Trust the Voice, Hide the Source: Anonymous Provenance for Verifiably Edited Audio

Xiyuan Fu, Zixing Wang, Yuncong Zhang, and Yu Chen(✉)

School of Cyber Science and Technology, Shandong University, Qingdao 266237, China
{xiyuanfu, zixing}@mail.sdu.edu.cn, yuncong@sdu.edu.cn, yuchen.prc@gmail.com

Abstract

As synthetic speech becomes increasingly realistic, the need to authenticate audio recordings has become more urgent. Such authentication should allow a released recording to demonstrate that it was captured by an authorized device and modified only through the declared edits, while concealing both the unreleased content and the identity of the particular recorder. In this work, we propose PPAAS, a privacy-preserving framework for authenticated audio editing under a formal security model. The framework integrates a privacy-preserving capture-authentication scheme and a specialized zero-knowledge proof protocol for authenticated editing. The capture-authentication scheme establishes that the recording was produced by an authorized device group while concealing both the unreleased content and the individual recorder. The proof protocol certifies that the published audio results from a valid editing process and that the declared edits were applied correctly, while avoiding general-purpose circuit conversion and redundant proof generation over the full recording. It also naturally supports a broad class of audio transformations. We implement PPAAS and benchmark it against a general-purpose zkSNARK baseline for the same edit relations. The results show that our capture-authentication scheme is practical for resource-constrained recording devices, and for a 2^{14}-sample removal from a 2^{19}-sample recording, our proof protocol achieves 398× faster proof generation and 160× faster verification, reduces prover peak memory by over 1600×, and produces a 45% smaller proof than the baseline.

Contents

1 Introduction **3**
- 1.1 Technical Challenges . . . 4
- 1.2 Our Contributions . . . 5
- 1.3 Technical Overview . . . 6
- 1.4 Related Work . . . 6
- 1.5 Paper Organization . . . 7

2 Preliminaries **7**
- 2.1 Notation and Audio Representation . . . 7
- 2.2 Cryptographic Tools . . . 8
- 2.3 Zero-Knowledge Proofs . . . 10

3 Privacy-Preserving Capture Authentication **11**
- 3.1 Privacy Limitations of Hash-then-Sign . . . 12
- 3.2 Capture-Authentication Construction . . . 12

4 Zero-Knowledge Proofs for Audio Editing **13**
- 4.1 A Unified Relation for Audio Editing . . . 13
- 4.2 Removing Source Entries Determined by the Release . . . 15
- 4.3 Proving the Audio-Editing Relation . . . 16
- 4.4 Security and Efficiency Analysis . . . 17

5 Privacy-Preserving Audio Authentication System **20**
- 5.1 Formal Definition . . . 20
- 5.2 Generic Construction . . . 22

6 Implementation and Evaluation **24**
- 6.1 Implementation Details . . . 24
- 6.2 Audio Editing Proof Results . . . 25
- 6.3 Comparing the Two Recording Schemes . . . 26

7 Conclusion **28**

1 Introduction

Audio recordings are routinely used as evidence in legal, journalistic, and investigative settings. At the same time, advances in voice cloning have made convincing audio forgeries widely accessible, weakening the evidentiary value of content that merely sounds authentic. The research community has documented this shift through dedicated surveys and benchmark efforts on synthetic and spoofed speech detection [KWJ23, YWT+21]. The threat is no longer hypothetical: AI-generated robocalls imitating U.S. President Joseph R. Biden, Jr. [Fed24] and fabricated election-related audio in Slovakia [Swe24] illustrate how synthetic speech can mislead the public. Such incidents make provenance, rather than plausibility alone, central to the authentication of digital audio.

For this reason, provenance for edited audio must establish both origin and alteration: the source must be authenticated at capture, and every change from that source to the released audio must be justified by a declared transformation. This is difficult because authentic recordings often cannot be released in full: they may contain victims' or witnesses' voices, private conversations, or other sensitive information that must be removed before disclosure. Body-worn-camera guidance, for example, recommends review and redaction when recordings raise privacy concerns [MT14]; investigative reporting and other sensitive releases create the same need to remove intervals, mask private speech, or apply fixed filters. These requirements expose the limits of conventional audio-authentication techniques, which typically detect tampering or verify signal integrity without proving the provenance of a privacy-preserving edit [RL18, PZL12]. A signature on the source does not authenticate the edited result, while releasing the signed source defeats redaction. Moreover, when source authentication relies on a device-specific public key, as in settings requiring hardware accountability, it may reveal the recorder even when the intended claim is only that the source came from an authorized device group. The challenge is therefore to preserve both capture provenance and edit provenance without exposing either the hidden recording or the identity of the recording device.

To meet these requirements, a conventional provenance record is not enough: the mechanism must protect the hidden source and recorder throughout public release even when the editor is untrusted for transformation correctness. Existing provenance mechanisms do not provide this guarantee. The Coalition for Content Provenance and Authenticity (C2PA) defines signed, tamper-evident manifests for images, video, audio, and documents [Coa25]. To support post-capture transformations, its trust chain may include editing applications that append and sign new provenance assertions; this preserves disclosed provenance but requires the editing environment to protect signing keys and behave correctly. Recent systems for edited images and videos instead use zero-knowledge proofs to show that a released output was derived from a hidden authenticated source, preserving source privacy without trusting the editor's execution [NT16, DCB25, MVVZ25, DEH25, ZYO+25, GMS26]. These systems show the promise of private media provenance, but they leave two obstacles for the audio workflow above.

The first obstacle is privacy leakage during capture. Several media-provenance systems authenticate a source by publishing a hash value and a conventional signature. The hash value then serves as the public reference for proving knowledge of the hidden source. When a release determines most of that source and the removed audio has a small candidate set, however, an adversary can enumerate candidates and test them against the public hash value. This is an offline guessing attack, not a collision attack. Replacing the hash value with a hiding commitment prevents this test, as also observed for predictable image tiles in TilesProofs [MVVZ25], but an ordinary signature still identifies the verification key of the recording device. The challenge is to bind authentication to a hidden source without leaving either an offline test for the source or an identifier for the recorder.

The second obstacle is proof overhead during editing. Existing media-provenance systems commonly decompose source authentication and transformation correctness into separate proofs, which requires an additional mechanism to bind both proofs to the same hidden source. Converting the transformation into a general arithmetic circuit also introduces substantial construction and proving costs; HyperVerITAS identifies the translation to R1CS as a major source of overhead and instead proves affine transformations directly [GMS26]. Even when the transformation is handled efficiently, the private witness generally contains the complete source. For a local audio edit, the proof therefore processes a long recording although the public release already determines most of its samples. The editor-time problem is to establish both source

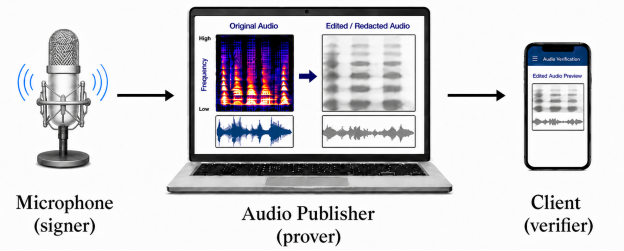

Figure 1: The three-party PPAAS workflow.

authentication and transformation correctness for the same hidden recording without paying for cross-proof binding, general-circuit conversion, or source values already determined by the release.

Taken together, these limitations leave a clear gap: existing approaches do not simultaneously hide the source and recorder while avoiding cross-proof binding, general-circuit conversion, and redundant work over source data already determined by the release. This gap motivates the central question of this work.

Can we efficiently verify both capture provenance and edit provenance while hiding the source recording and the recording device?

We answer affirmatively with PPAAS, a unified framework for capture authenticity, edit correctness, audio privacy, and device anonymity. A single legality relation binds both claims to one hidden source. At capture, a commitment-based ring signature replaces hash-then-sign and hides both the source and the recorder. At editing, the commitment opening and declared transformation are proved in one algebraic relation, avoiding separate binding and general-circuit conversion. The compiler also removes source entries determined by the public release, reducing source-dependent proof work from the complete recording to the private edited portion. The same reduction applies to vector-encoded media when the public release determines part of the authenticated source and its contribution is linear in both relations.

Figure 1 shows the three-party workflow considered in this paper. A provisioned recorder captures and authenticates a source recording. An editor receives the recording and its capture state, applies an allowed edit, and generates a proof for public release. A verifier receives only the edited audio, its public edit description, the capture object, and the proof, and checks both capture authenticity and edit correctness. We call this workflow a *Privacy-Preserving Audio Authentication System* (PPAAS).

The editor is entrusted with the source and capture opening but is not trusted for transformation correctness; verification is public. This matches deployments where editing environments may be manipulated and the root of trust remains in secure capture hardware. Similar assumptions are used by provenance frameworks such as VeriTAS [DCB25]. Before deployment, each recording device receives protected signing state, its verification key is certified, and it is assigned to a fixed authenticated ring; different device groups may use different rings, and anonymity holds only among members of the same ring. The recorder is trusted to canonicalize and authenticate the signal presented through its interface and to protect its signing state. Users may operate an authorized recorder and choose the presented audio, but cannot extract its signing state or make it authenticate an external file. Thus, an accepting publication proves that a device in the ring captured a source and that the declared edit maps that source to the release, while hiding unreleased source information and the particular device. As in VeriTAS [DCB25], the model begins after digitization: acoustic replay and recorder compromise are excluded, while the released audio, edit description, audio profile, and disclosed lengths are public.

Having established the problem setting and trust model, we next identify the key technical challenges, present our contributions, and give a high-level overview of PPAAS.

1.1 Technical Challenges

Formalizing privacy-preserving audio authentication. Capture authenticity, edit correctness, content privacy, and device anonymity are not compositional by default. Separate capture and editing mech-

anisms may refer to different sources, and stand-alone privacy definitions may omit leakage from the final publication. The first challenge is therefore to define the parties, public and hidden data, allowed edits, trust boundary, and accepting claim in one model, with both claims referring to the same hidden source.

Privacy-preserving capture authentication. Hash-then-sign exposes a predictable public hash and a device-specific verification key. Replacing only one component leaves the other leakage channel, while proving both in zero knowledge shifts unnecessary work to the editing proof. The second challenge is to authenticate the captured source while providing hiding, binding, unforgeability, and anonymity within an authorized ring.

Efficient proofs for authenticated editing. The editor must prove that one hidden source both opens the capture object and maps to the released audio. A direct construction must avoid duplicate source vectors, cross-proof equality checks, general-circuit compilation, and private processing of source entries already recoverable from the release. The third challenge is to remove these redundancies without weakening the common-source or soundness guarantees.

1.2 Our Contributions

We introduce and formally define PPAAS through four algorithms and five security requirements: completeness, capture authenticity, edit correctness, audio privacy, and device anonymity. A single legality relation binds capture authentication and edit correctness to the same hidden source, while the security model makes the public leakage and the anonymity set explicit. We give a generic construction that combines the capture and editing mechanisms and prove that it achieves all five requirements.

For capture authentication, we design a commit-then-ring-sign construction that eliminates both disclosure channels in hash-then-sign. A randomized commitment hides and binds the source recording, while a ring signature authenticates the commitment without identifying the signing member. The construction establishes content hiding and signer anonymity at capture without placing ordinary-signature verification and key-membership checks in the later editing proof.

For editing, we develop a zero-knowledge proof for audio transformations expressible as affine equations with range conditions. The protocol unifies the commitment-opening and transformation equations, proves their algebraic constraints directly, and thereby eliminates both a separate cross-proof binding mechanism and general-circuit conversion for the complete relation. By subtracting the source contribution determined by the public release, it removes the unedited part from both sets of equations. If D of the N encoded source entries remain private, the source-dependent private dimension is reduced from N to D. More generally, the same reduction applies to image and video transformations for which the public release and edit description determine a subset of source entries that occur linearly in both relations.

We implement a prototype and evaluate its capture and editing costs as the source size, edited portion, and operation vary.

To summarize, our contributions are threefold:

- PPAAS provides a unified formal framework for privacy-preserving audio authentication, with one legality relation and five security requirements for the complete release workflow;
- A commitment-based ring signature hides both the source recording and the recording device at capture, eliminating the two disclosure channels in hash-then-sign;
- A direct algebraic editing proof combines commitment opening and transformation correctness while removing source entries already determined by the public release.

Our techniques extend beyond audio to any setting where a released object and public transformation description determine part of a large authenticated witness, and the determined contribution is linear in the authentication and transformation relations. Examples include local image redaction, image cropping, and video-segment removal. However, our focus in this paper is on transformations applied to authenticated audio recordings.

1.3 Technical Overview

Commitment-based capture authentication. The recorder samples randomness r_{cap}, commits to the canonical source v, and signs the commitment using its provisioned ring. The public capture object contains its descriptor, commitment, and ring signature; the source and opening randomness are sent privately to the editor. Commitment hiding prevents offline source testing, binding prevents attaching another source to the capture object, and ring-signature anonymity hides the signing device.

A unified editing relation. Common audio operations such as interval removal, bounded-signal addition, and fixed-coefficient finite impulse response filters can be expressed as sparse affine equations with range conditions. We use a hiding and binding commitment whose opening equation is linear in the source and randomness. Because the opening and transformation equations share one source vector, they can be proved in one relation without duplicating the source or adding an equality argument.

Removing the unedited contribution. Given released audio y and public edit description F, a compiler partitions the source into entries recoverable from (F, y) and entries that remain private. It substitutes the recoverable entries into the transformation equations and moves their contribution to the public side. Linearity lets the verifier subtract the same contribution from the commitment-opening equation. The reduced relation contains only private source entries, commitment randomness, and private edit or auxiliary values. Thus, for D private entries in an N-entry source, private source-dependent work scales with D rather than N. The reduction depends on public recoverability and linearity, and extends to image cropping, local redaction, and video-segment removal when retained media and source positions are publicly recoverable.

Proving the reduced relation. We prove the reduced algebraic constraints directly instead of compiling them into a general circuit. Random linear combinations compress commitment-opening and transformation rows, and a final challenge combines them into one inner-product relation. One Inner Product Protocol proves the combined equality, while lookup arguments enforce the required byte, randomness, edit-value, and rounding ranges. The result avoids both a separate bridge and general-circuit representation of the complete relation.

System construction. PPAAS consists of Setup, Record, Prove, and Verify. Setup provisions device states, authenticated rings, the public audio profile, and cryptographic parameters. Record authenticates a source; Prove applies an allowed edit and generates a publication; Verify checks both claims from public data. The legality relation requires the same source to satisfy capture and transformation relations. The capture and editing mechanisms are developed independently in Sections 3 and 4, then combined in Section 5. The verifier checks the ring signature and the reduced editing proof against the same commitment. Binding and knowledge soundness connect the edit witness to a source authenticated by a provisioned device, while hiding, zero knowledge, and signer anonymity protect private publication data.

1.4 Related Work

Privacy-preserving authentication of edited media. PhotoProof [NT16] introduced cryptographic authentication under permissible image transformations, and ZK-IMG [KHSS22] studied attested images with zero-knowledge proofs. VerITAS [DCB25] combines a proof of a signed hash preimage with a proof of image transformation. TilesProofs [MVVZ25] divides an image into tiles and commits to each tile so that proof generation can be performed with smaller witnesses on commodity hardware. VIMz [DEH25] uses folding-based proofs for sequences of image transformations and proves knowledge of an authorized signer without revealing that signer. Eva [ZYO+25] extends privacy-preserving provenance to edited and lossily encoded video. HyperVerITAS [GMS26] uses multilinear polynomials to reduce prover memory and directly proves a broad class of affine transformations. Together, these systems establish the main line of privacy-preserving provenance for edited visual media, but their advances address different parts of the problem:

tiling reduces the size of individual image witnesses, folding supports sequences of transformations, and direct affine proofs reduce circuit overhead. They do not jointly hide both the recording and the signing member at capture, unify the commitment-opening and transformation constraints, and remove the publicly determined source contribution from both. PPAAS provides these properties in an audio-specific formal model.

