

Verifiable Audit for Revocation-Aware DICOM Exchange on Blockchain

Hyo Hyun Choi*

*Professor, Dept. of Computer Science & Engineering, Inha Technical College, Incheon, Korea

[Abstract]

Cross-institutional medical image exchange is fundamental to patient-centered care, but transfer histories of sensitive images must be retained in a trustworthy form. This paper presents VADREX (Verifiable Audit for Revocation-Aware DICOM Exchange), an audit protocol and implementation framework that lets patients verify the finality of an anchored consent-revocation entry within the checked log history. VADREX combines append-only logs and log-backed maps, and uses cross-referenced entries and signatures to resolve disputes when the required records are anchored. It does not authenticate the revocation-request time or prove the absence of unlogged physical transfers. In a two-institution DICOMweb evaluation, proof size less than doubled as the log grew from 10^3 to 10^6 entries, individual proof verification took a few milliseconds, and all 90 supplementary evaluation outcomes across ten repetitions matched expectations.

▶ **Key words:** Medical Imaging, Consent Revocation, Verifiable Log, Sparse Merkle Tree, Non-repudiation, DICOMweb, Blockchain

[요 약]

의료기관 간 의료영상 교환은 환자 중심 의료의 기본 요소이지만, 민감한 영상의 전송 이력은 신뢰할 수 있는 형태로 남아야 한다. 본 논문은 환자가 앵커된 철회 엔트리를 기준으로 확인한 로그 범위에서 해당 동의의 종결성을 검증하는 감사 프로토콜 및 구현 프레임워크 VADREX(Verifiable Audit for Revocation-Aware DICOM Exchange)를 제안한다. VADREX는 append-only 로그와 log-backed map을 결합하고, 필요한 원문이 앵커된 경우 상호 참조와 서명으로 전송 분쟁을 판별한다. 철회 요청 시각의 진정성이나 기록되지 않은 물리적 전송의 부재는 입증하지 않는다. 2기관 DICOMweb 평가에서 로그가 10^3 에서 10^6 개로 늘어도 증명 크기는 두 배 미만이었으며, 개별 증명 검증은 수 밀리초 이내였으며, 보충 평가의 90개 결과 모두 기대 판정과 일치하였다.

▶ **주제어:** 의료영상, 동의 철회, 검증 가능한 로그, Sparse Merkle Tree, 부인방지, DICOMweb, 블록체인

-
- First Author: Hyo Hyun Choi, Corresponding Author: Hyo Hyun Choi
 - *Hyo Hyun Choi (hchoi@inhac.ac.kr), Dept. of Computer Science & Engineering, Inha Technical College
 - Received: 2026. 08. 11, Revised: 2026. 09. 03, Accepted: 2026. 09. 21.

I. Introduction

의료기관 간에 CT나 MRI같은 의료 영상 교환은 환자 중심 의료의 기본 요소이지만, 민감한 영상의 전송 이력은 신뢰할 수 있는 형태로 남아야 한다. 유럽 일반개인정보보호법(GDPR)은 동의 철회권과, 철회 뒤 다른 법적 근거가 없는 개인정보의 삭제를 규정한다[1]. 접근 차단과 별개로, 환자에게는 철회 뒤의 기록을 직접 확인할 근거가 필요하다. 본 연구는 이 문제 중 앵커된 철회 엔트리 뒤에 같은 동의의 전송 성립 이벤트가 기록되어 있는지를 사후에 확인하는 범위를 다룬다.

개인정보를 불변 원장에 직접 기록하면 삭제권과 충돌한다. VADREX는 직접 식별정보가 없는 집계 앵커만 온체인에 기록하고, 영상 원본과 직접 식별자는 기관 내부에 둔다. 이는 유럽 개인정보보호이사회(EDPB)의 Guidelines[2]가 제시한 온체인 개인정보 최소화 방향을 따른 것이다. 연구 범위는 삭제의 구현이나 규제 준수의 입증이지 아니라 앵커된 기록에 대한 감사다.

Off-chain 저장과 루트 앵커링은 MedRec[3] 이후 널리 쓰이는 구조가 되었으며, KSI(Keyless Signature Infrastructure)와 에스토니아 e-Health[4, 5]는 이를 국가 규모에서 운영하였다. 이 구조는 제시된 로그의 무결성을 보이지만, 앵커된 이력에서 항목이 빠진 경우까지 앵커 값만으로 밝히지는 못한다. 각 기관이 자기 로그만 앵커링하는 환경에서는 전송 분쟁을 제3자가 판별하기도 어렵다.

본 연구는 RFC 6962 로그[6], 변조 탐지 로깅[7], 검증 가능한 log-backed map[8, 9, 10, 11]을 의료 동의 철회의 요건에 적용하였다. 기여는 기존 기법을 철회 장벽, 동의별 head와 기관 간 상호 서명에 연결한 감사 프로토콜 및 구현 프레임워크에 있다. 이를 두 기관 DICOM(Digital Imaging and Communications in Medicine)web 테스트베드에 구현하고 정상 동작 비용과 위반 주입 탐지 결과를 측정하였다.

평가는 로그 완전성(S1), 철회 종결성(S2), 전송 부인방지(S3)를 대상으로 한다. S1은 앵커된 로그의 변조와 삭제를 확인하고, S2는 환자가 보관한 철회 엔트리의 로그 위치 이후부터 최신 검증 앵커가 포괄하는 범위까지 종결성을 확인한다. S3는 양 기관의 필요한 원문이 모두 앵커되고 상호 참조와 서명이 일치할 때 전송 분쟁을 판별한다. 철회 요청 시각과 접수 시각과 처음부터 기록되지 않은 전송은 이 증거만으로 확인할 수 없다.

II장은 배경과 관련 연구, III장은 위협 모델과 보안 속성, IV장과 V장은 설계와 구현을 설명한다. VI장은 평가 결과, VII장은 한계와 논의, VIII장은 결론을 제시한다.

II. Preliminaries

1. Append-Only Logs and Consistency Proofs

RFC 6962의 Merkle 로그는 항목을 오른쪽 끝에만 추가하고 하나의 루트로 전체 이력을 요약한다. 항목 수를 n 이라 할 때 포함 증명은 특정 항목의 존재를 형제 해시 $O(\log n)$ 개로 보이며, 일관성 증명은 나중 루트가 이전 항목을 보존하면서 확장되었음을 보인다[6]. 항목이 100만 개여도 형제 해시는 스무 개다. 검증자는 로그 전체 대신 루트 두 개와 이 정도의 해시로 로그가 바뀌지 않고 이어졌는지 확인할 수 있다.

그러나 일관성 증명은 앵커 사이의 확장 관계를 확인할 뿐, 지금 제시된 로그 원문이 그 앵커와 같은지는 확인하지 않는다. 공개된 두 루트를 그대로 둔 채 기관 내부의 로그만 바꾸면 기존 일관성 증명은 여전히 유효할 수 있다. 이 경우에는 제시된 로그 원문에서 루트를 다시 계산해 앵커와 비교해야 한다. 누적 루트와 항목 수를 공개 원장에 기록하는 과정을 앵커링이라 한다.

2. Log-Backed Maps and Current-Value Proofs

일관성 증명은 시간 순서의 무결성을 보이지만 개별 최신값에는 직접 답하지 못한다. VADREX는 값의 포함 증명과 비포함 증명을 제공하는 SMT(Sparse Merkle Tree)를 로그와 함께 둔다. 로그가 변경 이력을 맡고 맵이 특정 키의 존재와 부재를 나타내는 log-backed map은 CONIKS 이후 키 투명성 시스템에서 사용된 구조다[8].

