Preimage-Type Attacks for Reduced Ascon-Hash

Application to Ed25519

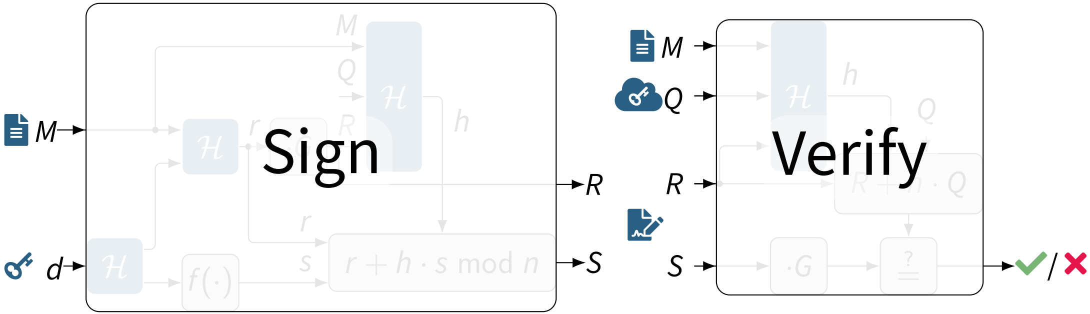
Marcel Nageler Lorenz Schmid Maria Eichlseder

SAC 2025 – Toronto

isec.tugraz.at

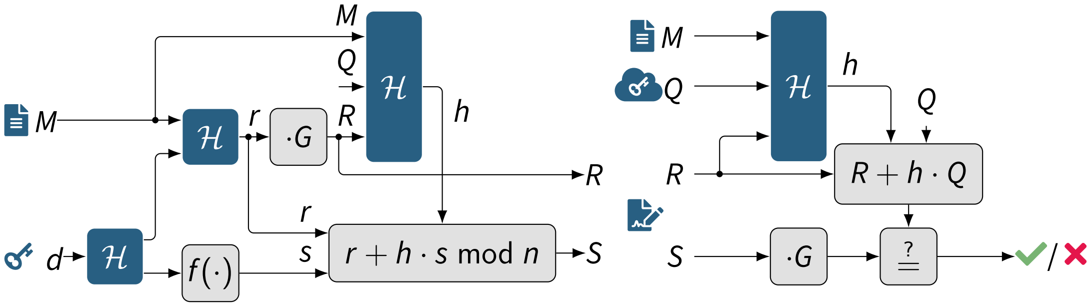
Ed25519 Signature Scheme [Nat23]

- Designed by Bernstein et al.standardized by NIST [Ber+11; Nat23]
- Used in SSH, Signal Protocol, ...
- Security based on elliptic curve and hash function



Ed25519 Signature Scheme [Nat23]

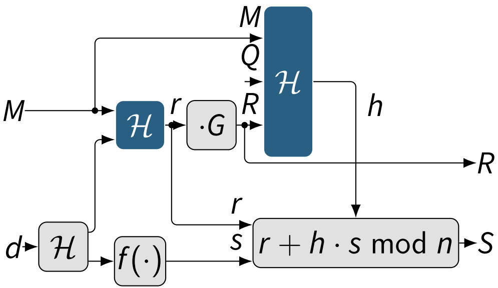
- Designed by Bernstein et al.standardized by NIST [Ber+11; Nat23]
- Used in SSH, Signal Protocol, …
- Security based on elliptic curve and hash function



Replace SHA-512 with Ascon-XOF [Nat24]

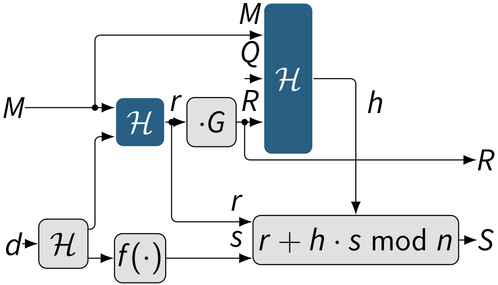
Security of Ed25519

- Based on **discrete log** on Curve25519
- Based on hash function
  - **secure PRF**
  - **secure commitment scheme**
- Ascon-XOF claims 128 bits of security
- No previous analysis in commitment scheme setting



Security of Ed25519

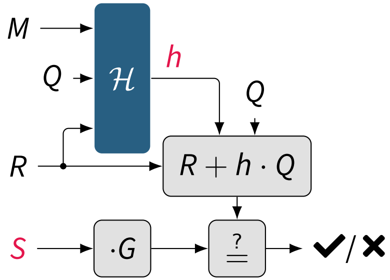
- Based on **discrete log** on Curve25519
- Based on hash function
  - **secure PRF**
  - **secure commitment scheme**
- Ascon-XOF claims 128 bits of security
- No previous analysis in commitment scheme setting