Audio authentication and private audio sharing. Audio watermarking, perceptual hashing, and signal forensics provide complementary methods for detecting tampering and analyzing recording history [HHS+16, ÖSMA05, RL18, PZL12, HLW+21]. Overo [LKYL22] shows that privacy processing and integrity protection can be combined in a practical audio-sharing system. These techniques provide signal-level evidence or protected audio sharing, but do not give a public zero-knowledge proof that released audio was derived from a hidden authenticated source by a declared edit. PPAAS targets this verification claim while keeping the source recording, sensitive edit values, and recording device outside the public view.

1.5 Paper Organization

Section 2 introduces the cryptographic preliminaries, audio profile, and notation used throughout. Section 3 presents our privacy-preserving capture-authentication scheme. Section 4 develops the algebraic proof for audio editing, establishes its security, and explains why its private proof work scales with the edited region. Section 5 formally defines PPAAS and combines the two mechanisms in a generic construction. Section 6 presents the implementation and experimental evaluation. Section 7 concludes the paper.

2 Preliminaries

This section defines the notation and audio representation used throughout the paper, and then recalls the cryptographic primitives and proof protocols used by our constructions.

2.1 Notation and Audio Representation

Basic notation. We denote the security parameter by λ and use PPT to mean probabilistic polynomial time. A function $\epsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is negligible if, for every $c > 0$, there exists λ_0 such that $\epsilon(\lambda) < \lambda^{-c}$ for all $\lambda \geq \lambda_0$. We write $x \leftarrow \mathcal{A}(\cdot)$ for the output of an algorithm and $x \xleftarrow{R} S$ for an element sampled uniformly from a finite set S. For an integer $k \geq 0$, let $[k] = \{0, \ldots, k-1\}$; for $a \leq b$, let $[a : b] = \{a, a+1, \ldots, b\}$. The symbol $\perp$ denotes failure.

We use $\mathbb{F}_q$ for the prime field of order q and $\mathbb{K}$ for a field containing $\mathbb{F}_q$. Vectors are indexed from zero. For vectors u and v, $\langle u, v \rangle$ denotes their inner product and $u \parallel v$ their concatenation. We write I_n for the $n \times n$ identity matrix, $\mathbf{e}_i$ for its i-th standard basis vector, and $\mathbf{1}_n$ for the all-one vector of length n.

Following the standard relation notation for proofs of knowledge [BG93, AFK23], a relation R is a set of statement–witness pairs. In a tuple $(\mathsf{stmt}; \mathsf{wit})$, the values before the semicolon are public and those after it are private. We write $(\mathsf{stmt}; \mathsf{wit}) \in \mathsf{R}$ when the witness satisfies the relation for the given statement.

Audio profiles. An audio profile fixes the sampling rate, the number $c \geq 1$ and order of channels, signed PCM sample depth b, and the deterministic rules for decoding an accepted file. We consider $b > 8$ divisible by eight. After decoding, T sample frames are flattened in temporal order, and the entries within each frame follow the channel order fixed by the profile. This gives a sequence $x = (x_0, \ldots, x_{n-1})$, where $n = cT$ and x_{ct+j} is the entry of channel j in frame t, with

$$x_i \in \{-2^{b-1}, \ldots, 2^{b-1} - 1\}.$$

This signed representation, interleaved channel order, and least-significant byte first convention agree with the PCM/WAVE format specified in Recommendation ITU-R BS.1352-4 [Int23].

Canonical representation. The preceding conventions describe the output of the profile-dependent file decoder. The encoding below is a separate representation for cryptographic operations, rather than the byte string of the input container. Let $L = b/8$. For each signed sample x_i, define its offset value $\widehat{x}_i = x_i + 2^{b-1}$ and write it in little-endian byte order as

$$\widehat{x}_i = \sum_{j=0}^{L-1} 2^{8j} v_{Li+j}, \qquad v_{Li+j} \in \{0, \ldots, 255\}.$$

This gives a bijection $\mathsf{Encode} : x \mapsto v \in \{0, \ldots, 255\}^N$, where $N = Ln$; we write Decode for its inverse. The public deterministic procedure Canon first decodes an accepted file according to the audio profile and then applies Encode:

$$\mathsf{Canon}(\mathsf{file}) = \mathsf{Encode}(\mathsf{PCMDecode}(\mathsf{file})).$$

It returns $\perp$ if the file does not conform to the profile. Container metadata outside the decoded PCM sequence is not included in the vector authenticated by the cryptographic protocols. We identify v with its canonical embedding in $\mathbb{F}_q^N$, where $q > 255$. The field and public range bounds are chosen so that every equation over $\mathbb{F}_q$ is equivalent to the corresponding bounded integer equation.

Edit descriptions. The symbol F denotes a public edit description and θ the private edit values. The deterministic procedure $\mathsf{Apply}(F, v; \theta)$ decodes v, applies the operation described by F to the signed samples, and returns the released sample sequence y under the same public profile. Throughout the formal model, y denotes this canonical sample sequence; any released file is represented by the sequence obtained from the same deterministic profile decoder. The procedure Derive in Section 4.1 expresses this sample-level operation as affine conditions over the byte vector v.

2.2 Cryptographic Tools

We next recall the primitives used to authenticate the canonical recording and hide both its contents and the identity of the recording device.

Definition 2.1 (Hash-function family [RS04])**.** A hash-function family $\mathcal{H} = (\mathsf{KGen}_\mathsf{H}, \mathsf{Eval}_\mathsf{H})$ has a probabilistic algorithm $hk \leftarrow \mathsf{KGen}_\mathsf{H}(1^\lambda)$ and a deterministic algorithm $\mathsf{Eval}_\mathsf{H}(hk, m)$. We write $H_{hk}(m) = \mathsf{Eval}_\mathsf{H}(hk, m)$. The family is collision resistant if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ϵ_H such that

$$\Pr\left[\; m_0 \neq m_1 \wedge H_{hk}(m_0) = H_{hk}(m_1) : \begin{array}{l} hk \leftarrow \mathsf{KGen}_\mathsf{H}(1^\lambda); \\ (m_0, m_1) \leftarrow \mathcal{A}(hk) \end{array} \right] \leq \epsilon_\mathsf{H}(\lambda).$$

Throughout the paper, H denotes the public function H_{hk} for a key generated at security parameter λ. Collision resistance prevents an adversary from producing two colliding inputs, but does not make a public hash value a hiding representation of its input.

Definition 2.2 (Commitment scheme [Ped92])**.** A commitment scheme $\Pi_\mathsf{com} = (\mathsf{Setup}_\mathsf{com}, \mathsf{Commit}, \mathsf{Open})$ consists of three PPT algorithms. Setup generates $pp_\mathsf{com} \leftarrow \mathsf{Setup}_\mathsf{com}(1^\lambda)$. For a message m and randomness $r \in \mathcal{R}_\mathsf{com}$, $C \leftarrow \mathsf{Commit}(pp_\mathsf{com}, m; r)$ produces a commitment, and $\mathsf{Open}(pp_\mathsf{com}, C, m, r)$ outputs a bit.

Correctness requires every honestly generated commitment to open successfully. For hiding, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT adversary, where $\mathcal{A}_1$ outputs two messages in the same public message space. There exists a negligible function ϵ_hide such that

$$\left| \Pr\left[b' = b : \begin{array}{l} pp_\mathsf{com} \leftarrow \mathsf{Setup}_\mathsf{com}(1^\lambda); \\ (\mathsf{st}, m_0, m_1) \leftarrow \mathcal{A}_1(pp_\mathsf{com}); \\ b \xleftarrow{\mathrm{R}} \{0,1\};\ r \xleftarrow{\mathrm{R}} \mathcal{R}_\mathsf{com}; \\ C \leftarrow \mathsf{Commit}(pp_\mathsf{com}, m_b; r); \\ b' \leftarrow \mathcal{A}_2(\mathsf{st}, C) \end{array} \right] - \frac{1}{2} \right| \leq \epsilon_\mathsf{hide}(\lambda).$$

Binding requires that, for every PPT adversary $\mathcal{A}$, there exists a negligible function ϵ_{bind} such that

$$\Pr\left[\begin{array}{c} m_0 \neq m_1 \wedge \\ \mathsf{Open}(pp_{\mathsf{com}}, C, m_0, r_0) = 1 \wedge \\ \mathsf{Open}(pp_{\mathsf{com}}, C, m_1, r_1) = 1 \end{array} : \begin{array}{l} pp_{\mathsf{com}} \leftarrow \mathsf{Setup}_{\mathsf{com}}(1^\lambda); \\ (C, m_0, r_0, m_1, r_1) \leftarrow \mathcal{A}(pp_{\mathsf{com}}) \end{array}\right] \leq \epsilon_{\mathsf{bind}}(\lambda).$$

Definition 2.3 (Commitment with a linear opening relation [ESS+19])**.** For vector messages in $\mathcal{M}_{\mathsf{com}} \subseteq \mathbb{F}_q^N$, we say that Π_{com} has a linear opening relation over $\mathbb{F}_q$ if pp_{com} specifies public matrices $A_{\mathsf{com}} \in \mathbb{F}_q^{h \times N}$ and $B_{\mathsf{com}} \in \mathbb{F}_q^{h \times \mu}$ such that

$$\mathsf{Commit}(pp_{\mathsf{com}}, m; r) = A_{\mathsf{com}} m + B_{\mathsf{com}} r$$

for every $m \in \mathcal{M}_{\mathsf{com}}$ and $r \in \mathcal{R}_{\mathsf{com}} \subseteq \mathbb{F}_q^\mu$, and Open accepts exactly when the message and randomness belong to these sets and the equation holds. This linear form is an additional algebraic requirement; the scheme must still satisfy the correctness, hiding, and binding properties above.

Commitments of this form are known. After coefficient embedding over $\mathbb{F}_q$, for example, the module-lattice commitment of Esgin et al. has the form $G_r r + G_m m$, with computational binding based on Module-SIS and hiding based on Module-LWE [ESS+19].

Definition 2.4 (Digital signature [GMR88])**.** A digital signature scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ generates a key pair (vk, sk), signs a message with sk, and verifies the resulting signature with vk. Correctness requires honestly generated signatures to verify. Existential unforgeability under chosen-message attacks requires that a PPT adversary with access to a signing oracle cannot produce a valid signature on a message that it did not submit to that oracle.

Definition 2.5 (Ring signature [RST01])**.** A ring-signature scheme authenticates a message with respect to a ring of verification keys without identifying the signing member. It is a tuple $\Pi_{\mathsf{ring}} = (\mathsf{KeyGen}_{\mathsf{ring}}, \mathsf{Sign}_{\mathsf{ring}}, \mathsf{Verify}_{\mathsf{ring}})$ of three PPT algorithms. Key generation outputs $(vk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\mathsf{ring}}(1^\lambda)$. For an ordered ring $R = (vk_1, \ldots, vk_n)$ containing vk_i, $\mathsf{Sign}_{\mathsf{ring}}(sk_i, R, m)$ outputs a signature σ, and $\mathsf{Verify}_{\mathsf{ring}}(R, m, \sigma)$ outputs a bit. Correctness requires signatures generated by every member of R to verify.

The rings used in this work are fixed before signing queries and contain honestly generated, provisioned verification keys. We therefore use the basic-anonymity and fixed-ring-unforgeability notions of Bender, Katz, and Morselli, specialized to this setting [BKM06].

Definition 2.6 (Fixed-ring security)**.** For $n = \mathrm{poly}(\lambda)$, the challenger generates $(vk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\mathsf{ring}}(1^\lambda)$ for $i \in [1:n]$ and fixes $R = (vk_1, \ldots, vk_n)$. The signing oracle $\mathcal{O}_{\mathsf{sign}}(i, m)$ returns $\mathsf{Sign}_{\mathsf{ring}}(sk_i, R, m)$. In the unforgeability experiment, Q records every message submitted to this oracle.

The scheme is fixed-ring unforgeable if, for every PPT adversary $\mathcal{A}$, there exists a negligible function ϵ_{uf} such that

$$\Pr\left[\begin{array}{c} \mathsf{Verify}_{\mathsf{ring}}(R, m^*, \sigma^*) = 1 \wedge \\ m^* \notin Q \end{array} : \begin{array}{l} (vk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\mathsf{ring}}(1^\lambda),\ i \in [1:n]; \\ R \leftarrow (vk_1, \ldots, vk_n);\ Q \leftarrow \emptyset; \\ (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sign}}}(R) \end{array}\right] \leq \epsilon_{\mathsf{uf}}(\lambda).$$

The scheme has signer anonymity if, for every PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function ϵ_{anon} such that

$$\left|\Pr\left[b' = b : \begin{array}{l} (vk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\mathsf{ring}}(1^\lambda),\ i \in [1:n]; \\ R \leftarrow (vk_1, \ldots, vk_n); \\ (\mathsf{st}, i_0, i_1, m) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{sign}}}(R); \\ b \xleftarrow{\mathrm{R}} \{0,1\};\quad \sigma^* \leftarrow \mathsf{Sign}_{\mathsf{ring}}(sk_{i_b}, R, m); \\ b' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{sign}}}(\mathsf{st}, \sigma^*) \end{array}\right] - \frac{1}{2}\right| \leq \epsilon_{\mathsf{anon}}(\lambda),$$

where $i_0 \neq i_1$ and both indices belong to $[1:n]$.

Definition 2.7 (Multilinear extension [Set20])**.** For a function $f : \{0,1\}^\ell \to \mathbb{K}$, its multilinear extension is the unique multilinear polynomial $\widetilde{f}$ that agrees with f on $\{0,1\}^\ell$. For $u \in \{0,1\}^\ell$ and $r \in \mathbb{K}^\ell$, define

$$\mathrm{eq}(u, r) = \prod_{s=1}^{\ell} \big(u_s r_s + (1 - u_s)(1 - r_s)\big).$$

Then

$$\widetilde{f}(r) = \sum_{u \in \{0,1\}^\ell} f(u)\, \mathrm{eq}(u, r).$$

For Boolean r, $\mathrm{eq}(u, r)$ is one when $u = r$ and zero otherwise.

We use the following polynomial identity bound [Sch80, Zip79].

Lemma 2.1 (Schwartz–Zippel)**.** *Let $f \in \mathbb{F}[X_1, \ldots, X_\mu]$ be a nonzero polynomial of total degree at most d, and let S be a finite subset of $\mathbb{F}$. If $r_1, \ldots, r_\mu$ are sampled independently and uniformly from S, then*

$$\Pr[f(r_1, \ldots, r_\mu) = 0] \le \frac{d}{|S|}.$$

The following tests apply Lemma 2.1 to a univariate polynomial and to a multilinear extension, respectively.

Corollary 2.2 (Random-combination tests)**.** *Let $\delta = (\delta_0, \ldots, \delta_{m-1}) \in \mathbb{K}^m$ be nonzero. For $\xi \xleftarrow{\mathrm{R}} \mathbb{K}$,*

$$\Pr\left[\sum_{i=0}^{m-1} \delta_i \xi^i = 0\right] \le \frac{m-1}{|\mathbb{K}|}.$$

Let $\ell = \lceil \log_2 \max\{2, m\} \rceil$, pad δ with zeros to 2^ℓ entries, and let $\mathrm{bin}_\ell(i)$ be the ℓ-bit encoding of i. For $r \xleftarrow{\mathrm{R}} \mathbb{K}^\ell$,

$$\Pr\left[\sum_{i=0}^{m-1} \delta_i\, \mathrm{eq}(\mathrm{bin}_\ell(i), r) = 0\right] \le \frac{\ell}{|\mathbb{K}|}.$$

Proof. The first sum is a nonzero univariate polynomial of degree at most $m-1$. The second is the multilinear extension of the padded nonzero table and has total degree at most ℓ. Both bounds follow from Lemma 2.1. □

Section 4 applies the univariate test to the rows of the commitment-opening equation and the multilinear test to the rows of the transformation equation.

