# Efficient Verifiable Image Redacting based on zk-SNARKs

Hankyung Ko
Hanyang University
Seoul, Korea
hankyungko@hanyang.ac.kr

Ingeun Lee
Kookmin University
Seoul, Korea
ingeunlee@kookmin.ac.kr

Seunghwa Lee
Kookmin University
Seoul, Korea
ttyhgo@kookmin.ac.kr

Jihye Kim
Kookmin University
Seoul, Korea
jihyek@kookmin.ac.kr

Hyunok Oh
Hanyang University
Seoul, Korea
hoh@hanyang.ac.kr

## ABSTRACT

Image is a visual representation of a certain fact and can be used as proof of events. As the utilization of the image increases, it is required to prove its authenticity with the protection of its sensitive personal information. In this paper, we propose a new efficient verifiable image redacting scheme based on zk-SNARKs, a commitment, and a digital signature scheme. We adopt a commit-and-prove SNARK scheme which takes commitments as inputs, in which the authenticity can be quickly verified outside the circuit. We also specify relations between the original and redacted images to guarantee the redacting correctness. Our experimental results show that the proposed scheme is superior to the existing works in terms of the key size and proving time without sacrificing the other parameters. The security of the proposed scheme is proven formally.

## CCS CONCEPTS

• **Security and privacy → Cryptography**; **Privacy-preserving protocols**.

## KEYWORDS

zk-SNARK, image authentication, privacy, filtering

**ACM Reference Format:**
Hankyung Ko, Ingeun Lee, Seunghwa Lee, Jihye Kim, and Hyunok Oh. 2021. Efficient Verifiable Image Redacting based on zk-SNARKs. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Virtual Event, Hong Kong*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3433210.3453110

## 1 INTRODUCTION

Photos are a prevalent data format that visually displays a fact and are increasingly being used today as more and more people have their own IoT devices such as mobile phones, web cameras, and dashboard cameras. The integrity and authenticity of the photographic information becomes very important, when it serves as proof of an identity or an event. In particular, since the spread of COVID-19, the need for a non-contact authentication technique using photo images of an ID card has tremendously increased. Also, there is an ISO standard [9] which allows construction of Mobile Driving License (mDL) applications and users can carry and use it in their phone instead of the plastic card. However, there is a critical risk that privacy can be compromised because photos often contain important personal information. If a photo of an ID card like Figure 1 (a) is leaked, the adversary can steal the victim's identity and may exploit it for criminal activity. While many countries have adopted identity verification using photographic information, to date, they have been used in parallel with other authentication techniques to supplement the limitations of this system. Images can also be used as legal evidence for certain events. Revealing the entire original image without any processing can invade the privacy of unrelated people.

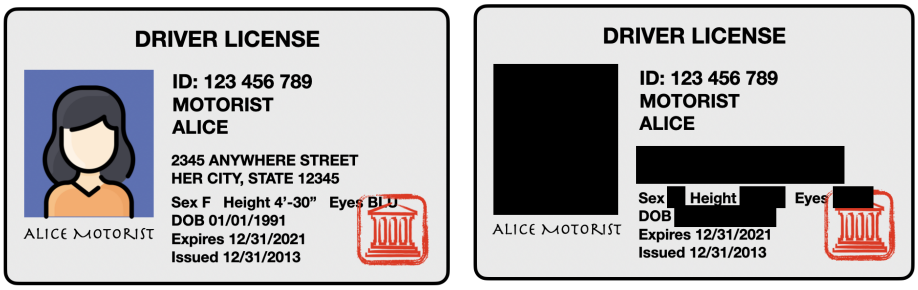
In order to minimize the exposure of personal information, it is recommended to delete all information except for information that is minimally necessary for identification or evidence, as shown in Figure 1 (b). Then the remaining work is to guarantee that the redacted image is identical to the original authenticated one while hiding sensitive information within the image. This paper claims three security requirements for verifiable image redacting: *originality*, *redacting correctness*, and *area zero-knowledge*. First, the originality means that the redacted image should have been generated from the authenticated original image. The prover must be able to attest that she/he has knowledge of the authenticated image. Second, redacting correctness is that the output image is a result of being redacted properly from the input. Lastly, it should not be possible to infer any information about the erased part from the redacted image and proofs.

### 1.1 Related Work

There have been many studies on redactable signature schemes (RSS) which support deletion of signed documents. In RSS systems, any person can black out any part of the signed document. RSS was first proposed by Steinfeld et al. [17] using Merkle hash tree. The security notion of privacy was formalized by Brzuska et al. [2]. RSS systems have been developed based on various data structures, such as tree structures [2, 15], graph structures [12], and cryptographic accumulators [6]. However, the signature size in their schemes was proportional to the message length. A state-of-the-art RSS [16] reduces it to constant by sacrificing its key size; it increases the


*ASIA CCS '21, June 7–11, 2021, Virtual Event, Hong Kong*

ACM ISBN 978-1-4503-8287-8/21/06…$15.00
https://doi.org/10.1145/3433210.3453110




(a) Original image

(b) Redacted image

**Figure 1: An example of ID card**

key size into the square of the message length in order to cover all possible redacting cases. Since it is not known which parts will be redacted before the key setup, the approach in [16] seems to be unavoidable to have a constant size signature. If the conditional statement for the input can be verified in a way that can express all cases of output, it will be possible to design an RSS scheme with a linear complexity.

There is another approach to construct the RSS schemes with a more sophisticated primitive zk-SNARKs (zero-knowledge succinct non-interactive arguments of knowledge) [7, 8, 14]. A zk-SNARKs scheme is a cryptographic tool that succinctly verifies any function's computation without revealing private information such as intermediate values generated in its computation. When zk-SNARKs are applied to RSS, one generic method is to encode all required computations as a function statement for proof; the function includes the signature verification as well as the redacting operation (consisting of conditional statements). Photoproof [13] adopting a proof-carrying data technique [5] basically puts the signature verification process of the input data into the proving statement directly. Signature verification typically consists of multiplication-intensive group operations and nonlinear hash function computations. Encoding them into an arithmetic circuit for zk-SNARKs results in the circuit size being so large that it has a direct impact on proving performance and common reference string (CRS) size. Its proving computation and CRS size complexities are proportional to the cube of the input, roughly taking 15 hours and requiring 460GB of CRS to generate a proof for a 100x100 size image. To mitigate this problem, verifiable document redacting (VDR) proposed in [4] verifies the signature outside the circuit, but hash computation and conditional statements still remain in the circuit. A widely used standard hash function, SHA256, is more than 20,000 lines in arithmetic circuit and a conditional statement represented as bit-wise operation requires 32 lines per 32-bit value. Although the CRS size and proving computation in VDR are reduced to linear complexities to the input, constant factors increases to at least the bit-length due to the conditional statements as shown in Table 1.

## 1.2 Main idea

In this paper, we propose an efficient construction for the verifiable image redacting scheme using zk-SNARKs, a Pedersen commitment scheme, and a digital signature scheme. Figure 2 shows an overview of our verifiable image redacting construction. The verifiable image redacting scheme requires three entities: a certificate authority, a prover, and a verifier. The certificate authority commits the image, signs the committed value, and then publishes the committed value and signature so that anyone can check the authenticity of the committed value while the original image keeps hidden. The prover is a person who wants to use an image as evidence. After redacting the private part, she proves that it is correctly redacted from the committed image. Lastly, the verifier verifies the originality and redacting correctness of the redacted image.

Our construction follows the philosophy of VDR [4] in that the signature is verified outside the circuit, but it does not even include the hash computation and the conditional operations in the circuit, improving the proving time and the crs size into a practical level. We achieve this goal by utilizing cc-SNARK (commit-carrying SNARK) and cp-SNARK (commit-and-prove SNARK) [3]. In cc-SNARK, a commitment is provided as SNARK input. If the Pedersen commitment is used as commitment then the commitment computation becomes for free. In addition, cp-SNARK can efficiently prove that the identical data are used in two Pedersen commitments. It is possible to efficiently prove the equality of data in commitments used for the signature scheme and the zk-SNARK scheme. Note that cc-SNARK is a snark scheme with commitment and cp-SNARK is a proving scheme to connect commits. Therefore, it is recommended to use both cp-SNARK and cc-SNARK together. [3]

To ensure existential unforgeability for RSS, it should be impossible to create a new proof without witness knowledge (the redacted image) even after seeing many proofs under different instances in VDR. I.e., the SNARK scheme used as a building in VDR needs to be simulation-extractable to provide non-malleability that prevents cheating in the presence of simulated proofs. To achieve non-malleability, we adopt se-cp-SNARK scheme (simulation-extractable cp-SNARK) [3, 10] and devise a new se-cc-SNARK scheme (simulation-extractable cc-SNARK) for the proposed verifiable image redacting scheme.

We replace the conditional statement for redacting with a new relation we propose. Since the redacting operation converts each pixel value into zero, the sum of the public output image $u$ and the private hidden image $\overline{u}$ is equivalent to the original input image $m$, i.e., $m = u + \overline{u}$. In addition, it is required that one of $u$ and $\overline{u}$ should be zero. Note that if one of $u$ and $\overline{u}$ values is not forced

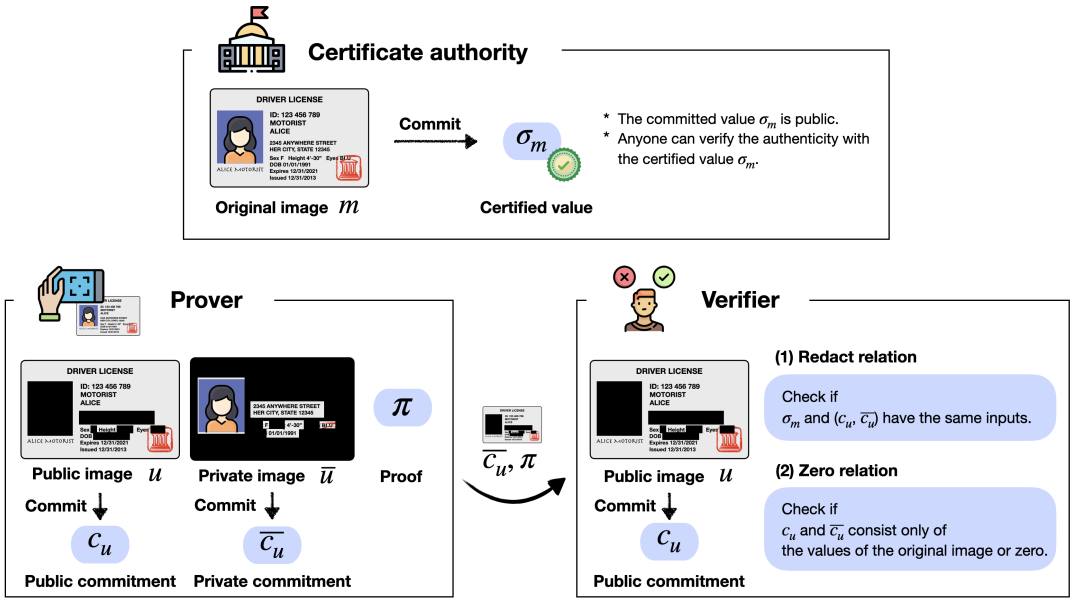



**Figure 2: Verifiable image redacting scheme overview**

to be 0, a malicious prover can manipulate the public output image. For example, given an unredacted pixel value $m = 4$, the pair corresponding to $(u, \overline{u})$ should be $(4, 0)$. However, if only the relationship of $m = u + \overline{u}$ is checked, the verification passes even if the prover intentionally sets $(u, \overline{u}) = (2, 2)$ to manipulate the original pixel value. Relations $\mathcal{R}_{redact}$ and $\mathcal{R}_{zero}$ in Algorithm 1 for se-cp-SNARK and se-cc-SNARK, respectively, denote the relations of sum and zero. Figure 2 shows the proposed VIR system in which the original image is divided into a public part $u$ and a private part $\overline{u}$, and committed to $C_u$ and $\overline{C_u}$.

