High-Assurance Cryptographic Software for Post-Quantum and Distributed Secure Computation

Abstract. Computer Aided Cryptography (CAC) aims to develop tools for the analysis and implementation of cryptographic protocols, including specification and implementation languages, compilers and formal verification tools. The formal verification dimension opens the way for High-Assurance Cryptographic Software. The objective of this work is to progress beyond the state of the art at both the foundational and applied research levels by developing new extensions to the Jasmin tool-chain and demonstrating its applicability to concrete examples that can have real-world impact. A recent SoK paper [2] published at IEEE S&P 2021 summarises challenges in this area; two of these challenges serve as our main motivation: 1) removing scalability bottlenecks in the analysis of distributed cryptographic protocols and 2) enabling the transition to post-quantum cryptography. This proposal will focus on concrete cryptographic constructions and implementations of post-quantum cryptography (PQC) and secure distributed computation that can serve as inputs to standardisation processes or as contributions to widely deployed cryptographic libraries and protocol software stacks.

Keywords: High-Assurance Cryptographic Software · Program Verification · Post-Quantum Cryptography · Secure Distributed Computation.

State of the art

Jasmin [1] is a programming language designed to allow assembly in the head (a mixture of high-level and low-level, platform-specific, programming); it is supported by a formally verified (certified in Coq), predictable, compiler which empowers programmers to write highly efficient fine-tuned code. The generated (verified) assembly code matches the performance of the best implementations for this primitive. Jasmin code can be proved correct and secure via an equivalence proof to a high-level specification in EasyCrypt: the Jasmin compiler is able to create an EasyCrypt translation of its source. End-to-end security and correctness follow from the certification of the Jasmin compiler (source-to-assembly) and from the EasyCrypt proof (source-to-spec).

Our experience using EasyCrypt and Jasmin [2] shows that both foundational and applied research are needed to tackle two classes of cryptographic protocols that are within our reach: 1) post-quantum cryptography (PQC) and 2) secure distributed (multiparty) computation.

PQC should not be confused with quantum computing or quantum cryptography; its goal is to create cryptographic schemes that can be used today and resist potential quantum attacks in the future. PQC relies on different maths abstractions and computational assumptions than classical crypto such as lattice-based [6] and isogeny-based assumptions [7]. Implementing such primitives raises new challenges for the Jasmin framework; e.g., we have not yet considered rejection sampling mechanisms, which are crucial in PQC.

Orthogonal challenges arise in the verification of interactive protocols in Jasmin and EasyCrypt. Each cryptographic primitive is proved secure in a security model that captures its use in the real world. Primitives such as key exchange have been proved secure

in a variety of models [3], with surprising complexity for two-party protocols: the goal
is to capture concurrent executions and deal with composition within the security proof
itself. General approaches to composability [5] exist, but these exclude some of the more
efficient instantiations, or require ad-hoc adaptations to capture the associated caveats.

HACSpec [4] is a common specification language for cryptography that can be used by
technical standards, software developers, and feed formal verification tools. Syntactically,
HACSpec is a subset of Rust, and hence is familiar to developers; most importantly for
standards, specifications are executable, they can be tested for correctness and interoper-
ability, and to generate test vectors. HACSpec comes with a translator tool that can feed
various formal verification frameworks. This proposal will contribute to the HACSpec de-
velopment and ensure that the extensions we make to Jasmin can have a greater impact
via integration into the HACSpec tool-chain.

1   *Objectives

The goal of this proposal is to progress beyond the SotA at both the foundational
and applied research levels by 1) developing new extensions to the HACSpec, and Jasmin
tool- chain, and 2) demonstrating its applicability to concrete examples that can have
real-world impact.

We will focus on concrete cryptographic constructions and implementations of PQC
and secure distributed computation that can serve as inputs to standardisation or as
contributions to widely deployed cryptographic libraries and software stacks. These use-
cases are just outside the reach of existing CAC tool-chains, as described in the SotA
analysis.

In what follows, we explain the approach that we will adopt to change this state of
affairs by working in the concrete setting of the HACSpec-Jasmin tool-chain to reduce
scalability bottlenecks in the analysis of distributed cryptographic protocols and enable
the transition to post-quantum cryptography.

With this in mind, we will create a repository of Jasmin artifacts allowing community
uploads, from small libraries to large developments. This need arises from the fact that
these artificats will require maintenance to ensure they are up to date with the current
tool versions. Furthermore all implementations will be backward compatible and, when
needed, we will update and expand the existing results so that they can be applied in
other distributed protocols.

2   *Descrição de trabalhos

This proposal follows from a previous work in the student's Master thesis where the
protocol MPC-in-the-head was studied as part of an algorithm called PICNIC. During
the first year of PHD, the student will explore the MPC-in-the-head protocol in depth
by implemeting it using the Jasmin language and verify its correct implementation using
EasyCrypt, thus improving its knowledge about both tools and the protocol. Furthermore,
the student will attend to the Summer School on real-world crypto and privacy in order
to understand and explore new protocols and tools to work them.

O trabalho no ZKBoo em Jasmim serve como porta pra estudar o MPC refirir o
mestrado na medida em que contactei com o MPC -checked

Falar do EasyCrypt como forma de formalizar provas de segurança. -> garantir que
as implementações estão corretamente implementadas

Referência ao MPC-in-the-head (T4)

1ž aprofundar temas e participar em eventos

O que vamos fazer neste primeiro ano - explorar o protocolo PQC (MPC in the head) Continuação do trhlo de mestrado - dominar as ferramentS DE JASMIM E EASYCR-RYPT BEM COM O HACSPEC. Ir á escola de verão da área de criptografia. -

References

[1] J. Almeida et al. "Jasmin: High-Assurance and High-Speed Cryptography." In: *ACM CCS* (2017).

[2] M. Barbosa et al. "SoK: Computer-Aided Cryptography." In: (2021).

[3] M. Bellare, D. Pointcheval, and P. Rogaway. "Authenticated Key Exchange Secure against Dictionary Attacks." In: *IACR EUROCRYPT* (2000).

[4] K. Bhargavan, F. Kiefer, and P.-Y. Strub. "HACSpec: Towards Verifiable Crypto Standards." In: *SSR* (2018).

[5] R. Canetti. "Universally Composable Security." In: *J. ACM* (2020).

[6] *CRYSTALS: Cryptographic Suite for Algebraic Lattices.* 2021.

[7] *SIKE: Supersingular Isogeny Key Encapsulation.* 2021.