원 플랫폼이 제출한 y^{plat} 와 일치하는지 검증
C6	데이터 비공개성	- 원천 로그·계약 조건의 평문이 공개 입력·출력에 포함되지 않았는지 검증

이러한 조건들은 개별적으로 증명되지 않고 하나의 영지식 증명 π 로 압축된다. 따라서 검증자는 단일 증명 검증만으로 원천 이용 로그의 무결성, 임계치 승인 충족, 계약 조건의 적법한 활성화, 정산 규칙의 정확한 적용 및 민감 정보의 비공개성을 동시에 확인할 수 있다. 이는 비밀 공유, 머클 트리 및 영지식 증명을 단순히 병렬적으로 적용하는 방식과 달리, 각 기술의 보안 속성을 하나의 검증 관계로 결합한다는 점에서 제안 프로토콜의 핵심적인 특징이다.

오프체인 계층에서 최종적으로 생성된 검증 트랜잭션은 허가형 블록체인의 검증 컨트랙트로 전달되며, 다수 피어의 독립 검증과 보증 정책 확인을 거쳐 검증 결과가 분산원장에 기록된다. 검증 성공·실패 및 보증 불일치 이력은 감사와 분쟁 해결을 위한 추적 정보로 활용되도록 설계된다(R6). 실제 정산금 지급 및 음원 권리자별 자동 분배는 본 연구 범위에 포함하지 않는다.

3.4 무결성 검증 프로세스

제안 프로토콜의 온체인 계층은 검증 컨트랙트 실행, 다수 피어의 독립 검증·보증, 검증 결과의 분산원장 기록으로 구성된다.

3.4.1 검증 트랜잭션 제출 및 컨트랙트 실행

선별된 머클 루트 rt , 정산 결과값 y^{plat} , 영지식 증명 π 가 검증 요청 식별자 id 와 함께 허가형 블록체인 네트워크에 트랜잭션으로 제출되면, 다수의 보증 피어는 원장에 사전 배포된 영지식 검증 컨트랙트를 각자 독립적으로 실행하여 아래식과 같이 π 가 공개 입력에 대해 유효한지 판정한다. 영지식 증명의 공개 입력은 식 (6)과 동일하게 머클 루트와 정산 결과값으로 구성된다.

$$Verify(vk, (rt, y^{plat}), \pi) = 1 \quad (8)$$

3.4.2 분산 합의 기반 교차 검증 및 앵커링

각 피어는 원천 데이터를 열람하지 않고 증명의 유효성만으로 정당성을 판정하며(R4), 다수 피어의 결과가 보증 정책을 충족할 때 검증 통과로 확정된다. 동일 입력을 복수 노드가 검증하므로 특정 노드가 조작·매수되더라도 정상 노드와 어긋나 즉시 탐지된다. 검증을 통과한 트랜잭션은 요청 식별자 id , 머클 루트 rt , 정산 결과값 y^{plat} , 영지식 증명 해시값 $H(\pi)$ 와 함께 분산원장에 앵커링되며, 검증 실패·합의 불일치 이력도 이벤트로 기록되어 감사 증거로 남도록 설계된다. 앵커링된 검증 결과를 근거로 하는 실제 정산금 지급·권리자별 분배는 별도 파이프라인의 몫으로 본 연구의 범위에 포함하지 않는다.

제안 기법의 교차 검증은 (a) 계약 조건이 임계치 분산되어 복수 주체의 승인 없이는 검증이 개시되지 않는 권한적 교차와, (b) 동일 증명을 여러 피어가 독립적으로 검증·대조하는 수평적 교차의 이중 구조를 이룬다. 이처럼 승인 권한과 검증 신뢰를 동시에 분산함으로써, 어느 한 주체의 오류나 악의가 전체 정산 검증의 무결성을 훼손하지 못한다. 두 축의 결합을 통해 '원천 데이터는 가명화 상태로 은닉하고, 결과 검증에 대한 신뢰는 특정 기관에 집중시키지 않고 다수에게 분산한다'는 제안 기법의 설계 목표가 달성된다.

