Anonymous C2PA: Feasibility of Large-Scale Anonymous Image Provenance

Kelsey Merrill Sarah Scheffler
Carnegie Mellon University

Abstract

High-quality AI-generated images are easy to create and near-indistinguishable from real photos, heightening the need to reliably establish image authenticity. The Coalition for Content Provenance and Authenticity (C2PA) has developed a technical standard under which cameras and editing software cryptographically sign the images they create, along with metadata and edit history. The standard is widely deployed in products by Google, Sony, Adobe, and others. However, C2PA offers no anonymity for signers. Signed images can be linked to a specific device or person, creating serious surveillance risks. C2PA also lacks formal security definitions, making it difficult for other systems to build on it with provable guarantees.

We formalize C2PA v2.5, and then we derive and formalize security and privacy properties desirable for an ideal anonymous image provenance system. Our primary building block for formalization is ring signatures, which let users sign on behalf of a group while staying anonymous within it. We give a new construction of shrinkable ring signatures that supports arbitrary subrings; this can be used to revoke compromised keys by "shrinking" these keys out of existing signatures. Prior work supported only a small, fixed set of subrings.

In our formalization, we find that several desiderata for an anonymous image provenance system are fundamentally incompatible. In particular, we prove that shrinking ring signatures requires work linear in the number of signatures shrunk, and in some settings, linear-sized public evidence; both are impractical at a global scale. We then explore the resulting trade-off space and analyze candidate systems making different choices among the strength of authenticity guarantees, network access requirements, and post-quantum security. Our work answers calls for a more private image provenance system and identifies several paths toward practical solutions.

1 Introduction

The authenticity of online images has been difficult to determine since the emergence of photo editing software. However, generative AI has intensified the problem considerably because "fake" images are now easy for laypeople to generate. To combat this issue, the Coalition for Content Provenance and Authenticity (C2PA) has put forward a technical standard for cryptographically verified image provenance [1]; the standard is also called C2PA. Under the standard, "generators" (cameras, generative AI models, and photo editing software) cryptographically sign images they create along with associated metadata. If the image is later edited by a second generator (e.g. image editing software), that generator signs the new image along with a hash of the old image, creating a provenance chain. If generators are trusted to record metadata and sign images appropriately, this creates a cryptographically verified provenance record.

C2PA is increasingly accepted and deployed in industry. Companies such as Adobe, Amazon, the BBC, Google, Meta, Microsoft, OpenAI, and Sony are members of the coalition's steering committee, and hundreds more are involved as members [2]. Google has implemented the spec

in their Pixel phones [3], Sony and Adobe have deployed it across their products [4, 5], Ring has deployed it in their doorbell cameras [6], the BBC and Reuters use it in the newsroom [7, 8], and users can check content credentials in Google and Adobe software [9, 10].

If deployed broadly, C2PA has the power to build trust online and help people avoid "AI slop" and disinformation. If a newsworthy but unflattering image or video has content credentials showing it came from a real camera, it is much harder to falsely claim the image is "fake news" or otherwise inauthentic. Social media sites, messaging apps, and other websites could check content credentials and add user-visible trust markers depending on the image's provenance. C2PA also brings potential benefits for digital artists who can use C2PA to make time-stamped claims of ownership. Provenance information provides artists with strong evidence that they were the first to create a particular artifact, demonstrating that similar works with later timestamps are likely copies or theft. Additionally, having content credentials specifying that the image is AI-generated helps avoid data poisoning in training [11] and allows AI providers to take credit for impressive image generation.

However, C2PA as it currently exists has dangerous privacy problems for generators. If C2PA is deployed according to the baseline specification, the same unique key is used to sign all images from a particular device. Therefore, the verification key links any C2PA-certified photos to each other, to that device, and often to a particular person. These keys can therefore be used for tracking and surveillance, as images and many kinds of metadata, including timestamps and GPS coordinates, can now be tied to an individual's device [12]. In more extreme cases, this tracking can lead to human rights abuses and threats to physical safety. For example, someone who photographs a politically sensitive event under an authoritarian regime could face physical danger if identified, and cryptographic evidence that their phone took the photos could undeniably link them to the images. The C2PA coalition acknowledges this danger in its Harms Modelling document [13], and C2PA documentation specifically states that the spec does not provide any anonymity guarantees [14]. Furthermore, downstream cryptographic systems meant to protect the anonymity of the signer (e.g., [15, 16]) can only provide this property as designed if the C2PA signer is anonymous.

Beyond the privacy issues, C2PA also has shortcomings in its presentation of its security goals. C2PA describes security goals in its documentation [14], but it does not formalize these properties nor prove that its specification meets them. This is an issue for other cryptographic systems that want to build on C2PA and use signed images in downstream functionalities [15, 17–19]. Without definitions of the security that C2PA claims to provide, creators of these systems cannot write their own security proofs based on C2PA. Additionally, abstract security definitions separate from the primitives that C2PA uses allow other systems to work with variants of C2PA that may be implemented differently but still provide the same security guarantees.

In this paper, we first provide a formalization of C2PA v2.5 that can be used in cryptographic proofs including C2PA as a module. Then, we consider various C2PA use cases to arrive at a list of privacy, security, and system properties desirable in an ideal version of C2PA. This idealized version would have C2PA's core integrity guarantee, support a global network of devices, offer anonymous signing, and support revocation in the case of compromised device keys.

Ring signatures are a natural tool towards formalizing this ideal C2PA system. In fact, we find the privacy and authenticity properties desirable in C2PA are neatly captured by ring signature security definitions [20, 21]. In particular, we find *shrinkable ring signatures* to be useful since they allow for "repairing" a signature in the case that a member of the anonymity set is later found to be malicious [22].

However, we find the existing literature on shrinkable ring signatures is under-developed for this use case. To our knowledge, the one existing construction only supports shrinking to a small, fixed set of subrings [22]. We give a simple generic construction for shrinkable ring signatures that

allows for shrinking to arbitrary subrings based on any claimable ring signature scheme (a scheme that allows signers to later reveal themselves as the real signer and *claim* signatures [21]). Given the transformation from a regular ring signature scheme to a claimable one in [21], this can be seen as a transform from a standard ring signature scheme.

In addition, we prove a lower bound on the amount of work required to shrink a list of signatures, and we find this bound conflicts with efficiency goals necessary for deployment on a global scale. This is one of several fundamental incompatibilities among the full desiderata for an ideal anonymous image provenance system; others are known in the literature, but outside the context of C2PA.

Finally, given the incompatibilities we find, we propose several candidate systems that make different trade-offs, and we analyze their viability, benefits, and drawbacks. We discuss practical ways forward for bringing anonymity to C2PA in spite of the impossibility of a version that meets all desired constraints.

Our main contributions are:

1. We formalize the stated security properties of the existing C2PA system, enabling C2PA to be combined with other cryptographic tools with formal guarantees (Section 4).

2. We derive formal cryptographic and system property desiderata from needs of different actors and systems using C2PA, including generator anonymity, key revocation, and generators' efficiency (Section 5).

3. We provide a novel generic construction for shrinkable ring signatures allowing for shrinking to arbitrary subrings (Section 6).

4. We demonstrate that the full set of properties one might desire from an anonymous C2PA system is unachievable in a single scheme in a scalable way. In particular, we prove that shrinking a list of signatures requires linear work in the number of signatures being shrunk (Section 7).

5. We propose candidate systems in the trade-off space and analyze their viability (Section 8).

2 Related Works

2.1 Private C2PA

2.1.1 Calls for privacy in C2PA

Existing literature highlights privacy risks in C2PA and calls for their remediation. Castellanos Rivadeneira [12] assesses surveillance risk posed by C2PA through analysis and review by expert stakeholders in journalism, tech, policy, human rights, and other fields. They identify several technical and regulatory threats to privacy in C2PA, including the issues around generator anonymity we discuss. They argue that the surveillance risk posed by C2PA for journalists and everyday users is particularly problematic due to the highly specific and targeted nature of the data revealed by C2PA. Golaszewski et al. [23, App. H] also find that information leakage during normal operation of C2PA could endanger photographers. Finally, some systems that treat C2PA as a root-of-trust assume or call for privacy in C2PA to enable the privacy of downstream functionality (see Section 2.3.1).

2.1.2 Systems proposing versions of privacy in C2PA

Fotos et al. and follow-on work by Buitrago [24, 25], Marinho et al. [26], Della Monica et al. [27], Loth et al. [28], and Awan et al. [29] all provide versions of private C2PA, but they consider different aspects of privacy and do not address generator anonymity. Instead, they address asset privacy, verifier privacy, and metadata privacy.

Google claims to implement C2PA in a way that is "private-by-design" in their Pixel phones [3]. Their implementation uses one-time-use certificates with an ephemeral key pair for each image, so images from the same user are not linked together by a common public key. Certificates to these ephemeral keys are issued by Google certificate authorities (CAs), which verify the authenticity of the device through Android Attestation [30]. Though Android Attestation does not directly reveal the identity of the user to Google, it does reveal the device's long-lived public key, and Google could link requests coming from the same device and correlate certificates. Google claims to prevent such linkage by segmenting its backend architecture [30], but there are no cryptographic guarantees preventing this.

Yu and Yao propose a similar version of private C2PA [31] with one-time-use certificates, but they provide more privacy by using anonymous credentials to authenticate with the CA; this makes requests for certificates by the same user unlinkable. However, the system only provides unlinkability for certificates issued one at a time, which is impractical for devices that may go offline for long periods, and the system is implemented with soon-to-be deprecated classical cryptography. The paper also does not consider the issue of revocation which is known to be difficult when using anonymous credentials [32].

Apple has proposed an alternative image provenance system called Apple Reference Image [33]. Instead of a manifest, the device captures a reference image "negative" straight from the camera's hardware, before any processing or editing takes place. The user may then later process that reference image by uploading it to Apple. This system provides generator privacy by signing all images under Apple's key after the user authenticates with Apple. Privacy from Apple is provided through Apple's Private Cloud Compute system [34]. Though the system does provide good generator privacy, it provides no system for redaction or privacy of the asset and no clear pathway for achieving those features.

2.2 C2PA security

Beyond its privacy issues, there is also a line of work demonstrating the security issues with C2PA. Golaszewski et al. [23] extract security goals from the C2PA specification, though the paper does not formalize them. This paper also proposes additional security goals for C2PA, and it highlights ways that current implementations and deployments of C2PA fall short of these goals. However, these additional properties do not consider generator anonymity.

Park et al. [35] and Jang [36] describe attacks on the C2PA system to get content credentials on inauthentic images. Though these security issues are important considerations for improvement to C2PA, we consider this out of scope. We identify and formalize C2PA's stated security goals in this work, but we do not consider systems-level improvements to C2PA security.

2.3 Systems using C2PA

2.3.1 C2PA as root-of-trust

In a line of works descending from PhotoProof [37], several works prove the correctness of specific image transformations, potentially with privacy constraints. Modern works in this space often

use C2PA signatures as a root of trust for this type of verifiable or privacy-preserving image processing, including VIMz [15], VerITAS [17], zkREAL [19], and their respective successors [38,39]. Additionally, VIMz and VerITAS propose changes to C2PA towards signer anonymity. VerITAS suggests using group signatures for C2PA, but signers are still traceable by the group manager in this setting. VIMz suggests storing verification keys in a Merkle tree and proving possession of a signature by a member of the tree, but this system is not efficient enough for practical use.

Other works utilizing C2PA as a root-of-trust include Zhang et al. [40], which proves correctness of image transforms for compressed images, and Zk-Cinema [18], a similar system for private and verifiable video processing. Ramos Chavez et al. [41] demonstrate how to use C2PA to authenticate streamed video content and prevent man-in-the-middle attacks. All of these works only make use of a signature on an asset from a trusted key (e.g. a C2PA-enabled camera) rather than utilizing the C2PA manifest.

2.3.2 Using the C2PA manifest

Balan et al. [11, 42] describe systems that make use of the C2PA manifest for AI training. [42] uses C2PA manifests to store opt-in or opt-out requests for AI training and additionally describes how manifests could be used to maintain payment or attribution information and enable automatic compensation for the image's use. [11] describes a framework for cryptographically verifying generative AI from end-to-end from data collection, to data refinement, training, inference, and evaluation. The paper makes use of C2PA manifests as a way of providing provenance for data ingested by AI training systems. These works demonstrate that C2PA's utility goes beyond just a signature over the image and its system for tracking metadata in the manifest is useful.