**Algorithm 1** Relations for verifiable image redacting

$\mathcal{R}_{redact}(\mathbf{h}, \mathbf{f}, \mathbf{f}', C_u, \overline{C_u}, C_m, \mathbf{u}; r, \overline{\mathbf{u}}, m_0, \mathbf{m})$
parse $\mathbf{h} = (h_0, h_1, \ldots, h_n)$, $\mathbf{f} = (f_0, f_1, \ldots, f_n)$, $\mathbf{f}' = (f_0', f_1', \ldots, f_n')$, $\mathbf{u} = (u_1, \ldots, u_n)$, $\overline{\mathbf{u}} = (\overline{u_1}, \ldots, \overline{u_n})$, and $\mathbf{m} = (m_1, m_2, \ldots, m_n)$
assert $C_u = \prod_{i=1}^{n} f_i^{u_i}$
assert $\overline{C_u} = f_0^r \cdot \prod_{i=1}^{n} f_{i+n}^{\overline{u_i}}$
assert $m_i = u_i + \overline{u_i}$ $(s.t.\ i \in [1, n])$
assert $C_m = h_0^{m_0} \cdot \prod_{i=1}^{n} h_i^{m_i}$

$\mathcal{R}_{zero}(\mathbf{u}, \overline{\mathbf{u}};\ )$
parse $\mathbf{u} = (u_1, \ldots, u_n)$, and $\overline{\mathbf{u}} = (\overline{u_1}, \ldots, \overline{u_n})$
assert $u_i \cdot \overline{u_i} = 0$ $(s.t.\ i \in [1, n])$

## 1.3 Our contributions

We summarize our contributions. First, we formally define security notions of a verifiable image redacting scheme. There are three notions: originality, redacting correctness, and area zero-knowledge. We construct a simulation-extractable cp-SNARK scheme (se-cp-SNARK) and a simulation-extractable commit-carrying SNARK (se-cc-SNARK) and propose an efficient verifiable image redacting scheme satisfying all three security requirements, using se-cp-SNARK, se-cc-SNARK, digital signature, and a Pedersen commitment scheme.

Our proposed scheme improves CRS size ($O(n)$) and proving time ($O(n \log n)$) compared with related works without sacrificing the other parameters as shown in Table 1. For FHD images, the proving time is 1.1s in our approach, while it is 6.7s in RSS [16] and 26.2s in VDR [4]. The key size is 2.7MB in our scheme, 2.1GB in RSS, and 670MB in VDR, respectively.

## 1.4 Organization

In Section 2, we describe the preliminaries of the verifiable image redacting scheme. Section 3 defines the security notions, and provides the construction and proof of our scheme. Experimental results are shown in Section 4. Finally, we conclude this paper in Section 5.

# 2 PRELIMINARIES

## 2.1 Notation

If $x$ is a arbitrary string then $|x|$ denotes its bit length. If $S$ is a set then $|S|$ denotes its size and $x \xleftarrow{\$} S$ denotes assigning a member uniformly from S to x. We use the notion of a security parameter $\lambda$ which will be provided as input to scheme and $1^\lambda$ to denote its unary representation. A lowercase bold character denotes a vector, and an uppercase bold character denotes a matrix.

Throughout this paper, $\mathbf{m} = (m_1, \cdots, m_n)$ means the original image, $\mathbf{u} = (u_1, \cdots, u_n)$ means the public image, and $\overline{\mathbf{u}} = (\overline{u_1}, \cdots, \overline{u_n})$ means the private image. Each $u_i$ or $\overline{u_i}$ represents a pixel (or pixel block) of the image, and the empty spaces of $\mathbf{u}$ and $\overline{\mathbf{u}}$ are filled with zeros.

**Table 1: Comparison of verifiable redacting systems. $n$ is the message length, $r$ is the redacted message length, $b$ is the bit length of a pixel(32-bit), and $h$ is the circuit size of a hash function.**

| | Photoproof [13] | VDR [4] | RSS [16] | Ours |
|---|---|---|---|---|
| \|CRS\| | $O(n^3)$ | $O(b \cdot n + h)$ | $O(n^2)$ | $O(n)$ |
| \|Proof\| | $O(1)$ | $O(1)$ | $O(1)$ | $O(1)$ |
| Prover | $O(n^3 \log(n))$ | $O((b \cdot n + h)\log(b \cdot n + h))$ | $O(r \cdot (n - r))$ | $O(n \log(n))$ |
| Verifier | $O(r)$ | $O(r)$ | $O(r)$ | $O(r)$ |

The multiplication of a field element vector $\mathbf{k}$ and a group element $a$ is calculated as $a^{k_0}, a^{k_1}, \ldots, a^{k_n}$.

## 2.2 Bilinear Group

We assume an bilinear group generator $BG$ that is run with $1^k$ and efficiently returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$. "efficiently" mean an algorithm running in time $poly(1^k)$. We will use bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$ with the following properties:

- p is prime in $poly(1^k)$ size.
- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of order $p$ with generator $G_1 \in \mathbb{G}_1, G_2 \in \mathbb{G}_2$.
- e is an efficiently computable pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- $e(G_1, G_2)$ are uniformly chosen generators $\mathbb{G}_T$.

## 2.3 Pedersen Vector Commitment

The Pedersen Vector commitment is a perfectly binding scheme based on the discrete log problem with providing additive homomorphism.

*Definition 2.1.* The Pedersen Commitment scheme for vectors of size n has the triple of PPT algorithms (Keygen, Commit, VerifyCommit) defined as follows. and We consider an instantiation on a group $\mathbb{G}_1$.

- $\mathbf{ck} \leftarrow \mathsf{Keygen}(1^\lambda)$: The key generation algorithm take as input security parameter $\lambda$, and returns a commitment key $\mathbf{ck}$.
- $(C_m, o) \leftarrow \mathsf{Commit}(\mathbf{ck}, \mathbf{w})$: The commit algorithm takes as input a commitment key $\mathbf{ck}$ and a vector $\mathbf{w}$, and returns an opening $o$ and a commitment $C_m$.
- $1/\perp \leftarrow \mathsf{VerifyCommit}(\mathbf{ck}, C_m, \mathbf{w}, o)$: The verifier algorithm takes as input a commitment key $\mathbf{ck}$, a commitment $C_m$, a vector $\mathbf{w}$ and an opening o, and returns 1(accept) or $\perp$(reject).

The scheme is perfectly hiding and computationally binding.
**Computational binding**: For every computational adversary $\mathcal{A}$ having knowledge of $\mathbf{ck}$:

$$Pr\left[\begin{array}{c}\mathsf{VerifyCommit}(\mathbf{ck}, C_m, \mathbf{w}, o) \wedge \\ \mathsf{VerifyCommit}(\mathbf{ck}, C_m, \mathbf{w}', o') \wedge \\ \mathbf{w} \neq \mathbf{w}'\end{array} \middle| (C_m, \mathbf{w}, o, \mathbf{w}', o') \leftarrow \mathcal{A}(\mathbf{ck})\right] = negl(1^\lambda)$$

**Perfect hiding**: For all messages s.t. $\mathbf{m_0} \neq \mathbf{m_1}$ and $|\mathbf{m_0}| = |\mathbf{m_1}|$ and for any adversary $\mathcal{A}$ the following probability is $\frac{1}{2}$:

$$Pr\left[\mathcal{A}(\mathbf{ck}, C_{m_b}) = b \middle| \begin{array}{c}\mathbf{ck} \leftarrow \mathsf{Keygen}(1^\lambda) \\ m_0, m_1 \leftarrow \mathcal{A}(\mathbf{ck}) \\ b \xleftarrow{\$} \{0,1\} \\ (C_{m_b}, r_b) \leftarrow \mathsf{Commit}(\mathbf{ck}, \mathbf{m_b})\end{array}\right] = \frac{1}{2}$$

## 2.4 Digital signature

*Definition 2.2.* The digital signature scheme has triple of PPT algorithms (Keygen, Sign, Verify) defined as follows.

- $(pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$: The key generation algorithm takes security parameter $\lambda$ as input, and returns a public key $pk$ and a signing key $sk$.
- $\sigma_m \leftarrow \mathsf{Sign}(sk, m)$: The signing algorithm takes a signing key $pk$ and a message $m$ as input, and returns a signature $\sigma_m$.
- $1/\perp \leftarrow \mathsf{Verify}(vk, \sigma_m, m)$: The verification algorithm takes a verification key $vk$, a signature $\sigma_m$, and a message $m$, and returns 1(accept) or $\perp$(reject).

The digital signature scheme satisfies correctness and unforgeability.
**Correctness**: For all $m$, the following probability is 1.

$$Pr\left[\mathsf{Verify}(vk, \sigma_m, m) = 1 \middle| \begin{array}{c}(vk, sk) \leftarrow \mathsf{Setup}(1^\lambda) \\ \sigma_m \leftarrow \mathsf{Sign}(vk, m)\end{array}\right] = 1$$

**Unforgeability**: For every PPT adversary $\mathcal{A}$ the following probability is $negl(1^\lambda)$:

$$Pr\left[\begin{array}{c}\mathsf{Verify}(crs, \sigma^*_m, m^*) = 1 \\ \wedge (\sigma^*_m, m^*) \notin \mathcal{Q}\end{array} \middle| \begin{array}{c}(vk, sk) \leftarrow \mathsf{Keygen}(1^\lambda) \\ (\sigma^*_m, m^*) \leftarrow \mathcal{A}^{O_{Sign}}(vk)\end{array}\right] = negl(1^\lambda)$$

where $O_{Sign}(m_i)$ returns $\sigma_{m_i} \leftarrow \mathsf{Sign}(sk, m_i)$ and adds $(\sigma_{m_i}, m_i)$ to $\mathcal{Q}$.

## 2.5 Simulation-Extractable Commit and Prove SNARK

The simulation-extractable commit and prove SNARK (se-cp-SNARK) scheme is a zk-SNARK scheme to prove the knowledge of $(\phi, w)$ such that $u$ is a message of commitment $c$ and a relation $R(\phi, w) = 1$ where the witness $u \in w$. Its definition follows LegoSNARK's cp-SNARK definition [3], except for simulation extractability, which allows attackers to access a simulated proof oracle. Since it is knowledge sound even if an attacker accesses the simulated proof oracle, it provides non-malleability.

*Definition 2.3.* A se-cp-SNARK scheme includes the quadruple PPT algorithms (KeyGen, Prove, Verify, Sim) defined as follows.

- $(crs, td) \leftarrow \mathsf{Setup}(\mathbf{ck}, R)$: The setup algorithm takes a relation $R \in \mathcal{R}_\lambda$ and commitment key ck as input, and returns a common reference string $crs$ and a trapdoor $td$.

