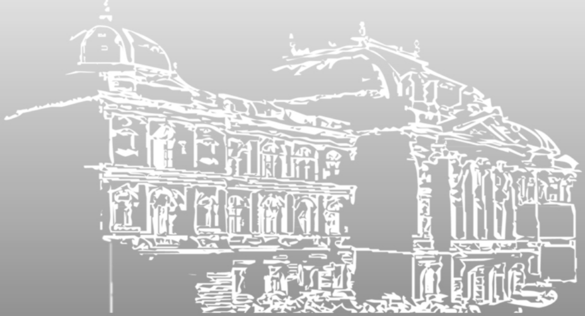


Skyscraper: Super Fast Hash for Big Primes

Clémence Bouvier Lorenzo Grassi Dmitry Khovratovich **Katharina Koschatko** Christian Rechberger **Fabian Schmid** Markus Schofnegger

Motivation

- Why a new hash function?
  - Cost of proving related to algebraic description
  - Traditional hashes have complex representations
  - Tailored designs improve many applications
- Focus on large primes
  - Used in pairing-based ZK protocols
  - Small proofs, fast verification
  - Efficiently used *on-chain*

Motivation

- Why a new hash function?
  - Cost of proving related to algebraic description
  - Traditional hashes have complex representations
  - Tailored designs improve many applications
- Focus on large primes
  - Used in pairing-based ZK protocols
  - Small proofs, fast verification
  - Efficiently used *on-chain*

Applications for Large Primes

- Generic Merkle tree proofs
  - Prove coin ownership
  - … or in general, set membership
- Provable randomness
  - Make proof non-interactive via Fiat–Shamir
  - Prove correct derivation of challenges/randomness
- Many more use cases
  - Authenticated and verifiable encryption
  - …

Applications for Large Primes

- Generic Merkle tree proofs
  - Prove coin ownership
  - … or in general, set membership
- Provable randomness
  - Make proof non-interactive via Fiat–Shamir
  - Prove correct derivation of challenges/randomness
- Many more use cases
  - Authenticated and verifiable encryption
  - …

Applications for Large Primes

- Generic Merkle tree proofs
  - Prove coin ownership
  - … or in general, set membership
- Provable randomness
  - Make proof non-interactive via Fiat–Shamir
  - Prove correct derivation of challenges/randomness
- Many more use cases
  - Authenticated and verifiable encryption
  - …

Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

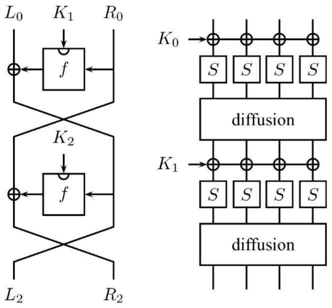
Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

How we achieve it:

- Feistel structure instead of SPN.



Feistel cipher versus SP network [CBP06, Fig. 4]

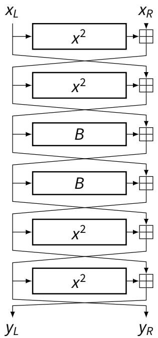
Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

How we achieve it:

- Non-invertible S-Boxes and Squarings.



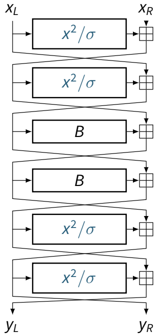
Design Strategy

What do we want:

- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

How we achieve it:

- Account for Montgomery reduction.



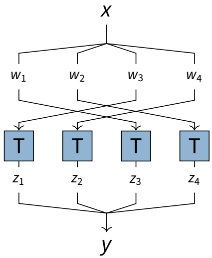
Design Strategy

What do we want:

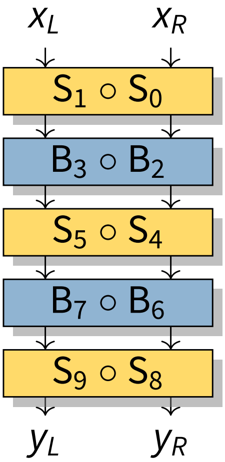
- Improve hashing performance for big prime fields.
- Make design as simple as possible to minimize attack vectors.
- Use lookups to increase the algebraic degree.

How we achieve it:

- Use simple 8-bit lookup tables. (cf. `Monolith` [GKL+24])
- S-box combined with circular shift.

