Decentralizing Photo Forensics for Public and Verifiable Trust

Mauro Clavijo
Universitat Rovira i Virgili
Tarragona, Spain
mauro.clavijo@urv.cat

Rolando Trujillo-Rasua
Universitat Rovira i Virgili
Tarragona, Spain
rolando.trujillo@urv.cat

Carles Anglés-Tafalla
Universitat Rovira i Virgili
Tarragona, Spain
carles.angles@urv.cat

***Abstract*—The integrity and traceability of digital photographic evidence represent a critical factor during forensic investigations, especially when this evidence undergoes technical transformations, such as cropping or resolution enhancement. Ensuring that these modifications remain transparent, verifiable, and attributable is essential to maintaining the value of the evidence during an investigation. To meet these requirements, existing systems typically rely on blockchain-based implementations within permissioned networks or provide only limited support for image transformations. As a result, they often lack the flexibility and transparency required for open or decentralized forensic scenarios. In this paper, we propose an endorsement-based image forensics system that leverages public blockchain to record the lifecycle and verify the authenticity of images. Our system employs hybrid encryption to provide confidentiality of uploaded images while simultaneously ensuring that they remain auditable and non-repudiable. The system supports different trust models and enables users to assess the trustworthiness of an image's provenance data directly and indirectly. Direct trust is achieved by validating an image transformation through reproducible functions or zero-knowledge proofs; indirect trust is enabled through publicly recorded endorsements. Our design achieves low gas costs and provides confidentiality, verifiability, and traceability guarantees, improving upon previous approaches without relying on permissioned infrastructures.**

***Index Terms*—blockchain, photo forensics, confidentiality, transformation validation.**

I. Introduction

The presentation of reliable evidence is central across diverse fields, including criminal investigations, journalism, and insurance claims. Digital forensics, in particular, relies heavily on robust methods to securely collect, preserve, and analyze digital evidence, ensuring its technical integrity. If concerns arise about the authenticity and tampering of such evidence, especially when the origin (or any edition to it) is not correctly documented, the evidence might become legally inadmissible. These concerns have recently increased due to Generative Artificial Intelligence (GenAI), which can lead to the production and manipulation of convincing images [5]. In that sense, there is a growing need for systems that ensure the authenticity of digital images throughout their lifecycle.

Authenticity extends beyond static integrity or non-repudiation. Recent studies [12], [6], [7], [9] converge on a more comprehensive view: authentic images must be traceable, meaning their transformation history should be verifiable and reproducible, from the original capture to the final version. Achieving these properties involves three key guarantees: (1) verification of the source and editors (e.g., cryptographic signatures or C2PA [7]), (2) assurance that image transformations are non-malicious (e.g., validated through digital forensics or auditability), and (3) reproducibility or verifiability of those modifications (either through reproducible or zero-knowledge proofs).

The ability to reproduce transformation chains from a legitimate source is the central goal of image authenticity. Yet, this level of transparency introduces a fundamental tension with confidentiality. Revealing the full chain of transformations may expose sensitive or protected information, such as identities of witnesses or classified visual content. Existing solutions approach this tension through two opposing trust models. *Indirect trust* relies on central authorities, such as C2PA, to guarantee authenticity via certified metadata. *Direct trust* allows users to independently verify edits using zero-knowledge proofs, confirming correctness without revealing content. Each model, however, is limited: C2PA lacks verification of transformation correctness or support for alternative trust mechanisms, while ZK-based systems [12], [6] remain computationally expensive and applicable only to predefined operations. A decentralized, selective confidentiality model—where content remains private but transformation chains are still verifiable—is therefore needed.

In addition to full-traceability and protection of sensitive information, modern forensics systems would benefit from the inclusion of all members of the society in the data collection and validation process. Today, people have camera devices (smartphones), making them an important source of forensic evidence, and some like to participate in the fight against crime by interacting with other members of the community. Phenomena such as Human Fleash Search (HFS), where online communities (so-called "netizens") collaboratively identify suspects or verify content [2], [13], illustrate how public involvement can enhance digital investigations when properly harnessed.

In this context, blockchain emerges not merely as a practical implementation tool, but as a foundational design element inherently aligned with the system's core requirements. Prior work [11], [15], [1], [10], [18] has consistently identified blockchain as a key enabler for ensuring traceability in digital forensics and multimedia authenticity [14], [4]. Its properties of immutability, decentralization, and auditability address the critical challenges of traceability, transparency, and community-driven validation.

However, none of these works handles confidentiality and openness in a satisfactory way. On the one hand, systems such as Custody Chain [11] and Sharma et al. [15] focus on registering metadata of digital evidence using permissioned blockchain technology, which represents an entry barrier for users. On the other hand, systems using public blockchains, such as PhotoChain [10] and Yan et al. [17], do not provide transformation validation or access control policies to ensure that images remain confidential. To the best of our knowledge, no existing blockchain-based system offers a flexible model that supports an open and community-driven validation of forensics data by means of both: indirect and direct trust.