- $\pi \leftarrow \mathsf{Prove}(crs, \phi, \{c_j, u_j, o_j\}_{j=1}^{l}, w)$: The prover algorithm takes as input a $crs$ for a relation $R$, $(\phi, w) \in R$, commitments $c_j$, inputs $u_j$ and opening $o_j$, and returns a proof $\pi$.
- $0/1 \leftarrow \mathsf{Verify}(crs, \phi, \{c_j\}_{j=1}^{l}, \pi)$: The verifier algorithm takes as input a $crs$, a statement $\phi$, commitments $c_j$ and a proof $\pi$, and returns 0 (reject) or 1 (accept).
- $\pi \leftarrow \mathsf{Sim}(crs, td, \phi, \{c_j\}_{j=1}^{l})$: The simulator algorithm takes a $crs$, a trapdoor $td$, a statement $\phi$, and commitments $c_j$ as input, and returns a proof $\pi$.

The se-cp-SNARK satisfies the correctness, succinctness, simulation-extractability, and zero-knowledge.

**Completeness**: An argument is complete if given true statement $\phi$, a prover with a witness can convince the verifier. For all $(\phi, w) \in R$, the probability of completeness is 1:

$$Pr\left[ \mathsf{Verify}(crs, \phi, \{c_j\}_{j=1}^{l}, \pi) = 1 \middle| \begin{array}{c} (crs, td) \leftarrow \mathsf{Setup}(\mathbf{ck}, \mathcal{R}), \\ \pi \leftarrow \mathsf{Prove}(crs, \phi, \{c_j, u_j, o_j\}_{j=1}^{l}, w) \end{array} \right] = 1$$

**Succinctness**: $\Pi_{cc}$ is said succinct if the running time of Verify is $poly(\lambda)(\lambda + |\phi| + \log|w|)$ and the size of the proof is $poly(\lambda)(\lambda + \log|w|)$

**Simulation-Extractability**: An argument is simulation-extractable if the prover must know a witness and such knowledge can be efficiently extracted from the prover by using a knowledge extractor. Simulation-Extractability requires that for a PPT adversary $\mathcal{A}$ generating an accepting proof with the oracle $O_{Sim}$, there must be an extractor $\chi_{\mathcal{A}}$ that, given the same input of $\mathcal{A}$, outputs a valid witness such that

$$Pr\left[ \begin{array}{c} \mathsf{Verify}(crs, \phi, \{c_j\}_{j=1}^{l}, \pi) = 1 \\ \wedge(\phi, \pi, \{c_j\}_{j=1}^{l}) \in \mathcal{Q} \\ \wedge \mathcal{R}(\phi, w) = 0 \end{array} \middle| \begin{array}{c} (crs, td, z) \leftarrow \mathsf{Setup}(\mathcal{R}), \\ (\phi, \{c_j\}_{j=1}^{l}, \pi) \leftarrow \mathcal{A}^{O_{Sim}}(R, crs, z) \\ (\{u_j, o_j\}_{j=1}^{l}, w) \leftarrow \chi_{\mathcal{A}}(R, crs, z) \end{array} \right] = negl(1^{\lambda})$$

where

- $z$ is auxiliary input.
- $O_{Sim}(crs, td, \phi_i, \{c_j\}_{j=1}^{l})$ returns a proof $\pi_i \leftarrow \mathsf{Sim}(crs, td, \phi_i, \{c_j\}_{j=1}^{l})$ and adds $(\phi_i, \pi_i, \{c_j\}_{j=1}^{l})$ to $\mathcal{Q}$.

**Zero-Knowledge**: A scheme $\Pi_{cc}$ has zero-knowledge for a relation $R$ if for every adversary $\mathcal{A}$ there exists a simulator Sim such that both following conditions hold for all adversaries $\mathcal{A}$:

$$Pr\left[ \mathcal{A}(crs, z, \phi, \{c_j\}_{j=1}^{l}, \pi) = 1 \middle| \begin{array}{c} (crs, td, z) \leftarrow \mathsf{Setup}(\mathbf{ck}, \mathcal{R}), \\ \wedge \mathcal{R}(\phi, w) = 1 \end{array} \middle| \pi \leftarrow \mathsf{Prove}(crs, \phi, \{c_j, u_j, o_j\}_{j=1}^{l}, w) \right]$$
$$\approx Pr\left[ \mathcal{A}(crs, z, \phi, \{c_j\}_{j=1}^{l}, \pi) = 1 \middle| \begin{array}{c} (crs, td, z) \leftarrow \mathsf{Sim}_{\mathsf{Setup}}(\mathbf{ck}, \mathcal{R}), \\ \wedge \mathcal{R}(\phi, w) = 1 \end{array} \middle| (\pi) \leftarrow \mathsf{Sim}(crs, td, \phi, \{c_j, u_j, o_j\}_{j=1}^{l}) \right]$$

where $z$ is auxiliary input.

## 2.6 Simulation-Extractable Commit and Carry SNARK

Similar to the case of se-cp-SNARK, the simulation-extractable commit and carry SNARK (se-cc-SNARK) schemes proves a relation with commitment, but it generates a commitment while proving the relation.

*Definition 2.4.* The se-cc-SNARK scheme has the quintuple of PPT algorithms $\Pi_{cc}$ = (KeyGen, Prove, Verify, VerifyCom, Sim) defined as follows.

- $(\mathbf{ck}, crs, td) \leftarrow \mathsf{Setup}(R)$: The setup algorithm takes as input a relation $R \in \mathcal{R}_{\lambda}$, and returns a commitment key $\mathbf{ck}$, a $crs$, and a simulation trapdoor $td$.
- $(\overline{C_u}, \pi, r) \leftarrow \mathsf{Prove}(crs, u, \overline{u}, w)$: The prover algorithm takes as a $crs$ for a relation $R$ and $((u, \overline{u}), w) \in R$, and returns a commitment $\overline{C_u}$ where the input is $\overline{u}$, a proof $\pi$, and an opening $r$.
- $0/1 \leftarrow \mathsf{Verify}(crs, u, \overline{C_u}, \pi)$: The verifier algorithm takes as input a $crs$, a statement $u$, commitments $\overline{C_u}$ and a proof $\pi$, and returns 0(reject) or 1(accept).
- $0/1 \leftarrow \mathsf{VerifyCom}(\mathbf{ck}, \overline{C_u}, \overline{u}, r)$: The verifier algorithm takes as input a commitment key $\mathbf{ck}$, a commitments $\overline{C_u}$, a message $\overline{u}$, and an opening $r$, and returns 0(reject) or 1(accept).
- $(\overline{C_u}, \pi, r) \leftarrow \mathsf{Sim}(crs, td, \phi)$: The simulator algorithm takes as a $crs$, a simulation trapdoor $td$, and a statement $u$, and returns a commitment $\overline{C_u}$, a proof $\pi$, and an opening $r$.

The se-cc-SNARK satisfies the correctness, succinctness, simulation-extractability, zero-knowledge, and binding.

**Completeness**: An argument is complete if given true statement $\phi$, a prover with a witness can convince the verifier. For all $(u, \overline{u}, w) \in R$, the probability of completeness is:

$$Pr\left[ \mathsf{Verify}(crs, \phi, cm, \pi) = 1 \middle| \begin{array}{c} (\mathbf{ck}, crs, td) \leftarrow \mathsf{Setup}(\mathcal{R}), \\ (\overline{C_u}, \pi, r) \leftarrow \mathsf{Prove}(crs, u, \overline{u}, w) \end{array} \right] = 1$$

**Succinctness**: $\Pi_{cc}$ is said succinct if the running time of Verify is $poly(\lambda)(\lambda + |u| + |\overline{u}| + \log|w|)$ and the size of the proof is $poly(\lambda)(\lambda + \log|w|)$

**Simulation-Extractability**: An argument is simulation-extractable if the prover must know a witness and such knowledge can be efficiently extracted from the prover by using a knowledge extractor. Simulation-Extractability requires that for a PPT adversary $\mathcal{A}$ generating an accepting proof with the oracle $O_{Sim}$, there must be an extractor $\chi_{\mathcal{A}}$ that, given the same input of $\mathcal{A}$, outputs a valid witness such that

$$Pr\left[ \begin{array}{c} \mathsf{Verify}(crs, u, \overline{C_u}, \pi) = 1 \wedge (\phi, \pi) \in \mathcal{Q} \\ \wedge(\mathsf{VerifyCom}(\mathbf{ck}, \overline{C_u}, \overline{u}, r) = 0 \\ \vee \mathcal{R}((u, \overline{u}), w) = 0) \end{array} \middle| \begin{array}{c} (crs, td, z) \leftarrow \mathsf{Setup}(\mathcal{R}), \\ (u, \overline{C_u}, \pi) \leftarrow \mathcal{A}^{O_{Sim}}(R, crs, z) \\ (r, w) \leftarrow \chi_{\mathcal{A}}(R, crs, z) \end{array} \right] = negl(1^{\lambda})$$

where

- $z$ is auxiliary input.
- $O_{Sim}(crs, td, (u_i, \overline{u_i}))$ returns a proof $\pi_i \leftarrow \mathsf{Sim}(crs, td, (u_i, \overline{u_i}))$ and adds $((u_i, \overline{u_i}), \pi_i)$ to $\mathcal{Q}$.

**Zero-Knowledge**: A scheme $\Pi_{cc}$ has zero-knowledge for a relation $R$ if for every adversary $\mathcal{A}$ there exists a simulator Sim such that both following conditions hold for all adversaries $\mathcal{A}$:

$$Pr\left[\begin{array}{c}\mathcal{A}(crs,z,cm,\pi)=1\\ \wedge\mathcal{R}(\phi,w)=1\end{array}\middle|\begin{array}{c}(\mathbf{ck},crs,td,z)\leftarrow \mathsf{Setup}(\mathcal{R}),\\ (cm,\pi,r)\leftarrow \mathsf{Prove}(crs,\phi,w)\end{array}\right]$$
$$\approx Pr\left[\begin{array}{c}\mathcal{A}(crs,z,cm,\pi)=1\\ \wedge\mathcal{R}(\phi,w)=1\end{array}\middle|\begin{array}{c}(\mathbf{ck},crs,td,z)\leftarrow \mathsf{Sim}_{\mathsf{Setup}}(\mathcal{R}),\\ (cm,\pi)\leftarrow \mathsf{Sim}(crs,td,\phi)\end{array}\right]$$

where $z$ is auxiliary input.

**Binding**: For every polynomial-time adversary $\mathcal{A}$ the following probability is $negl(1^\lambda)$:

$$Pr\left[\begin{array}{c}\mathsf{VerifyCom}(\mathbf{ck},\overline{C_u},\overline{u},r)\wedge\\ \mathsf{VerifyCom}(\mathbf{ck},\overline{C_u},\overline{u'},r')\wedge\\ u\neq u'\end{array}\middle|\begin{array}{c}(\mathbf{ck},crs,td,z)\leftarrow \mathsf{Setup}(\mathcal{R}),\\ (\overline{C_u},\overline{u},r,\overline{u'},r')\leftarrow \mathcal{A}(\mathcal{R},crs,z)\end{array}\right]$$

where $z$ is auxiliary input.