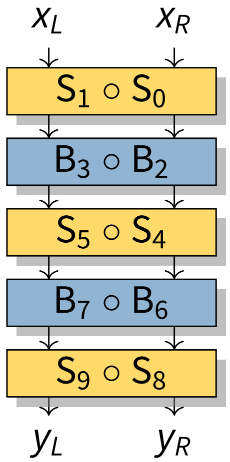


Skyscraper Design: Overview

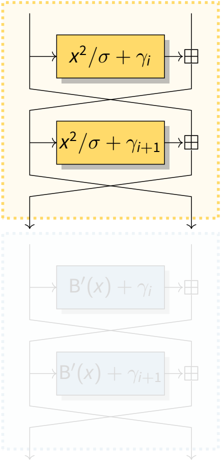


- Square operation S_i
  - Non-invertible x^2
  - Good statistical properties
  - Speed-up via Montgomery
- Bars operation B_i
  - Non-invertible S-Box B'
  - Applicable to any prime
  - High algebraic degree
  - Speed-up via efficient bit operations

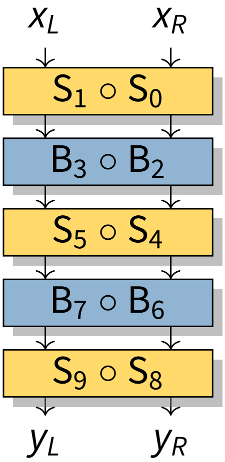
Skyscraper Design: Overview



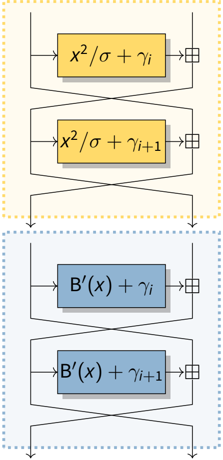
- Square operation S_i
  - Non-invertible x^2
  - Good statistical properties
  - Speed-up via Montgomery
- Bars operation B_i
  - Non-invertible S-Box B'
  - Applicable to any prime
  - High algebraic degree
  - Speed-up via efficient bit operations



Skyscraper Design: Overview



- Square operation S_i
  - Non-invertible x^2
  - Good statistical properties
  - Speed-up via Montgomery
- Bars operation B_i
  - Non-invertible S-Box B'
  - Applicable to any prime
  - High algebraic degree
  - Speed-up via efficient bit operations



Skyscraper Design: Wide-State Dilemma

Question: Feistel takes only two inputs. How to "increase statesize"?

Skyscraper Design: Wide-State Dilemma

Question: Feistel takes only two inputs. How to "increase statesize"?

Switch to extension field $\mathbb{F}_{p^n}$

Skyscraper Design: Wide-State Dilemma

Question: Feistel takes only two inputs. How to "increase statesize"?

Switch to extension field $\mathbb{F}_{p^n}$

- $\mathbb{F}_{p^n} \equiv \mathbb{Z}_p[X]/G(x)$, where G is an irreducible polynomial of degree n
- Field element $x \in \mathbb{F}_{p^n}$ can be interpreted as element of $\mathbb{F}_p^n$

$$x_0 + x_1 \cdot X + \cdots + x_{n-1} \cdot X^{n-1} \equiv (x_0, x_1, \ldots, x_{n-1})$$

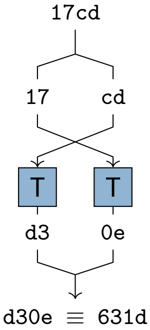
	Bar	Sq
$\mathbb{F}_p$	8	14
$\mathbb{F}_{p^2}$	12	52
$\mathbb{F}_{p^3}$	20	128

Table: Runtime [ns]

Skyscraper Design: S-Box component B′

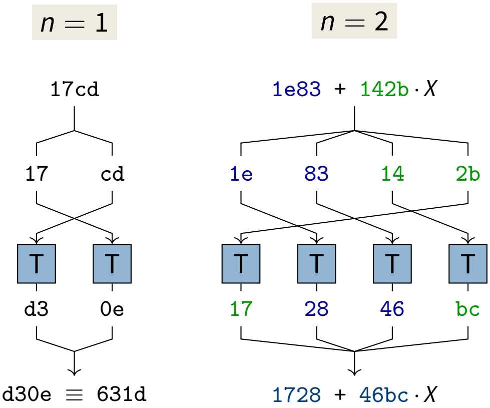
Examples: $B' : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$ for $p = 28657$ (15-bit prime)

