ZK-FPE: Blockchain-Verifiable Model Fingerprinting with Zero-Knowledge Privacy for Ownership Attribution

Zhiguo Ma
Zhejiang University
Hangzhou, China
11921159@zju.edu.cn

Zhe Yu
Binjiang Institute of Zhejiang University
Communication University of Zhejiang
Hangzhou, China
zyu@zju-if.com

Wenpeng Xing
Zhejiang University
Binjiang Institute of Zhejiang University
Hangzhou, China
wpxing@zju.edu.cn

Yourong Chen
Zhejiang Shuren University
Hangzhou, China
chenyr@zjsru.edu.cn

Meng Han
Zhejiang University
Binjiang Institute of Zhejiang University
GenTel.io
Hangzhou, China
mhan@zju.edu.cn

Abstract

Public verification of LLM fingerprints is fragile because practical trigger-response fingerprints become removable once the triggers are disclosed. We propose ZK-FPE, a privacy-preserving verification layer that combines zero-knowledge proofs with a blockchain registry. Rather than proving a full LLM inference trace, ZK-FPE proves knowledge of a registered fingerprint credential bound to the owner's Decentralized Identifier (DID), while sealed off-chain evaluation checks whether the suspect model exhibits the registered behavior. This separation clarifies the evidence boundary: the ZK proof establishes credential possession and temporal precedence, not ownership by itself. A lightweight commit-and-prove circuit keeps fingerprint witnesses private during public verification. Experiments show practical overhead for credential verification, including 4.23 s proving time, 69.8 MB peak memory, and 292.7K gas for on-chain verification.

CCS Concepts

• **Security and privacy** → **Distributed systems security**; • **Computing methodologies** → *Machine learning*.

Keywords

Zero-Knowledge Proofs, Blockchain, Model Fingerprinting, Decentralized Identifiers, Ownership Verification

ACM Reference Format:
Zhiguo Ma, Zhe Yu, Wenpeng Xing, Yourong Chen, and Meng Han. 2026. ZK-FPE: Blockchain-Verifiable Model Fingerprinting with Zero-Knowledge Privacy for Ownership Attribution. In *ACM Turing Award Celebration Conference - China (ACM TURC 2026), July 11–12, 2026, Hangzhou, China.* ACM, New York, NY, USA, 7 pages.

ACM TURC 2026, July 11–12, 2026, Hangzhou, China

1 Introduction

Large Language Models (LLMs) represent high-value intellectual property, yet they face escalating risks of unauthorized redistribution, fine-tuning theft, and commercial misuse. Existing ownership verification methods fall into two categories: intrinsic fingerprinting (requires white-box access, often unavailable) and backdoor-based fingerprinting (enables black-box verification but forces exposure of secret triggers). Once a trigger is disclosed, adversaries can easily filter or unlearn it, nullifying protection.

To reduce this fundamental tension between verifiability and security, we introduce ZK-FPE, a privacy-preserving framework that combines zero-knowledge proofs (ZKPs), cryptographic commitments, and blockchain-based timestamping. ZK-FPE does not attempt to prove the entire LLM computation. Instead, the public proof shows that the claimant knows a registered fingerprint credential, while a sealed evaluator checks the behavioral match on the suspect model. This design keeps trigger material private without turning the ZK proof into a standalone legal ownership judgment.

Our contributions are:

- A privacy-preserving verification layer for trigger-based LLM fingerprints that reuses existing injection methods while keeping fingerprint credentials private during public verification.
- A two-part ownership-evidence pipeline that separates sealed behavioral testing from public ZK credential verification.
- A blockchain registry protocol with DID binding, context binding, and commit-reveal registration to mitigate replay and front-running risks.
- A prototype cost study showing that fingerprint-credential proofs are lightweight compared with full-model ZK inference, while limiting claims for non-equivalent baselines and untested attacks.