A. Contributions

This paper presents a public blockchain-based image forensic system that enhances provenance records via explicit transformation validation to ensure that digital images can be traced, cannot be denied, and remain confidential. Our proposed framework emphasizes explicit validation of image modifications via an auditable list of individual validators, unlike existing solutions that primarily focus on fundamental integrity checks or access limitations.

Our system is designed to accommodate both scenarios: It can operate in private contexts, like traditional forensic investigations, using hybrid encryption to safeguard sensitive data, or in open scenarios, like the Human Flesh Search (HFS) phenomenon, where selected public validators participate in the investigation of a crime. Importantly, it supports reproducible validation and zero-knowledge validation, allowing transformations to be replicated when possible or maintaining confidentiality when necessary.

Critically, our system does not impose a fixed notion of trust. Instead, it delegates trust assessment to the end-user, who may rely on their own criteria, such as a list of trusted validators or quorum thresholds, to determine whether a transformation chain is acceptable. In addition, it incorporates hybrid encrypted off-chain storage and on-chain logs to maintain security and traceability. The core contributions of the system include:

- **Flexible Transformation Validation:** Motivated by the need to support transformation chains that may involve both reproducible and privacy-sensitive operations, we introduce a flexible validation policy. In this model, validators, which may include experts or community members (e.g., in HFS-like scenarios), can endorse the authenticity of image transformations using reproducible or zero-knowledge methods. This policy maintains a clear and auditable record of who validated each transformation, when, and under what conditions.
- **Selective Access Control for reproducible transformations:** To balance reproducibility with confidentiality, our system restricts access to transformation details where necessary, while still permitting public validation when suitable.
- **Auditable Provenance via Public Blockchain:** Using the inherent immutability of public blockchain technology, our system provides transparent and auditable logs of all changes to the image, endorsements, and user actions. Our solution addresses a significant drawback in current forensic blockchain systems by combining selective confidentiality with public transparency in a novel way.

B. Organization

The content of this article proceeds as follows. Section II reviews related work. Section III introduces the proposed endorsement-based image forensics system. Section IV details the functionality of each feature of the system. Then, Section V presents a security analysis. Section VI presents a preliminary empirical evaluation. Lastly, Section VII presents conclusions and describes future research directions.

II. Related Work

The authenticity of digital images has been a growing area of research, especially in the face of AI-generated content and manipulated media. Two main directions in the recent literature address this issue: centralized provenance systems and privacy-preserving cryptographic verification.

The C2PA standard [7] provides a framework for associating cryptographically signed metadata with images, enabling source verification and non-repudiation. However, C2PA relies on trusted certificate authorities and cannot verify whether a transformation is legitimate or reproducible, which limits its use in open or adversarial environments. In contrast, zero-knowledge (ZK) proof systems [12], [6], [9] validate transformations without revealing content but remain computationally expensive and limited to predefined operations.

Given these limitations, blockchain-based systems have emerged as a decentralized alternative to traceable and auditable image provenance. Existing approaches can be classified as *permissioned* or *public* depending on their consensus model and access control. Permissioned systems, such as Custody Chain [11], ForensiBlock [1], and the work by Sharma et al. [15], commonly employ Hyperledger Fabric or private Ethereum instances to manage evidence within controlled networks. These systems achieve confidentiality

and integrity but sacrifice transparency and openness, and typically omit transformation validation.

More recent research explores public blockchain solutions to improve openness and auditability. Zou et al. [18] proposed a photo forensics system that validates transformations by comparing edited results on-chain, though without privacy guarantees. Yan et al. [17] introduced access-controlled notarization using CP-ABE and BLS signatures, while Li et al. [10] presented PhotoChain, which leverages a Trusted Execution Environment (TEE) and IPFS for secure multimedia validation. However, these systems restrict transformation rights to owners or hardware-specific devices and lack flexible confidentiality controls.

In summary, most blockchain-based forensic frameworks ensure data integrity but are limited either by closed participation (permissioned networks) or by insufficient confidentiality and transformation validation (public networks). Our proposed framework bridges this gap by combining hybrid encryption, public blockchain transparency, and validator-based transformation verification—supporting both reproducible and zero-knowledge modes. Table I compares our approach with existing systems in terms of openness, confidentiality, access control, and authenticity of transformation.

TABLE I: Comparison of state-of-the-art forensic systems. Legend: PBC = Public Blockchain, IC = Image Confidentiality, AC = Access Control, AT = Authenticity of Transformations.