$n = 1$



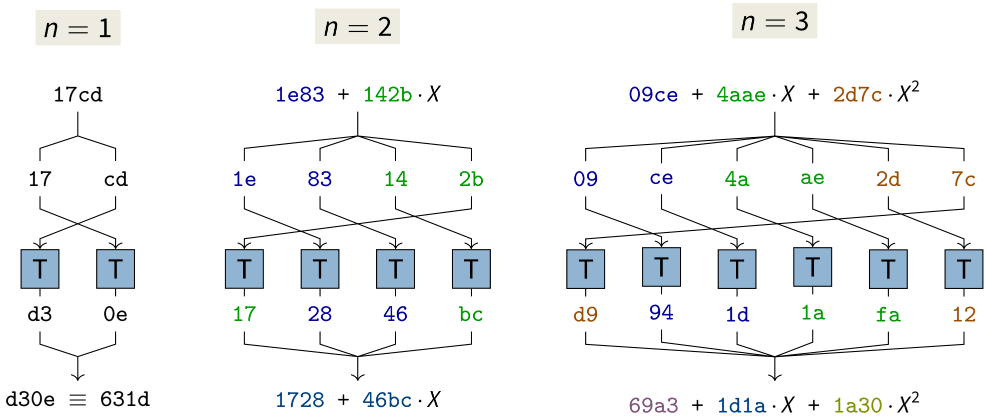
Skyscraper Design: S-Box component B′

Examples: $B' : \mathbb{F}_{p^n} \rightarrow \mathbb{F}_{p^n}$ for $p = 28657$ (15-bit prime)



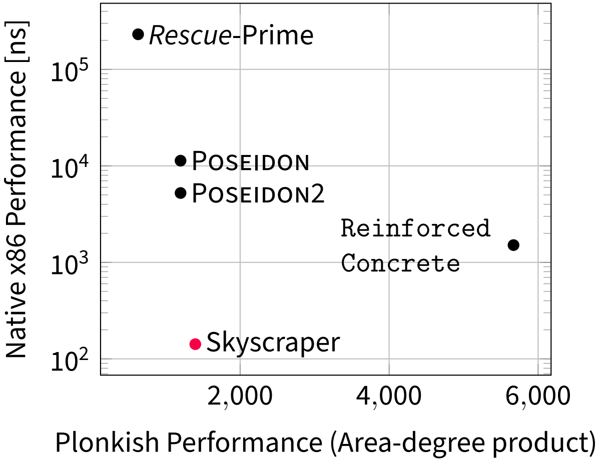
Skyscraper Design: S-Box component B′

Examples: $\mathrm{B}' : \mathbb{F}_{p^n} \rightarrow \mathbb{F}_{p^n}$ for $p = 28657$ (15-bit prime)



Performance Comparison for BN254

Hash Function	x86	ZK
Skyscraper	142	1 398
RC	1 510	5 670
POSEIDON	11 324	1 200
POSEIDON2	5 233	1 200
Rescue-Prime	230 950	630



Area-degree product = size of witness matrix × max. degree of polynomial that encodes a gate

Security Issues and Update

What happened:

- Rebound attack by Antoine Bak [Bak25]
- Attack on 9 round version
- No security margin

Skyscraper update:

- Increase number of Rounds
- Additional Squares impact native performance
- Additional Bars impact ZKP performance