From Preimages to Forgeries

1. Choose h and S
2. Calculate $R = h \cdot Q - S \cdot G$
3. Find preimage M such that $\mathcal{H}(R \parallel Q \parallel M) = h$

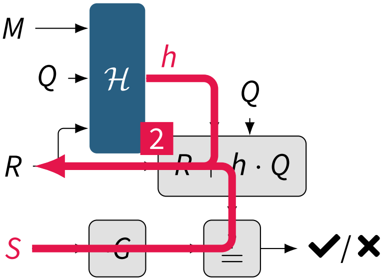
➔ $M, (R, S)$ is a forgery



From Preimages to Forgeries

1. Choose h and S
2. Calculate $R = h \cdot Q - S \cdot G$
3. Find preimage M such that $\mathcal{H}(R \parallel Q \parallel M) = h$

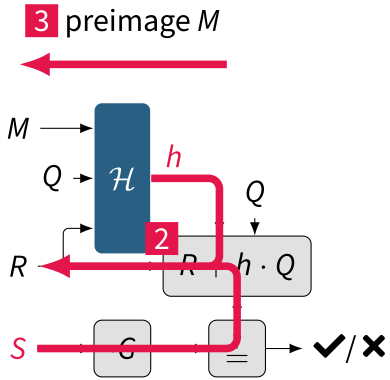
➔ $M, (R, S)$ is a forgery



From Preimages to Forgeries

1. Choose h and S
2. Calculate $R = h \cdot Q - S \cdot G$
3. Find preimage M such that $\mathcal{H}(R \parallel Q \parallel M) = h$

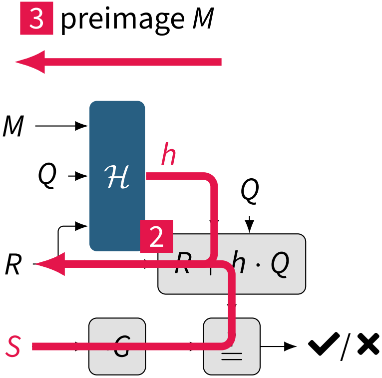
➔ $M, (R, S)$ is a forgery



From Preimages to Forgeries

1. Choose h and S
2. Calculate $R = h \cdot Q - S \cdot G$
3. Find preimage M such that $\mathcal{H}(R \parallel Q \parallel M) = h$

→ $M, (R, S)$ is a forgery



Related Work on outputs $\geq$ 256 bit

Output	Setting	#R	Complexity	Strategy	Reference
256 bit	preimage*	3	2^{163}	Meet-in-the-Middle	[Don+24]
	preimage*	3	2^{184}	Differential-Linear	[Niu+24]
	preimage*	4	2^{185}	Meet-in-the-Middle	[Don+24]
	preimage*	4	2^{189}	Differential-Linear	[Niu+24]
	preimage*	5	2^{191}	Meet-in-the-Middle	[Don+24]
any (n-bit)	2nd/rpp preimage	$\star$	2^{128}	Generic	
	2nd preimage	1	2^{64} GE	Linearization	Ours
	preimage	1	$2^{64}\ \mathrm{GE} + 2^{n-128}$	Linearization	Ours
	rpp preimage	1	$2^{29.7}\ \mathrm{GE} \approx 2^{35.3}$	Linearization	Ours

rpp: random-prefix preimage, GE: Gaussian eliminations, $\star$: uses increased 2^{192} bound [LM22]

Related Work on outputs $\geq$ 256 bit

Output	Setting	#R	Complexity	Strategy	Reference
256 bit	preimage*	3	2^{163}	Meet-in-the-Middle	[Don+24]
	preimage*	3	2^{184}	Differential-Linear	[Niu+24]
	preimage*	4	2^{185}	Meet-in-the-Middle	[Don+24]
	preimage*	4	2^{189}	Differential-Linear	[Niu+24]
	preimage*	5	2^{191}	Meet-in-the-Middle	[Don+24]
any (n-bit)	2nd/rpp preimage	*	2^{128}	Generic	
	2nd preimage	1	2^{64} GE	Linearization	Ours
	preimage	1	$2^{64}\ \mathrm{GE} + 2^{n-128}$	Linearization	Ours
	rpp preimage	1	$2^{29.7}\ \mathrm{GE} \approx 2^{35.3}$	Linearization	Ours

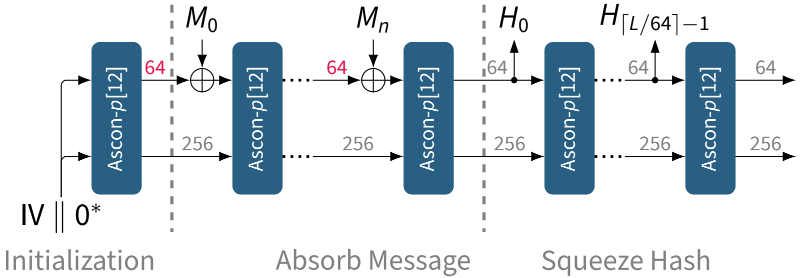
rpp: random-prefix preimage, GE: Gaussian eliminations, $\star$: uses increased 2^{192} bound [LM22]

