Automatic Preimage Attack Framework on Ascon Using a Linearize-and-Guess Approach

Huina Li[2,1], Le He[1], Shiyao Chen[3,1], Jian Guo[1], Weidong Qiu[2]

1 Nanyang Technological University, Singapore, Singapore

2 Shanghai Jiao Tong University, Shanghai, China

3 Strategic Centre for Research in Privacy-Preserving Technologies and Systems, Nanyang Technological University, Singapore, Singapore

March 26, 2024

Table of contents

- Designed by Dobraunig, Eichlseder, Mendel, Schläffer
- A family of lightweight cryptographic algorithms for AEAD(ASCON-128/ASCON-128a) and hashing functionality(ASCON-HASH/ASCON-XOF)
- NIST Lightweight Cryptography Standard (in February 2023)

Sponge[1]-based Hashing

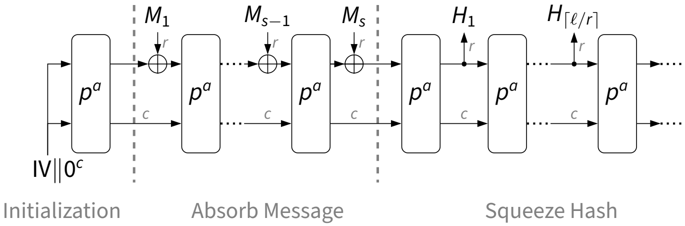


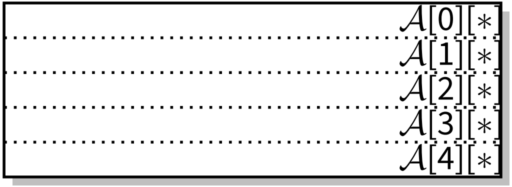
Figure: Hashing mode $\mathcal{X}_{h,r,a}$ in Ascon-Hash, Ascon-Xof

Name	Algorithm	Capacity c	Rate r
Ascon-Hash	$\mathcal{X}_{256,64,12}$	256	64
Ascon-Xof	$\mathcal{X}_{0,64,12}$ with arbitrary output length	256	64

[1] Guido Bertoni et al. "Sponge functions". In: *ECRYPT hash workshop*. Vol. 2007. 9. 2007

Ascon Permutation

- p^a permutation is the underlying permutaion of Ascon-Hash and Ascon-Xof
- Let $\mathcal{A}[y][x], 0 \leq y < 5, 0 \leq x < 64$ denote the bit in the 320-bit state $\mathcal{A}$.
- $\mathcal{A}$ is split into five 64-bit registers words $\mathcal{A}[y][*]$.



- 12 rounds
- Each round consists of 3 operations, i.e., $R := p_L \circ p_S \circ p_C$

Round Function of Ascon

- p_C: $\mathcal{A}[2][*] \leftarrow \mathcal{A}[2][*] \oplus c_r$. No impacts on preimage attacks.
- p_S: update the state $\mathcal{A}$ with 64 parallel applications of the 5-bit S-box. The only Non-Linear operation.

  Let $\mathcal{A}[*][i] = (a_{0i}, a_{1i}, \cdots, a_{4i}), 0 \leq i < 64$ denote the inputs in $i-$th S-box and the outputs are $(b_{0i}, b_{1i}, \cdots, b_{i4})$. The ANF of the S-box layer p_S,

$$\begin{aligned}
b_{0,i} &= a_{4,i}a_{1,i} \oplus a_{3,i} \oplus a_{2,i}a_{1,i} \oplus a_{2,i} \oplus a_{1,i}a_{0,i} \oplus a_{1,i} \oplus a_{0,i} \\
b_{1,i} &= a_{4,i} \oplus a_{3,i}a_{2,i} \oplus a_{3,i}a_{1,i} \oplus a_{3,i} \oplus a_{2,i}a_{1,i} \oplus a_{2,i} \oplus a_{1,i} \oplus a_{0,i} \\
b_{2,i} &= a_{4,i}a_{3,i} \oplus a_{4,i} \oplus a_{2,i} \oplus a_{1,i} \oplus 1 \\
b_{3,i} &= a_{4,i}a_{0,i} \oplus a_{4,i} \oplus a_{3,i}a_{0,i} \oplus a_{3,i} \oplus a_{2,i} \oplus a_{1,i} \oplus a_{0,i} \\
b_{4,i} &= a_{4,i}a_{1,i} \oplus a_{4,i} \oplus a_{3,i} \oplus a_{1,i}a_{0,i} \oplus a_{1,i}
\end{aligned} \tag{1}$$