2.3 Zero-Knowledge Proofs

The editing construction uses public-coin zero-knowledge arguments for linear equations and range conditions. We first recall interactive arguments of knowledge and then define the polynomial-oracle relations used in Section 4.

Definition 2.8 (Interactive argument of knowledge [BCG+17])**.** Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be a public-coin proof system for a polynomial-time relation R, where $\mathcal{P}$ and $\mathcal{V}$ share a statement stmt and $\mathcal{P}$ additionally holds a witness wit. The verifier's messages are sampled uniformly and independently of the prover's messages.

The proof system has *perfect completeness* if, for every PPT algorithm $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} (pp, \mathsf{stmt}, \mathsf{wit}) \notin \mathsf{R} \vee \\ \langle \mathcal{P}(pp, \mathsf{stmt}, \mathsf{wit}), \mathcal{V}(pp, \mathsf{stmt}) \rangle = 1 \end{array} : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(pp) \end{array}\right] = 1.$$

It has *computational knowledge soundness* if, for every deterministic polynomial-time prover $\mathcal{P}^*$, there is an expected polynomial-time extractor $\mathcal{E}$ such that, for every PPT algorithm $\mathcal{A}$,

$$\Pr\left[\; b = 1 \;\wedge\; (pp, \mathsf{stmt}, \mathsf{wit}) \notin \mathsf{R} : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ (\mathsf{stmt}, \mathsf{st}) \leftarrow \mathcal{A}(pp); \\ \mathsf{wit} \leftarrow \mathcal{E}^{\langle \mathcal{P}^*(\mathsf{st}), \mathcal{V}(pp, \mathsf{stmt}) \rangle}(pp, \mathsf{stmt}) \end{array} \right] \leq \mathsf{negl}(\lambda).$$

The extractor's oracle runs a complete protocol execution, returns its transcript, and permits rewinding with fresh public-coin challenges; b is the verifier's output in the first such execution. A perfectly complete proof system with computational knowledge soundness is an interactive argument of knowledge.

Following Goldwasser–Micali–Rackoff [GMR85], the protocol is computational zero knowledge if, for every PPT verifier $\mathcal{V}^*$, there is a PPT simulator $\mathcal{S}$ whose output is computationally indistinguishable from the view of $\mathcal{V}^*$ in a real execution. Honest-verifier zero knowledge (HVZK) restricts this requirement to the honest verifier. In special honest-verifier zero knowledge (SHVZK), the simulator receives the verifier's public coins and produces a transcript without the witness [BCG+17].

The Fiat–Shamir transform replaces public-coin verifier challenges with hashes of the preceding transcript [FS87, BR93]. For multi-round protocols, its knowledge guarantees depend on protocol-specific conditions and can incur a round-dependent loss [AFK23]. Interactive oracle proofs (IOPs), introduced by Ben-Sasson, Chiesa, and Spooner, allow the verifier oracle access to the prover's messages rather than requiring it to read those messages in full [BSCS16].

Definition 2.9 (Polynomial interactive oracle proof [CBBZ23])**.** A polynomial interactive oracle proof (PIOP) is a public-coin interactive proof for a polynomial oracle relation whose tuples are written $(\mathsf{i}, \mathsf{x}; \mathsf{w})$. The index i and instance x may contain oracles to multivariate polynomials, with the corresponding polynomials contained in the public parameters and witness, respectively. Each prover message contains multivariate polynomial oracles, and each verifier message is a random challenge. We write $[\![f]\!]$ for an oracle to a polynomial f.

Definition 2.10 (Lookup relation [CBBZ23])**.** Let $\mathbb{B} = \{0,1\}$, let $t = (t_1, \ldots, t_{2^\mu - 1}) \in \mathbb{F}^{2^\mu - 1}$ be a public lookup table, and let $f \in \mathbb{F}[X_1, \ldots, X_\mu]$ be multilinear. The lookup relation is

$$\mathsf{R}_{\mathsf{lk}} = \left\{ \; \big((t, [\![f]\!]); (f, \mathsf{addr})\big) : \begin{array}{l} \mathsf{addr} : \mathbb{B}^\mu \to [1 : 2^\mu - 1], \\ f(x) = t_{\mathsf{addr}(x)} \text{ for every } x \in \mathbb{B}^\mu \end{array} \right\}.$$

HyperPlonk gives a PIOP for this relation [CBBZ23].

Inner-product arguments establish a claimed inner product of committed vectors; Bootle et al. give a logarithmic-communication argument of knowledge for this task [BCC+16]. In polynomial-oracle form, the vector entries are the evaluations of multilinear polynomials over the Boolean hypercube.

Definition 2.11 (Inner-product relation [LFKN92, Set20])**.** Let $f, g \in \mathbb{F}[X_1, \ldots, X_\mu]$ be multilinear. The inner-product relation is

$$\mathsf{R}_{\mathsf{ip}} = \left\{ \big(([\![f]\!], g, c); f\big) : c = \sum_{x \in \mathbb{B}^\mu} f(x) g(x) \right\}.$$

This relation is proved by applying sum-check to the degree-two polynomial $f \cdot g$ [LFKN92, Set20].

3 Privacy-Preserving Capture Authentication

Let x denote a canonical recording. This section develops a privacy-preserving capture-authentication scheme that prevents public verifiers from testing unreleased source content or identifying which recorder in an authorized device group produced the capture. We first identify why hash-then-sign does not provide these guarantees, then present and analyze our construction.

3.1 Privacy Limitations of Hash-then-Sign

A common capture procedure first computes a hash value of the recording and then signs the resulting value. In a simplified form, an enrolled device with signing key sk_i computes

$$h = H(x), \qquad \sigma = \mathsf{Sign}(sk_i, h), \tag{1}$$

where H is a public hash function. The verifier checks the signature under the device's public key vk_i. This mechanism is used by several privacy-preserving image-provenance constructions [DCB25, GMS26]. There, h remains available as the public reference for proving that a hidden source both hashes to h and produces the released media. The source need not be published, but its hash becomes part of the public authentication data.

This exposure creates the first privacy limitation: a public hash value does not by itself provide hiding. For clarity, write the recording as $x = x_U \parallel x_E$, where the public release determines x_U and leaves a small part x_E unpublished. An adversary can enumerate candidates u for x_E and test

$$H(x_U \parallel u) = h.$$

For a small or predictable candidate set, this becomes an offline test for the hidden content. The attack does not find a collision in H; it uses the public value to test candidate openings, so collision resistance does not imply hiding of the recording.

The ordinary signature creates a second privacy limitation. Verification under vk_i identifies the key that produced the authentication object, and an enrolled key or certificate can therefore reveal the recording device. VIMz instead hides the signature and verification key and proves their validity and set membership inside a zkSNARK [DEH25]. This hides the signer, but adds both checks to the zero-knowledge relation and increases proof generation and verification costs.

Changing either component alone is therefore insufficient. A privacy-preserving capture-authentication scheme must hide unreleased source information while authenticating the recording without identifying the recorder within the authorized group.

3.2 Capture-Authentication Construction

The preceding analysis shows that both leakage channels should be eliminated by the capture mechanism itself, rather than compensated for in the later zero-knowledge proof. Our key design is to make a randomized commitment serve simultaneously as a hiding reference to the captured source and as the value authenticated with respect to an authorized device group. We authenticate this commitment with a ring signature. The same commitment later anchors the editing proof to the authenticated source, so capture authentication and edit correctness share one source reference without placing ordinary-signature verification or device-membership checks inside that proof. This commitment-based ring-signature construction is the capture-authentication primitive used by PPAAS.

Let $\Pi_{\mathsf{com}} = (\mathsf{Setup}_{\mathsf{com}}, \mathsf{Commit}, \mathsf{Open})$ be a correct, hiding, and binding commitment scheme, where Open verifies whether (x, r) opens C. Fix public parameters $pp_{\mathsf{com}} \leftarrow \mathsf{Setup}_{\mathsf{com}}(1^\lambda)$, and write $\mathcal{R}_{\mathsf{com}}$ for its randomness space. Let $\Pi_{\mathsf{ring}} = (\mathsf{KeyGen}_{\mathsf{ring}}, \mathsf{Sign}_{\mathsf{ring}}, \mathsf{Verify}_{\mathsf{ring}})$ be a ring signature scheme. A ring is an ordered list $R = (vk_1, \ldots, vk_n)$, and the holder of sk_i may sign a message with respect to R without revealing i. We assume the standard correctness, unforgeability, and signer anonymity properties of Π_{ring}. A canonical public capture descriptor d fixes the commitment-parameter identifier, authorized-ring snapshot, and audio profile, including the encoded length $N(d)$. Write $pp_{\mathsf{com}}[d]$ for the corresponding fixed-length commitment parameters and $\mathcal{M}_d$ for their message space. The canonical encoding of each admissible pair (d, C) belongs to the message space of Π_{ring}. The ring signature authenticates the canonical encoding of (d, C). For compactness, write $\mathsf{Com}_d(x; r) = \mathsf{Commit}(pp_{\mathsf{com}}[d], x; r)$ and use Open_d analogously. The signing algorithm returns $\perp$ when $x \notin \mathcal{M}_d$.

Using these components, we define the commitment-based ring-signature scheme

$$\Pi_{\mathsf{cap}} = (\mathsf{KeyGen}_{\mathsf{cap}}, \mathsf{Sign}_{\mathsf{cap}}, \mathsf{Verify}_{\mathsf{cap}}), \qquad \mathsf{KeyGen}_{\mathsf{cap}} := \mathsf{KeyGen}_{\mathsf{ring}}.$$

The signing and verification algorithms are shown in Figure 2.

$\mathsf{Sign}_{\mathsf{cap}}(sk_i, R, d, x)$	$\mathsf{Verify}_{\mathsf{cap}}(R, x, \Sigma)$
1. $r \xleftarrow{R} \mathcal{R}_{\mathsf{com}}$.	1. Parse $\Sigma = (d, C, r, \sigma)$.
2. $C \leftarrow \mathsf{Com}_d(x; r)$.	2. $b_1 \leftarrow [x \in \mathcal{M}_d] \wedge \mathsf{Open}_d(C, x, r)$.
3. $\sigma \leftarrow \mathsf{Sign}_{\mathsf{ring}}(sk_i, R, (d, C))$.	3. $b_2 \leftarrow \mathsf{Verify}_{\mathsf{ring}}(R, (d, C), \sigma)$.
4. $\Sigma \leftarrow (d, C, r, \sigma)$.	4. return $b_1 \wedge b_2$.

Figure 2: Descriptor-bound commitment-based ring-signature scheme.

For formal analysis, $\Sigma = (d, C, r, \sigma)$ is the complete output of the signing algorithm. PPAAS discloses only (d, C, σ) and keeps (x, r) private from public verifiers.

Theorem 3.1 (Authentication properties of the capture scheme)**.** *If* Π_{com} *satisfies correctness and binding, and* Π_{ring} *satisfies correctness, unforgeability, and signer anonymity, then* Π_{cap} *satisfies correctness, unforgeability, and signer anonymity for every fixed ring* R *over the admissible descriptors.*

Proof. Correctness follows directly from the correctness of the commitment and ring signature schemes. An honestly generated commitment under d opens to x, and the ring signature on (d, C) verifies.

For unforgeability, fix R and consider an accepting signature for (d^*, x^*) that was not previously submitted to the signing oracle. Let $\Sigma^* = (d^*, C^*, r^*, \sigma^*)$. If (d^*, C^*) was not signed in answering a previous query, then σ^* is a forgery against Π_{ring}. Otherwise, a previous signing query (d', x') produced the same signed message, so $d' = d^*$ and its commitment is C^*. Since (d^*, x^*) was not submitted to the signing oracle, $x' \neq x^*$. The opening returned for x' and the accepting opening (x^*, r^*) are therefore two openings under the same fixed-length parameters to different messages, contradicting the binding of Π_{com}.

For signer anonymity, consider two members of the same ring signing the same (d, x). In the reduction, sample r and compute C independently of the challenge signer. The tuple (d, C, r) is therefore identical in the two experiments and independent of the signer. A distinguisher for the resulting signatures consequently distinguishes the two members signing the same (d, C), contradicting signer anonymity of Π_{ring}. □

The public capture object eliminates both disclosures identified above. Commitment hiding makes commitments to candidate recordings consistent with the release computationally indistinguishable, so the public tuple (d, C, σ) cannot serve as an offline candidate test. This guarantee requires r to remain private; otherwise, the opening procedure again enables such tests. Signer anonymity hides which member of R, and hence which recorder in the authorized group, generated the signature on (d, C), without adding ordinary-signature verification and key membership to a separate zero-knowledge proof. Theorem 3.1 establishes that the resulting capture scheme retains correctness, unforgeability, and signer anonymity.

4 Zero-Knowledge Proofs for Audio Editing

Given an encoded source v, its capture commitment C_{cap}, and a publication (y, F), the editor proves that the same v both opens C_{cap} and maps to y under the declared edit F. We first unify these conditions and then remove source entries determined by the publication.

4.1 A Unified Relation for Audio Editing

Audio representation. Let $x = (x_0, \ldots, x_{n-1})$ be a sequence of single-channel signed PCM samples. For sample depth b divisible by eight, let $L = b/8$ and encode

$$\widehat{x}_i = x_i + 2^{b-1} = \sum_{j=0}^{L-1} 2^{8j} v_{Li+j}, \qquad v_{Li+j} \in \{0, \ldots, 255\}.$$

This yields $v \in \mathbb{F}_q^N$, where $N = Ln$; edit intervals align with sample boundaries. Substituting this encoding into sample-domain edits gives affine conditions over $\mathbb{F}_q$, so the capture commitment and edit relation use the same v.

Affine audio transformations. This representation captures audio transformations expressible as affine relations with range conditions, including sample selection and removal, bounded additions, and fixed-coefficient finite impulse response filters. Following the general affine representation used for media transformations [GMS26], we use one relation for this class: F determines the public coefficients and index structure, while source entries not determined by the release and private edit values remain hidden. We evaluate interval removal, bounded additive noise, and fixed three-tap equalization; their matrix forms are as follows.

Let $x \in \mathbb{Z}^n$ be the input and let $I = [a, a + d)$ be the edited interval. Let $S_I \in \{0,1\}^{d\times n}$ be the interval-selection matrix, so that $(S_I x)_j = x_{a+j}$, let I_n be the $n \times n$ identity matrix, and let $\mathbf{1}_d$ be the all-one vector.

Interval removal. Let $R_I \in \{0,1\}^{(n-d)\times n}$ be the matrix obtained by retaining the rows of the identity whose indices are outside I:

$$R_I = \begin{bmatrix} \mathbf{e}_0 & \cdots & \mathbf{e}_{a-1} & \mathbf{e}_{a+d} & \cdots & \mathbf{e}_{n-1} \end{bmatrix}^{\mathsf{T}}, \qquad y = R_I x,$$

where $\mathbf{e}_i$ is the i-th standard basis vector. The corresponding block is omitted when the removed interval begins with the first sample or ends with the last sample; any block with an empty index range is interpreted as having no columns. The samples in I are absent from the output, while the remaining samples retain their original order.