Related Work on outputs $\geq$ 256 bit

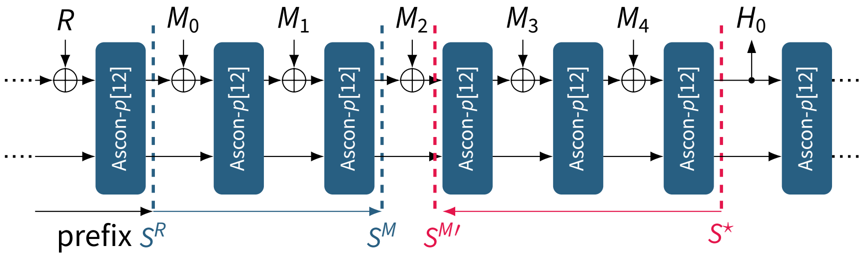
Output	Setting	#R	Complexity	Strategy	Reference
256 bit	preimage*	3	2^{163}	Meet-in-the-Middle	[Don+24]
	preimage*	3	2^{184}	Differential-Linear	[Niu+24]
	preimage*	4	2^{185}	Meet-in-the-Middle	[Don+24]
	preimage*	4	2^{189}	Differential-Linear	[Niu+24]
	preimage*	5	2^{191}	Meet-in-the-Middle	[Don+24]
any (n-bit)	2^{nd}/rpp preimage	$\star$	2^{128}	Generic	
	2^{nd} preimage	1	$2^{64}\,\mathrm{GE}$	Linearization	Ours
	preimage	1	$2^{64}\,\mathrm{GE} + 2^{n-128}$	Linearization	Ours
	rpp preimage	1	$2^{29.7}\,\mathrm{GE} \approx 2^{35.3}$	Linearization	Ours

rpp: random-prefix preimage, GE: Gaussian eliminations, $\star$: uses increased 2^{192} bound [LM22]

Ascon-XOF – Mode



Generic random-prefix preimage attack (2^{128})



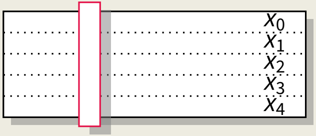
1. Choose $S^\star$, and absorb prefix $R \parallel Q$
2. Generate 2^{128} M_0, M_1
3. Generate 2^{128} M_3, M_4

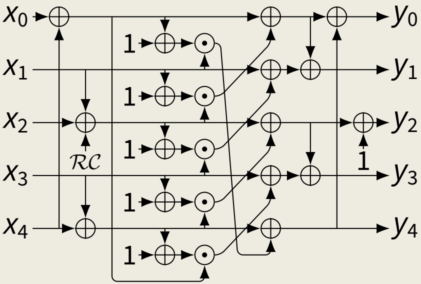
Find match in the 256 capacity bits

Can we exploit control of h
for a **dedicated attack**?

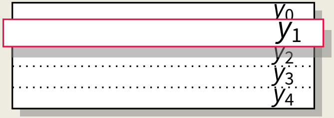
Ascon-XOF – Round Function

S-box layer





Linear layer



$$y_0 \oplus (y_0 \ggg 19) \oplus (y_0 \ggg 28) \rightarrow x_0$$
$$y_1 \oplus (y_1 \ggg 61) \oplus (y_1 \ggg 39) \rightarrow x_1$$
$$y_2 \oplus (y_2 \ggg 1) \oplus (y_2 \ggg 6) \rightarrow x_2$$
$$y_3 \oplus (y_3 \ggg 10) \oplus (y_3 \ggg 17) \rightarrow x_3$$
$$y_4 \oplus (y_4 \ggg 7) \oplus (y_4 \ggg 41) \rightarrow x_4$$

