Fully Homomorphic Encryption Back-End for CirC
Assignment 5: Project Proposal Milestone Report
07-300, Fall 2021

William Seo
`https://northrim.github.io/07400-s22`

December 2021

1 Specific Items

My name and project web page URL are displayed above. I will be working with Professor Wenting Zheng and PhD student Edward Chen from the Computer Science department.

2 Progress Report

So far, I have read two research papers which are relevant to my project. The first paper, titled "SoK: Fully Homomorphic Encryption Compilers" is a paper that surveys and evaluates currently existing FHE tools, libraries, and compilers [3]. The most important section of this paper for me was section VI, which provided an in-depth overview of existing FHE compilers, such as Alchemy, Cingulata, and EVA. The paper provides evaluations of the features, optimizations, and accessibility of these different compilers. In the paper's final section, the authors make several suggestions on how the development of FHE tools should proceed from here. These suggestions include the construction of a "common intermediate representation language", and the building of more general tools that can support Multi-Party Computation and Zero-Knowledge Proofs in addition to FHE [3]. This is noteworthy because the project that I am working on is a realization of these two suggestions.

The second paper, titled "EVA: An Encrypted Vector Arithmetic Language and Compiler for Efficient Homomorphic Computation" [1], introduces the FHE language/compiler EVA, which was arguably the most capable compiler of the ones evaluated in the SoK survey [3]. The most important sections of this paper for me were section 4, which provides an overview of the EVA compiler, and section 5, which describe transformations (optimizations) in the compiler. Reading these sections provided me with a stronger understanding of how FHE compilers are structured, how they function, and how they can be optimized. EVA will be a valuable reference for me while I develop my FHE backend for CirC.

3 Reflection on Initial Plan

3.1 Major changes:

There have been no major changes.

3.2 Meeting your milestone:

I have met the initial milestone of getting started on the background reading by the end of the semester. On a side note, my proposal also has a milestone to be completed by the end of winter break, which is to familiarize myself with Rust and the CirC code base, and also complete some starter compiler tasks. I have made progress towards these winter break milestones by beginning to read the Rust Handbook [2], and attempting to set up the CirC codebase on my local computer.

3.3 Surprises:

I am running into errors when trying to build the CirC codebase on my local machine. My mentor and I are currently looking into ways to fix this.

3.4 Revisions to your 07-400 milestones:

One revision we may have to make is to allocate more time for the February 1st milestone, which is integrating an FHE backend library (SEAL) into CirC.

3.5 Resources needed:

For this section, nothing has changed from my original proposal.

References

[1] Roshan Dathathri, Blagovesta Kostova, Olli Saarikivi, Wei Dai, Kim Laine, and Madanlal Musuvathi. EVA: an encrypted vector arithmetic language and compiler for efficient homomorphic computation. *CoRR*, abs/1912.11951, 2019.

[2] S. Klabnik and C. Nichols. *The Rust Programming Language (Covers Rust 2018)*. No Starch Press, 2019.

[3] Alexander Viand, Patrick Jattke, and Anwar Hithnawi. Sok: Fully homomorphic encryption compilers. *CoRR*, abs/2101.07078, 2021.