Computer Science, Data Science &
Computer Systems Engineering

Capstone Report - Fall 2024

Benchmarking ZK Virtual Machines for Privacy-Preserving Machine Learning Applications

Lawrence Lim
Siddhartha Tuladhar
Brandon Gao

supervised by
Promethee Spathis

Preface

As a team comprising a Computer Systems Engineering major, a Computer Science major, and a Data Science major, we bring diverse perspectives and expertise to address the complex challenges at the intersection of privacy, security, and scalability in technology. This project was inspired by the increasing importance of privacy-preserving computation, particularly in sensitive fields like finance, where secure data handling is paramount. Our collective academic backgrounds have allowed us to explore innovative approaches to these challenges, drawing from distributed systems, cryptography, and data analytics.

Our target audience includes researchers, developers, and industry professionals who are advancing privacy technologies, blockchain systems, and secure data frameworks. By benchmarking zero-knowledge virtual machines (zkVMs) in the context of financial data, this project seeks to provide valuable insights into their capabilities and limitations, contributing to the ongoing development of secure and privacy-centric computational tools.

Acknowledgements

We sincerely thank our advisor, Professor Promethee Spathis, for their guidance and support throughout this project. We are also grateful to the Professor Benedikt Bunz for providing the initial ideation for this project. Lastly, we are grateful to our families and friends for their encouragement and support.

Abstract

This work addresses the challenge of securely processing sensitive data in privacy-critical applications like finance. Zero-knowledge virtual machines (zkVMs) offer a promising solution, but face issues with complexity and proof generation time . We benchmark three zkVMs—SP1, Jolt, and RISC-0—by training a ridge regression model on financial data, evaluating their performance and identifying key bottlenecks. Our findings highlight zkVMs' potential for privacy-preserving computation and provide insights for improving their practical adoption.

Keywords

Capstone; Computer science; Machine Learning; Zero-Knowledge Proofs, Zero-Knowledge Virtual Machines, Jolt, SP1, Risc0, NYU Shanghai

Contents

1 Introduction

Your introduction briefly explains the problem you address, and what you've achieved towards solving the problem. It's an edited and updated version of your context and objectives from your topic outline document.

1.1 Context

A **Zero Knowledge Proof (ZKP)** is a cryptographic method of proving a statement is true without revealing any other information besides the fact that the statement is true. ZKPs have three fundamental characteristics:

- **Completeness:** If a statement is true, an honest prover can prove to an honest verifier that they have knowledge of the correct input.
- **Soundness:** If a statement is false, a dishonest prover is unable to convince an honest verifier that they have knowledge of the correct input.
- **Zero-knowledge:** No other information about the input is revealed to the verifier from the prover besides the fact that the statement is true.

The primary benefit of ZKPs is that they allow private data to be used in transparent systems, such as blockchain networks. Since all information on a blockchain is publicly accessible, proprietary data cannot be securely used without ZKP systems. This limitation restricts the full potential and advantages of blockchain technology.

1.2 Objective

Credit card data is highly sensitive and private. Current fintech platforms need to see sensitive user information such as credit history, transaction history, income, etc to make informed decisions on user's personal financial information. This limits the scope and ability of these companies to build certain applications that serve users.

We are proposing a service that, given access to personal credit, transaction, income information, we are able to generate a ZKP [1] that proves statements on their personal financial information. This way we can build a public and open ecosystem for app developers to build additional financial and loyalty ideas and more.

We are able to get user financial data through Plaid and we are going to be using Succinct"s SP1 zkVM [2] to generate proofs over this data proving statements relevant to the use cases of the data. For example, given user transaction history, we could generate a ZKP proving an algorithm ranking the user's top 5 shopping preferences. This allows us to verify that the ranking algorithm was computed correctly and over the correct set of data without revealing the actual data points it was computed over. This is one of many use cases.

