Cryptographic Primitives for Zero-Knowledge: Theory and Implementation

CANDIDATE

Stefano Trevisani

SUPERVISOR

Dr. Arnab Roy

CO-SUPERVISORS

Prof. Alberto Policriti
Prof. Elisabeth Oswald

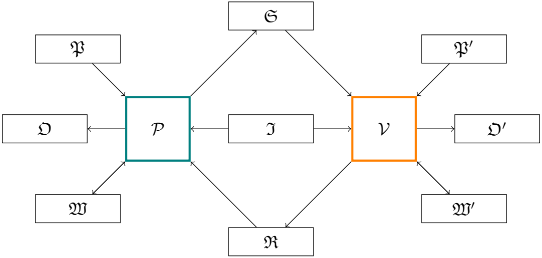
TUTOR

M.Sc. Matthias Steiner

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

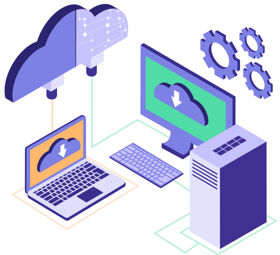
UNIVERSITÀ DEGLI STUDI DI UDINE
hic sunt futura

Interactive Proof Systems [GMR89]



- Prover: wants to prove a statement by creating a proof.
- Verifier: wants to check the soundness of the proof.
- Modeled as *interactive I/O probabilistic Turing machines*.
- Verifier is polynomially bounded.
- Verifier might be fooled with *negligible* probability.
- IP = PSPACE [Sha92].

Verifiable Computation



Proof systems can be used for *verifiable computation*:

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

ZK-SNARK systems

Completeness

Soundness

Zero-Knowledge

ZK-SNARK systems:

- Prover might be **dishonest** $\implies$ proof system.
- 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*.

SNARKs via QAPs [GGPR12, PGHR13, Gro16]

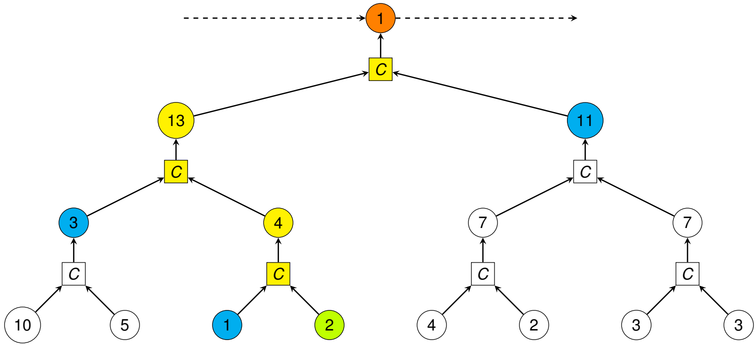
high-level language		low-level logic	satisfiability encoding	verification protocol
C →	IL →	ARITH GATES →	QAP →	ECC
	C exprs	+, *, *split*	*polynomials*	verification

Setting up a SNARK for some computable function:

1. Bounded computations represented through *arithmetic circuits*.
2. *Rank-1 constraint systems (R1CSs)* encode circuit invariants.
3. *Quadratic Arithmetic Programs (QAPs)* “compress” R1CSs.
4. *Private key* to build the proof, *public key* to verify it.
5. Exploit *bilinear maps*, work in the exponent: discrete $\log$ is hard!
6. Inject random noise for statistical zero-knowledge.

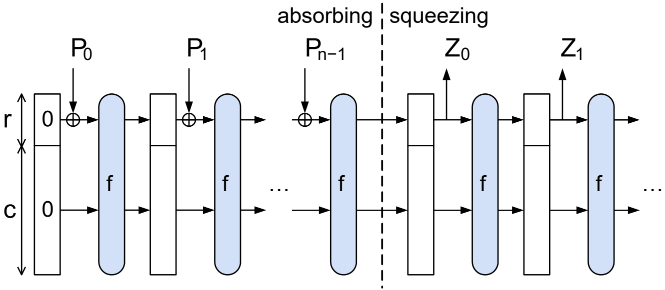
Generating the keys incurs into the *toxic waste* problem. . .

The Blockchain



- Groups of transactions are leaves of a *Merkle tree* [Mer88].
- Bottom-up computation using a **compression function**.
- The root contains the *commitment* (among other data).
- Verify a commitment following the *authentication path*.

