New Modes on the Block: Security and Efficiency of Novel AO Compression Modes

ZKSC 2026, Vienna

Stefano Trevisani

Joint work with: E. Andreeva, R. Bhattacharyya, A. Roy

Hash functions and ZK-SNARKs

Cryptographic Hash Functions play a central role in SNARKs:

- In the Fiat-Shamir transform, enabling the 'N' in SNARK
- In the **Merkle Tree** commitment scheme:
  - Used in the ZK-STARK/FRI PCS.
  - In recursive SNARKs and IVC protocols.

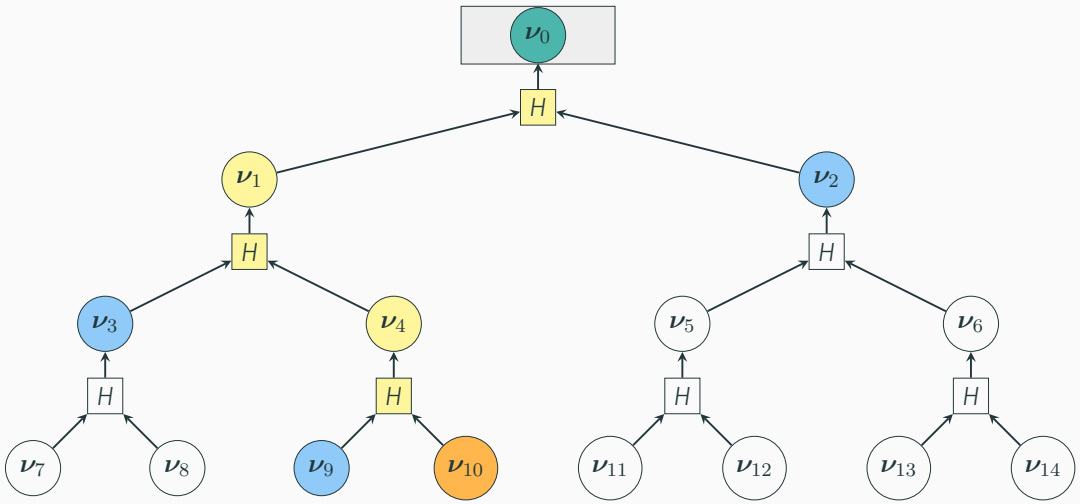
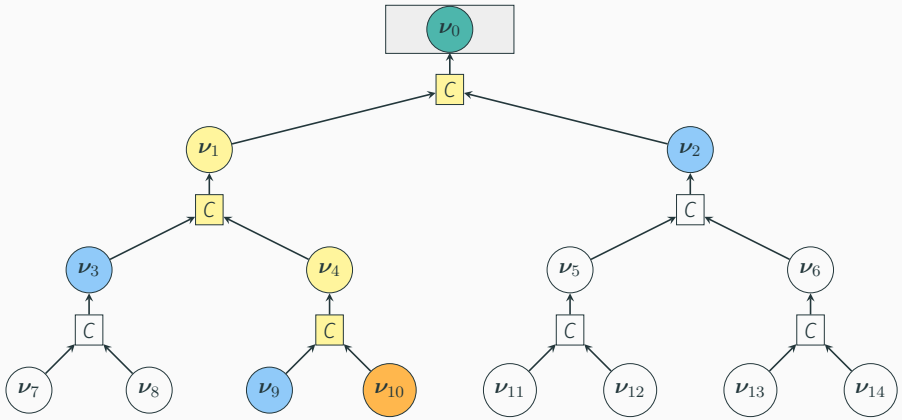


Figure 1: A binary Merkle Tree. Highlighted is an authentication path.

Merkle Trees Commitments

Merkle Trees used to instantiate a **vector commitment scheme**:

