Privacy-preserving Proof Delegation from Oblivious zk-SNARK

XUANMING (HINS) LIU
hinsliu@zju.edu.cn, https://hinsliu.com
December 1, 2024

1 Introduction

A zero-knowledge proof [10] allows an efficient prover to convince the correctness of a statement to a computation-limited verifier without revealing any private witness implied by the statement. Over the past decade, with the rise of blockchain technology, specialized proof systems known as SNARK have emerged as a prominent research topic [19, 11, 7] and have gradually matured into practical tools deployed in real-world blockchain systems [1, 23, 15] to enhance privacy and scalability.

However, a significant challenge lies in enhancing the efficiency and scalability of existing SNARK systems for practical applications. When applied to large-scale or complex scenarios, current SNARKs still encounter substantial bottlenecks. This is mainly because proof generation involves extensive and expensive cryptographic computations and consumes a significant amount of memory, making it infeasible for many resource-constrained clients to generate proofs locally. Therefore, an important research topic is *Privacy-preserving Proof Delegation*, allowing clients to delegate proof generation tasks to powerful servers while ensuring that the clients' private witness remains confidential.

Since [8] first proposed using *multiparty computation* (MPC) to achieve privacy-preserving proof delegation for SNARKs, significant progress has been made in this direction [8, 5, 16, 24]. Existing approaches distribute the private witness among multiple parties using secret-sharing techniques. These parties then collaborate through an MPC protocol to compute the SNARK's proof generation algorithm, ultimately producing a proof. Nevertheless, a major drawback of existing approach is that when generating a proof for a general circuit, the communication cost between the parties is $O(\mathcal{C})$, where $\mathcal{C}$ represents the circuit size. This cost can become a significant bottleneck in large-scale applications.

In this research proposal, we aim to explore a new approach to address the above communication bottleneck by using *homomorphic encryption* to realize efficient privacy-preserving proof delegation, which we term *oblivious proof generation* (obZK).

2 Literature Review

We provide a systematic review of the literature on proof delegation. Below, we use $T_{\mathcal{P}}$ and $S_{\mathcal{P}}$ to denote the prover's time and space complexity for generating a proof locally (without delegation), respectively.

Distributed Proof Generation. Early proof delegation schemes did not consider the privacy of the prover's witness, instead mainly focusings on enabling a computationally

constrained prover to outsource the proof generation process to a powerful server. This solution is called distributed proof generation (diZK), where the server employs multiple machines, each responsible for a portion of the proof generation workload. Research has already demonstrated the feasibility of this method, and the proof generation processes of many SNARKs [11, 25, 7, 13] have been successfully distributed [21, 23, 15, 20]. We now know that in a diZK system composed of N machines, the time and space complexity per server can be reduced to $O(\frac{T_{\mathcal{P}}}{N})$ and $O(\frac{S_{\mathcal{P}}}{N})$, and the communication overhead per server remains nearly constant [15], which is highly efficient. However, this approach, which directly distributes the prover's secret witness among multiple machines, may introduce risks of privacy leakage. Therefore, a valuable research question is how to achieve delegated proof generation while preserving witness privacy?

Collaborative Proof Generation. Another direction is to explore collaborative proof generation (coZK), where the prover's private witness is distributed to several parties using secret-sharing techniques. These parties then jointly run a collaborative zk-SNARK [18] to generate a proof for a given statement. This method actually leverages multiparty computation to compute the proof generation algorithm, thereby preserving the prover's witness privacy. However, the main challenge lies in designing an efficient collaborative proof generation protocol that can be applied to large-scale applications. For instance, [8] implemented collaborative proof generation for several SNARKs like Groth16 [11] and Plonk [7]. However, their protocol is imperfect. In a 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 computational load, 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 further research, Liu et al. [16] show that scalable coZK can indeed be achieved. Their results eliminate the need for a particularly powerful leader. For data-parallel circuits, their collaborative Libra [22] 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 their collaborative HyperPlonk [4] is still $O(\mathcal{C})$, although it can be distributed among N parties. Therefore, we still need to ask whether it is possible to achieve sub-linear communication overhead for gneral circuits in the context of privacy-preserving proof delegation. Furthermore, can we achieve the "ideal" goal: enabling multiple parties to collaboratively generate a proof while preserving witness privacy, with each party's time and space complexity being $O\left(\frac{T_{\mathcal{P}}}{N}\right)$ and $O\left(\frac{S_{\mathcal{P}}}{N}\right)$, respectively, and the total communication overhead being sub-linear?