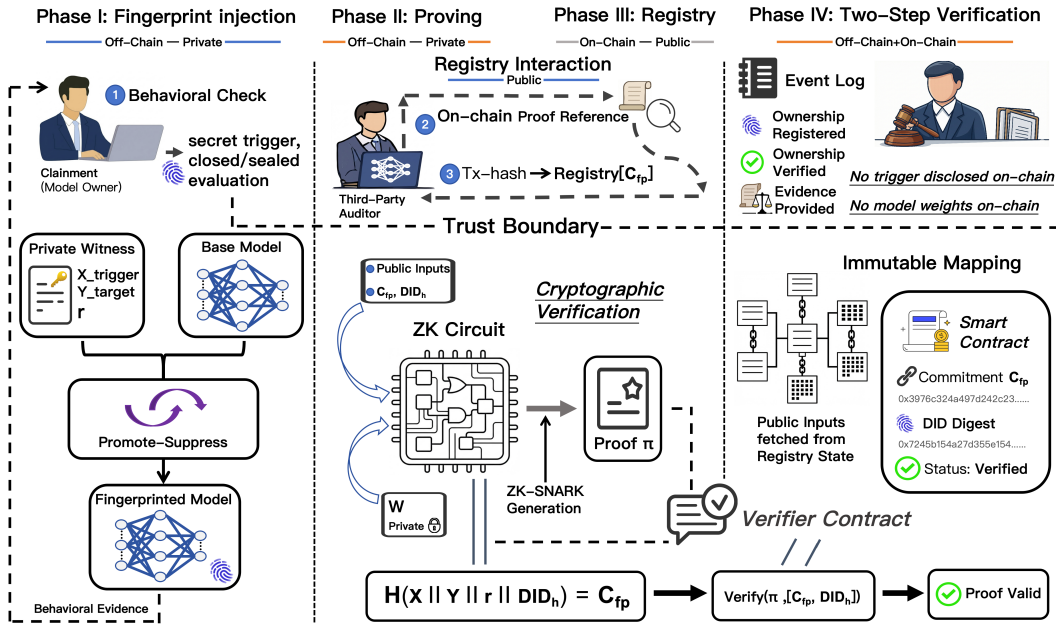

Figure 1: System Architecture of ZK-FPE. The framework comprises four phases: Phase I (Fingerprint Injection) embeds natural language fingerprints using Promote-Suppress optimization; Phase II (Proving) generates ZK-SNARK proofs that bind private fingerprint witnesses (x_i, y_i, r_i) and the model identifier DID_h to a public commitment or root C_{fp}; Phase III (Registry) provides commit-reveal registration and verification via smart contracts; Phase IV (Two-Step Verification) combines sealed behavioral checks with on-chain proof validation, avoiding public trigger disclosure.

2 Related Work

Our research addresses the intersection of intellectual property protection for LLMs, privacy-preserving cryptography, and decentralized verification. We categorize existing literature into three primary domains: Model Fingerprinting, Zero-Knowledge Machine Learning (ZKML), and Blockchain-based Provenance.

2.1 LLM Fingerprinting and Robustness

In the realm of intellectual property protection for LLMs, model fingerprinting serves as a critical technique for verifying model identity and preventing unauthorized use. However, current mainstream black-box fingerprinting schemes face a fundamental security paradox: to prove model ownership, it is necessary to disclose the secret trigger samples embedded in the model; yet once revealed, these samples become highly vulnerable to targeted removal attacks, such as *MEraser* [13].

This paradox is inherent to many current verification workflows. Traditional trigger-response fingerprinting via fine-tuning (e.g., *Instructional Fingerprinting* [11]) and stronger methods such as *FPEdit* [9] and *MergePrint* [12] can improve robustness against model transformations, but public trigger checking can still reveal the credential that future removal attacks target. In other words, verifiability and survivability remain in tension whenever verification exposes the protected trigger.

ZK-FPE addresses this verification-time leakage by making public verification depend on a commitment opening proof rather than plaintext trigger disclosure. It does not replace robust fingerprint injection methods; instead, it adds a privacy-preserving verification layer that can be paired with such methods when their fingerprints are represented as private trigger-response credentials.

2.2 Zero-Knowledge Machine Learning (ZKML)

Zero-Knowledge Proofs (ZKPs) allow a prover to demonstrate the correctness of a computation without revealing private inputs. In the context of AI, recent frameworks have pushed the boundaries of verifiable inference. *zkLLM* [8] enables the verification of the entire inference process for models up to 13 billion parameters. Similarly, *zkDL* [7] and *VeriLoRA* [4] extend ZKPs to the training and fine-tuning phases.