3 Preliminaries

We provide background on our main cryptographic primitive, ring signatures, and the C2PA system. Background on other primitives used can be found in Section 3.2.

3.1 Ring Signatures

Ring signatures, first introduced by Rivest et al. in [43], allow signing a message as one member of a group of signers called a "ring;" others can verify that *someone* in the ring signed the message, but cannot determine which ring member is the signer. Ring signatures do not require central coordination, and rings can be formed ad hoc at signing time without other signers knowing they were even included. A ring signature scheme has the following algorithms:

- $\text{KeyGen}(1^\lambda) \to (sk, vk)$ - generate a signing key / verification key pair.
- $\text{Sign}(sk, m, R) \to \sigma$ - sign message m with signing key sk on behalf of ring R (set of verification keys).
- $\text{Verify}(m, \sigma, R) \to \{0, 1\}$ - check the signature against the ring and the message.

Ring signatures have two security properties [20]:

- **Unforgeability** - no one without a secret key corresponding to one of the verification keys in the ring can generate a signature on behalf of that ring.
- **Anonymity** - given a signature on behalf of some ring, no one can determine which member of the ring generated that signature.

We are particularly interested in ring signatures with signature size sublinear in the ring size. A common pattern introduced in [44] for building such signatures collects verification keys in the ring in a cryptographic accumulator (see Section 3.2), and a signature is a zero-knowledge proof (ZKP) of membership. The message is tied to the signature by including the message in the ZKP's Fiat-Shamir hash.

Some ring signatures have additional properties described by Park and Sealfon [21] and Nguyen et al. [22] including *unrepudiability*, *claimability*, and *shrinkability*.

3.1.1 Unrepudiability

Unrepudiability is the property that non-signers *cannot* prove that they did not generate a particular signature, even if all secret keys are revealed. Park and Sealfon provide a formal definition [21, Defn. 14], and they show that unrepudiability of ring signatures is the same as anonymity with respect to full key exposure.

3.1.2 Claimability

Claimability is the ability of a signer to later cryptographically *claim* a given signature as having been generated by them. A claimable ring signature has the following additional algorithms as part of its API:

- $\text{Claim}(R, sk, \sigma) \to \zeta$ - takes as input a set of verification keys R, a signing key sk, and a ring signature σ and outputs a claim ζ.
- $\text{VerClaim}(R, vk, \sigma, \zeta) \to \{0, 1\}$ - takes as input a set of verification keys R, a specific verification key vk, a signature σ, and a claim ζ, and it outputs whether ζ is a valid claim for σ and vk.

These have the following properties [21, Defn. 17]:

1. *Honest signer can claim*: honestly generated signatures can always be claimed by the signer.
2. *Non-signers cannot claim*: there does not exist an efficient adversary that can generate a valid claim for a signature it did not generate.
3. *Non-framing*: there does not exist an efficient adversary that can generate a signature-claim pair that verifies under public key vk without knowledge of the corresponding secret key.

Park and Sealfon [21] prove that claimability and unrepudiability are incompatible because a failed run of $\text{VerClaim}(R, vk, \sigma, \text{Claim}(R, sk, \sigma))$ acts as a repudiation.

3.1.3 Shrinkability

Shrinkability is the ability for signers to later verifiably shrink the ring corresponding to a particular signature. It allows the signer to maintain anonymity within the smaller ring while asserting that some parties in the original ring are definitely not the signer. Nguyen et al. [22] introduced the concept of shrinkable ring signatures (SRS). An SRS has the following additional algorithms:

- $\text{Shrink}(sk, \sigma, R, R') \to \gamma$ given a set of verification keys R, a subset $R' \subset R$, signing key sk, and signature σ, outputs a shrink certificate γ demonstrating that σ was signed by someone in the subring R'.

- VerShrink$(\sigma, R, R', \gamma) \to \{0, 1\}$ takes as input a set of verification keys R, a subset of R called R', a signature σ, and a shrink certificate γ, and it outputs whether γ is valid for R' and σ.

In Section 6.1, we provide a different formalization for SRS security properties than Nguyen et al., and we discuss the reasoning for the differences in Appendix A. The security properties are informally:

1. **Shrink correctness** (honest signer can shrink): signers can always shrink honestly generated signatures to well-formed rings.
2. **Shrink unforgeability** (non-signers cannot shrink): no efficient adversary can generate a valid shrink certificate for a signature it did not generate.
3. **Shrink soundness** (no shrinking to invalid rings): no efficient adversary can generate a valid shrink certificate for $R' \not\subseteq R$ or $vk \notin R'$, even with access to the corresponding secret key sk.

Nguyen et al. [22] note that shrinkability implies claimability. If a signer may shrink to a ring of one (just the signer), that shrink evidence acts as a claim. Their work only provides a construction that allows shrinking to particular subrings, and shrinking to arbitrary subrings is left as an open question which we answer in Section 6.

3.2 Other Cryptography

This section provides background information on cryptographic primitives that we reference in passing.

NIZK-PoK. A non-interactive zero-knowledge proof-of-knowledge (NIZK-PoK) is a verifiable statement that the prover knows a private w corresponding to some public x such that $(x, w) \in \mathcal{R}$ for some binary relation $\mathcal{R}$ [45]. Unlike a plain NIZK which only requires *there exists* w, in a proof-of-knowledge, the prover must actually *know* w. They have the following security properties:

- **Correctness** - honestly generated proofs for $(x, w) \in \mathcal{R}$ should verify.
- **Knowledge soundness** - only a prover that knows a w such that $(x, w) \in \mathcal{R}$ can generate a valid proof.
- **Zero knowledge** - the verifier does not learn anything about w from the proof.

Accumulators. Accumulators are cryptographic tools for committing to a set [46]. They allow for proofs of (non-)membership in the set corresponding to a given commitment, sometimes without revealing the full contents of the set and/or the element whose membership is being proven. They may also allow for additions and deletions.

Blind signatures. Blind signatures, introduced by Chaum [47], allow a user to obtain a signature from another signer without revealing the message to the signer. The signer cannot distinguish requests to sign any two different messages. Similar to standard digital signature unforgeability, blind signatures have *one-more* unforgeability—anyone without the signer's secret key who receives q signatures cannot obtain $q + 1$ signatures on messages of their choice. Formal definitions can be found in [48].

Anonymous credentials. Anonymous credentials, introduced later by Chaum [49], are a cryptographic primitive allowing a user to prove that they have specific attributes without revealing any more information. An anonymous credential scheme has three parties: issuers, users, and verifiers. Issuers generate credentials and give them to users, who then present the credential to

verifiers. A single credential may have several *attributes*, and a user may present only a subset of the attributes to any particular verifier. The verifier learns nothing except that the user has the attributes presented; it learns nothing about the user's other attributes or the identity of the user. Additionally, credentials are unforgeable such that only the issuer may generate new credentials that verify. Formal definitions can be found in [50].

3.3 C2PA

In this section, we provide technical background on the C2PA system and the notation we use in our model of it.

3.3.1 Data

We define the following terms and notation for the C2PA system [1, Sec. 2].

- **asset** (I) - an image, plus metadata necessary to read/decode the image (e.g. file type).
- **assertion** (a): a statement about an asset, its metadata, or its provenance.
- **hard binding** (h): a special assertion specifying a cryptographic digest of the asset.
- **claim** ($C = \{\{a_j\}; h\}$): a set of assertions that must include exactly one hard binding.
- **signature** (σ): a signature over a claim.
- **manifest** ($M = (C, \sigma)$): a claim and signature pair.

Note that C2PA does not directly sign the asset (e.g. image). The spec requires that the claim include a hard binding over the asset, and then the whole claim is signed, effectively implementing hash-then-sign over the asset.

Provenance is provided through assertions called *ingredient assertions*. To indicate a previous asset was used in the creation of this asset, a generator adds an assertion including the manifest of the previous asset. Timestamps or other metadata can also be included through an assertion.

3.3.2 Parties

Generators are trusted devices with signing keys that create or modify assets and manifests. Generators include cameras, generative AI models, image editing software, or any other device or software that creates or modifies assets. They are trusted to be honest about the origins of assets they create and the edits they make to existing assets. *Validators* are devices that validate assets and manifests.

3.3.3 Algorithms

We formalize C2PA as a system with the following algorithms:

- $\text{KeyGen}(1^\lambda) \to (sk, vk)$. Generates a signing key pair. To be run for each generator device.
- $\text{HardBind}(m) \to h$. The function used to compute hard bindings.
- $\text{VerHardBind}(m, h) \to \{0, 1\}$. Checks whether h corresponds to message m.
- $\text{Sign}(sk, C) \to \sigma$. The function used to generate signatures over claims.
- $\text{Verify}(vk, I, M) \to \{0, 1\}$. The function run by validators to verify assets and their provenance.

4 Formal Definitions for C2PA v2.5

Two aspects of C2PA make it difficult to use for cryptography. First, C2PA's technical specification specifies a system without separation between its security guarantees, functionality, and construction. However, it is generally desirable to separate the "interface" of a cryptosystem (including its security guarantees) from any particular instantiation; multiple constructions may suffice to implement the same interface. It is useful to be able to replace components of C2PA for efficiency or compatibility reasons, as illustrated by cryptosystems in Section 2.3.1 (e.g., replacing the hash function SHA2 with the more zero-knowledge-proof-friendly Poseidon). Second, the definitions in the C2PA specification are not sufficiently formal for many cryptographic applications. In this section, we describe a formalization for the current version of C2PA. Our formalization enables integration of C2PA with other cryptosystems and allows extending C2PA with other features like anonymity (see Section 5).

We both formalize C2PA's security guarantees and analyze the "standard" construction of the C2PA system given in its specification to assess whether it meets those properties. Sections 4.3 and 4.4 formalize C2PA's claimed security properties of *integrity* and *non-repudiation* respectively. We show that C2PA's standard construction achieves integrity, but whether it achieves non-repudiation depends on the particular interpretation intended by C2PA's informal description of this property and the choice of signature scheme. This difference additionally emphasizes the need for formal security definitions and the usefulness of separating the security properties from the construction.

We begin by defining our abstraction of the C2PA system. Then, we discuss the security properties from C2PA's documentation. Finally, we provide formal definitions for the cryptographically-relevant security properties and show whether C2PA's construction does or does not achieve them.

4.1 Standard C2PA Construction

Constructing C2PA requires two primitives: a hash function that we treat as a random oracle, and a digital signature scheme. The allowed hash functions are SHA2-256, 384, and 512, and the allowed digital signature schemes are ECDSA, RSASSA-PSS, EdDSA, or ML-DSA [1, Sec. 12]. Call the selected hash function Hash and the selected digital signature scheme Sig. The algorithms in Section 3.3.3 are then implemented as follows:

- $\text{KeyGen}(1^\lambda) = \mathsf{Sig}.\,\text{KeyGen}(1^\lambda)$
- $\text{HardBind}(m) = \mathsf{Hash}(m)$
- $\text{VerHardBind}(m, h) = (\mathsf{Hash}(m) == h)$.
- $\text{Sign}(sk, C) = \mathsf{Sig}.\,\text{Sign}(sk, C)$
- $\text{Verify}(vk, I, M) =$
  $(C, \sigma) := M$
  $\{a_j\}, h := C$
  $\mathsf{Sig}.\,\text{Verify}(vk, C, \sigma) \wedge \text{VerHardBind}(I, h)$

Note that C2PA in full has more functionality than this. For example, C2PA validators also take care to validate timestamps, C2PA also includes a system for uniquely identifying assets, and C2PA specifies many details about data formats and manifest structure. We abstract C2PA in this way to allow us to analyze the parts relevant to cryptographic security, and we consider the rest to be out of scope. This formalization serves as a foundation on which other works can formalize other aspects of C2PA as necessary.

4.2 C2PA security considerations

C2PA lists the following security principles in their security considerations document [14, Sec. 3].

Integrity - "Attempts to compromise the integrity of C2PA provenance data associated with a respective asset will be detected upon validation. This includes all data stored within C2PA Manifests."

