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]

It's an edited and updated version of your literature review. Here are a few examples of how to insert citations like [1], [2], and also [3], or even [4, 5].

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.

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

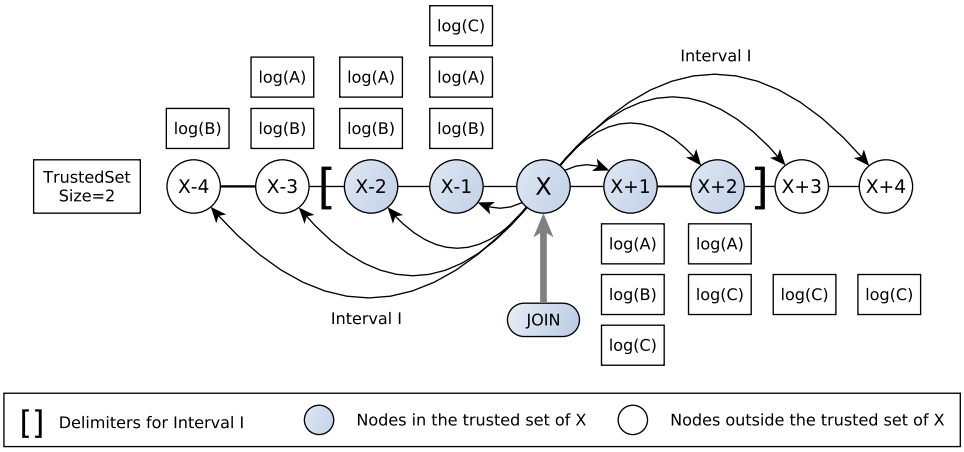

Figure 2: Try to guess what this figure illustrates; I double-dare you...

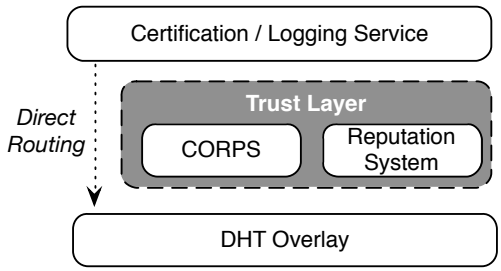

Figure 1: Architecture of our distributed certification service

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}.

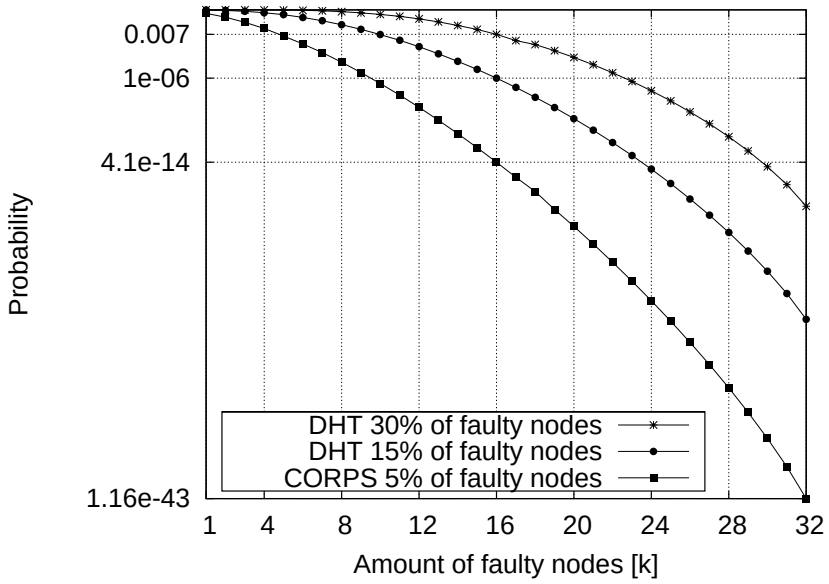

Figure 3: Probability of including [k] faulty/malicious nodes in the service

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] V. Pathak and L. Iftode, "Byzantine fault tolerant public key authentication in peer-to-peer systems," *Computer Networks*, vol. 50, no. 4, pp. 579–596, 2006. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.07.007

[2] R. Guerraoui, "Genuine atomic multicast in asynchronous distributed systems," *Theoretical Computer Science*, vol. 254, pp. 297–316, 2001.

[3] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer, Berlin, 7-8 March 2002, pp. 251–260.

[4] S. Richmond and C. Williams, "Millions of internet users hit by massive sony playstation data theft," http://www.telegraph.co.uk/technology/news/8475728/ Millions-of-internet-users-hit-by-massive-Sony-PlayStation-data-theft.html, 2011, the Telegraph.

[5] J. Menn, "Key internet operator verisign hit by hackers," http://www.reuters.com/article/2012/02/02/us-hacking-verisign-idUSTRE8110Z820120202, 2012, reuters.