GlassDB는 로그와 검증 가능한 색인을 결합하여 최신 값 증명(current-value proof)의 비용을 줄이는 구조를 설명한다[11]. VADREX는 이 구조를 이용하여 동의의 최신 상태가 계속 철회 상태인지를 확인한다. 특정 순번의 비포함만 검사하면 seq를 건너뛴 기록을 탐지하지 못하므로, 최신값을 대표하는 head를 함께 확인한다. 다만 본 구현은 깊이 256을 그대로 사용하므로 SMT 증명 크기는 키 수와 무관한 상수다.

3. Related Work

VAMS[10]는 Trillian 기반 log-backed map과 맵 헤드를 이용해 데이터 접근을 감사하고, 감사자와 사용자의 가십(gossip)으로 서로 다른 이력이 제시되는지를 확인한다. CONIKS[8]와 SEEMless[9]는 메시징 키의 최신값과 변경 이력을 검증한다. VADREX는 이 연구들의 최신값 검증 기법을 바탕으로 하지만, 의료영상 철회 엔트리 뒤의 종결성을 검사한다는 점에서 검증 대상이 다르다.

GlassDB[11]는 로그와 색인을 결합한 검증 가능한 데이터베이스다. VADREX는 최신값 검증을 동의별 head에 적용하되, 환자가 보관한 철회 응답과 온체인 앵커를 기준점으로 사용한다. 키 유도의 공개 검증이 필요한 CONIKS의 VRF와 달리, 본 구현은 비밀값을 가진 환자의 자가검증을 위해 HMAC을 사용한다.

AS2[12]는 메시지 다이제스트에 서명한 영수증으로 수신 증거를 남기며, InterSnap[13]은 교차 체인 영수증과 원장 스냅샷으로 거래를 감사한다. VADREX는 두 기관의 통상적인 로그 앵커에 상호 참조를 남겨, 별도의 전송별 온체인 교차 기록 없이 필요한 증거를 대조한다.

FHIRChain[14]과 WiraChain[15]은 FHIR 기반 임상데이터 공유와 권한 관리를 다룬다. Tajima[16]는 Orthanc와 DICOMweb에 블록체인 접근과 정산을 결합하고, Esperança 등[17]은 의료영상을 환자 지갑과 DID에 연결한다. Redact-Chain[18]은 수정 가능한 원장으로 공유 의료데이터의 관리와 삭제 문제에 대응한다. Table 1은 이 연구들의 주된 대상과 구조를 비교하였다.

Table 1. Scope and evidence of related work

Work	Main target	Evidence or integration
CONIKS [8], SEEMless [9]	Key-directory updates	Verifiable maps; user monitoring
VAMS [10]	Data-access auditing	Log-backed map; gossip
GlassDB [11]	Ledger history and current values	Authenticated log and index
AS2 [12], InterSnap [13]	Message / cross-chain disputes	Signed receipts / ledger snapshots
FHIRChain [14], WiraChain [15]	Clinical-data sharing and permissions	FHIR permissions on blockchain
Tajima [16], Esperança [17]	Medical-image exchange / reuse	DICOMweb / patient wallet
Redact-Chain [18]	Shared-data modification and deletion	Redactable blockchain
VADREX	Anchored revocation finality; transfer disputes	Revocation-bound head; anchored cross-signatures

선행 연구는 검증 가능한 상태 관리와 의료데이터의 권한 관리, 전송, 원장 수정을 다룬다. VADREX는 환자가 보관한 철회 응답을 기준으로 기록 종결성을 사후 검사하고, 양 기관의 앵커된 증거로 전송 분쟁을 판별한다. 환자 지갑은 필요하지 않다. Table 1의 비교는 감사 대상과 증거 구조에 한정한다.

III. Threat Model

1. Participants and Adversary

참여자는 환자, 제공기관과 수신기관의 게이트웨이, 블록체인, 검증자다. 환자는 지갑을 사용하지 않고 consentId, secret_c와 철회 응답을 보관한다. 동의의 seq는 홈 기관인 제공기관만 부여한다. 두 게이트웨이는 각자 로그와 SMT를 유지하여 앵커링하며, 검증자는 온체인 앵커와 이에 대해 검증한 증명만 신뢰한다.

악의적 기관은 자기 로그와 SMT를 바꾸거나 앵커링 시점을 조작할 수 있다. 또한 로그를 처음부터 남기지 않거나, 철회 후에 전송하거나, seq를 건너뛰거나, 메시지를 지연 또는 재전송을 할 수 있다. 그러나 상대 기관의 키를 위조하거나 온체인 앵커를 사후에 변경할 수는 없다. 해시 충돌, HMAC 위조와 정직한 상대 기관의 로그에 임의의 기록을 추가할 수 없다고 가정한다.

2. Trust Assumptions

위협 모델이 다루는 범위는 다음 여섯 전제에 의존한다. T1-T5는 증거와 운영 범위의 전제이며, T6은 정직한 제공기관의 철회 처리 규칙을 설명한다.

- T1 키 비유출 - 기관 서명 키는 유출되지 않는다. 키가 유출되면 부인방지(S3)의 근거를 잃는다.
- T2 블록체인 무결성 - 기록된 앵커는 변조하거나 삭제할 수 없으며 임의 시점에 조회할 수 있다. 이 가정이 깨지면 모든 검증 근거가 무너진다.
- T3 비담합 - 두 기관은 동시에 담합하지 않는다. 정직한 기관은 수신 완료로 기록한 뒤 영수증을 반환한다. 이는 정직한 구현의 동작 규칙이며, 비담합만으로 악의적 기관의 기록을 강제하지는 않는다.
- T4 환자 기준점 보관 - 환자(검증자)는 철회 응답 {seq, entryHash}를 정직하게 보관한다. 이를 잃으면 철회 검증(S2)의 기준점도 사라진다.
- T5 게이트웨이 종단성 - 모든 전송은 게이트웨이를 경유한다. 게이트웨이를 우회한 전송은 현재 검증 방법으로 확인할 수 없다.
- T6 정상 철회 장벽 - 정직한 제공기관은 접수 후 신규 승인을 막고 전송 정산 뒤 철회를 기록한다. 이는 정상 운영 규칙이지 악의적 기관의 준수를 미리 보장하는 가정은 아니다. S2는 기록 후 위반을 검사하나 접수와 기록 간 악의적 지연은 판별하지 못한다.