Round Function of Ascon

- p_L: provide diffusion within each 64-bit register word $\mathcal{A}[y][*]$.

$$
\begin{aligned}
\mathcal{A}[0][*] &\leftarrow \mathcal{A}[0][*] \oplus (\mathcal{A}[0][*] \ggg 19) \oplus (\mathcal{A}[0][*] \ggg 28) \\
\mathcal{A}[1][*] &\leftarrow \mathcal{A}[1][*] \oplus (\mathcal{A}[1][*] \ggg 61) \oplus (\mathcal{A}[1][*] \ggg 39) \\
\mathcal{A}[2][*] &\leftarrow \mathcal{A}[2][*] \oplus (\mathcal{A}[2][*] \ggg 1) \oplus (\mathcal{A}[2][*] \ggg 6) \\
\mathcal{A}[3][*] &\leftarrow \mathcal{A}[3][*] \oplus (\mathcal{A}[3][*] \ggg 10) \oplus (\mathcal{A}[3][*] \ggg 17) \\
\mathcal{A}[4][*] &\leftarrow \mathcal{A}[4][*] \oplus (\mathcal{A}[4][*] \ggg 7) \oplus (\mathcal{A}[4][*] \ggg 41)
\end{aligned}
\tag{2}
$$

Preimage Resistance

What we are looking at...?

Given H, find a preimage M such that $\text{Hash}(M, IV) = H$ equals to solving an algebraic polynomial system.

For one hash function with h-bit hash value.

- Brute-force: 2^h time complexity in average
- Preimage Attack: Find a technique that is faster than brute-force.

Summary of Techniques in Preimage Attack against Sponge-based Hashing

Target	Technique	Max. Rounds	Reference
Keccak challenges	SAT	2	[MS13]
Keccak-224/256	Linear structure	4/4	[GLS16]
Keccak-224/256	Linear structure + Allocating model(Lin.s.A.m)	4/4	[LS19]
Keccak-224/256	Lin.s.A.m + Partial linearization	4/4	[HLY21]
Keccak-384/512	Non-linear structure	4/3	[Raj19]
Keccak-384/512	Non-Lin.s.A.m + Relinearization	4/3	[LIMY20]
Keccak-384/512	Non-Lin.s.A.m + Relinearization + Extra linear dependence	3/3	[HLY23]
Keccak-512	MitM + Linear structure + MILP	4	[QHDYW23]
Keccak-384/512	MitM + Weak-diffusion structure + MILP	4	[QZHDW23]
Ascon-Xof	Algebraic(without c_r and IV)	2	[DEMS19]
Ascon-Xof	MitM + Weak-diffusion structure + MILP	4	[QZHDW23]
Ascon-Xof	MitM + Weak-diffusion structure + SAT	4	[DDLS24]
Ascon-Xof	Non-Lin.s.A.m + SAT	4	Our

We extend the preimage attack framework on Keccak to Ascon with the help of SAT tools.

Linear Structure[2]

- Given a 1.5-round linear structure (after 2-round entire linearization) by manually designing.

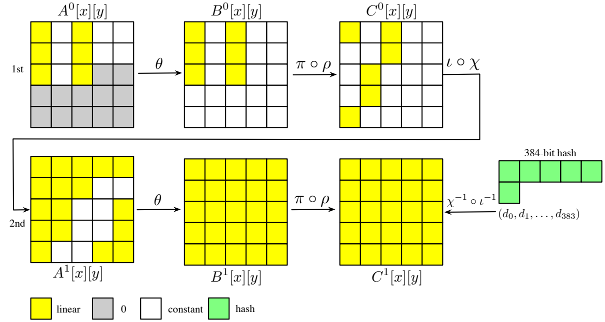


Figure: 1.5-round linear structure with 256 degrees of freedom used in preimage attack on 2-round Keccak-384

[2] Jian Guo, Meicheng Liu, and Ling Song. "Linear structures: applications to cryptanalysis of round-reduced Keccak". In: *ASIACRYPT 2016*. Springer. 2016, pp. 249–274

Main Idea of Linear Structure

- Construct a system of m linear equations leaked by the hash value in n free variable bits, assume $n \geq m$, then the system has a non-trivial solution.
- For one hash function with h-bit hash value.