Availability - "An attacker may degrade availability of C2PA data, but C2PA should provide means to mitigate this impact." This largely refers to C2PA's *soft binding* mechanism for matching manifests with assets even if C2PA data has been stripped from the asset. This refers to statistical techniques we consider out of scope.

Confidentiality - The documentation lists confidentiality, but then it says, "C2PA does not specify confidentiality properties for assets, provenance model primitives, private address books, or cryptographic secrets. [...] C2PA does not specify guarantees of confidentiality of personal information." Thus, we do not formalize confidentiality properties.

Non-repudiation[1] - "Manifests are signed with a digital signature to maintain the integrity of the data. This signature also provides the identity of the signing individual, organization, software system or hardware device." It then discusses how validators may specify their own "trust anchors," or lists of CAs or generators they trust, and thus validators need to know which generator did the signing. In other words, the signer is identified and cannot deny they did the signing.

The two properties here that can provide cryptographic guarantees are **integrity** and **non-repudiation**.

4.3 Integrity

The C2PA integrity property says that validation should detect all attempts to compromise the integrity of provenance data. Validation failure means that either the signature over the claim does not verify, or the hard binding contained in the claim does not match the asset. Thus, integrity captures two notions: (1) An adversary cannot modify the asset without modifying the manifest, i.e., the adversary cannot switch out a different asset for a given manifest; (2) An adversary without access to a generator secret key cannot cause a new manifest to verify with any asset.

The first property is captured by the standard definition for commitment binding, and the second is captured by the standard definition for EUF-CMA secure digital signatures. We provide two definitions based on these standard definitions, and we say C2PA integrity requires satisfying both.

Definition 4.1 (C2PA hard binding)**.** We say that C2PA has the property of *hard binding* if, for all PPT adversaries $\mathcal{A}$:

$$\Pr\left[\begin{array}{l|l} b_0 = 1 \wedge & (sk, vk) \leftarrow \text{KeyGen}(1^\lambda) \\ b_1 = 1 \wedge & (I_0, I_1, M) \leftarrow \mathcal{A}(sk, vk) \\ I_0 \neq I_1 & b_0 \leftarrow \text{Verify}(vk, I_0, M) \\ & b_1 \leftarrow \text{Verify}(vk, I_1, M) \end{array}\right] \leq \text{negl}(\lambda)$$

[1]Note that C2PA's *non-repudiation* property is different than ring signature *unrepudiability* discussed in Section 3.1.1. Non-repudiation refers to *signers'* inability to deny having generated a signature; unrepudiability refers to *non-signers'* inability to deny having generated a signature.

Definition 4.2 (C2PA signature unforgeability)**.** Define oracle $\mathcal{O}_{\mathsf{Sign}}$ as $\mathcal{O}^{sk}_{\mathsf{Sign}}(C) \to \mathrm{Sign}(sk, C)$ such that $\mathcal{O}^{sk}_{\mathsf{Sign}}$ signs the given claim with respect to signing key sk. We say that C2PA has the property of *signature unforgeability* if, for all PPT adversaries $\mathcal{A}$ which make queries Q to $\mathcal{O}_{\mathsf{Sign}}$:

$$\Pr\left[\begin{array}{l|r} b = 1 \wedge & (sk, vk) \leftarrow \mathrm{KeyGen}(1^\lambda) \\ C \notin Q & (I, M) \leftarrow \mathcal{A}^{\mathcal{O}^{sk}_{\mathsf{Sign}}}(vk) \\ & (C, \sigma) := M \\ & b \leftarrow \mathrm{Verify}(vk, I, M) \end{array}\right] \leq \mathrm{negl}(\lambda)$$

Definition 4.3 (C2PA integrity)**.** We say that C2PA has the property of *integrity* if it satisfies both hard binding (Definition 4.1) and signature unforgeability (Definition 4.2).

Lemma 4.1. *Standard C2PA has the integrity property as defined in Definition 4.3 in the random oracle model.*

Proof sketch. The adversary must either find two assets that verify against the same manifest, or forge a new signature.

In order for two assets to verify according to the same manifest, both assets must have the same hard binding in order to pass the VerHardBind check in the verify function (Section 4.1). With SHA2 modeled as a random oracle, by definition, the adversary cannot predict the output of h nor find a collision without knowing the preimage, except with negligible probability. Thus, the adversary will only be able to achieve the first win condition with negligible probability.

By the EUF-CMA security of the digital signature schemes permitted under the C2PA spec, the adversary can only forge a signature on a new claim with at most negligible probability, so she has negligible probability of achieving the second win condition. Therefore, the probability that the adversary can achieve either win condition is negligible. □

Note that if SHA2 is not modeled as a random oracle, and no salt, key, or other randomization is used, this property does not hold; some adversary has hard-coded collisions for any deterministic hash function. Outside the random oracle model, the hard binding property could be achieved with a keyed hash function or a cryptographic commitment scheme.

4.4 Non-repudiation

C2PA's treatment of non-repudiation [14, Sec. 3] quoted in Section 4.2 is ambiguous based on whether the adversary can generate keys that the verifier will consider trustworthy. If the adversary cannot generate keys, then non-repudiation is covered by the integrity property above. If adversarially generated keys are considered, the stronger property of *exclusive ownership* is necessary. Exclusive ownership says that no adversary can generate another bitstring that works as a verification key for an existing signature [51]. If adversarially generated keys are considered and exclusive ownership is not provided, signers can repudiate by generating another verification key that verifies with that signature. We expect that in some instantiations, adversarial keys can be generated but not vouched for in a certificate hierarchy, making exclusive ownership unnecessary. However, should adversarially generated keys be considered in scope (e.g., if the adversary can take advantage of automated open key generation processes), we provide a definition based on *strong universal exclusive ownership* (S-UEO) [52].

Definition 4.4 (C2PA Non-repudiation)**.** Define oracle $\mathcal{O}_{\mathsf{Sign}}$ as in Definition 4.2. Define $Q = \{(C_i, \sigma_i)\}$ to be the set of queries made to $\mathcal{O}_{\mathsf{Sign}}$ paired with their responses. We say that C2PA has the property of *non-repudiation under adversarially generated keys* if, for all PPT adversaries $\mathcal{A}$ which generate set Q:

$$\Pr\left[\begin{array}{l|r} b = 1 \wedge & (sk, vk) \leftarrow \mathrm{KeyGen}(1^\lambda) \\ (C_0, \sigma) \in Q \wedge & (C_0, \sigma, I, C_1, vk') \leftarrow \mathcal{A}^{\mathcal{O}^{sk}_{\mathsf{Sign}}}(vk) \\ vk \neq vk' & M_1 := (C_1, \sigma) \\ & b \leftarrow \mathrm{Verify}(vk', I, M_1) \end{array}\right] \leq \mathrm{negl}(\lambda)$$

In order to win this game, the adversary must find an (asset, manifest) pair and a new public key that verifies against an oracle-generated signature using an honestly-generated public key.

C2PA's list of allowed signature algorithms [1, Sec. 12.3.1] includes two schemes satisfying S-UEO as defined in [52]: ML-DSA [53] and Ed25519 [52].

Lemma 4.2. *Standard C2PA does not have the non-repudiation property as defined in Definition 4.4.*

Proof sketch. To win, the adversary must find a claim (C_1, perhaps new, perhaps not) and a new verification key vk' that verify against an honestly-generated signature. Since the adversary is allowed to provide any asset I along with C_1, the adversary can trivially pass the hard binding check; the challenge for the adversary is to find a new claim and verification key pair that verify against the existing signature. This maps directly onto the definition of S-UEO.

C2PA supports signature schemes that do not satisfy S-UEO: [54] proves this for RSA, and [55] proves this for ECDSA. Thus, when these signature schemes are used, Definition 4.4 does not hold, and thus the C2PA standard as a whole does not have this notion of non-repudiation. □

5 Idealized Anonymous C2PA

In this section, we propose a list of desirable properties for an anonymous image provenance system. Existing literature about systems building on C2PA provides some insight into properties the community finds useful. First, all of the systems discussed in Section 2.3 rely on some authenticity or integrity property: they require that C2PA manifests and the signatures over those manifests are trustworthy. Additionally, many of the systems discussed in Section 2.3.1 consider the idea of an anonymous signer for C2PA, demonstrating interest in signer anonymity. We also analyze other C2PA use cases and deployment concerns to derive a final list of properties, which we then translate into formal cryptographic and system properties.

5.1 Illustrative C2PA use cases

In order to determine which security and privacy properties may be important to C2PA users, we considered several different C2PA use cases in different settings. We find the following examples particularly illustrative and motivating for our final list of desirable properties.

Journalist in authoritarian regime. Motivated by resolving the issues mentioned in [13], our first use case considers content provenance in authoritarian regimes. A journalist photographing politically sensitive or contested events in a hostile regime may face imprisonment or even death if their identity is linked to the photo. The adversary is powerful: it has vast computational resources, it may seize the camera, it may coerce organizations or companies into cooperating, and it may

question the journalist under duress. Of course, the regime may also do these things in the absence of a C2PA-like system, but we wish to avoid the C2PA scenario being *worse* for the journalist.

Because the photos are politically contested, the journalist needs *authenticity*, or proof to others that the photo is credible (e.g. from a real camera and not AI-generated). Because the journalist is under threat for taking the pictures, they need *anonymity* among a set of trusted generator devices. They also need *unrepudiability*, because otherwise the adversary could demand the journalist "prove their innocence" through a repudiation process which non-signers could complete but the actual signer could not. A journalist may also want *claimability* to take credit for the photo after the fact once it is safe to do so. Unfortunately, these last two goals conflict (see Section 7).

The same set of concerns could apply to a regular smartphone user in one of these places taking pictures of sensitive events in their own community.

Free society citizen. Consider someone in a place with substantial civil rights protections. Photographing a politically sensitive event, like a protest or police misconduct, carries lesser but real risks: retaliation from powerful people, social consequences from people in their private life that are upset by the political implications of the photo, or abuse from online trolls. Device seizure and questioning under duress are unlikely, so they have little use for unrepudiability, but they may still desire authenticity, anonymity, and claimability. The same holds for mundane photos: users don't want their camera to enable tracking or link anonymous online posts to their real-world identity, so anonymity is still a useful property even for trivial photos.

Digital artist. In contrast, consider an artist who wants their identity attached with timestamped proof that a particular device or software installation created the asset. This person does not have privacy concerns, and in fact, they are hindered by anonymity properties. They would prefer to have identifying signatures, or at least claimable ones.

Summary. From these use cases, we derive properties authenticity, anonymity, claimability, and unrepudiability.

5.2 C2PA global deployment concerns

In addition to security and privacy properties, an ideal image provenance system needs particular system properties to allow for deployment on a global scale.

Revocation and shrinkability. Like any system involving digital signatures, C2PA needs public key infrastructure (PKI) so validators know which keys are trustworthy. Devices are routinely lost, stolen, retired, or compromised, and the inability to revoke would mean that *one* compromise globally would potentially put the *entire* system into an untrustworthy state. Thus, a global-scale deployment must consider revocation as part of its PKI.

The traditional notion of revocation applies from a certain point in time going *forwards* from the revocation event; signatures generated before the revocation event remain valid, but future signatures by the revoked key will fail to verify. This is sufficient for settings like HTTPS where there is no delay between signing and verifying, and if a user visits a website with some certificate later determined to be malicious, there is no benefit to retroactive revocation [56].

In an image provenance system, signatures are expected to verify in perpetuity and be used for provenance long after the image was signed. If an attacker signs an image with a compromised key, and the compromise is only discovered after a delay, it is useful to revoke the key *backwards* so that signatures it produced before revocation also stop verifying.

Without anonymity, a revocation list is sufficient to accomplish this. However, with anonymous signatures, retroactive revocation would also invalidate honest signatures whose anonymity set includes the revoked key. This is a problem: anyone signing with a maximally large anonymity set, as we would expect many parties described in Section 5.1 to do, would have all their signatures

invalidated each time any signing key in the world is revoked—and since revocation events are likely to be common globally, most credentials would not be useful. An incident with Nikon's C2PA deployment in August 2025 underscores the relevance of this issue [57]. Nikon had to revoke all of its certificates and stop validating all signatures signed with those certificates. If those keys had been part of a global anonymity set, then *every image* signed during that time period would be affected, not just Nikon images.