Security Issues and Update

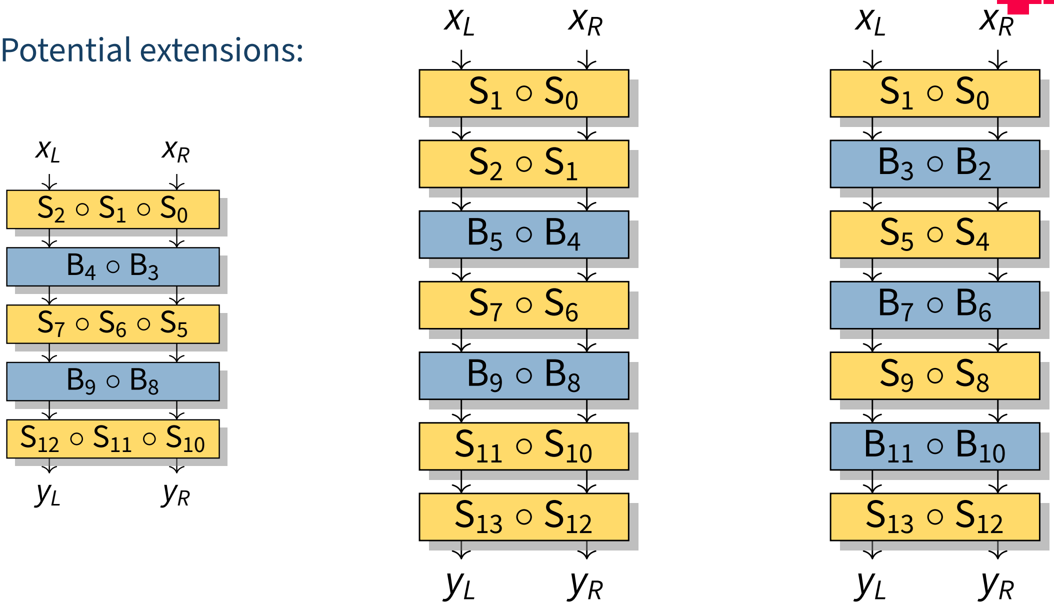
What happened:

- Rebound attack by Antoine Bak [Bak25]
- Attack on 9 round version
- No security margin

Skyscraper update:

- Increase number of Rounds
- Additional Squares impact native performance
- Additional Bars impact ZKP performance

Potential extensions:



Native performance for hashing over $\mathbb{F}_p$

Hash Function		Message 500 bit	Message 1 Mbit
SHA-256		1	1
Rescue-Prime		5610	10 541
Poseidon		197	561
Poseidon2		116	277
`Reinforced Concrete`		33	72
`Monolith-64`		2.9	1.3
Skyscraper	$\mathbb{F}_p$	3.2 *(4.1)*	15.1 *(19.6)*
	$\mathbb{F}_{p^2}$	8.2 *(12.8)*	9.6 *(15.3)*

Table: Native performance for hashing compared to SHA-256.

	Sq	Bar
$\mathbb{F}_p$	9.6%	4.4%
$\mathbb{F}_{p^2}$	14.7%	2.9%
$\mathbb{F}_{p^3}$	15.2%	2.1%

Table: Cost of Round functions

Conclusion

- New hash function Skyscraper
  - Efficient in plain and in proof systems for large primes
  - Plain performance comparable to SHA-3
- Feistel design strategy
  - Allows for non-invertible components
  - Remove affine layer (costly for large state or large round number)
  - Simple design (no growing state) due to extension field usage
- Generic Bar construction
  - Not fixed to specific prime
  - High algebraic degree

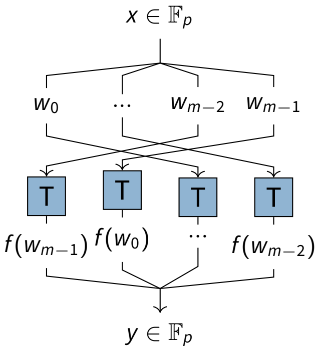
Conclusion

- New hash function Skyscraper
  - Efficient in plain and in proof systems for large primes
  - Plain performance comparable to SHA-3
- Feistel design strategy
  - Allows for non-invertible components
  - Remove affine layer (costly for large state or large round number)
  - Simple design (no growing state) due to extension field usage
- Generic Bar construction
  - Not fixed to specific prime
  - High algebraic degree

Conclusion

- New hash function Skyscraper
  - Efficient in plain and in proof systems for large primes
  - Plain performance comparable to SHA-3
- Feistel design strategy
  - Allows for non-invertible components
  - Remove affine layer (costly for large state or large round number)
  - Simple design (no growing state) due to extension field usage
- Generic Bar construction
  - Not fixed to specific prime
  - High algebraic degree

Algebraic Cryptanalysis: S-Box component B′



- Range constraints (w_i fits into s bits):

$$0 = \prod_{i=0}^{2^s-1} (w_k - i) \quad \forall\, 0 \leq k < m$$

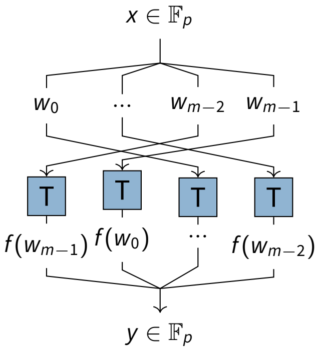
- Composition constraint:

$$x = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot w_k$$

- $\mathtt{Rot}_r$, $\mathtt{Sbox}$, and $\mathtt{Comp}$:

$$y = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot f(w_{k-r \bmod m})$$