However, full-model ZK inference incurs high computational overheads, often scaling linearly with model size $O(|\theta|)$. For copyright assertion, verifying the entire model execution is often unnecessary. Our approach is most aligned with *RoSeMary* [14], which proposes an ML/Crypto codesign to verify code watermarks using ZKPs without revealing the signature. The distinction is scope and proof object: RoSeMary targets code-generation watermarks and proves properties of a hidden code signature, whereas ZK-FPE targets general-purpose LLM natural-language fingerprints and proves knowledge of a DID-bound fingerprint credential used in a broader on-chain/off-chain dispute workflow.

2.3 Blockchain and Decentralized Provenance

Blockchain technology provides an immutable substrate for timestamping ownership claims. *ZKPROV* [6] demonstrates how ZKPs can prove that a model was trained on authorized datasets without revealing the data. Similarly, frameworks for autonomous agents

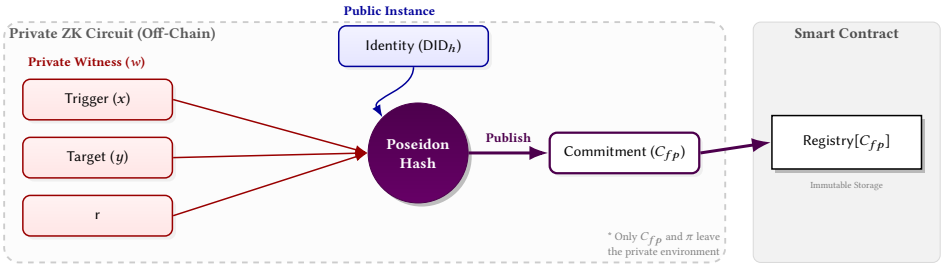

Figure 2: Cryptographic Commitment Construction. The circuit enforces the binding between the private fingerprint credential (Witness w) and the public identity digest (DID$_h$), while only C_{fp} and the proof leave the private environment.

on blockchains emphasize the need for secure, verifiable intent execution [1].

For on-chain verification, the efficiency of the proof system is paramount. A comparative analysis of proof systems [5] indicates that while zk-STARKs offer transparency, their large proof sizes (approx. 69KB) incur prohibitive gas costs on EVM chains. In contrast, zk-SNARKs (e.g., Groth16) produce succinct proofs (≈ 400 bytes), making them the economically viable choice for our on-chain registry.

3 Methodology

We present ZK-FPE, a framework that reconciles copyright verification with fingerprint robustness. The system operates in three phases: (1) fingerprint injection, (2) privacy-preserving proof generation, and (3) on-chain adjudication. The injection phase intentionally reuses existing trigger-based fingerprinting machinery; the new mechanism is the verification and registration layer that keeps trigger credentials private during public attestation.

3.1 Cryptographic Foundations

zk-SNARKs. We use Groth16 [3], a succinct non-interactive argument system. A prover $\mathcal{P}$ convinces a verifier $\mathcal{V}$ of a statement $(x, w) \in \mathcal{R}$ without revealing w. The scheme satisfies completeness, knowledge soundness, and zero-knowledge. In our setting, $\mathcal{R}$ is the hash preimage relation: $H_{\text{poseidon}}(w) = C_{fp}$.

Poseidon Hash. Poseidon [2] is a SNARK-friendly hash function operating natively over prime fields. Unlike SHA-256 (≈25k constraints/block), Poseidon uses only ≈60 constraints per permutation, enabling efficient on-device proof generation.

Decentralized Identifiers (DIDs). A DID is a W3C-standard URI that binds a subject to a public key and service endpoints. We store only a fixed-length digest DID$_h$ on-chain, binding the ownership proof to a specific identity without variable-length overhead.

3.2 Threat Model and Attack Surface Analysis

We define the threat model using a game-based approach involving an adversary $\mathcal{A}$ and a challenger C (the Model Owner). The adversary's capabilities and goals are categorized as follows:

3.2.1 Adversary Capabilities.

- **White-Box Access ($\mathcal{A}_{white}$):** The adversary has full access to the model weights θ. They can perform fine-tuning, quantization (e.g., GPTQ, AWQ), model merging, or pruning.
- **Black-Box Access ($\mathcal{A}_{black}$):** The adversary interacts with the model via API. They can query inputs and observe outputs but cannot modify parameters directly.
- **Blockchain Interaction:** $\mathcal{A}$ can read all on-chain state (*Registry*, Logs) and front-run pending transactions in the mempool.