4. 시스템 구현 및 평가

본 장에서는 3장에서 제안한 영지식 비밀 공유(zk-SSS) 기반 음원 정산 무결성 검증 시스템을 프로토타입으로 구현하고, 실측을 통해 해당 프로토콜이 실제 환경에서 동작함을 확인하여 성능을 평가한다. 제시하는 모든 수치는 예시값이 아닌 실제 측정 결과이며, 재현에 필요한 구성과 스크립트는 프로토타입 저장소에 포함하였다.

4.1 실험 환경 및 데이터셋

실험은 단일 호스트에서 수행하였으며, 상세 구성은 표 4와 같다. 영지식 연산 구조는 이를 기술·컴파일하는 도메인 특화 언어인 circom으로 작성하고, snarkjs의 Groth16(BN254)로 증명을 생성하였다. 온체인 검증은 Hyperledger Fabric v2.5.13 상의 Go 체인코드에서 gnark-crypto로 수행하였다. 블록체인 네트워크는 3장의 설계에 따라 조직 4개, 보증 피어 노드 4개, Raft 오더러 노드 3개로 구성하였다. 벤치마크는 동일 호스트의 도커 컨테이너에서 실행되므로 물리 네트워크 지연은 포함되지 않는다.

표 4. 실험 환경
Table 4. Experimental Environment

구분	사양
CPU / 메모리	- AMD Ryzen 9 5950X (16코어/32스레드) - 67 GB
OS / 런타임	- Linux 6.18 (WSL2) - Node.js v20.18.1
증명 도구	- circom 2.2.3 - snarkjs 0.7.4 (Groth16/BN254)
블록체인 / 검증	- Hyperledger Fabric v2.5.13 - gnark-crypto v0.20.1

성능 평가에는 실제 음원 소비의 롱테일 특성을 모사한 합성 데이터셋을 사용하였다. 각 레코드는 트랙 식별자, 재생 수(cnt), 곡당 단가(price), 지역(region), 상품 유형(product)으로 구성된다. 소수의 인기 음원이 전체 재생의 대부분을 차지하는 실제 소비 분포를 반영하기 위해 재생 수는 bounded Pareto 분포(형상 지수 $\alpha=1.5$)로 생성하였고, 단가는 상품 유형별 고정 티어를 적용하였으며, 지역·상품은 균등 분포로 배정하였다. 모든 난수는 시드 고정 의사난수 생성기로 산출하여 동일 시드에 대해 항상 같은 데이터셋이 재현되도록 하였으며, 유한체 오버플로를 방

지하기 위해 각 필드 값의 범위를 제한하였다.

다만 합성 데이터셋은 실제 플랫폼 로그의 스키마 다양성, 결제 취소·중복 재생 등 이상 레코드, 실제 시장 규모 분포를 반영하지 못하므로 정규화 규칙의 실무 적합성은 본 실험으로 검증되지 않는다. 그러나 온체인 검증 비용은 공개 입력 수에만 의존하여 데이터 값과 무관하고, 머클 구성·증명 생성 비용은 레코드 값이 아닌 레코드 수와 배치 수에 의해 결정되므로, 값의 분포가 달라져도 규모별 성능 추세는 유지된다. 실제 이용 로그 확보 시 동일한 재현 스크립트로 재평가할 수 있도록 데이터 입력을 표준 스키마로 분리하였다.

4.2 시스템 구현

프로토타입은 계약 조건의 Shamir (t, n) 분할, 이용 로그의 머클 약정, 단일 Groth16 연산 구조 내부의 계약 조건 재구성·정산 규칙 적용·결과값 일치 증명, 그리고 Hyperledger Fabric 다수 피어의 온체인 검증과 결과 앵커링을 구현한다. 반면, 승인 서명 기반 임계치 확인(C3의 승인 부분)과 비밀 조각 커밋먼트 일치 검증(C2)은 3장의 설계 수준에 머물러 있으며, 4.3절의 성능 수치에는 해당 비용이 포함되지 않는다.