System	PBC	IC	AC	AT
Custody Chain [11]		✓	✓	
Sharma et al. [15]		✓	✓	
PhotoChain [10]	✓	✓		✓
Zou et al. [18]		✓	✓	✓
ForensiBlock [1]		✓	✓	
Yan et al. [17]	✓	✓	✓	
Proposed System	✓	✓	✓	✓

III. Endorsement-based image forensics system

This section presents a general overview of our proposed endorsement-based image forensics system.

A. Actors

Our system involves different entities and user roles. We employ the UML standard [8], where an *actor* represents an external entity interacting with the system, and *role* represents a behavior that an actor can assume.

- **Certified cameras (C):** Like C2PA, our system relies on a set of certified devices capable of capturing digital images and signing them with their long-term private key sk_c to ensure authenticity and non-repudiation. We assume that certified cameras are tamper-proof and that sk_c is securely embedded at the manufacturing stage.
- **Users (U):** The set of human actors in the system. Each user can dynamically assume one or more of the following roles:
  - **Owner (O):** Captures photos with certified cameras and uploads them. Owners may also apply access control rules to their images when they are stored off-chain.
  - **Editor (E):** Edits or transforms images. To publish a transformation, editors must provide sufficient metadata and proof (e.g., reproducible function or zero-knowledge proofs) to enable validation by others.
  - **Validator (V):** Endorses transformations that can be verified correct. Validators may act independently or be recognized through system-defined incentives (see Section IV-E).
- **Blockchain (BC):** A public ledger that maintains records of image provenance, including image metadata, transformation histories, and access permissions.
- **Decentralized Database (DDB):** Decentralized off-chain storage that stores cryptographically encrypted images.
- **Public Key Directory (PKD):** A decentralized registry actor that stores public keys of editors and validators.

B. General overview

The proposed scheme, shown in Figure 1, is designed to maintain a privacy-preserving record of image transformations. The process starts when a user (owner O) registers an original image. If confidentiality is needed, the image is encrypted using hybrid encryption while its cryptographic hash is published on the blockchain as a tamper-evident anchor.

Subsequent transformations are performed by users acting as editors. Every time an editor transforms an image, a new version is created. This new version is stored off-chain, while the hash and transformation metadata are published on-chain.

One or more users may serve as validators to ensure that the transformation is authentic. They do this by independently checking that the transformation is correct using one of the following validation modes:

- **Reproducible** transformation: Validators reapply the transformation to determine if it provides the same result.
- **Zero-knowledge** validation: Validators verify a proof of correctness without accessing the image content, preserving confidentiality.

Our system presents a workflow where all roles and validation modes are defined per-image and are not mutually exclusive: a user may act as an owner, editor, or validator depending on the instance. Likewise, validation modes are defined per transformation, allowing different steps in a transformation chain to use different validation mechanisms. For example, a final image may result from a sequence in which one transformation was reproducibly validated while

another was validated via zero knowledge. This per-instance flexibility illustrates the core design principles of our system, which we describe in detail next.

1) **Register an original picture p_0:** An owner captures a signed picture p_0 with a camera. Hybrid encryption is applied only if confidentiality is required; in that case, the image is encrypted and uploaded to DDB with an access control policy. Otherwise, the image may remain publicly accessible.
2) **Publish metadata of p_0:** The system computes a cryptographic hash of p_0 and records it on-chain along with the camera signature, serving as immutable evidence of provenance.
3) **Editors securely obtain p:** If p is securely stored at DDB with encryption key k, the editor e can check whether k appears encrypted with e's public key, namely pk_e. If so, e might decrypt p.
4) **Edit p and obtain p':** The editor applies a transformation function f to the image p, generating a modified version denoted p'. The transformation may fall under any of the supported validation modes, depending on its nature and sensitivity.
5) **Upload transformed picture p':** If confidentiality is required, the editor e encrypts the transformed image p' using a hybrid encryption scheme and uploads it to DDB. The transformed picture inherits, by default, the access control of the previous image. However, the editor may optionally restrict access further (e.g., limiting users access to sensitive transformation), but cannot expand access beyond the owner's original policy.
6) **Publish metadata of p', f:** The system calculates and publishes $h(p')$, the metadata required to reproduce the transformation (e.g., function or zk-proof, if applicable), and the selected validation mode to BC.
7) **Validators securely obtain p (if needed):** If the validation mode is reproducible, validators may need to access p to verify the transformation. If p was encrypted, they follow the same decryption process as in Step 3, provided they have decryption rights.
8) **Obtain metadata from p, p', f:** Validators query BC to access validation metadata: $h(p), h(p')$, f, validation mode and optional proof data.
9) **Calculate transformation and validate:** Validation proceeds based on the declared mode (reproducible, zero-knowledge, or unverifiable).
10) **Publish validation result:** Validators who confirm the transformation publish an endorsement on-chain. End-users can later consult these endorsements to assess the trustworthiness of the transformation based on their own criteria. We discuss possible trust policies in Section IV-D.