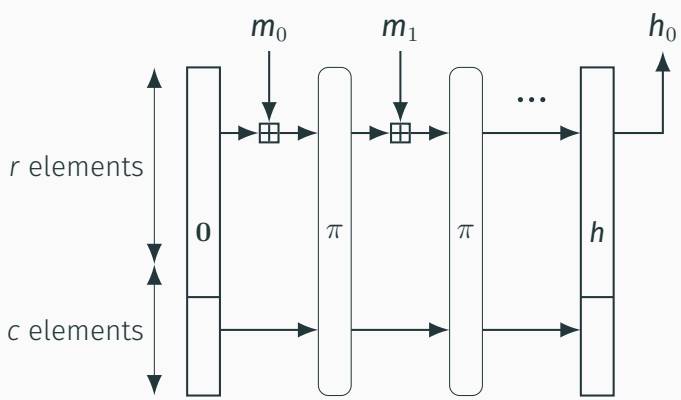
- The internal hash function is a $t:1$ compression function.
  - Usually $t = 2$.
- $\mathrm{Com}(\boldsymbol{m})$:
  - Split $\boldsymbol{m}$ into leaf nodes $\boldsymbol{m}_1 = \boldsymbol{\nu}_{2^h-1}, \ldots, \boldsymbol{m}_{2^h} = \boldsymbol{\nu}_{2^{h+1}-2}$.
  - Compute and output commitment $\boldsymbol{c} = \boldsymbol{\nu}_0$.
- $\mathrm{Open}(i)$: output $\boldsymbol{m}_i$'s co-path to the root.
  - Importantly: opening $\boldsymbol{\rho}$ is logarithmic in $|\boldsymbol{m}|$.
- $\mathrm{Ver}(\boldsymbol{c}, i, \boldsymbol{m}_i, \boldsymbol{\rho})$: compute $\boldsymbol{c}'$ from $(i, \boldsymbol{m}_i, \boldsymbol{\rho})$, accept iff $\boldsymbol{c}' = \boldsymbol{c}$.



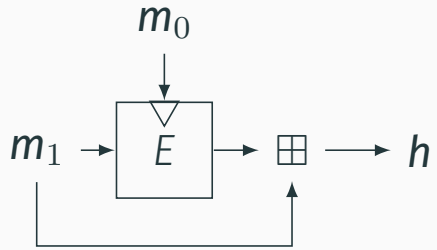
Compression Modes in the Wild

Which compression modes are used in practice?

- Blockcipher-based modes
  - PGV modes (e.g. Davies-Meyer, SHA-2).
- Permutation-based modes
  - Sponge (e.g. SHA-3).
    - Provably secure in the ideal-permutation model.
  - `Jive` (proposed with `Anemoi`).
  - Trunc (proposed with GRIFFIN, POSEIDON2, ...).
    - No provable security analysis.



Sponge



Davies-Meyer

Why Sponge?

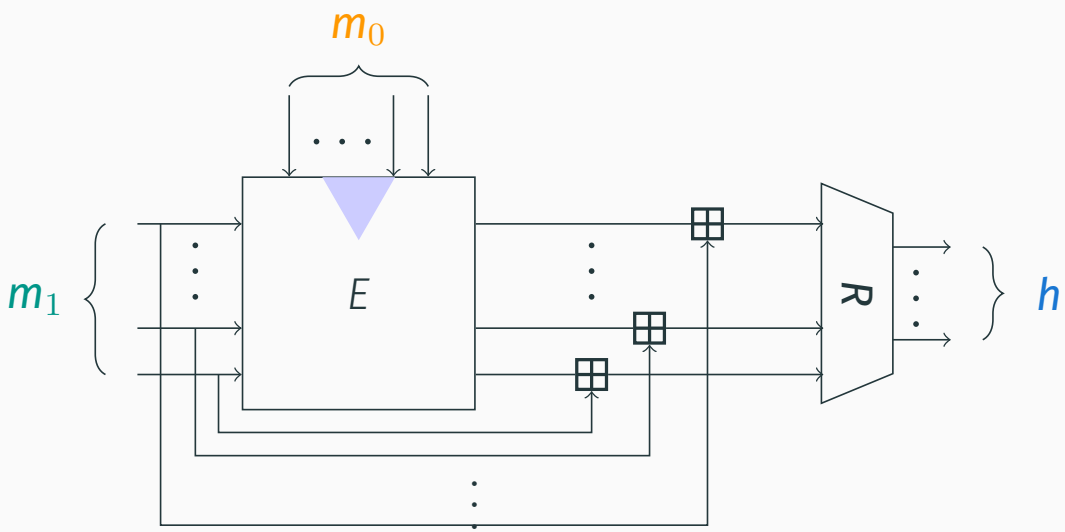
Sponge mode:

- RO-indifferentiable in the ideal permutation model.
- Single-round sponge (sponge-1):
  - CICO-1, CICO-k, CICO-(k_1, k_2)...
- Limitations of Sponge:
  - Suboptimal security bounds (especially collision/preimage resistance).
  - Most efficient?
- Many AO permutation (Poseidon-π, *Rescue*) are blockcipher-based.
  - Construct modes directly from the block cipher.
  - Better permutation-based modes.