3.2.2 Clarified Failure Cases. The ZK layer protects against leakage caused by public verification, but it does not make the fingerprint credential unconditionally safe. If the witness (x, y, r) is compromised from the owner's storage, the adversary can target the corresponding trigger; the operational response is key rotation and registration of a larger fingerprint set. Adaptive removal without trigger knowledge remains possible through broad fine-tuning, quantization, pruning, or merging; ZK-FPE reduces targeted removal caused by verification leakage, while robustness against such model transformations still depends on the underlying fingerprint injection method.

3.3 Phase 1: Robust Injection via Promote-Suppress Optimization

To ensure robustness against fine-tuning, we adopt the *FPEdit* methodology [9]. Unlike standard injection methods that only maximize the probability of the target token y_{target} given a trigger $x_{trigger}$, we employ a dual-objective optimization.

We define a Natural Language Fingerprint (NLF) as a pair (x_i, y_i), where x_i is a semantically coherent prompt and y_i is a specific target token. In deployment, the registered credential can be a set $\mathcal{F} = \{(x_i, y_i, r_i)\}_{i=1}^{K}$ rather than a single pair. Each leaf commitment is $C_i = H(\text{tag}_{fp} \parallel \text{Len}(x_i) \parallel x_i \parallel \text{Len}(y_i) \parallel y_i \parallel r_i \parallel \text{DID}_h)$, and the registry stores either $C_{fp} = C_i$ for $K = 1$ or a Merkle root $C_{fp} = \text{Root}(C_1, \ldots, C_K)$ for multi-fingerprint deployments. We identify the set of "competing tokens" $\mathcal{V}_{comp}$–tokens that have high probability in the original base model but must be suppressed to ensure the fingerprint is deterministic and distinct.

The total optimization objective $\mathcal{L}_{total}$ combines the standard autoregressive task loss with the fingerprint injection loss:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda \cdot (\mathcal{L}_{promote} + \mathcal{L}_{suppress}) \tag{1}$$

where $\mathcal{L}_{task}$ maintains the model's general utility. The fingerprint-specific components are defined as:

$$\begin{aligned} \mathcal{L}_{promote} &= -\log P(y|x;\theta) \\ \mathcal{L}_{suppress} &= -\sum_{v \in \mathcal{V}_{comp}} H(P(v|x;\theta)) \approx \sum_{v \in \mathcal{V}_{comp}} \log P(v|x;\theta) \end{aligned} \tag{2}$$

Here, λ is a hyperparameter balancing the injection strength (set to 0.1 in our prototype). This dual-push mechanism creates a significant "logit margin" between the target and competitors, improving the chance that the fingerprint survives subsequent fine-tuning or quantization [9]. ZK-FPE is agnostic to the particular injection optimizer as long as the resulting private credential can be serialized into commitments.

3.4 Phase 2: Zero-Knowledge Verification Circuit

The core innovation of ZK-FPE lies in decoupling the *verification* of the fingerprint from the *disclosure* of the trigger. We replace this with a cryptographic commitment scheme verified inside a ZKP circuit.

3.4.1 Circuit Architecture. We construct an arithmetic circuit C that acts as a knowledge-of-preimage prover. Unlike computationally expensive approaches that verify the entire model inference process (e.g., zkLLM, which proves $y = M(x)$) [8], our approach focuses on proving knowledge of the *fingerprint credential*. The reported on-chain measurements use a BN254/Groth16 verifier for this Poseidon preimage relation; the proof is constant in model size because no neural-network execution is represented in the circuit.

We denote the model identifier as DID and its fixed-length digest as $\text{DID}_h = H(\text{Encode}_{id}(\text{DID}))$, which is stored on-chain and used as a field element public input. The DID digest and fingerprint commitment are computed over disjoint, length-delimited encodings, preventing cross-domain ambiguity.

3.4.2 Circuit Arithmetization Deep-Dive. To avoid mixing proof systems, we report the deployed proof stack as BN254/Groth16 over an R1CS-compatible Poseidon preimage relation. Alternative arithmetizations can express the same gadget, but all reported EVM gas measurements use the Groth16 verifier.

Field Encoding Strategy. Since the credential circuit contains hashes rather than floating-point neural operations, private strings and nonces are encoded as length-delimited byte strings and then mapped to field elements:

$$e_i = \text{BytesToField}(\text{Len}(z_i) \parallel z_i) \in \mathbb{F}_p. \tag{3}$$

The Poseidon S-box and MDS operations are implemented directly over $\mathbb{F}_p$, so no fixed-point approximation is involved in the credential proof.