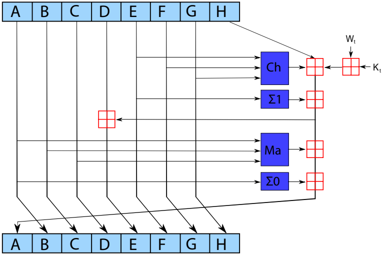
Cryptographic Compression Functions



One-way compression functions (OWCF):

- Many inputs reduced to a few outputs (e.g. 2-to-1).
- Easy to compute, but hard to invert and find collisions.
- Usually derived from one-way permutations.
- Constructed through secure schemes, like Sponge [BDPVA07].

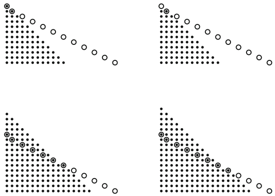
SHA [Dan15]



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 $\implies$ emulation.
- SHA-256 $\approx$ 25000 constraints.
- Can we do better?

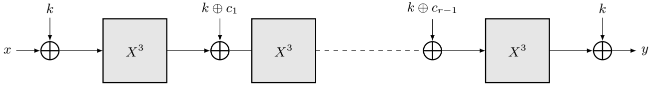
Arithmetization Oriented Primitives



Arithmetization-Oriented (AO) cryptographic primitives:

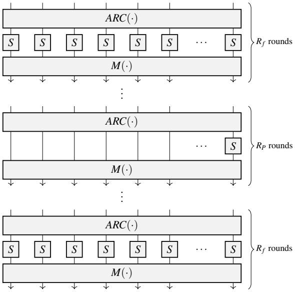
- Build keyed permutations using prime field sum and multiplication.
- Apply a secure scheme to get a compression function.
- AO primitives can be modeled as polynomials.
- Must be protected against *classic* and *algebraic* attacks.

MiMC [AGR+16]



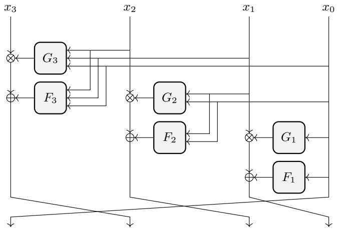
- MiMC: Minimal Multiplicative Complexity.
- Extremely simple: *round function* is $x^3 + c$.
- Exponent is the lowest integer in $\mathbb{F}_p$ coprime with $p - 1$.
- Many rounds to be secure against *algebraic attacks*.
- MiMCHash-256: 640 constraints.

POSEIDON [GKR+21]



- POSEIDON: Partial *substitution-permutation network* (SPN).
- Full rounds defend against classic attacks.
- Partial rounds defend against algebraic attacks.
- POSEIDON-256: 240 constraints.

GRIFFIN [GHR+22]



- GRIFFIN is 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 constraints.

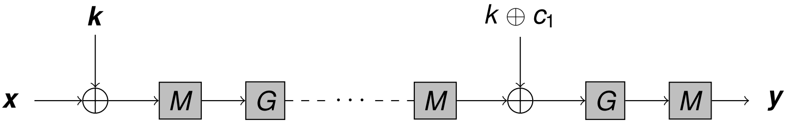
The GTDS [RS22]

$$
\begin{array}{lcl}
x_1 & \longmapsto & f(x_n, x_{n-1}, \ldots, x_2, x_1) \\
x_2 & \longmapsto & f(x_n, x_{n-1}, \ldots, x_2) \\
\cdots & \longmapsto & \cdots \\
x_{n-1} & \longmapsto & f(x_n, x_{n-1}) \\
x_n & \longmapsto & f(x_n)
\end{array}
$$

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

- Includes and improves previous design strategies.
- $f(x_n) = y_n = x^{1/d_2}; \quad f(x_n, \ldots, x_i) = y_i = x_i^{d_1} g_i(\sigma_{i+1}) + h_i(\sigma_{i+1}).$
- $\sigma_i = \sum_{j=i}^{n} x_j + y_j; \quad g_i(x) = x^2 + \alpha_i x + \beta_i; \quad h_i(x) = x^2 + \gamma_i x.$
- π-equivalence: constraint systems unaffected by permutations.

Arion and ArionHash [RST23]



Arion, a new keyed permutation from the GTDS:

- Exponent d_2: easy to exponentiate by, inverse is big.
- Affine layer is an MDS circulant matrix easy to multiply by.
- Achieves degree overflow in just one round.
- ArionHash: OWCF based on Arion in sponge mode.
- α-ArionHash: 76 constraints, same guarantees as competitors.

Comparisons

`libsnark`: used by ZCash [BSCG$^+$14] for its blockchain.
We used it to implement:

- Several primitives designed for ZK-SNARK, including ours.
- A self-parametrizing Merkle tree.
- A new mode of hash, the Augmented Binary tRee [ABR21].

Proof generation times for MT commitments over 256-bit prime fields

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

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.

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.

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.