The PGV-LC modes

At CSF'24, we introduced the PGV-LC **modes**:

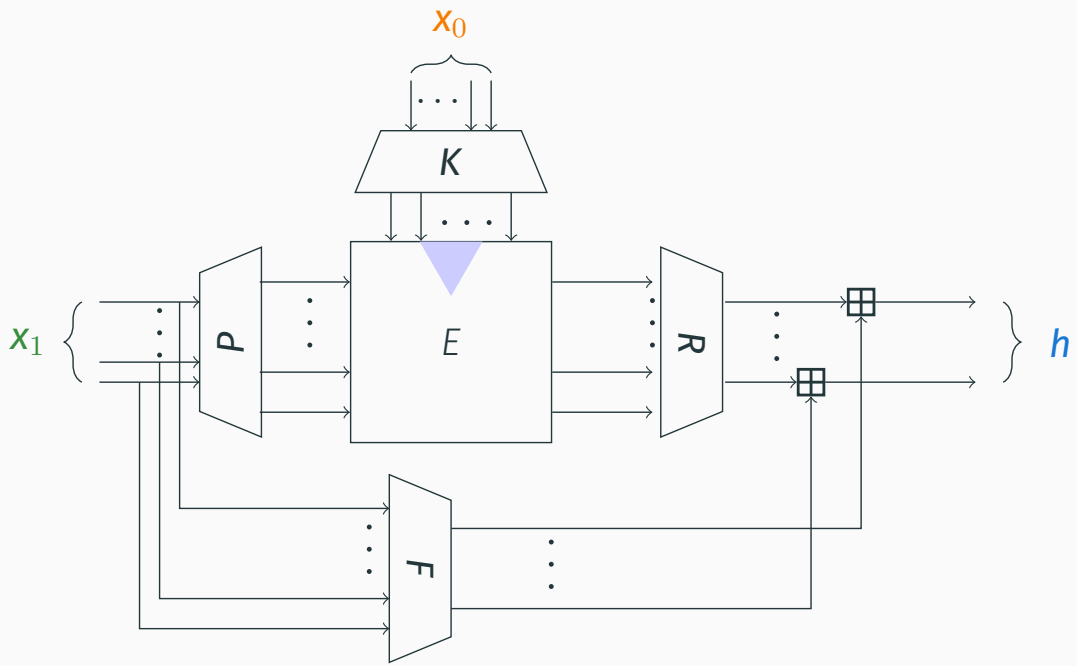
- Uses a blockcipher $E\colon \mathbb{F}_p^{\kappa} \times \mathbb{F}_p^n \to \mathbb{F}_p^n$.
- Matrix $\boldsymbol{R} \in \mathbb{F}_p^{\ell \times n}$ parametrizes output size.
  - ◇ Compresses its input $\Rightarrow \ell \leq n$.
  - ◇ Algebraic generalization of e.g. truncation and chopping.



The PGV-ELC mode

And an immediate extension thereof, called PGV-ELC:

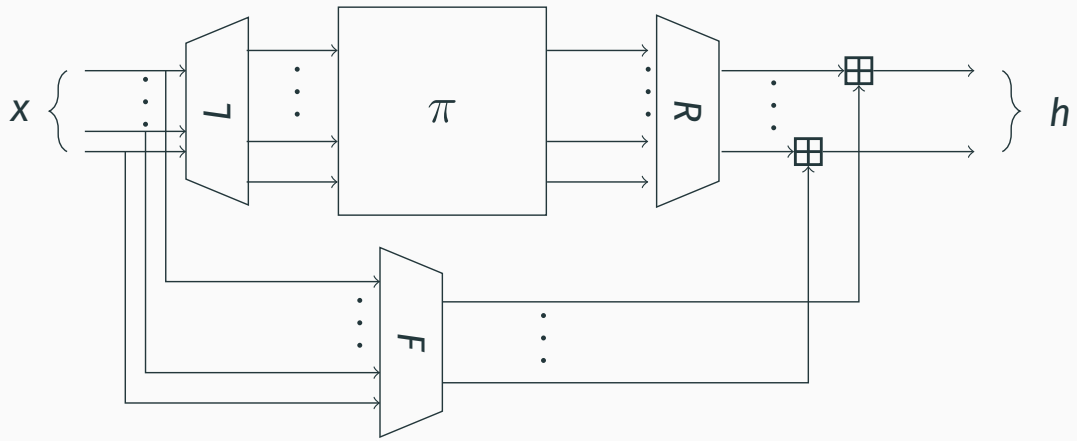
- Matrices $K \in \mathbb{F}_p^{\kappa \times \kappa'}$ and $P \in \mathbb{F}_p^{n \times n'}$ parametrize input size.
  - ◇ Expand their inputs $\Rightarrow \kappa' \leq \kappa$ and $n' \leq n$.
  - ◇ Algebraic generalization of e.g. zero-padding.
- Matrix $F \in \mathbb{F}_p^{\ell \times n'}$ adapts input to output size.