C. System requirements

Now that we have a general overview of the proposed framework, we establish the core requirements that the system must fulfill to support image transformation in a blockchain-based forensic scenario.

1) Functional requirements:

- **FR1: Cost-optimized public blockchain integration:** The system must be compatible with consolidated public blockchain platforms (e.g., Ethereum) while minimizing transaction fees to ensure practical deployment and economic feasibility. This includes the efficient use of smart contracts and on-chain storage to reduce gas costs.
- **FR2: Correct provenance chain:** Each transformation must be linked to its predecessor with associated metadata published on-chain to enable traceability.
- **FR3: Validator endorsement visibility:** Validator endorsements for modifications must be kept in a way that is publicly accessible to assist users judge trustworthiness.
- **FR4: Sound provenance and validation data:** Given an image p, it must be possible for a user to trace p back to its sources.

2) Security requirements:

- **SR1: Integrity of evidence:** Image hashes, transformation metadata, and validator endorsements must be tamper-evident.
- **SR2: Non-repudiation of actions:** Actions, such as image uploads or validations, must be registered on-chain in a way that prevents denial of authorship by any party involved (e.g., cameras, editors, validators).
- **SR3: Conditional confidentiality:** When required, image data stored off-chain (e.g., in DDB) must be accessible only to users with sufficient permissions. Public images may remain unencrypted.
- **SR4: Authenticity of source images:** All source images published at BC must come from certified cameras.
- **SR5: Resilience to validator-based denial-of-service:** The system must prevent malicious validators from degrading the endorsement process by discouraging or limiting false validations and ensuring that only trusted, reputable endorsements are counted.

IV. The system in detail

This section presents a detailed description of the system features. We describe the processes of registering, transforming, and validating an image. Then, we outline a description of possible validation mechanisms and, finally, motivation for validators.

Unless otherwise specified, all algorithmic descriptions and processes below assume the general scenario where confidentiality is required, that is, images and associated data are encrypted and access-controlled by default. In cases

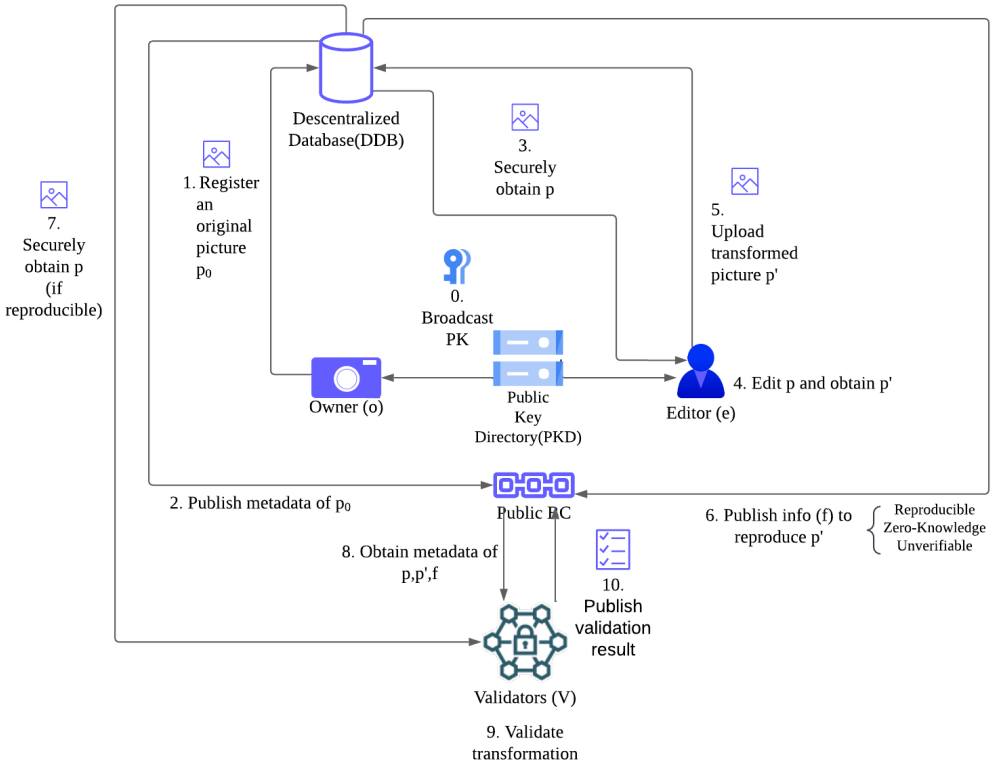


Fig. 1: An overview of our endorsement-based image forensics system. The diagram illustrates the complete scenario, where confidentiality is enabled and the validation mode is reproducible. For simplicity, the diagram assumes all communication channels to be secure.

where confidentiality is not needed, the same algorithms can be applied by omitting the encryption and access control steps, resulting in a simplified workflow.