Backwards revocation with anonymous signatures is captured by the additional property of shrinkability (Section 3.1): the ability to after-the-fact *shrink* the anonymity set to remove the now-revoked key.

Scalability, sublinearity, and concrete efficiency. Deploying image provenance around the world creates a system of billions of devices. The set of generator devices includes every phone or computer with a camera, every traditional camera, every instance of photo editing software, and every generative AI model. Furthermore, many of these devices will take several photos every day. Thus, scalability is of utmost importance. Given the global scale, it is likely impractical for any individual device to do linear work in either the number of devices in the system or the number of photos devices have signed. It may be possible for large service providers to do such linear work, but only sparingly.

Revocation events are also likely to be common in a global system, often warranting shrinking of many signatures. Thus, it is also likely impractical for any individual generator or validator to do linear work in the number of revocation events or shrinking events.

Offline tolerance. Many generator devices will have consistent network access, but it is very possible that a generator device may go offline for weeks or months at a time [3]. Consider a traveler taking photos in a remote area or a journalist taking photos in a war zone. These photos should still have trustworthy content credentials. An ideal image provenance system should not require network access at signing time, and it should be tolerant to devices going offline for long periods. Although our formalization of C2PA did not mention interaction between parties, some implementations of C2PA rely on connectivity to a service provider to be able to execute algorithms in Section 3.3.3.

Post-quantum security. Due to anticipated developments in quantum computing, the U.S. standards body, the National Institute of Standards and Technology, plans to deprecate all quantum-vulnerable algorithms by 2030 and disallow their use by 2035 [58]. A global deployment is slow to change. If such a deployment is to happen now, it should use post-quantum secure (PQS) cryptography.

5.3 Formalization of desirable properties

Accounting for all of the concerns discussed in the previous sections, we find ring signature security definitions naturally encapsulate the desirable security properties. In their basic form, they provide the authenticity and anonymity properties we need [20], and they can provide claimability, unrepudiability, and shrinkability with extensions [21, 22]. Also, ring signatures with a singleton ring allow for identifying signatures if desired. We provide the following list of properties for an ideal anonymous image provenance system with definitions based on ring signatures.

Authenticity. Formally, both C2PA's hard binding property (Definition 4.1) and ring signature unforgeability as defined in [21, Defn. 8]. Informally, no adversary not in the ring can sign on behalf of the ring, and as in C2PA v2.5, no adversary can swap a new asset for the same manifest.

Anonymity. Formally, ring signature anonymity with respect to adversarially chosen keys as defined in [21, Defn. 5]. Informally, no adversary can determine which ring member generated a given signature, even given the ability to add adversarially-generated public keys to the ring.

$\mathsf{Exp}^{\mathsf{corr}}(n, m, i)$

$\{(sk_j, vk_j)\}_{j=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$R := \{vk_j\}_{j=1}^{n}$
$\sigma \leftarrow \mathrm{Sign}(sk_i, m, R)$
return $\mathrm{Verify}(m, \sigma, R)$

$\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A}}(n)$

$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$(R^*, m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$
if $R^* \not\subseteq \{vk_i : i \in [n] \setminus \mathcal{Q}_{\mathsf{Corr}}\}$
  return 0
if $(\cdot, m^*, R^*) \in \mathcal{Q}_{\mathsf{Sign}}$ **then return** 0
return $\mathrm{Verify}(m^*, \sigma^*, R^*)$

$\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(n)$

$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$(R^*, m^*, i_0^*, i_1^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$
$R := R^* \cup \{vk_{i_0^*}, vk_{i_1^*}\}$
$b \leftarrow \{0, 1\}$
$\sigma \leftarrow \mathrm{Sign}(sk_{i_b^*}, m^*, R)$
$\mathcal{O}' := (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}_{\mathsf{Shrink}}^{(\{i_0^*, i_1^*\}, \{\sigma\})})$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}'}(z, \sigma)$
if $i_0^* \in \mathcal{Q}_{\mathsf{Corr}} \vee i_1^* \in \mathcal{Q}_{\mathsf{Corr}}$ **then return** 0
return $(b = b')$

Figure 1: Experiments for SRS correctness (left), unforgeability (middle), and anonymity (right). $\mathcal{O} = (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}_{\mathsf{Shrink}})$, and $\mathcal{Q}_{\mathsf{Corr}}$ and $\mathcal{Q}_{\mathsf{Sign}}$ are the sets of queries made to $\mathcal{O}_{\mathsf{Corr}}$ and $\mathcal{O}_{\mathsf{Sign}}$, respectively.

We select this notion of anonymity because it is the strongest in the literature that allows for claimability and shrinkability.

Claimability. Formally, claimability as defined for ring signatures in [21, Defn. 15]. Informally, a signer may later publish additional data linking a signature they generated to their public key. No adversary may claim signatures she did not sign or generate claims for other signers.

Unrepudiability. Formally, ring signature anonymity with respect to full key exposure as defined in [21, Defn. 6]. Informally, even if all secret keys are revealed, no adversary can determine which ring member generated a given signature. This is the same as signer/non-signer indistinguishability, and thus the same as unrepudiability [21].

Shrinkability. Formally, ring signature shrinkability as defined in Section 6.1. Informally, a signer may later publish additional evidence removing public keys from the original signing ring. No adversary may shrink signatures generated by a secret key she does not have, and no signer may shrink to an invalid ring (i.e., a ring not containing the signer or that is not a subset of the original ring).

Revocation. If a particular generator is found to be malicious or compromised at some point in time, its signatures can be prevented from verifying for all time after that point.

Sublinearity. No linear work in the number of generators, linear work in the total number of photos generators have created, or linear work in the number of revocation events should be done by individuals. Big service providers may be capable of handling linear work in limited circumstances, but no other party should.

Concretely efficient. Any online operations during signing and verifying should be efficient enough to be practical for everyday use, and any additional cryptographic data (keys, signatures, credentials, etc.) should be small enough to be practical to store on generator devices.

Offline tolerant. The system should be resilient to devices remaining offline for weeks or months at time while still allowing for trustworthy provenance information to be generated for photos captured by these devices in that time.

Post-quantum secure. The system should be plausibly post-quantum secure.

$\mathsf{Exp}^{\mathsf{sh\text{-}corr}}(n, m, n', s)$	$\mathsf{Exp}^{\mathsf{sh\text{-}unf}}_{\mathcal{A}}(n)$	$\mathsf{Exp}^{\mathsf{sh\text{-}sound}}_{\mathcal{A}}(n)$
$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \text{KeyGen}(1^\lambda)$	$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \text{KeyGen}(1^\lambda)$	$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \text{KeyGen}(1^\lambda)$
$R := \{vk_i\}_{i=1}^{n}$	$(R^*, m^*, i^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$	$(R^*, m^*, i^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$
$R' := \{vk_i\}_{i=1}^{n'}$	$R := R^* \cup \{vk_{i^*}\}$	$R := R^* \cup \{vk_{i^*}\}$
$\sigma \leftarrow \text{Sign}(sk_s, m, R)$	$\sigma \leftarrow \text{Sign}(sk_{i^*}, m^*, R)$	$\sigma \leftarrow \text{Sign}(sk_{i^*}, m^*, R)$
$\gamma \leftarrow \text{Shrink}(sk_s, \sigma, R, R')$	$(\tilde{R}^*, \gamma^*) \leftarrow \mathcal{A}_2^{\mathcal{O}}(z, \sigma)$	$(\tilde{R}^*, \gamma^*) \leftarrow \mathcal{A}_2^{\mathcal{O}}(z, \sigma)$
return $\text{VerShrink}(\sigma, R, R', \gamma)$	**if** $i^* \in \mathcal{Q}_{\mathsf{Corr}}$ **then return** 0	**if** $\tilde{R}^* \subseteq R \wedge vk_{i^*} \in \tilde{R}^*$ **then return** 0
	if $(i^*, \sigma, \cdot, \tilde{R}^*) \in \mathcal{Q}_{\mathsf{Shrink}}$ **then return** 0	**return** $\text{VerShrink}(\sigma, R, \tilde{R}^*, \gamma^*)$
	return $\text{VerShrink}(\sigma, R, \tilde{R}^*, \gamma^*)$	

Figure 2: Experiments for SRS shrink correctness (left), shrink unforgeability (middle), and shrink soundness (right). $\mathcal{O} = (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}_{\mathsf{Shrink}})$, and $\mathcal{Q}_{\mathsf{Corr}}$ and $\mathcal{Q}_{\mathsf{Shrink}}$ are the sets of queries made to $\mathcal{O}_{\mathsf{Corr}}$ and $\mathcal{O}_{\mathsf{Shrink}}$, respectively.

6 Improved Definitions and Construction for Shrinkable Ring Signatures

As noted in Section 3.1, the existing literature on shrinkable ring signature schemes only allows for shrinking to specific non-arbitrary subrings [22]. In order to implement revocation as described in Section 5.2, shrinking to arbitrary subrings is necessary.

We present a generic construction for such a scheme from a claimable ring signature scheme and a NIZK-PoK (see Section 3.2). Park and Sealfon [21] present a generic construction for claimable ring signatures from a ring signature, a standard digital signature, a PRF, and a commitment scheme, so this can be thought of as a generic construction for shrinkable ring signatures from those primitives.

We begin by providing formal definitions for the security properties of a shrinkable ring signature scheme. Then, we provide our construction and discuss its security.

6.1 Shrinkable ring signature security properties

In this section, we provide formalizations of the security properties for a shrinkable ring signature. The basic security definitions for ring signatures (correctness, anonymity, and unforgeability) need to be modified slightly in the SRS setting to ensure that these properties still hold in the presence of shrinking. Then, we provide definitions for the security properties of shrinking discussed in Section 3.1.3.

Define the following oracles for use in these definitions:

- $\mathcal{O}_{\mathsf{Sign}}(i, m, R) \rightarrow \text{Sign}(sk_i, m, R)$
- $\mathcal{O}_{\mathsf{Corr}}(i) \rightarrow sk_i$
- $\mathcal{O}^{P,S}_{\mathsf{Shrink}}(i, \sigma, R, R')$
  1. If $\sigma \in S$ and $\{vk_p : p \in P\} \not\subseteq R' \rightarrow \perp$
  2. If $\sigma \in S$ and $\{vk_p : p \in P\} \subseteq R' \rightarrow \text{Shrink}(sk_{i_\sigma}, \sigma, R, R')$ such that i_σ is the true signer of σ
  3. If $\sigma \notin S \rightarrow \text{Shrink}(sk_i, \sigma, R, R')$

Definitions reference experiments defined in Figure 1 and Figure 2.

Definition 6.1 (SRS Correctness)**.** For all $n = \text{poly}(\lambda)$, all m such that $|m| = \text{poly}(\lambda)$, and all $i \in [n]$:

$$\Pr[\mathsf{Exp}^{\mathsf{corr}}(n, m, i) = 1] \geq 1 - \text{negl}(\lambda)$$

Definition 6.2 (SRS Unforgeability)**.** For all $n = \text{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{unf}}_{\mathcal{A}}(n) = 1] \leq \text{negl}(\lambda)$$

Definition 6.3 (SRS Anonymity)**.** For all $n = \text{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(n) = 1] \leq \frac{1}{2} + \text{negl}(\lambda)$$

Definition 6.4 (SRS Shrink Correctness)**.** For all $n = \text{poly}(\lambda)$, all m such that $|m| \leq \text{poly}(\lambda)$, all $n' \in [n]$, and all $s \in [n']$:

$$\Pr[\mathsf{Exp}^{\mathsf{sh\text{-}corr}}(n, m, n', s) = 1] \geq 1 - \text{negl}(\lambda)$$

Note that defining R' in $\mathsf{Exp}^{\mathsf{sh\text{-}corr}}$ as the first n' keys is without loss of generality since keys can be re-ordered or re-indexed.