음원 권리자의 계약 조건 S 는 소수점 오차 없이 다루기 위해 고정소수점 정수($1e4$ 스케일)로 표현하고, 이를 Shamir (t, n) 비밀 공유로 분할하여 서로 다른 검증 노드에 분산 보관한다. 원천 이용 로그의 각 레코드는 유한체 위에서 직접 동작하도록 설계된 영지식 친화적 해시 함수인 Poseidon[12]으로 $leaf = Poseidon(cnt, price)$ 와 같이 커밋하며, 이 리프들을 Poseidon 머클 트리로 압축하여 외부에는 루트 rt 만 공개한다.

정산의 정합성은 단일 Groth16 연산 구조(Settlement)로 입증하며, 이 구조는 다음 세 가지를 하나의 증명으로 결합한다. ① 사용된 각

레코드의 머클 포함 경로가 공개 루트 rt 와 일치함(C1), ② t 개의 비밀 조각을 라그랑주 보간법으로 결합하여 지분율을 연산 구조 내부에서 복원함(C3의 재구성 부분, 복원값 비노출로 C6 충족), ③ 해당 지분율로 재계산한 정산액 y^{calc} 가 음원 플랫폼이 제출한 공개 입력 y^{plat} 과 일치함(C4·C5). 이와 같은 세 조건을 단일 증명으로 통합한 연산 구조는 로그 16건($B=16$, 트리 깊이 $D=4$, 임계치 $t=3$)을 한 배치로 처리하며, 41,588개의 산술 제약으로 이루어진다. 제약 수는 연산 구조의 복잡도와 증명 생성 비용을 가늠하는 지표이며, 외부에 공개되는 입력은 rt 와 y^{plat} 두 개뿐이다. 반면, 조각 커밋먼트 일치 검증(C2)과 승인 서명·임계치 충족 검증(C3의 승인 부분)은 t 개 조각을 활성화된 비공개 입력으로 사용하는 데까지만 구현하였으며, 이를 연산 구조 및 블록체인 스마트 컨트랙트 제약으로 통합하는 것은 향후 과제로 남긴다.

온체인 검증은 허가형 블록체인 Hyperledger Fabric에서 이루어지며, snarkjs(증명 생성 도구)와 gnark-crypto(검증기)의 데이터 형식이 달라 증명·검증기를 검증기가 이해하는 BN254 곡선 점 형식으로 변환하는 브리지를 구현하였다. 각 피어는 (rt, y^{plat}, π) 만으로 유효성을 판정하며 원본 로그는 열람하지 않는다. 이때 검증 요청 식별자 id 는 영지식 증명의 입력이 아니라, 정산 회차를 식별하고 승인 서명·검증 결과를 해당 회차에 바인딩하는 용도로만 트랜잭션에 포함된다. 검증에 성공하면 검증 요청 식별자인 id 를 키로 하여 $(rt, y^{plat}, H(\pi))$ 와 함께 검증 결과를 원장에 앵커링한다. 검증 컨트랙트는 Hyperledger Fabric의 체인코드(스마트 컨트랙트)로 구현하여 CCaaS(ChainCode-as-a-Service) 방식으로 배포하였으며, 보증 정책은 4개 조직 중 3개 이상의 서명을 요구하는 $OutOf(3, 4)$ 로 설정하였다.