3. Security Properties

- S1 로그 완전성: 입력은 온체인 앵커, 일관성 증명과 로그 원문이다. T2와 해시의 안전성을 전제로 연속 앵커의 확장 관계 및 원문에서 재계산한 루트를 확인한다. 불일치하면 변조나 누락으로 본다. 성공은 제시된 로그와 앵커의 일치 여부를 뜻하며, 처음부터 미기록인 사건이나 앵커링의 적시성까지 보장하지 않는다.
- S2 철회 종결성: 입력은 환자가 보관한 consentId, secret_c, {seq=k, entryHash}, 양 기관의 앵커와 증명이다. T2와 T4를 전제로 철회 포함을 확인하고, 제공기관의 seq 추가와 건너뛰기는 head로, head 미갱신은 전체 로그 교차 확인으로 검사하며, k+1 비포함은 보조 검사로 사용한다. 수신기관은 첫 적격 앵커 이후 head의 변화를 검사한다(IV장 3절). 모두 통과하면 제공기관의 기록 종결성을 확인하고 수신기관의 보조 검사도 통과한 것으로 판정한다. 증거 불일치는 실패, 철회 또는 적격 수신기관 앵커가 없으면 waiting으로 처리한다. 이를 실제 전송의 준수로 해석할 범위는 T3, T5, T6에 제한되며, 요청 시각의 진정성과 미기록 전송의 부재는 입증하지 않는다.
- S3 전송 부인방지: 입력은 전송 승인(TRANSFER_APPROVED), 수신 완료(RECEIVE_COMPLETED), 전송 완료(TRANSFER_COMPLETED)의 원문, 포함 증명과 기관 공개키다. T1과 T2를 전제로 세 원문이 모두 앵커되고 상호 참조와 서명이 일치하면 전송 근거를 인정하고 수신 부인을 기각한다. 증거가 부족하면 전송 성립 주장을 인정하지 않는다. T3와 T5는 운영 범위의 전제이며, 비답함만으로 미기록 기관의 책임이 특정되지는 않는다.

IV. System Design

1. Overview and Design Principles

기관별 감사로그만으로는 환자와 검증자가 병원 내부 상태와 분리된 근거를 얻기 어렵다. 이를 위해 VADREX는 기관별 게이트웨이, Orthanc PACS, 로그 트리, SMT와 앵커 컨트랙트로 구성된다(Fig. 1). 게이트웨이는 동의 검증부터 앵커링까지 담당하고, 환자와 검증자는 기관 내부 DB가 아니라 API와 체인 RPC를 사용한다.

감사 자료를 원문째 온체인에 기록하면 의료영상 교환의 기밀정보가 노출되고 기록 비용도 커진다. 따라서 영상 원본, 식별정보, consentId, 비밀값, DICOM UID와 영상

해시는 온체인에 올리지 않고 앵커의 다섯 필드만 남기며, 등록 권한은 기관별 소유자로 제한한다. 환자는 블록체인 지갑을 사용하지 않고 오프체인에서 동의를 생성하고 철회하며, 채널과 키 관리는 V장에서 다룬다.

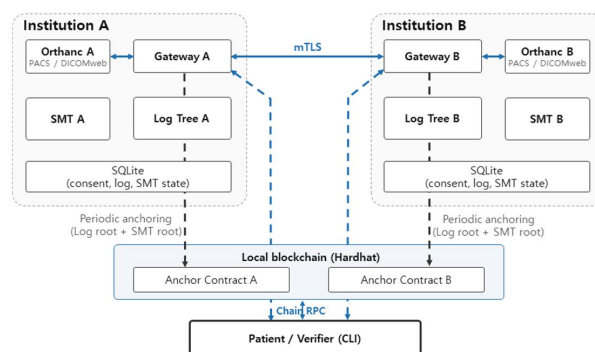


Fig. 1. System Architecture

2. Log Completeness (S1)

각 기관은 감사 이벤트를 RFC 6962 트리에 추가하고 누적 루트를 주기적으로 앵커링한다. 검증자는 연속 앵커 쌍의 일관성 증명을 확인하고, 앵커된 리프 수만큼 로그에서 루트를 재계산해 온체인 값과 비교한다. 이전 이력을 바꾼 새 루트는 일관성 검사에서, 앵커와 다른 원문을 제시하는 행위는 재계산 검사에서 드러난다(II장 1절). 둘을 통과해야 로그가 앵커와 일치하며 연속적으로 누적되었다고 판정한다. treeSize가 증가하지 않으면 컨트랙트가 등록을 거부하지만, 이것만으로 미기록 사건이나 앵커 지연을 탐지하는 것은 아니다.

이 판정은 기록되어 앵커링된 로그를 대상으로 한다. 전송이 처음부터 기록되지 않으면 그 기관의 트리에 흔적이 없으므로 S1의 대상이 아니다. 상대 기관의 앵커에 남은 엔트리는 단서가 되지만, 어느 기관이 기록을 누락했는지 판단하기에는 충분하지 않다(4절에서 이에 더 설명한다).

3. Revocation Finality (S2)

철회 엔트리가 로그에 포함되었다는 사실만으로는 그 뒤에 같은 동의의 전송 이벤트가 없는지 알 수 없다. "아무 일도 없었다"를 기록에서 확인하려면 원칙적으로 모든 기록을 뒤져야 하므로, VADREX는 이를 "동의를 마지막으로 가리키는 값이 그대로인가"라는 물음으로 바꾼다. k+1 키의 비포함만 검사하면 k+7처럼 순번을 건너뛴 기록을 놓칠 수 있으므로 head와 로그를 함께 확인한다.

철회 시점은 환자의 요청 전송 t_req, 게이트웨이의 접수 t_recv, 철회 엔트리 기록 t_rev, 그 엔트리를 포함한 앵커 등록 t_anchor로 구분한다. S2는 t_anchor에서 확

보된 증거로 철회 엔트리의 로그 위치 이후부터 최신 검증 앵커까지 검사한다. 엔트리의 timestamp는 요청 시각과 접수 시각을 인증하는 외부 증거가 아니다.

3.1. Consent Chain and Revocation Barrier

전송 성립 계열 이벤트(생성, 승인, 완료, 수신완료, 철회)만 seq를 받아 체인과 SMT에 기록한다. 요청, 거부와 실패 등은 seq 없이 로그 트리엔만 남으므로 철회 종결성 판정에 영향을 주지 않는다(Fig. 2). seq는 동의의 승인 제공기관만 부여하고, 각 이벤트는 같은 기관 로그의 직전 체인 엔트리를 prevHash로 가리킨다. CONSENT_REVOKED는 종결자(terminator)이며, 기록된 뒤에는 더 큰 seq를 허용하지 않는다.

Consent event chain - CONSENT_REVOKED terminates



Only CREATED, APPROVED, COMPLETED, RECEIVE_COMPLETED, and REVOKED enter the chain/SMT with seq. DENIED, FAILED, and REPLAY_BLOCKED use seq=null and remain log-only.

Fig. 2. Consent Event Chain

진행 중인 전송이 남은 채로 종결자를 기록하면 체인의 마지막 위치가 어긋난다. 따라서 제공기관은 철회 요청을 받으면 (1) 신규 승인을 즉시 차단하고, (2) 진행 중 전송의 결과가 TRANSFER_COMPLETED 또는 타임아웃에 따른 TRANSFER_UNCONFIRMED로 확정될 때까지 기다린다. 그 뒤 (3) CONSENT_REVOKED를 기록하고 {seq, entryHash}를 환자에게 반환한다. 이 철회 장벽(revocation barrier)을 거치면 정직한 시스템에서는 종결자보다 큰 seq의 체인 이벤트가 존재하지 않는다.

3.2. Head Key and Patient Verification

각 기관은 로그 트리와 별도로 SMT를 유지한다. consentId와 seq를 단순 해시하면 외부인이 식별자를 추측할 수 있으므로, EDPB의 keyed hash 권고[2]에 따라 $HMAC(secret_c, consentId \parallel ":" \parallel seq)$ 를 키로 사용한다. 동의별 최신 이벤트를 가리키는 $HMAC(secret_c, consentId \parallel ":"head)$ 도 함께 둔다. head의 값은 가장 최근 체인 이벤트의 entryHash이며, 로그 기록과 head 갱신은 같은 트랜잭션에서 처리한다. 정상적으로 철회되면 head는 철회 엔트리 e_rev를 가리키며, 이후 어떤 seq로 이벤트를 추가해도 값이 바뀐다(Fig. 3). 따라서 head 포함 증명을 주 검사로 사용하고, 제공기관의 k+1 키 비포함 증명을 보조로 확인한다.