Definition 6.5 (SRS Shrink Unforgeability)**.** For all $n = \text{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{sh\text{-}unf}}_{\mathcal{A}}(n) = 1] \leq \text{negl}(\lambda)$$

Definition 6.6 (SRS Shrink Soundness)**.** For all $n = \text{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{sh\text{-}sound}}_{\mathcal{A}}(n) = 1] \leq \text{negl}(\lambda)$$

6.2 Construction for shrinkable ring signatures

Recall the APIs for claimable and shrinkable ring signatures presented in Section 3.1. Given we have a claimable ring signature scheme as defined there, we can convert this into a shrinkable ring signature scheme with a NIZK-PoK. Suppose we have signature σ over ring R, and the signer wants to shrink ring R to ring $R' \subset R$. The signer can do so given a ZK-PoK for the following relation $\mathcal{R}$.

$$\mathcal{R} := \left\{ \begin{array}{l} ((\sigma, R, R'), (vk, \zeta)) : \\ \qquad \text{VerClaim}(R, vk, \sigma, \zeta) = 1 \wedge vk \in R' \end{array} \right\}$$

In other words, to shrink, the signer proves, "I know claim evidence ζ and a public key vk such that ζ, vk verify for signature σ, and also vk is in the subring R'." The proof serves as the shrink evidence. To verify the shrink evidence, VerShrink checks the proof and checks that $R' \subseteq R$ (this can be done outside the proof since R, R' are both public).

Since $\mathcal{R} \in \mathsf{NP}$, this transformation can be instantiated efficiently with any zk-SNARK.

Security. Recall the security properties for shrinkable ring signatures and claimable ring signatures presented in Section 3.1. To show this construction is a shrinkable ring signature (SRS) scheme, we must show it has the three shrinkability properties, and also has anonymity and unforgeability.

This scheme achieves shrink correctness (Definition 6.4) by the first property of a CRS scheme (honest signers can claim) and the correctness of the ZK-PoK.

Shrink unforgeability (Definition 6.5) is achieved by the second property of a CRS scheme (non-signers cannot claim) and the knowledge soundness property of the NIZK-PoK. By knowledge soundness, in order to generate the ZK-PoK needed to shrink, one must know a claim on the signature. However, by the properties of a CRS, only signers can generate a claim, so it must be that only the signer can generate the ZK-PoK, and thus only the signer can shrink. Note that it is important that the prover *know* the claim (i.e., knowledge soundness is important over regular soundness) because a claim trivially exists by the first property of a CRS scheme.

Shrink soundness (Definition 6.6) holds by the soundness property of the NIZK-PoK: vk must be in R' in order to generate a proof of $\mathcal{R}$. Additionally, the verifier checks directly that $R' \subset R$, ensuring R' is a valid subset.

Anonymity in R' still holds after shrinking due to the zero knowledge property of the NIZK-PoK. Unforgeability is unaffected by this transform, and it holds by the unforgeability property of a CRS scheme.

7 Impossibilities and Incompatibilities

In Section 5 we identified several desirable properties for an "idealized" version of C2PA with anonymity and other properties as desired from the literature and use cases. Unfortunately, as we will show in this section, there are fundamental incompatibilities between some of these properties; no system can achieve all of them simultaneously.

7.1 Incompatibility of claimability/shrinkability and unrepudiability from literature

From the existing ring signature literature, we see that claimability and unrepudiability are incompatible [21]. Failure to produce a claim that verifies is a repudiation, so these two cannot exist simultaneously. Similarly, since shrinkability implies claimability [22] (shrinking to the singleton ring containing the signer is a claim), then we can conclude that shrinkability and unrepudiability are similarly incompatible.

7.2 Shrinking requires linear work

We also find incompatibility between shrinkability and the sublinearity property we specify in Section 5.3. As we discussed in Section 5.2, we want to avoid linear work in the number of revocation or shrinking events. One may try to mitigate this by grouping such events together; instead of every generator taking action to remedy affected signatures as soon as a revocation happens, perhaps they handle events in groups at some acceptable fixed time interval. However, there is a clear trade-off between the length of this time period and the number of signatures that need to be shrunk in each group: the longer the time between shrinking and thus the fewer times revocation events need to be handled, the more signatures that need to be handled at each shrinking.

In an attempt to resolve this tension, we investigate batch shrinking. Perhaps it is more efficient to shrink many signatures at once than to shrink each one individually. We refer to such a scheme as a "batch shrinkable ring signature" (BSRS). A BSRS has all of the properties of an SRS, but it also has two additional algorithms:

$\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(n)$

$\{(sk_i, vk_i)\}_{i=1}^n \leftarrow \text{KeyGen}(1^\lambda)$

$(R^*, m^*, i_0^*, i_1^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^n)$

$R := R^* \cup \{vk_{i_0^*}, vk_{i_1^*}\}$

$b \leftarrow \{0,1\}$

$\sigma \leftarrow \text{Sign}(sk_{i_b^*}, m^*, R)$

$P := \{i_0^*, i_1^*\};\ S := \{\sigma\}$

$\mathcal{O}' := (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}^{(P,S)}_{\mathsf{Shrink}}, \mathcal{O}^{P,S}_{\mathsf{BShrink}})$

$b' \leftarrow \mathcal{A}_2^{\mathcal{O}'}(z, \sigma)$

if $i_0^* \in \mathcal{Q}_{\mathsf{Corr}} \vee i_1^* \in \mathcal{Q}_{\mathsf{Corr}}$ **then return** 0

return $(b = b')$

Figure 3: The BSRS anonymity experiment. $\mathcal{O} = (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}_{\mathsf{Shrink}}, \mathcal{O}_{\mathsf{BShrink}})$, and $\mathcal{Q}_{\mathsf{Corr}}$ is the set of queries made to $\mathcal{O}_{\mathsf{Corr}}$.

- $\text{BatchShrink}(\{sk_i\}, \Sigma, R, R') \to \Gamma$. Generates shrink evidence shrinking all signatures in Σ from ring R to subring R'.
- $\text{VerBatchShrink}(\Sigma, R, R', \Gamma) \to \{0,1\}$. Verifies that shrink evidence Γ demonstrates that *all* signatures in Σ are shrunk from R to R'.

Security definitions for a BSRS are provided below. Such a scheme has all of the properties of a regular SRS, but the adversary also has access to a batch shrinking oracle in addition to its other oracles. The batch shrinking oracle is defined as follows:

$\mathcal{O}^{P,S}_{\mathsf{BShrink}}(T, R, R')$

foreach $(i_t, \sigma_t) \in T$ **do**

  if $\sigma_t \in S$ **then**

    if $\{vk_p : p \in P\} \not\subseteq R'$ **then return** $\perp$

    $i_t :=$ true signer of σ_t

return $\text{BatchShrink}(\{sk_i\}_{(i,\cdot)\in T}, \{\sigma_t\}_{(\cdot,\sigma_t)\in T}, R, R')$

In the anonymity definition, the batch shrink oracle is punctured analogously to the shrink oracle, shown in Definition 7.1. Other SRS properties are trivially modified to include the $\mathcal{O}_{\mathsf{BShrink}}$, so we do not include them here. The anonymity definition requires the experiment defined in Figure 3.

Definition 7.1 (BSRS Anonymity)**.** For all $n = \text{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{anon}}_{\mathcal{A}}(n) = 1] \leq \frac{1}{2} + \text{negl}(\lambda)$$

Batch shrinking security properties are defined as follows. They require the experiments defined in Figure 4.

Definition 7.2 (BSRS BShrink Correctness)**.** For all $n = \text{poly}(\lambda)$, all $n' \in [n]$, all multisets $L = \{s_1, ..., s_{|L|}\} \subseteq [n']$ such that $|L| = \mu = \text{poly}(\lambda)$, and all $\{m_\ell\}_{\ell\in[\mu]}$ such that $|m_\ell| \leq \text{poly}(\lambda)$:

$$\Pr[\mathsf{Exp}^{\mathsf{bsh\text{-}corr}}(n, n', L, \{m_\ell\}) = 1] \geq 1 - \text{negl}(\lambda)$$

$\mathsf{Exp}^{\mathsf{bsh\text{-}corr}}(n, n', L, \{m_\ell\}_{\ell\in[\mu]})$

$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$R := \{vk_i\}_{i=1}^{n}$
$R' := \{vk_i\}_{i=1}^{n'}$
for $\ell \in [\mu]$ **do**
  $\sigma_\ell \leftarrow \mathrm{Sign}(sk_{s_\ell}, m_\ell, R)$
$\Sigma := \{\sigma_\ell\}_{\ell\in[\mu]}$
$\Gamma \leftarrow \mathrm{BatchShrink}(\{sk_{s_\ell}\}_{\ell\in[\mu]}, \Sigma, R, R')$
return $\mathrm{VerBatchShrink}(\Sigma, R, R', \Gamma)$

$\mathsf{Exp}^{\mathsf{bsh\text{-}unf}}_{\mathcal{A}}(n)$

$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$(R^*, m^*, i^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$
$R := R^* \cup \{vk_{i^*}\}$
$\sigma \leftarrow \mathrm{Sign}(sk_{i^*}, m^*, R)$
$(\Sigma^*, \tilde{R}^*, \Gamma^*) \leftarrow \mathcal{A}_2^{\mathcal{O}}(z, \sigma)$
if $\sigma \notin \Sigma^*$ **then return** 0
if $i^* \in \mathcal{Q}_{\mathsf{Corr}}$ **then return** 0
if $(i^*, \sigma, \cdot, \tilde{R}^*) \in \mathcal{Q}_{\mathsf{Shrink}}$ **then return** 0
if $\exists (T, \cdot, \tilde{R}^*) \in \mathcal{Q}_{\mathsf{BShrink}} : (i^*, \sigma) \in T$
  return 0
return $\mathrm{VerBatchShrink}(\Sigma^*, R, \tilde{R}^*, \Gamma^*)$

$\mathsf{Exp}^{\mathsf{bsh\text{-}sound}}_{\mathcal{A}}(n)$

$\{(sk_i, vk_i)\}_{i=1}^{n} \leftarrow \mathrm{KeyGen}(1^\lambda)$
$(R^*, m^*, i^*, z) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\{vk_i\}_{i=1}^{n})$
$R := R^* \cup \{vk_{i^*}\}$
$\sigma \leftarrow \mathrm{Sign}(sk_{i^*}, m^*, R)$
$(\Sigma^*, \tilde{R}^*, \Gamma^*) \leftarrow \mathcal{A}_2^{\mathcal{O}}(z, \sigma)$
if $\sigma \notin \Sigma^*$ **then return** 0
if $\tilde{R}^* \subseteq R \wedge vk_{i^*} \in \tilde{R}^*$ **then return** 0
return $\mathrm{VerBatchShrink}(\Sigma^*, R, \tilde{R}^*, \Gamma^*)$

Figure 4: Experiments for BSRS batch shrink correctness (left), unforgeability (middle), and soundness (right). $\mathcal{O} = (\mathcal{O}_{\mathsf{Sign}}, \mathcal{O}_{\mathsf{Corr}}, \mathcal{O}_{\mathsf{Shrink}}, \mathcal{O}_{\mathsf{BShrink}})$, and $\mathcal{Q}_{\mathsf{Corr}}$, $\mathcal{Q}_{\mathsf{Shrink}}$, and $\mathcal{Q}_{\mathsf{BShrink}}$ are the sets of queries made to $\mathcal{O}_{\mathsf{Corr}}$, $\mathcal{O}_{\mathsf{Shrink}}$, and $\mathcal{O}_{\mathsf{BShrink}}$, respectively.

Definition 7.3 (BSRS BShrink Unforgeability)**.** For all $n = \mathrm{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{bsh\text{-}unf}}_{\mathcal{A}}(n) = 1] \leq \mathrm{negl}(\lambda)$$

Definition 7.4 (BSRS BShrink Soundness)**.** For all $n = \mathrm{poly}(\lambda)$ and all PPT adversaries $\mathcal{A}$:

$$\Pr[\mathsf{Exp}^{\mathsf{bsh\text{-}sound}}_{\mathcal{A}}(n) = 1] \leq \mathrm{negl}(\lambda)$$

7.2.1 Minimum total shrinking work

In Theorem 7.1, we prove that any batch shrinker satisfying the security properties of a batch shrinkable ring signature must read at least one bit of each signature being shrunk with probability $1 - \mathrm{negl}(\lambda)$, thereby implying work linear in the number of signatures shrunk.