# 3 VERIFIABLE IMAGE REDACTING SCHEME

## 3.1 Definition

In this section, we define a verifiable image redacting scheme and its security notions. The scheme can be divided into an initializing phase and a redacting phase. In the initializing phase, the certificate authority authenticates the original image, by generating a commitment and its signature. A private key $sk$ and a public key $pk$ are generated using the AuthSetup algorithm. Then, the Authenticate algorithm commits to and signs the original image $\mathbf{m}$ with the private key $sk$. In the redacting phase, the image is redacted and the Prove algorithm generates a proof of the redacting process. The prover splits the original image $\mathbf{m}$ into a public image $\mathbf{u}$ and a private image $\overline{\mathbf{u}}$. Note that the Prove algorithm proves that $\mathbf{m}=\mathbf{u}+\overline{\mathbf{u}}$. Lastly, anyone can verify the image authenticity using the Verify algorithm without private image $\overline{\mathbf{u}}$. The complete definition is as follows:

*Definition 3.1.* A verifiable image redacting scheme includes the algorithms (AuthSetup, Authenticate, ProofSetup, Prove, Verify, Sim) defined as follows:

- $(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)$: The authenticate key generation algorithm takes as input security parameter $\lambda$, and returns a public key $pk$ and a private key $sk$.
- $(C_m,\sigma_m,o)\leftarrow \mathsf{Authenticate}(sk,\mathbf{m})$: The authentication algorithm takes as input a private key $sk$ and a message vector $\mathbf{m}$, and returns a commitment $C_m$, a certificate $\sigma_m$, and a commitment opening $o$.
- $(crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})$: The setup algorithm takes a relation $\mathcal{R}$ and a public key $pk$ as input, and returns a common reference string $crs$ and a trapdoor $td$.
- $(\pi,\overline{C_u},r)\longleftarrow \mathsf{Prove}(crs,C_m,\mathbf{u};\overline{\mathbf{u}},m_0)$: The prover algorithm takes a $crs$ for a relation $R$, a commitment $C_m$, a public image $\mathbf{u}$, a private image $\overline{\mathbf{u}}$, and opening $m_0$ as input, and returns a proof $\pi$, a private image commitment $\overline{C_u}$, and an opening $r$ for private image.
- $(1/\perp)\longleftarrow \mathsf{Verify}(crs,pk,C_m,\sigma_m,\mathbf{u},(\pi,\overline{C_u}))$: The verifier algorithm takes a $crs$, a public key $pk$, a commitment $C_m$, a signature $\sigma_m$, a public image $\mathbf{u}$, a proof $\pi$, and a private image commitment $\overline{C_u}$ as input, and returns 1 (accept) or $\perp$ (reject).
- $(\pi,\overline{C_u},r)\longleftarrow \mathsf{Sim}(crs,td,C_m,\mathbf{u})$: The simulation algorithm takes a $crs$, a trapdoor $td$, a commitment $C_m$, and a public image $\mathbf{u}$ as input, and returns a proof $\pi$, a private image commitment $\overline{C_u}$, and an opening $r$ for the private.

A verifiable image redacting scheme must satisfy the following properties:

**Completeness** : For all $((\overline{C_u},C_m,\mathbf{u}),(r,m_0,\overline{\mathbf{u}},\mathbf{m}))\in\mathcal{R}$, the following probability is 1.

$$Pr\left[\mathsf{Verify}\begin{pmatrix}crs,\mathbf{u}^*,C_{m^*},\\ \sigma_m,(\pi,\overline{C_{u^*}})\end{pmatrix}=1\middle|\begin{array}{c}(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)\\ (C_m,\sigma_m,m_0)\leftarrow \mathsf{Authenticate}(sk,\mathbf{m})\\ (crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})\\ (\pi,\overline{C_u},r)\leftarrow \mathsf{Prove}(crs,C_m,\mathbf{u};\overline{\mathbf{u}},m_0)\end{array}\right]=1$$

**Redacting correctness** : For all PPT adversaries $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that the following probability is negligible with security parameter $\lambda$.

$$Pr\left[\begin{array}{c}\mathsf{Verify}\begin{pmatrix}crs,\mathbf{u}^*,C_{m^*},\\ \sigma_{m^*},(\pi,\overline{C_{u^*}})\end{pmatrix}=1\\ \wedge\mathcal{R}\begin{pmatrix}(\overline{C_{u^*}},C_{m^*},\mathbf{u}^*),\\ (r^*,m_0^*,\overline{\mathbf{u}}^*,\mathbf{m}^*)\end{pmatrix}=0\\ \wedge\mathbf{m}^*\in\mathcal{M}\end{array}\middle|\begin{array}{c}(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)\\ (crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})\\ \begin{pmatrix}\pi,C_{m^*},\overline{C_{u^*}},\\ r^*,\mathbf{u}^*\end{pmatrix}\leftarrow \mathcal{A}^{O_A,O_P}(\mathcal{R},crs,pk,z)\\ (m_0^*,\mathbf{m}^*,\overline{\mathbf{u}}^*)\leftarrow \chi_{\mathcal{A}}(\mathcal{R},crs,pk,z)\end{array}\right]=negl(1^\lambda)$$

where

- $z$ is auxiliary input.
- $O_A(\mathbf{m_i})$ returns $(C_{m_i},\sigma_{m,i},m_{0,i})\leftarrow \mathsf{Authenticate}(sk,\mathbf{m_i})$ and adds $(\mathbf{m_i},C_{m_i},\sigma_{m,i},m_{0,i})$ to $\mathcal{M}$.
- $O_P(cm_i,\mathbf{u_i};\overline{\mathbf{u_i}},m_{0,i})$ returns $(\pi_i,\overline{C_{u_i}},r_i)\leftarrow \mathsf{Prove}(crs,C_{m_i},\mathbf{u_i};\overline{\mathbf{u_i}},m_{0,i})$.

**Originality** : For all PPT adversaries $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that the following probability is negligible with security parameter $\lambda$.

$$Pr\left[\begin{array}{c}\mathsf{Verify}\begin{pmatrix}crs,\mathbf{u}^*,C_{m^*},\\ \sigma_{m^*},(\pi,\overline{C_{u^*}})\end{pmatrix}=1\\ \wedge\mathcal{R}\begin{pmatrix}(\overline{C_{u^*}},C_{m^*},\mathbf{u}^*),\\ (r^*,m_0^*,\overline{\mathbf{u}}^*,\mathbf{m}^*)\end{pmatrix}=1\\ \wedge\mathbf{m}^*\notin\mathcal{M}\end{array}\middle|\begin{array}{c}(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)\\ (crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})\\ \begin{pmatrix}\pi,C_{m^*},\overline{C_{u^*}},\\ r^*,\mathbf{u}^*\end{pmatrix}\leftarrow \mathcal{A}^{O_A,O_P}(R,crs,pk,z)\\ (m_0^*,\mathbf{m}^*,\overline{\mathbf{u}}^*)\leftarrow \chi_{\mathcal{A}}(\mathcal{R},crs,pk,z)\end{array}\right]=negl(1^\lambda)$$

where

- $z$ is auxiliary input.
- $O_A(\mathbf{m_i})$ returns $(C_{m_i},\sigma_{m,i},m_{0,i})\leftarrow \mathsf{Authenticate}(sk,\mathbf{m_i})$ and adds $(\mathbf{m_i},C_{m_i},\sigma_{m,i},m_{0,i})$ to $\mathcal{M}$.
- $O_P(crs,C_{m_i},\mathbf{u_i};\overline{\mathbf{u_i}},m_{0,i})$ returns $(\pi_i,\overline{C_{u_i}},r_i)\leftarrow \mathsf{Prove}(crs,C_{m_i},\mathbf{u_i};\overline{\mathbf{u_i}},m_{0,i})$.

**Area zero-knowledge** : For all PPT adversaries $\mathcal{A}$, the following two probabilities are statically close.

$$Pr\left[\begin{array}{c}\mathcal{A}\begin{pmatrix}crs,pk,cm,\sigma_m,\\ \overline{C_u},\mathbf{u},\pi,z\end{pmatrix}=1\\ \wedge\mathcal{R}\begin{pmatrix}(\overline{C_u},C_m,\mathbf{u}),\\ (r,m_0,\overline{\mathbf{u}},\mathbf{m})\end{pmatrix}=1\end{array}\middle|\begin{array}{c}(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)\\ (\sigma_m,C_m,m_0)\leftarrow \mathsf{Authenticate}(sk,\mathbf{m})\\ (crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})\\ (\pi,\overline{C_u},r)\leftarrow \mathsf{Prove}(crs,C_m,\mathbf{u};\overline{\mathbf{u}},m_0)\end{array}\right]$$
$$\approx Pr\left[\begin{array}{c}\mathcal{A}\begin{pmatrix}crs,pk,cm,\sigma_m,\\ \overline{C_u},\mathbf{u},\pi,z\end{pmatrix}=1\\ \wedge\mathcal{R}\begin{pmatrix}(\overline{C_u},C_m,\mathbf{u}),\\ (r,m_0,\overline{\mathbf{u}},\mathbf{m})\end{pmatrix}=1\end{array}\middle|\begin{array}{c}(pk,sk)\leftarrow \mathsf{AuthSetup}(1^\lambda)\\ (\sigma_m,C_m,m_0)\leftarrow \mathsf{Authenticate}(sk,\mathbf{m})\\ (crs,td)\leftarrow \mathsf{ProofSetup}(pk,\mathcal{R})\\ (\pi,\overline{C_u},r)\leftarrow \mathsf{Sim}(crs,C_m,\sigma_m,\mathbf{u},td)\end{array}\right]$$

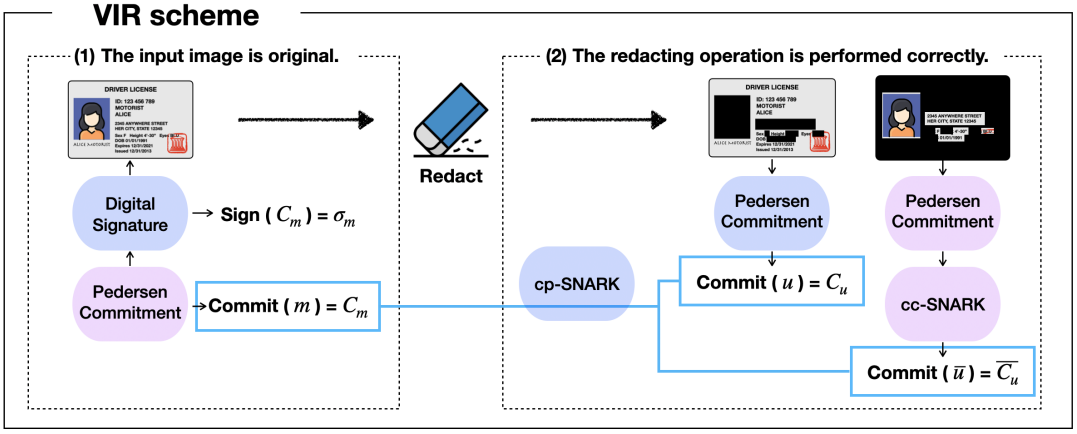



**Figure 3: Verifiable image redacting scheme**

## 3.2 Proposed VIR Scheme

In this section, we provide an efficient construction of verifiable image redacting (VIR) using se-cp-SNARK, se-cc-SNARK, a commitment, and a digital signature. The VIR construction should achieve the *originality* and the *redacting correctness* for its soundness and the *area zero-knowledge* for its privacy protection. Figure 3 overviews the system design of the VIR scheme. The originality is proved through the signing process in the Authenticate algorithm. The digital signature scheme signs on a hashed message in general. In our scheme, we use the Pedersen commitment scheme for the hash function and allows public verifiability of the signature on the committed message. The main reason of using the Pedersen commitment is to apply cp-SNARK. The *originality* is satisfied by the computational binding of the Pedersen commitment and unforgeability of digital signature scheme. Secondly, the redacting correctness is proved through Prove algorithm of the redacting phase. After executing a redacting operation, the prover commits each public and private image, respectively. To publicly verify the legitimacy of the private image, its commitment $\overline{C_u}$ is generated through the cc-SNARK technique. The cp-SNARK proves that $m_i = u_i + \overline{u_i}$ in the commitment form. In the cc-SNARK, the relation that $u_i \cdot \overline{u_i} = 0$ is included to specify that $u_i$ or $\overline{u_i}$ is zero. For the non-malleability of zk-SNARK proofs, simulation extractability is required. We devise se-cp-SNARK and se-cc-SNARK for VIR based on se-qa-NIZK [10], and se-SNARK [11], respectively.