Algebraic Cryptanalysis: S-Box component B′



- Range constraints (w_i fits into s bits):

$$0 = \prod_{i=0}^{2^s-1} (w_k - i) \quad \forall\, 0 \leq k < m$$

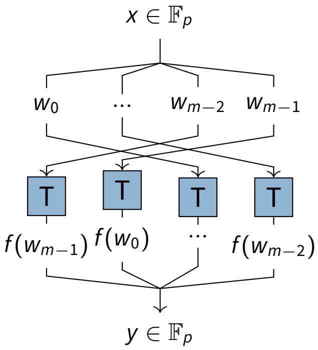
- Composition constraint:

$$x = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot w_k$$

- $\texttt{Rot}_r$, $\texttt{Sbox}$, and $\texttt{Comp}$:

$$y = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot f(w_{k-r \bmod m})$$

Algebraic Cryptanalysis: S-Box component B′



- Range constraints (w_i fits into s bits):

$$0 = \prod_{i=0}^{2^s-1} (w_k - i) \quad \forall\, 0 \leq k < m$$

- Composition constraint:

$$x = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot w_k$$

- $\texttt{Rot}_r$, $\texttt{Sbox}$, and $\texttt{Comp}$:

$$y = \sum_{k=0}^{m-1} 2^{s(m-1-k)} \cdot f(w_{k-r \bmod m})$$

Cryptanalysis (GB) results for Skyscraper

Instance				Alg. model			Gröbner basis computation (step 1)				
p	m	N	R	n_v	n_e	$d_{\max}$	F4 [s]	mem. [MB]	d_{reg}	Degrees GB_{DRL}	#Sol.
37	2	1	6	7	9	16	1	33	16	2-2-2-1-1-1-1-1	3
109	2	1	6	7	9	16	2	79	16	1-1-1-1-1-1-1	1
163	2	1	6	7	9	16	3	146	16	1-1-1-1-1-1-1	1
191	2	1	6	7	9	16	4	149	16	1-1-1-1-1-1-1	1
587	3	1	6	9	11	16	243	3661	17	1-1-1-1-1-1-1-1-1	1
1237	3	1	6	9	11	16	7550	31871	18	2-1-1-1-1-1-1-1-1	2
1361	3	1	6	9	11	16	18974	56732	18	1-1-1-1-1-1-1-1-1	1
1399	3	1	6	9	11	16	18912	56828	18	1-1-1-1-1-1-1-1-1	1
2297	3	1	6	9	11	16	51103	194271	20	1-1-1-1-1-1-1-1-1	1
2953	3	1	6	9	11	16	123706	385124	22	1-1-1-1-1-1-1-1-1	1

Table: GB attack against Skyscraper-CICO for a chunk size of $s = 4$, R rounds (2 squarings, 2 bars).

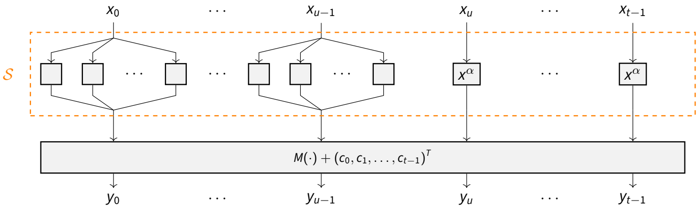
Cryptanalysis (GB) results for Skyscraper

Instance				Alg. model			Gröbner basis computation (step 1)				
p	m	N	R	n_v	n_e	$d_{\max}$	F4 [s]	mem. [MB]	d_{reg}	Degrees GB_{DRL}	#Sol.
10973	2	1	5	4	5	256	103	1665	256	1-1-1-1	1
12277	2	1	5	4	5	256	207	2033	256	1-1-1-1	1
12809	2	1	5	4	5	256	574	3660	256	2-1-1-1	2
17033	2	1	5	4	5	256	483	3407	256	1-1-1-1	1
25057	2	1	5	4	5	256	2604	11395	256	2-2-2-2-2-2-1	4
28837	2	1	5	4	5	256	1348	7956	256	1-1-1-1	1
45943	2	1	5	4	5	256	3282	17224	256	1-1-1-1	1
51647	2	1	5	4	5	256	4210	21075	256	1-1-1-1	1
52541	2	1	5	4	5	256	4312	21810	256	1-1-1-1	1