한편, 실제 정산에서는 같은 음원이라도 소비된 지역(국내·국외)과 상품(정기 스트리밍·다운로드 등)에 따라 지분율이 달라지므로, 이를 반영하여 지역·상품 조합마다 다른 지분율을 적용하는 확장 연산 구조를 구현하였다. 각 레코드의 리프를 *Poseidon(cnt, price, region, product)*로 확장하여 지역·상품을 커밋에 함께 묶고, 연산 구조 내부에서 각 레코드가 속한 조합을 판별한 뒤 조합별 매출에 해당 지분율을 곱해 정산 총액을 증명한다. 소비 지역 5종 및 상품 4종(조합 20개, $B=16$) 기준 제약 46,596개, 상품만 구분하는 축소형(조합 4개)은 43,332개로, 두 경우 모두 기본 연산 구조(41,588개)와 큰 차이가 없으며, 정상 증명·검증과 위조 입력 거부를 확인하였다.

4.3 성능 평가

모든 성능 수치는 각 측정 설정마다 20회씩 측정하여 평균과 표준편차로 산출하였으며, 1회성 비용인 신뢰 설정은 회차마다 반복되는 운영 성능을 왜곡하지 않도록 측정에서 제외하였다. 그림 2는 SSS 조각 생성 시간을 나타내며, 최대 설정인 ($t=9, n=15$)에서도 평균 0.222 ms에 불과하여, 전체 처리 시간에서 무시할 수 있는 수준이다.

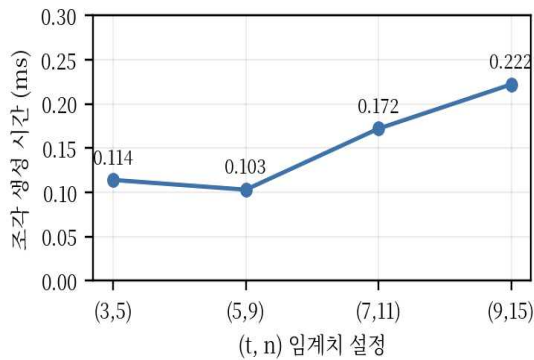


그림 2. SSS 조각 생성 오버헤드
Fig 2. SSS Share Generation Overhead

그림 3은 원천 이용 로그를 가명화하는 머클 트리 구성 시간(레코드별 Poseidon 해시 + 트리 구성)을 규모별로 나타낸 그래프이다.

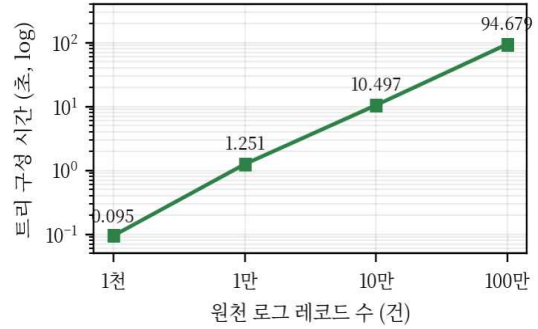


그림 3. 머클 트리 구성 오버헤드
Fig 3. Merkle Tree Construction Overhead

측정 결과 처리 시간은 레코드당 약 95 μ s로 선형 증가하여 100만 건에서 약 94.7초가 소요되며, 가명화가 오프체인 전처리 단계에서 가장 큰 비용을 차지함을 보여준다. 그러나, 이 비용은 로그 규모에 선형 증가에 그치고 정산 회차당 1회만 수행되는 오프체인 연산이므로, 규모와 무관하게 일정한 온체인 검증(약 0.87 ms)과 분리된다. 즉, 대용량 로그 처리의 부담은 오프체인 전처리에 집중되고 온체인 검증으로 전가되지 않으므로써, 제안 구조의 설계 목표를 실측으로 뒷받침한다.

증명 생성은 배치(로그 16건) 단위로 수행되며, 배치당 0.909초(± 0.148)로 측정되었다. 검증 시간은 Groth16 검증이 로그·배치 수에 관계없이 고정된 횟수의 페어링 연산만 수행하므로 데이터 양과 무관한 상수이다. 검증 연산 자체는 동일하나 실행 언어와 필드·페어링 연산의 최적화 수준에 따라 속도가 달라, JavaScript 기반 로컬 검증(snarkjs)은 10.69 ms(± 2.30), 네이티브 Go로 BN254 연산을 최적화한 온체인 검증(gnark)은 0.872 ms(± 0.120)로 Go 구현이 약 12배 빨랐다.