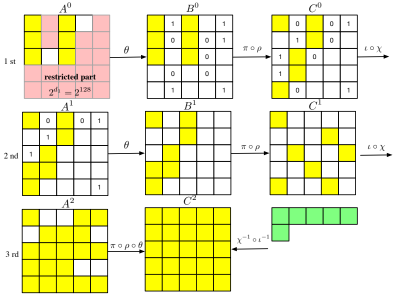
$$\begin{cases} f_0(x_0, x_1, \cdots, x_{n-1}) \oplus c_0 = h_0 \\ f_1(x_0, x_1, \cdots, x_{n-1}) \oplus c_1 = h_1 \\ \cdots \\ f_{m-1}(x_0, x_1, \cdots, x_{n-1}) \oplus c_{n-1} = h_{m-1} \end{cases} \quad (3)$$

  - total gain: 2^m
  - matching probability of one guess: 2^{m-h}
  - total complexity of finding a preimage: 2^{h-m} guesses

Linear Structure with the Allocating Model[34]

- Extend to a longer 2.5-round linear structure via an allocating model.
    1. Find an inner part that satisfies certain conditions for the initial state of the linear structure, 2^{d}_{1}.
    2. Construct the algebraic systems according to the linear structure and then solve these systems for finding a multi-block preimage, 2^{h-m}
- Search complexity of finding this restricted inner part:$2^{d_1} = 2^{128}$.
- $n = 64, n \geq m \Rightarrow m = 64$
- Random space of linear structure: $2^{d_r} = 2^{256}$
- Final complexity:

$$\max(2^{d_1} \times 2^{h-m-d_r},\ 2^{h-m})$$



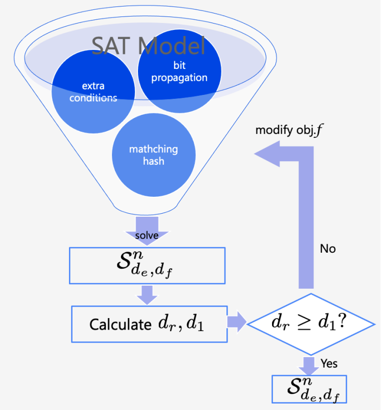
[3]Ting Li and Yao Sun. “Preimage attacks on round-reduced Keccak-224/256 via an allocating approach”. In: *EUROCRYPT 2019*

Table of contents

Hard to Design a Structure manually

- small rate
- the linear duffision layer is highly flexible (independently shift each row)
- the non-linear layer is more complex than KECCAK

Overview of Framework



1. Optimal Structures $\mathcal{S}^n_{d_e,d_f}$ Search Stage
2. Validity of Optimal Structures Verification Stage: $d_r \geq d_1$ holds?
   - Construct a system of d_e linear equations in d_f free variable bits ($d_f \geq d_e$), then the system has a non-trivial solution.
3. Final Complexity Computation Stage:

$$2^{h-d_e}$$

Automatic Optimal Structure Search Model

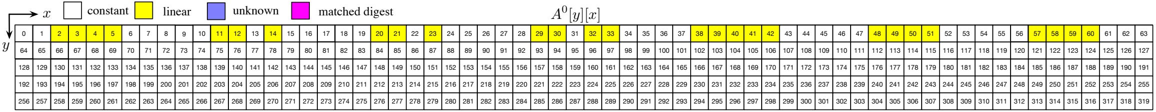
$$A^0_{x,y} \xrightarrow{p_S \circ p_C} B^0_{x,y} \xrightarrow{p_L} A^1_{x,y} \xrightarrow{p_S \circ p_C} B^1_{x,y} \xrightarrow{p_L} A^2_{x,y} \xrightarrow{p_S \circ p_C} B^2_{x,y} \xrightarrow{p_L} A^3_{x,y} \xrightarrow{p_S \circ p_C} B^3_{x,y} \xleftarrow{p_L^{-1}} A^4_{x,y}$$

3-round Structure —> 1-round LC + 1-round LCQ + 1-round LQ

Figure: 3-round Quartic Structure used in 4-round Preimage Attack

- Modeling the bit propagation
  - Modeling the Substitution Layer p_S
  - Modeling the Linear Layer p_L
- Modeling the matching hash bit
- Modeling extra condition
  - Modeling the initial state
  - Modeling the objective functions

Modeling the Initial State



- Separate all state bits into three types: linear bits (v), constant bits(c), unknown bits(q).
- In the initial state A^0:
  - constant bits are fixed constants, including the padding bit
  - linear bits only exist in the outer part