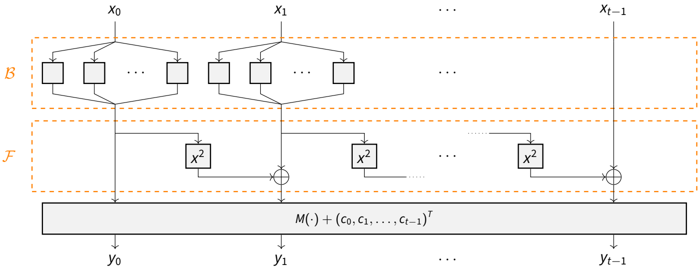
Table: GB attack against Skyscraper-CICO for a chunk size of $s = 8$, R rounds (2 squarings, 1 bar).

What about other lookup-based primitives?

Tip5



Monolith



Some cryptanalysis (GB) results for `Monolith`

CICO model	Alg. model				Gröbner basis		Basis conversion		Elimination	
	R	n_v	n_e	$d_{\max}$	F4 [s]	d_{reg}	FGLM [s]	$d_{\mathcal{I}}$	ELIM [s]	#Sol.
$(?, ?, 0) \mapsto (0, ?, ?)$	1	8	8	16	0.69	30	2.28	512	0.33	246
	2	12	13	16	7292.55	32	0.89	296	0.32	296
$(?, 0, 0) \mapsto (0, ?, ?)$	1	7	8	16	0.05	18	0.0	4	0.0	4
	2	11	13	16	54.23	30	0.0	1	0.0	1
	3	15	18	16	11031.2	32	0.0	1	0.0	1

Table: GB attack against `Monolith`-CICO over $\mathbb{F}_p$ for $p = 2^8 - 2^4 + 1$ for a chunk size of $s = 4$, state size $t = 3$, and $u = 1$ decomposition S-Boxes per round.

Observations and Questions

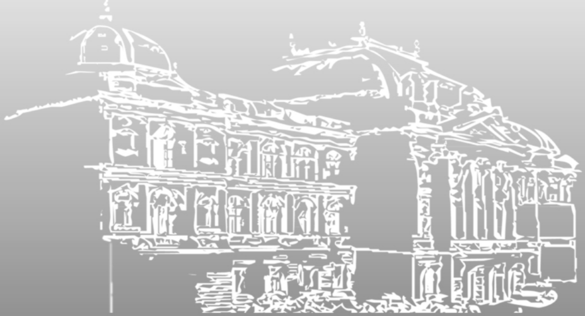
- Security based only on the first step in GB attack?
  - Only low degrees after GB step
  - Low quotient space dimension $d_{\mathcal{I}}$
- But: d_{reg} does not grow (fast)
  - Security from large number of variables in model
  - Better to have larger primes (= more chunks)
- Place for improvement
  - Use dedicated monomial order to skip first step?
  - Better way of modeling the lookup table?

Observations and Questions

- Security based only on the first step in GB attack?
  - Only low degrees after GB step
  - Low quotient space dimension $d_{\mathcal{I}}$
- But: d_{reg} does not grow (fast)
  - Security from large number of variables in model
  - Better to have larger primes (= more chunks)
- Place for improvement
  - Use dedicated monomial order to skip first step?
  - Better way of modeling the lookup table?

Observations and Questions

- Security based only on the first step in GB attack?
  - Only low degrees after GB step
  - Low quotient space dimension $d_{\mathcal{I}}$
- But: d_{reg} does not grow (fast)
  - Security from large number of variables in model
  - Better to have larger primes (= more chunks)
- Place for improvement
  - Use dedicated monomial order to skip first step?
  - Better way of modeling the lookup table?



Skyscraper: Super Fast Hash for Big Primes

Clémence Bouvier Lorenzo Grassi Dmitry Khovratovich **Katharina Koschatko** Christian Rechberger **Fabian Schmid** Markus Schofnegger

Bibliography I

[AAE+20] A. Aly, T. Ashur, Eli Ben-Sasson, S. Dhooghe, and A. Szepieniec. **Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols.** *IACR Trans. Symmetric Cryptol.* 2020.3 (2020), pp. 1–45.