Dedicated Attack

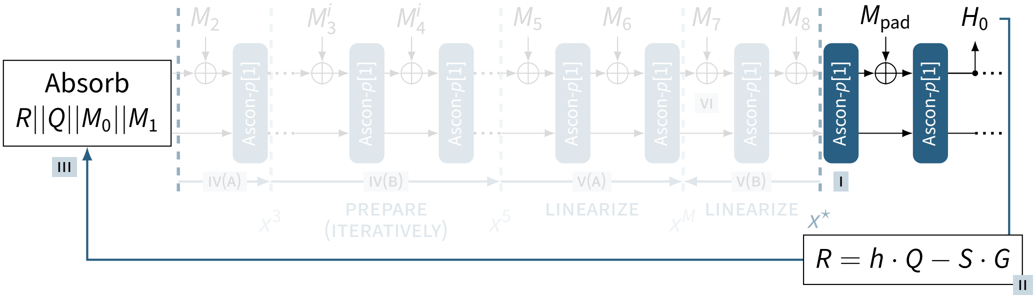


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Dedicated Attack

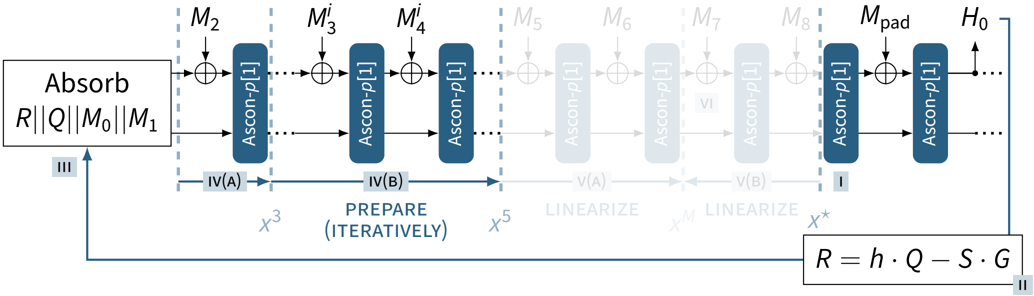


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Dedicated Attack

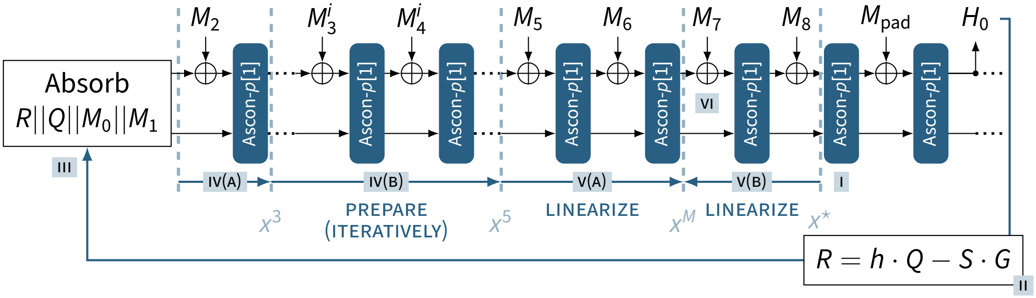


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Dedicated Attack

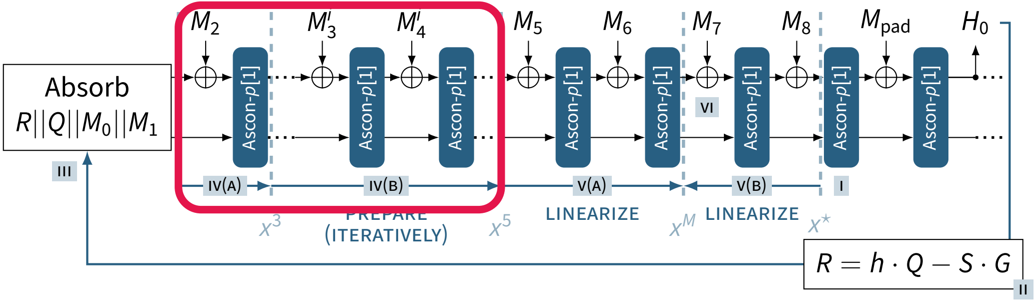
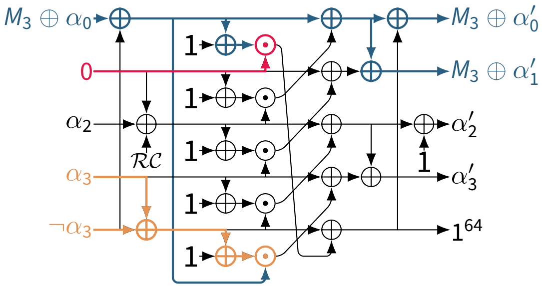