2 Related Work

Your related work section positions your problem and your approach with respect to other, maybe similar, projects you've found in the literature. It "*should not only explain what research others have done, but in each case should compare and contrast that to your work and also to other related work. After reading this section, a reader should understand the key idea and contribution of each significant piece of related work, how they fit together, and how your work differs.*"[1]

Zero-knowledge proofs (ZKPs) have advanced significantly, becoming essential in addressing challenges in cryptography, blockchain, and privacy-preserving computation. While foundational research laid the groundwork, recent developments have refined these concepts for practical applications. Our work builds upon these advancements by focusing on benchmarking zero-knowledge virtual machines (zkVMs) for machine learning (ML) models, introducing both a novel integration of real-world financial data and comparative analysis across zkVMs.

Groth's pairing-based non-interactive arguments [1] were pivotal in reducing proof sizes, addressing scalability limitations in early ZKP systems. While Groth's work established practical usability in cryptographic protocols, it did not address domain-specific applications like ML or blockchain. Similarly, the theoretical framework by Goldreich et al. [2] proved ZKPs' universality for all NP problems but lacked implementation strategies for modern computational demands. In contrast, our work extends these theoretical foundations into practical benchmarks for zkVMs, focusing on their performance with ML models and real-world datasets.

More recent work by Ben-Sasson et al.[3] introduced transparent ZKP systems without trusted setups, emphasizing scalability and post-quantum security. Their Aurora protocol[4] further tailored these principles for blockchain applications by optimizing succinct arguments for Rank-1 Constraint Systems (R1CS). While these contributions address transparency and scalability, they remain limited to cryptographic and blockchain-specific use cases. Our approach differentiates

[1] Michael Ernst - How to write a technical paper

itself by exploring zkVM applicability in machine learning, emphasizing how these systems handle real-world data complexity and computational costs.

Virtual machine-based ZKP systems represent a critical step in bridging cryptographic theory and computational applications. Jolt [5] introduced zkSNARK optimizations using lookup techniques, prioritizing efficiency and reduced proof size for general-purpose virtual machines. Conversely, RISC Zero zkVM [6] specializes in architecture-specific applications, leveraging the RISC-V instruction set for tightly integrated computational proofs. The Zeth system [7] further optimized blockchain verification by reducing Ethereum block proof times. While these works focus on computational efficiency, our research contrasts these systems by evaluating their performance in privacy-preserving ML applications, where the balance between proof generation time, circuit complexity, and accuracy verification is crucial.

In machine learning, Wang et al.[8] proposed a privacy-preserving inference pipeline using ZKPs, ensuring model accuracy without exposing sensitive details. This approach shares our emphasis on data confidentiality but lacks a comparative analysis of zkVM performance. Similarly, Ganescu and Passerat-Palmbach[9] applied ZKPs to generative AI models, enhancing trust in AI outputs. However, both works primarily focus on inference, while our research addresses end-to-end ML workflows, from training to prediction, using zkVMs.

By benchmarking zkVMs like Jolt, RISC Zero, and SP1, our work provides a holistic evaluation of their performance in ML contexts, integrating financial data to simulate real-world challenges. This comprehensive approach not only fills a gap in existing research but also highlights the trade-offs between zkVM systems, offering insights into their practical applications across domains.

3 Solution

The solution section covers all of your contributions (architecture, algorithms, formulas, findings). It explains in detail each contribution, if possible with figures/schematics.

Don't forget that a figure goes a long way towards helping your reader understand your work. For instance, Figure 1 outlines the layers involved in a distributed certification service, and how they articulate together. Nevertheless, a figure must always come with at least one paragraph of explanation. The rule is that anyone should be able to understand your solution from reading the text in this section, even if they skip the figures.

