Cryptographic Primitives for Zero-Knowledge: Theory and Implementation

CANDIDATE
Stefano Trevisani

SUPERVISOR
Dr. Arnab Roy

CO-SUPERVISORS
Prof. Alberto Policriti
Prof. Elisabeth Oswald

TUTOR
M.Sc. Matthias Steiner

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

UNIVERSITÀ DEGLI STUDI DI UDINE
hic sunt futura

Verifiable Computation

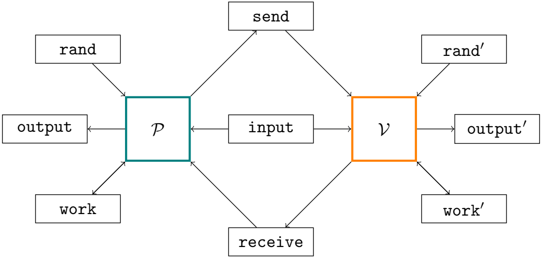
Verifiable computation:

- Server: computes some function.
- Client(s): verify the correctness of the results.

Many applications:

- Delegating heavy loads to the cloud [ACK+02].
- Calculating household due bills [PGHR13].
- **Verifying transactions on the blockchain.** [BSCG+14]

Interactive Proof Systems [GMR89]



A pair of **interactive probabilistic Turing machines**:

- Prover: wants to prove a statement by creating a proof.
- Verifier: wants to check the soundness of the proof.
- Verifier is PTIME, might be fooled with negligible probability.
- IP = PSPACE [Sha92].

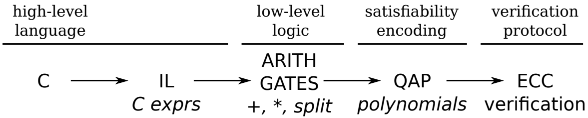
ZK-SNARK systems

ZK-SNARK systems:

- Verifier might be curious $\implies$ **Zero-Knowledge.**
- Verification must be fast $\implies$ **Succinct.**
- There may be many verifiers $\implies$ **Non-interactive.**
- Prover is polynomially bounded $\implies$ **Argument of Knowledge.**

Complexity dominated by proof generation time.

SNARKs via QAPs [GGPR12, PGHR13, Gro16]

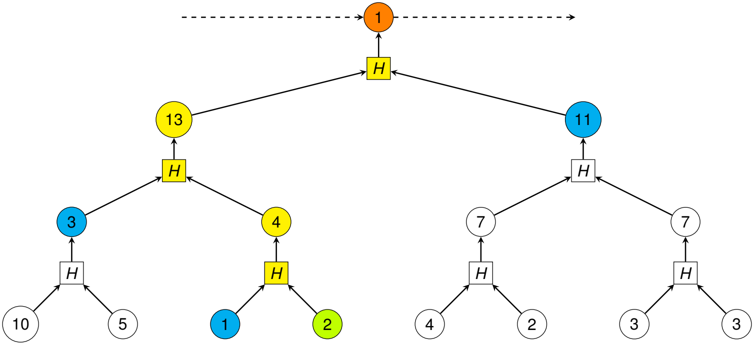


Bounded computations via *arithmetic circuits* over $\mathbb{F}_p$:

1. *Rank-1 constraint systems (**R1CS**)* encode circuit invariants.
2. *Quadratic Arithmetic Programs (**QAP**)* “compress” R1CSs.
3. Prover/Verifier keys to build/check the proof: ensure integrity.
4. Exploit bilinear maps, work in the exponent: discrete $\log$ is hard!
5. “Inject” randomness in the polynomials to get zero-knowledge.

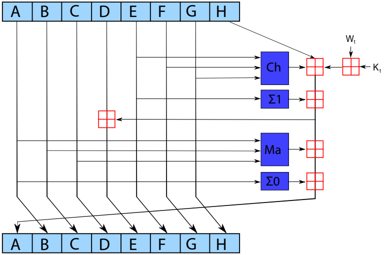
Complexity depends on the number of R1CS constraints.