Let $\Pi_{cp}$ = (Setup, Prove, Verify, Sim) be a se-cp-SNARK scheme, $\Pi_{cc}$ = (Setup, Prove, Verify, VerifyCommit, Sim) be a se-cc-SNARK scheme, $\Pi_\sigma$ = (Keygen, Sign, Verify) be a digital signature scheme, and $\Pi_{cm}$ = (Keygen, Commit, VerifyCommit) be a Pedersen vector commitment scheme. Algorithm 2 describes the VIR construction using $\Pi_{cp}$, $\Pi_{cc}$, $\Pi_\sigma$, and $\Pi_{cm}$. Note that we provide the concrete constructions of $\Pi_{cp}$ in Section 3.3 and $\Pi_{cc}$ in Section 3.4. The redacting correctness is guaranteed by the computational knowledge soundness property of zk-SNARKs and the area zero-knowledge is satisfied by the perfect hiding of Pedersen commitment and zero-knowledge property of zk-SNARKs.

Theorem 3.2. *Assuming that a se-cp-SNARK scheme $\Pi_{cp}$ and a se-cc-SNARK scheme $\Pi_{cc}$ satisfy simulation extractability with negligible probability $\epsilon_{cp}$ and $\epsilon_{cc}$, the VIR scheme satisfies redacting correctness with negligible error $\epsilon \leq \epsilon_{cp} + \epsilon_{cc}$.*

Sketch of proof. If there is an adversary $\mathcal{A}$ to break the redacting correctness with a non-negligible probability, there exists a queried message $\mathbf{m}^*$ that is accepted by the Verify algorithm but does not satisfy the relation $R$. It means that $\mathcal{A}$ breaks either the $\Pi_{cp}$ or $\Pi_{cc}$ soundness with a non-negligible probability. However, the computational knowledge soundness errors for each scheme $\Pi_{cp}$ and $\Pi_{cc}$ are $\epsilon_{cp}$ and $\epsilon_{cc}$, respectively, which are negligible. Therefore the $\mathcal{A}$ can break the redacting correctness of the VIR scheme with probability $\epsilon \leq \epsilon_{cp} + \epsilon_{cc}$, which is negligible. □

The proof is available in Appendix A.1

Theorem 3.3. *Assuming that the commitment scheme $\Pi_{cm}$ satisfies computational binding with negligible error $\epsilon_{cm}$ and the digital signature scheme $\Pi_\sigma$ satisfies satisfies unforgeability with negligible error $\epsilon_\sigma$, the VIR scheme satisfies originality with negligible error $\epsilon \leq \epsilon_{cm} + \epsilon_\sigma$.*

Sketch of proof. If there is an adversary $\mathcal{A}$ to break the unforgeability with a non-negligible probability, there exists a non queried message $\mathbf{m}^*$ that is accepted by the Verify algorithm and satisfies the relation $R$. It is possible only if $\mathcal{A}$ breaks either the binding property or the unforgeability property, which contradicts the assumption. Therefore $\mathcal{A}$ can break originality of the VIR scheme with probability $\epsilon \leq \epsilon_{cm} + \epsilon_\sigma$, which is negligible. □

The proof is available in Appendix A.2

Theorem 3.4. *Assuming that a cp-SNARK scheme $\Pi_{cp}$ and a cc-SNARK scheme $\Pi_{cc}$ satisfy zero-knowledge, and the commitment scheme $\Pi_{cm}$ satisfies perfect hiding, the VIR scheme satisfies area zero-knowledge.*

Proof. $\Pi_{cm}$ has the perfect hiding property, so the commitment $cm$ has no information about private image. Also, since the $\Pi_{cp}$ and $\Pi_{cc}$ has zero-knowledge property, the proof $\pi$, $\overline{C_m}$, and public image $\mathbf{u}$ have no information. Formally, since there exists Sim algorithm which can generate a valid proof $\pi$, $\overline{C_m}$ without

**Algorithm 2** Verifiable image redacting (VIR)

$(pk, sk) \longleftarrow \mathsf{AuthSetup}(1^\lambda)$
$\mathbf{ck} \longleftarrow \Pi_{cm}.\mathsf{Keygen}(1^\lambda)$
$vk_\sigma, sk_\sigma \longleftarrow \Pi_\sigma.\mathsf{Keygen}(1^\lambda)$
$pk = (\mathbf{ck}, vk_\sigma)$
$sk = (\mathbf{ck}, sk_\sigma)$
return $(pk, sk)$

$(\sigma_m, C_m, m_0) \longleftarrow \mathsf{Authenticate}(\mathbf{m}, sk)$
parse $sk = (\mathbf{ck}, sk_\sigma)$
$(C_m, m_0) \longleftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck}, \mathbf{m})$
$\sigma_m \longleftarrow \Pi_\sigma.\mathsf{Sign}(C_m, sk_\sigma)$
return $(\sigma_m, C_m, m_0)$

$(crs, td) \longleftarrow \mathsf{ProofSetup}(pk, \mathcal{R}_{redact} \wedge \mathcal{R}_{zero})$
$(\mathbf{ck_{cc}}, crs_{cc}, td_{cc}) \longleftarrow \Pi_{cc}.\mathsf{Setup}(\mathcal{R}_{zero})$
parse $\mathbf{ck_{cc}} = (f_0, f_1, \ldots, f_{2n})$
parse $pk = (\mathbf{ck}, vk_\sigma)$
set $\mathbf{h} = \mathbf{ck} = (ck_0, \ldots, ck_n)$, $\mathbf{f} = (f_0, f_1, \ldots, f_n)$, and $\mathbf{f'} = (f_0, f_{n+1}, \ldots, f_{2n})$
$(crs_{cp}, td_{cp}) \longleftarrow \Pi_{cp}.\mathsf{Setup}(\mathbf{h}, \mathbf{f}, \mathbf{f'}, \mathcal{R}_{redact})$
$crs = (\mathbf{ck_{cc}}, crs_{cc}, crs_{cp})$
$td = (td_{cc}, td_{cp})$
return $(crs, td)$

$(\overline{C_m}, \pi, r') \longleftarrow \mathsf{Prove}(crs, C_m, \mathbf{u}; \overline{\mathbf{u}}, m_0)$
parse $crs = (\mathbf{ck_{cc}}, crs_{cc}, crs_{cp})$
$(\overline{C_u}, \pi_1, r') \longleftarrow \Pi_{cc}.\mathsf{Prove}(crs_{cc}, \mathbf{u}, \overline{\mathbf{u}};\ )$
parse $\mathbf{ck_{cc}} = (f_0, f_1, \ldots, f_{2n})$ and set $\mathbf{ck'} = (1, f_1, \ldots, f_n)$
$(C_u, r) \longleftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck'}, \mathbf{u})$
$\pi_2 \longleftarrow \Pi_{cp}.\mathsf{Prove}(crs_{cp}, (C_u, \overline{C_u}, C_m); (0, r', m_0, \mathbf{u}, \overline{\mathbf{u}}))$
$\pi = (\pi_1, \pi_2)$
return $(\overline{C_u}, \pi, r')$

$(1/\perp) \longleftarrow \mathsf{Verify}(crs, pk, \mathbf{u}, cm, (\pi, \overline{C_u}))$
parse $crs = (\mathbf{ck_{cc}}, crs_{cc}, crs_{cp})$ and $\pi = (\pi_1, \pi_2)$
parse $\mathbf{ck_{cc}} = (f_0, f_1, \ldots, f_{2n})$ and set $\mathbf{ck'} = (1, f_1, \ldots, f_n)$
$(C_u, r) \longleftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck'}, \mathbf{u})$
parse $pk = (\mathbf{ck}, vk_\sigma)$
assert $\Pi_\sigma.\mathsf{Verify}(vk_\sigma, C_m, \sigma_m) = 1$
assert $\Pi_{cc}.\mathsf{Verify}(crs_{cc}, \mathbf{u}, \overline{C_u}, \pi_1) = 1$
assert $\Pi_{cp}.\mathsf{Verify}(crs_{cp}, (C_u, \overline{C_u}, C_m), \pi_2) = 1$
return 1

$(\overline{C_u}, \pi, r') \longleftarrow \mathsf{Sim}(crs, td, cm, \mathbf{u})$
parse $crs = (\mathbf{ck_{cc}}, crs_{cc}, crs_{cp})$
parse $td = (td_{cc}, td_{cp})$
$(\overline{C_u}, \pi_1, r') \longleftarrow \Pi_{cc}.\mathsf{Sim}(crs_{cc}, td_{cc}, \mathbf{u})$
parse $\mathbf{ck_{cc}} = (f_0, f_1, \ldots, f_{2n})$ and set $\mathbf{ck'} = (1, f_1, \ldots, f_n)$
$(C_u, r) \longleftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck'}, \mathbf{u})$
$\pi_2 = \Pi_{cp}.\mathsf{Sim}(crs_{cp}, td_{cp}, C_u, \overline{C_u}, C_m)$
$\pi = (\pi_1, \pi_2)$
return $(\overline{C_u}, \pi, r')$

the corresponding private image, the VIR scheme satisfies area zero-knowledge property. □

## 3.3 Simulation-extractable commit and prove SNARK for VIR

We adopt the existing se-cp-SNARK (simulation-extractable commit and prove SNARK) scheme in [3, 10] for VIR in Algorithm 3. The matrix $\mathbf{M}$ is set to commitment keys for original ($\mathbf{h}$), public ($\mathbf{f}$), and private image ($\mathbf{f'}$):

$$\mathbf{M} = \begin{bmatrix} f_0 & 0 & 0 & f_1 & \cdots & f_n & 0 & \cdots & 0 \\ 0 & f_0 & 0 & 0 & \cdots & 0 & f_{n+1} & \cdots & f_{2n} \\ 0 & 0 & h_0 & h_1 & \cdots & h_n & h_1 & \cdots & h_n \end{bmatrix}$$

And this linear operation is proved by se-qa-NIZK [10]. We provide the concrete construction in Algorithm 3.