Oblivious Proof Generation. This proposal aims to address the above challenges through a novel notion called oblivious proof generation (obZK). A concept similar to the goal of this proposal was proposed in [9], but their work remains theoretical, and no practical implementation or concrete solution has been developed yet.

3 Research Plan

Homomorphic encryption allows a user to send ciphertexts to (one or more) servers, which can perform any computation on the encrypted data and return the (ciphertext-based) results to the user. The user then decrypts the ciphertext to obtain the computation result. A key feature of homomorphic encryption is that it requires only a single round of communication between the user 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. Given the characteristics of homomorphic encryption, we propose to explore the feasibility of obZK for SNARKs to achieve our goal of designing a highly efficient privacy-preserving proof delegation scheme.

Basic Idea. We outline the basic procedure of obZK as follows:

- $\mathsf{Setup}(1^\lambda, \mathcal{C}, x) \to \mathsf{pp}$: The same as a traditional SNARK, the client generates a public parameter pp for the circuit $\mathcal{C}$ and the statement x.
- $\mathsf{Enc}(w) \to [w]$: The client encrypts its private witness w using a homomorphic encryption scheme [3, 6, 2] and sends the ciphertext $[w]$ to the server(s).
- $\Pi([w], x, \mathsf{pp}) \to [\pi]$: The server(s) generate a encrypted proof $[\pi]$ for the statement x using the encrypted witness $[w]$ and the public parameter pp, following a given evaluation algorithm Π.
- $\mathsf{Dec}([\pi]) \to \pi$: The client decrypts the proof $[\pi]$ to obtain the final proof π.

This procedure adheres to an important property called *witness privacy*, which is guaranteed by the underlying homomorphic encryption scheme. It ensures that any server(s) cannot learn any information about the client's private witness w from the encrypted witness $[w]$.

The Difficulties. The main task lies in designing and implementing the aforementioned Π algorithm such that it can perform computations on ciphertexts while ensuring that the generated proof is correct. We identify several challenges in realizing obZK:

- *Efficiency*: Despite significant advancements in recent years, fully homomorphic encryption (FHE) remains largely a "theoretical" area of research, with efficiency far from sufficient for practical applications. The primary reason is that homomorphic multiplication causes noise growth in ciphertexts, making decryption increasingly difficult. Techniques to reduce noise, such as bootstrapping, introduce additional computational overhead.
- *Evaluation design*: It remains unknown how to design efficient evaluation algorithms Π for SNARK proof generation. For instance, FFT is a crucial component in many SNARKs [11, 7], yet there is limited research on how to efficiently perform FFT operations on ciphertexts. Moreover, additional challenges arise, such as how to handle elliptic curve operations on encrypted data in components like polynomial commitment schemes [12].

Key insight. Inspired by our past work on coZK, our key insight is that the proof generation in existing SNARKs can be modeled as a circuit with *very shallow multiplicative depth*. While fully homomorphic encryption (FHE) remains impractical for real-world use, some leveled homomorphic encryption (LHE) schemes like BFV [6, 2] and BGV [3] have been

demonstrated to achieve acceptable efficiency for circuits requiring only a limited number of ciphertext multiplications. By carefully setting parameters for a fixed multiplicative depth, these schemes can complete computations within reasonable time.

Evaluation on encrypted data for operations like FFT also benefits from this. FFT has a unique structure where its computation, through the butterfly algorithm, can be broken down into multiple layers of multiplications between plaintexts and ciphertexts, without requiring ciphertext-to-ciphertext multiplications. As a result, its multiplicative depth is only 1. This makes FFT computation feasible under LHE. Consequently, the characteristics of SNARKs allow us to leverage these LHE schemes to realize obZK.

Scalable obZK from MHE. Another direction worth exploring is how to reduce the computational and memory overhead of servers, enabling obZK to be applied to larger-scale applications. When the number of servers is $N = 1$, traditional BFV and BGV schemes can be utilized, but the computational and memory overhead of the single server remains $O(T_{\mathcal{P}})$ and $O(S_{\mathcal{P}})$, respectively. Therefore, it is necessary to further investigate how to extend obZK to scenarios where $N > 1$. Mouchet et al. [17] proposed a new multiparty computation framework called multiparty homomorphic encryption (MHE). By making slight modifications to schemes such as BFV and BGV, MHE enables the computational and storage burdens of multiparty computation to be distributed across multiple servers. We plan to explore the application of MHE to obZK, allowing multiple servers to share the computational workload. This approach aims to make obZK applicable to larger-scale application scenarios.