Bounded additive noise. The output has the same length as the input and satisfies

$$y = x + S_I^{\mathsf{T}}\eta, \qquad \eta \in \{-B, \ldots, B\}^d,$$

where the bound B is public. For $\bar{\eta} = \eta + B\mathbf{1}_d$, the equations on the edited interval become

$$S_I x + \bar{\eta} = S_I y + B\mathbf{1}_d, \qquad \bar{\eta} \in \{0, \ldots, 2B\}^d.$$

The PCM range conditions reject additions that clip or saturate a sample.

Fixed three-tap equalization. Let $H \in \mathbb{Z}^{n\times n}$ be the causal filter matrix

$$H_{i,j} = \begin{cases} 1, & j = i \text{ or } j = i - 2, \\ 2, & j = i - 1, \\ 0, & \text{otherwise.} \end{cases}$$

Indices before the first sample are assigned value zero. Samples outside I are unchanged, and the edited samples satisfy

$$(\mathrm{I}_n - S_I^{\mathsf{T}} S_I)y = (\mathrm{I}_n - S_I^{\mathsf{T}} S_I)x,$$
$$S_I H x + e - 4S_I y = \mathbf{1}_d, \qquad e \in \{0,1,2,3\}^d.$$

Writing $s_i = (Hx)_i$, the second equality is equivalent to $y_i = \lfloor (s_i + 2)/4 \rfloor$ for $i \in I$; the vector e records the rounding value. All filter outputs are computed from the same input vector x.

The matrices R_I, S_I, and H are sparse: the selection matrices have one nonzero entry per row, and each row of H has at most three. For the two length-preserving operations, the first equation enforces that samples outside I remain unchanged. The three examples therefore share the same affine form despite their different audio semantics. Under the canonical byte encoding, direct substitution of these matrix forms gives

the corresponding Apply semantics. In each case, the released audio determines the source entries outside the edited interval, while the edited interval contains the source entries retained in the reduced witness. Thus these instantiations satisfy the semantic and reconstruction requirements imposed on Derive below.

On input (F, y), a deterministic public procedure Derive either rejects or returns the public coefficients, range information, and data required by the reduction below. With private edit and auxiliary values $z = (\theta, z_{\mathsf{aux}})$, the returned matrices define

$$G_F v + H_F z = b_F.$$

Here $G_F \in \mathbb{F}_q^{m \times N}$, $H_F \in \mathbb{F}_q^{m \times |z|}$, and $b_F \in \mathbb{F}_q^m$ are public. The procedure also fixes the allowed set $\mathcal{Z}_F$; the entries of v remain byte-valued.

Definition 4.1 (Audio transformation relation)**.** The audio transformation relation R_{tr} is defined as

$$\mathsf{R}_{\mathsf{tr}} = \left\{ \big((y, F); (v, z)\big) : \begin{array}{l} \mathsf{Derive}(F, y) \neq \bot, \\ G_F v + H_F z = b_F, \\ v \in \{0, \ldots, 255\}^N, \\ z \in \mathcal{Z}_F \end{array} \right\}.$$

For every supported, well-formed F, canonical source v, and admissible private edit value θ, the procedure accepts $(F, \mathsf{Apply}(F, v; \theta))$. Moreover, whenever $\mathsf{Derive}(F, y) \neq \bot$,

$$\mathsf{Apply}(F, v; \theta) = y \quad \Longleftrightarrow \quad \exists z_{\mathsf{aux}} : \big((y, F); (v, (\theta, z_{\mathsf{aux}}))\big) \in \mathsf{R}_{\mathsf{tr}}.$$

R_{tr} establishes the declared edit but does not bind its private source to the authenticated capture. We therefore use the hiding and binding commitments with a linear opening relation defined in Definition 2.3:

$$C_{\mathsf{cap}} = A_{\mathsf{cap}} v + B_{\mathsf{cap}} r_{\mathsf{cap}}, \tag{2}$$

where $A_{\mathsf{cap}} \in \mathbb{F}_q^{h \times N}$ and $B_{\mathsf{cap}} \in \mathbb{F}_q^{h \times \mu}$ are public, $C_{\mathsf{cap}} \in \mathbb{F}_q^h$, and $r_{\mathsf{cap}} \in \mathcal{R}_{\mathsf{cap}} \subseteq \mathbb{F}_q^\mu$. Including any additional commitment description pp_{com}, write $pp_{\mathsf{cap}} = (pp_{\mathsf{com}}, A_{\mathsf{cap}}, B_{\mathsf{cap}}, \mathcal{R}_{\mathsf{cap}})$.

Definition 4.2 (Audio-editing relation)**.** The audio-editing relation R_{AE} is defined as

$$\mathsf{R}_{\mathsf{AE}} = \left\{ \big((pp_{\mathsf{cap}}, C_{\mathsf{cap}}, y, F); (v, r_{\mathsf{cap}}, z)\big) : \begin{array}{l} C_{\mathsf{cap}} = A_{\mathsf{cap}} v + B_{\mathsf{cap}} r_{\mathsf{cap}}, \\ r_{\mathsf{cap}} \in \mathcal{R}_{\mathsf{cap}}, \\ \big((y, F); (v, z)\big) \in \mathsf{R}_{\mathsf{tr}} \end{array} \right\}.$$

By sharing v, the relation links capture and editing without a separate equality proof between independently committed source witnesses.

4.2 Removing Source Entries Determined by the Release

Although the unified relation links capture and editing, it treats the complete v as private. For local edits, the publication often determines the unchanged entries, which can therefore be omitted from both parts of the relation.

For an edit description F and published audio y, Derive identifies disjoint sets U_F and E_F such that

$$[N] = U_F \dot{\cup} E_F, \; v_U = \mathsf{Recover}(F, y), \; v = P_U v_U + P_E v_E. \tag{3}$$

Here $P_U \in \{0,1\}^{N \times |U_F|}$ and $P_E \in \{0,1\}^{N \times |E_F|}$ are public insertion matrices, with P_E ordered by source position. The set U_F contains the entries reconstructed by Derive, while E_F contains the remaining entries, including any additional context retained for the affine conditions. If $U_F = \emptyset$, the relation still applies without a reduction. Correctness of Derive requires every (v, z) satisfying R_{tr} for (y, F) to agree with the recovered entries: $P_U^{\mathsf{T}} v = v_U$.

Substituting (3) into the affine relation gives

$$G_F P_E v_E + H_F z = b_F - G_F P_U v_U.$$

Conditions containing no private value are checked directly, and the procedure rejects if any fails. The remaining affine conditions are

$$\widetilde{G}_F v_E + \widetilde{H}_F z = \widetilde{b}_F.$$

The procedure also rejects invalid recovered bytes. For every v_E, z, these conditions hold if and only if the complete affine relation holds after substituting $v = P_U v_U + P_E v_E$.

The partition similarly reduces the capture-opening equation. Let

$$A_U = A_{\mathsf{cap}} P_U, \qquad A_E = A_{\mathsf{cap}} P_E. \tag{4}$$

Then (2) becomes

$$C_{\mathsf{cap}} - A_U v_U = A_E v_E + B_{\mathsf{cap}} r_{\mathsf{cap}}. \tag{5}$$

The editor proves this equation together with the reduced affine and range conditions above; all derived quantities are computed deterministically from $(pp_{\mathsf{cap}}, C_{\mathsf{cap}}, F, y)$.

Definition 4.3 (Reduced audio-editing relation)**.** The reduced audio-editing relation $\mathsf{R}^{\mathsf{red}}_{\mathsf{AE}}$ is defined as

$$\mathsf{R}^{\mathsf{red}}_{\mathsf{AE}} = \left\{ \big((pp_{\mathsf{cap}}, C_{\mathsf{cap}}, y, F); (v_E, r_{\mathsf{cap}}, z)\big) : \begin{array}{l} \mathsf{Derive}(F, y) \neq \bot, \\ C_{\mathsf{cap}} - A_U v_U = A_E v_E + B_{\mathsf{cap}} r_{\mathsf{cap}}, \\ \widetilde{G}_F v_E + \widetilde{H}_F z = \widetilde{b}_F, \quad z \in \mathcal{Z}_F \\ v_E \in \{0, \ldots, 255\}^{|E_F|}, \quad r_{\mathsf{cap}} \in \mathcal{R}_{\mathsf{cap}} \end{array} \right\}. \tag{6}$$

The common partition preserves both components of the relation simultaneously.

Lemma 4.1 (Relation equivalence)**.** *Assume that* $\mathsf{Derive}(F, y) \neq \bot$*, and use the corresponding public output. For every* v_E*,* r_{cap}*, and* z*, let* $v = P_U v_U + P_E v_E$*. Then*

$$\begin{aligned} &\big((pp_{\mathsf{cap}}, C_{\mathsf{cap}}, y, F); (v, r_{\mathsf{cap}}, z)\big) \in \mathsf{R}_{\mathsf{AE}} \\ &\iff \big((pp_{\mathsf{cap}}, C_{\mathsf{cap}}, y, F); (v_E, r_{\mathsf{cap}}, z)\big) \in \mathsf{R}^{\mathsf{red}}_{\mathsf{AE}}. \end{aligned} \tag{7}$$

Proof. Using (4), the commitment equation in R_{AE} is

$$C_{\mathsf{cap}} = A_U v_U + A_E v_E + B_{\mathsf{cap}} r_{\mathsf{cap}}.$$

Subtracting $A_U v_U$ gives (5). Correctness of Derive guarantees that every complete witness agrees with v_U on U_F, so it has the unique decomposition $v = P_U v_U + P_E v_E$ with $v_E = P_E^{\mathsf{T}} v$. Under this substitution, the retained affine conditions are equivalent to the complete relation, while every condition containing no private value has already been checked publicly. Derive also checks the byte range of v_U, so the byte range of v is equivalent to that of v_E together with this public check. The conditions on z and r_{cap} are unchanged. Adding $A_U v_U$ and reversing these steps proves the converse. □

Every complete witness maps to $v_E = P_E^{\mathsf{T}} v$, and an extracted reduced witness reconstructs $v = P_U v_U + P_E v_E$.

4.3 Proving the Audio-Editing Relation

Apart from coordinate-wise range conditions, the reduced relation is linear, allowing a direct proof without general-purpose circuit conversion. We compress its two linear checks to scalar equalities, combine them into one inner product, and enforce the ranges through lookup protocols.

Let $\mathbb{K} = \mathbb{F}_{q^{d_\mathbb{K}}}$ be a finite extension of $\mathbb{F}_q$ with a fixed embedding, and let pp_{proof} parameterize the hiding, binding commitments shared by the Inner Product and Lookup Protocols of Section 2.3, all defined over $\mathbb{K}$. For readability, we leave the commitment randomness implicit in the notation.

For each supported operation, let $\mathcal{T}_{F,z}$ and $\mathcal{T}_r$ be public finite tables such that $\mathcal{Z}_F = \mathcal{T}_{F,z}^{|z|}$ and $\mathcal{R}_{\mathsf{cap}} = \mathcal{T}_r^{\mu}$; cross-coordinate conditions remain in the affine relation. Each nonempty block has a public logical length fixed by the statement and is canonically padded with a fixed table member; its inner-product coefficients are padded with zeros. Lookup covers the complete padded block, while empty blocks are omitted. The commitment and lookup steps below apply only to nonempty blocks.

Commitment-opening check.

Set $C_E := C_{\mathsf{cap}} - A_U v_U$. The verifier samples $\xi \xleftarrow{\mathrm{R}} \mathbb{K}$, sets $\rho = (1, \xi, \ldots, \xi^{h-1})$, and computes

$$g_A = A_E^{\mathsf{T}}\rho, \qquad g_B = B_{\mathsf{cap}}^{\mathsf{T}}\rho, \qquad d_{\mathsf{open}} = \langle \rho, C_E \rangle.$$

The compressed opening check is

$$\Delta_{\mathsf{open}} = \langle g_A, v_E \rangle + \langle g_B, r_{\mathsf{cap}} \rangle - d_{\mathsf{open}} = 0.$$

Affine transformation check. Let m_F be the number of reduced transformation rows and let $\ell_F = \lceil \log_2 \max\{2, m_F\} \rceil$. The extension degree $d_\mathbb{K}$ is chosen so that $(h + \ell_F)/|\mathbb{K}|$ is negligible in the security parameter for every supported relation. Using the multilinear-extension equality polynomial from Definition 2.7, pad the row domain to 2^{ℓ_F} positions and, for $u \in \{0,1\}^{\ell_F}$ and $r \in \mathbb{K}^{\ell_F}$, define

$$\mathrm{eq}(u, r) = \prod_{j=1}^{\ell_F} \big(u_j r_j + (1 - u_j)(1 - r_j)\big).$$

For $r_{\mathsf{row}} \xleftarrow{\mathrm{R}} \mathbb{K}^{\ell_F}$, set $\omega = (\omega_0, \ldots, \omega_{m_F - 1})$, with $\omega_i = \mathrm{eq}(\mathrm{bin}(i), r_{\mathsf{row}})$ where $\mathrm{bin}(i)$ is the ℓ_F-bit encoding of i, and compute

$$g_v = \widetilde{G}_F^{\mathsf{T}}\omega, \qquad g_z = \widetilde{H}_F^{\mathsf{T}}\omega, \qquad d_{\mathsf{tr}} = \langle \omega, \widetilde{b}_F \rangle.$$

The compressed affine check is

$$\Delta_{\mathsf{tr}} = \langle g_v, v_E \rangle + \langle g_z, z \rangle - d_{\mathsf{tr}} = 0.$$

The weighted sum is the multilinear extension of the zero-padded row-discrepancy vector evaluated at r_{row} and has degree at most ℓ_F.

Batched inner-product claim. For $\lambda_{\mathsf{open}}, \lambda_{\mathsf{tr}} \xleftarrow{\mathrm{R}} \mathbb{K}$, combine the two equalities as

$$\lambda_{\mathsf{open}}\Delta_{\mathsf{open}} + \lambda_{\mathsf{tr}}\Delta_{\mathsf{tr}} = 0. \tag{8}$$

With

$$w = (v_E \,\|\, r_{\mathsf{cap}} \,\|\, z),$$
$$g = (\lambda_{\mathsf{open}} g_A + \lambda_{\mathsf{tr}} g_v \,\|\, \lambda_{\mathsf{open}} g_B \,\|\, \lambda_{\mathsf{tr}} g_z),$$
$$d = \lambda_{\mathsf{open}} d_{\mathsf{open}} + \lambda_{\mathsf{tr}} d_{\mathsf{tr}},$$

this is $\langle g, w \rangle = d$, proved by one Inner Product Protocol; the same commitments support the range checks. Figure 3 summarizes the protocol.

4.4 Security and Efficiency Analysis

We first bound the soundness loss introduced by batching the two linear checks into a single inner-product claim.