The PAX modes

We are also proposing the PAX **family of modes** [ongoing work]:

- Based on a permutation $\pi : \mathbb{F}_p^n \to \mathbb{F}_p^n$.
- $\mathcal{C}_{L,F,R,\pi}$ maps $\boldsymbol{x} \in \mathbb{F}_p^m$ to $\boldsymbol{h} \in \mathbb{F}_p^\ell$, with $\ell \leq m$.
- Expansion matrix $\boldsymbol{L} \in \mathbb{F}_p^{n \times m}$.
- Compression matrices $\boldsymbol{F} \in \mathbb{F}_p^{\ell \times m}$ and $\boldsymbol{R} \in \mathbb{F}_p^{\ell \times n}$.



PAX as a Generalization of Existing Modes

The PAX family generalizes most of the used constructions:

- $\mathrm{Trunc}\ C(x) = \lfloor \pi(x) + x \rfloor_\ell$:
  - Fix $n = m$, L, F, R are all (pseudo-)identities.
- `Jive` $C(x) = \sum_i x_i + \sum_i y_i$, with $y = \pi(x)$:
  - Fix $n = m$, L is the identity, F and R are circulant matrices on the repeated ith canonical vector of $\mathbb{F}_p^\ell$.
- First Sponge round: $C(x) = \lfloor \pi(x \parallel \mathbf{0}) \rfloor_\ell$
  - L, R pseudo-identities, $R = \mathbf{0}$.
  - Note: R is (clearly) not right-invertibile.

Provable Security

Black-box security of cryptographic constructions:

- Well-established model.
- Some primitives $\mathcal{P}_1, \ldots, \mathcal{P}_n$ used in a *mode*.
  - Often one primitive $\mathcal{P}$.
  - $\mathcal{P}$ is *ideal* (randomly sampled).
  - An adversary $\mathcal{A}$ can query the primitive(s).
- Security of a mode with respect to some property is expressed via an advantage function $\mathbf{Adv}(\mathcal{A}, q)$.

Compression Modes Security

For a compression mode $\mathcal{C}$:

- Often $\mathcal{P}$ is a blockcipher or a permutation.
  - Say over alphabet $\mathbb{F}_p$.
- Adversary makes q queries to $\mathcal{P}$.
- Some standard security properties:
  - Collision resistance:

$$\mathbf{Adv}^{\mathsf{COL}}_{\mathcal{C}}(\mathcal{A}, q) = \Pr\left[\mathcal{C}^{\mathcal{P}}(x) = \mathcal{C}^{\mathcal{P}}(x') \wedge x \neq x' \;\middle|\; (x, x') \leftarrow \mathcal{A}^{\mathcal{P}}()\right]$$

  - Preimage resistance:

$$\mathbf{Adv}^{\mathsf{PRE}}_{\mathcal{C}}(\mathcal{A}, q) = \Pr\left[\mathcal{C}^{\mathcal{P}}(x) = \mathcal{C}^{\mathcal{P}}(x') \;\middle|\; x \xleftarrow{\$} M; x' \leftarrow \mathcal{A}^{\mathcal{P}}\left(\mathcal{C}^{\mathcal{P}}(x)\right)\right]$$

  - Second-Preimage resistance (implied by collision resistance):

$$\mathbf{Adv}^{\mathsf{PRE2}}_{\mathcal{C}}(\mathcal{A}, q) = \Pr\left[\mathcal{C}^{\mathcal{P}}(x) = \mathcal{C}^{\mathcal{P}}(x') \wedge x \neq x' \;\middle|\; x \xleftarrow{\$} M; x' \leftarrow \mathcal{A}^{\mathcal{P}}(x)\right]$$

Random-Oracle Indifferentiability

Random-oracle indifferentiability is the ideal notion:

- Let $\mathcal{H}$ be a random oracle with compatible domain and range.
- Let $\mathcal{S}$ be a simulator algorithm
  - Makes $q_{\mathcal{S}}$ queries to $\mathcal{H}$, with $q_{\mathcal{S}}$ small.
- Let $\mathcal{D}$ be a differentiator:
  - Makes $q_{\mathcal{P}}$ queries to its left interface.
  - Makes $q_{\mathcal{H}}$ queries to its right interface.
- Then, letting $q = q_{\mathcal{P}} + q_{\mathcal{H}}$:

$$\mathbf{Adv}^{\mathsf{INDIF}}_{\mathcal{C},\mathcal{S}}(\mathcal{D}, q) = \left|\Pr\left[\mathcal{D}^{\mathcal{P},\mathcal{C}^{\mathcal{P}}}() = \top\right] - \Pr\left[\mathcal{D}^{\mathcal{S}^{\mathcal{H}},\mathcal{H}}() = \top\right]\right|$$

Merkle Tree Security

Which properties do we need for Merkle Tree commitments?