This theorem assumes that there is no "short" way to reference signatures, and the only way the batch shrinker knows what to shrink is by taking in a list of signatures. We believe this is reasonable. If one instead assumes some way to reference some known set of past signatures, these results no longer hold.

Theorem 7.1. *Assume a BSRS satisfying batch shrink correctness (Definition 7.2) and batch shrink unforgeability (Definition 7.3). For all $\mu = \mathrm{poly}(\lambda)$, any ordered list of signers $L = [s_1, ..., s_\mu]$ with honestly-generated keys, any ordered list of messages $M = [m_1, ..., m_\mu]$, and all R, R' such that $\{vk_i : i \in L\} \subseteq R, R'$ and $R' \subseteq R$, define $\Sigma := [\mathrm{Sign}(sk_{L[\ell]}, M[\ell], R)]_{\ell\in[\mu]}$. Any batch shrinker $\mathrm{BatchShrink}(\{sk_i\}, \Sigma, R, R')$ must read at least one bit of each entry of Σ with probability $\geq 1 - \mathrm{negl}(\lambda)$.*

Proof. Assume towards contradiction that there exists some position t in Σ that the batch shrinker reads with probability $1 - p_t$ such that p_t is non-negligible. We can then construct an adversary for batch shrink unforgeability that has non-negligible probability of winning as follows:

1. Request signatures over ring R from the signing oracle for (signer, message) $(L[\ell], M[\ell])_{\ell\in[\mu]}$. Call this list Σ.

2. Query the batch shrink oracle on Σ with ring R' to obtain Γ.

3. Output $i^* = L[t]$ as the challenge identity, arbitrary fresh challenge message m^*, and R as the challenge ring.

4. Receive σ_{chall}. Replace $\Sigma[t]$ with σ_{chall} to get Σ^*. Output (Σ^*, R', Γ).

Note that the freshness conditions for batch shrink unforgeability are satisfied: no signers are corrupted, no individual shrinker calls were made, and the batch shrinker was never called with σ_{chall} in the list of signatures.

If the batch shrinker never reads position t, then Γ is distributed identically to what the batch shrinker would produce for signature list Σ^*. Thus, by batch shrink correctness,

$$\Pr[\text{VerBatchShrink}(\Sigma^*, R, R', \Gamma) = 1] \geq p_t - \text{negl}(\lambda)$$

By assumption, p_t is non-negligible, and the adversary wins the game with non-negligible probability—a contradiction. Thus, there must not exist a position t with non-negligible p_t, and the batch shrinker must read at least one bit of each signature in the list with probability $\geq 1 - \text{negl}(\lambda)$. □

7.2.2 Minimum per-signer work assuming no key sharing

Given Theorem 7.1, one might consider delegating the shrinking work to a service provider with more computational capabilities. However, we prove that this also does not solve the problem. If we assume that signers keep their secret keys private, we prove that *each signer* must do work linear in the number of their signatures they shrink.

Note that signers may still work together; perhaps there is some MPC protocol that allows signers to keep their secret keys private and their signatures anonymous but produce some batch shrink evidence. However, with keys kept private, we know that each signer must do at least some work as part of this protocol—by shrink unforgeability, only the holder of the signer's secret key can shrink a given signature.

Like Theorem 7.1, this theorem also assumes that there does not exist a short way to reference batches of signatures.

Theorem 7.2. *Assume a BSRS satisfying batch shrink correctness (Definition 7.2) and batch shrink unforgeability (Definition 7.3). For any set of participants I with $|I| = \text{poly}(\lambda)$ and all (sk_i, vk_i) honestly generated, for each $i \in I$, let M_i be a message list such that $|M_i| = \mu_i = \text{poly}(\lambda)$. Let $\Sigma_i = [\text{Sign}(sk_i, M_i[\ell], R)]_\ell$ for all $i \in I$ such that each identity has signed exactly M_i, in order. Let Σ be the concatenation of all Σ_i's. For all $R' \subseteq R$ such that $vk_i \in R'$ $\forall i \in I$, in an honest execution of* BatchShrink *among I on (Σ, R, R'), for every $j \in I$ and every position t of Σ holding one of j's signatures, j's role must read at least one bit of $\Sigma[t]$ with probability at least $1 - \text{negl}(\lambda)$.*

Proof sketch. The intuition is similar to Theorem 7.1: at least one bit of each signature must be read during shrinking or an adversary can break batch shrink unforgeability. However, since each signer keeps their secret key private, each signer must read all of their own signatures also by batch shrink unforgeability.

The proof and construction of the adversary follow the same structure as Theorem 7.1 but modified to account for the fact that BatchShrink is now a protocol. □

7.2.3 Minimum batch shrink evidence length assuming no collaboration

If we assume that signers do not collaborate, we prove a stronger result that signers must additionally publish shrink evidence of size at least linear in the number of signatures shrunk, except with negligible probability. In other words, if all signers must shrink their own signatures, then signers do not gain anything over shrinking each signature individually.

Theorem 7.3. *Assume a BSRS satisfying ring signature anonymity (Definition 7.1), batch shrink correctness (Definition 7.2), and batch shrink unforgeability (Definition 7.3). Let signers j, k such that $j \neq k$ have honestly-generated key pairs and $vk_j, vk_k \in R' \subseteq R$. Then, $\forall \mu$ such that $\mu = \text{poly}(\lambda)$ and for all messages M such that $|M| = \mu$:*

$$\Pr[|\operatorname{BatchShrink}(sk, \Sigma, R, R')| \leq \mu \cdot c\log(\lambda)] \leq \text{negl}(\lambda)$$

Proof. First, note that batch shrink evidence for signature list Σ must have size at least $\omega(\log \lambda) > c \log \lambda$ by batch shrink unforgeability. Otherwise, an adversary could forge batch shrink evidence in polynomial time by simply brute force enumerating all possible evidence strings.

Then, we show that the size of the batch shrink evidence for a list of signatures must have the same distribution as if the signatures were shrunk individually. Intuitively, this must be the case because otherwise signatures become linkable. If there existed short shrink evidence for two signatures, and since by assumption signers do not collaborate, one would learn that these two signatures must have been shrunk by the same party. This linkability violates anonymity.

More formally, we use a hybrid argument. Let $X_i(M)$ be the size of the shrink evidence returned by the batch shrinker given signatures on the messages in M by signer i. Define $H_t = X_j(M[1..t]) + \sum_{\ell=t+1}^{\mu} X_k(M[\ell])$. We will show that $H_0 \approx_c H_\mu$. Assume towards contradiction that there exists some PPT distinguisher D that can distinguish between some pair of hybrids H_{t-1}, H_t with non-negligible probability. We can then construct an adversary for the anonymity game using this distinguisher:

1. Get signatures on $M[1..t-1]$ from identity j and $M[t+1..\mu]$ from identity k. Output j, k as challenge identities.

2. Upon receiving the challenge signature, append it to the list of j's signatures and query the batch shrink oracle. Then query the batch shrink oracle with singleton lists for all of k's signatures.

3. Give the total size of all shrink evidence to D, and return D's output.

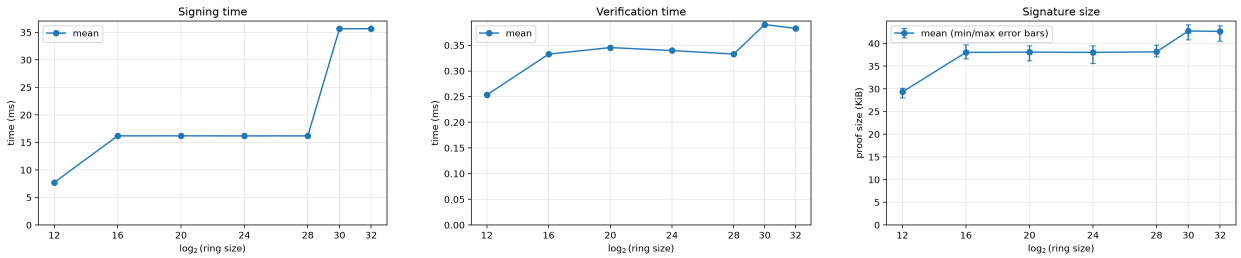
Define Z to be the total shrink evidence size. Note that if $b = 0$, the challenge signature is a signature from j over $M[t]$, so Z is distributed exactly as H_t. Likewise, if $b = 1$, Z is distributed exactly as H_{t-1}. Since the batch shrink oracle shrinks according to the true signer, it must shrink the challenge signature from k as a singleton, separate from the list of signatures from j. So, if D is able to distinguish these two hybrids with non-negligible probability, then the adversary can also guess the challenge bit with non-negligible probability.

Putting this together, we find that the size of shrink evidence must be greater than $\mu \cdot c \log \lambda$ except with negligible probability. Then, we conclude that shrink evidence must be $\Omega(\mu)$ except with negligible probability. □

8 Candidate Proposals

In this section, we review possible system designs with different trade-offs among the desiderata in Section 5.3. In addition to generators and validators, we consider a service provider party with large computation and storage. The trust model for the service provider depends on the scheme.

We first identify three small changes which can be applied under many architectures to get around the impossibilities presented in Section 7 concerning unrepudiability, required shrinking work, and sublinearity concerns.



(a) Online Signing Time (b) Online Verifying Time (c) Signature Size

Figure 5: Xue et al. [61] extended benchmarks run single-threaded on a Linux server machine with AMD EPYC 9124 processor, 32 CPU cores, and 604GB of RAM. Note that this scheme has a non-deterministic signature size, and size is averaged over 100 runs.

Achieving unrepudiability. Since unrepudiability and claimability/shrinkability are mathematically incompatible, these properties clearly cannot exist for the same signature. However, a signer can choose at signing time which properties they wish to have for a particular signature (i.e., which scheme they wish to use). As discussed in Section 6, claimability and shrinkability can both be implemented as a transform on top of a regular ring signature scheme. The transform works by including additional information along with the basic ring signature to later allow for claiming/shrinking. To sign in an unrepudiable way, the signer simply does not include this information.

Shrinking only as necessary. To circumvent issues regarding required work for shrinking discussed in Section 7, one could avoid or delay shrinking on each revocation event on most assets. Instead, only do the work of shrinking for particularly important photos that need up-to-date trust signals, or shrink only on an as-needed basis. This avoids the computational overhead of unnecessary shrinking for assets where a strong trust signal is currently unimportant, while still leaving the possibility that those assets *can* have their trust signal be fully updated in the future.

Small rings can do linear work. In some cases, an anonymity set consisting of all C2PA-enabled devices in the world is not necessary. For example, one could consider rings consisting only of the devices of journalists who work for a particular newspaper, or drivers for a particular package delivery company. For these small rings, the sublinearity concerns discussed in Section 5.2 do not apply. There exist highly efficient post-quantum ring signature constructions for rings up to thousands of members such that linear asymptotic scaling is not problematic [59, 60].

8.1 An accumulator ring signature scheme for full anonymity and authenticity

Applying the fixes for unrepudiability and required shrinking work above, we now aim to propose a scheme that achieves the rest of the desired properties from Section 5.3. In order to achieve sublinearity, the basis of this system is a ring signature scheme using an accumulator in the style of Dodis et al. [44]. The ring members are stored in an accumulator, and signers produce proofs of membership in the accumulator as part of their ring signatures.

The accumulator must be managed by a service provider, both due to its large size and because the provider is likely the only party aware of newly added devices. However, the service provider does not need to be trusted to do this faithfully. If all updates to the accumulator are recorded in a tamper-evident log, auditors can replay the log history and ensure that the published accumulator value is correct.

Xue et al. [61] present a PQS linkable ring signature construction based on Merkle trees that is concretely efficient for "online" operations (i.e., assuming the Merkle tree is already constructed) even for very large ring sizes. That paper only reports measurements for rings up to 2^{13}, but we extend their benchmarks to rings up to 2^{32} (see Figure 5).[2] We find that even for rings of this size (over four billion), online signing time is approximately 36 milliseconds and online verifying time is approximately 380 microseconds. Signature sizes are somewhat large at approximately 50KB, but still relatively practical given a modern smartphone photo is typically a few MB in size.

Though the construction of Xue et al. [61] is purposefully linkable, it can easily be made unlinkable (as desired for this setting) by simply ignoring or removing the extra linkability constraints and checks. This will be at least as efficient as the linkable scheme. Claimability and shrinkability can be provided using the transforms in [21] and Section 6.