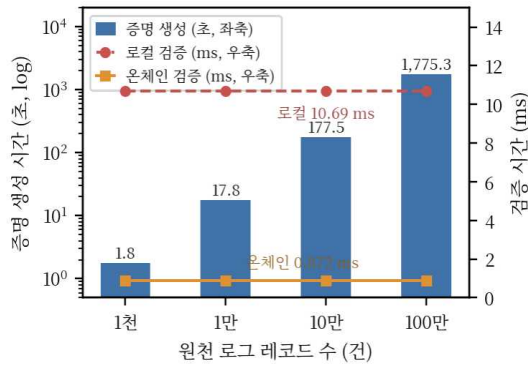


그림 4. 로그 규모별 총 증명 생성 및 검증 시간
Fig 4. Proof Generation & Verification Time by Log Scale

로그 규모별 총 증명 생성 시간은 배치당 실측치에 배치 수를 곱하고 병렬 워커 수(32코어)로 나누어 산출하였다. 그림 4는 로그 규모별 총 증명 생성/검증 시간을, 표 5는 로그 규모에 비례하는 오프체인 처리 시간(머클 구성·증명 생성)을 나타내며, 온체인 비용은 표 5의 어느 규모에서도 동일하다. 온체인에 제출되는 데이터는 id , rt , y^{plat} 와 Groth16 증명 π (128바이트)로 구성되어 로그 규모와 무관하게 1 KB 미만으로 일정하며, 표 6의 기본 구성 기준에서와 같이 온체인 검증 시간 역시 $0.872 \text{ ms}(\pm 0.120)$ 로 일정하다.

표 5. 로그 규모별 종합 성능 측정 결과
Table 5. Performance Measurements by Log Scale

로그 규모 (건)	머클 구성 (초)	증명 생성 (초)
1,000	0.095	1.79
10,000	1.251	17.75
100,000	10.497	177.53
1,000,000	94.679	1,775.27

목표 토폴로지(4조직/4피어/3 Raft 오더러, *OutOf*(3, 4) 보증)에서 20회 측정한 온체인 검증 지연은 표 6과 같다. 순수 검증 시간은 토폴로지와 연산 구조 복잡도(제약 수)에 모두 무관

하게 약 0.87 ms 내외로 거의 일정한데, 이는 Groth16 검증 비용이 공개되는 입력 수(2개; rt , y^{plat})에만 의존하고 연산 구조 규모에 비례하지 않기 때문이다. 실제로 제약 수가 41,588개인 기본 구조와 46,596개인 지역·상품 확장 구조의 검증 시간이 사실상 동일하여, 확장 연산 구조에서도 온체인 부담이 증가하지 않음을 실측으로 확인하였다. 반면, 중단 지연은 대부분 오더러의 배치 타임아웃(기본 2초)에서 기인하며 검증 연산 비용이 아니므로 참고치로만 제시한다.

표 6. 온체인 검증 지연 (20회 평균)
Table 6. On-Chain Verification Latency (20-run Avg.)

구 성	제약 수	순수 검증 (ms)	중단 지연 (ms)
기본, 4조직·3오더러	41,588	0.872 ± 0.120	$2,356 \pm 120$
확장, 4조직·3오더러	46,596	0.874 ± 0.246	$2,421 \pm 157$
기본, 2조직·1오더러	41,588	0.826 ± 0.067	$2,256 \pm 122$

제안 방식의 대규모 확장성을 검증하기 위해, 배치 크기를 기존 $B=16$ 에서 $B=1,024$ (트리 깊이 $D=10$)로 확대하여 연산 구조를 컴파일한 결과, 비선형 제약 약 277만 개(2,768,897개)·선형 제약 약 309만 개(3,087,363개) 규모였으며, 컴파일에 약 2분 51초, 최대 4.4 GB의 메모리가 소요되었다. 이로써 제안 연산 구조가 대규모 배치에서도 구조적으로 표현·생성될 수 있음과 그 제약 규모·컴파일 비용을 확인하였다. 다만 이 규모의 증명 생성·검증은 2^{22} (약 419만) 이상의 대용량 신뢰 설정을 요구하여 본 실험에서는 수행하지 않았으며, 실제 대규모 처리의 확장성은 작은 배치($B=16$)를 다수 생성하여 집계하는 4.2절의 배치·집계 구조로 달성된다.