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

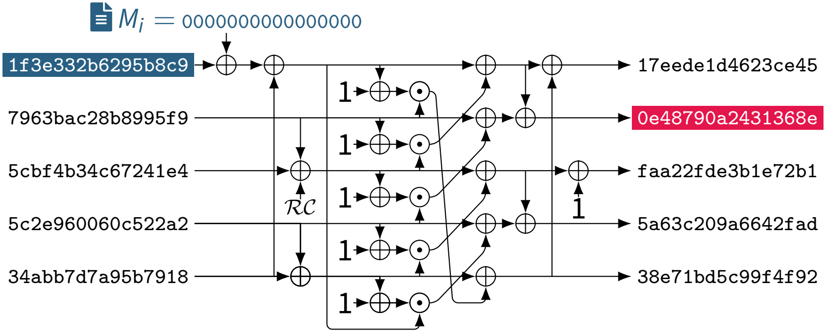
Initial Conditions: Motivation



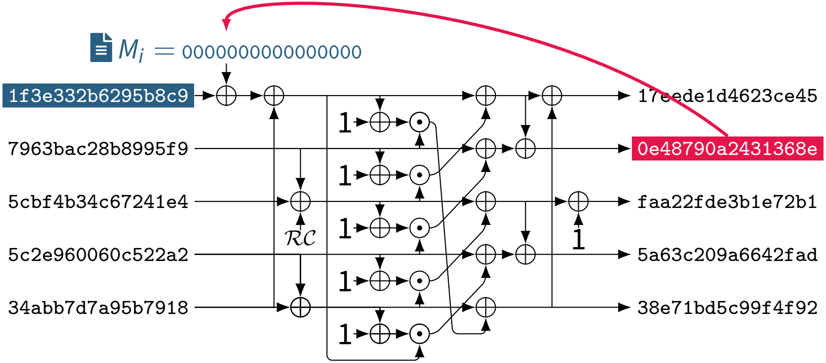
$x_1 = 0,$ $x_3 = \neg x_4$

Message Diffusion

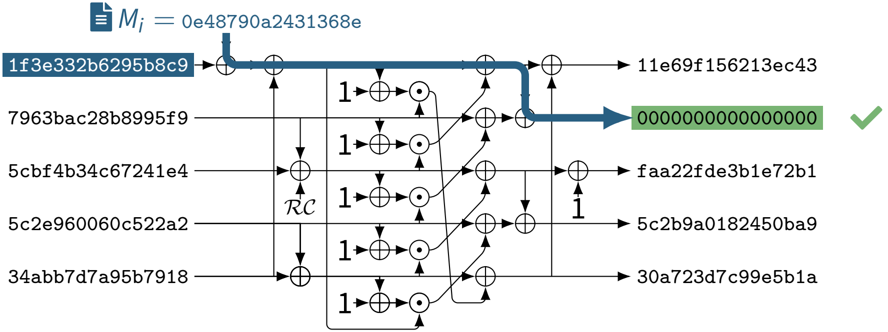
Initial Conditions: $x_1 = 0$



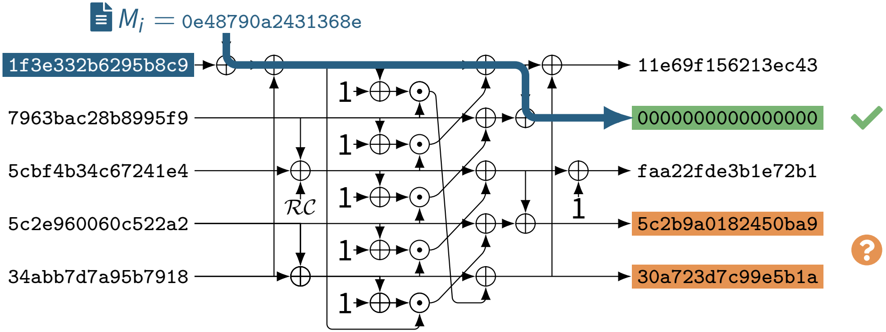
Initial Conditions: $x_1 = 0$



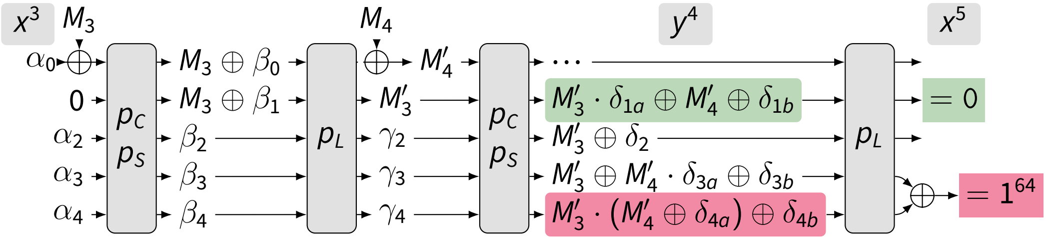
Initial Conditions: $x_1 = 0$



Initial Conditions: $x_1 = 0$



Initial Conditions: $x_3 \oplus x_4 = 1^{64}$



- maintain $x_1 = 0$
- increase Hamming weight of $x_3 \oplus x_4$
- repeat until $x_3 \oplus x_4 = 1^{64}$ ($\approx$ 8 times)