Managing ring updates. The accumulator must be updated frequently as members are added and deleted from the ring. Xue et al. [61] suggest that their scheme is most suited to static rings due to the cost of updating the Merkle tree; in the worst case, this could require rebuilding the whole tree and become linear in the number of generators. Additionally, for a Merkle tree like in [61], all members need to update their membership proof for each update, which is infeasible. However, these issues can be mitigated.

To avoid needing to reconstruct the Merkle tree on each update, make the initial tree significantly larger than necessary and add dummy values in unused slots (which is still efficient even for very large rings). Then, adding a new key only requires replacing a dummy value and updating that path, which is $O(\log n)$ work for the service provider. Likewise, deleting a key is simply replacing that key with a dummy value and updating that path. The tree then only needs to be rebuilt very infrequently, when the ring grows even beyond a large overestimate.

One way to keep membership proofs updated is to update the accumulator in epochs and have devices contact the service provider non-anonymously to retrieve the updated accumulator value and their updated witness at some regular time once per epoch. It is important that devices reach out for updates on a regular schedule regardless of their signing activity; otherwise, signers could be deanonymized with timing correlations. Updating in epochs also streamlines auditing, as auditors only have to run their update check at a predetermined regular time.

Offline tolerance. This scheme can be modified to accommodate devices that go offline for significant periods by allowing devices to sign against the most recent accumulator view they had before going offline and then shrinking (if necessary) to the intersection of that accumulator and the new accumulator state upon regaining network access. However, this disrupts the regular epoch-based updating and thus provides more opportunity for timing correlation attacks.

Summary. This scheme is highly concretely efficient at signing/verifying time and supports full anonymity and authenticity, but it does allow for possible timing correlation attacks for devices that go offline for extended periods.

8.2 One-time certificates, weakened authenticity

Existing proposals for anonymous C2PA (Google [3], Yu and Yao [31]) use signing based on one-time-use certificates. Certificate-based signing can sometimes be desirable as it easily slots into legacy infrastructure. Both of these proposals go a long way towards the list of desirable properties in Section 5.3, and in the rest of this section, we will discuss modifications towards achieving more of these properties.

The high-level workflow of these schemes is as follows. Before signing time, a generator contacts the service provider for a one-time-use certificate associated with an ephemeral key pair. The

[2] `https://github.com/clasp-research/Xue-et-al-24-ring-sig`

device authenticates with the service provider in some way; Google uses Android Attestation [30], and [31] uses anonymous credentials. Provided the authentication passes, the service provider issues a certificate, which the device uses the next time it needs to sign an asset.

Weakened authenticity. Note that this workflow slightly weakens authenticity: the service provider must be trusted to faithfully authenticate C2PA-enabled devices and issue certificates appropriately. One possible mitigation is for the server to use remote attestation to convince clients they are actually running the expected code, similar to Signal's use of attestation to prove it runs the expected code [62].

Achieving full anonymity from the service provider. As discussed in Section 2.1.2, both of these schemes do not fully meet the anonymity property we provide in Section 5.3. In Google's system, certificates issued to the same device could be linkable by Google, thus making assets signed with those certificates linkable. In [31], they note that devices may want to request certificates in batches; this lowers the number of requests that need to be made to the service provider, and it allows for offline tolerance since devices may request many certificates before going offline and then use those over that time. However, they also note that certificates acquired in batches are linkable. To fix this, we suggest the service provider sign certificates with blind signatures. Thus, the service provider does not ever see the public keys in the certificates it signs, and even if many certificates are issued in a batch, it cannot link them together as belonging to the same generator.

Post-quantum security. As noted in Section 2.1.2, [31] uses classical cryptography. Both the anonymous credentials used in [31] and the blind signature addition we propose can be instantiated instead with PQS constructions. Existing PQS anonymous credential and blind signature schemes are near practical, if not already practical. [63] presents a blind signature scheme with signing times in the 10s of milliseconds and signature sizes about 10KB. [32] presents an anonymous credential scheme with presentation + verification time of about 500ms and credential size in the 10s of KB. More recent work reduces credential size further to single KBs [64].

Claimability and shrinkability. Claimability can be provided using a modified transform [21] by substituting the signature from the ephemeral key pair on the certificate in place of the ring signature. Shrinkability again uses the method from Section 6.

Revocation. Traditional, forwards revocation is difficult in this setting due to the general difficulties of revocation for anonymous credentials [32]. The standard technique in the anonymous credentials literature is signature revocation lists (SRLs), which allow for revoking the credential of someone without revealing their secret key. However, SRLs do not scale well, and they become impractical by only a few thousand revocations [65]. To circumvent this issue, credentials should be short-lived and issued in epochs so any revocation list can be cleared after each epoch.

Summary. The scheme presented in [31] can be modified to additionally achieve full anonymity from the service provider, post-quantum security, and practical revocation: use blind signatures over certificates, use PQS anonymous credential and blind signature schemes, keep credentials short-lived and re-issue in epochs. However, any scheme of this form does have a slightly weakened authenticity guarantee in that the service provider must be trusted to faithfully authenticate devices and issue certificates appropriately.

8.3 Non-post-quantum-secure cryptography

If we weaken the requirement for PQS cryptography, the scheme presented in Section 8.2 becomes concretely efficient. A recent measurement of anonymous credential schemes shows prover and verifier times on the order of 10s of milliseconds with credential sizes on the order of hundreds of bytes to single KBs [66]. Likewise, classical blind signature schemes have fast signing and verifying times in the single milliseconds or hundreds of microseconds, and signature sizes are 10s to 100s of

bytes [67].

Removing the requirement for PQS cryptography does *not* give efficiency gains for the accumulator-based ring signature scheme in Section 8.1. To our knowledge, state-of-the-art classical ring signature schemes are significantly slower than [61] and not efficient enough for practical deployment at rings of this size [68, 69].

9 Conclusion

C2PA is a system for cryptographically verified image provenance. It has enormous promise for combating misinformation, but it also has dangerous privacy problems. We formalize the existing C2PA system, and we derive and formalize properties desirable in an ideal large-scale anonymous image provenance system. Towards this construction, we present a new generic transform for a shrinkable ring signature scheme. We also identify incompatibilities between some of the desirable properties: some exist in the literature in different contexts, and we prove new lower bounds on the amount of work required to shrink a list of signatures. Finally, we present and analyze candidate solutions, and we find several practical paths forward for anonymous C2PA.

References

[1] Content Credentials : C2PA Technical Specification :: C2PA Specifications. [Online]. Available: https://spec.c2pa.org/specifications/specifications/2.5/specs/C2PA_Specification.html

[2] Coalition for Content Provenance and Authenticity. C2PA membership. [Online]. Available: https://c2pa.org/membership/

[3] E. Lynch and S. Hanna. How Pixel and Android are bringing a new level of trust to your images with C2PA Content Credentials. Google Online Security Blog. [Online]. Available: https://security.googleblog.com/2025/09/pixel-android-trusted-images-c2pa-content-credentials.html

[4] Sony electronics launches industry first video-compatible camera authenticity solution for news organizations and broadcasters - sony pro. [Online]. Available: https://pro.sony/ue_US/press/video-authenticity-launch

[5] A. Parsons. Content authenticity arrives for enterprises. Adobe for Business. [Online]. Available: https://business.adobe.com/blog/content-authenticity-arrives-for-enterprises

[6] S. Perez. Ring is adding a new content verification feature to videos. TechCrunch. [Online]. Available: https://techcrunch.com/2026/01/22/ring-is-adding-a-new-content-verification-feature-to-videos/

[7] C. Halford. Mark the good stuff: Content provenance and the fight against disinformation. BBC Research and Development. [Online]. Available: https://www.bbc.co.uk/rd/articles/2024-03-c2pa-verification-news-journalism-credentials

[8] A. Zammit. Sony and Reuters Demonstrate World-first* Near-live Newsroom Workflow to Protect Content Authenticity. Sony Europe. [Online]. Available: https://www.sony.eu/presscentre/sony-and-reuters-demonstrate-world-first-near-live-newsroom-workflow-to-protect-content-authenticity-9etv7j

[9] Content Credentials — Gemini Enterprise Agent Platform. Google Cloud Documentation. [Online]. Available: https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/content-credentials

[10] Content Credentials overview. [Online]. Available: https://helpx.adobe.com/content/help/en/creative-cloud/apps/adobe-content-authenticity/content-credentials/overview.html

[11] K. Balan, R. Learney, and T. Wood, "A Framework for Cryptographic Verifiability of End-to-End AI Pipelines," in *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, ser. IWSPA '25. Association for Computing Machinery, 2025, pp. 49–59. [Online]. Available: https://dl.acm.org/doi/10.1145/3716815.3729011

[12] J. Castellanos Rivadeneira. C2PA Content Credentials and the Surveillance Risk: Adversarial Scenarios and Governance Gaps in the Content Provenance Ecosystem. [Online]. Available: https://papers.ssrn.com/abstract=6986118

[13] C2PA Harms Modelling :: C2PA Specifications. [Online]. Available: https://spec.c2pa.org/specifications/specifications/2.4/security/Harms_Modelling.html

[14] C2PA Security Considerations :: C2PA Specifications. [Online]. Available: https://spec.c2pa.org/specifications/specifications/2.4/security/Security_Considerations.html

[15] S. Dziembowski, S. Ebrahimi, and P. Hassanizadeh, "VIMz: Private Proofs of Image Manipulation using Folding-based zkSNARKs," vol. 2025, no. 2, pp. 344–362, 2025. [Online]. Available: https://petsymposium.org/popets/2025/popets-2025-0065.php

[16] K. Merrill, A. Woo, W. Zheng, and S. Scheffler, "Zero knowledge barcode decoding with application to private online attribute verification," *IACR Cryptol. ePrint Arch.*, vol. 2026, p. 1567, 2026. [Online]. Available: https://eprint.iacr.org/2026/1567

[17] T. Datta, B. Chen, and D. Boneh, "VerITAS: Verifying Image Transformations at Scale," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 4606–4623. [Online]. Available: https://ieeexplore.ieee.org/document/11023367/

[18] A. Frolov, J. Guo, X. Zhao, T. Datta, D. Boneh, and I. Miers. (2026) Zk-Cinema: Proving Video Provenance in Zero Knowledge. [Online]. Available: https://eprint.iacr.org/2026/1598

[19] A. Koyama, K. Toyoda, M. Fujimoto, and T. H. Tran, "Zk-REAL: A Zero-Knowledge-Based Protocol for Repeated Image Edit Authenticity Proof With Lattice Hashing," vol. 7, pp. 443–454, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/11372060

[20] A. Bender, J. Katz, and R. Morselli, "Ring Signatures: Stronger Definitions, and Constructions Without Random Oracles," in *Theory of Cryptography*, S. Halevi and T. Rabin, Eds. Springer, 2006, pp. 60–79. [Online]. Available: https://link.springer.com/chapter/10.1007/11681878_4

[21] S. Park and A. Sealfon, "It Wasn't Me!" in *Advances in Cryptology – CRYPTO 2019*, A. Boldyreva and D. Micciancio, Eds. Springer International Publishing, 2019, pp. 159–190. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-26954-8_6

[22] T. Ngoc Nguyen, W. Susilo, D. Hoang Duong, F. Guo, K. Fukushima, and S. Kiyomoto, "Shrinkable Ring Signatures: It Wasn't Them!" in *Information Security Practice and Experience*, Z. Xia and J. Chen, Eds. Springer Nature, 2025, pp. 75–93. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-97-9053-1_5

[23] E. Golaszewski, N. Krawetz, A. T. Sherman, E. Zieglar, S. K. Matukumalli, R. Yus, C. L. Kegley, M. Barthel, W. Bowman, B. Barot, and K. Kullman. (2026) Verifying Provenance of Digital Media: Security Analysis of C2PA and its Implementation. [Online]. Available: https://eprint.iacr.org/2026/804

[24] N. Fotos and J. Delgado, "Towards Privacy-Enhancing Provenance Annotations for Images," in *2024 IEEE International Conference on Image Processing (ICIP)*, 2024, pp. 3785–3791. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10647277

[25] S. Buitrago Cervilla, "Privacy of provenance metadata." [Online]. Available: https://hdl.handle.net/2117/451934

