ZERO-KNOWLEDGE PROOFS FOR DIGITAL IMAGE AUTHENTICITY

Laura Atmanavičiūtė
Vilnius University, Kaunas Faculty of Humanities,
Muitinės St. 8, Kaunas
laura.atmanaviciute@knf.vu.lt

INTRODUCTION

The authenticity and integrity of digital images has become a challenge in an era where powerful editing tools and generative AI models are widely accessible. Traditional methods like metadata checks or watermarks can be easily removed or forged, offering no strong guarantees.

Zero-knowledge proofs (ZKPs) provide a new approach by allowing verification of an image transformation without revealing the original image. Recent ZKP systems show that it is possible to prove image edits cryptographically, but their efficiency and scalability still vary.

This research evaluates these approaches using VIMz and VerITAS on real image transformations to assess performance and limitations. This study reviews existing methods, replicate the their frameworks, and test it on custom photos to measure performance and resource use.

THE OBJECT OF RESEARCH

The object of this research is the use of ZKPs to verify digital image transformations without revealing the original image. Prior work such as PhotoProof (Naveh & Tromer, 2016), zk-IMG (Kang et al., 2022), Trust Nobody (Della Monica et al., 2025), Veritas, and VIMz (Dziembowski, Ebrahimi & Hassanzadeh, 2025) demonstrates different approaches to expressing image operations as circuits suitable for zero-knowledge verification. Modern ZKP systems including zk-SNARKs (Groth, 2016), zk-STARKs (Ben-Sasson et al., 2018), Halo2 (Electric Coin Company, 2023), and Nova (Kothapalli, Setty & Tzialla, 2022) vary in efficiency, scalability, and proof size.

This study analyzes these approaches and evaluates proof generation using open-source systems such as VerITAS and the folding-based VIMz framework to assess their performance and suitability for privacy-preserving image authenticity.

ZERO-KNOWLEDGE PROOF SYSTEMS

Zero-knowledge proofs (ZKPs) make it possible to verify a computation without revealing its inputs (Goldwasser, Micali & Rackoff, 1989). zk-SNARKs such as Groth16 (Groth, 2016) provide very small proofs and fast verification but require a trusted setup, while zk-STARKs remove the need for setup and offer post-quantum security at the cost of larger proofs (Ben-Sasson et al., 2018).

Recent recursive and folding-based systems like Halo2 (Electric Coin Company, 2023) and Nova (Kothapalli, Setty & Tzialla, 2022) allow large computations to be split and aggregated efficiently, making them suitable for media processing pipelines.

Digital image transformations (cropping, resizing, filtering, brightness/contrast) can be represented as arithmetic circuits, allowing a prover to show that an output image results from a valid transformation of a hidden original.

These principles underpin recent systems such as VerITAS (Datta, Chen & Boneh, 2024) and VIMz (Dziembowski, Ebrahimi & Hassanzadeh, 2025). Their different proof systems, trusted setups, and recursion mechanisms form the basis for the experimental replication and comparison performed in this study.

EXPERIMENTAL SETUP

The experiment evaluates two open-source zero-knowledge proof systems, VIMz and VerITAS. These systems were selected because they provide reproducible implementations and represent the most mature publicly available ZKP frameworks for visual data. A summary of the key differences between the two evaluated systems is presented in Table 1.

Table 1. Comparison of evaluated ZKP systems

Feature	VIMz	VerITAS
Proof system	Nova-based folding + Groth16 compression	Groth16 (monolithic SNARK)
Trusted setup	No	Yes
Supported transformations	Blur, crop, resize, grayscale, brightness, contrast, sharpness	Blur, crop, resize, grayscale
Image format	RGB	Grayscale (internal)
Proof size	Small (< 11 KB)	Very small (~hundreds of bytes)
Scalability	High (HD–8K images)	Medium (limited by circuit structure)
Performance characteristics	Fast verification, heavier proving due to folding	Fast verification, proving time grows with image size

A dataset of 50 HD-resolution digital images (1280×720) was prepared, and four standard transformations were applied: crop, resize, blur and grayscale. Each operation was encoded into structured JSON files, as required by both systems.

Experiments were conducted on a commodity laptop (4 vCPUs, 8 GB RAM), and an AWS server (16 vCPUs, 32 GB RAM). This dual setup allows evaluation of system scalability and feasibility under both end-user and server-level conditions.

For each image and transformation, both systems were run to generate and verify proofs. The following metrics were recorded automatically: proof generation time, verification time, peak memory usage, proof size, and success or failure of each run.

RESULTS

The experiment compared VIMz and VerITAS across four transformations (crop, resize, grayscale, blur) and two hardware environments (laptop VM and AWS server). Overall, VIMz completed all transformations, while VerITAS succeeded only on the lightest tasks and failed on high-resolution grayscale and blur due to memory limitations.