Initial Conditions: Example

M_i	x_0	x_1	x_2	x_3	x_4	Comment
	9b1e5494e934d681	4bc3a01e333751d2	ae65396c6b34b81a	3c7fd4a4d56a4db3	1a5c464906c5976d	Ascon-Hash256 IV
685903260457ea53	7970a54f7a956205	e9a17ea5020eaf64	d414b335e550a038	ce2b77bea3c33fae	6c4b2bc95f2b8612	randomize
af1cb14c5d612ec1	916deea6ca0c72b3	0a6160777409cca1	70870fd1dfdfed47	085e826fa9975606	f358a8241564bdab	randomize
18caa93c20b674b7	387d44f426400f5d	0000000000000000	20cd0294856ce9d9	e661109dca462fe5	a5ae5f097f76c9e7	$x_1 = 0$
02034e8424004000	7b5fd54cc3442713	79b3c813ba6959a9	8e981997454cb201	eb022c6a4ca098c2	8d1249789d7b6069	
01f669c9a1d2ac91	f8a9dabce0271af2	0000000000000000	5e954963f707cbff	93940c511e30a8b1	b0a4a03ea5cf574c	$\mathrm{HW}(x_3 \oplus x_4) = 44$
00002c0313ccfde0	08e2144b56261f9d	ce0ae2d036a6b480	5fea35b137080f08	ade02e27b0350c3a	eeab32c89a199854	
15a3f9744b803c32	00e92efb764370dc	0000000000000000	daae749cd5a45915	c38ee7010cc0932f	e458bcdef33f6cd0	$\mathrm{HW}(x_3 \oplus x_4) = 53$
00844284fede47b6	39714d7a7ada4901	ba0e93d81e0db0a2	45855c2dc2b33c60	867ea1897720a565	36660897bf93eb2d	
f0ecba1838f63fe3	ae7c216a92878d36	0000000000000000	d08ff4f28c946d24	67e6360c2c81f91f	ea19c9f3d37e06e0	$\mathrm{HW}(x_3 \oplus x_4) = 60$
84651c261793f7ac	f0b342b5edf85168	0b62ef4c276c5728	3640848a0a9f0819	1227e8b7b5b1f1c1	8d1bffffffc6ffff	
40cfd2b5adc1516c	3ad693c7c7a2c56b	0000000000000000	051854c4920ed35c	fbdc2a1e3a528196	4423d5e1c5ad7e69	$\mathrm{HW}(x_3 \oplus x_4) = 63$
12ea247b8325fbbc	76c4d6d49a7b27a3	0bad11bcb3f78c5c	065d504c96d30a8b	16bd4fa276b7e65b	bf7fffffffdfffff	
d44b76d580c036c3	e52f8615a999649f	0000000000000000	eb86bd4233149d27	b7aa2d4e2d66f7f0	4c55d2b1d299080f	$\mathrm{HW}(x_3 \oplus x_4) = 63$
701b354e4cb8b2d3	0bb9f7786ebb76c5	58a0a5c730f2b3ff	034a6a4a21329678	01f5d6d1c2c6be3e	fbf7ffffffffdffff	
abb1f71862b2a4f7	e12feb45172b0c5b	0000000000000000	b864c997c9c9cc64	d5174e2ae7415ebf	aae8b1d518bea140	$\mathrm{HW}(x_3 \oplus x_4) = 63$
68cf246ff07f4737	e2614bcbe9faa9bb	ada27d29b6866f6f	1a7f8a7d4d76f992	aa62f368370fa547	7effffffffbfffff	
2b434b61edbc88fa	7c76c3ee03ffe1f7	0000000000000000	56927a3259d256c6	4f1fd664b03eb777	70e0299b4fc14888	$\mathrm{HW}(x_3 \oplus x_4) = 62$
1a442a5a23dc2146	07643ea2da87265e	b008f46aaa6d1b88	3f2d4cccc6bd2280	a15eca1a17a2f10e	3e7ffffff9fffff	
a66c7fabd8c72266	40f931a2dc9d06e2	0000000000000000	ec3d2628b747674a	455dd67eb2e95fa1	baa229814d16a05e	$\mathrm{HW}(x_3 \oplus x_4) = 64$