The Blockchain



- Groups of transactions are leaves of a **Merkle tree** [Mer88].
- Bottom-up computation using an **hash function**.
- The root contains a binding commitment.
- Check commitment via the short authentication path.

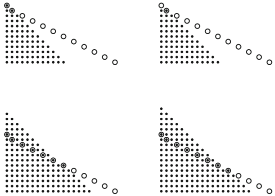
SHA [Dan15]



Standard designs, like SHA, are designed over *boolean fields*:

- Bitwise AND, XOR, rotation, modulo 2^k addition...
- Extremely efficient hardware and software implementations.
- However, ZK-SNARKs work over prime fields $\Longrightarrow$ emulation.
- SHA-256 $\approx$ **25000 R1CS constraints**.

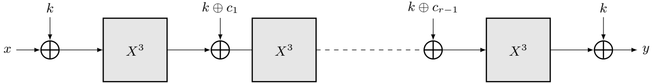
Arithmetization Oriented Primitives



Arithmetization-Oriented (AO) cryptographic primitives over $\mathbb{F}_p$:

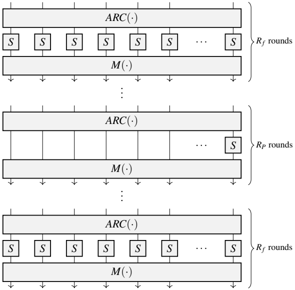
- Use only field **sum** and **multiplication**.
- Can be modeled as polynomials.
- Security depends on the feasibility of **algebraic attacks**.

MiMC [AGR+16]



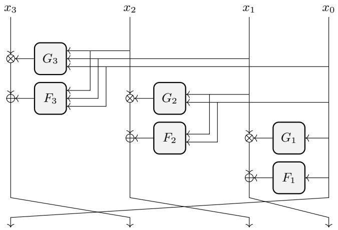
- MiMC: Minimal Multiplicative Complexity.
- Extremely simple: *round function* is $x^3 + c$.
- Exponent is the lowest integer in $\mathbb{F}_p$ coprime with $p - 1$.
- $\lceil \frac{\log(p)}{\log(3)} \rceil$ rounds to achieve **degree overflow**.
- MiMC-256: **640 R1CS constraints** (40$\times$ less than SHA-256).

POSEIDON [GKR+21]



- POSEIDON: *substitution-permutation network* (from AES [DR99]).
- Full rounds defend against classic attacks.
- **Partial rounds** defend against algebraic attacks.
- POSEIDON-256: **240 R1CS constraints** (2.5$\times$ less than MiMC).

GRIFFIN [GHR+22]



- GRIFFIN: based on the `Horst` scheme: $(x, y) \mapsto (y, x \otimes G(y))$.
- Circulant MDS matrix in the linear layer.
- Inverse power to achieve faster degree growth [AABS+19].
- GRIFFIN-256: **96 R1CS constraints** (2.5× less than POSEIDON).

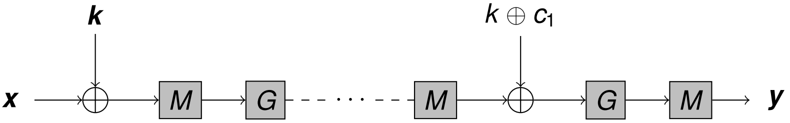
The GTDS [RS22]

The new *Generalized Triangular Dynamical System* (GTDS):

$$
\begin{aligned}
x_i \quad &\longmapsto \quad y_i = x_i^{d_1} g_i\left(\sum_{j=i+1}^{n} x_j + y_j\right) + h_i\left(\sum_{j=i+1}^{n} x_j + y_j\right) \\
x_n \quad &\longmapsto \quad y_n = x_n^{1/d_2} \\
g_i(x) &= x^2 + \alpha_i x + \beta_i \\
h_i(x) &= x^2 + \gamma_i x
\end{aligned}
$$

- Algebraic framework to design secure permutations.
- Encompasses existing strategies (Feistel, `Horst`, SPN...).
- Extends previous designs, allows more flexibility.
- **High degree permutation** representable using a **small R1CS**!