SMT key: $HMAC(secret_c, consentId \parallel ":" \parallel seq)$
head tracks the latest chain event.

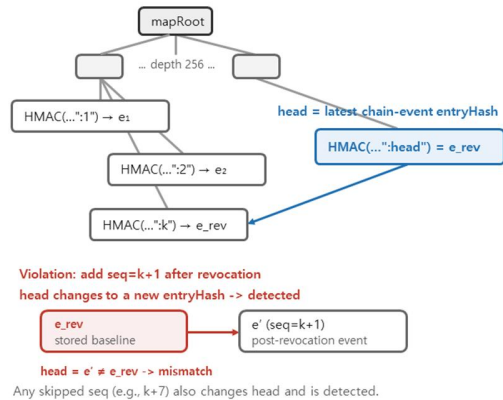


Fig. 3. SMT Head Key Verification after Revocation

환자는 consentId, secret_c와 철회 응답 {seq=k, e_rev}를 보관하고 앵커값은 블록체인에서 직접 읽는다. 절차를 의사코드로 나타내면 다음과 같다. 각 확인은 제공한 자료를 앵커 기준으로 재검증하는 단계다.

VerifyS2(consentId, secret_c, k, e_rev, Δ_{max}):

```

A, B ← 체인 RPC에서 읽은 양 기관 앵커
H_i ← e_rev가 포함된 제공기관 앵커
if H_i가 없으면: return waiting
확인: e_rev의 포함 증명, 동의, CONSENT_REVOKED, seq=k
확인: H_i부터 최신 A까지 모든 연속 앵커의 일관성
for H in A[i..latest]:
    확인: head=e_rev의 포함 증명, k+1의 비포함 증명
확인: 최신 A의 전체 로그 루트, 동의 목록과 링크, 철회 종결자
확인: B의 모든 연속 앵커의 일관성
E ← anchoredAt ≥ H_i.anchoredAt +  $\Delta_{max}$ 인 B의 앵커
if E가 비어 있으면: return waiting
확인: E의 head는 계속 부재이거나 동일한 값
값이 있으면 확인: 포함 증명,
    같은 동의의 RECEIVE_COMPLETED, seq<k
return success // 어느 확인이든 불일치하면 failure
  
```

head만 확인하면 로그에는 이벤트를 추가하고 head는 갱신하지 않은 경우를 발견하지 못할 수 있다. 따라서 검증자는 head가 가리키는 원문이 CONSENT_REVOKED (seq=k)인지, 그 뒤에 같은 동의의 체인 이벤트가 없는지를 로그와 교차 확인한다. 게이트웨이의 목록을 그대로 믿지 않고 각 엔트리의 리프 해시와 포함 증명을 확인하며, 전체 로그의 RFC 6962 루트를 다시 계산해 온체인 값과 비교한다. 이 $O(n)$ 검사는 검증할 때 한번만 수행하고, 각 앵커에서는 크기 변화가 작은 head 증명과 일관성 증명만 확인한다. 비용과 프라이버시 문제는 VII장에서 설명한다.

수신기관은 독립적으로 앵커링하므로 철회부터 다음 수신기관 앵커까지 지연이 생긴다. 이를 유예 구간(grace window)이라 하며, 정책 상한은 Δ_{max} 이다. 제공기관은

전체 로그를 다시 계산해 동의 체인을 확인하지만, 수신기관은 head 값과 해당 엔트리만 확인한다.

4. Transfer Non-Repudiation (S3)

기관마다 자기 로그만 남기면 제공기관이 전송을 주장하거나 수신기관이 수신을 부인할 때 어느 한쪽의 기록만으로 사실관계를 정하기 어렵다. VADREX는 상대 엔트리 해시와 서명을 서로의 로그에 남겨 두 로그가 같은 전송을 가리키게 한다(Fig. 4).

핸드셰이크는 요청, 승인, 전송, 수신, 완료의 다섯 단계로 진행되며 각 단계에서 상대 엔트리 해시에 서명한다. B의 요청 해시 h_{req} 가 A의 승인 엔트리에 들어가고, 이때 A가 부여한 seq와 파생 키가 B에 전달된다. 이어 A의 승인 해시 h_{app} 이 B의 수신 엔트리에, B의 영수증 해시 h_{rcv} 가 A의 완료 엔트리에 들어간다. 전송이 성립하면 양쪽 로그에 상대 기관의 서명과 이를 가리키는 엔트리가 함께 남는다.

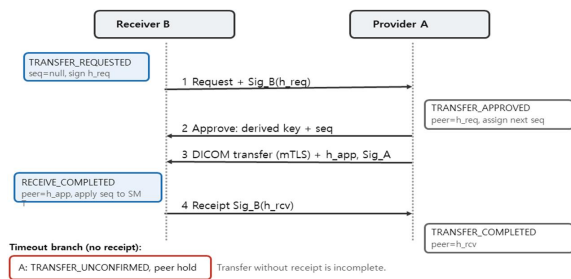


Fig. 4. Transfer Handshake

상호 참조 엔트리는 기관의 앵커에 포함되며, 본 연구에서는 이를 간접 앵커링(indirect anchoring)이라 부른다. 별도의 온체인 교차 기록이 없어 전송별 온체인 쓰기가 추가되지 않는다. B가 수신을 부인하면 A는 B의 서명이 담긴 완료 엔트리와 포함 증명을 제시한다. 검증자는 B의 앵커된 로그에서 수신 엔트리 원문을 받아 영수증 해시가 같은 동의에 속하는지 확인한다. 두 값이 일치하면 B는 자신의 서명을 부인하기 어렵기 때문에(T1) 검증자는 수신 부인을 기각한다. 영수증이 없으면 미완료로 판정한다.

B가 수신 엔트리 원문을 제공하지 않으면 영수증 해시를 특정 동의와 연결할 근거가 없다. 원문 없이 해시와 서명만 인정하면 다른 전송의 상대 서명을 재사용할 수 있다. 따라서 검증자는 전송이 입증되었다고 판단하지 않고 증거 불충분으로 판정한다.

이 판정은 한 기관이 기록을 누락한 경우와 상대 기관이 근거 없이 전송을 주장한 경우를 구분하지 않는다. B가 영

수증만 반환하고 수신 엔트리를 기록하지 않으면 A가 가진 것은 해시와 서명뿐이다. 앵커된 증거만 보면 이는 A가 승인만 하고 전송하지 않은 상태와 같다. 승인 엔트리에 대한 A의 서명도 DICOM 조회 전에 생성되므로 승인이 있었다는 사실만 보일 뿐, 실제 전송을 증명하지 않는다.

V. Implementation

VADREX는 TypeScript(Node.js 20+) 언어로 구현하였다. 스키마와 정규 직렬화, Merkle 자료구조, 게이트웨이, 검증기를 각각 독립 패키지로 두어 생성 측과 검증 측이 같은 코드로 해시를 계산하게 하였다. 블록체인은 Hardhat 로컬 노드에 Solidity ^0.8.20 앵커 컨트랙트를 배포하고 ethers.js v6을 통해 컨트랙트를 호출한다.

