ASCON

Authenticated Encryption
CAESAR and NIST LWC Winner

Rithvika Pervala

Department of EECS
Indian Institute of Technology Bhilai

May 2, 2023

Outline

T1 - Construction
○●○○○○○○○○

T2 - Cryptanalysis
○○○○○○○○○○○○

T3 - Verilog
○○○○○○

T4 - Automated Analysis
○○○○○○

Conclusion
○○○

Abstract

Abstract. Authenticated encryption satisfies the basic need for authenticity and confidentiality in our information infrastructure. In this paper, we provide the specification of Ascon-128 and Ascon-128a. Both authenticated encryption algorithms provide efficient authenticated encryption on resource-constrained devices and on high-end CPUs. Furthermore, they have been selected as the "primary choice" for lightweight authenticated encryption in the final portfolio of the CAESAR competition. In addition, we specify the hash function Ascon-Hash, and the extendable output function Ascon-Xof. Moreover, we complement the specification by providing a detailed overview of existing cryptanalysis and implementation results.

Abstract

Abstract.   Authenticated encryption satisfies the basic need for authenticity and confidentiality in our information infrastructure. In this paper, we provide the specification of Ascon-128 and Ascon-128a. Both authenticated encryption algorithms provide efficient authenticated encryption on resource-constrained devices and on high-end CPUs. Furthermore, they have been selected as the "primary choice" for lightweight authenticated encryption in the final portfolio of the CAESAR competition. In addition, we specify the hash function Ascon-Hash, and the extendable output function Ascon-Xof. Moreover, we complement the specification by providing a detailed overview of existing cryptanalysis and implementation results.

Key Points

- **Authentication** - AEAD, Single Pass, Online
- **Device Versatility** - High End CPU, Small Devices
- **Primitive Options** - Encryption, Hash, XOF
- **Cryptanalysis** - Linear, Differential, Zero-Sum, Integral

Cipher Specifications

Recommended parameters for ASCON authenticated encryption

Cipher	Bit size of					Rounds	
	key	nonce	tag	rate	capacity	p^a	p^b
ASCON-128	128	128	128	64	256	12	6
ASCON-128a	128	128	128	128	192	12	8

Sponge Construction

- **State** (**s**) - The sponge operates on a state of 320 bits
- **Rate** (**r**) - Rate at which message is consumed
- **Capacity** (**c**) - Encryption, Hash, XOF
- **Tag** (**t**)- Associated with Authentication.
- **a**, **b** - Number of Rounds in **End** and **Core** Permutations

State (S)

$S = x_0 || x_1 || x_2 || x_3 || x_4$

x_0
x_1
x_2
x_3
x_4

State (S)

$S = x_0||x_1||x_2||x_3||x_4$

$S = S_r||S_c$

ASCON-128
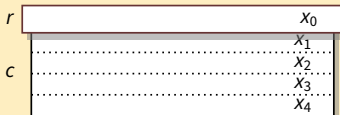
- **Rate** (**r**) - 64 Bits Block
- **Capacity** (**c**) - 256 Bits

ASCON-128a

- **Rate** (**r**) - 128 Bits Block
- **Capacity** (**c**) - 192 Bits

Authenticated Encryption
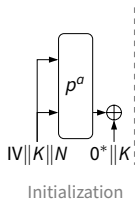
Initialization

Duplex-Sponge Construction

- **Initialization** - Initialize the 320-Bit State S with Initialization Vector IV, key K, and Nonce N. Run it through **End Permutation** p^a and inject padded K.

Authenticated Encryption
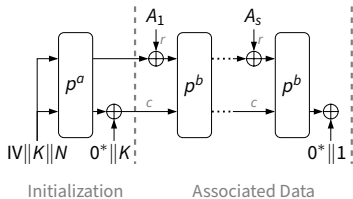
Initialization   Associated Data

Duplex-Sponge Construction

- **Initialization** - Initialize the 320-Bit State S with Initialization Vector IV, key K, and Nonce N. Run it through **End Permutation** p^a and inject padded K.
- **Associated Data Processing** - Inject S with Associated Data blocks A_i and and digest through **Core Permutation** p^b.