- Tree $\mathcal{T}$ using $C = \mathcal{C}^{\mathcal{P}}$ as the internal compression function.
  - $\mathcal{T}^C$ is also a compression function.
- Local opening security:

$$\mathbf{Adv}^{\mathrm{OPEN}}_{\mathcal{T}}(\mathcal{A}, q) = \Pr\left[\mathrm{Ver}\left(\mathcal{T}^C(m), i, m'_i, \rho'\right) = \top \wedge m'_i \neq m_i \;\middle|\; (m, i, m'_i, \rho') \leftarrow \mathcal{A}^{\mathcal{P}}()\right]$$

- It can be shown that:

$$\mathbf{Adv}^{\mathrm{OPEN}}_{\mathcal{T}}(q) \leq \mathbf{Adv}^{\mathrm{COL}}_{\mathcal{C}}(q)^1$$

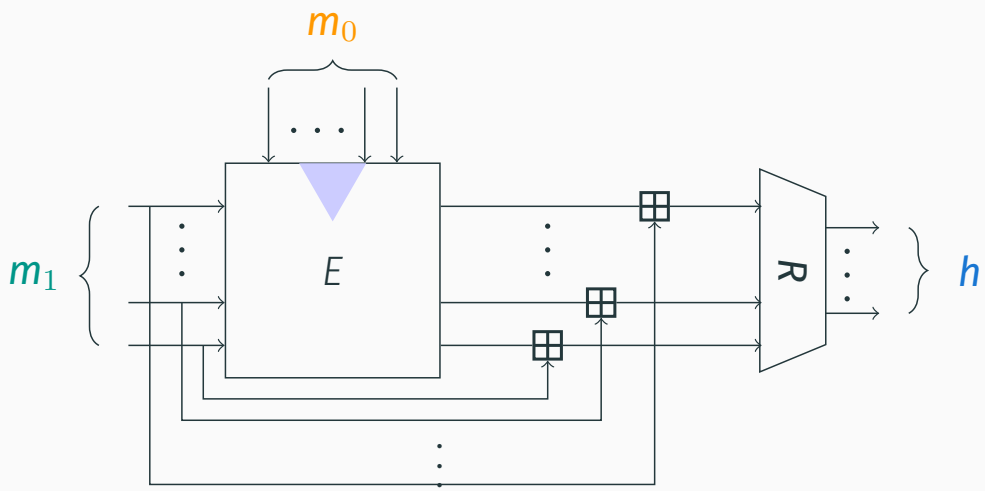
- Indifferentiability can however still be valuable.

[1] Or $2q$ in a slightly different notion.

PGV-LC Security

Collision resistance of PGV-LC:

1. Consider R right-invertible (full row rank).
2. $\mathbb{F}_p^n$ is partitioned into p^ℓ equivalence classes.
3. $\mathcal{A}$ can exploit partition unbalances from oracle replies.
4. Still, $\mathbf{Adv}_C^{\mathrm{COL}}(q) \leq \frac{q^2+q}{p^\ell - q}$ ($\approx$ birthday attack).
5. Similar reasoning for preimages: $\mathbf{Adv}_C^{\mathrm{PRE}}(q) \leq \frac{q}{p^\ell - q}$.



PGV-ELC Security

Collision resistance of PGV-ELC:

1. Consider K and P left-invertible, F right-invertible.
   - Also induce partitions of respective spaces.
2. 'Meaningless' queries, can be exploited only indirectly.
3. Again, $\mathbf{Adv}_C^{\mathrm{COL}}(q) \leq \frac{q^2+q}{p^\ell - q}$.
4. And for preimage resistance: $\mathbf{Adv}_C^{\mathrm{PRE}}(q) \leq \frac{q}{p^\ell - q}$.

Security of PAX

Indifferentiability of PAX based on the following simulator:

function $\mathcal{S}^{\mathcal{H}}(b \in \{-1,1\}, z \in \mathbb{F}_p^n) \to \mathbb{F}_p^n$

- **static** $T \leftarrow \emptyset$
- **if** $b = 1$ **then** ▷ *Forward query*
  - **if** $z \in \mathrm{Dom}(T)$ **then return** $T(z)$
  - $v \leftarrow L^{+}z$
  - **if** $z = Lv$ **then** ▷ *Queried point has a preimage*
    - $h \leftarrow \mathcal{H}(v)$
    - $y \xleftarrow{\$} [h - Fv]_R \setminus \mathrm{Ran}(T)$ ▷ $[x]_R$ *equiv. class of* x
  - **else** ▷ *Queried point is preimage-free*
    - $y \xleftarrow{\$} \mathbb{F}_p^n \setminus \mathrm{Ran}(T)$
  - $T \leftarrow T \cup \{(z, y)\}$
  - **return** y
- **else** ▷ *Backward query*
  - **if** $z \in \mathrm{Ran}(T)$ **then return** $T^{-1}(z)$
  - $x \xleftarrow{\$} \mathbb{F}_p^n \setminus \mathrm{Dom}(T)$
  - $T \leftarrow T \cup \{(x, z)\}$
  - **return** x