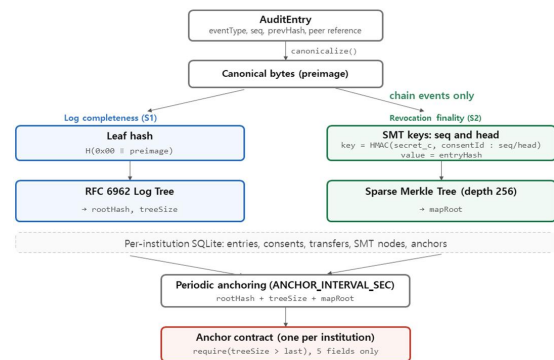


Fig. 5. Log Tree and SMT Anchoring Pipeline

기관 상태가 섞이지 않도록 동의, 트리, SMT와 앵커 이력은 기관별 SQLite 파일에 저장하고, PACS는 Orthanc의 DICOMweb 표준 인터페이스[19]를 제공한다. 기관 간 채널은 사실 CA 기반 mTLS로 일원화하며, 각 기관은 자기 키만 로컬 파일로 보관하고 상대 기관에서는 공개키만 참조한다. 그러나 실험을 위해 두 게이트웨이가 단일 호스트에서 같은 커널을 공유하므로 기관 분리는 논리적인 수준에 머문다. OS 수준의 격리는 구현하지 않았다. 상대 기관의 키를 위조할 수 없고 정직한 상대 기관의 로그에 임의의 기록을 추가할 수 없다는 가정은 두 기관이 서로 다른 관리 영역에서 각자의 키와 로그를 독립적으로 관리한다는 전제에 기반한다. 감사 엔트리가 두 트리에 반영되고 기관별 앵커로 기록되는 경로는 Fig. 5와 같다.

앵커 컨트랙트는 기관마다 한 인스턴스를 배포한다. 핵심 부분은 Fig. 6과 같다. 구조체는 다섯 필드뿐이며, 'require' 한 줄은 누적 리프 수가 계속 증가하도록 강제

한다. 따라서 줄어든 로그를 앵커하려는 조작은 오프체인 증명을 확인하기 전에 거부된다. 'batchId'와 기록 시각은 호출자가 아니라 컨트랙트가 부여하므로 앵커의 순서와 시각은 소급해 위조할 수 없다.

```
contract Anchor is Ownable {
    struct AnchorRecord {
        uint64 batchId;
        bytes32 rootHash;
        uint64 treeSize;
        bytes32 mapRoot;
        uint64 anchoredAt;
    }
    ...
    function registerAnchor(bytes32 rootHash, uint64 treeSize,
        bytes32 mapRoot) external onlyOwner {
        require(treeSize > lastTreeSize, "treeSize must increase");
        uint64 batchId = count + 1;
        uint64 anchoredAt = uint64(block.timestamp);
        anchors[batchId] = AnchorRecord({
            batchId: batchId,
            rootHash: rootHash,
            treeSize: treeSize,
            mapRoot: mapRoot,
            anchoredAt: anchoredAt
        });
        count = batchId;
        lastTreeSize = treeSize;
    }
    ...
}
```

Fig. 6. Anchor Contract Excerpt

트리 알고리즘은 외부 Merkle 라이브러리 없이 구현하였고, 순수 검증 함수는 Node 전용 API에 의존하지 않으므로 브라우저에서도 재사용할 수 있다. SMT는 값을 갱신할 때마다 해당 키의 256단계 경로에 있는 노드를 새로 생성하고, 변경되지 않은 나머지 노드는 기존 노드를 재사용한다. 과거 어느 앵커 시점의 mapRoot로도 증명을 만들 수 있는 것이 이 때문이다. 수신기관 파생 키는 HKDF-SHA256으로 얻는다.

게이트웨이는 한 프로세스에서 동의 관리, 전송 핸드셰이크, 감사 기록과 주기적 앵커링을 담당한다. 철회 장벽의 정산 대기는 영수증 타임아웃에 여유를 더한 시간으로 제한한다. 정산되지 않은 전송은 TRANSFER_UNCONFIRMED로 기록해 상대 기관의 후속 승인을 막는다. 저장 계층은 consentId 인덱스로 체인 이벤트를 기록 순서대로 조회하고 정규 바이트를 반환한다. 게이트웨이는 검증 자료만 제공한다. 신뢰의 근거는 체인에서 직접 읽은 값과 검증기의 재계산 결과다. 앵커 컨트랙트 주소와 Ed25519 공개키도 게이트웨이에서 받지 않고 신뢰할 수 있는 로컬 파일에 미리 지정한다. 모든 실험과 종단 시나리오에 이 방식을 적용하였다.

수신기관이 철회를 아직 반영하지 않은 상태를 실패로 처리하면 정상적인 앵커링 지연을 공격으로 오인할 수 있다. 따라서 검증기는 철회 앵커 시각에 $\Delta_{\max}$ 를 더한 기준 시각 이후의 앵커만 검사하고, 해당 앵커가 없으면 waiting을 반환한다. 판단하기 이른 상태는 위반으로 보지 않으며, 기준 시각 전후의 실행으로 확인하였다.

각 패키지는 단위 및 통합 테스트를 갖추며, 위변조 탐지는 성공 케이스와 탐지 케이스를 함께 확인한다. 전송 성립, 분쟁 판별과 철회 검증의 정상 및 위반 흐름은 종단 시나리오 스크립트로 재현할 수 있다.

평가 전용 기능은 일반 경로와 분리하였다. 앵커 증가와 위반 주입 엔드포인트는 환경변수를 설정한 경우에만 활성화된다. 전송 기준선은 평가 프로그램에서 측정하고, 앵커링 주기도 환경변수로 설정해 독립 변수로 사용한다.

VI. Evaluation

1. Experimental Setup

측정은 Windows 호스트(Intel Core i7-11700K, 메모리 32 GB, Node.js v20.16)의 Docker Compose 스택과 Hardhat 로컬 노드에서 수행하였다. 기존 평가(paper-v1)는 DB와 체인을 초기화한 한 번의 실행 안에서 각 조건을 10회 반복하였다. 여섯 측정군을 고정된 순서로 실행해 총 360행을 남겼으며, 반복마다 초기화한 10개의 독립 실행을 뜻하지 않는다. 본문의 $\pm$ 는 반복값의 평균과 표준편차이며, 기존 집계와 같게 표준편차의 분모는 $n=10$ 을 사용한다.

VI장 2절의 정상 동작 수치는 기존 평가 자료다. 앞선 측정군의 앵커가 뒤 측정군에 누적되므로 전체 검증 시간에는 그 이력이 포함된다. VI장 3절의 시간 분리와 gas는 별도 로컬 환경에서 보충 측정하였다. 보충 측정도 반복 사이 상태를 누적되 기존 자료와 구분해 기록하였다. 기존 시스템과 평가 프로그램은 GitHub에 공개하였으며 [20], 보충 측정의 계측 도구와 원자료도 같은 저장소에 실행별로 구분해 공개하였다.