Authenticated Encryption
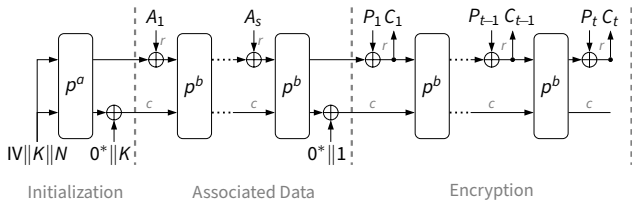
Initialization   Associated Data   Encryption

Duplex-Sponge Construction

- **Initialization** - Initialize the 320-Bit State S with Initialization Vector IV, key K, and Nonce N. Run it through **End Permutation** p^a and inject padded K.

- **Associated Data Processing** - Inject S with Associated Data blocks A_i and and digest through **Core Permutation** p^b.

- **Encryption** - Inject S with plaintext blocks P_i and extract ciphertext blocks C_i after each run of p_b.

Authenticated Encryption
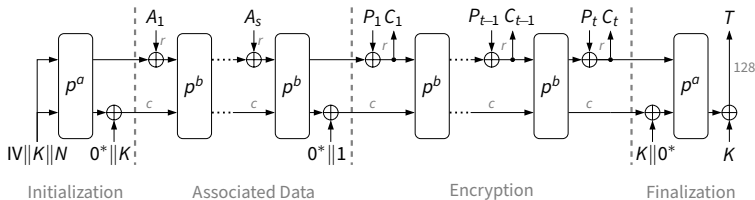

Duplex-Sponge Construction

- **Initialization** - Initialize the 320-Bit State S with Initialization Vector IV, key K, and Nonce N. Run it through **End Permutation** p^a and inject padded K.
- **Associated Data Processing** - Inject S with Associated Data blocks A_i and and digest through **Core Permutation** p^b.
- **Encryption** - Inject S with plaintext blocks P_i and extract ciphertext blocks C_i after each run of p_b.
- **Finalization** - Inject padded K again and extract the Tag T for authentication after running through p^a.

Initialization and Associated Data Processing

Initialization

$$IV_{k,r,a,b} \leftarrow k||r||a||b||0^{160-k}$$
$$S \leftarrow IV||K||V$$
$$S \leftarrow p^a(S) \oplus \left(0^{320-k}||K\right)$$

- $|IV|$ - 64, $|K|$ - 128, $|N|$ - 128
- ASCON-128 - 80400c0600000000
- ASCON-128a - 80800c0800000000

Initialization and Associated Data Processing

Initialization

$$IV_{k,r,a,b} \leftarrow k||r||a||b||0^{160-k}$$
$$S \leftarrow IV||K||V$$
$$S \leftarrow p^a(S) \oplus \left(0^{320-k}||K\right)$$

- $|IV|$ - 64, $|K|$ - 128, $|N|$ - 128
- ASCON-128 - 80400c0600000000
- ASCON-128a - 80800c0800000000

Associated Data Processing

- Padding the Associated Data A and splitting s r-Bit Blocks

  $$A_1, ..., A_s \leftarrow A||1||0^{r-1-(|A| \bmod r)}$$

- Digesting Associated Data Blocks

  $$S \leftarrow p^b\left((S_r \oplus A_i)||S_c\right), \ 1 \le i \le s$$

- Adding 1-bit Domain Separation Constant

  $$S \leftarrow S \oplus \left(0^{319}||1\right)$$

T1 - Construction
0000000●000

T2 - Cryptanalysis
000000000000

T3 - Verilog
000000

T4 - Automated Analysis
000000

Conclusion
000

Encryption and Finalisation

Encryption

- Padding the Plaintext P and splitting t r-Bit Blocks

$$P_1, \dots, P_t \leftarrow P||1||0^{r-1-(|P| \bmod r)}$$

- Injecting Plaintext and Extracting Ciphertext

$$C_i \leftarrow S_r \oplus P_i \quad 1 \leq i \leq t$$

$$S \leftarrow \begin{cases} p^b\left(C_i||S_c\right) & \text{if } 1 \leq i < t \\ C_i||S_c & \text{if } i = t \end{cases}$$