Algebraic Editing Protocol
$(pp_{\mathsf{proof}}, pp_{\mathsf{cap}}, C_{\mathsf{cap}}, y, F); (v_E, r_{\mathsf{cap}}, z)$

1: $\mathcal{V}$ runs $\mathsf{Derive}(F, y)$, rejects if it returns $\perp$, and obtains v_U, P_U, P_E, and the reduced relation. It computes A_U, A_E by (4) and $C_E = C_{\mathsf{cap}} - A_U v_U$.
2: $\mathcal{P}$ calculates $\mathsf{com}_v \leftarrow \mathsf{Commit}_{\mathsf{proof}}(pp_{\mathsf{proof}}, v_E)$, $\mathsf{com}_z \leftarrow \mathsf{Commit}_{\mathsf{proof}}(pp_{\mathsf{proof}}, z)$, and $\mathsf{com}_r \leftarrow \mathsf{Commit}_{\mathsf{proof}}(pp_{\mathsf{proof}}, r_{\mathsf{cap}})$.
3: $\mathcal{V}$ samples $\xi \xleftarrow{\mathrm{R}} \mathbb{K}$, $r_{\mathsf{row}} \xleftarrow{\mathrm{R}} \mathbb{K}^{\ell_F}$, and $\lambda_{\mathsf{open}}, \lambda_{\mathsf{tr}} \xleftarrow{\mathrm{R}} \mathbb{K}$, and sends them to $\mathcal{P}$.
4: $\mathcal{P}$ and $\mathcal{V}$ compute ρ, ω, g_A, g_B, g_v, g_z, d_{open}, and d_{tr}, and then form the public g and d of (8).
5: $\mathcal{P}$ and $\mathcal{V}$ engage in one Inner Product Protocol for (8), using com_v, com_r, and com_z.
6: $\mathcal{P}$ and $\mathcal{V}$ engage in Lookup Protocols for $v_E \in \{0, \ldots, 255\}^{|E_F|}$, $z \in \mathcal{Z}_F$, and $r_{\mathsf{cap}} \in \mathcal{R}_{\mathsf{cap}}$, using the same commitments.
7: $\mathcal{V}$ accepts if all the arguments accept.

Figure 3: Protocol for the reduced audio-editing relation.

Lemma 4.2 (Soundness of constraint batching)**.** *Assume that the committed witness vectors are fixed before ξ, r_{row}, λ_{open}, and λ_{tr} are sampled. If the commitment-opening equation in (5) is false, then $\Delta_{\mathsf{open}} = 0$ with probability at most $(h-1)/|\mathbb{K}|$. If at least one reduced affine condition is violated, then $\Delta_{\mathsf{tr}} = 0$ with probability at most $\ell_F/|\mathbb{K}|$. Conditioned on $(\Delta_{\mathsf{open}}, \Delta_{\mathsf{tr}}) \neq (0,0)$, Equation (8) holds with probability at most $1/|\mathbb{K}|$. Consequently, if the opening equality or any reduced affine condition is violated, Equation (8) holds with probability at most $(h+\ell_F)/|\mathbb{K}|$.*

Proof. For the commitment-opening equation, let $\delta_a = (A_E v_E + B_{\mathsf{cap}} r_{\mathsf{cap}} - C_E)_a$. If the equation is false, the polynomial $p(X) = \sum_{a=0}^{h-1} \delta_a X^a$ is nonzero and has degree at most $h-1$. The equality $\Delta_{\mathsf{open}} = 0$ is exactly $p(\xi) = 0$, so the claim follows from the Schwartz–Zippel bound. For the reduced affine conditions, let e_i be the discrepancy in row i and consider the polynomial $\widehat{e}(r) = \sum_i e_i \,\mathrm{eq}(\mathrm{bin}(i), r)$. It is nonzero whenever some row discrepancy is nonzero, has total degree at most ℓ_F, and $\Delta_{\mathsf{tr}} = 0$ is exactly $\widehat{e}(r_{\mathsf{row}}) = 0$. Finally, if at least one compressed discrepancy is nonzero, then $\lambda_{\mathsf{open}}\Delta_{\mathsf{open}} + \lambda_{\mathsf{tr}}\Delta_{\mathsf{tr}}$ is a nonzero linear polynomial in the two batching challenges. The probability that a violated condition vanishes in either of the first two compressions is at most $(h-1+\ell_F)/|\mathbb{K}|$. Outside this event, the final linear combination vanishes with probability at most $1/|\mathbb{K}|$, which gives the stated bound. □

Combining this bound with the security of the underlying arguments gives the main security statement.

Theorem 4.3 (Security of the editor proof)**.** *Suppose that the capture commitment is hiding and binding and that the proof commitments are hiding and binding. Suppose that the Inner Product and Lookup Protocols are complete, knowledge sound, and SHVZK. Then the protocol in Figure 3 is a complete SHVZK interactive argument of knowledge for $\mathsf{R}^{\mathsf{red}}_{\mathsf{AE}}$. If ϵ_{arg} is the combined knowledge error of the underlying arguments and $\epsilon_{\mathsf{base}} := \epsilon_{\mathsf{arg}} + \epsilon_{\mathsf{bind}}$, the knowledge error of the resulting protocol is at most*

$$\epsilon_{\mathsf{base}} + \frac{h + \ell_F}{|\mathbb{K}|}. \tag{9}$$

Proof. Completeness. Let $(v_E, r_{\mathsf{cap}}, z)$ be a witness for $\mathsf{R}^{\mathsf{red}}_{\mathsf{AE}}$. Derive reconstructs the prescribed v_U, all public conditions hold, and $C_E = C_{\mathsf{cap}} - A_U v_U$. The reduced commitment-opening and affine checks therefore hold before compression. By linearity, both compressed discrepancies are zero, so Equation (8) holds for every choice of the verifier challenges. The witness also satisfies the required ranges, so completeness of the Inner Product and Lookup Protocols implies that the verifier accepts.

Knowledge soundness. Consider an accepting prover after the commitments com_v, com_z, and com_r have been fixed. By knowledge soundness of the underlying arguments, an extractor obtains values v_E, z, and r_{cap} that satisfy the combined inner-product equality and the corresponding range conditions, except with probability ϵ_{arg}. Binding guarantees, except with probability ϵ_{bind}, that every use of each pre-challenge commitment refers to the same challenge-independent extracted value. Conditioned on these events, the

discrepancy vectors meet the premise of Lemma 4.2. By Lemma 4.2, if the opening equality or any reduced affine condition is violated, the combined equality holds with probability at most $(h+\ell_F)/|\mathbb{K}|$. Together with the underlying-argument and binding errors, this gives the bound in (9). Outside this event, the extracted values satisfy (5), the reduced affine relation, and all range conditions. Together with the public checks performed in the first protocol step, these are exactly the conditions of $\mathsf{R}^{\mathsf{red}}_{\mathsf{AE}}$.

Special honest-verifier zero knowledge. The values v_U, A_U, A_E, C_E, and the reduced relation are determined by the public statement and reveal no additional witness value. Fix the honest verifier's public coins. We prove SHVZK by a sequence of hybrids. First retain the honestly generated proof commitments and replace, one at a time, the Inner Product and Lookup transcripts by the outputs of their special simulators for the prescribed verifier coins. The arguments are invoked in a fixed order with fresh prover randomness, so standard sequential composition makes the resulting view computationally indistinguishable from a real execution.

After all subargument transcripts have been simulated, replace com_v, com_r, and com_z, one at a time, by fresh commitments to fixed dummy vectors of the same public lengths. After each replacement, rerun the simulated subarguments that depend on the modified commitment tuple. Each transition follows from commitment hiding: a distinguisher between adjacent hybrids would yield a hiding adversary that inserts its challenge commitment into the tuple, runs the polynomial-time subargument simulators, and forwards the resulting view to the distinguisher.

Crucially, we use the SHVZK guarantee only while the original commitments make all subinstances valid. After replacing a commitment, the hiding reduction merely runs the polynomial-time simulator algorithms on the modified public input, without applying the SHVZK guarantee to the potentially invalid subinstances. The final hybrid is witness-independent and computationally indistinguishable from the real accepting view, and therefore also accepts with overwhelming probability. □

The evaluated implementation makes these checks non-interactive.

By Lemma 4.1, a complete source that opens C_{cap} and satisfies the edit can be reconstructed from an extracted reduced witness.

Efficiency. Let $D = |E_F|$ and $Z = |z|$. The reduction lowers the private dimension from $N + \mu + Z$ to $D + \mu + Z$; for the supported interval edits, the source-dependent proof work is governed by the edited interval. Public recovery and capture-time commitment computation still depend on the complete source.

Following the notation for the underlying arguments, let $T^{\mathsf{ip}}_{\mathcal{P}}(M)$, $T^{\mathsf{ip}}_{\mathcal{V}}(M)$, and $\pi^{\mathsf{ip}}(M)$ denote the prover time, verifier time, and proof size of one Inner Product Protocol on committed vectors of total length M. For a witness block of logical length n, let $T^{\mathsf{lk}}_{\mathcal{P}}(n)$, $T^{\mathsf{lk}}_{\mathcal{V}}(n)$, and $\pi^{\mathsf{lk}}(n)$ denote the corresponding costs of its lookup argument, and let $T^{\mathsf{com}}_{\mathcal{P}}(n)$ and π^{com} denote the time required to compute and the size of a commitment. We omit the fixed lookup tables from the notation. The cost functions include the canonical public padding required by the underlying arguments; zero-length blocks and their commitment and lookup terms are omitted.

Theorem 4.4 (Complexity of the Algebraic Editing Protocol)**.** *Let $D = |E_F|$, $Z = |z|$, $M = D + \mu + Z$, and let s_F be the total number of nonzero entries in $\widetilde{G}_F$ and $\widetilde{H}_F$. Let $\mathcal{B}_F$ be the multiset of positive lengths among D, μ, and Z. The prover time, verifier time excluding recovery of the public source entries, and proof size of the protocol in Figure 3 are as follows, where the $O(\cdot)$ terms count $\mathbb{K}$-field operations:*

$$T_{\mathcal{P}} = O\big(h(D+\mu) + m_F + s_F + M\big) + T^{\mathsf{ip}}_{\mathcal{P}}(M) + \sum_{n\in\mathcal{B}_F} \big(T^{\mathsf{com}}_{\mathcal{P}}(n) + T^{\mathsf{lk}}_{\mathcal{P}}(n)\big)\,,$$

$$T_{\mathcal{V}} = O\big(h(D+\mu) + m_F + s_F + M\big) + T^{\mathsf{ip}}_{\mathcal{V}}(M) + \sum_{n\in\mathcal{B}_F} T^{\mathsf{lk}}_{\mathcal{V}}(n),$$

$$|\pi| = \pi^{\mathsf{ip}}(M) + |\mathcal{B}_F|\pi^{\mathsf{com}} + \sum_{n\in\mathcal{B}_F} \pi^{\mathsf{lk}}(n).$$

Proof. Computing g_A and g_B takes $O(h(D+\mu))$ field operations. Computing ω, g_v, g_z, and d_{tr} takes $O(m_F + s_F)$ additional operations, and forming the combined coefficients takes $O(M)$. The protocol creates

one commitment for each nonempty witness block, invokes one Inner Product Protocol on the concatenated witness, and invokes one lookup argument for each of the same committed blocks. Adding the costs of these components gives the stated bounds. □

A formulation that retains the complete source vector would use a private vector of length $N + \mu + Z$, whereas the reduced protocol uses $M = D + \mu + Z$. For an interval of d PCM samples, interval removal has $D = Ld$ and no private transformation rows or auxiliary edit values. Bounded additive noise has $D = Ld$, $m_F = d$, and $Z = d$; fixed three-tap equalization has the same asymptotic dimensions, with one rounding value and only a constant number of source coefficients per edited sample. Hence, for these representative operations, the prover bounds in Theorem 4.4 depend on the edited interval rather than on the $N - D$ source entries recovered from the publication.

This reduction does not make every system operation independent of N. Before running the protocol, the verifier recovers v_U and computes $A_U v_U$; for a dense capture matrix, these public computations take $O(h(N-D))$ field operations. Thus, including this preprocessing, the verifier time is $T_{\mathcal{V}} + O(h(N-D))$. The capture commitment also processes the complete source vector at recording time. The reduction concerns the private witness and the proof computation over its hidden values, not these public or recording-time computations.

5 Privacy-Preserving Audio Authentication System

We formalize PPAAS and its security model, then instantiate it by combining the privacy-preserving capture-authentication scheme of Section 3 with the specialized zero-knowledge editing protocol of Section 4.

5.1 Formal Definition

Let $\mathcal{I}$ index the provisioned recording devices and $\mathcal{G}$ the authorized device groups. Device $i \in \mathcal{I}$ belongs to $g_i \in \mathcal{G}$, identified in its public capture object by ρ_{g_i}. Each capture object contains a canonical public descriptor d whose group identifier is denoted by $\rho(d)$ and whose audio profile fixes the encoded source length. The capture relation $\mathsf{R}_{\mathsf{cap}}$ contains $\big(\mathsf{cap}; (v, w_{\mathsf{cap}})\big)$ when cap authenticates the canonical source $v = \mathsf{Canon}(\mathsf{file})$ with witness w_{cap}. For a public edit F and private edit values θ, let $z_{\mathsf{aux}} \leftarrow \mathsf{Aux}(F, v, \theta)$ and write $z = (\theta, z_{\mathsf{aux}})$.

Definition 5.1 (Privacy-Preserving Audio Authentication System)**.** A privacy-preserving audio authentication system (PPAAS) consists of the following PPT algorithms.

- $\mathsf{Setup}(1^\lambda)$: outputs public parameters pp and device-specific secret states $\mathcal{D} = \{\mathsf{st}_i\}_{i \in \mathcal{I}}$, with st_i provisioned only on device i. The public parameters fix the audio profile, public procedures, capture and proof parameters, and authorized device groups.
- $\mathsf{Record}(pp, \mathsf{st}_i, \mathsf{file})$: returns $\perp$ if canonicalization fails; otherwise outputs a public capture object cap and private capture data (v, w_{cap}).
- $\mathsf{Prove}(pp, \mathsf{cap}, v, w_{\mathsf{cap}}, F, \theta)$: lets $y \leftarrow \mathsf{Apply}(F, v; \theta)$ and $z_{\mathsf{aux}} \leftarrow \mathsf{Aux}(F, v, \theta)$, sets $X = (\mathsf{cap}, y, F)$ and $w = (v, w_{\mathsf{cap}}, z)$, and outputs $\mathsf{pub} = (X, \pi)$, where π together with the public verification checks establishes the existence of w such that $(X; w) \in \mathsf{R}_{\mathsf{legal}}$.
- $\mathsf{Verify}(pp, \mathsf{pub})$: parses $\mathsf{pub} = (X, \pi)$ and outputs 1 iff the public conditions and the proof π for X accept.

Let R_{tr} be the audio transformation relation of Section 4.1. All relations below are parameterized by pp. The system legality relation is

$$\mathsf{R}_{\mathsf{legal}} = \left\{ \big((\mathsf{cap}, y, F); (v, w_{\mathsf{cap}}, z)\big) : \begin{array}{l} \big(\mathsf{cap}; (v, w_{\mathsf{cap}})\big) \in \mathsf{R}_{\mathsf{cap}}, \\ \big((y, F); (v, z)\big) \in \mathsf{R}_{\mathsf{tr}} \end{array} \right\}. \tag{10}$$

Because both conditions in Relation (10) share the hidden source v, every valid witness binds capture authentication and editing to the same recording.

Adversarial model. Let $\mathcal{A}$ be a PPT adversary against PPAAS. In each experiment, the challenger runs $(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda)$, gives pp to $\mathcal{A}$, retains the protected device states in $\mathcal{D}$, and initializes $Q_{\mathsf{cap}} = \emptyset$. The adversary has adaptive access to $\mathcal{O}_{\mathsf{Record}}(i, \mathsf{file})$. The oracle returns $\perp$ if $i \notin \mathcal{I}$; otherwise it runs $(\mathsf{cap}, (v, w_{\mathsf{cap}})) \leftarrow \mathsf{Record}(pp, \mathsf{st}_i, \mathsf{file})$, returning $\perp$ on failure, or parsing d from cap, adding (d, v) to Q_{cap} and returning $(\mathsf{cap}, (v, w_{\mathsf{cap}}))$ on success. All other PPAAS algorithms are public and may be run directly by $\mathcal{A}$. In the two-stage experiments below, both adversarial stages retain access to $\mathcal{O}_{\mathsf{Record}}$.