4.4 기존 연구와의 비교 분석

Pan 등[13]은 Hyperledger Fabric 체인코드에

ZKP 검증을 결합할 때 보증 정책과 검증 연산의 상호작용으로 트랜잭션 처리 성능이 크게 저하될 수 있음을 보고하였다. 본 논문의 순수 검증 시간은 4조직·3오더러 구성에서 0.872 ms로 중단 지연(약 2.36초)의 0.04%에 불과하고 2조직·1오더러 구성과도 차이가 없어, 네이티브 Go(gnark-crypto) 검증기와 2개로 제한한 공개 입력에 의해 검증 연산이 병목이 되지 않음을 확인하였다. 증명 생성 측면에서 Siddiqui 등[14]은 Poseidon 머클 트리(깊이 5, 배치 32건)와 PLONK 기반 Fabric ZK-Rollup에서 8코어 기준 배치당 35-45초를 보고한 반면, 본 논문은 Groth16과 깊이 4(배치 16건) 연산 구조에서 배치당 0.909초를 측정하였다. 다만 증명 시스템, 연산 구조의 내용(머클 루트 정합성 대비 머클 포함 경로·지분율 재구성·정산 규칙), 하드웨어가 서로 달라 직접 비교는 어려우며, 두 연구 모두 증명 생성이 오프체인 병목이고 온체인 검증은 상수라는 동일한 구조적 결론을 지지한다는 점에서 의미가 있다. 한편, Kuznetsov 등[15]은 circom/snarkjs 기반 Groth16을 1개에서 약 151만 개 제약에 이르는 8중 연산 구조에서 벤치마크하여, 증명 크기와 검증 시간이 연산 구조 복잡도와 무관하게 상수(변동계수 5% 미만)로 유지됨을 통계적으로 확인하였다. 이는 본 논문에서 제약 수 41,588개와 46,596개의 두 연산 구조 간 온체인 검증 시간이 동일하게 측정된 결과와 일치한다. 다만 해당 연구의 검증 시간(약 0.66 초)과 증명 크기(JSON 직렬화 약 805바이트)는 측정 단위와 직렬화 방식이 본 논문(함수 단위 측정, 압축 곡선점 128바이트)과 다르므로 절대 값은 직접 비교할 수 없다.

5. 결 론

본 논문에서는 음원 정산 검증의 투명성과 비

밀 보호가 상충하는 문제를 해소하기 위하여, 영지식 증명과 임계치 비밀 공유를 결합한 zk-SSS 기반 프라이버시 보존형 음원 정산 무결성 검증 시스템을 제안하였다. 민감 계약 조건은 (t, n) 비밀 공유로 다수의 독립 노드에 분산 보관되어 임계치 승인 시에만 영지식 연산 구조의 비공개 입력으로 활성화되며, 대용량 원천 이용 로그는 Poseidon 머클 약정으로 가명화된 상태에서 정산 규칙 적용의 정합성이 단일 연산 구조의 결합 제약으로 증명된다. 허가형 블록체인의 다수 피어는 원본을 열람하지 않고 결과값·증명·머클 루트만으로 결과를 교차 검증한다. circom/snarkjs 및 Fabric 기반 프로토타입 실험 결과, 원천 이용 로그 100만 건 규모에서 머클 구성 약 94.7초, 병렬 증명 생성 약 1,775초, 증명 크기 128바이트를 보였으며, 온체인 검증은 로그 규모·연산 구조 복잡도와 무관하게 약 0.87 ms, 제출량은 1 KB 미만으로 일정하게 유지되었다. 검증·전송 비용의 규모 독립성은 대규모 정산에서도 온체인 부담이 증가하지 않음을 의미한다. 이는 권리자가 원천 데이터와 타인의 계약 조건을 열람하지 않고도 정산 결과의 정당성을 독립적으로 확인할 수 있게 하여, 투명성과 비밀성의 충돌을 실용적인 비용으로 해소함을 보여준다. 아울러 지역·상품 조합별 지분율을 적용하는 확장 연산 구조와 대규모 배치($B=1,024$) 연산 구조의 구성 가능성, 위조 입력 거부(건전성)를 실험으로 확인하였다.