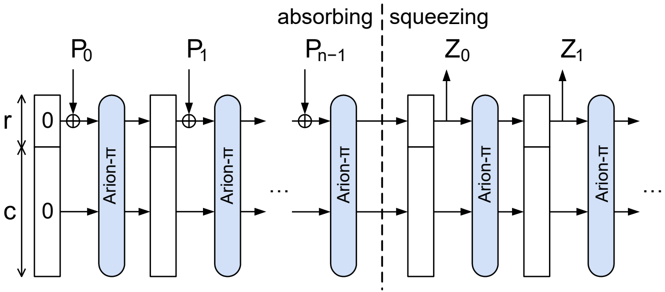
Arion and Arion-π **[RST23]**



Arion, a new keyed permutation from the GTDS:

- Exponent d_2: inverse is large, few constraints for exponentiation.
- Affine layer: matrix with linear matrix-vector product complexity.
- Achieves degree overflow in just one round.
- Arion-π: one-way permutation obtained from fixed-key Arion.

ArionHash and α-ArionHash



The ArionHash hash function:

- Derived from Arion-π in **sponge mode** [BDPVA07].
- ArionHash-256: only **76 R1CS constraints**.
- 330$\times$ less than SHA-256!
- 3.2$\times$ less than POSEIDON, 25% less than GRIFFIN.

Experiments

Proof generation times for Merkle commitments over the BN-254 [BN05] field.

Tree height	α-**ArionHash**	GRIFFIN	POSEIDON	MiMC
4	**73** ms	88 ms	186 ms	350 ms
8	**145** ms	181 ms	386 ms	735 ms
16	**278** ms	338 ms	745 ms	1460 ms
32	**509** ms	622 ms	1422 ms	2930 ms

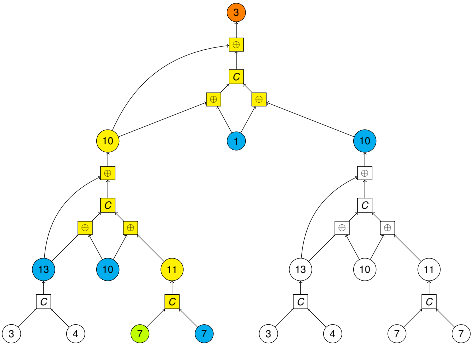
`libsnark`: standard C++ library for ZK-SNARK (ZCash [BSCG$^+$14]).
I used it to write a library containing:

- Arion and other arithmetization-oriented cryptographic primitives.
- A self-configuring Merkle tree, and the ABR mode [ABR21].
- A unified interface to write, test and benchmark new designs.
- Will be open-sourced once the article is published.

$\mathcal{T}$he $\mathcal{E}$nd

Thank you for your attention!

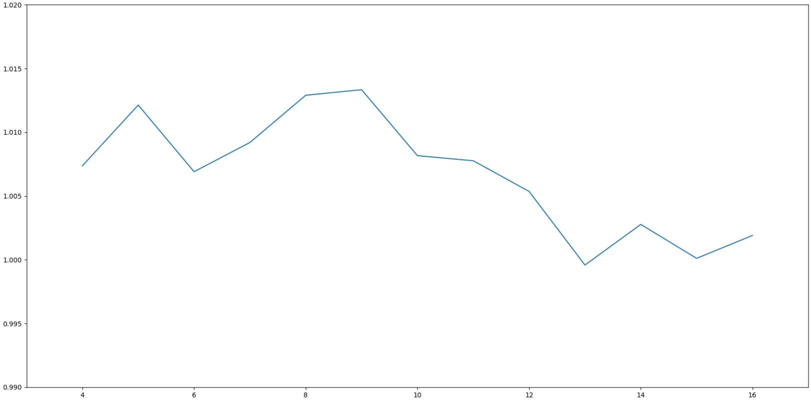
The Augmented Binary tRee [ABR21]