Completeness. For every provisioned device i, every file conforming to the public audio profile, every supported, well-formed F, and every admissible θ for $v = \mathsf{Canon}(\mathsf{file})$, we require

$$\Pr\left[\mathsf{Verify}(pp, \mathsf{pub}) = 0 : \begin{array}{l}(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda);\\ (\mathsf{cap}, (v, w_{\mathsf{cap}})) \leftarrow \mathsf{Record}(pp, \mathsf{st}_i, \mathsf{file});\\ \mathsf{pub} \leftarrow \mathsf{Prove}(pp, \mathsf{cap}, v, w_{\mathsf{cap}}, F, \theta)\end{array}\right] \le \mathsf{negl}(\lambda).$$

Capture authenticity. PPAAS has *capture authenticity* if, for every PPT adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{c}\mathsf{cap}^* \text{ parses with descriptor } d^* \wedge\\ (\mathsf{cap}^*; (v^*, w^*_{\mathsf{cap}})) \in \mathsf{R}_{\mathsf{cap}} \wedge\\ (d^*, v^*) \notin Q_{\mathsf{cap}}\end{array} : \begin{array}{l}(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda);\\ (\mathsf{cap}^*, (v^*, w^*_{\mathsf{cap}})) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Record}}}(pp)\end{array}\right] \le \mathsf{negl}(\lambda).$$

Edit correctness. We formalize edit correctness as knowledge soundness. PPAAS has *edit correctness* if, for every PPT adversary $\mathcal{A}$, there exists an expected polynomial-time extractor $\mathcal{E}$ such that, with (X^*, π^*) denoting the candidate publication output by $\mathcal{A}$ in $\mathcal{E}$'s main execution,

$$\Pr\left[\begin{array}{c}\mathsf{Verify}(pp, (X^*, \pi^*)) = 1 \wedge\\ (w^* = \perp \ \vee \ (X^*; w^*) \notin \mathsf{R}_{\mathsf{legal}})\end{array} : \begin{array}{l}(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda);\\ ((X^*, \pi^*), w^*) \leftarrow \mathcal{E}^{\mathcal{A}, \mathcal{O}_{\mathsf{Record}}}(pp)\end{array}\right] \le \mathsf{negl}(\lambda).$$

Audio privacy. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any PPT adversary. For $b \in \{0, 1\}$, let $v_b = \mathsf{Canon}(\mathsf{file}_b)$ and $y_b = \mathsf{Apply}(F^*, v_b; \theta_b)$. We restrict $\mathcal{A}_1$ to outputs for which $i^* \in \mathcal{I}$, (F^*, θ_b) is admissible for v_b for each $b \in \{0, 1\}$, $|v_0| = |v_1|$, and $y_0 = y_1$. PPAAS has *audio privacy* if

$$\left|\Pr\left[\beta' = \beta : \begin{array}{l}(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda);\\ (\mathsf{st}_{\mathcal{A}}, i^*, F^*, \mathsf{file}_0, \theta_0, \mathsf{file}_1, \theta_1)\\ \qquad \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Record}}}(pp);\\ \beta \xleftarrow{\mathrm{R}} \{0, 1\};\\ (\mathsf{cap}^*, (v_\beta, w_{\mathsf{cap},\beta})) \leftarrow \mathsf{Record}(pp, \mathsf{st}_{i^*}, \mathsf{file}_\beta);\\ \mathsf{pub}^* \leftarrow \mathsf{Prove}(pp, \mathsf{cap}^*, v_\beta, w_{\mathsf{cap},\beta}, F^*, \theta_\beta);\\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Record}}}(\mathsf{st}_{\mathcal{A}}, \mathsf{pub}^*)\end{array}\right] - \frac{1}{2}\right| \le \mathsf{negl}(\lambda).$$

Device anonymity. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any PPT adversary. We restrict $\mathcal{A}_1$ to outputs with distinct indices $i_0, i_1 \in \mathcal{I}$ belonging to the same authorized device group, namely $g_{i_0} = g_{i_1}$, and (F^*, θ^*) is admissible for $v^* = \mathsf{Canon}(\mathsf{file}^*)$. PPAAS has *device anonymity* if

$$\left|\Pr\left[\beta' = \beta : \begin{array}{l}(pp, \mathcal{D}) \leftarrow \mathsf{Setup}(1^\lambda);\\ (\mathsf{st}_{\mathcal{A}}, i_0, i_1, \mathsf{file}^*, F^*, \theta^*) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Record}}}(pp);\\ \beta \xleftarrow{\mathrm{R}} \{0, 1\};\\ (\mathsf{cap}^*, (v^*, w^*_{\mathsf{cap}})) \leftarrow \mathsf{Record}(pp, \mathsf{st}_{i_\beta}, \mathsf{file}^*);\\ \mathsf{pub}^* \leftarrow \mathsf{Prove}(pp, \mathsf{cap}^*, v^*, w^*_{\mathsf{cap}}, F^*, \theta^*);\\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{Record}}}(\mathsf{st}_{\mathcal{A}}, \mathsf{pub}^*)\end{array}\right] - \frac{1}{2}\right| \le \mathsf{negl}(\lambda).$$

A PPAAS is secure if it satisfies all five properties above.

5.2 Generic Construction

We instantiate the definition using the capture scheme Π_{cap} of Section 3 and the editing protocol of Section 4. Let Π_{com} and Π_{ring} be the two components of Π_{cap}, and choose Π_{com} with the linear opening relation of Definition 2.3. For each admissible descriptor d, canonical source vectors belong to the fixed-length message space $\mathcal{M}_d$. The corresponding public description $pp_{\mathsf{cap}}[d]$ uses $A_{\mathsf{cap}}[d] = A_{\mathsf{com}}[d]$, $B_{\mathsf{cap}} = B_{\mathsf{com}}$, and $\mathcal{R}_{\mathsf{cap}} = \mathcal{R}_{\mathsf{com}}$. Write pp_{cap} for this descriptor-indexed family. Let $\Pi_{\mathsf{edit}} = (\mathsf{Setup}_{\mathsf{edit}}, \mathsf{Prove}_{\mathsf{edit}}, \mathsf{Verify}_{\mathsf{edit}})$ be a non-interactive argument of knowledge for $\mathsf{R}_{\mathsf{AE}}^{\mathsf{red}}$.

Let pp_{sys} collect the descriptor syntax, public audio profiles, and the descriptions of Canon, Apply, Derive, and Aux. Let $\mathsf{ValidDesc}(pp, d)$ check the canonical descriptor syntax, commitment-parameter identifier, and audio profile. Setup fixes an injective map $\mathsf{Reg} = \{\rho_g \mapsto R_g\}_{g\in\mathcal{G}}$ from group identifiers to ordered rings. We write $R_\rho = \mathsf{Reg}[\rho]$ and let $\mathsf{ValidRing}(pp, \rho)$ check that ρ identifies a setup-registered ring.

The public capture object is $\mathsf{cap} = (d, C_{\mathsf{cap}}, \sigma)$ and its private witness is $w_{\mathsf{cap}} = r_{\mathsf{cap}}$. Accordingly, the abstract capture relation is instantiated as

$$\mathsf{R}_{\mathsf{cap}} = \left\{ \big((d, C_{\mathsf{cap}}, \sigma); (v, r_{\mathsf{cap}})\big) : \begin{array}{l} \mathsf{ValidDesc}(pp, d) = 1, \quad v \in \mathcal{M}_d, \\ \mathsf{ValidRing}(pp, \rho(d)) = 1, \\ \mathsf{Verify}_{\mathsf{ring}}(R_{\rho(d)}, (d, C_{\mathsf{cap}}), \sigma) = 1, \\ \mathsf{Open}(pp_{\mathsf{com}}[d], C_{\mathsf{cap}}, v, r_{\mathsf{cap}}) = 1 \end{array} \right\}. \tag{11}$$

The verifier checks the public descriptor, ring, and signature conditions directly, while Π_{edit} proves the opening and transformation conditions through $\mathsf{R}_{\mathsf{AE}}^{\mathsf{red}}$, which is equivalent to R_{AE} by Lemma 4.1. Together, these checks establish $\mathsf{R}_{\mathsf{legal}}$; Figure 4 gives the resulting algorithms.

Setup(1^λ)

1: $pp_{\mathsf{com}} \leftarrow \mathsf{Setup}_{\mathsf{com}}(1^\lambda)$; derive the descriptor-indexed family $pp_{\mathsf{cap}}[d]$ as above and fix pp_{sys}.
2: Run $\mathsf{Setup}_{\mathsf{edit}}(1^\lambda, pp_{\mathsf{cap}}, pp_{\mathsf{sys}})$ to obtain pp_{edit}.
3: Each device i runs $(vk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\mathsf{cap}}(1^\lambda)$ and retains sk_i.
4: For every group g, set $R_g = (vk_i)_{i:g_i=g}$ in a fixed order and fix the injective map $\mathsf{Reg} = \{\rho_g \mapsto R_g\}_{g\in\mathcal{G}}$.
5: Device i stores $\mathsf{st}_i = (sk_i, \rho_{g_i})$.
6: Set $pp \leftarrow (pp_{\mathsf{cap}}, pp_{\mathsf{edit}}, pp_{\mathsf{sys}}, \mathsf{Reg})$; **return** $(pp, \mathcal{D})$, where $\mathcal{D} = \{\mathsf{st}_i\}_{i\in\mathcal{I}}$.

Record$(pp, \mathsf{st}_i, \mathsf{file})$

1: Parse $\mathsf{st}_i = (sk_i, \rho)$; set $R_\rho \leftarrow \mathsf{Reg}[\rho]$ and $v \leftarrow \mathsf{Canon}(\mathsf{file})$; if parsing, lookup, or canonicalization fails, **return** $\perp$.
2: Form the canonical descriptor d from ρ, the commitment-parameter identifier, and the public profile of v, including its length.
3: $\Sigma = (d, C_{\mathsf{cap}}, r_{\mathsf{cap}}, \sigma) \leftarrow \mathsf{Sign}_{\mathsf{cap}}(sk_i, R_\rho, d, v)$.
4: $\mathsf{cap} \leftarrow (d, C_{\mathsf{cap}}, \sigma)$.
5: **return** $(\mathsf{cap}, (v, r_{\mathsf{cap}}))$.

Prove$(pp, \mathsf{cap}, v, r_{\mathsf{cap}}, F, \theta)$

1: Parse $\mathsf{cap} = (d, C_{\mathsf{cap}}, \sigma)$ and validate d; on failure, **return** $\perp$.
2: Compute $y \leftarrow \mathsf{Apply}(F, v; \theta)$ and $z_{\mathsf{aux}} \leftarrow \mathsf{Aux}(F, v, \theta)$; run $\mathsf{Derive}(F, y)$ and return $\perp$ if it rejects.
3: Using the returned partition, set $v_E \leftarrow P_E^{\mathsf{T}} v$ and $z \leftarrow (\theta, z_{\mathsf{aux}})$.
4: $\mathsf{stmt} \leftarrow (pp_{\mathsf{cap}}[d], C_{\mathsf{cap}}, y, F)$; $\mathsf{wit} \leftarrow (v_E, r_{\mathsf{cap}}, z)$.
5: $\pi \leftarrow \mathsf{Prove}_{\mathsf{edit}}(pp_{\mathsf{edit}}, \mathsf{stmt}, \mathsf{wit})$.
6: Set $X \leftarrow (\mathsf{cap}, y, F)$; **return** $\mathsf{pub} = (X, \pi)$.

Verify(pp, pub)

1: Parse $\mathsf{pub} = (X, \pi)$, $X = (\mathsf{cap}, y, F)$, and $\mathsf{cap} = (d, C_{\mathsf{cap}}, \sigma)$; on failure, **return** 0.
2: $b_{\mathsf{ring}} \leftarrow \mathsf{ValidDesc}(pp, d) \wedge \mathsf{ValidRing}(pp, \rho(d))$; if $b_{\mathsf{ring}} = 0$, **return** 0. Set $R_{\rho(d)} \leftarrow \mathsf{Reg}[\rho(d)]$ and $b_{\mathsf{sig}} \leftarrow \mathsf{Verify}_{\mathsf{ring}}(R_{\rho(d)}, (d, C_{\mathsf{cap}}), \sigma)$.
3: $\mathsf{stmt} \leftarrow (pp_{\mathsf{cap}}[d], C_{\mathsf{cap}}, y, F)$.
4: $b_{\mathsf{edit}} \leftarrow \mathsf{Verify}_{\mathsf{edit}}(pp_{\mathsf{edit}}, \mathsf{stmt}, \pi)$.
5: **return** $b_{\mathsf{sig}} \wedge b_{\mathsf{edit}}$.

Figure 4: Generic construction of PPAAS from the capture and editing protocols.

Theorem 5.1 (PPAAS composition)**.** *Assume that* Canon, Apply, Derive, *and* Aux *satisfy their stated correctness conditions for every supported edit and that provisioned signing keys remain hidden from $\mathcal{A}$. If the descriptor-indexed commitment family is correct, hiding, and binding and has the linear opening relation of Definition 2.3, Π_{ring} is correct and satisfies fixed-ring unforgeability and signer anonymity, and Π_{edit} is a complete and computationally zero-knowledge non-interactive argument of knowledge for $\mathsf{R}_{\mathsf{AE}}^{\mathsf{red}}$ in the programmable random-oracle model, then the construction in Figure 4 satisfies all five PPAAS properties defined in Section 5.1.*

Proof. We prove the five properties in turn.

Completeness. Let d be the descriptor produced by the honest recording algorithm. An honest execution uses the ring identifier stored in st_i, so $\mathsf{ValidRing}(pp, \rho_{g_i}) = 1$. Correctness of the commitment and ring-signature schemes gives a commitment that opens to the canonical source v under d and a valid ring signature on (d, C_{cap}). Thus, by the construction of Π_{cap}, Relation (11) is satisfied by (v, r_{cap}). Correctness of Apply and Aux gives y and z, while Derive gives v_U, the partition, and the reduced affine relation. Setting $v_E = P_E^{\mathsf{T}} v$ satisfies that relation, and $C_E = C_{\mathsf{cap}} - A_U v_U$ satisfies the reduced opening equation. Thus, $(v_E, r_{\mathsf{cap}}, z)$ satisfies $\mathsf{R}_{\mathsf{AE}}^{\mathsf{red}}$ for the statement $(pp_{\mathsf{cap}}[d], C_{\mathsf{cap}}, y, F)$, and completeness of Π_{edit} makes $b_{\mathsf{edit}} = 1$. The two public capture checks also accept, so Verify returns 1.

Capture authenticity. Suppose an adversary outputs $\mathsf{cap}^* = (d^*, C^*, \sigma^*)$ and (v^*, r^*) satisfying Relation (11), with $(d^*, v^*) \notin Q_{\mathsf{cap}}$. Because d^* passes the public checks, $R_{\rho(d^*)}$ is one of the rings fixed by setup. The tuple $\Sigma^* = (d^*, C^*, r^*, \sigma^*)$ is therefore an accepting Π_{cap} signature on (d^*, v^*) for that fixed ring. Moreover, $(d^*, v^*) \notin Q_{\mathsf{cap}}$ means that the same descriptor and source were not returned by an honest recording request. Guessing the target among the polynomially many provisioned rings gives, with only a polynomial loss, a forger against Π_{cap} for one fixed ring. As established in Theorem 3.1, such a forgery yields either a new-message forgery against Π_{ring} or two openings of one commitment to different sources. The former contradicts fixed-ring unforgeability and the latter contradicts binding of Π_{com}. Hence capture authenticity holds.