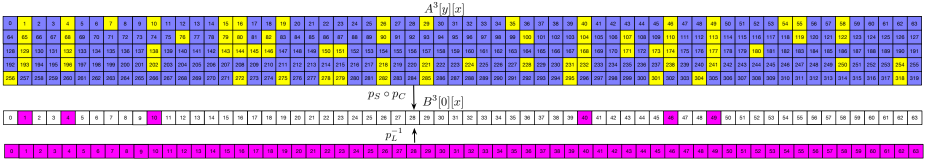


Figure: The propagation of the last round in 4-round quartic structure

- For 5-bit inputs of each sbox $(a_0, a_1, \ldots, a_4)$, the output bit b_0 is known(recovered from hash bit by p_L^{-1})

$$b_0 = a_4 a_1 \oplus a_3 \oplus a_2 a_1 \oplus a_2 \oplus a_1 a_0 \oplus a_1 \oplus a_0 \tag{4}$$

Generate Guess Linear Equations leaked by the Hash Value

Observation 1

If the algebraic degrees of a_3, a_2, a_0 are at most 1, then the guess linear equation $a_3 \oplus a_2 \oplus a_0 = b_0$ holds with a probability of $\frac{3}{4}$ when 5 input bits are uniformly distributed.

$$\begin{aligned} b_0 &= a_4 a_1 \oplus a_3 \oplus a_2 a_1 \oplus a_2 \oplus a_1 a_0 \oplus a_1 \oplus a_0 \\ &= (a_4 \oplus a_2 \oplus a_0 \oplus 1) a_1 \oplus a_3 \oplus a_2 \oplus a_0 \end{aligned}$$

Note that the quadratic term $(a_4 \oplus a_2 \oplus a_0 \oplus 1)a_1 = 0$ holds in fact with a probability of $\frac{3}{4}$. Thus, one linear equation $a_3 \oplus a_2 \oplus a_0 = b_0$ can be obtained by excluding the quadratic term, which can still bring a gain of $\frac{3}{4}/\frac{1}{2} \approx 2^{0.585}$.

Modeling the Matching Hash Bit($A^{n-1} \rightarrow B^{n-1}[0]$)

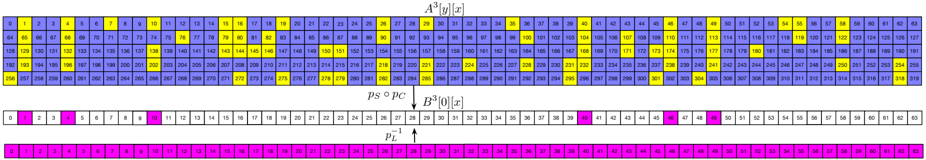


Figure: The propagation of the last round in 4-round quartic structure

For each column (sbox), if $A^2[0][i], A^2[2][i], A^2[3][i]$ $(0 \leq i < 64)$, are all linear bits or constant bits, we say there is a 1-bit hash matching, and one guess linear equation is constructed with a probability of $\frac{3}{4}$.

Additionally, we introduce 64 variables, denoted by $E_i, 0 \leq i < 64$ for $A^2[\star][i]$ to indicate which column matches successfully, if $E_i = 1$ means there is a 1-bit hash matching; otherwise, $E_i = 0$.

Objective Function

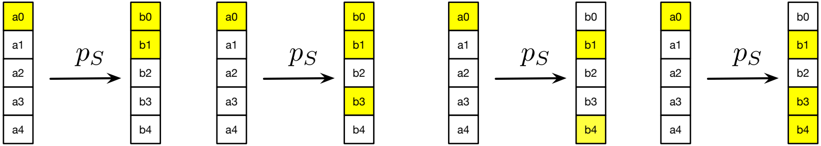
- our objective is to maximize the number of guess linear equations d_e

where d_f is the number of free linear bits(degrees of freedom).

$$\begin{cases} d_f = \sum A^0_{0,x} \\ Maximize: \quad d_e = \sum E_x \end{cases} \tag{5}$$

Diffusion of Linear Bits through the first $p_S(A^0 \rightarrow B^0)$

- Adding some specific conditions imposed on the inputs of sbox that can significantly control the diffusion of linear bits.



$a_1 = 0; a_3 + a_4 = 1$ $a_1 = 0; a_3 + a_4 = 0$ $a_1 = 1; a_3 + a_4 = 1$ $a_1 = 1; a_3 + a_4 = 0$

- $b_2 = a_4 a_3 \oplus a_4 \oplus a_2 \oplus a_1 \oplus 1$ must be a constant bit

Diffusion of Linear Bits through the second $p_S(A^1 \to B^1)$

- Due to the independent calculation of p_L within each row, the third input bit a_2 of the second p_S must be a constant bit.