Modes security comparison

Adversarial advantages for a compression function $\mathbb{F}_p^m \to \mathbb{F}_p^\ell$:

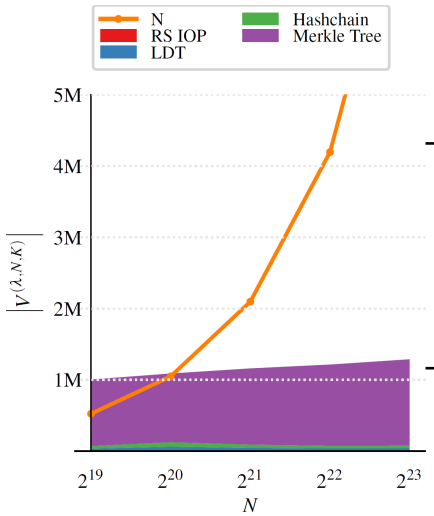
Mode	Primitive	COL	PRE	DIF
PGV-ELC	Block$(m/2+c, m/2+c)$	q^2/p^ℓ	q/p^ℓ	q/p^c
PAX	Perm$(m+c)$	q^2/p^ℓ	q/p^ℓ	q/p^c
Sponge-1	Perm$(m+c)$	$\min\{q^2/p^\ell, q/p^c\}$	$q/p^{\min\{\ell,c\}}$	q/p^c
Sponge	Perm$(r+c)$	$q^2/p^{\min\{\ell,c\}}$	$q/p^{\min\{\ell,c\}}$	q^2/p^c

- PAX and PGV-LC have optimal COL and PRE resistance.
- Single-iteration sponge (Sponge-1) has better bounds than multi-iteration.
- COL and PRE resistance of Sponge are sub-optimal.
- PGV-ELC indifferentiability does not depend on the key 'capacity'.

Impact of Hashing

How much is hashing relevant in a SNARK computation?

- In FRACTAL's verifier circuit [6], about 90% of constraints are from hash functions.
- In the RISC Zero chess example [13], about 37% of total time is spent on hashing.
- *Circuit* and *plain* performance are both important.
  - ZKProofMarket: by 2030 2^{27} proofs per day.
  - Even a 1ms saving per proof: $\approx 2^{24}$ seconds per day.



N	2^{20}	2^{21}	2^{22}	2^{23}
RS-IOP	2,948	3,036	3,124	3,212
LDT	58,586	42,658	36,190	37,598
Hashchain	65,184	46,878	37,566	37,566
Merkle Tree	962,544	1,068,122	1,137,818	1,210,418
- leaf hash	235,840	235,642	284,746	309,298
- 2-to-1 hash	663,830	760,624	781,216	829,264
- cap hash	62,874	71,856	71,856	71,856
Total	**1,132,666**	**1,210,292**	**1,264,299**	**1,338,406**

Caveat: Heuristic Security

Instantiating a compression functions:

- Target a specific security level (e.g. 128-bits for collision resistance).
- Choose a concrete underlying primitive:
  - parametrization based on cryptanalytical bounds.
- For PGV-ELC modes, this is not completely clear.
  - Need for cryptanalysis on AO blockcipher-based compression.
  - Experiments still relevant to establish efficiency margin.
- For PAX modes, results about previous permutation-based modes apply.

Arithmetization-Oriented Hash Functions

How to address the bottleneck?

- Plain performance: traditional solutions (e.g. SHA) work well.
- Circuit performance:
  - Arithmetic circuit over a field $\mathbb{F}_{p^d}$.
  - In SNARKs that rely on pairing-friendly EC: $p \approx 2^{256}$, $d = 1$.
  - For STARKs that rely on FRI IOPP: $p \approx 2^{32}$, $p \approx 2^{64}$, $d = 1$.
  - Multiplicative complexity/depth is an important starting point.
  - But cost is really tied to the *arithmetization technique*.
- *Arithmetization-Oriented* hash functions.

Arithmetization Techniques

Which arithmetization technique?