Edit correctness. Fix an accepting publication (X^*, π^*). Write $X^* = (\mathsf{cap}^*, y^*, F^*)$. Parse $\mathsf{cap}^* = (d^*, C_{\mathsf{cap}}^*, \sigma^*)$. Acceptance establishes $\mathsf{ValidDesc}(pp, d^*) = 1$ and $\mathsf{ValidRing}(pp, \rho(d^*)) = 1$. It also establishes $\mathsf{Verify}_{\mathsf{ring}}(R_{\rho(d^*)}, (d^*, C_{\mathsf{cap}}^*), \sigma^*) = 1$, while acceptance of the editing proof implies that $\mathsf{Derive}(F^*, y^*)$ succeeds and deterministically fixes v_U^* and $C_E^* = C_{\mathsf{cap}}^* - A_U v_U^*$. By knowledge soundness of Π_{edit}, the extractor obtains $(v_E^*, r_{\mathsf{cap}}^*, z^*)$ for the accepting publication, satisfying $\mathsf{R}_{\mathsf{AE}}^{\mathsf{red}}$ for the statement $(pp_{\mathsf{cap}}[d^*], C_{\mathsf{cap}}^*, y^*, F^*)$, except with negligible probability. Lemma 4.1 reconstructs $v^* = P_U v_U^* + P_E v_E^*$ and establishes

$$\mathsf{Open}(pp_{\mathsf{com}}[d^*], C_{\mathsf{cap}}^*, v^*, r_{\mathsf{cap}}^*) = 1,$$
$$\big((y^*, F^*); (v^*, z^*)\big) \in \mathsf{R}_{\mathsf{tr}}.$$

Together with the two public capture checks, the extracted opening satisfies Relation (11); hence these conditions give $(X^*; (v^*, r_{\mathsf{cap}}^*, z^*)) \in \mathsf{R}_{\mathsf{legal}}$. Thus, every accepting publication has an extractable witness for the system legality relation, except with negligible probability.

Audio privacy. Consider valid challenge sources v_0, v_1 of the same length, and let their edits produce the same (y, F). We use three hybrid transitions. In H_0, the challenger publishes the real challenge publication generated from v_0. In H_1, it replaces the editing proof with a simulated proof for the same public statement; zero knowledge makes H_0 and H_1 computationally indistinguishable. In H_2, it replaces the commitment to v_0 with a commitment to v_1. Couple the two worlds using the same admissible public descriptor d. The value $v_U = \mathsf{Recover}(F, y)$ is identical in both challenge worlds, so $C_E = C_{\mathsf{cap}} - A_U v_U$ is an efficiently computed function of the challenge commitment and common public values. After replacing the commitment, the challenger regenerates the ring signature on (d, C) and the simulated editing proof. A hiding reduction can perform both operations using an honestly generated device signing key and the editing-proof simulator. A distinguisher between H_1 and H_2 therefore breaks hiding of Π_{com}. Finally, H_3 replaces the simulated proof with the real proof for v_1; zero knowledge again makes the two hybrids indistinguishable. The last hybrid is the real challenge publication generated from v_1, establishing audio privacy.

Device anonymity. Because the challenge devices belong to the same authorized group, they use the same ring identifier and ring. Couple the two challenge experiments by using the same source, edit, commitment

randomness, and editing-proof randomness. The resulting descriptor, C_{cap}, y, F, reduced statement, and editing witness are identical and independent of the challenge device. The only signer-dependent value is the ring signature σ on (d, C_{cap}). A distinguisher between the two complete publications therefore directly yields a distinguisher for which of the two ring members signed the same descriptor and commitment, contradicting signer anonymity of Π_{ring}. This establishes all five properties. □

Complexity analysis. We analyze the computation and communication costs of the PPAAS authentication procedure, covering capture, proof generation, and verification. For the ring R_ρ identified in a capture object, let $n_R = |R_\rho|$, let N be the length of the canonical source vector, and retain the notation of Theorem 4.4: $D = |E_F|$, $\mu = |r_{\mathsf{cap}}|$, $Z = |z|$, and $M = D + \mu + Z$. Write $T_{\mathsf{com}}(N)$ for the capture-commitment time, $T^{\mathsf{ring}}_{\mathsf{sign}}(n_R)$ and $T^{\mathsf{ring}}_{\mathsf{vrfy}}(n_R)$ for ring signing and verification, and T_{reg} for validating a ring identifier and retrieving its ring from Reg. Let S_{rid}, S_{com}, and $S_{\mathsf{sig}}(n_R)$ denote the encoded sizes of a ring reference, capture commitment, and ring signature. Write $T^{\mathsf{edit}}_{\mathsf{P}}(M)$, $T^{\mathsf{edit}}_{\mathsf{V}}(M)$, and $S_{\mathsf{edit}}(M)$ for the costs of Π_{edit}, with its remaining parameters suppressed. Finally, let $T_{\mathsf{pub}}(N - D, F)$ denote Derive, including recovery of v_U and computation of $A_U v_U$. The cryptographic costs of the three principal stages are

$$\begin{aligned} T_{\mathsf{Record}} &= T_{\mathsf{com}}(N) + T^{\mathsf{ring}}_{\mathsf{sign}}(n_R), \\ T_{\mathsf{Prove}} &= T_{\mathsf{pub}}(N - D, F) + T^{\mathsf{edit}}_{\mathsf{P}}(M), \\ T_{\mathsf{Verify}} &= T_{\mathsf{reg}} + T^{\mathsf{ring}}_{\mathsf{vrfy}}(n_R) + T_{\mathsf{pub}}(N - D, F) + T^{\mathsf{edit}}_{\mathsf{V}}(M). \end{aligned}$$

Excluding the released audio y and its edit description F, the authentication-data size is

$$S_{\mathsf{auth}} = S_{\mathsf{rid}} + S_{\mathsf{com}} + S_{\mathsf{sig}}(n_R) + S_{\mathsf{edit}}(M).$$

The authenticated ring itself is public setup data shared by all recordings in its group and is therefore not retransmitted with every publication. Thus, the system adds one commitment and one ring signature at capture, and one editing proof at publication; verification performs the corresponding public checks once.

6 Implementation and Evaluation

This section evaluates the prototype implementation of our privacy-preserving audio authentication system. We report setup time, proof generation time, proof verification time, prover peak memory, and authentication-proof payload. For editing proofs, prover setup and proof generation are measured separately; for the capture procedure, we separately report source processing and signature generation. Experiments were performed on a machine with an Intel Xeon w5-3433 processor, 32 logical CPUs, 256 GB of RAM, and Ubuntu 22.04.5 LTS. We evaluate our construction and the general-purpose zkSNARK baseline (hereafter denoted by Π_{zkSNARK}) on the same machine and with the same source and edit instances. The inputs are synthetically generated mono 16-bit PCM recordings at 16 kHz and are deterministic across the two constructions. We write N_s for the number of source samples and E_s for the number of samples in the edited interval.

6.1 Implementation Details

We implement the system in Rust as two prototype instantiations. The capture procedure in our construction computes the randomized linear commitment described in Section 3 and authenticates its public descriptor and commitment with a fixed-ring Schnorr–OR signature over Ristretto255. Both constructions use authorized groups of 256 devices. The editor-side argument uses sumcheck for the algebraic checks, a transparent BaseFold polynomial commitment scheme [ZCF23] for polynomial commitments and openings, and the VEIL masking compiler [DHRR26, Suc26]. Following Plonky2 [Pol21], its rate-1/8 profile uses 28 proximity queries and 16 bits of grinding for a nominal 100-bit local target under the conjectural $1 - \rho$ regime; this is not an end-to-end AoK claim.

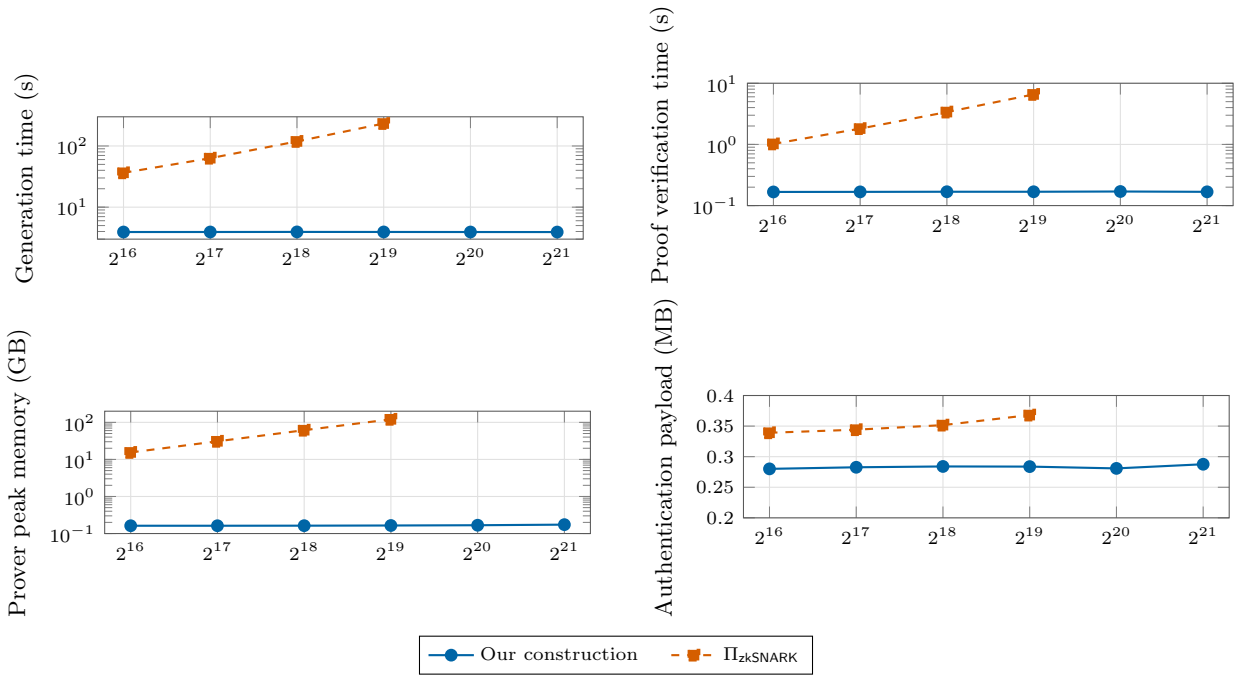

Figure 5: Proof costs for AddNoise with $E_s = 2^{16}$ as the source size grows. The time and memory axes are logarithmic, while the authentication-payload axis is linear. Missing Π_{zkSNARK} points at $N_s = 2^{20}$ and 2^{21} are executions that exceeded the 240 GB memory limit.

For comparison, the Plonky2 baseline Π_{zkSNARK} [Pol21] uses a public Poseidon source hash [GKR+21] and separate general-purpose zkSNARKs for ECDSA registry membership [Pol23] and editing under the same PCM semantics. The device public key, signature, and membership path are private witnesses, so its authentication proof also hides the selected member; its deterministic source hash, however, does not provide source-content hiding. The reported ratios compare the two aggregate proof-system architectures rather than isolate individual design choices.

The prototype supports the three transformations defined in Section 4.1. The release and edit description are public, whereas the source and auxiliary values form the witness; Π_{zkSNARK} encodes the same semantics in a global full-source circuit.

6.2 Audio Editing Proof Results

The editing experiments vary the source size and edited interval. For source-size scaling, we use *AddNoise* because it preserves the source length while exercising private edit values and range checks. We use *setup* for relation or circuit construction and its associated inputs and witness, not for a one-time public-parameter or trusted setup. T_{proofV} measures only proof verification; public-context reconstruction, proof decoding, native ring authentication, and other publication-workflow costs are excluded.

Figure 5 examines AddNoise at $E_s = 2^{16}$. As N_s grows from 2^{16} to 2^{21}, our proof generation remains within 3.92–3.94 s, proof verification within 0.167–0.170 s, and prover peak memory within 0.162–0.174 GB. Across the baseline's completed points, our construction is 9.3–59.3× faster in proving, 6.1–39.0× faster in proof verification, uses 95.1–733.1× less prover memory, and produces 17.4–22.9% smaller authentication-proof payloads under the convention that excludes the published PCM. Moreover, the baseline does not complete at $N_s = 2^{20}$ or 2^{21} under the 240 GB limit. The nearly flat post-setup proving, proof verification, and memory costs are consistent with the reduced relation depending primarily on the edited interval. To determine whether the behavior in Figure 5 also holds for the other transformations, Table 1 reports all

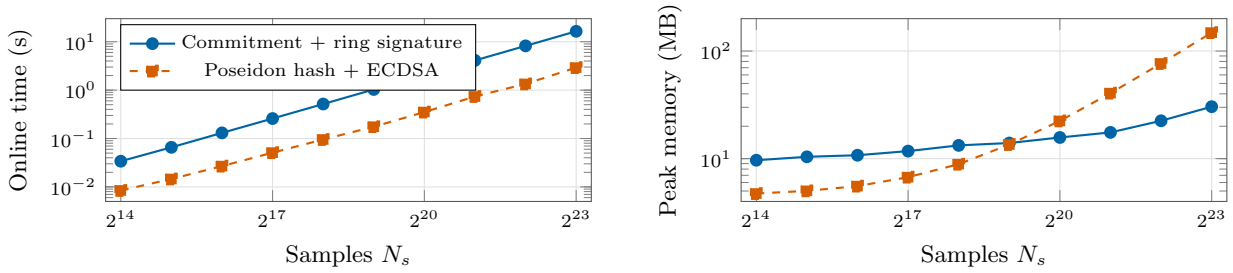

Figure 6: Recording cost as the source recording grows.

three transformations in both experiments.

We use $N_s = 2^{19}, E_s = 2^{16}$ as a representative point because it is shared by the two experiments and $N_s = 2^{19}$ is the largest source size for which Π_{zkSNARK} completes all three relations in the source-size experiment. At this point, PPAAS is 269.4–284.5× faster in setup, 59.1–85.6× faster in proving, 37.6–43.7× faster in proof verification, and uses 714.9–733.1× less prover memory across the three relations.

The two experiments also show which input dimension determines the cost. At fixed $E_s = 2^{16}$, our proof-generation and proof-verification times remain nearly constant for each relation as N_s increases, while setup grows with the source size. This separates source-dependent instance construction from the cost of proving the edited relation. In contrast, the proof-generation time of Π_{zkSNARK} rises with N_s, consistent with the growth of its full-source circuit, and it exceeds the 240 GB limit at $N_s = 2^{20}$ and 2^{21}. At fixed $N_s = 2^{19}$, our proof-generation time instead grows with E_s: from 0.58 to 10.89 s for Remove and from approximately 0.82 to 32.44 s for AddNoise and FixedEq. The corresponding time for Π_{zkSNARK} remains close to 229–238 s for AddNoise and FixedEq because the complete source still determines the circuit size. For Remove, the shorter published recording coincides with a smaller padded circuit at $E_s = 2^{18}$, reducing the baseline proof-generation time to 119.04 s. Remove is reported only when $N_s > E_s$, and dashes denote executions that exceeded the memory limit. These scaling trends are consistent with proving the reduced affine relation directly rather than constructing a general-purpose circuit over the full source.

Excluding the published PCM, our authentication-proof payloads are 0.20–0.32 MB, compared with 0.34–0.37 MB for Π_{zkSNARK}, a reduction of 11.0–44.9% across the completed pairs. Under this accounting, PPAAS comprises one specialized editing argument and a native ring signature, whereas the baseline comprises two general-purpose zkSNARK proofs. The largest observed advantages occur for Remove at $N_s = 2^{19}$ and $E_s = 2^{14}$: 398.0× in proving, 160.2× in proof verification, 1667.0× in prover memory, and a 44.9% smaller authentication-proof payload.

6.3 Comparing the Two Recording Schemes

We measure source processing and signature generation as N_s grows from 2^{14} to 2^{23}. Our construction uses a randomized linear commitment and fixed-ring signature; the baseline uses a Poseidon hash and ECDSA. Reported output excludes the audio and public object metadata.