Dedicated Attack

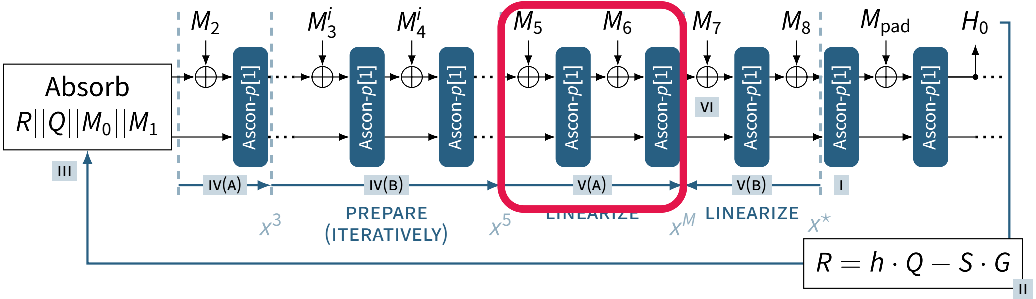
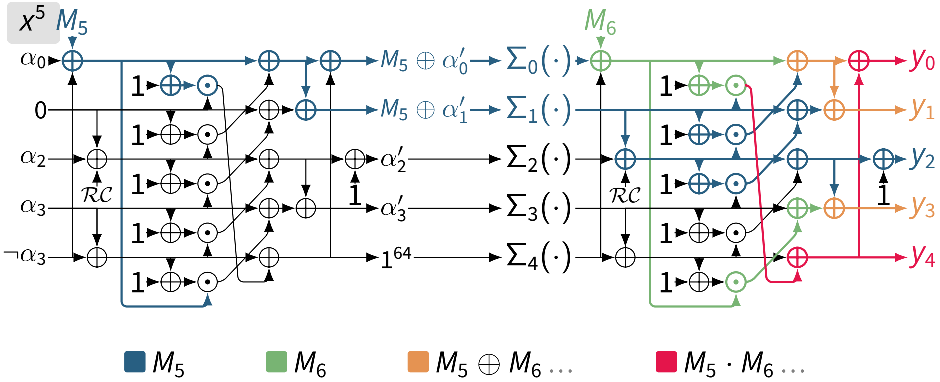
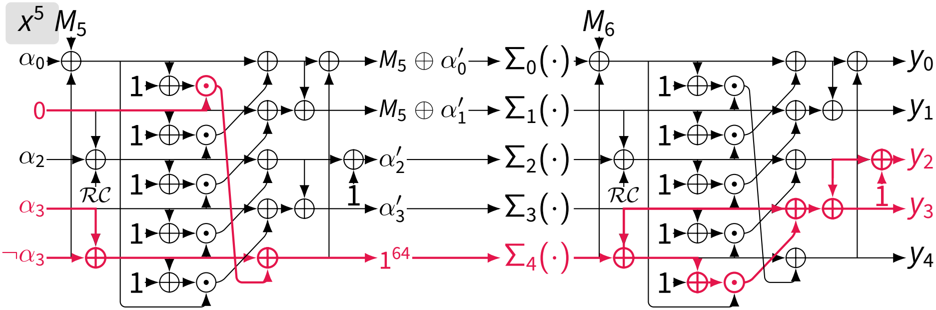


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Linearization over 2 rounds



Linearization over 2 rounds



- $y_2 \oplus y_3 = 1^{64}$ with $p = 75\,\%$ (per bit)

Dedicated Attack

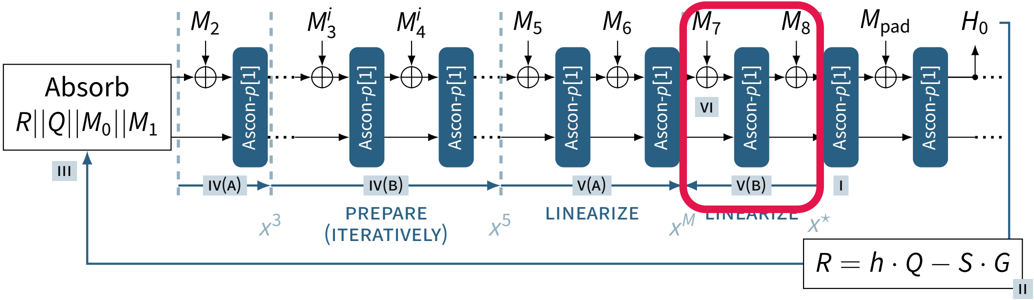


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Inverse Linearization

	x_0	x_1	x_2	x_3	x_4
$\mathcal{S}^{-1}(y \parallel 0000)$	1	y	$y+1$	0	0
$\mathcal{S}^{-1}(y \parallel 0001)$	1	$y+1$	y	1	0
$\mathcal{S}^{-1}(y \parallel 0010)$	0	y	$y+1$	1	1
$\mathcal{S}^{-1}(y \parallel 0011)$	y	$y+1$	$y+1$	0	1
$\mathcal{S}^{-1}(y \parallel 0100)$	0	0	0	y	y
$\mathcal{S}^{-1}(y \parallel 0101)$	0	$y+1$	y	0	1
$\mathcal{S}^{-1}(y \parallel 0110)$	y	1	1	$y+1$	0
$\mathcal{S}^{-1}(y \parallel 0111)$	1	y	y	1	y
$\mathcal{S}^{-1}(y \parallel 1000)$	y	$y+1$	y	1	y
$\mathcal{S}^{-1}(y \parallel 1001)$	y	y	$y+1$	1	y
$\mathcal{S}^{-1}(y \parallel 1010)$	$y+1$	$y+1$	1	0	$y+1$
$\mathcal{S}^{-1}(y \parallel 1011)$	0	y	0	0	$y+1$
$\mathcal{S}^{-1}(y \parallel 1100)$	$y+1$	1	y	y	1
$\mathcal{S}^{-1}(y \parallel 1101)$	$y+1$	y	1	0	$y+1$
$\mathcal{S}^{-1}(y \parallel 1110)$	1	0	0	$y+1$	$y+1$
$\mathcal{S}^{-1}(y \parallel 1111)$	$y+1$	$y+1$	$y+1$	1	0