A. Registering an original image

The registration of an original image is a fundamental step in our forensic framework, as it ensures that owners upload pictures with a verifiable and traceable origin. This step establishes the initial trust anchor for all subsequent image transformations and validations. Algorithm 1 illustrates the procedure for securely registering an original image.

In this process, the owner o captures a picture p_0 using a certified camera, which generates a digital signature over the image hash, $sig_{sk_C}(h(p_0))$. To enforce access control, a hybrid encryption scheme is used: a random symmetric key k encrypts p_0, and k is itself encrypted with the public keys of all authorized users listed in the public key directory (PKD). Both the encrypted image and the set of encrypted keys are stored in the decentralized database (DDB), ensuring that only designated users can decrypt the image.

Finally, the system computes the hash of p_0 and records it on-chain, together with the camera signature, by invoking a smart contract on the blockchain (BC).

Algorithm 1 Register an original image

1: **Input:** Original picture p_0, signature $sig_{sk_C}(h(p_0))$, public keys of authorized users PK_U
2: **Output:** Hash $h(p_0)$, data stored in BC and DDB
3: $enc(p_0, k_0) \leftarrow$ Symmetric encryption of p_0 using randomly generated key k_0
4: $enc_{PK_U}(k_0) \leftarrow$ Asymmetric encryption of k_0 for each $pk_u \in PK_U$
5: $e_0 \leftarrow (enc(p_0, k_0), enc_{PK_U}(k_0))$
6: Store e_0 in DDB and set $success \leftarrow$ `True` if storage succeeds; otherwise, set $success \leftarrow$ `False`
7: **if** $success =$ `True` **then**
8: Publish hash $h(p_0), sig_{sk_C}(h(p_0))$ in BC
9: **return** $h(p_0)$
10: **else**
11: **return** `Fail`
12: **end if**

B. Image transformation

Once an image has been registered, either originally captured or previously transformed, an editor first decrypts the symmetric key (using his/her private key) and then decrypts

the image data from DDB.

Given an image p, the editor applies a transformation function f to produce a new image p'. The transformed image is then encrypted with a newly generated symmetric key, which is itself encrypted for each authorized user, and all resulting encrypted data are uploaded to off-chain storage. Metadata, including $h(p')$, f, and the chosen validation mode—is published to BC, enabling traceability and (possibly) verification.

The validation mode determines how subsequent validators will assess the legitimacy of the transformation. In reproducible mode, the validators can reconstruct and independently verify the transformation by applying f to the original image. In zero-knowledge mode, the editor must generate a cryptographic proof that demonstrates both possession of the original and transformed images, and that the claimed transformation is correct, without disclosing the image data itself. This is typically accomplished using zero-knowledge proof (ZKP) protocols, such as those described in [12], [6], where the user proves knowledge of (p, p') and that $p' = f(p)$ holds. If confidentiality must be strictly preserved, only the proof is published on-chain without revealing the image.

Algorithm 2 Transform an image

Input: Encrypted image $enc(p, k)$, encrypted key $enc_{pk_e}(k)$, transformation function f, validation mode
Output: Hash of transformed image $h(p')$, data stored in BC and DDB
$k \leftarrow$ Decrypt $enc_{pk_e}(k)$ using sk_e
$p \leftarrow$ Decrypt $enc(p, k)$ using symmetric key k
$p' \leftarrow f(p)$
Generate random symmetric key k'
$enc(p', k') \leftarrow$ Symmetric encryption of p' with k'
$enc_{PK}(k') \leftarrow$ Asymmetric encrypt k' under authorized users' public keys
$e' \leftarrow (enc(p', k'), enc_{PK}(k'))$
Upload e' to DDB
if validationMode = reproducible **then**
    $f_{\text{publish}} \leftarrow f$
else if validationMode = zero-knowledge **then**
    Generate zk-proof π of $f(p) = p'$
    $f_{\text{publish}} \leftarrow \pi$
else if validationMode = unverifiable **then**
    $f_{\text{publish}} \leftarrow null$
end if
$t \leftarrow (h(p'), f_{\text{publish}}, validationMode)$ to BC
Publish $t, sig_{sk_e}(t)$ on BC
return $h(p')$

C. Validate the authenticity of an image transformation

When an image transformation is published, users are notified and have the option to validate the transformation. In the reproducible mode, a validator obtains both the input and output images—either by decrypting them from the DDB (in the confidential scenario) or accessing them directly if they are public—and independently applies the published transformation function f to verify that the output matches the recorded hash $h(p')$.

In the zero-knowledge case, the user verifies the ZK cryptographic proof obtained from the BC without accessing the contents of the image. Successful validations result in signed endorsements recorded on-chain in the validation result for transparency and auditability.

