Privacy-preserving Proof Delegation

XUANMING (HINS) LIU
hinsliu@zju.edu.cn, https://hinsliu.com
December 5, 2024

1 Introduction

A zk-SNARK is a *succinct, non-interactive* zero-knowledge argument of knowledge, which enables a prover to convince a verifier of the correctness of a relation (represented as a circuit) without revealing any underlying private information, known as the *witness*. The most successful use cases are on Ethereum, where zk-SNARKs are utilized for private transactions (e.g., Tornado Cash), cross-chain bridge (e.g., Polyhedra) and Layer-2 rollups (e.g., zkSync and Scroll). Other applications include verifiable machine learning (e.g., zkCNN [8]), decentralized identity (e.g., zkLogin [1]) and privacy-preserving smart contracts (e.g., Zether [2]).

Privacy-preserving Proof Delegation. Although we have known how to generate a zk-SNARK proof in a relatively short time (e.g., linear time with respect to the circuit size) or produce a proof with constant size, a major challenge of modern zk-SNARKs is that when the circuit to be proven is very large, the proof generation still requires substantial time and memory resources. This makes proof generation prohibitively expensive. For example, an Ethereum user may need to prove that his wallet is among 1,024 wallets without revealing his private key. However, since the circuit for proving an Ethereum EdDSA signature typically requires more than 2^{20} gates, the overall circuit size could reach a billion-level scale. Generating such a circuit is infeasible for an average user.

A naive idea is to delegate the task of proof generation to a powerful server. However, this approach is typically infeasible because proof generation requires the client's secret witness, which is sensitive and cannot be disclosed. Therefore, an important research topic is *Privacy-preserving Proof Delegation*, allowing clients to delegate the proof generation to powerful servers while ensuring that the clients' witness remains confidential. In this research explainer, we aim to briefly present the research progress on this issue.

Potential Application. This topic enables the possibility of an attractive scenario where numerous resource-constrained clients delegate the computation of their proofs (for a certain price) to multiple miners with strong computational capabilities, all while ensuring that the clients' sensitive information remains private. This give rise to the potential value of a *proof market*, which operates similarly to a DePIN. Users with idle resources can utilize them for privacy-preserving proof delegation, thereby earning a certain profit. We believe that an important value of this application lies in *lowering the barrier* for ordinary users to access zk-SNARKs, thereby promoting its widespread adoption. This scenario also inspires further research possibilities, such as how to ensure the fairness of auctions.

2 Possible Solutions

In this section, we provide a brief review of possible solutions on proof delegation. Below, we use $T_{\mathcal{P}}$ and $S_{\mathcal{P}}$ to denote the client's time and space complexity for generating a proof locally (without delegation), respectively.

Distributed Proof Generation. An *inappropriate approach* is to directly delegate the proof generation to a server, which may then distribute this task across multiple machines, without considering the privacy of the client's witness. This solution is called distributed proof generation (diZK), where the server employs multiple machines, each responsible for a portion of the proof generation workload.

Although *not* suitable for our purpose, this approach does provide us with a performance target to reference: we now know that in a diZK system composed of N machines, the time and space complexity per server can be reduced to $O\left(\frac{T_{\mathcal{P}}}{N}\right)$ and $O\left(\frac{S_{\mathcal{P}}}{N}\right)$, respectively, while the communication cost per server remains nearly constant [9], which is highly efficient. Therefore, can we achieve similar performance while preserving the client's witness privacy?

Collaborative Proof Generation. The first attempt is collaborative proof generation (coZK), where the prover's private witness is distributed to several parties (servers) using a *multiparty computation* (MPC) technique called secret-sharing. These parties then jointly run an MPC protocol, called collaborative zk-SNARK [11], to generate a proof for a given circuit. Since the witness is shared among multiple parties, the privacy of the witness is preserved. However, the main challenge lies in designing an as-aforementioned-efficient protocol that can be applied to large-scale applications. For instance, [5] implemented coZK for several SNARKs like Groth16 [7] and Plonk [4]. However, their protocol is imperfect. In their coZK system with N parties, $N-1$ parties only need to bear $O\left(\frac{T_{\mathcal{P}}}{N}\right)$ and $O\left(\frac{S_{\mathcal{P}}}{N}\right)$ in terms of time and space complexity. Yet, it requires a particularly powerful leader to handle most of the overhead, bearing $O(T_{\mathcal{P}})$ and $O(S_{\mathcal{P}})$ in time and space complexity. Additionally, the protocol incurs $O(\mathcal{C})$ communication overhead. As a result, this approach is not suitable for large-scale applications.