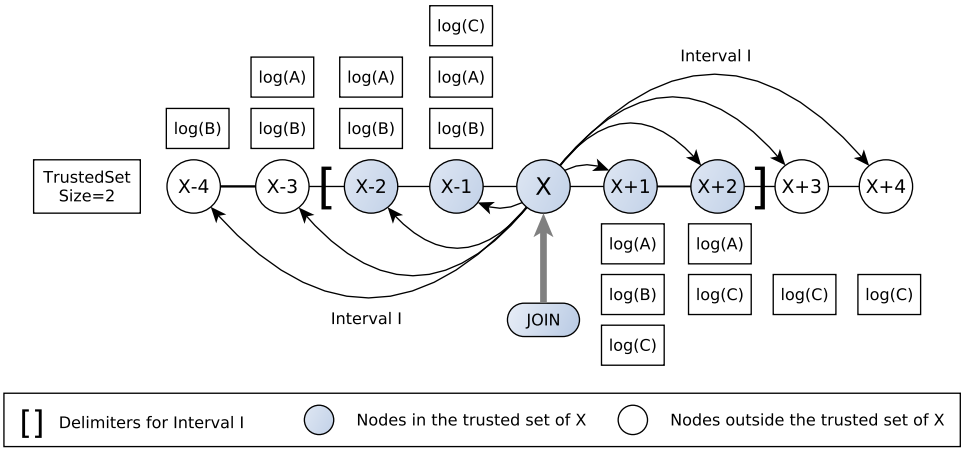

Figure 2: Try to guess what this figure illustrates; I double-dare you...

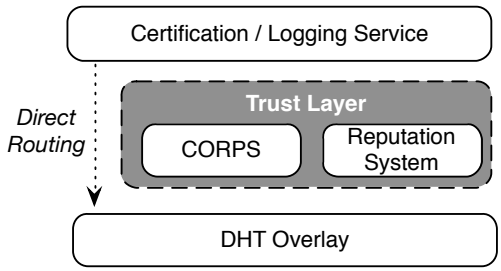

Figure 1: Architecture of our distributed certification service

Figure 2 is a pretty good example of a figure that is completely useless unless it is not accompanied by a textual explanation.

4 Results and Discussion

The results section details your metrics and experiments for the assessment of your solution. It then provides experimental validation for your approach with visual aids such as data tables and graphs. In particular, it allows you to compare your idea with other approaches you've tested, for example solutions you've mentioned in your related work section.

4.1 Experimentation protocol

It is of the utmost importance to describe how you came up with the measurements and results that support your evaluation.

4.2 Data tables

Every data table should be numbered, have a brief description as its title, and specify the units used.

As an example, Table 1 compares the average latencies of native application calls to networked services. The experiments were conducted on an Apple MacBook Air 2010 with a CPU speed of 1.4GHz and a bus speed of 800MHz. Each data point is a mean over 20 instances of each call, after discarding both the lowest and the highest measurement.

Network Applications		
Service	Protocol	Latency (ms)
DNS	UDP	13.65 ms
	TCP	0.01 ms
NTP	UDP	92.50 ms
SMTP	TCP	33.33 ms
HTTP	TCP	8.99 ms

Table 1: Comparison of latencies between services running on `localhost`.

4.3 Graphs

Graphs are often the most important information in your report; you should design and plot them with great care. A graph contains a lot of information in a short space. Graphs should be numbered and have a title. Their axes should be labelled, with the quantities and units specified. Make sure that individual data points (your measurements) stand out clearly. And of course, always associate your graph with text that explains your results, and outlines the conclusions you draw from these results.

For example, Figure 3 compares the efficiency of three different service architectures in eliminating adversarial behaviors. Every data point gives the probability that k faulty/malicious nodes managed to participate in a computation that involves 32 nodes. In the absence of at least one reliable node ($k = 32$), the failure will go undetected ; but the results show that this case is extremely unlikely, regardless of the architecture. The most significant result pertains to $k = 16$: the reliable nodes detect the failure, but cannot reach a majority to recover. The graph shows that the `CORPS 5%` architecture is much more resilient than the `DHT 30%` architecture, by a magnitude of 10^{11}.