Besides verifying correctness, validators check that the editor was authorized to access p at the transformation time by consulting its on-chain access control record. Algorithm 3 summarizes this validation process.

Algorithm 3 Validation of image transformations

Input: Encrypted previous image $enc(p, k)$, encrypted key $enc_{pk_v}(k)$, hash of transformed image $h(p')$, published transformation data f_{publish}, $validationMode$
Output: Endorsement recorded in validation result on BC
Check that the editor is authorized to access p via its on-chain access control record; **abort** if not authorized
if validationMode = reproducible **then**
    $k \leftarrow$ Decrypt $enc_{pk_u}(k)$ using private key sk_u
    $p \leftarrow$ Decrypt $enc(p, k)$ using symmetric key k
    Compute $success \leftarrow (h(f_{\text{publish}}(p)) = h(p'))$
else if $validationMode$ = zero-knowledge **then**
    Verify zero-knowledge proof f_{publish}
    $success \leftarrow$ Result of proof verification
else if $validationMode$ = unverifiable **then**
    $success \leftarrow$ False {Unverifiable mode cannot be validated}
end if
if $success$ = true **then**
    Set $validationResult[h(p')][u] \leftarrow$ true
else
    Set $validationResult[h(p')][u] \leftarrow$ false
end if
$val \leftarrow validationResult[h(p')][u]$
Publish $val, sig_{sk_v}(val)$ on BC

D. Trust evaluation

A key feature of our system is that it allows end-users (e.g., a judge, jurors, editors or netizens) to assess the trustworthiness of an image based on four properties of a transformation: i) reproducibility, ii) zero-knowledge verification, iii) editor authentication and iv) endorsements. We call these *trust properties*, as they contribute to the consumer's belief that a transformation did not tampered with the data in a dishonest way.

We argue that trust is a function on the properties satisfied by a transformation. Formally, given a transformation t, we define $\mathsf{trust}(t)$ to be a logical formula whose literals are trust properties. For example, we can express that a transformation is trusted if it is reproducible or if the editor is trusted. We extend the trust function to a provenance chain $\tau = t_1 \cdots t_n$ by letting $\mathsf{trust}(\tau)$ to be equal to $\mathsf{trust}(t_1) \wedge \cdots \wedge \mathsf{trust}(t_n)$. In words, a provenance chain is trusted if all transformations involved in the production of the data are trusted.

It is worth remarking that other trust functions can be used within our system. For example, by letting each property have a weight, then $\mathsf{trust}(t)$ could be the weighted sum of all properties satisfied by t, which expresses the fact that some properties may have more impact on trust than others. This is particularly relevant for the evaluation of endorsements, since an image endorsed by a single trusted validator may not be as trustworthy as an image endorsed by many. Another way to evaluate trust based on endorsements is via a threshold scheme. This has the advantage of staying within propositional logic, hence being compatible within our trust function formulation. Ultimately, the application domain or end-user will determine the most appropriate formulation for the trust function.

Another important point is that when we say an editor or validator is trusted, we mean that their identity (or cryptographic public key) is included in a user's list of trusted entities. This list may be maintained personally, managed by a community, or integrated into a public key infrastructure such as C2PA.

E. Motivation of validators

Validators play the most critical role in the image authentication process. To ensure the active work of the validators and prevent malicious behavior, it is essential to motivate them correctly. We propose two mechanisms to encourage a validator to participate actively and correctly endorse authenticity.

1) **Economic-based rewards:** In conventional digital forensics, expert witnesses are employed to assess the authenticity and integrity of the data [3]. Instead of relying on a single expert's interpretation, the system can employ the same economic resources to motivate validators, by providing economic incentives.
2) **Reputation-based System:** Validators increase their reputation score every time they successfully endorse an image. The reputation score can better influence the validation process and have public recognition with a list of top validators always available on the blockchain.

V. Security analysis

This section provides informal security arguments on how the proposed framework meets the security requirements defined in Section III-C.

In our analysis, we assume a Dolev-Yao attacker in full control of the network, i.e. capable of observing/intercepting all messages sent to the network and of injecting new messages. The goal of the attacker is to tamper with image data in an undetected way.

We assume idealized cryptography, meaning that hash functions are one-way and collision-resistant, encryption reveals no information about the plaintext, and cryptographic signatures are unforgeable.