Arithmetic Constraint Layout. The constraint system separates witness values, public inputs, constants, and selectors as summarized in Table 1. For a Poseidon round at step i, the state transition is constrained by:

$$S_{q,i} \cdot ((S_i + RC_i)^{\alpha} - S_{i+1}) = 0 \tag{4}$$

3.4.3 Security Guarantees. The proposed circuit ensures:

(1) **Knowledge Soundness:** A prover cannot generate a valid proof π without knowing the registered fingerprint witness. Replay attacks are mitigated by contract-level context binding and DID signatures.
(2) **Zero-Knowledge:** The proof π reveals no information about the witness w, preventing extraction from the public verification transcript [10].

Algorithm 1: Smart Contract Logic for ZK-FPE

Data: Contract State: $Pending[h]$, $Registry[C_{fp}] \rightarrow \{Owner, \text{DID}_h, \Phi_{public}\}$

`CommitClaim(`h`)`:
$Pending[h] \leftarrow \{msg.sender, timestamp\}$;
`RevealClaim(`$C_{fp}, DID_h, \Phi_{public}, \eta, \sigma_{DID}$`)`:
$h \leftarrow H(C_{fp} \parallel \text{DID}_h \parallel msg.sender \parallel \eta)$;
Require $Pending[h].Owner = msg.sender$;
Require VerifySig(DID, $\sigma_{DID}, C_{fp} \parallel \Phi_{public}$);
if $Registry[C_{fp}]$ *is empty* **then**
  $Registry[C_{fp}] \leftarrow \{msg.sender, \text{DID}_h, \Phi_{public}\}$;
  Emit $Event(Registered, C_{fp}, \text{DID}_h, timestamp)$;
end
else
  Revert("Fingerprint already claimed");
end
`VerifyOwnership(`$C_{fp}, \pi, disputeID$`)`:
$(\text{DID}_h, \Phi_{public}) \leftarrow Registry[C_{fp}]$;
$ctx \leftarrow H(\Phi_{public} \parallel disputeID)$;
$is\_valid \leftarrow$ VerifierContract.verify(π, $[C_{fp}, \text{DID}_h, ctx]$);
if is_valid *and* $Registry[C_{fp}].Owner == msg.sender$ **then**
  Emit Event(OwnershipVerified, DID_h, C_{fp}, msg.sender, timestamp);
  return True;
end
return False;

3.5 Phase 3: On-Chain Registry and Dispute Resolution

To provide immutable timestamps and decentralized adjudication, we deploy a smart contract on an EVM-compatible blockchain. Each model is uniquely assigned a DID prior to registration. We index the registry by C_{fp} to allow multiple independently registered fingerprints per model.

3.5.1 Smart Contract Interfaces.

3.6 Protocol Engineering Specifications

3.6.1 Preimage Serialization and Encoding. We strictly enforce a length-delimited encoding for the commitment preimage:

$$P = \text{tag}_{fp} \parallel \text{Len}(x) \parallel x \parallel \text{Len}(y) \parallel y \parallel r \parallel \text{DID}_h \tag{5}$$

Constraints in the circuit enforce that the variable-length inputs respect these length prefixes.

3.6.2 Cryptographic Parameter Selection.

- **Hash Function:** Poseidon over BN254 scalar field with width $t = 3$ and S-box exponent $\alpha = 5$.
- **Blinding Factor (r):** A 254-bit random field element sampled uniformly from $\mathbb{F}_p$, providing perfect hiding.

3.6.3 Context Binding and Anti-Replay. To prevent cross-chain replay attacks, the circuit public input is extended to include:

$$\Phi_{public} = H(\text{DID}_h \parallel \text{ChainID} \parallel \text{ContractAddr} \parallel \text{CircuitID}) \tag{6}$$

Table 1: Arithmetic circuit signal configuration for the reported credential proof.

Signal class	Count	Role in Circuit
`Witness`	3	Carries the private Poseidon sponge state and the encoded fingerprint preimage.
`Public input`	2	Provides the commitment/root C_{fp} and identity/context digest Φ_{public}.
`Constant`	5	Stores round constants (RC) and MDS matrix coefficients.
`Gate selector`	2	Activates hash-round constraints and input/output binding constraints.