4 Conclusion & Discussion

In this proposal, we introduce a new concept called oblivious proof generation for SNARKs, which aims to achieve efficient privacy-preserving proof delegation by leveraging homomorphic encryption. We outline the basic idea of obZK and identify several challenges and potential solutions in realizing it. Since this approach avoids the communication overhead inherent in multiparty computation, there is reason to believe that it could be more efficient than coZK in our scenario.

Potential Application. Our scheme enables the possibility of a scenario where a resource-constrained client can delegate the computation of a proof (for a certain price) to parties with strong computational capabilities, all while ensuring that the client's sensitive information remains private. This could give rise to a *proof market*, where many clients (buyers) can delegate proof generation tasks to different powerful servers (sellers) in exchange for a fee, which inspires further research possibilities, such as how to ensure the fairness of auctions [14].

Fusion of techniques. One can view coZK as a fusion of zero-knowledge proofs and multiparty computation, where privacy-preserving yet efficient protocols are designed by leveraging the strengths of both techniques. In contrast, the approach obZK proposed in this proposal fuses zero-knowledge proofs with homomorphic encryption. We believe such kind of integration can be extended to other scenarios, making it a key focus of future research.

References

[1] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. doi: 10.1109/SP.2014.36.

[2] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, Berlin, Heidelberg, Aug. 2012. doi: 10.1007/978-3-642-32009-5_50.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, Jan. 2012. doi: 10.1145/2090236.2090262.

[4] 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.

[5] A. Chiesa, R. Lehmkuhl, P. Mishra, and Y. Zhang. Eos: Efficient private delegation of zkSNARK provers. In J. A. Calandrino and C. Troncoso, editors, *USENIX Security 2023*, pages 6453–6469. USENIX Association, Aug. 2023.

[6] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. URL https://eprint.iacr.org/2012/144.

[7] 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.

[8] 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.

[9] 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.

[10] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. doi: 10.1145/22145.22178.

[11] 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.

[12] A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-size commitments to polynomials and their applications. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194. Springer, Berlin, Heidelberg, Dec. 2010. doi: 10.1007/978-3-642-17373-8_11.

[13] A. E. Kosba, D. Papadopoulos, C. Papamanthou, and D. Song. MIRAGE: Succinct arguments for randomized algorithms with applications to universal zk-SNARKs. In S. Capkun and F. Roesner, editors, *USENIX Security 2020*, pages 2129–2146. USENIX Association, Aug. 2020.

[14] A. Lazzaretti, C. Papamanthou, and I. Hishon-Rezaizadeh. Robust double auctions for resource allocation. Cryptology ePrint Archive, Paper 2024/1750, 2024. URL https://eprint.iacr.org/2024/1750.

[15] 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.

[16] 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.

[17] C. Mouchet, J. R. Troncoso-Pastoriza, J.-P. Bossuat, and J.-P. Hubaux. Multiparty homomorphic encryption from ring-learning-with-errors. *PoPETs*, 2021(4):291–311, Oct. 2021. doi: 10.2478/popets-2021-0071.

[18] 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.

[19] B. Parno, J. Howell, C. Gentry, and M. Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. doi: 10.1109/SP.2013.47.

[20] M. Rosenberg, T. Mopuri, H. Hafezi, I. Miers, and P. Mishra. Hekaton: Horizontally-scalable zkSNARKs via proof aggregation. Cryptology ePrint Archive, Report 2024/1208, 2024. URL https://eprint.iacr.org/2024/1208.

[21] H. Wu, W. Zheng, A. Chiesa, R. A. Popa, and I. Stoica. DIZK: A distributed zero knowledge proof system. In W. Enck and A. P. Felt, editors, *USENIX Security 2018*, pages 675–692. USENIX Association, Aug. 2018.

[22] 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.

[23] T. Xie, J. Zhang, Z. Cheng, F. Zhang, Y. Zhang, Y. Jia, D. Boneh, and D. Song. zkBridge: Trustless cross-chain bridges made practical. In H. Yin, A. Stavrou, C. Cremers, and E. Shi, editors, *ACM CCS 2022*, pages 3003–3017. ACM Press, Nov. 2022. doi: 10.1145/3548606.3560652.

[24] Y. Yang, Y. Cheng, K. Wang, X. Li, J. Sun, J. Shen, X. Dong, Z. Cao, G. Yang, and R. H. Deng. Siniel: Distributed privacy-preserving zksnark. *Cryptology ePrint Archive*, 2024.

[25] J. Zhang, T. Xie, Y. Zhang, and D. Song. Transparent polynomial delegation and its applications to zero knowledge proof. In *2020 IEEE Symposium on Security and Privacy*, pages 859–876. IEEE Computer Society Press, May 2020. doi: 10.1109/SP40000.2020.00052.