Security Claim 1 (Integrity and authenticity of transformations)**.** *Each transformation $(h(p), f, h(p'))$ recorded on the blockchain is guaranteed to reference an existing image p and to be submitted with a valid digital signature from the editor who performed the transformation.*

Proof Sketch. The smart contract accepts a transformation only if $h(p)$ is already registered and the transaction carries a valid signature from the editor's registered public key. With blockchain immutability and unforgeable signatures, an adversary cannot alter metadata. □

Security Claim 2 (Non-repudiation of actions)**.** *Each action recorded on the blockchain (such as image registration, transformation, or validation) is irrefutably linked to the corresponding user, and users cannot deny their authorship of these actions.*

Proof Sketch. All relevant transactions are digitally signed by the respective actor's private key (sk_e, sk_v, sk_C). The blockchain ensures immutability, so these signatures cannot be removed or altered after inclusion. Verification of the signature with the public key proves authorship, ensuring that users cannot plausibly deny their recorded actions. □

Security Claim 3 (Conditional confidentiality)**.** *Unauthorized entities cannot access the data of the plaintext image, ensuring the confidentiality of the evidence even in public or adversarial environments.*

Proof Sketch. Symmetric encryption (like AES) is used to encrypt image data, and authorized users' public keys (like RSA) are used to encrypt the symmetric keys. Only users who have the right private key can decrypt the image under normal IND-CCA conditions. Even if encrypted data are publicly available (e.g., in DDB), confidentiality is preserved. In zero-knowledge mode, no image data may be shared, further minimizing information exposure. □

Security Claim 4 (Auditability of transformation chains)**.** *Every transformation chain stored on the blockchain is auditable and independently verifiable, ensuring detection of any tampering or inconsistency.*

Proof Sketch. The use of cryptographic hashes to link each state of the image and transformation $(h(p) \xrightarrow{f} h(p'))$ allows any validator to reconstruct and verify the entire sequence. In reproducible mode, validators can use the transformation

f on their own and make sure that $f(p) = p'$. In zero-knowledge mode, validators check a cryptographic proof that the transformation is correct, as in [12], [6]. Any inconsistency or tampering breaks the cryptographic link or fails proof verification, ensuring that the audit trail is always correct. □

Security Claim 5 (Resilience to denial-of-service by validators)**.** *The system discourages and limits denial-of-service attacks and fraudulent validations by malicious validators, ensuring that the integrity and throughput of the endorsement process are maintained.*

Proof Sketch. First, every attempt at validation, whether honest or malicious, costs an abundance of gas, which makes it difficult to conduct large-scale flooding attacks. Second, the system uses reputation- or trust-based rules. For example, validators who submit fake endorsements repeatedly may lose rewards or reputation, and trust policies may stop their votes from being counted (See Section IV-E). Threshold or trust-list systems make sure that only trustworthy validators can change the results of validation. These steps stop both large-scale and targeted DoS attacks, which keeps the system secure and available.

□

VI. Empirical Validation

In this section, we empirically validate our system as described in the functional requirements of Section III-C. Specifically, we focus on validating FR1: Cost-optimized public blockchain integration by measuring gas costs associated with blockchain integration.

A. Experimental Setting

To evaluate the gas cost of our system, we implemented the smart contract in Solidity using the Remix IDE (version 0.62.1), which provides a streamlined development environment for Ethereum Virtual Machine (EVM). The contract was deployed and tested on Ganache 2.7.1[1], a local blockchain simulator for development and testing. Ganache was run on a 13th Gen Intel® Core™ i7-13620H with 16 GB RAM and 1 TB SSD, running Ubuntu 24.04.2 LTS.

B. Gas Cost

We measured gas consumption in an Ethereum Virtual Machine (EVM) environment to evaluate our system's (FR1) blockchain cost-efficiency. We concentrate on the three main functions: image registration, transformation validation, and contract deployment.

We evaluate two main smart contract functions:

- The function `Store_Image` is responsible for registering original image hashes and transformed image metadata such as image hash, parent-child relationships, and transformation identifiers.

[1]https://trufflesuite.com/ganache/

TABLE II: Gas Usage for Core Contract Operations

Operation	Gas Used
Contract Deployment	634975
storeImage	51750
confirmValidation	92430

- The `Confirm_Validation` function allows validators to endorse an image transformation. Ensures that a transformation exists and that the validator has not already validated the same image, and then publishes a $validationResult$.

Table II shows the gas usage for three fundamental operations: total contract deployment, image registration, and validation functions. The deployment of our smart contract consumes 634975 gas, which is the most expensive operation, but it only needs to be executed once to deploy the system. The `Store_Image` function, responsible for registering both original and transformed image metadata, consumes 51750 gas per execution. Finally, the `Confirm_Validation` function, which allows validators to endorse transformations, consumes approximately 92430 gas per operation.

For context, the base gas cost of the simplest transaction in Ethereum is 21,000 gas [16]. Our image registration and validation operations require only about 2.5x and 4.4x this base cost, respectively. This confirms that our system operates well within feasible gas limits for public deployment and validates our initial premise (FR1) that a cost-effective, blockchain-based image forensics system is achievable.

VII. Discussion, conclusions and future work

This work introduces an endorsement-based image forensics system that leverages public blockchain and hybrid encryption to guarantee the integrity, traceability and confidentiality of digital evidence. Using a public blockchain to store hashes of images, transformation information, and validation endorsements, the system minimizes on-chain storage while maintaining verifiability.