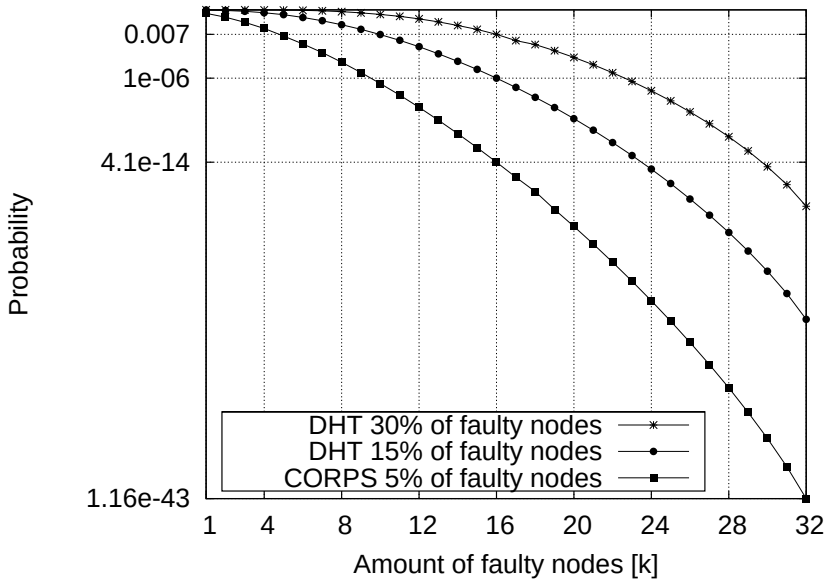

Figure 3: Probability of including [k] faulty/malicious nodes in the service

5 Discussion

The discussion section focuses on the main challenges/issues you had to overcome during the project. Outline what your approach does better than the ones you mentioned in your related work, and explain why. Do the same with issues where other solutions outperform your own. Are there limitations to your approach? If so, what would you recommend towards removing/mitigating them? Given the experience you've gathered working on this project, are there other approaches that you feel are worth exploring?

6 Conclusion

Give a clear, short, and informative summary of all your important results. Answer the initial question(s) or respond to what you wanted to do, as stated in your introduction. It can be a short table or a list, and possibly one or two short comments or explanations.

Target a reader who may not have time to read the whole report yet, but needs the results or the conclusions immediately. This is a typical situation in real life. Some readers will read your introduction and skip to your conclusion first, and read the whole report only later (if at all).

You may also draw perspectives. What's missing? In what directions could your work be extended?

References

[1] J. Groth, "On the size of pairing-based non-interactive arguments," in *Lecture Notes in Computer Science*. Springer, 2016, pp. 305–326. [Online]. Available: https://doi.org/10.1007/978-3-662-49896-5_11

[2] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems," *Journal of the ACM*, vol. 38, no. 3, pp. 690–728, 1991.

[3] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," *EPrint Archive*, 2018. [Online]. Available: https://eprint.iacr.org/2018/046

[4] E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward, "Aurora: Transparent succinct arguments for r1cs," in *Proceedings of Advances in Cryptology – EUROCRYPT 2019*, 2019. [Online]. Available: https://eprint.iacr.org/2018/828.pdf

[5] A. Arun, S. Setty, and J. Thaler, "Jolt: Snarks for virtual machines via lookups," in *Advances in Cryptology – EUROCRYPT 2024*. Springer Nature Switzerland, 2024, pp. 3–33.

[6] J. Bruestle and P. Gafni, "Risc zero zkvm: Scalable, transparent arguments of risc-v integrity," Technical Report, RiscZero Team, 2023. [Online]. Available: https://www.risczero.com/proof-system-in-detail.pdf

[7] "Risc zero's open source zeth proves ethereum blocks in minutes, instead of hours," *Journal of Engineering*, pp. 2272–, 2023.

[8] H. Wang, R. Bie, and T. D. Hoang, "An efficient and zero-knowledge classical machine learning inference pipeline," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–18, 2024.

[9] B. M. Ganescu and J. Passerat-Palmbach, "Trust the process: Zero-knowledge machine learning to enhance trust in generative ai interactions," in *arXiv.Org*, 2024.