본 연구의 벤치마크는 동일 호스트의 도커 구성에서 수행되어 물리 네트워크 지연을 포함하지 않고, 합성 데이터셋을 사용하였으므로 실제 이용 로그의 스키마 다양성과 이상 레코드에 대한 검증은 향후 과제로 남는다. 또한, 조각 커밋먼트 일치 검증(C2)과 승인 서명·임계치 충족 검증(C3의 승인 부분)은 아직 프로토타입 수준에 머물러 있어, 추후 이를 영지식 연산 구조 내부 제약으로 통합하는 작업이 요구된다. 나아가 창작자

와 권리자 및 이해관계자 간 권리 관계 정보를 공신력 있게 검증·추적할 수 있는 통합 검증 기능으로 확장하여, 신뢰 체계로 연결하는 것을 향후 연구로 추진할 계획이다.

본 연구는 문화체육관광부 및 한국콘텐츠진흥원 부설 한국문화기술기획평가원의 2026년도 글로벌 저작권 문제해결 사업으로 수행되었음 (과제명 : K-POP 창작자 저작권 보호를 위한 블록체인 및 AI 기반 음원 사용 이력 관리 기술 개발, 과제번호 : RS-2026-25520563, 기여율 : 100%)

참 고 문 헌

- [1] J. Jung, "The Levy System of Music Service and Revenue Structure of Copyright's Royalty in the On-line Music Market in Korea", *Journal of Digital Convergence*, 13(1), pp.429-438, 2015.1, DOI : 10.14400/JDC.2015.13.1.429
- [2] T. H. Kim, "A Study on the Advancement of the Music Copyright Management System: Focusing on the Copyright Registration System and Royalty Collection and Distribution", *Journal of the Korea Entertainment Industry Association*, 18(4), pp.395-413, 2024, DOI : 10.21184/jkeia.2024.6.18.4.395
- [3] Z. Zheng, S. Xie, H. N. Dai, X. Chen, H. Wang, "Blockchain Challenges and Opportunities: A Survey", *International Journal of Web and Grid Services*, 14(4), pp.352-375, 2018, DOI : 10.1504/IJWGS.2018.095647
- [4] J. W. Choe, "A Study on the Determination of the Music Copyright Royalty: Focused on the Collection Regulations of KOMCA, FKMP and RIAK", *Culture, Media, and Entertainment Laws*, 8(2), pp.61-90, 2014.12, UCI : I410-ECN-0102-2015-300-002191602
- [5] M. O'Dair, Z. Beaven, "The Networked Record Industry: How Blockchain Technology Could Transform the Record Industry", *Strategic Change*, 26(5), pp.471-480, 2017, DOI : 10.1002/jsc.2147
- [6] S. Goldwasser, S. Micali, C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems", *SIAM Journal on Computing*, 18(1), pp.186-208, 1989, DOI : 10.1137/0218012
- [7] J. Groth, "On the Size of Pairing-based Non-interactive Arguments", *Advances in Cryptology - EUROCRYPT 2016*, LNCS 9666, pp.305-326, Springer, 2016, DOI : 10.1007/978-3-662-49896-5_11
- [8] H. Kang, T. Dai, N. Jean-Louis, S. Tao, X. Gu, "FabZK: Supporting Privacy-Preserving, Auditable Smart Contracts in Hyperledger Fabric", *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp.543-555, IEEE, 2019, DOI : 10.1109/DSN.2019.00061
- [9] G. R. Blakley, "Safeguarding Cryptographic Keys", *1979 International Workshop on Managing Requirements Knowledge (MARK)*, pp.313-317, IEEE, 1979, DOI : 10.1109/MARK.1979.8817296
- [10] A. Shamir, "How to Share a Secret", *Communications of the ACM*, 22(11), pp.612-613, 1979.11, DOI : 10.1145/359168.359176
- [11] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function", *Advances in Cryptology - CRYPTO '87*, LNCS 293, pp.369-378, Springer, 1988, DOI : 10.1007/3-540-48184-2_32
- [12] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, M. Schofnegger, "Poseidon: A New Hash Function for Zero-Knowledge Proof Systems", *Proceedings of the 30th USENIX Security Symposium*, pp.519-535, USENIX Association, 2021, <https://www.usenix.org/conference/usenixsecurity21/presentation/grassi>
- [13] R. Pan, Z. Shi, A. Belloum, Z. Zhao,