Figure 6 shows approximately linear processing time but different memory growth. At $N_s = 2^{23}$, our commitment takes 16.43 s and 30.40 MB, versus 2.92 s and 148.31 MB for the baseline hash. Ring and ECDSA signing take 1.087 and 2.267 ms, respectively, so source processing dominates. At 16 kHz, our commitment processes this 8.7-minute input at about 32 times its acquisition rate, trading additional computation for slower memory growth.

At this point, our procedure is 5.6× slower and uses 4.9× less peak memory; its 18,506-B authentication payload, versus 96 B, reflects the anonymity and hiding mechanisms. Prior work demonstrates capture-time hashing and signing in commercial cameras and Schnorr-style ring signing on constrained hardware [MVVZ25, GMS26, KM20]. Our measurements establish server-side throughput only; recording-hardware evaluation remains future work.

Operation	Size	Construction	T_{setup}	T_{prove}	T_{proofV}	M_P	$\lvert\pi\rvert_{\text{auth}}$
(a) Fixed $E_s = 2^{16}$ and varying N_s							
Remove	2^{17}	Ours	0.265	2.697	0.149	0.161	0.229
		Π_{zkSNARK}	35.889	36.234	1.007	15.454	0.337
Remove	2^{18}	Ours	0.529	2.714	0.149	0.166	0.225
		Π_{zkSNARK}	144.190	117.206	3.318	60.909	0.355
Remove	2^{19}	Ours	1.057	2.697	0.150	0.168	0.225
		Π_{zkSNARK}	296.695	230.871	6.547	120.307	0.364
Remove	2^{20}	Ours	2.106	2.688	0.150	0.171	0.224
		Π_{zkSNARK}	–	–	–	–	–
Remove	2^{21}	Ours	4.204	2.681	0.149	0.180	0.228
		Π_{zkSNARK}	–	–	–	–	–
AddNoise	2^{16}	Ours	0.137	3.922	0.167	0.162	0.280
		Π_{zkSNARK}	37.967	36.438	1.019	15.431	0.339
AddNoise	2^{17}	Ours	0.269	3.926	0.168	0.162	0.283
		Π_{zkSNARK}	70.686	63.156	1.814	30.821	0.344
AddNoise	2^{18}	Ours	0.533	3.937	0.168	0.163	0.284
		Π_{zkSNARK}	140.240	118.687	3.378	61.410	0.352
AddNoise	2^{19}	Ours	1.061	3.931	0.168	0.165	0.284
		Π_{zkSNARK}	285.756	232.978	6.566	120.804	0.368
AddNoise	2^{20}	Ours	2.113	3.922	0.170	0.168	0.281
		Π_{zkSNARK}	–	–	–	–	–
AddNoise	2^{21}	Ours	4.211	3.916	0.168	0.174	0.288
		Π_{zkSNARK}	–	–	–	–	–
FixedEq	2^{16}	Ours	0.137	3.951	0.174	0.162	0.287
		Π_{zkSNARK}	39.298	36.878	1.146	15.423	0.337
FixedEq	2^{17}	Ours	0.269	3.925	0.175	0.165	0.283
		Π_{zkSNARK}	73.088	64.194	1.850	30.458	0.347
FixedEq	2^{18}	Ours	0.535	3.927	0.173	0.166	0.281
		Π_{zkSNARK}	145.504	120.507	3.494	61.410	0.356
FixedEq	2^{19}	Ours	1.060	3.937	0.175	0.169	0.280
		Π_{zkSNARK}	301.596	232.826	6.567	120.796	0.359
FixedEq	2^{20}	Ours	2.112	3.921	0.174	0.178	0.281
		Π_{zkSNARK}	–	–	–	–	–
FixedEq	2^{21}	Ours	4.210	3.932	0.172	0.193	0.280
		Π_{zkSNARK}	–	–	–	–	–
(b) Fixed $N_s = 2^{19}$ and varying E_s							
Remove	2^{14}	Ours	1.056	0.584	0.041	0.072	0.201
		Π_{zkSNARK}	304.673	232.296	6.600	120.815	0.365
Remove	2^{15}	Ours	1.056	1.358	0.078	0.097	0.205
		Π_{zkSNARK}	304.840	232.806	6.666	120.145	0.370
Remove	2^{16}	Ours	1.057	2.697	0.150	0.168	0.225
		Π_{zkSNARK}	296.695	230.871	6.547	120.307	0.364
Remove	2^{17}	Ours	1.056	5.380	0.295	0.307	0.238
		Π_{zkSNARK}	300.207	228.709	6.428	119.879	0.364
Remove	2^{18}	Ours	1.055	10.886	0.574	0.588	0.251
		Π_{zkSNARK}	128.727	119.043	3.494	62.163	0.350
AddNoise	2^{14}	Ours	1.059	0.826	0.046	0.074	0.254
		Π_{zkSNARK}	284.780	229.058	6.485	120.450	0.365
AddNoise	2^{15}	Ours	1.058	1.999	0.088	0.096	0.263
		Π_{zkSNARK}	286.443	229.299	6.509	120.903	0.366
AddNoise	2^{16}	Ours	1.061	3.931	0.168	0.165	0.284
		Π_{zkSNARK}	285.756	232.978	6.566	120.804	0.368
AddNoise	2^{17}	Ours	1.063	7.863	0.330	0.307	0.299
		Π_{zkSNARK}	284.142	229.052	6.636	120.584	0.363
AddNoise	2^{18}	Ours	1.070	15.959	0.651	0.589	0.311
		Π_{zkSNARK}	282.019	230.955	6.819	121.651	0.362
AddNoise	2^{19}	Ours	1.080	32.412	1.308	1.153	0.324
		Π_{zkSNARK}	284.223	233.004	7.031	122.606	0.363
FixedEq	2^{14}	Ours	1.058	0.817	0.048	0.073	0.251
		Π_{zkSNARK}	302.628	232.733	6.623	120.177	0.364
FixedEq	2^{15}	Ours	1.058	1.960	0.090	0.098	0.255
		Π_{zkSNARK}	300.765	232.117	6.615	120.648	0.366
FixedEq	2^{16}	Ours	1.060	3.937	0.175	0.169	0.280
		Π_{zkSNARK}	301.596	232.826	6.567	120.796	0.359
FixedEq	2^{17}	Ours	1.064	7.838	0.341	0.309	0.295
		Π_{zkSNARK}	300.298	236.183	7.061	120.834	0.366
FixedEq	2^{18}	Ours	1.070	15.841	0.679	0.592	0.312
		Π_{zkSNARK}	293.799	236.482	6.809	121.160	0.363
FixedEq	2^{19}	Ours	1.082	32.436	1.351	1.152	0.322
		Π_{zkSNARK}	288.544	238.461	7.331	122.295	0.362

Blue marks the lower value in completed pairs and our values when Π_{zkSNARK} is unavailable. T_{setup}, T_{prove}, and T_{proofV} are prover setup, proof generation, and proof-system verification; M_P is prover peak RSS, and $\lvert\pi\rvert_{\text{auth}}$ excludes the separately published PCM. Dashes mark 240 GB memory-limit failures; Remove requires $N_s > E_s$; $N_s = 2^{19}$, $E_s = 2^{16}$ appears in both parts.

Table 1: Editing-proof performance for all evaluated configurations. Times are in seconds, prover peak RSS is in GB, and authentication-proof payload is in MB.

Scheme	Processing	Signature	Memory	Output
Commitment + ring signature	16.430 s	**1.087 ms**	**30.40 MB**	18,506 B
Poseidon hash + ECDSA	**2.918 s**	2.267 ms	148.31 MB	**96 B**

Table 2: Representative recording-side measurements at $N_s = 2^{23}$. Source processing is commitment or hashing time. Output is the commitment or hash plus its signature, excluding the audio and public object metadata.

7 Conclusion

We presented PPAAS, which combines privacy-preserving capture authentication with a specialized zero-knowledge protocol for publicly verifiable audio editing. It hides from public verifiers unreleased source information and the recorder within its disclosed group, and directly proves affine edits while avoiding general-circuit conversion and redundant full-source work. Our prototype benchmarks characterize recorder-side costs and show substantial reductions relative to the general-purpose zkSNARK baseline.

References

[AFK23] Thomas Attema, Serge Fehr, and Michael Klooß. Fiat–Shamir transformation of Multi-Round interactive proofs (extended version). *Journal of Cryptology*, 36(4):36, 2023.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *Advances in Cryptology – EUROCRYPT 2016, Part II*, pages 327–357. Springer, 2016.

[BCG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In *Advances in Cryptology – ASIACRYPT 2017, Part III*, pages 336–365. Springer, 2017.

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *Advances in Cryptology – CRYPTO '92*, pages 390–420. Springer, 1993.

[BKM06] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. In *Theory of Cryptography – TCC 2006*, pages 60–79. Springer, 2006.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[BSCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Theory of Cryptography – TCC 2016-B, Part II*, pages 31–60. Springer, 2016.

[CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. Hyperplonk: Plonk with linear-time prover and high-degree custom gates. In *Advances in Cryptology – EUROCRYPT 2023, Part II*, pages 499–530. Springer, 2023.

[Coa25] Coalition for Content Provenance and Authenticity. Content credentials: C2PA technical specification. Technical report, Coalition for Content Provenance and Authenticity, 2025. Version 2.3. `https://spec.c2pa.org/specifications/specifications/2.3/specs/C2PA_Specification.html`.

[DCB25] Trisha Datta, Binyi Chen, and Dan Boneh. Veritas: Verifying image transformations at scale. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 4606–4623. IEEE, 2025.

[DEH25] Stefan Dziembowski, Shahriar Ebrahimi, and Parisa Hassanizadeh. VIMz: Private proofs of image manipulation using folding-based zkSNARKs. *Proceedings on Privacy Enhancing Technologies*, 2025.

[DHRR26] Rahul Dalal, Tamir Hemo, Eugene Rabinovich, and Ron Rothblum. VEIL: Lightweight zero-knowledge for hash-based multilinear proof systems. Cryptology ePrint Archive, Paper 2026/683, 2026.

[ESS+19] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *Applied Cryptography and Network Security – ACNS 2019*, volume 11464 of *Lecture Notes in Computer Science*, pages 67–88. Springer, 2019.

[Fed24] Federal Communications Commission. FCC proposes $6 million fine for deepfake robocalls around NH primary. Federal Communications Commission, May 2024.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology – CRYPTO '86*, pages 186–194. Springer, 1987.

[GKR+21] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. Poseidon: A new hash function for zero-knowledge proof systems. In *30th USENIX Security Symposium*, pages 519–535, 2021.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304. ACM, 1985.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988.

[GMS26] Garrett Greiner, Toshi Mowery, and Pratik Soni. Hyperveritas: Verifying image transformations at scale on boolean hypercubes. *Proceedings on Privacy Enhancing Technologies*, 2026(2):58–75, 2026.

[HHS+16] Guang Hua, Jiwu Huang, Yun Q. Shi, Jonathan Goh, and Vrizlynn L. L. Thing. Twenty years of digital audio watermarking: A comprehensive review. *Signal Processing*, 128:222–242, 2016.

[HLW+21] Guang Hua, Han Liao, Qingyi Wang, Haijian Zhang, and Dengpan Ye. Detection of electric network frequency in audio recordings: From theory to practical detectors. *IEEE Transactions on Information Forensics and Security*, 16:236–248, 2021.

[Int23] International Telecommunication Union. File format for the exchange of audio programme materials with metadata on information technology media. Technical Report Recommendation ITU-R BS.1352-4, ITU Radiocommunication Sector, 2023.

[KHSS22] Daniel Kang, Tatsunori Hashimoto, Ion Stoica, and Yi Sun. ZK-IMG: Attested images via zero-knowledge proofs to fight disinformation. *arXiv preprint arXiv:2211.04775*, 2022.

[KM20] Dusan Klinec and Vashek Matyas. Privacy-friendly Monero transaction signing on a hardware wallet. In *ICT Systems Security and Privacy Protection – 35th IFIP TC 11 International Conference, SEC 2020*, volume 580 of *IFIP Advances in Information and Communication Technology*, pages 338–351. Springer, 2020.

[KWJ23] Zahra Khanjani, Gabrielle Watson, and Vandana P. Janeja. Audio deepfakes: A survey. *Frontiers in Big Data*, 5:1001063, 2023.

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[LKYL22] Jaemin Lim, Kiyeon Kim, Hyunwoo Yu, and Suk-Bok Lee. Overo: Sharing private audio recordings. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1933–1946. ACM, 2022.

[MT14] Lindsay Miller and Jessica Toliver. Implementing a body-worn camera program: Recommendations and lessons learned. Technical report, Office of Community Oriented Policing Services, U.S. Department of Justice, 2014.

[MVVZ25] Pierpaolo Della Monica, Ivan Visconti, Andrea Vitaletti, and Marco Zecchini. Trust nobody: Privacy-preserving proofs for edited photos with your laptop. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 4624–4642. IEEE, 2025.

[NT16] Assa Naveh and Eran Tromer. Photoproof: Cryptographic image authentication for any set of permissible transformations. In *2016 IEEE Symposium on Security and Privacy*, pages 255–271. IEEE, 2016.

[ÖSMA05] Hamza Özer, Bülent Sankur, Nasir Memon, and Emin Anarım. Perceptual audio hashing functions. *EURASIP Journal on Advances in Signal Processing*, 2005(12):658950, 2005.

[Ped92] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO '91*, pages 129–140. Springer, 1992.

[Pol21] Polygon Zero. Plonky2: A SNARK implementation based on PLONK and FRI. GitHub repository, `https://github.com/0xPolygonZero/plonky2`, 2021. Version 0.2.2 used in the prototype.

[Pol23] Polygon Zero. ECDSA gadgets for plonky2. GitHub repository, `https://github.com/0xPolygonZero/plonky2-ecdsa`, 2023.

[PZL12] Xunyu Pan, Xing Zhang, and Siwei Lyu. Detecting splicing in digital audios using local noise level estimation. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1841–1844. IEEE, 2012.

[RL18] Diego Renza and Camilo Lemus. Authenticity verification of audio signals based on fragile watermarking for audio forensics. *Expert Systems with Applications*, 91:211–222, 2018.

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Fast Software Encryption – FSE 2004*, pages 371–388. Springer, 2004.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *Advances in Cryptology – ASIACRYPT 2001*, pages 552–565. Springer, 2001.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Set20] Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In *Advances in Cryptology – CRYPTO 2020, Part III*, pages 704–737. Springer, 2020.

[Suc26] Succinct Labs. SLOP-VEIL: Reference implementation of the VEIL compiler and stacked BaseFold. Source code in the SP1 repository, `https://github.com/succinctlabs/sp1`, 2026. Prototype adapted from commit f998ff15a7ff6d3293194c7dbbf1235c4f459768.

[Swe24] Ali Swenson. Tests find AI tools readily create election lies from the voices of well-known political leaders. Associated Press, May 2024.

[YWT+21] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Hector Delgado. ASVspoof 2021: Accelerating progress in spoofed and deepfake speech detection. In *Proceedings of the 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 1–5, 2021.

[ZCF23] Hadas Zeilberger, Binyi Chen, and Ben Fisch. Basefold: Efficient field-agnostic polynomial commitment schemes from foldable codes. *Cryptology ePrint Archive*, 2023. Report 2023/1705.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79*, pages 216–226. Springer, 1979.

[ZYO+25] Chengru Zhang, Xiao Yang, David Oswald, Mark Ryan, and Philipp Jovanovic. Eva: Efficient privacy-preserving proof of authenticity for lossily encoded videos. In *2025 IEEE Symposium on Security and Privacy*, pages 4643–4662. IEEE, 2025.