**Algorithm 3** Simulation-extractable commit and prove SNARK for VIR

$(crs, td) \longleftarrow \mathsf{Setup}(\mathbf{h}, \mathbf{f}, \mathbf{f'}, \mathcal{R}_{redact})$
parse $\mathbf{h} = (h_0, \cdots, h_n)$, $\mathbf{f} = (f_0, f_1, \cdots, f_n)$, and $\mathbf{f'} = (f_0, f_{n+1}, \cdots, f_{2n})$
$\mathbf{k} \xleftarrow{\$} \mathbb{Z}_q^3$
$(k_0^*, k_1^*) \xleftarrow{\$} \mathbb{Z}_q^2$
$\mathbf{M} = \begin{bmatrix} f_0 & 0 & 0 & f_1 & \cdots & f_n & 0 & \cdots & 0 \\ 0 & f_0 & 0 & 0 & \cdots & 0 & f_{n+1} & \cdots & f_{2n} \\ 0 & 0 & h_0 & h_1 & \cdots & h_n & h_1 & \cdots & h_n \end{bmatrix}$
$\mathbf{P} = \mathbf{M}^\mathsf{T} \times \mathbf{k}$
$a \xleftarrow{\$} \mathbb{G}_2$ in $\mathcal{D}_k$
$b \xleftarrow{\$} \mathbb{G}_1$ in $\mathcal{D}_k$
$(d_0, d_1) = (a^{k_0^*}, a^{k_1^*})$
$(p_0, p_1) = (b^{k_0^*}, b^{k_1^*})$
$\mathbf{c} = (\mathbf{k} \cdot a)$
$\tau \xleftarrow{\$} \mathbb{Z}_q$
$crs = (\mathbf{P}, \mathbf{c}, a, b, d_0, d_1, p_0, p_1, \tau)$
$td = \mathbf{k}$
return $(crs, td)$

$\pi \longleftarrow \mathsf{Prove}(crs, C_u, \overline{C_u}, C_m; r, r', m_0, \mathbf{u}, \overline{\mathbf{u}})$
parse $crs = (\mathbf{P}, \mathbf{c}, a, b, d_0, d_1, p_0, p_1, \tau)$
$s \xleftarrow{\$} \mathbb{Z}_p$
$\mathbf{w} = (r, r', m_0, \mathbf{u}, \overline{\mathbf{u}})$
$\pi = (\mathbf{w} \times \mathbf{P} \cdot (p_0 \cdot p_1^\tau)^s, b^s)$
return $\pi$

$(1/\perp) \longleftarrow \mathsf{Verify}(crs, C_u, C_m, (\pi, \overline{C_u}))$
parse $crs = (\mathbf{P}, (c_0, c_1, c_2), a, b, d_0, d_1, p_0, p_1, \tau)$
parse $\pi = (\pi_1, \pi_2)$
assert $e(\pi_1, a) = e(C_u, c_0) \cdot e(\overline{C_u}, c_1) \cdot e(C_m, c_2) \cdot e(\pi_2, d_0 \cdot d_1^\tau)$
return 1

$\pi \longleftarrow \mathsf{Sim}(crs, td, C_u, \overline{C_u}, C_m)$
parse $crs = (\mathbf{P}, (c_0, c_1, c_2), a, b, d_0, d_1, p_0, p_1, \tau)$
parse $td = (k_0, k_1, k_2)$
$s \xleftarrow{\$} \mathbb{Z}_p$
$\pi = (C_u^{k_0} \cdot \overline{C_u}^{k_1} \cdot C_m^{k_2} \cdot (p_0 \cdot p_1^\tau)^s, b^s)$
return $\pi$

Theorem 3.5. *Assuming that the $\mathcal{D}_k - MDDH$ assumption in $\mathbb{G}_1$ holds and the $\mathcal{D}_k - KerMDH$ assumption in $\mathbb{G}_2$, the se-cp-SNARK for VIR satisfies simulation extractability and zero-knowledge [10].*

### 3.4 Simulation-extractable commit carrying SNARK for VIR

We build se-cc-SNARK (simulation-extractable commit carrying SNARK) for VIR utilizing [3] and [11] in Algorithm 4. Since the I/O part in the verification is similar to the Pedersen vector commitment, it is calculated in the proof instead of in the verification. The difference between this operation and commitment is that there is no opening part for the hiding property, so CRS ($\eta$ parts) and the opening ($\nu$) are added for this.

**Algorithm 4** Simulation-extractable commit carrying SNARK for VIR

$(crs, td) \longleftarrow \mathsf{Setup}(\mathcal{R}_{zero})$
$\alpha, \beta, \gamma, \delta, x, \eta \xleftarrow{\$} \mathbb{Z}_p^{*6}$.
$crs = (G_1^{\alpha}, G_1^{\beta}, G_1^{\delta}, G_1^{\alpha\delta}, G_1^{\eta}, G_1^{\eta\gamma}, \{G_1^{\gamma x^i}\}_{i=0}^{d_x-1}, \{G_1^{\gamma\delta x^i}\}_{i=0}^{d_x-1},$
$\{G_1^{\beta u_i(x)+\alpha v_i(x)+\gamma w_i(x)}\}_{i=0}^{l},$
$\{G_1^{\beta\gamma u_i(x)+\alpha\gamma v_i(x)+\gamma^2 w_i(x)}\}_{i=l+1}^{m},$
$\{G_1^{\gamma^2 x^i \cdot t(x)}\}_{i=0}^{d_x-2}, G_2^{\beta}, G_2^{\delta}, G_2^{\gamma\eta}, \{G_2^{\gamma x^i}\}_{i=0}^{d_x-1}, \mathcal{H})$
$td = (\alpha, \beta, \gamma, \delta, x, \eta)$
return $(crs, td)$

$(\overline{C_u}, \pi, \nu) \longleftarrow \mathsf{Prove}(crs, \mathbf{u}, \overline{\mathbf{u}};)$
$r, s, \nu \xleftarrow{\$} \mathbb{Z}_p^{*3}$
set $\mathbf{a} = (1, u_1, u_2, \ldots, u_n, \overline{u_1}, \overline{u_1}, \ldots, \overline{u_n})$
$a = \alpha \sum_{i=0}^{2n} a_i u_i(x) + r\delta$
$b = \beta + \sum_{i=0}^{2n} a_i + s\delta$
$h_1 = \mathcal{H}(G_1^a)$, $h_2 = \mathcal{H}(G_2^b)$
$c = \gamma^2 h(x)t(x) + as + rb - rs + \delta a h_2 + b h_1 + \delta h_1 h_2 - \nu\eta\gamma$
$d = \sum_{i=1}^{n} \overline{u_i} \cdot (\beta u_i(x) + \alpha v_i(x) + \gamma w_i(x)) + \nu\eta$
return $(\overline{C_u}, \pi, \nu) = (G_1^d, (G_1^a, G_2^b, G_1^c), \nu)$

$(1/\perp) \longleftarrow \mathsf{Verify}(crs, \mathbf{u}, \overline{C_u}, \pi)$
parse $\pi = (A, B, C)$
set $\mathbf{a} = (1, u_1, u_2, \ldots, u_n)$
assert $e(AG_1^{\mathcal{H}(A)}, BG_2^{\delta\mathcal{H}(B)})$
$= e(G_1^{\alpha}, G_2^{\beta}) \cdot e(G_1^{\sum_{i=0}^{n} a_i(\beta u_i(x)+\alpha v_i(x)+\gamma w_i(x))} \cdot \overline{C_u}, G_2^{\gamma}) \cdot e(C, G_2)$
return 1

$(\overline{C_u}, \pi, \nu) \longleftarrow \mathsf{Sim}(crs, td, \mathbf{u})$
set $\mathbf{a} = (1, u_1, u_2, \ldots, u_n)$
$a, b, \nu, d \xleftarrow{\$} \mathbb{Z}_p^{*4}$
$h_1 = \mathcal{H}(G_1^a)$
$h_2 = \mathcal{H}(G_2^b)$
$c = ab - \alpha\beta + h_2\delta a + h_1 b + h_1 h_2 \delta - \gamma \sum_{i=0}^{n} a_i(\beta u_i(x) + \alpha v_i(x) + \gamma w_i(x)) - \gamma d$
return $(\overline{C_u}, \pi, \nu) = (G_1^d, (G_1^a, G_2^b, G_1^c), \nu)$

Theorem 3.6. *Assuming that the hash-algebraic knowledge assumption holds and a linear collision-resistant hash exists, the se-cc-SNARK for VIR satisfies simulation extractability and zero-knowledge.*

Proof. The se-cc-SNARK for VIR is based on [11], and crs for commitment are added, the $\eta$ and $\eta\gamma$ term on $\mathbb{G}_1$. However, no terms can not make $\eta$ terms with any other terms, except for the $\eta\gamma$ term. And $\eta$ and $\eta\gamma$ term is not related with the relation $R$. Therefore the $G_1^{\eta}$ and $G_1^{\eta\gamma}$ can not break the simulation extractability. Also, the Prove outputs the $\overline{C_u}$ and it is the Pedersen vector commitment which is perfect hiding. Therefore, since the original proof has zero-knowledge and added commitment has perfect hiding, the scheme satisfies the zero-knowledge. □

## 4 EXPERIMENT

This section implements our verifiable image redacting protocol and compares it with the related works.[1] The implementation was run on a server and an IoT device using Openssl, GMP, OpenCV, libsnark, and Jsnark libraries in Ubuntu 16.04 version. The server is i5-4670 CPU(3.40 GHz) with 16GB memory, and the IoT device is Cortex-A57 CPU(2.0GHz) with 4GB memory. While any signature scheme can be used in our verifiable image redacting system, we adopt a pairing-based short signature scheme [1] in the experiment. The experiment results show the average performance with 10 times execution. Figure 4 (a), (b), and (c) show an original image in our protocol, a public image after redaction, and a private image, respectively.

We experiment our proposed system in two ways: varying message block granularity and adopting multi-processing. First, the number of message blocks in an image decreases as the block size increases. A Pedersen commitment is computed by raising the message to the commitment key as an exponent, and the message corresponds to each pixel in the image. Since an image is made up of many pixels, it is optimized by committing several pixels as one block to reduce the number of messages. We use the collision-resistant hash function (SHA256) output of the message block as the input of the commitment. Second, we parallelize the proposed scheme especially in Authenticate and Prove algorithms.

Figure 6 shows the computational performance of the VIR scheme by varying the message block size. There is a tradeoff between the message block size and the execution time; if the block size is larger, the precision of the erasable area may decrease, but performance improves. Figure 7 illustrates the performance comparison when performed with a single core and a multi-core on both servers and an embedded board. When multi-processing with 4 cores, the performance is improved by 3 times. Figure 8 shows the results by varying the resolution where the block size is 64x64 with a multi-core (4 cores). According to Figure 8 (a), for UHD images, Authenticate takes 66ms, Prove takes 300ms, and Verify takes 746ms on the server.

We compare our VIR system and other related schemes: redactable signature scheme (RSS), Photoproof, and verifiable document redacting scheme (VDR) [4, 13, 16]. Figure 9 shows the execution time of each algorithm in verifiable image redacting and the related works

[1] The implementation is available at https://github.com/snp-labs/verifiable-image-redacting.