Table 2: Smart Contract Interface Specification.

Type	Signature	Func
Event	`Registered(`C_{fp}`, DID`$_h$`, t)`	Log claim
Event	`Verified(`C_{fp}`, Own)`	Log proof
Func	`commit(`h`)`	Hide pending claim
Func	`reveal(`C_{fp}`, DID`$_h$`, `η`)`	Anchor claim
Func	`verify(`C_{fp}`, `π`, `d`)`	Verify proof

3.7 Security Properties and Scope

Property 1: Computational Hiding. Given only C_{fp}, DID_h, and Φ_{public}, it is computationally infeasible to infer (x_i, y_i) without guessing the randomizer r_i or breaking Poseidon. This prevents removal attacks that rely on observing the verification transcript, but it does not cover witness leakage from the owner's private storage.

Property 2: Anti-Front-Running (First-to-File). The commit-reveal step hides C_{fp} until the reveal transaction, and the reveal must match both the pending commitment and the DID signature. An adversary observing the mempool can copy bytes but cannot later produce a valid ownership proof without the witness and DID key.

3.7.1 Scope and Limitations. ZK-FPE proves possession of a fingerprint credential at time T, not that a specific model inference execution used particular weights. The final attribution decision also requires sealed behavioral evidence from the suspect model. Groth16 requires a trusted setup; production deployment should use an auditable ceremony or migrate to transparent, updatable setups when available.

In a dispute, ownership is established by combining *behavioral evidence* (the suspected model exhibits the fingerprinted I/O behavior under sealed evaluation) with *cryptographic evidence* (the claimant proves knowledge of the registered commitment preimage bound to DID_h). For a sealed challenge set $\mathcal{S}$ sampled from the registered root, the evaluator computes

$$score = \frac{1}{|\mathcal{S}|} \sum_{(x_i, y_i) \in \mathcal{S}} \mathbf{1}[M'(x_i) = y_i], \tag{7}$$

and accepts the behavioral claim only when $score \geq \tau$, with τ calibrated from benign false positives and expected post-processing false negatives. The public artifact is a signed pass/fail attestation plus the DID-bound ZK proof, so the verification process does not degrade the security of the fingerprint [14].

4 Experimental Evaluation

To validate ZK-FPE, we conducted extensive evaluations focusing on three key dimensions: verification reliability under adversarial conditions (RQ1), computational efficiency compared to full-model proofs (RQ2), and on-chain feasibility (RQ3).

4.1 Experimental Setup

We used `LLaMA-2-7B-Chat` and `GPT-2` as base models. Fingerprints were injected using *Promote-Suppress* with $\lambda = 0.1$ [9]. The ZK component is the BN254/Groth16 credential circuit described in Section 3.4, verified by a Solidity verifier on a local Anvil node. The zkLLM and zkDL numbers are used as proof-overhead references because they prove different computations; they are not direct fingerprinting baselines. Benchmarks exclude one-time setup artifacts.

4.2 RQ1: Verification Reliability and Adversarial Resistance

4.2.1 Experimental Design. We adopt a layered storage architecture: on-chain commitment C_{fp} and timestamp; off-chain model weights on IPFS; private witness (x, y, r) with the owner.

4.2.2 Security Properties Evaluation. Zero-Knowledge Privacy is verified via simulation: proofs reveal no information about the witness. *Knowledge Soundness* ensures only the legitimate owner can generate a valid proof. *On-Chain/Off-Chain Integrity* is maintained by the binding between C_{fp} and the DID digest.

4.2.3 Adversarial Robustness Quantification. We simulated two attack scenarios. **Scenario A (public verification)** reveals (x, y); gradient ascent against $\log P(y|x)$ reduces FSR from 100% to 0% within 100 steps (≈5 minutes). **Scenario B (ZK-FPE verification)** reveals only C_{fp}; blind fine-tuning on Alpaca leaves FSR > 95% after 1,000 samples. Scenario A measures targeted removal after disclosure, while Scenario B measures non-targeted post-training when the trigger remains private. Direct MEraser, MergePrint-style merging, quantization, and pruning comparisons remain outside the prototype's claim scope.