[Bak25] A. Bak. **A practical distinguisher on the full Skyscraper permutation.** Cryptology ePrint Archive, Paper 2025/102. 2025 (cit. on pp. 24, 25).

[CBP06] C. Cannière, A. Biryukov, and B. Preneel. **An introduction to Block Cipher Cryptanalysis.** *Proceedings of the IEEE* 94 (Mar. 2006), pp. 346–356. DOI: 10.1109/JPROC.2005.862300 (cit. on p. 10).

[GKL+22] L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch. **Reinforced Concrete: A Fast Hash Function for Verifiable Computation.** CCS 2022. ACM, 2022, pp. 1323–1335. DOI: 10.1145/3548606.3560686.

Bibliography II

[GKL+24] L. Grassi, D. Khovratovich, R. Lüftenegger, C. Rechberger, M. Schofnegger, and R. Walch. **Monolith: Circuit-Friendly Hash Functions with New Nonlinear Layers for Fast and Constant-Time Implementations.** *IACR Trans. Symmetric Cryptol.* 2024.3 (2024), pp. 44–83. DOI: 10.46586/TOSC.V2024.I3.44-83 (cit. on pp. 13, 44).

[GKR+21] L. Grassi, D. Khovratovich, C. Rechberger, A. Roy, and M. Schofnegger. **Poseidon: A New Hash Function for Zero-Knowledge Proof Systems.** USENIX 2021. USENIX Association, 2021, pp. 519–535.

[GKS23] L. Grassi, D. Khovratovich, and M. Schofnegger. **Poseidon2: A Faster Version of the Poseidon Hash Function.** AFRICACRYPT 2023. Vol. 14064. LNCS. Springer, 2023, pp. 177–203. DOI: 10.1007/978-3-031-37679-5_8.

[SAD20] A. Szepieniec, T. Ashur, and S. Dhooghe. **Rescue-Prime: a Standard Specification (SoK).** Cryptology ePrint Archive, Report 2020/1143. https://ia.cr/2020/1143. 2020.

Plain Performance

	BN254			Goldilocks [GKL+24]		
Hashing algorithm	(t,deg.)	Rounds	Time [ns]	(t,deg.)	Rounds	Time [ns]
POSEIDON	(3,5)	64 (4+56+4)	11 324	(8,7)	30 (4+22+4)	1 898
POSEIDON2	(3,5)	64 (4+56+4)	5 233	(8,7)	30 (4+22+4)	1 292
Rescue-Prime	(3,5)	14	230 950	(8,7)	8	12 128
`Reinforced Concrete`	(3,5)	7 (3+1+3)	1 510			
Skyscraper	(2,2)	10 (6 Sq., 4 B)	142			
`Monolith`				(8,2)	6	130

Table: Native performance of compressing 512 bits (2-1 compression) over $\mathbb{F}_p^t$, where *t* denotes the statesize. For POSEIDON,POSEIDON2, *Rescue*-Prime, and `Reinforced Concrete`, "deg." denotes the non-linear permutation degree, that is, the smallest positive integer *d* such that $\gcd(d, p-1) = 1$ (e.g., $d = 5$ for BN254 and $d = 7$ for Goldilocks). In `Monolith` and Skyscraper, (non-invertible) squaring is employed as part of a Feistel construction.

Plonkish Performance

Component	Witnesses	Constraints (deg.)	Lookups	Area-degree product
		$n = 1$		
Double squaring	3	3 (2)	0	–
All squarings	9	9 (2)	0	–
`Bars`	56	10 (1)	32	–
All `Bars`	224	40 (1)	128	–
Skyscraper	233	49 (1, 2)	128	1398
`Reinforced Concrete`	378	24 (1, 3, d)	267	5670
POSEIDON, POSEIDON2	80	80 (d)	0	1200
Rescue-Prime	42	42 (d)	0	630

Table: Circuit performance of all components of Skyscraper, where we assume the BN254 setting. This also includes a comparison with `Reinforced Concrete`, POSEIDON, POSEIDON2, and *Rescue*-Prime, where d is the smallest positive integer such that $\gcd(d, p-1) = 1$ (e.g., $d = 5$ for BN254).