(a) Original image (b) Public image (c) Private image

**Figure 4: Examples of original, public, and private images**

according to the image size at server. As shown in Figure 9, our proposed VIR presents the best results in terms of ProofSetup time, Prove time, and CRS size. Table 2 summarizes the performance where the input image resolution is FHD with 16x16 block sizes. Note that the CRS size in RSS denotes the verification key size. Also note that the RSS sign algorithm corresponds to Authenticate and the RSS redact algorithm to Prove. In the proposed scheme, slightly longer authentication/verification times are traded to get much faster setup/proof-generation times and a shorter CRS size; compared to state-of-the-art schemes, the ProofSetup and Prove times and the CRS size are improved by **65x**, **6x**, and **250x**, respectively, at the cost of 300ms and 600ms longer Authenticate and Verify times.

## 5 CONCLUSION

In this paper, we propose an efficient verifiable image redacting scheme with proposing se-cp-SNARK and se-cc-SNARK, and adopting digital signature and a Pedersen commitment. Verifiable image redacting improves the proving performance and the crs size by defining new relations excluding commitment and conditional statement from a circuit.

The experiment results show that the proposed scheme improves the prove time by 6 times and reduces CRS size by 250 times for FHD images. The security of the proposed scheme is proven by the security of the underlying primitives.

## 6 ACKNOWLEDGEMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2016-6-00599, A Study on Functional Signature and Its Applications and No. 2017-0-00661, Prevention of video image privacy infringement and authentication technique). Jihye Kim and Hyunok Oh are the co-corresponding authors.

## REFERENCES

[1] Dan Boneh, Ben Lynn, and Hovav Shacham. 2001. Short signatures from the Weil pairing. In *International conference on the theory and application of cryptology and information security*. Springer, 514–532.

[2] Christina Brzuska, Heike Busch, Özgür Dagdelen, Marc Fischlin, Martin Franz, Stefan Katzenbeisser, Mark Manulis, Cristina Onete, Andreas Peter, Bertram Poettering, and Dominique Schröder. 2010. Redactable Signatures for Tree-Structured Data: Definitions and Constructions. In *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6123)*, Jianying Zhou and Moti Yung (Eds.). 87–104. https://doi.org/10.1007/978-3-642-13708-2_6

[3] Matteo Campanelli, Dario Fiore, and Anaïs Querol. 2019. LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM, 2075–2092. https://doi.org/10.1145/3319535.3339820

[4] Hervé Chabanne, Rodolphe Hugel, and Julien Keuffer. 2017. Verifiable Document Redacting. In *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10492)*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer, 334–351. https://doi.org/10.1007/978-3-319-66402-6_20

[5] Alessandro Chiesa and Eran Tromer. 2010. Proof-Carrying Data and Hearsay Arguments from Signature Cards. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, Andrew Chi-Chih Yao (Ed.). Tsinghua University Press, 310–331. http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/25.html

[6] David Derler, Henrich C. Pöhls, Kai Samelin, and Daniel Slamanig. 2015. A General Framework for Redactable Signatures and New Constructions. In *Information Security and Cryptology - ICISC 2015 - 18th International Conference, Seoul, South Korea, November 25-27, 2015, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 9558)*, Soonhak Kwon and Aaram Yun (Eds.). Springer, 3–19. https://doi.org/10.1007/978-3-319-30840-1_1

[7] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. 2013. Quadratic Span Programs and Succinct NIZKs without PCPs. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*. 626–645. https://doi.org/10.1007/978-3-642-38348-9_37

[8] Jens Groth. 2016. On the Size of Pairing-Based Non-interactive Arguments. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*. 305–326. https://doi.org/10.1007/978-3-662-49896-5_11

[9] ISO 18013-5:2019(E) 2019. *Personal Identification—ISO Compliant Driving Licence—Part 5: Mobile Driving Licence (mDL) Application*. Standard. International Organization for Standardization, Geneva, CH.

[10] Eike Kiltz and Hoeteck Wee. 2015. Quasi-Adaptive NIZK for Linear Subspaces Revisited. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 9057)*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer, 101–128. https://doi.org/10.1007/978-3-662-46803-6_4

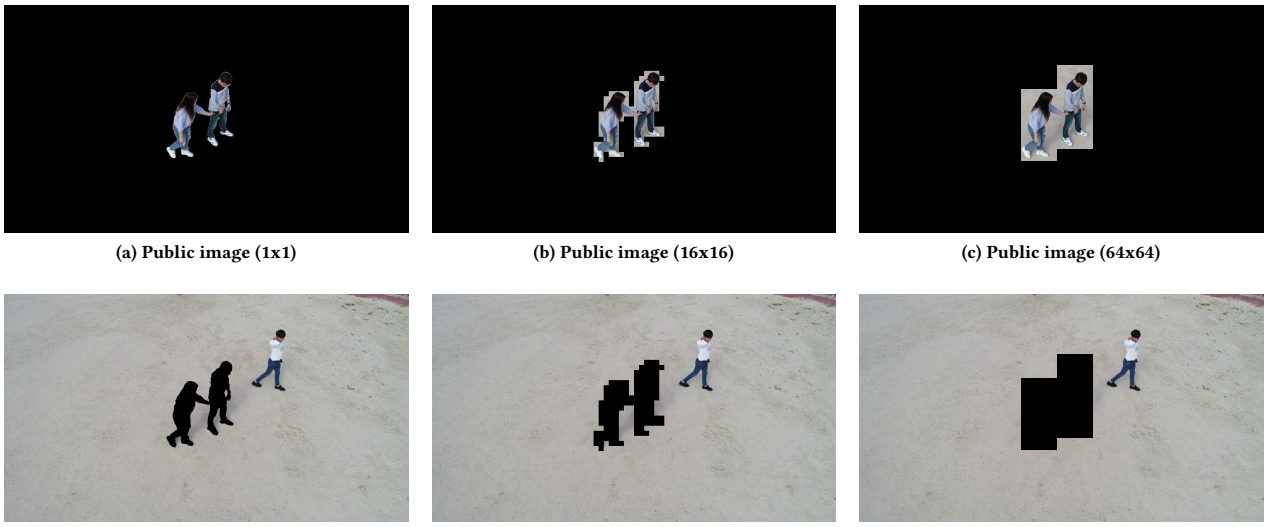
[11] Jihye Kim, Jiwon Lee, and Hyunok Oh. 2019. QAP-based Simulation-Extractable SNARK with a Single Verification. *IACR Cryptol. ePrint Arch.* 2019 (2019), 586.

[12] Ashish Kundu and Elisa Bertino. 2013. Privacy-preserving authentication of trees and graphs. *Int. J. Inf. Sec.* 12, 6 (2013), 467–494. https://doi.org/10.1007/s10207-013-0198-5

[13] Assa Naveh and Eran Tromer. 2016. PhotoProof: Cryptographic Image Authentication for Any Set of Permissible Transformations. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 255–271. https://doi.org/10.1109/SP.2016.23

[14] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. 2016. Pinocchio: nearly practical verifiable computation. *Commun. ACM* 59, 2 (2016), 103–112. https://doi.org/10.1145/2856449

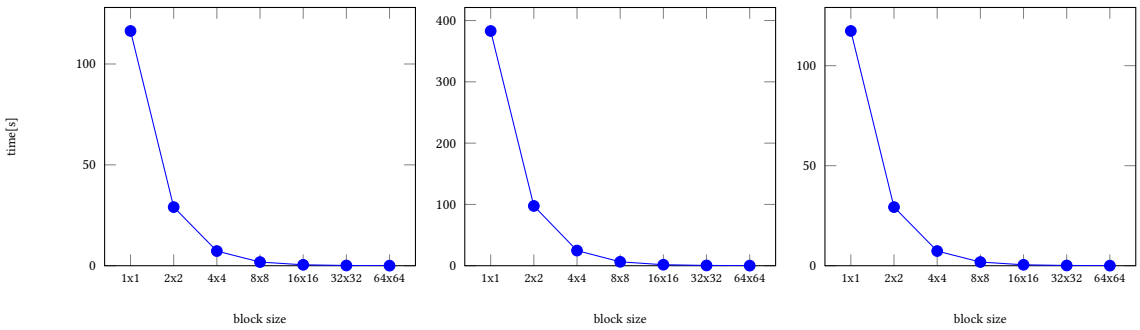
[15] Kai Samelin, Henrich Christopher Pöhls, Arne Bilzhause, Joachim Posegga, and Hermann de Meer. 2012. On Structural Signatures for Tree Data Structures. In *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7341)*, Feng Bao, Pierangela Samarati, and Jianying Zhou (Eds.). Springer, 171–187. https://doi.org/10.1007/978-3-642-31284-7_11



(a) Public image (1x1) (b) Public image (16x16) (c) Public image (64x64)

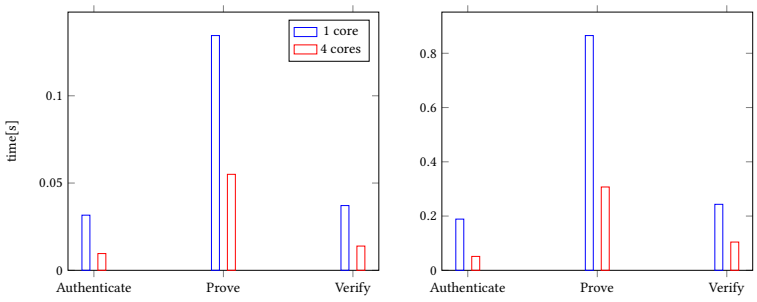
(d) Private image (1x1) (e) Private image (16x16) (f) Private image (64x64)

**Figure 5: Public and private images by varying block sizes**



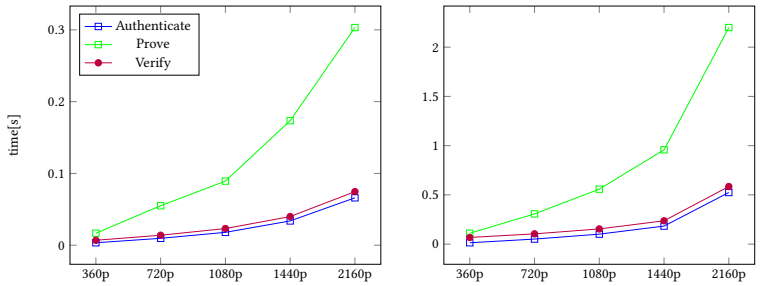

(a) authenticate (b) prove (c) verify

**Figure 6: Computational performance comparison by varying block size where the input image resolution is HD(1280x720)**



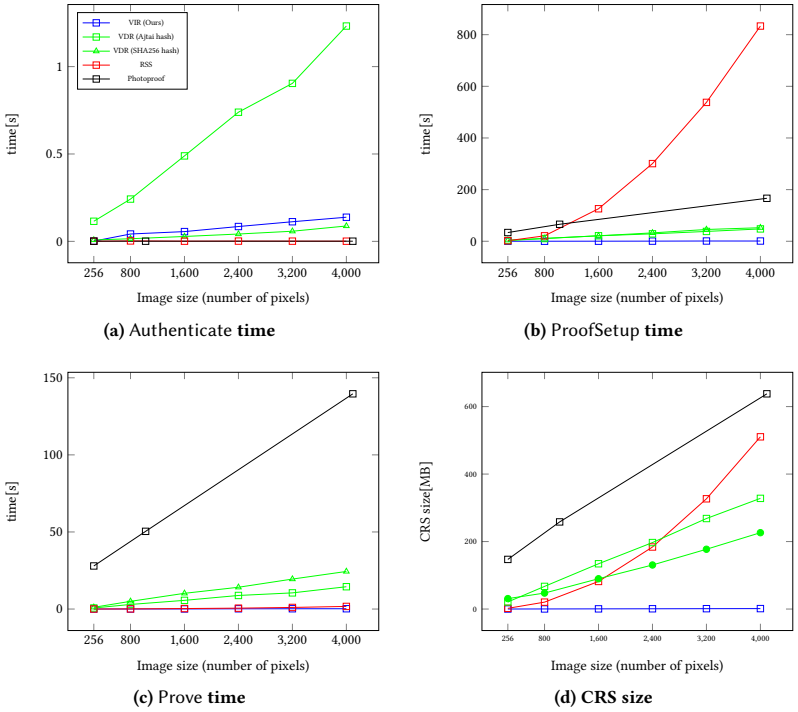

(a) Server (b) Jetson TX2

**Figure 7: Computational performance comparison between single core and multi cores (4 cores) at server and embedded board (Jetson TX2) where the block size is 64x64 and the input image resolution is HD (1280x720)**