- *Compactness* $\approx$ blocks compressed per function call.
- Merkle Tree compactness is $2/3$.
- ABR interleaves OWCF calls with field addition.
- ABR processes 50% more messages: compactness is 1.
- Additions are basically free in ZK-SNARK!

Experimental results

ABR vs. Merkle Tree relative performance for increasing height:



Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec.
Design of symmetric-key primitives for advanced cryptographic protocols.
Cryptology ePrint Archive, Paper 2019/426, 2019.
https://eprint.iacr.org/2019/426.

Elena Andreeva, Rishiraj Bhattacharyya, and Arnab Roy.
Compactness of hashing modes and efficiency beyond merkle tree.
Cryptology ePrint Archive, Paper 2021/573, 2021.
https://eprint.iacr.org/2021/573.

David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer.
Seti@home: An experiment in public-resource computing.
Commun. ACM, 45(11):56–61, nov 2002.

Martin Albrecht, Lorenzo Grassi, Christian Rechberger, Arnab Roy, and Tyge Tiessen.
Mimc: Efficient encryption and cryptographic hashing with minimal multiplicative complexity.
In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 191–219, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Sponge functions.
In *ECRYPT hash workshop*, volume 2007, 2007.

Paulo S. L. M. Barreto and Michael Naehrig.
Pairing-friendly elliptic curves of prime order.
Cryptology ePrint Archive, Paper 2005/133, 2005.
https://eprint.iacr.org/2005/133.

Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza.
Zerocash: Decentralized anonymous payments from bitcoin.
In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, 2014.

Quynh H. Dang.
Secure Hash Standard.
National Institute of Standards and Technology, Jul 2015.

Joan Daemen and Vincent Rijmen.
Aes proposal: Rijndael, 1999.

Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova.
Quadratic span programs and succinct nizks without pcps.
Cryptology ePrint Archive, Paper 2012/215, 2012.
https://eprint.iacr.org/2012/215.

Lorenzo Grassi, Yongling Hao, Christian Rechberger, Markus Schofnegger, Roman Walch, and Qingju Wang.
A new feistel approach meets fluid-spn: Griffin for zero-knowledge applications.
IACR Cryptol. ePrint Arch., 2022:403, 2022.

Lorenzo Grassi, Dmitry Khovratovich, Christian Rechberger, Arnab Roy, and Markus Schofnegger.
Poseidon: A new hash function for zero-knowledge proof systems.
In *USENIX Security Symposium*, 2021.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff.
The knowledge complexity of interactive proof systems.
SIAM Journal on Computing, 18(1):186–208, 1989.

Jens Groth.
On the size of pairing-based non-interactive arguments.
Cryptology ePrint Archive, Paper 2016/260, 2016.
https://eprint.iacr.org/2016/260.

Ralph C. Merkle.
A digital signature based on a conventional encryption function.
In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pages 369–378, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova.
Pinocchio: Nearly practical verifiable computation.
Cryptology ePrint Archive, Paper 2013/279, 2013.
https://eprint.iacr.org/2013/279.

Arnab Roy and Matthias Steiner.
Generalized triangular dynamical system: An algebraic system for constructing cryptographic permutations over finite fields, 2022.
https://arxiv.org/abs/2204.01802.

Arnab Roy, Matthias Steiner, and Stefano Trevisani.
Arion: Arithmetization-oriented permutation and hashing from generalized triangular dynamical systems.
Undergoing submission, 2023.

Adi Shamir.
Ip = pspace.
J. ACM, 39(4):869–877, oct 1992.