Dedicated Attack

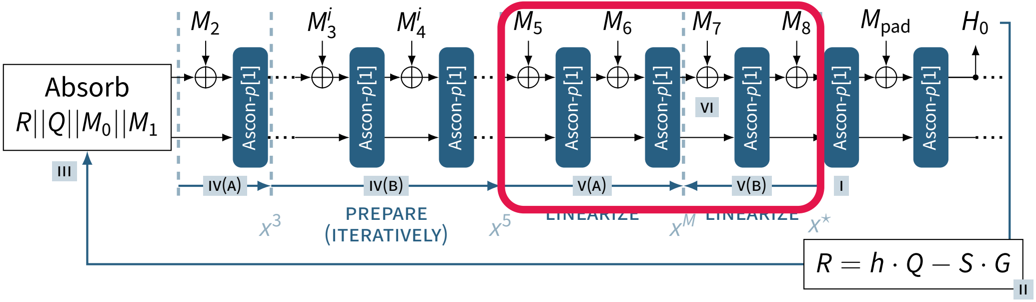
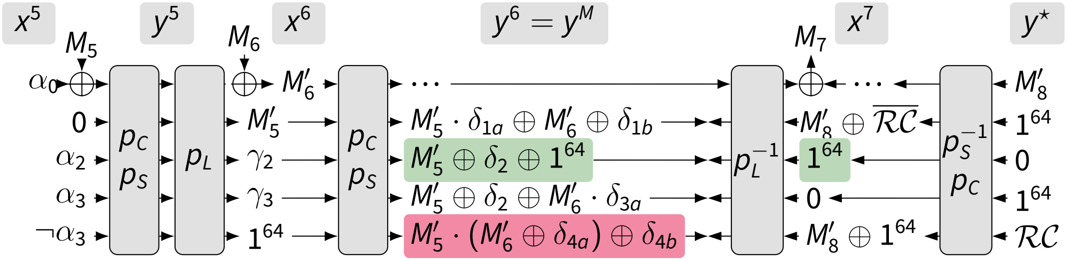


Figure: Finding forgeries with random-prefix preimages on Ascon-XOF128.

Linearization Procedure



- 192 variables, 256 constraints
- ≈ 32 free constraints due to $y_2 \oplus y_3 \approx 1^{64}$
- repeat $\approx 2^{26.6}$ times for solution

Conclusion

- Random-prefix preimage attack on 1-round Ascon in $2^{29.7}$ Gaussian eliminations, implementation takes $\approx$3 core-hours
- First analysis of **sponge-based hash function** in random-prefix preimage setting
- Using Ascon-XOF in Ed25519 maintains the desired security level

`marcel.nageler@tugraz.at`

Acknowledgments

This paper was supported by the European Research Council Starting Grant KEYLESS

Bibliography I

[Ber+11] Daniel J. Bernstein et al. **High-Speed High-Security Signatures.** CHES 2011. Vol. 6917. LNCS. Springer, 2011, pp. 124–142. DOI: 10.1007/978-3-642-23951-9_9.

[Dob+21] Christoph Dobraunig et al. **Ascon v1.2: Lightweight Authenticated Encryption and Hashing.** Journal of Cryptology 34.3 (2021), p. 33. DOI: 10.1007/s00145-021-09398-9.

[Don+24] Xiaoyang Dong et al. **Generic MitM Attack Frameworks on Sponge Constructions.** CRYPTO 2024. Vol. 14923. LNCS. Springer, 2024, pp. 3–37. DOI: 10.1007/978-3-031-68385-5_1.

[LM22] Charlotte Lefevre and Bart Mennink. **Tight Preimage Resistance of the Sponge Construction.** CRYPTO 2022. Vol. 13510. LNCS. Springer, 2022, pp. 185–204. DOI: 10.1007/978-3-031-15985-5_7.

[Nat23] National Institute of Standards and Technology. **Digital Signature Standard (DSS).** Tech. rep. Federal Information Processing Standards Publications (FIPS) 186-5. 2023. DOI: 10.6028/nist.fips.186-5.

Bibliography II

[Nat24] National Institute of Standards and Technology. **Ascon-Based Lightweight Cryptography Standards for Constrained Devices: Authenticated Encryption, Hash, and Extendable Output Functions.** Tech. rep. NIST Special Publication (SP) 800-232 (Initial Public Draft). 2024. DOI: 10.6028/nist.sp.800-232.ipd.

[Niu+24] Zhongfeng Niu et al. **Speeding Up Preimage and Key-Recovery Attacks with Highly Biased Differential-Linear Approximations.** CRYPTO 2024. Vol. 14923. LNCS. Springer, 2024, pp. 73–104. DOI: 10.1007/978-3-031-68385-5_3.