“Operating ZKPs on Blockchain: A Performance Analysis Based on Hyperledger Fabric”, 2024 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS), IEEE, 2024, DOI : 10.1109/DAPPS61106.2024.00018

- [14] S. Siddiqui, Neha, K. Hari Babu, “ZK-Rollup for Hyperledger Fabric: Architecture and Performance Evaluation”, 2026, arXiv, <https://arxiv.org/abs/2602.08870> DOI : 10.48550/arXiv.2602.08870
- [15] O. Kuznetsov, Y. Kuznetsova, G. Ziyatbekova, Y. Kovalenko, R. Palahusynets, “Performance Evaluation of zk-SNARK Protocols for Privacy-Preserving Sensor Data Verification: A Systematic Benchmarking Study”, Sensors, 26(8), Art. 2486, 2026.4, DOI : 10.3390/s26082486

저 자 소 개



최창준(Chang-Jun Choi)

2019.02 : 상명대학교 컴퓨터공학과 졸업
 2021.08 : 세종대학교 정보보호학과 석사
 2021.09-현재 : 엘에스웨어(주) 소프트웨어연구소
 연구개발본부 책임연구원
 <주관심분야> 정보보호, 블록체인, 네트워크 보안, 분산신원증명, 저작권 기술



장성일(Sung-II Jang)

2019.08 : 숭실대학교 컴퓨터학과 석사
 2021.08 : 숭실대학교 소프트웨어학과 박사수료
 2021.09-현재 : 엘에스웨어(주) 소프트웨어연구소
 연구개발본부 수석연구원
 <주관심분야> 시스템 프로그래밍, 분산 컴퓨팅, 블록체인



조인희(In-Hee Jo)

2013.03 : 숭실대학교 컴퓨터학과 석사
 2025.08 : 숭실대학교 컴퓨터학과 박사수료
 2012.10-2025.10 : 디지캡(주) 책임연구원
 2025.11-현재 : 엘에스웨어(주) 소프트웨어연구소
 연구개발본부 수석연구원
 <주관심분야> 저작권 기술, 인공지능, 온톨로지



신동명(Dong-Myung Shin)

2003.02 : 대전대학교 컴퓨터공학과 박사
 2001-2006 : 한국정보보호진흥원(KISA)
 응용기술팀 선임연구원
 2006-2014 : 한국저작권위원회
 저작권기술팀 팀장
 2014-2016 : 한국스마트그리드사업단
 보안인증팀 팀장
 2016-현재 : 엘에스웨어(주) 소프트웨어연구소
 연구개발본부 연구소장/전무이사
 <주관심분야> 오픈소스 라이선스, 저작권 기술, 시스템/
 네트워크 보안, SW 취약점 분석·감정, 블록체인