- R1CS arithmetization:

$$Ax \odot Bx = Cx$$

- (basic) $\mathcal{PlonK}$ arithmetization:

$$\left\{ q_{\mathrm{x}} x + q_{\mathrm{y}} y + q_{\mathrm{z}} z + q_{\mathrm{xy}} xy + q_c = 0 \right.$$

- AIR arithmetization:

$$T(x_{\mathrm{in}}, x_{\mathrm{out}}) \qquad \left\{ B(x) \right.$$

- Lookup arguments: efficient range-checks and more.
- Traditional solutions become arithmetization-oriented?

PGV-LC **Proof Generation**

Time to generate a MT opening proof on Groth16:

- Scalar field of the BLS12-381 elliptic curve: $\log_2(p) \approx 255$.

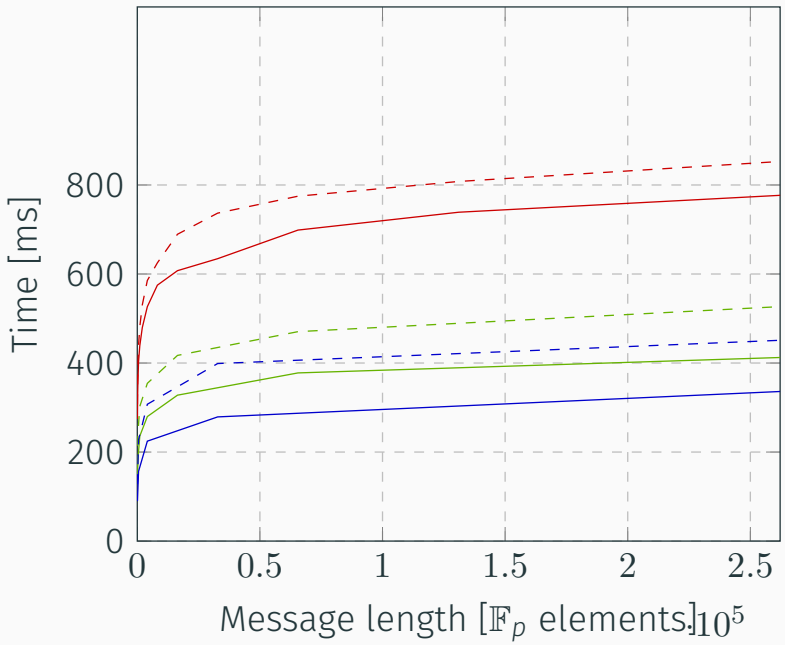


Figure 3: Dashed: Poseidon, Solid: Hades with PGV-LC

PGV-LC **Root Computation**

Time to compute the root of a Merkle Tree:

- Scalar field of the BLS12-381 elliptic curve: $\log_2(p) \approx 255$.