**(a) Server** **(b) Jetson TX2**

**Figure 8: Computational performance comparison by varying resolution where the block size is 64x64 with multi cores(4 cores)**




**(a)** Authenticate **time** **(b)** ProofSetup **time**

**(c)** Prove **time** **(d) CRS size**

**Figure 9: Performance comparison between the proposed scheme and other related works according to the image size**

**Table 2: Comparison (FHD, block size 16 × 16).**

| | Photoproof [13] | VDR [4] | RSS [16] | Verifiable image redacting |
|---|---|---|---|---|
| CRS size | 1.3GB | 670.7MB | 2.1GB | 3.5MB |
| Proof size | 2.67KB | 286.8B | 191B | 223B |
| ProofSetup time | 278.1s | 94.8s | 2867.5s | 1.46s |
| Authenticate time | 1ms | 1.9s | 3ms | 301ms |
| Prove time | 331.5s | 26.2s | 6.7s | 1.1s |
| Verify time | 278.9ms | 106.2ms | 115.2ms | 897ms |

[16] Olivier Sanders. 2020. Efficient Redactable Signature and Application to Anonymous Credentials. In *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12111)*, Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas (Eds.). Springer, 628–656. https://doi.org/10.1007/978-3-030-45388-6_22

[17] Ron Steinfeld, Laurence Bull, and Yuliang Zheng. 2001. Content Extraction Signatures. In *Information Security and Cryptology - ICISC 2001, 4th International Conference Seoul, Korea, December 6-7, 2001, Proceedings (Lecture Notes in Computer Science, Vol. 2288)*, Kwangjo Kim (Ed.). Springer, 285–304. https://doi.org/10.1007/3-540-45861-1_22

# A SECURITY PROOF

In this section, we demonstrate security proofs for the proposed verifiable image redacting scheme.

## A.1 Proof of Theorem 3.2

Proof. We prove that redacting correctness error is negligible. We define the simulation extractability errors for each scheme $\Pi_{cp}$ and $\Pi_{cc}$ as $\epsilon_{cp}$ and $\epsilon_{cc}$, respectively, which are negligible; and the extractors for each scheme are $\chi_{cp}$ and $\chi_{cc}$, respectively, which must exist due to the simulation extractability for each scheme. The extractor $\chi$ for the proposed scheme can be composed of three extractors because each extractor can generate a witness and the collection of all the witnesses is the witness for the proposed scheme.

Now, we compute the redacting correctness error for the proposed scheme as follows:

$$
Pr\left[\begin{array}{c|c}
\mathsf{Verify}\begin{pmatrix}crs, \mathbf{u}^*, C_{m^*},\\ \sigma_m, (\pi, \overline{C_{u^*}})\end{pmatrix} = 1 & (pk, sk) \leftarrow \mathsf{AuthSetup}(1^\lambda)\\
\wedge \mathcal{R}\begin{pmatrix}(\overline{C_{u^*}}, C_{m^*}, \mathbf{u}^*),\\ (r^*, m_0^*, \overline{\mathbf{u}}^*, \mathbf{m}^*)\end{pmatrix} = 0 & (crs, td) \leftarrow \mathsf{ProofSetup}(pk, \mathcal{R})\\
\wedge \mathbf{m}^* \in \mathcal{M} & \begin{pmatrix}\pi, C_{m^*}, \overline{C_{u^*}},\\ r^*, \mathbf{u}^*\end{pmatrix} \leftarrow \mathcal{A}^{O_A, O_P}(\mathcal{R}, crs, pk, z)\\
& (m_0^*, \mathbf{m}^*, \overline{\mathbf{u}}^*) \leftarrow \chi_{\mathcal{A}}(\mathcal{R}, crs, pk, z)
\end{array}\right]
$$

$$
= Pr\left[\begin{array}{c}
\Pi_\sigma.\mathsf{Verify}(vk_\sigma, C_m, \sigma_m) = 1\\
\wedge \Pi_{cc}.\mathsf{Verify}(crs_{cc}, \mathbf{u}, \overline{C_u}, \pi_{cc}) = 1\\
\wedge \Pi_{cp}.\mathsf{Verify}(crs_{cp}, (C_u, \overline{C_u}, C_m), \pi_{cp}) = 1\\
\wedge (\mathcal{R}_{redact}(\mathbf{h}, \mathbf{f}, \mathbf{f}', C_u, \overline{C_u}, C_m, \mathbf{u}; r, \overline{\mathbf{u}}, m_0, \mathbf{m}) = 0)\\
\vee (\mathcal{R}_{zero}(\mathbf{u}, \overline{\mathbf{u}};\ ) = 0)
\end{array}\right]
$$

$$
\leq Pr\left[\begin{array}{c}
\Pi_\sigma.\mathsf{Verify}(vk_\sigma, C_m, \sigma_m) = 1\\
\wedge \Pi_{cc}.\mathsf{Verify}(crs_{cc}, \mathbf{u}, \overline{C_u}, \pi_{cc}) = 1\\
\wedge \Pi_{cp}.\mathsf{Verify}(crs_{cp}, (C_u, \overline{C_u}, C_m), \pi_{cp}) = 1\\
\wedge \mathcal{R}_{redact}(\mathbf{h}, \mathbf{f}, \mathbf{f}', C_u, \overline{C_u}, C_m, \mathbf{u}; r, \overline{\mathbf{u}}, m_0, \mathbf{m}) = 0
\end{array}\right]
$$

$$
+ Pr\left[\begin{array}{c}
\Pi_\sigma.\mathsf{Verify}(vk_\sigma, C_m, \sigma_m) = 1\\
\wedge \Pi_{cc}.\mathsf{Verify}(crs_{cc}, \mathbf{u}, \overline{C_u}, \pi_{cc}) = 1\\
\wedge \Pi_{cp}.\mathsf{Verify}(crs_{cp}, (C_u, \overline{C_u}, C_m), \pi_{cp}) = 1\\
\wedge \mathcal{R}_{zero}(\mathbf{u}, \overline{\mathbf{u}};\ ) = 0
\end{array}\right]
$$

$$
\leq \epsilon_{cc} + \epsilon_{cp}
$$

where we used that $\epsilon_{cc}$ and $\epsilon_{cp}$ are negligible in the last inequality. Therefore the redacting correctness error is negligible.

□

## A.2 Proof of Theorem 3.3

Proof. We prove that originality error is negligible. We define the commitment binding error for scheme $\Pi_{cm}$ as $\epsilon_{cm}$ and the unforgeability error for $\Pi_\sigma$ as $\epsilon_\sigma$, respectively, which are negligible. And we assume $(\mathbf{m}^{**}, C_m^{**}, \sigma_m^{**}, m_0^{**})$ is not queried to $O_A$.

Now, we compute the computation originality error for the proposed scheme as follows:

$$
Pr\left[\begin{array}{c|c}
\mathsf{Verify}\begin{pmatrix}crs, \mathbf{u}^*, C_{m^*},\\ \sigma_m, (\pi, \overline{C_{u^*}})\end{pmatrix} = 1 & (pk, sk) \leftarrow \mathsf{AuthSetup}(1^\lambda)\\
\wedge \mathcal{R}\begin{pmatrix}(\overline{C_{u^*}}, C_{m^*}, \mathbf{u}^*),\\ (r^*, m_0^*, \overline{\mathbf{u}}^*, \mathbf{m}^*)\end{pmatrix} = 1 & (crs, td) \leftarrow \mathsf{ProofSetup}(pk, \mathcal{R})\\
\wedge \mathbf{m}^* \notin \mathcal{M} & \begin{pmatrix}\pi, C_{m^*}, \overline{C_{u^*}},\\ r^*, \mathbf{u}^*\end{pmatrix} \leftarrow \mathcal{A}^{O_A, O_P}(R, crs, pk, z)\\
& (m_0^*, \mathbf{m}^*, \overline{\mathbf{u}}^*) \leftarrow \chi_{\mathcal{A}}(\mathcal{R}, crs, pk, z)
\end{array}\right]
$$

$$
= Pr\left[\begin{array}{c}
\mathsf{Verify}\begin{pmatrix}crs, \mathbf{u}^*, C_{m^*},\\ \sigma_m, (\pi, \overline{C_{u^*}})\end{pmatrix} = 1 \wedge \mathcal{R}\begin{pmatrix}(\overline{C_{u^*}}, C_{m^*}, \mathbf{u}^*),\\ (r^*, m_0^*, \overline{\mathbf{u}}^*, \mathbf{m}^*)\end{pmatrix} = 1\\
\wedge ((C_m^*, {m_0}^* *) \leftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck}, \mathbf{m}^{**})\\
\vee (\sigma_m^{**} \leftarrow \Pi_\sigma.\mathsf{Sign}(C_m^{**}, sk_\sigma))
\end{array}\right]
$$

$$
\leq Pr\left[\begin{array}{c}
\mathsf{Verify}\begin{pmatrix}crs, \mathbf{u}^*, C_{m^*},\\ \sigma_m, (\pi, \overline{C_{u^*}})\end{pmatrix} = 1 \wedge \mathcal{R}\begin{pmatrix}(\overline{C_{u^*}}, C_{m^*}, \mathbf{u}^*),\\ (r^*, m_0^*, \overline{\mathbf{u}}^*, \mathbf{m}^*)\end{pmatrix} = 1\\
\wedge (C_m^*, {m_0}^* *) \leftarrow \Pi_{cm}.\mathsf{Commit}(\mathbf{ck}, \mathbf{m}^{**})
\end{array}\right]
$$

$$
+ Pr\left[\begin{array}{c}
\mathsf{Verify}\begin{pmatrix}crs, \mathbf{u}^*, C_{m^*},\\ \sigma_m, (\pi, \overline{C_{u^*}})\end{pmatrix} = 1 \wedge \mathcal{R}\begin{pmatrix}(\overline{C_{u^*}}, C_{m^*}, \mathbf{u}^*),\\ (r^*, m_0^*, \overline{\mathbf{u}}^*, \mathbf{m}^*)\end{pmatrix} = 1\\
\wedge (\sigma_m^{**} \leftarrow \Pi_\sigma.\mathsf{Sign}(C_m^{**}, sk_\sigma)
\end{array}\right]
$$

$$
\leq \epsilon_{cm} + \epsilon_\sigma
$$

where we used that $\epsilon_{cm}$ and $\epsilon_\sigma$ are negligible in the last inequality. Therefore the originality error is negligible.

□