An informal security analysis demonstrates that the system upholds essential properties, including evidence integrity, non-repudiation, and confidentiality, under standard cryptographic assumptions. These properties collectively address the core security requirements for digital forensic workflows.

Our preliminary empirical evaluation demonstrates that gas costs were shown to remain close to the Ethereum baseline transaction fees, confirming the feasibility of public blockchain integration (FR1).

Future work will focus on extending the current encryption model by exploring shared encryption schemes which will enable a group of validators to access an image without requiring individual key management. Finally, a full-scale performance evaluation will be performed to assess

the system in real world scenarios and enable a proper comparison with state-of-the-art approaches.

ACKNOWLEDGMENT

Mauro Clavijo was supported by Martí i Franqués grant (2023PMF-PIPF-46) from Universitat Rovira i Virgili and project CRISES: Security and Privacy (2021 SGR 00115) funded by Catalunya government.

Rolando Trujillo-Rasua was supported by a Ramón y Cajal grant (RYC2020-028954-I) from the Spanish Ministry of Science and Innovation and the EU, as well as by projects PROVTOPIA (PID2023-150098OB-I00), funded by MICIU/AEI/10.13039/501100011033 and FEDER (EU), and HERMES, funded by INCIBE and the EU's NextGenerationEU/PRTR.

REFERENCES

[1] Asma Jodeiri Akbarfam, Mahdieh Heidaripour, Hoda Maleki, Gokila Dorai, and Gagan Agrawal. Forensiblock: A provenance-driven blockchain framework for data forensics and auditability. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, page 136–145, Atlanta, GA, USA, November 2023. IEEE.

[2] Lennon YC Chang and Jinxin Zhu. Taking justice into their own hands: Predictors of netilantism among cyber citizens in hong kong. *Frontiers in Psychology*, 11:556903, 2020.

[3] Frederick B Cohen. *Digital forensic evidence examination*. Asp Press Livermore, 2010.

[4] Thomas K Dasaklis, Fran Casino, and Constantinos Patsakis. Sok: Blockchain solutions for forensics. In *Technology Development for Security Practitioners*, pages 21–40. Springer, 2021.

[5] Rebecca A. Delfino. Deepfakes on trial: A call to expand the trial judge's gatekeeping role to protect legal proceedings from technological fakery. *Hastings Law Journal*, 74(2):293–349, 2023.

[6] Shahriar Ebrahimi, Lei Zhang, Prateek Bhargava, et al. Vimz: Private proofs of image manipulation using folding-based zksnarks. *IEEE Transactions on Information Forensics and Security*, 2025. To appear.

[7] Coalition for Content Provenance and Authenticity. C2pa specification v2.2, 2025. Available at https://c2pa.org/specifications/specifications/2.2/.

[8] Object Management Group. Unified modeling language (uml) specification, 2017. https://www.omg.org/spec/UML.

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

[10] Xudong Li, Lingbo Wei, Lidong Wang, Ying Ma, Chi Zhang, and Muhammad Sohail. A blockchain-based privacy-preserving authentication system for ensuring multimedia content integrity. *International Journal of Intelligent Systems*, 37(5):3050–3071, May 2022.

[11] Monia Lusetti, Luca Salsi, and Andrea Dallatana. A blockchain based solution for the custody of digital files in forensic medicine. *Forensic Science International: Digital Investigation*, 35:301017, December 2020.

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

[13] Johnny Nhan, Laura Huey, and Ryan Broll. Digilantism: An analysis of crowdsourcing and the boston marathon bombings. *British Journal of Criminology*, 57(2):341–361, 2017.

[14] Yousef Sanjalawe, Sharif Makhadmeh, Hamzah Alqudah, et al. Digital forensics meets blockchain: Enhancing evidence authenticity and traceability. In *2024 2nd International Conference on Cyber Resilience (ICCR)*, pages 01–06. IEEE, 2024.

[15] Pratima Sharma, Rajni Jindal, and Malaya Dutta Borah. Blockchain-based distributed application for multimedia system using hyperledger fabric. *Multimedia Tools and Applications*, 83(1):2473–2499, January 2024.

[16] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger (yellow paper). *Ethereum Foundation*, 2014.

[17] Wenqi Yan, Jiachen Shen, Zhenfu Cao, and Xiaolei Dong. Blockchain based digital evidence chain of custody. In *Proceedings of the 2020 2nd International Conference on Blockchain Technology*, page 19–23, Hilo HI USA, March 2020. ACM.

[18] Renpeng Zou, Xixiang Lv, and Baocang Wang. Blockchain-based photo forensics with permissible transformations. *Computers and Security*, 87:101567, November 2019.