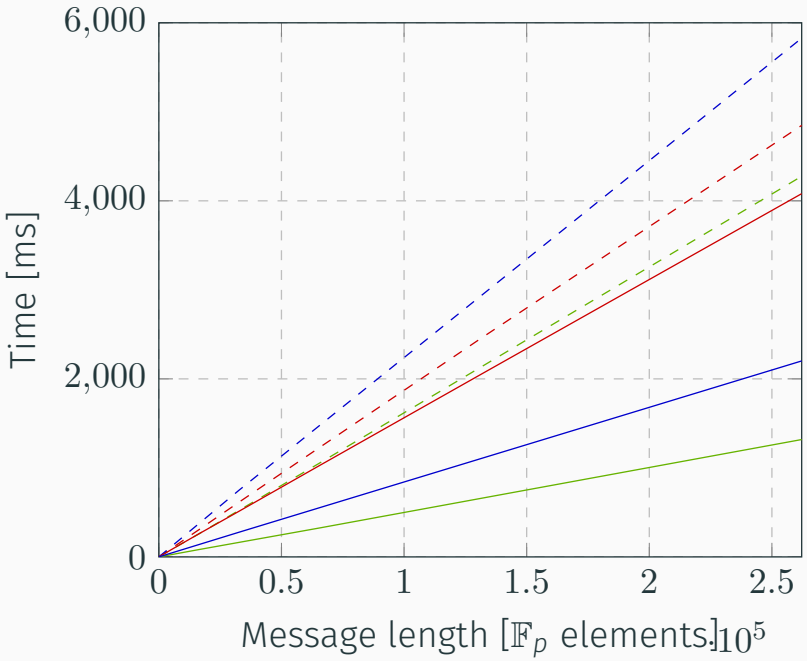


Figure 4: Dashed: Poseidon, Solid: Hades with PGV-LC

Plain performance

Comp. ratio	BN254 PAX	BN254 Sponge	Goldilocks PAX	Goldilocks Sponge
		Poseidon-π		
$2:1$	$9.4\,\mu s$	$17.5\,\mu s$	$3.5\,\mu s$	$7.3\,\mu s$
$4:1$	$28.2\,\mu s$	$42.4\,\mu s$	$12.2\,\mu s$	$18.6\,\mu s$
$8:1$	$103.7\,\mu s$	$130.4\,\mu s$	$141.5\,\mu s$	$265.4\,\mu s$
		Rescue		
$2:1$	$334.6\,\mu s$	$353.2\,\mu s$	$16.8\,\mu s$	$26.2\,\mu s$
$4:1$	$372.0\,\mu s$	$382.9\,\mu s$	$36.2\,\mu s$	$46.9\,\mu s$
$8:1$	$553.5\,\mu s$	$626.1\,\mu s$	$117.7\,\mu s$	$153.5\,\mu s$

Groth16 benchmarks

Preimage circuit in Groth16:

Comp. ratio	# R1CS constraints PAX	# R1CS constraints Sponge	Proof Generation time PAX	Proof Generation time Sponge
		Poseidon-π		
$2:1$	**221**	246	44.5 ms	47.3 ms
$4:1$	**268**	293	51.3 ms	54.9 ms
$8:1$	**368**	393	66.3 ms	71.5 ms
		Rescue		
$2:1$	**240**	252	41.2 ms	42.3 ms
$4:1$	**264**	270	44.9 ms	45.1 ms
$8:1$	**384**	432	63.8 ms	67.7 ms

Plonky2 benchmarks

Preimage circuit in *Plonky2*:

Comp. ratio	# gates		Proof Generation time	
	PAX	Sponge	PAX	Sponge
Poseidon-π				
$2:1$	**122**	259	11.3 ms	16.5 ms
$4:1$	**439**	668	26.0 ms	27.1 ms
$8:1$	**2065**	2864	90.8 ms	92.9 ms
Rescue				
$2:1$	**91**	175	10.9 ms	17.2 ms
$4:1$	**284**	418	16.8 ms	27.1 ms
$8:1$	**976**	1213	47.9 ms	49.0 ms

Roll credits

Fin

Thank you for your attention!

Any questions?

Bibliography i

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.

Bibliography ii

- Elena Andreeva, Rishiraj Bhattacharyya, Arnab Roy, and Stefano Trevisani.
  On Efficient and Secure Compression Functions for Arithmetization-Oriented Hashing.
  In *2024 IEEE 37th Computer Security Foundations Symposium (CSF)*, pages 1–16, Los Alamitos, CA, USA, Jul 2024. IEEE Computer Society.
- Amit Singh Bhati, Erik Pohle, Aysajan Abidin, Elena Andreeva, and Bart Preneel.
  Let's go eevee! a friendly and suitable family of aead modes for iot-to-cloud secure computation.

Bibliography iii

In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, pages 2546–2560, New York, NY, USA, 2023. Association for Computing Machinery.

John Black, Phillip Rogaway, and Thomas Shrimpton. **Black-box analysis of the block-cipher-based hash-function constructions from pgv.** In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 320–335, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

Bibliography iv


Joppe W. Bos and Peter L. Montgomery.
Montgomery arithmetic from a software perspective.
Cryptology ePrint Archive, Paper 2017/1057, 2017.


Alessandro Chiesa, Dev Ojha, and Nicholas Spooner.
Fractal: Post-quantum and transparent recursive proofs from holography.
In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 769–793, Cham, 2020. Springer International Publishing.


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

Bibliography v

Lorenzo Grassi, Dmitry Khovratovich, and Markus Schofnegger.
Poseidon2: A faster version of the poseidon hash function.

Cryptology ePrint Archive, Paper 2023/323, 2023.

Lorenzo Grassi, Reinhard Lüftenegger, Christian Rechberger, Dragos Rotaru, and Markus Schofnegger.
On a generalization of substitution-permutation networks: The hades design strategy.
Cryptology ePrint Archive, Paper 2019/1107, 2019.

Bibliography vi

Jens Groth.

On the size of pairing-based non-interactive arguments.

In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

Dmitry Khovratovich, Mario Marhuenda Beltrán, and Bart Mennink.

Generic security of the safe api and its applications.

In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 301–327, Singapore, 2023. Springer Nature Singapore.

Bibliography vii

Ralph Charles Merkle.

Secrecy, Authentication, and Public Key Systems.

PhD thesis, Stanford University, Stanford, CA, USA, 1979. AAI8001972.

Tomer Solberg.

Risc zero prover protocol & analysis.

`https://github.com/ingonyama-zk/papers/blob/main/risc0_protocol_analysis.pdf`, 04 2023.

Accessed: 2026-02-08.