Inputs	Outputs	Condition	Inputs	Outputs	Condition
ccccc	ccccc	-	vcccc	vvccc	$a_1 = 0; a_3 + a_4 = 1$
vcccc	cvccv	$a_1 = 0; a_3 + a_4 = 1$	cvccc	ccvvc	$a_2 = 0; a_3 = 1; a_0 + a_4 = 1$
ccccv	cvccv	$a_0 = 1; a_1 = 0; a_3 = 1$	ccccv	vvccc	$a_0 = 1; a_1 = 1; a_3 = 1$
vvccc	qvvvq	-	cvccv	qvvvq	-

Diffusion of Linear Bits through the third $p_S(A^2 \rightarrow B^2)$

- Due to the independent calculation of p_L within each row, unknown bits ('q') only exist in the first row and the last row of A^2.
- Suppose one 5-bit inputs of i-th sbox $A^2[*][i] = (a_0, a_1, a_2, a_3, a_4)$, the outputs $B^2[*][i] = (b_0, b_1, b_2, b_3, b_4)$

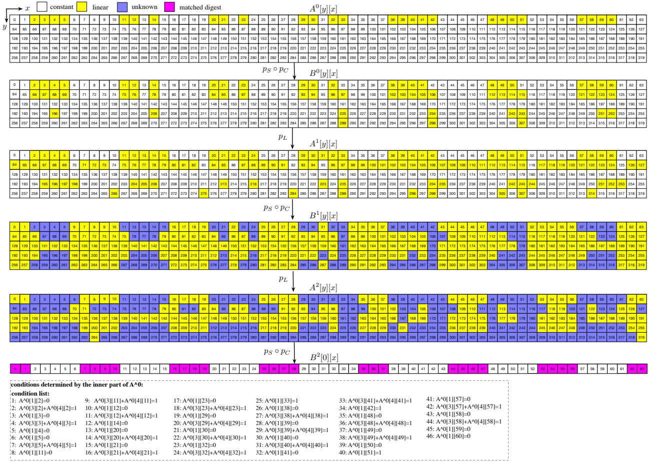
$$b_2 = a_4 a_3 \oplus a_4 \oplus a_2 \oplus a_1 \oplus 1$$
$$b_0 = a_4 a_1 \oplus a_3 \oplus a_2 a_1 \oplus a_2 \oplus a_1 a_0 \oplus a_1 \oplus a_0$$

  if any single unknown bit is among a_1, a_2, or a_3, then one of b_3,b_2,b_0 must be an unknown bit.
- Due to the independent calculation of p_L within each row, unknown bits ('q') must exist in the first row, the third row and the last row of A^3. (Too Bad! we miss at least 1-bit hash matching)

Table of contents

- Random space: $2^{d_r} = 2^{36}$
- $d_r > d_1$ holds
- Complexity(guesses):

$$2^{128-0.585d_e} = 2^{112.205}$$



Our Preimage Attacks on Ascon-Xof

Table 1: Summary of preimage attacks on Ascon-Xof. Hash: the length of the digest in bits. Size: the number of linear equations leaked by the hash value. Guesses: the number of required solutions. Solving Time: the average complexity of obtaining a single solution.

Round	Hash	Size	Guesses	Solving Time†	Final Time	Memory	Reference
2	64	25	2^{39}	$2^{-0.04}$	2^{39}	-	[DEMS19]
		64	$2^{27.56}$	2^{4}	$2^{31.56}$	-	Section 3
3	128	-	-	-	$2^{120.58}$	2^{39}	[QHD+23]
		-	-	-	$2^{114.53}$	2^{30}	[QZH+23]
		27	$2^{112.205}$	$2^{-0.29}$	$2^{112.205}$	-	Section 5
4	128	-	-	-	$2^{126.4}$	2^{45}	[QHD+23]
		-	-	-	$2^{124.67}$	2^{50}	[QZH+23]
		6	$2^{124.49}$	$2^{-1.01}$	$2^{124.49}$	-	Section 6

†Here 'Solving Time' is determined by the ratio of the number of bit operations required for one Gaussian Elimination turn to the number of bit operations in round-reduced Ascon.

Table of contents

Bibliography I

- Bertoni, Guido et al. "Sponge functions". In: *ECRYPT hash workshop*. Vol. 2007. 9. 2007.
- Guo, Jian, Meicheng Liu, and Ling Song. "Linear structures: applications to cryptanalysis of round-reduced Keccak". In: *ASIACRYPT 2016*. Springer. 2016, pp. 249–274.
- He, Le et al. *Improved Preimage Attacks on Round-Reduced Keccak-384/512*. Cryptology ePrint Archive, Paper 2022/788. 2022.
- Li, Ting and Yao Sun. "Preimage attacks on round-reduced Keccak-224/256 via an allocating approach". In: *EUROCRYPT 2019*.