[26] J. Marinho, E. Schiller, A. Debauge, and N. Foukia, "ZK-Disclosure: Privacy-Preserving Information Disclosure for Digital Evidence with C2PA and zk-SNARKs," in *2025 Annual Computer Security Applications Conference Workshops (ACSAC Workshops)*, 2025, pp. 309–315. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/11418029

[27] P. Della Monica, I. Visconti, A. Vitaletti, and M. Zecchini, "Trust Nobody: Privacy-Preserving Proofs for Edited Photos with Your Laptop," in *2025 IEEE Symposium on Security and Privacy (SP)*, 2025, pp. 4624–4642. [Online]. Available: https://ieeexplore.ieee.org/document/11023375

[28] A. Loth, D. C. Rosario, P. Ebinger, M. Kappes, and M.-O. Pahl. Origin Lens: A Privacy-First Mobile Framework for Cryptographic Image Provenance and AI Detection. arXiv.org. [Online]. Available: https://arxiv.org/abs/2602.03423v1

[29] M. Awan and J. Collomosse. Soft Redaction of Image Provenance via Zero-Knowledge Proofs. [Online]. Available: http://arxiv.org/abs/2608.07063

[30] Upgrading Android Attestation: Remote Provisioning. Android Developers Blog. [Online]. Available: https://android-developers.googleblog.com/2022/03/upgrading-android-attestation-remote.html

[31] L. Yu and Y. Yao. (2026) AIPA: Anonymous Image Provenance Authentication in Online Social Networks via Unlinkable Pseudonym Certificates and zk-SNARKs. [Online]. Available: https://eprint.iacr.org/2026/1914

[32] S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois, and O. Sanders, "Practical Post-Quantum Signatures for Privacy," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. ACM, pp. 1523–1537. [Online]. Available: https://dl.acm.org/doi/10.1145/3658644.3670297

[33] Apple Reference Image: A New Approach for Verified Photography. Apple Security Research. [Online]. Available: https://security.apple.com/blog/apple-reference-image/

[34] Private Cloud Compute: A new frontier for AI privacy in the cloud - Apple Security Research. Apple Security Research. [Online]. Available: https://security.apple.com/blog/private-cloud-compute/

[35] S. Park, A. Vilesov, J. Zhang, H. Khalili, Y. Tian, A. Kadambi, and N. Sehatbakhsh, "Chimera: Creating Digitally Signed Fake Photos by Fooling Image Recapture and Deepfake Detectors," 2025, pp. 4305–4324. [Online]. Available: https://www.usenix.org/conference/usenixsecurity25/presentation/park

[36] Y. Jang. Signing Right Away. [Online]. Available: http://arxiv.org/abs/2510.09656

[37] A. Naveh and E. Tromer, "Photoproof: Cryptographic image authentication for any set of permissible transformations," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 255–271.

[38] P. Mikołajczyk, P. Hassanizadeh, and S. Ebrahimi. (2025) Towards Trustless Provenance: A Privacy-Preserving Framework for On-chain Media Verification. [Online]. Available: https://eprint.iacr.org/2025/1024

[39] G. Greiner, T. Mowery, and P. Soni, "Hyperveritas: Verifying image transformations at scale on boolean hypercubes," *Proc. Priv. Enhancing Technol.*, vol. 2026, no. 2, pp. 58–75, 2026. [Online]. Available: https://doi.org/10.56553/popets-2026-0036

[40] Y. Zhang, M. Zhou, and B. Bünz. (2026) Verifiable Compression of Images. [Online]. Available: https://eprint.iacr.org/2026/1717

[41] R. Ramos Chavez, M. Raad, A. Goldberg, and A. Wagenaar, "When a Man-in-the-Middle tries crashing the live stream party, the C2PA bouncer steps in," in *Proceedings of the 5th Mile-High Video Conference*, ser. MHV '26. Association for Computing Machinery, pp. 27–33. [Online]. Available: https://dl.acm.org/doi/10.1145/3789239.3793271

[42] K. Balan, A. Gilbert, A. Black, S. Jenni, A. Parsons, and J. Collomosse, "DECORAIT - DECentralized Opt-in/out Registry for AI Training," in *Proceedings of the 20th ACM SIGGRAPH European Conference on Visual Media Production*, ser. CVMP '23. Association for Computing Machinery, pp. 1–10. [Online]. Available: https://dl.acm.org/doi/10.1145/3626495.3626506

[43] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in *Advances in Cryptology — ASIACRYPT 2001*, C. Boyd, Ed. Springer, 2001, pp. 552–565.

[44] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup, "Anonymous Identification in Ad Hoc Groups," in *Advances in Cryptology - EUROCRYPT 2004*, C. Cachin and J. L. Camenisch, Eds. Springer, 2004, pp. 609–626.

[45] A. De Santis and G. Persiano, "Zero-knowledge proofs of knowledge without interaction," in *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, 1992, pp. 427–436.

[46] F. Baldimtsi, R. Canetti, and S. Yakoubov, "Universally composable accumulators," in *Topics in Cryptology – CT-RSA 2020*, S. Jarecki, Ed. Cham: Springer International Publishing, 2020, pp. 638–666.

[47] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Boston, MA: Springer US, 1983, pp. 199–203.

[48] A. Juels, M. Luby, and R. Ostrovsky, "Security of blind digital signatures," in *Advances in Cryptology — CRYPTO '97*, B. S. Kaliski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 150–164.

[49] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[50] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology — EUROCRYPT 2001*, B. Pfitzmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 93–118.

[51] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, and C. Janson, "BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1696–1714. [Online]. Available: https://ieeexplore.ieee.org/document/9519420

[52] J. Brendel, C. Cremers, D. Jackson, and M. Zhao, "The Provable Security of Ed25519: Theory and Practice," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1659–1676. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9519456

[53] M. Meyer, P. Struck, and M. Weishäupl, "Exclusive Ownership of Fiat-Shamir Signatures: ML-DSA, SQIsign, LESS, and More," in *Advances in Cryptology – CRYPTO 2025*, Y. Tauman Kalai and S. F. Kamara, Eds. Springer Nature Switzerland, 2025, pp. 57–90.

[54] T. Pornin and J. P. Stern, "Digital Signatures Do Not Guarantee Exclusive Ownership," in *Applied Cryptography and Network Security*, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Springer, 2005, pp. 138–150.

[55] K. Emura, "On the BUFF security of ECDSA with key recovery," vol. 94, no. 9, p. 197. [Online]. Available: https://doi.org/10.1007/s10623-026-01942-7

[56] OpenSSH. OpenSSH: Post-quantum cryptography. [Online]. Available: https://www.openssh.org/pq.html

[57] Nikon Cameras C2PA Support: The Z6 III Vulnerability and Service Suspension. C2PA Viewer. [Online]. Available: https://c2paviewer.com/articles/nikon-cameras-c2pa

[58] A. Regenscheid, "Transition to Post-Quantum Cryptography Standards," p. NIST IR 8547 ipd, 2024. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2024/NIST.IR.8547.ipd.pdf

[59] X. Zhang, Z. Li, R. Steinfeld, R. K. Zhao, J. K. Liu, and T. H. Yuen. (2025) Fast Post-Quantum Ring Signature from Power Residue PRFs with QROM Security. [Online]. Available: https://eprint.iacr.org/2025/1841

[60] J. H.-y. Chiang, I. Damgård, W. R. Duro, S. Engan, S. Kolby, and P. Scholl, "Post-Quantum Threshold Ring Signature Applications from VOLE-in-the-Head," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '25. Association for Computing Machinery, pp. 4664–4678. [Online]. Available: https://dl.acm.org/doi/10.1145/3719027.3744854

[61] Y. Xue, X. Lu, M. H. Au, and C. Zhang, "Efficient Linkable Ring Signatures: New Framework and Post-quantum Instantiations," in *Computer Security – ESORICS 2024*, J. Garcia-Alfaro, R. Kozik, M. Choraś, and S. Katsikas, Eds. Springer Nature Switzerland, 2024, pp. 435–456. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-70903-6_22

[62] M. Marlinspike. Technology preview: Private contact discovery for Signal. Signal Messenger. [Online]. Available: https://signal.org/blog/private-contact-discovery/

[63] C. Baum, M. Beckmann, W. Beullens, S. Mukherjee, and C. Rechberger, "Concretely Efficient Blind Signatures Based on VOLE-in-the-Head Proofs and the MAYO Trapdoor," 2026, pp. 6325–6344. [Online]. Available: https://www.usenix.org/conference/usenixsecurity26/presentation/baum

[64] C. Jeudy and O. Sanders. (2026) Compact Lattice Anonymous Credentials from Tighter Approximate Range Proofs. [Online]. Available: https://eprint.iacr.org/2026/1920

[65] Z. Liu, Y. Xu, D. Tang, and N. Zeng, "Efficient Post-quantum EPID Signatures From VOLE-in-the-Head," in *Applied Cryptography and Network Security: 24th International Conference, ACNS 2026, Stony Brook, NY, USA, June 22–25, 2026, Proceedings, Part I*. Springer-Verlag, pp. 3–31. [Online]. Available: https://doi.org/10.1007/978-3-032-32560-0_1

[66] A. I. Lagos, D. Spinellis, and N. Alexopoulos, "Measuring the Performance of Candidate Verifiable Credential Schemes for the EU Digital Identity Wallet," in *Proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '26. Association for Computing Machinery, pp. 1240–1247. [Online]. Available: https://dl.acm.org/doi/10.1145/3748522.3779904

[67] E. Crites, C. Komlo, M. Maller, S. Tessaro, and C. Zhu. (2023) Snowblind: A Threshold Blind Signature in Pairing-Free Groups. [Online]. Available: https://eprint.iacr.org/2023/1228

[68] K. Hayakawa, T. Nakanishi, T. Kitasuka, and Z. Lian, "DL-Based Ring Signatures Using Curve Trees: Sublinear Signing and Verification with Precomputed Tree," in *Information and Communications Security*, C. Su, M. Conti, Y. Cao, and K. Omote, Eds. Springer Nature, 2026, pp. 443–463.

[69] T. H. Yuen, M. F. Esgin, J. K. Liu, M. H. Au, and Z. Ding, "DualRing: Generic Construction of Ring Signatures with Efficient Instantiations," in *Advances in Cryptology – CRYPTO 2021*, T. Malkin and C. Peikert, Eds. Springer International Publishing, 2021, pp. 251–281. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-84242-0_10

A New Definitions for Shrinkable Ring Signatures

Nguyen et al. [22] provide one formalization for the security properties of shrinkable ring signatures. We provide our own in Section 6.1. Our definitions have the following differences:

Soundness against malicious signer. Shrink soundness should hold even against the true signer, or the holder of the secret key. In [22, Fig. 5], the signer is not allowed to be corrupted, and thus the adversary does not have access to the secret key. Our definition (Definition 6.6) allows for an adversary that may corrupt the signer.

Adaptive oracle access. In [22, Fig. 4,5], the adversary is not given oracle access after receiving the challenge. We permit continued oracle access.

Separating shrink unforgeability from shrink soundness. If an adversary can win the game in [22, Fig. 5], it can then trivially win the game in [22, Fig. 4]. The freshness condition in Fig. 4 does not restrict the adversary because the adversary is not given oracle access after σ^* is defined. Thus, the win condition in Fig. 5 is the same as Fig. 4 but just with an additional condition, so any adversary to Fig. 5 can win in Fig. 4. With the two fixes above, we separate shrink unforgeability and shrink soundness.

Clarified shrink oracle. As defined in [22], the adversary can query the shrink oracle for the shrink evidence for any R, R', i, σ tuple. This trivially leads to a break in anonymity: the adversary

can guess a signer i and see if the oracle returns valid shrink evidence. The anonymity game in [22, Fig. 2] nods towards this by requiring the adversary to make queries of the form (R, R', i_b, σ) for challenge bit b, but the adversary does not know b, so it cannot necessarily make queries of this form. We define the shrink oracle with parameters and puncture it accordingly in the anonymity experiment (Definition 6.3).

Clarified notation. The definitions in [22] have some instances of undefined variables: σ^* in Fig. 4 and Fig. 5 and $\tilde{R}$ in Fig. 5.