PSI VESELY

psi@ucsd.edu

EDUCATION

University of California, San Diego *'20—Present*
PhD in Cryptography *San Diego, CA*

University College London *'18—'19*
MSc in Information Security *with distiction* *London, UK*
On the Design of Polynomial Commitment Schemes with Dr. Mary Maller
We introduce two new polynomial commitments (PCs) and a black box transformation for evaluating multiple PCs at multiple points given a PC supporting evaluation at a single point. One PC achieves full succinctness and enforceable degree bounds via an universal and updatable SRS; the other is transparent with constant commitments, log-sized proofs, and square root URS and prover opening time.

Hampshire College *'10—'15*
Bachelor of Mathematics, minor in Computer Science *Amherst, MA*
On McEliece-Type Cryptosystems as Post-Quantum Standards with Prof. Polanco Encarnación
An analysis of the suitability of McEliece-Type cryptosystems as post-quantum standards, with a focus on efforts to reduce public key size by replacing the binary Goppa code.
Evolving a Cryptographic Compression Function with Prof. Lee Spector
We introduce a new cryptographic compression function evolved using the PushGP genetic programming environment and a novel fitness heuristic.

PUBLICATIONS

Proofs for Inner Pairing Products and Applications *'21*
ASIACRYPT '21 `https://ia.cr/2019/1177`
We generalize previous works on inner product arguments (IPAs) to any bilinear map and introduce new IPAs for pairing-based languages–the first with log-time verifiers. Combining several new IPAs, we construct new polynomial commitments improving on the prover time and CRS size of KZG, provide the first concretely efficient protocol for aggregating Groth16 proofs without recursion, and construct a low-memory SNARK for machine computations with a significantly faster prover over prior state-of-the-art [Bitansky et al., STOC '13].

Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS *'20*
EUROCRYPT '20 `https://ia.cr/2019/1047`
We present a methodology to construct preprocessing zkSNARKs with universal and updatable SRS. Our construction improves on Sonic [Maller et al., CCS 2019], the prior state of the art in this setting, in all efficiency parameters: proving is an order of magnitude faster and verification is thrice as fast, even with smaller SRS size and argument size.

PREPRINTS

Plumo: An Ultralight Blockchain Client *'21*
In submission, Financial Cryptography '21 `https://ia.cr/2021/1361`
We introduce a consensus-agnostic methodology for constructing ultralight clients, providing highly efficient blockchain syncing via SNARK-based state transition proofs. We also present two new SNARK-friendly constructions: a BLS-based offline aggregate multisignature scheme in which signers do not have to know their multisignature group in advance, and a composite algebraic-symmetric cryptographic hash function.

WORK EXPERIENCE

Aleo *Oct '21—Present*
Scientist *San Francisco, CA*

cLabs *June '19—Oct '21*
Scientist *San Francisco, CA*
Designed an ultralight client for the Celo blockchain using SNARKS and circuit-friendly primitives including a novel aggregeate multisignature composite hash function. Contributed to designs for privacy-preserving contact discovery and private transaction comments.

University of California, Berkeley *Oct. '19—Oct. '20*
Research Assistant *Berkeley, CA*
With professor Alessandro Chiesa, researching topics in zero-knowledge proofs.

Information Security Services *Sept. '17—May '18*
Information security and cryptography related development, consulting, and training. Worked with non-profits including Data Cívica and Human Rights Data Analysis Group.

Freedom of the Press Foundation *Sept. '15—Aug. '17*
Security Engineer *San Francisco, CA*
`https://github.com/freedomofpress/securedrop`
Design and development of the SecureDrop open-source whistleblower submission platform. Stringent security requirements and a multi-machine, multi-OS architecture demanded wide-breadth domain knowledge including cryptographic, network, OS, and application-level security expertise.
`https://github.com/freedomofpress/fingerprint-securedrop`
Implemented a machine learning system to evaluate website fingerprinting attacks and defenses for Tor onion services. Led Tor developer conference sessions on the topic and worked closely with academic researchers.

TALKS

ASIACRYPT *Dec. '21*
Proofs for Inner Pairing Products and Applications

zkSummit 5 *May '20*
Inner Pairing Product Arguments and Applications

Scaling Bitcoin at Tel Aviv University *Sept. '19*
The Celo Ultralight Client *Tel Aviv, ISR*

COSIC Group at KU Leuven *Mar. '17*
Fingerprinting SecureDrop *Leuven, BE*

TEACHING EXPERIENCE

Introduction to Cryptography *Fall '21*
Teaching assistant *UCSD*
Co-taught discussion section, held office hours, and created and graded assignments with Prof. Nadia Heninger.

Hampshire College Quantitative Resource Center *Sept. '12—May '15*
Manager *Amherst, MA*
Tutored mathematics and computer science. Management duties included hiring, budgeting, and scheduling.

Linear Algebra *Fall '14*
Teaching Assistant *Hampshire College*
Guest taught a lecture, held weekly office hours, and marked coursework with Prof. Sarah Hews.

ACADEMIC SERVICE

Financial Cryptography *'20*
Subreviewer

SOFTWARE PROJECTS

ripp *'20*
Co-Author `https://github.com/arkworks-rs/ripp`
An implementation of several arguments from the "Proofs for Inner Pairing Products and Applications" paper.

marlin *'19*
Co-Author `https://github.com/arkworks-rs/marlin`
An implementation of the Marlin zkSNARK.

winternitz *'18*
Author `https://github.com/nvesely/winternitz`
The first standalone implementation of the post-quantum WOTS-T one-time signature scheme.

Rusty Secrets *'18*
Co-Author `https://github.com/SpinResearch/RustySecrets`
A Rust implementation of Shamir's Secret Sharing Scheme that provides authentication of shares. Used in the Sunder application.

SodiumOxide *'18*
Maintainer & Contributor `https://github.com/sodiumoxide/sodiumoxide`
A Rust interface to the C++ libsodium cryptography library that seeks to utilize Rust's featureset to improve on the usability and safety of the library.

libalpaca *'17*
Co-Author `https://github.com/camelids/libalpaca`
A library that implements ALPaCa, an application-layer defense against website fingerprinting.