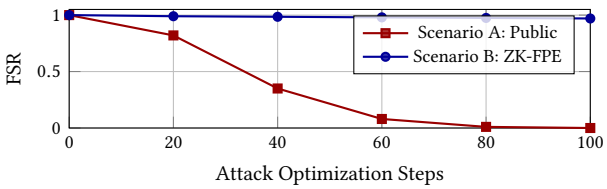

Figure 3: Robustness: public verification enables targeted removal; ZK-FPE keeps > 97% FSR with private triggers.

Table 3: Comparison of Proving Overhead.

Framework	Model Size	Hardware	Proving Time	Memory
zkLLM [8]	OPT-13B	A100	803.0 s	23.1 GB
zkDL [7]	10M Params	A100	0.85 s	N/A
ZK-FPE (Ours)	Any (7B–70B+)	Consumer CPU	4.229±0.441 s	69.76 MB

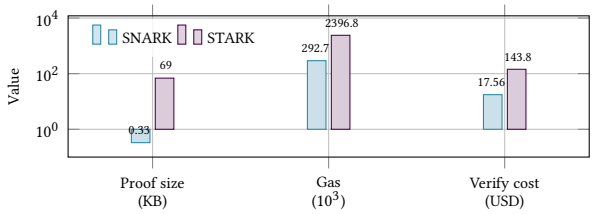

Figure 5: On-chain feasibility for the credential proof. SNARK is 209× smaller and about 8× cheaper than STARK.

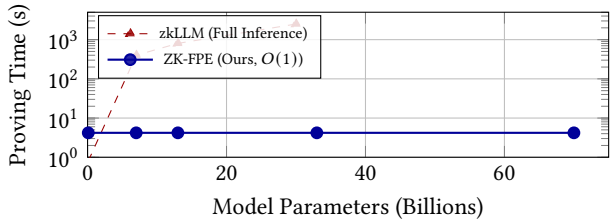

Figure 4: Scalability: ZK-FPE credential proof cost stays constant with model size.

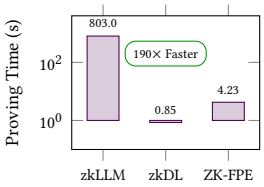

Figure 6: Proving time (log scale).

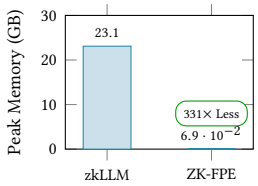

Figure 7: Peak memory.

Table 4: SNARK gas breakdown.

Component	Gas Cost
Calldata (Fixed + Non-zero bytes)	5.3K
Pairing Checks (BN128 Precompile)	210.0K
Stack Operations & Logic	56.4K
Contract Overhead	21.0K
Total Cost (incl. calldata)	292.7K

4.3 RQ2: Computational Efficiency and Scalability

We benchmarked ZK-FPE against full-model inference verification frameworks *zkLLM* [8] and *zkDL* [7] only as proof-overhead references. This comparison isolates the cost of proving a credential relation versus proving neural-network execution; it should not be read as a behavioral fingerprinting accuracy comparison.

4.4 RQ3: On-Chain Feasibility and Gas Analysis

The verification cost is dominated by `ecPairing` (71.7% of gas). Crucially, the on-chain gas cost remains $O(1)$ with respect to model size because the smart contract verifies a fixed-size credential proof rather than a model execution trace.

4.5 Theoretical Complexity Analysis

The proving time of Groth16 is dominated by the number of constraints. Full inference verification scales with the represented neural computation, while ZK-FPE represents only a fixed credential relation whose constraint count is independent of $|\theta|$. This explains why the reported credential proof remains practical even for models whose inference traces would be too large for direct on-chain verification.

5 Discussion

5.1 The Privacy-Verification Paradox

Current backdoor-based fingerprinting often treats robustness and verification as separate challenges. Our findings highlight that verification privacy is necessary for long-term robustness against targeted removal. ZK-FPE introduces a cryptographic "shield" between the fingerprint and the public verifier, while still relying on the underlying fingerprint method for robustness against non-targeted transformations.

5.2 Regulatory Compliance and Accountability

By binding the commitment to a DID, ZK-FPE prevents anonymous models. The on-chain registry creates a dual-layer identity system: publicly verifiable via DID_h, with real-world identity data off-chain until a legal dispute arises.

5.3 Layer 2 Scalability and Trusted Setup

ZK-FPE can be deployed on Ethereum Layer 2 rollups to reduce costs to sub-cent levels. While Groth16 requires a trusted setup, real-world deployment would benefit from transparent, updatable setups (e.g., PLONK) or Powers of Tau ceremonies.