2. Normal-Operation Costs

2.1. Proof Scaling (S1)

로그 규모를 10^3 ~ 10^6 리프로 늘려 세 증명의 크기와 검증 시간을 측정하였다. 로그가 1,000배 커지는 동안 증명 크기는 두 배가 채 되지 않았다(Fig. 7). 포함 증명은 753→1,449바이트(형제 해시 10→20개), 일관성 증명은 688→1,177바이트(9→16개)로 로그적으로 증가하였고, 깊이가 256으로 고정된 SMT 비포함 증명은 17,791바이트로 일정하였다. 검증 시간은 포함 증명과 일관성 증명이 0.18~0.29밀리초, SMT 비포함 증명이 2.7~3.4밀리초였으며, 표준편차 범위에서 규모에 따른 차이는 나타나지 않았다.

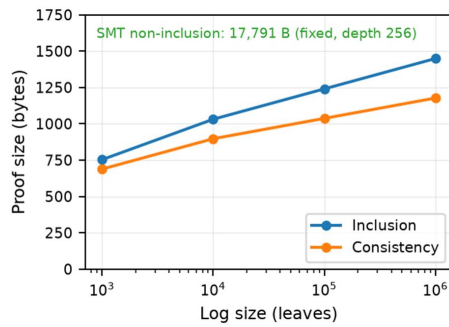


Fig. 7. Proof Size by Log Size

2.2. Revocation Anchoring Delay (S2)

철회 API 호출부터 앵커링까지의 지연은 설정한 주기 T와 거의 같았다(T = 10, 30, 60, 300초에서 평균±표준편차 10.4±2.3, 30.2±1.3, 60.2±1.6, 300.1±2.7초). 평가 프로그램이 철회 직후 다음 앵커링을 기다리는 과정을 순서대로 수행하므로, 철회 요청과 앵커링 시점 사이의 간격이 매번 비슷하게 유지된다. 따라서 측정된 지연은 주기 T에 가까운 상한값이며, 철회 요청이 앵커링 주기 내 균등하게 발생하고 추가 정산 지연이 없다면 다음 앵커까지의 평균 대기시간은 약 T/2가 된다.

2.3. Handshake Overhead (S3)

약 5, 50, 200 MiB(1 MiB = 1,048,576바이트)의 Study에서 게이트웨이 전송을 Orthanc 직접 복사와 비교하였다. 크기 순서대로 직접 복사는 0.161±0.016, 1.509±0.026, 6.146±0.135초, 게이트웨이 경로는 0.900±0.080, 2.708±0.116, 8.688±0.240초였다. 추가 지연은 0.739±0.079, 1.200±0.124, 2.542±0.233초다. 각 반복의 픽셀 바이트 수를 시간으로 나눈 유효 처리량은 직접 복사 31.39±2.80, 33.15±0.56, 32.56±0.71 MiB/s, 게이트웨이 5.60±0.46, 18.50±0.79, 23.04±0.63 MiB/s였다. 이는 프로토콜 부가 바이트를 제외한 값이다. 추가 지연에는 서명, 핸드셰이크, 감사 기록과 게이트웨이 간 base64 JSON 영상 중계 비용이 포함되며, 기존 원자료로는 각각의 비중을 구분하지 않았다.

2.4. End-to-End Verification Cost (S2)

제공기관 앵커를 1, 10, 50, 100개로 늘렸을 때 전체 검증 시간의 10회 평균±표준편차는 각각 1.66±0.22, 5.72±1.74, 28.60±8.73, 84.53±19.97초였다. 같은 기간에 누적된 수신기관 앵커도 함께 검사했으며, 그 수는 평균 8.5, 68, 388, 1,163개였다. 따라서 평균과 편차에는 반복 간 수신기관 앵커 누적의 영향도 포함된다. 개별 증

명 검증은 밀리초 수준이지만, 전체 시간은 앵커별 증명 순회와 전체 로그 교차 확인에 따라 증가한다. 누적 앵커 수뿐 아니라 재계산할 로그 크기도 비용에 영향을 준다. 검사별로 줄일 수 있는 범위는 VII장에서 구분한다.

3. Violation Injection Detection

기존 평가의 아홉 위반 조건 90개 결과는 모두 기대 판정과 일치하였다. 시간의 의미를 구분하기 위해 같은 조건을 별도 환경에서 각각 10회 보충 측정하였다. 정상 철회가 성공하는 것을 먼저 확인하고, 각 반복의 로컬 검증 계산 시간, 시나리오 전체 시간과 판정을 함께 기록하였다(Table 2). 계산 시간은 해당 판정까지 실제 수행한 해시, 증명, 서명, 로그 재계산과 조건 검사이며, API와 RPC 자료 조회와 대기는 제외한다. 첫 실패에서 종료하므로 성공 경로 전체의 비용과 같지는 않다.

Table 2. Evaluation outcomes (10 repetitions)

Property	Case	Correct	Compute (ms)	Total (s)
S1	(1) Historical log tampering (on-chain)	10/10	0.16±0.06	9.55±1.56
S1	(2) Log-entry omission	10/10	3.55±1.89	1.26±0.64
S1	(3) Post-anchor completion deletion	10/10	5.25±2.33	10.54±1.10
S1	(4) Reused treeSize (revert)	10/10	N/A	5.55±0.62
S2	(5) Append seq=k+1	10/10	20.54±2.24	17.62±0.96
S2	(6) Skip to seq=k+7	10/10	20.83±2.28	17.38±0.81
S2	(7) Log-only append; stale head	10/10	36.31±7.70	17.58±1.08
S3	(8) Receiver denial after anchoring	10/10	27.01±8.26	12.63±2.19
S3	(9) No timely completion pair	10/10	26.75±8.31	23.18±1.22

(1)-(3)과 (5)-(7)은 주입한 위반의 탐지를, (4)는 컨트랙트의 등록 거부, (8)은 앵커된 수신 근거에 대한 부인의 기각을, (9)는 적시에 앵커된 완료 쌍이 없는 전송 성립 주장의 기각을 나타낸다. Correct는 탐지, 거부와 기각을 구분하지 않고 각 항목에서 기대 판정과 일치한 반복 횟수를 뜻한다. 계산은 밀리초, 전체 시간은 초 단위의 평균±표준편차다. (4)의 N/A는 로컬 검증기 대신 컨트랙트가 거부한다는 뜻이다.

전체 시간은 사례 준비와 위반 주입부터 판정 또는 스크립트 종료까지이며, 통신, 앵커 형성, 영수증 대기, 내부 재시작을 포함한다. S2와 S3에는 계측용 검증기의 tsx 실행과 판정 재생 확인도 포함된다. 공통 환경 설정과 사전

빌드는 제외하고 S2는 사례별로 측정했다. 별도 DB와 체인의 예비 실행 뒤 10회 동안 앵커는 A 13→123, B 5→35개로 누적되었으며, 본 측정의 로그 리프는 A 26→235, B 8→68개였다. (2), (3), (7)-(9)의 계산에는 전체 로그 재계산이 포함되어 로그 크기의 영향을 받는다. Study는 0.25 MiB, 앵커 주기는 30초이며 수동 앵커도 사용했다. S2는 제공기관 검사에서 종료되므로 grace=0을 사용했고, (9)는 영수증 타임아웃 5초, 지연 10초로 실행했다. 기존 평가의 약 1,650개 수신기관 앵커 조건과 달라 속도 개선으로 해석하지 않는다.

