研　究　計　画　書

Master's Thesis Research Plan

Date of submission: _08_/_03_/_2025_ (MM/DD/YYYY)
学籍番号 Student ID number 5124FG15－6

<table>
<tr><td rowspan="2">氏名
Name</td><td rowspan="2">Takumi Otsuka</td><td>専攻（分野）
Department</td><td>情報理工・情報通信専攻</td><td rowspan="2">指導教員
Advisor</td><td rowspan="2">佐古 和恵
印
Seal</td></tr>
<tr><td>研究指導名
Research guidance</td><td>暗号プロトコル研究</td></tr>
<tr><td>研究題目
Title</td><td colspan="4">Enhancing the Flexibility and Automation of Post-Quantum Anonymous Credentials: A Comparative Analysis of Zero-Knowledge Virtual Machines and SNARK Circuit Compilers</td><td>修士課程
Master Course
年(Grade)</td></tr>
<tr><td>研究倫理教育
Research Ethics</td><td colspan="5">※該当するものをチェックしてください。/Please tick either of following options.

☑ 本計画書を提出時点で、所定の研究倫理教育を受講済み（本学設置科目の場合、単位修得済み）であり、受講した研究倫理教育について MyWaseda の申請フォームから報告済みである。
At the time of submitting this research plan, I have completed the required research ethics course (or earned credits for courses offered by the University). Besides, I have reported on the research ethics course through MyWaseda application form.

☐ 本計画書を提出時点で、所定の研究倫理教育を受講していない。
At the time of submitting this research plan, I have not completed the required research ethics course (or earned credits for courses offered by the University).</td></tr>
</table>

１．研　究　目　的　Purpose of Research

The advent of quantum computing demands a rapid transition to post-quantum cryptographic solutions. In digital identity, SNARK-friendly schemes like Loquat[1] underpin post-quantum anonymous credential systems such as BDEC[2]. However, BDEC's reliance on static, custom zkSNARK[3] circuits for credential verification leads to critical inflexibility, rendering it impractical for dynamic attribute management. Zero-Knowledge Virtual Machines[4] (zkVMs) promise a solution, offering to prove arbitrary programs and transform complex circuits into high-level code updates. This research will investigate the specific zero-knowledge properties of different zkVMs through comparative analysis of zkVMs and alternative SNARK circuit compilers[5], implementing and benchmarking the BDEC verifier within both approaches. This quantitative and qualitative analysis will determine which approach offers a more viable and agile foundation for the next generation of digital identity systems, specifically addressing the trade-offs between flexibility, performance with concrete metrics such as prover time, verification time, and memory usage.

２．従 来 の 研 究　Existing Research

- Loquat is built on symmetric-key primitives and the quadratic residuosity problem, minimizing the number of constraints required when translated into an arithmetic circuit.
- BDEC builts directly upon Loquat, integrating it with Merkle trees to achieve unlinkable and selectively disclosable credentials. In BDEC, a user's attributes are committed to in the leaves of a Merkle tree, and the issuer provides a Loquat signature on the tree's root. While this construction is secure and effective, its implementation relies on a fixed-size zkSNARK circuit that is tailored to a predefined number of attributes, which is the primary limitation we aim to address.
- Zero-Knowledge Virtual Machines (zkVMs), such as Jolt[6] and Risc Zero[7], represent a shift from hand-crafted circuits to general-purpose computation proofs. While they simplify development and broaden applicability, their performance on specialized cryptographic tasks remains underexplored. zkVMs vary in privacy guarantees depending on the underlying proof system. zk-SNARKs offer compact proofs but often require a trusted setup. In parallel, direct SNARK compilers like zkLLVM[8], and Circom[9] are maturing, offering an alternative path that forgoes the VM layer in favor of optimized circuits. This research will compare these approaches to assess trade-offs in efficiency, and auditability.

[1] Xinyu Zhang et al., "Loquat: A SNARK-Friendly Post-Quantum Signature based on the Legendre PRF," Cryptology ePrint Archive, Report 2024/868
[2] Zoey Z. Li et al., "BDEC: Enhancing Learning Credibility via Post-Quantum Digital Credentials," in Provable and Practical Security (ProvSec 2024), Springer LNCS, 2025
[3] Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Zerocash: Decentralized Anonymous Payments from Bitcoin. 2014 IEEE Symposium on Security and Privacy, 459-474.

[4] Tim Dokchitser, & Alexandr Bulkin. (2023). Zero Knowledge Virtual Machine step by step.
[5] Ryan Lavin, Xuekai Liu, Hardhik Mohanty, Logan Norman, Giovanni Zaarour, & Bhaskar Krishnamachari. (2024). A Survey on the Applications of Zero-Knowledge Proofs.
[6] Arasu Arun, Srinath Setty, & Justin Thaler. (2023). Jolt: SNARKs for Virtual Machines via Lookups.
[7] Bruestle, J., Gafni, P., & RISC Zero Team. (2023, August 11). RISC Zero zkVM: Scalable, Transparent Arguments of RISC-V Integrity. RISC Zero. https://dev.risczero.com/proof-system-in-detail.pdf
[8] Garrido, G., Riu, A., & Pardo, D. (2023). zkLLVM: A Zero-Knowledge Proof-Friendly LLVM-Based Compiler. 2023 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), 1-2.
[9] Bellés-Muñoz, M., Isabel, M., Munoz-Tapia, J., Rubio, A., & Baylina, J. (2023). Circom: A Circuit Description Language for Building Zero-Knowledge Applications. IEEE Trans. Dependable Secur. Comput., 20(6), 4733–4751.

３．研　究　計　画　Research Plan

2025				2026						
9月	10月	11月	12月	1月	2月	3月	4月	5月	6月	7月
① Initial zkVM Implementation										
			② BDEC Verifier Implementation							
							③ Benchmark			
								④ Finalise Thesis		

① **September to November 2025: Initial zkVM Implementation**
- Conduct a deep dive into the architectures of leading Zero-Knowledge Virtual Machines (zkVMs) such as Jolt, RISC Zero, based on existing research.
- Thoroughly deconstruct the cryptographic primitives in Loquat and BDEC.
- Identify and analyze viable direct SNARK circuit compilers (e.g., zkLLVM, Circom), understanding their core compilation approaches and claimed advantages.
- Implement a simple, non-cryptographic program within a chosen zkVM to practice the end-to-end workflow of writing, compiling, proving, and verifying code.

② **December 2025 to March 2026: BDEC Verifier Implementation**
- Develop the complete verification logic for BDEC as a Rust program, including a flexible Merkle path verifier that can handle variable numbers of attributes and a complete implementation of the Loquat signature verification algorithm.
- Compile the BDEC verifier program for a chosen zkVM's instruction set, debugging and optimizing the Rust code to be compatible and efficient within the zkVM's constraints. Concurrently, adapt the BDEC verification logic to be compiled into a SNARK circuit using one of the identified alternative SNARK circuit compilers.

③ **April to May 2026: Benchmark Testing**
- Run the compiled the BDEC verifier program in both zkVM and SNARK circuit environments.
- Conducting a comparative analysis of the theoretical zero-knowledge properties and performance implications, such as circuit size, prover time, verification time, for both zkVMs and direct compilers, specifically addressing the professor's concerns about zkVMs' "true zero-knowledgeness" and potential speed limitations.

④ **May to July 2026: Finalise Thesis**
- Finalise the analysis of all benchmark data and draw concrete conclusions.
- Perform drafting, reviewing feedback from Professor Sako to refine the entire thesis document.