Associate Professorship of Embedded Systems and Internet of Things
Department of Electrical and Computer Engineering
Technical University of Munich

# Gnark Implementation and Optimization of the TLS-CHACHA20-POLY1305-SHA256 TLS Cipher Suite

## Master Thesis

Jan Lauinger, February 17, 2023

| | |
|---|---|
| Title: | “Gnark Implementation and Optimization of the TLS-CHACHA20-POLY1305-SHA256 TLS Cipher Suite” |
| Supervisor: | Jan Lauinger |
| Period: | 24 weeks |

## Context

Zero-knowledge proofs continue to impact and reshape existing Internet protocols with their inherent features of confidentiality, compression, and credibility. With the rise of zk-SNARK frameworks, which allow developers to build practical zero-knowledge proofs of arbitrary statements, implementation and deployment of new types of Internet protocols becomes possible. When targeting to add zero-knowledge features to current Internet protocols, the excitement however may stop. This is due to the fact, that many standardized cryptographic algorithms are zk-SNARK non-friendly. Cryptographic algorithms, which are not SNARK friendly, may depend on optimized adjustments or quirks, which must be translated to the more constrained zk-SNARK circuit representation of computation. Additionally, not all computation steps of a cryptographic algorithm are of concern in a zk-SNARK circuit and can be bypassed if considered as public input.
In this thesis, it is your task to translate the TLS 1.3 cipher suite `TLS-CHACHA20-POLY1305-SHA256` into a zk-SNARK circuit and implement the circuit using gnark. In order to do so, you access the `CHACHA20-POLY1305` implementation found in the *crypto/tls* Golang standard library and use it to generate input/output mappings as well as intermediate values to generate test data. Next, you design the zk-SNARK circuit representation of `CHACHA20-POLY1305` and implement it in gnark. We have already implemented the first existing `SHA256` implementation in gnark and your task will be to further optimize the bit/byte representation of frontend variables in our `SHA256` implementation. Last, your task is to benchmark your implementation and optimizations and compare it with other existing implementations. Once you have generated all evaluation benchmarks, the last concern is to provide documentation of your repository and compile your thesis submission document.

## Requirements

- High: Coding with Golang and work with Go modules.
- High: Independent work ethics, conceptual thinking, and willingness to dive deep into implementation of cryptographic algorithms.
- Medium: Interest in zero-knowledge proof systems and cryptographic algorithms.

## Tasks

The thesis is separated into several working packages.

1. Familiarization with cryptographic algorithms of interest (generate input/output mappings and access to intermediate values) & the gnark framework
2. Design of `CHACHA20-POLY1305` zk-SNARK circuit representation & implementation its in gnark
3. Optimization and benchmarking of zk-SNARK circuit implementations & thesis writing

## Preliminary Schedule