S1의 (1)은 조작한 다음 루트를 격리 컨트랙트에 등록한 뒤 일관성 증명 실패로 탐지하였다. (2)는 일부 항목을 뺀 로그로 앵커 루트를 재현하지 못하는 경우다. (3)은 정상 전송을 앵커한 뒤 제공기관 DB 사본의 TRANSFER_COMPLETED를 삭제하여, 루트 불일치와 수신기관에 남은 대응 엔트리를 확인하였다. (2)와 (3)의 계산 시간에는 삭제한 로그의 루트 재계산과 비교가 포함된다. (4)는 최신 앵커와 같은 treeSize를 다시 등록해 컨트랙트의 증가 조건에 따라 거부되었다.

S2의 (5)-(7)은 각각 따로 실행하여 시간을 기록하였다. (5)와 (6)은 제공기관 head 변경으로, (7)은 head가 그대로인 상태에서 전체 로그의 후속 이벤트를 확인하여 탐지하였다. 철회 기록 이전의 악의적 지연은 현재 증거로 판별할 수 없어 주입 평가에 포함하지 않았다. 이는 T6의 정상 동작 규칙으로 해결된 문제가 아니라 S2의 검증 범위 밖에 남는 한계다.

S3의 (8)은 B가 RECEIVE_COMPLETED를 기록하고 앵커한 뒤 수신을 부인한 경우다. 검증기는 앵커된 원문과 B의 서명을 대조해 부인을 기각하였다. 이는 세 원문이 모두 앵커되고 상호 참조와 서명이 일치할 때 성립하는 부인 방지에 해당한다. (9)는 영수증이 타임아웃 이후 도착하여 A에는 TRANSFER_COMPLETED 대신 RECEIPT_LATE만 기록된 상태에서 전송 성립을 주장한 경우다. 검증기는 적시에 앵커된 완료 쌍이 없다는 이유로 주장을 기각하였다(IV장 4절). 이 기각은 물리적 수신 자체를 부정하지 않고, 어느 기관이 기록을 누락했는지는 특정하지 않는다.

VII. Limitations & Discussion

요청 시점의 증명 부재는 본 설계의 핵심 한계다. 제공기관이 철회 기록을 늦춘 사이 전송하면 현재 검증으로 판별할 수 없다. 환자가 보관한 {seq, entryHash}는 철회 엔

트리의 위치를 확인할 뿐 t_req나 t_recv의 진정성을 입증하지 못하기 때문이다. 따라서 S2의 성공을 요청 직후부터 물리적 전송이 전혀 없었다는 뜻으로 해석할 수 없다. 본 연구의 보장은 T1-T6이 정한 범위에서 확보된 앵커와 기록에 한정된다.

수신기관의 앵커링 지연으로 유예 구간(grace window)이 생기며 정책 상한은 $\Delta_{\max}$ 이나, 프로토타입은 이를 강제하거나 초과를 자동 판정하지 않는다. 수신기관 검사는 첫 적격 앵커부터 head가 없거나 같은 값으로 유지되는지 확인하고, 값이 있으면 seq<k인지 확인한다. 제공기관이 자기 로그에 남기지 않은 수신이 seq<k로 첫 적격 앵커에 이미 포함되면, 정직한 수신기관이 기록했어도 철회 전 수신과 구분하지 못해 통과할 수 있다. 수신기관의 로그↔SMT 불일치와 제공기관 은폐가 함께 일어나면 탐지할 수 없다. 수신기관에도 전체 로그 교차 확인을 적용하면 후자의 불일치를 검사할 수 있으나 비용이 늘어난다.

검사별 최적화 범위는 다르다. 연속 앵커 쌍의 일관성 증명은 누적 이력의 확장 관계를 확인하므로 모두 검사해야 한다. 제공기관은 최신 앵커에서 전체 로그를 교차 확인하므로 head 증명도 최신 앵커 하나로 줄이는 방안을 검토할 수 있다. 수신기관은 전체 로그 교차 확인을 하지 않으므로 모든 적격 앵커의 head 증명이 필요하다. 이 최적화는 이번 평가에 적용하지 않았다.

S3는 전송 승인, 수신 완료와 전송 완료의 세 원문이 모두 앵커된 경우에만 부인방지를 제공한다. 처음부터 기록되지 않은 엔트리는 증거 불충분으로 판정할 뿐 누락 기관을 가릴 수 없다. 이를 보완하려면 수신 원문을 영수증에 넣고, 제공기관이 DICOM 전달 후의 값에 서명해 수신기관이 보관해야 한다.

HMAC 기반 SMT 키는 식별자 은닉에 사용되지만, VRF와 달리 비밀값을 모르는 제3자가 키 파생 결과의 정당성을 공개적으로 검증하는 기능은 제공하지 않는다. 전체 로그 검사는 완전성을 높이는 대신 다른 동의의 가명 엔트리를 노출하고 로그 크기에 따른 비용을 만든다. 영수증이 오지 않으면 공정 교환을 완전히 보장하기 어렵고, TRANSFER_UNCONFIRMED와 RECEIPT_LATE로 증거를 남긴다.

로컬 Hardhat에서 Solidity 0.8.20, optimizer 200회, 컴파일 대상 EVM paris로 gas estimate를 10회 측정하였다. 배포는 444,021±0, 최초 앵커 등록은 160,481±0, 후속 등록은 143,369±0 gas였으며 영수증의 gasUsed와 같았다. 각 반복은 새 컨트랙트와 동일한 입력 해시를 사용했다. 이는 온체인 연산량이며 공개망의 수수료, 혼잡,

L1/L2 지연이나 실용성을 입증한 결과는 아니다.

평가는 합성 DICOM과 순차 실행을 사용했다. 동시 요청에서는 같은 동의의 순번과 head를 갱신하는 순서, 철회 장벽의 정산 순서를 유지해야 하며, DB 쓰기와 PACS 처리의 경합으로 지연이 늘 수 있다. 동시 전송 처리량이나 임상 환경의 응답시간은 이번 결과로 추정하지 않는다. 정상 검증 비용과 유예 구간에는 누적 수신기관 앵커의 영향이 포함된다.

VAMS, CONIKS, SEEMless[8, 9, 10]와는 상태 검증의 기반을 공유한다. 의료데이터 및 영상 공유 연구[14, 15, 16, 17, 18]와 비교하면, 본 연구는 권한 관리와 전송 자체보다 환자가 보관한 철회 응답을 기준으로 기록의 종결성을 확인하는 데 초점을 둔다. AS2[12]의 서명 영수증 보관에서 더 나아가 양 기관의 앵커와 증거를 대조하며, InterSnap[13]의 별도 스냅샷 아카이브 대신 기존 로그 앵커를 이용한다. 그 대신 세 원문이 확보되어야 하고 미기록 기관의 책임은 판별하지 못한다. 이 비교는 적용 범위와 비용의 차이를 설명한다. 선행 시스템보다 빠르거나 더 강한 보장을 제공한다는 뜻은 아니다.

규제 준수 여부는 다루지 않았다. HMAC 키 은닉은 익명화나 암호학적 삭제(cryptographic erasure)를 제공하지 않는다. append-only 감사로그는 책임성을 위한 가명 기록이지만, 그 보존을 삭제권의 예외로 인정하지는 별도 검토가 필요하다.