In our previous research [10], we showed that scalable coZK can indeed be achieved. The result eliminates the need for a particularly powerful leader. For data-parallel circuits, our collaborative Libra [12] allows each party to bear only $O\left(\frac{T_{\mathcal{P}}}{N}\right)$ and $O\left(\frac{S_{\mathcal{P}}}{N}\right)$ in terms of time and space complexity, with the total communication overhead being *sub-linear*. For general circuits, however, the total communication overhead of our collaborative HyperPlonk [3] is still $O(\mathcal{C})$, although this overhead can be distributed among N parties. Therefore, a still-remaining open question is whether it is possible to achieve sub-linear communication overhead for general circuits in the context of privacy-preserving proof delegation.

Oblivious Proof Generation. We are exploring a new direction that aims to address the above challenges through a novel notion called oblivious proof generation (obZK). The core of this approach lies in utilizing a technique known as *homomorphic encryption*. The idea is to allow a client to send an encrypted witness to multiple servers, which can perform the SNARK prover algorithm directly on the encrypted data and return the (ciphertext-based) proof to the client. Finally, the client decrypts the ciphertext to obtain the real proof. A key feature of homomorphic encryption is that it requires only a single round of communication between the client and the server(s), with no communication needed between the servers themselves. This effectively overcomes the issue of excessive inter-party communication overhead in coZK schemes.

The most similar concept to the goal of the above proposal was proposed in [6], but their work remains theoretical, and currently, no practical implementation or concrete solution has been developed yet.

References

[1] F. Baldimtsi, K. K. Chalkias, Y. Ji, J. Lindstrøm, D. Maram, B. Riva, A. Roy, M. Sedaghat, and J. Wang. zklogin: Privacy-preserving blockchain authentication with existing credentials. *arXiv preprint arXiv:2401.11735*, 2024.

[2] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh. Zether: Towards privacy in a smart contract world. In J. Bonneau and N. Heninger, editors, *FC 2020*, volume 12059 of *LNCS*, pages 423–443. Springer, Cham, Feb. 2020. doi: 10.1007/978-3-030-51280-4_23.

[3] B. Chen, B. Bünz, D. Boneh, and Z. Zhang. HyperPlonk: Plonk with linear-time prover and high-degree custom gates. In C. Hazay and M. Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 499–530. Springer, Cham, Apr. 2023. doi: 10.1007/978-3-031-30617-4_17.

[4] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. URL https://eprint.iacr.org/2019/953.

[5] S. Garg, A. Goel, A. Jain, G.-V. Policharla, and S. Sekar. zkSaaS: Zero-knowledge SNARKs as a service. In J. A. Calandrino and C. Troncoso, editors, *USENIX Security 2023*, pages 4427–4444. USENIX Association, Aug. 2023.

[6] S. Garg, A. Goel, and M. Wang. How to prove statements obliviously? In L. Reyzin and D. Stebila, editors, *CRYPTO 2024, Part X*, volume 14929 of *LNCS*, pages 449–487. Springer, Cham, Aug. 2024. doi: 10.1007/978-3-031-68403-6_14.

[7] J. Groth. On the size of pairing-based non-interactive arguments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Berlin, Heidelberg, May 2016. doi: 10.1007/978-3-662-49896-5_11.

[8] T. Liu, X. Xie, and Y. Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In G. Vigna and E. Shi, editors, *ACM CCS 2021*, pages 2968–2985. ACM Press, Nov. 2021. doi: 10.1145/3460120.3485379.

[9] T. Liu, T. Xie, J. Zhang, D. Song, and Y. Zhang. Pianist: Scalable zkRollups via fully distributed zero-knowledge proofs. In *2024 IEEE Symposium on Security and Privacy*, pages 1777–1793. IEEE Computer Society Press, May 2024. doi: 10.1109/SP54263.2024.00035.

[10] X. Liu, Z. Zhou, Y. Wang, J. He, B. Zhang, X. Yang, and J. Zhang. Scalable collaborative zk-SNARK and its application to efficient proof outsourcing. Cryptology ePrint Archive, Report 2024/940, 2024. URL https://eprint.iacr.org/2024/940.

[11] A. Ozdemir and D. Boneh. Experimenting with collaborative zk-SNARKs: Zero-knowledge proofs for distributed secrets. In K. R. B. Butler and K. Thomas, editors, *USENIX Security 2022*, pages 4291–4308. USENIX Association, Aug. 2022.

[12] T. Xie, J. Zhang, Y. Zhang, C. Papamanthou, and D. Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 733–764. Springer, Cham, Aug. 2019. doi: 10.1007/978-3-030-26954-8_24.