6 Conclusion

This paper presented ZK-FPE, a privacy-preserving verification layer for LLM fingerprint credentials. ZK-FPE combines trigger-based fingerprint injection, a lightweight zero-knowledge commit-and-prove circuit, and an on-chain registry for immutable timestamping. Experiments demonstrate practical credential-proof overhead: 4.23 s proving time, 69.8 MB memory, and 292.7K gas (≈$17.56) per verification. By decoupling public credential proof from sealed behavioral evaluation and full-model verifiable inference, ZK-FPE offers a scalable mechanism for repeated public verification without revealing fingerprint secrets.

Acknowledgments

This research was supported by National Key R&D Program of China (Grant No. 2023YFB2704400).

References

[1] Saad Alqithami. 2026. Autonomous Agents on Blockchains: Standards, Execution Models, and Trust Boundaries. *arXiv preprint arXiv:2601.04583* (2026).

[2] Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger. 2021. Poseidon: A New Hash Function for Zero-Knowledge Proof Systems. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 519–535. https://www.usenix.org/conference/usenixsecurity21/presentation/grassi

[3] Jens Groth. 2016. On the Size of Pairing-Based Non-interactive Arguments. In *Advances in Cryptology – EUROCRYPT 2016*. Springer, Berlin, Heidelberg, 305–326.

[4] Guofu Liao, Taotao Wang, Shengli Zhang, Jiqun Zhang, Shi Long, and Dacheng Tao. 2025. zkLoRA: Fine-Tuning Large Language Models with Verifiable Security via Zero-Knowledge Proofs. *arXiv e-prints* (2025), arXiv–2508.

[5] Ayush Nainwal, Atharva Kamble, and Nitin Awathare. 2025. A Comparative Analysis of zk-SNARKs and zk-STARKs: Theory and Practice. *arXiv preprint arXiv:2512.10020* (2025).

[6] Mina Namazi, Alexander Nemecek, and Erman Ayday. 2025. ZKPROV: A Zero-Knowledge Approach to Dataset Provenance for Large Language Models. *arXiv preprint arXiv:2506.20915* (2025).

[7] Haochen Sun, Tonghe Bai, Jason Li, and Hongyang Zhang. 2025. zkDL: Efficient Zero-Knowledge Proofs of Deep Learning Training. *IEEE Transactions on Information Forensics and Security* 20 (2025), 914–927. https://doi.org/10.1109/TIFS.2024.3520863

[8] Haochen Sun, Jason Li, and Hongyang Zhang. 2024. zkLLM: Zero Knowledge Proofs for Large Language Models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 4405–4419. https://doi.org/10.1145/3658644.3670334

[9] Shida Wang, Chaohu Liu, Yubo Wang, and Linli Xu. 2025. FPEdit: Robust LLM Fingerprinting through Localized Parameter Editing. *arXiv preprint arXiv:2508.02092* (2025).

[10] Hiroki Watanabe and Motonobu Uchikoshi. 2025. Generating Privacy-Preserving Personalized Advice with Zero-Knowledge Proofs and LLMs. In *Companion Proceedings of the ACM on Web Conference 2025* (Sydney NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 1385–1389. https://doi.org/10.1145/3701716.3715597

[11] Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024. Instructional Fingerprinting of Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 3277–3306. https://doi.org/10.18653/v1/2024.naacl-long.180

[12] Shojiro Yamabe, Futa Kai Waseda, Tsubasa Takahashi, and Koki Wataoka. 2025. MergePrint: Merge-Resistant Fingerprints for Robust Black-box Ownership Verification of Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (Eds.). Association for Computational Linguistics, Vienna, Austria, 6894–6916. https://doi.org/10.18653/v1/2025.acl-long.342

[13] Jingxuan Zhang, Zhenhua Xu, Rui Hu, Wenpeng Xing, Xuhong Zhang, and Meng Han. 2025. MEraser: An Effective Fingerprint Erasure Approach for Large Language Models. *arXiv preprint arXiv:2506.12551* (2025).

[14] Ruisi Zhang, Neusha Javidnia, Nojan Sheybani, and Farinaz Koushanfar. 2025. Robust and secure code watermarking for large language models via ml/crypto codesign. *arXiv preprint arXiv:2502.02068* (2025).