VIII. Conclusion

VADREX는 기관별 감사로그와 상호 서명 참조를 결합하여 DICOM 교환에서 앵커된 로그의 누락과 전송 부인을 검증한다. 여기에 log-backed map의 최신값 증명을 붙여 철회가 그 동의의 마지막 이벤트로 남아 있는지를 확인하게 하였다. 환자는 지갑 없이도 앵커된 철회 엔트리의 로그 위치 이후부터 최신 검증 앵커까지 종결성을 검사할 수 있다. 이는 요청 시각이나 미기록 전송의 부재에 대한 증명은 아니다. 평가에서 로그가 1,000배 커지는 동안 증명 크기는 두 배가 채 되지 않았고, 개별 증명 검증은 수 밀리초 이내였다. 보충 평가의 90개 결과도 모두 기대 판정과 일치하였다. 전체 검증 비용은 누적 앵커 순회와 전체 로그 교차 확인의 영향을 받았다.

이번 평가는 로컬 노드의 gas 연산량까지 확인했으며, 공개망의 실제 비용과 지연, 동시 전송과 임상 환경은 다루지 않았다. 요청 시점까지 검증 범위를 넓히려는 환자

서명 요청과 접수 확인서, 외부 시각 증거의 관계를 별도로 설계해야 한다. 영수증에 수신 원문과 전달 이후의 서명을 남겨 미기록 전송의 책임을 판별하는 방안도 후속 과제로 남는다.

ACKNOWLEDGEMENT

The author defined the research problem, designed the system, and interpreted the experimental results. Generative AI was used to assist with implementing the prototype, automating the evaluation, and checking consistency between the design and the implementation. The author personally reviewed and verified the generated code and check results. (Anthropic. Claude. Opus 4.8. <https://claude.ai/code>. June–July 2026. Gen AI.; OpenAI. Codex. 5.5. <https://chatgpt.com/codex>. June–July 2026. Gen AI.)

REFERENCES

- [1] European Parliament and Council of the European Union, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, L 119, pp. 1–88, May 2016, Articles 7(3) and 17(1)(b). <https://eur-lex.europa.eu/eli/reg/2016/679/oj/eng>
- [2] European Data Protection Board, Guidelines 02/2025 on Processing of Personal Data through Blockchain Technologies, Version 2.0, July 2026. https://www.edpb.europa.eu/system/files/2026-07/edpb_guidelines_202502_blockchain_v2_en.pdf
- [3] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," Proceedings of the 2nd International Conference on Open and Big Data (OBD), pp. 25–30, Vienna, Austria, August 2016. DOI: 10.1109/OBD.2016.11.
- [4] A. Buldas, A. Kroonmaa, and R. Laanoja, "Keyless Signatures' Infrastructure: How to Build Global Distributed Hash-Trees," Secure IT Systems (NordSec 2013), LNCS 8208, pp. 313–320, Ilulissat, Greenland, October 2013. DOI: 10.1007/978-3-642-41488-6_21.
- [5] J. Metsallik, P. Ross, D. Draheim, and G. Piho, "Ten Years of the e-Health System in Estonia," Proceedings of the 3rd International Workshop on (Meta)Modelling for Healthcare Systems (MMHS 2018), CEUR Workshop Proceedings, Vol. 2336, pp. 6–15, Bergen, Norway, June 2018; proceedings published March 2019.

- https://eur-ws.org/Vol-2336/MMHS2018_invited.pdf
- [6] B. Laurie, A. Langley, and E. Kasper, "Certificate Transparency," RFC 6962, IETF, June 2013. DOI: 10.17487/RFC6962.
- [7] S. A. Crosby and D. S. Wallach, "Efficient Data Structures for Tamper-Evident Logging," Proceedings of the 18th USENIX Security Symposium, pp. 317–334, Montreal, Canada, August 2009.
- [8] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman, "CONIKS: Bringing Key Transparency to End Users," Proceedings of the 24th USENIX Security Symposium, pp. 383–398, Washington, D.C., USA, August 2015.
- [9] M. Chase, A. Deshpande, E. Ghosh, and H. Malvai, "SEEMless: Secure End-to-End Encrypted Messaging with less Trust," Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, pp. 1639–1656, London, UK, November 2019. DOI: 10.1145/3319535.3363202.
- [10] A. Hicks, V. Mavroudis, M. Al-Bassam, S. Meiklejohn, and S. J. Murdoch, "VAMS: Transparent Auditing of Access to Data," arXiv preprint arXiv:1805.04772v5, May 2023. DOI: 10.48550/arXiv.1805.04772.
- [11] C. Yue, T. T. A. Dinh, Z. Xie, M. Zhang, G. Chen, B. C. Ooi, and X. Xiao, "GlassDB: An Efficient Verifiable Ledger Database System Through Transparency," Proceedings of the VLDB Endowment, Vol. 16, No. 6, pp. 1359–1371, February 2023. DOI: 10.14778/3583140.3583152.
- [12] D. Moberg and R. Drummond, "MIME-Based Secure Peer-to-Peer Business Data Interchange Using HTTP, Applicability Statement 2 (AS2)," RFC 4130, IETF, July 2005. DOI: 10.17487/RFC4130.
- [13] T. Sengupta, B. C. Ghosh, S. Chakraborty, and S. Sural, "Auditable Ledger Snapshot for Non-Repudiable Cross-Blockchain Communication," IEEE Transactions on Services Computing, Vol. 19, No. 1, pp. 181–194, January 2026. DOI: 10.1109/TSC.2025.3630671.
- [14] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data," Computational and Structural Biotechnology Journal, Vol. 16, pp. 267–278, July 2018. DOI: 10.1016/j.csbj.2018.07.004.
- [15] L. Grau, J. Urrunaga, and J. Santisteban, "WiraChain: A Blockchain and FHIR-Based Platform for Improving Clinical Data Interoperability in Healthcare," Frontiers in Public Health, Vol. 14, Article 1746772, February 2026. DOI: 10.3389/fpubh.2026.1746772.
- [16] Y. Tajima, "Decentralized Patient-Centric Medical Image Exchange: A Hybrid Blockchain and DICOMweb Architecture," medRxiv preprint, December 2025. DOI: 10.64898/2025.12.29.25343172.
- [17] M. Esperança, T. Galvão, D. Freitas, J. C. Ferreira, A. Cysneiros, and L. Bento, "Integrating Medical Imaging Datasets with Blockchain Wallets: A Case Study on ARDS-COVID19 Patients," PLOS ONE, Vol. 21, No. 3, e0338897, March 2026. DOI: 10.1371/journal.pone.0338897.
- [18] J. Hu, K. Huang, G. Bian, and Y. Cui, "Redact-Chain for Health: A Scheme Based on Redactable Blockchain for Managing Shared Healthcare Data," Electronics, Vol. 12, No. 20, Article 4240, October 2023. DOI: 10.3390/electronics12204240.
- [19] DICOM Standards Committee, DICOM PS3.18: Web Services, NEMA, <https://dicom.nema.org/medical/dicom/current/output/htm/part18/ps3.18.html>
- [20] VADREX, "VADREX: Verifiable Audit for Revocation-Aware DICOM Exchange - Source Code and Evaluation Harness," GitHub, <https://github.com/VADREX-git/VADREX>

Authors



Hyo Hyun Choi received his B.S., M.S., and Ph.D. degrees in computer science and engineering from Sogang University, Seoul, Korea in 1994, 1996, and 2005, respectively. He joined Telecommunication R&D Center of

Samsung Electronics as a senior engineer in 2005, where he was in charge of the research on the wireless sensor network. In 2009, he joined the faculty of the Department of Computer Science and Engineering, Inha Technical College, Incheon, Korea, as a professor. His research interests are blockchain, wireless sensor network, urban air mobility, unmanned aerial vehicle swarm control and internet of things.