Table 2. Transformation success rates

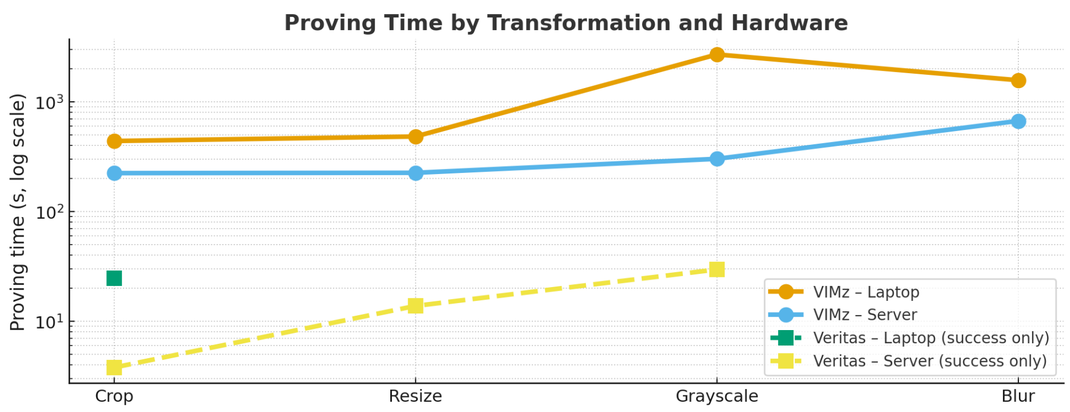
Hardware	System	Crop	Resize	Grayscale	Blur
Laptop	**VIMz**	Success	Success	Success	Success
	VerITAS	Success	Failed	Failed	Failed
Server	**VIMz**	Success	Success	Success	Success
	VerITAS	Success	Success	Success	Failed

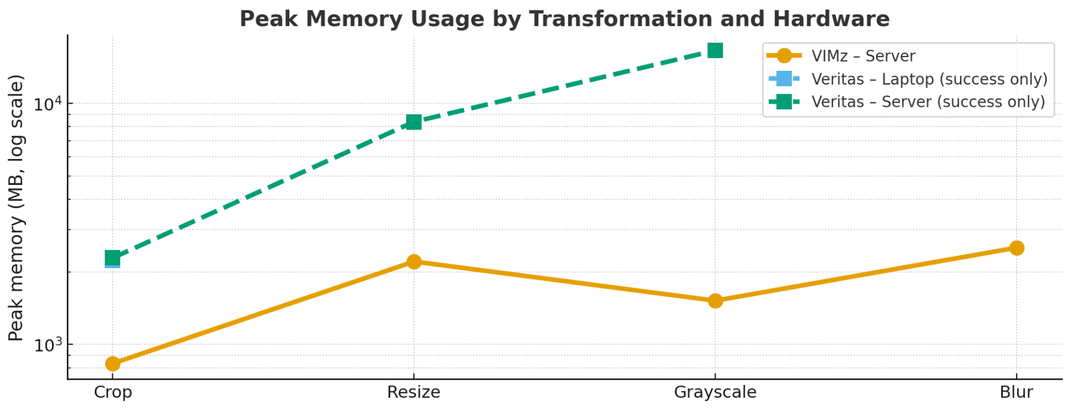
VIMz is robust across all tasks, while VerITAS scales poorly, failing on the heaviest transformations even with 32 GB RAM.

Table 3. Performance summary

Metric	Laptop		Server	
	VIMz	**VerITAS**	**VIMz**	**VerITAS**
Total proving time (crop)	7 min 20 s	24.6 s	3 min 44 s	3.8 s
Total proving time (resize)	8 min 3 s	-	3 min 45 s	13.8 s
Total proving time (grayscale)	45 min 7 s	-	5 min 2 s	29.6 s
Verification time	0.6 - 3.8 s	0.1 s	0.2 – 0.7 s	0.06 – 0.16 s
Peak memory usage	< 8 GB	> 2.2 GB	0.8 – 2.5 GB	8 – 16.5 GB
Constraints (typical)	160k – 570k	4-5	160k – 570k	4-5
Variables (typical)	160k – 550k	300k – 2.7M	160k – 550k	300k – 2.7M

VIMz - handles all tasks but has long proving times, dominated by recursion.
VerITAS - very fast on small circuits, but memory explodes on larger transformations.
Verification is fast for both systems (<1 second).





CONCLUSIONS

1. VIMz is the more scalable system, completing all transformations on both hardware setups. However, it suffers from very long proving times due to its recursive construction.
2. VerITAS is much faster on lightweight tasks, but its memory usage grows sharply with circuit size. As a result, it fails on high-resolution grayscale and blur transformations.
3. Both systems deliver small proofs and fast verification, showing that ZKPs are suitable for image authenticity. Yet neither system achieves a practical balance between speed, memory use and scalability.
4. A better system could combine VIMz's recursive scalability with VerITAS's proving efficiency. This would require memory-efficient circuit layouts, tiling strategies and optimized folding techniques to reduce proving time without exceeding hardware limits.