- Unpadding Last Ciphertext C_t

$$C_t' \leftarrow \lfloor C_t \rfloor_{|P| \bmod r}$$

Encryption and Finalisation

Encryption

- Padding the Plaintext P and splitting t r-Bit Blocks

$$P_1, \ldots, P_t \leftarrow P||1||0^{r-1-(|P| \bmod r)}$$

- Injecting Plaintext and Extracting Ciphertext

$$C_i \leftarrow S_r \oplus P_i \quad 1 \leq i \leq t$$

$$S \leftarrow \begin{cases} p^b\left(C_i||S_c\right) & \text{if } 1 \leq i < t \\ C_i||S_c & \text{if } i = t \end{cases}$$

- Unpadding Last Ciphertext C_t

$$C'_t \leftarrow \lfloor C_t \rfloor_{|P| \bmod r}$$

Finalization

- Adding Padded Key to State

$$S \leftarrow S \oplus \left(0^r||K||0^{c-k}\right)$$

- End Permutation

$$S \leftarrow p^a(S)$$

- Extracting Tag T from the Least Significant 128 Bits after adding Key

$$T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$$

Hash
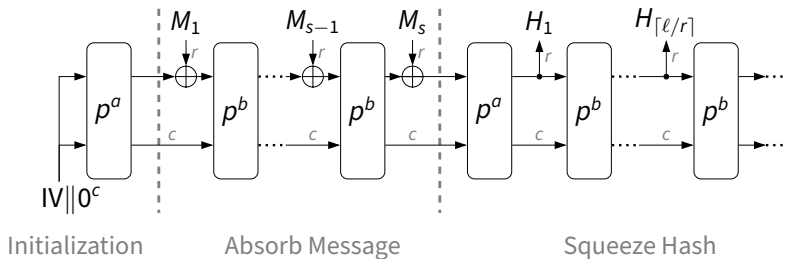
Round Constant (p_C)

Permutation ($p = p_L \circ p_S \circ p_C$)
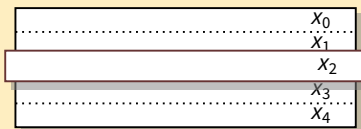
1 Byte Constant

- $x_2 := x_2 \oplus c_r$
- 12 Round Constants
- $p^a \rightarrow c_r$ and $p^b \rightarrow c_{a-b+r}$

Round Constant (p_C)

Permutation ($p = p_L \circ p_S \circ p_C$)

1 Byte Constant
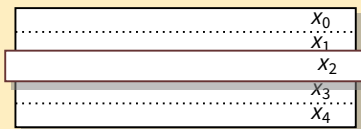
- $x_2 := x_2 \oplus c_r$
- 12 Round Constants
- $p^a \rightarrow c_r$ and $p^b \rightarrow c_{a-b+r}$

p^{12}	p^8	p^6	Constant	p^{12}	p^8	p^6	Constant
0			000000000000000000f0	6	2	0	00000000000000000096
1			000000000000000000e1	7	3	1	00000000000000000087
2			000000000000000000d2	8	4	2	00000000000000000078
3			000000000000000000c3	9	5	3	00000000000000000069
4	0		000000000000000000b4	10	6	4	0000000000000000005a
5	1		000000000000000000a5	11	7	5	0000000000000000004b

Sbox (p_S) and Linear Layer (p_L)

Permutation ($p = p_L \circ p_S \circ p_C$)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
S(x)	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17

Sbox (p_S) and Linear Layer (p_L)

Permutation ($p = p_L \circ p_S \circ p_C$)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
S(x)	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17
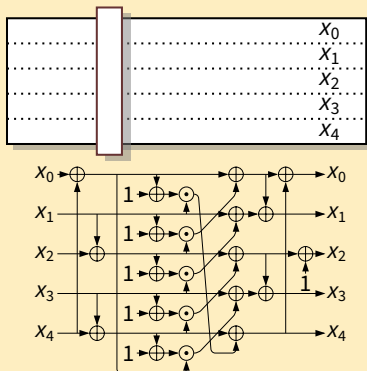
Sbox

Sbox (p_S) and Linear Layer (p_L)

Permutation ($p = p_L \circ p_S \circ p_C$)

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
S(x)	4	b	1f	14	1a	15	9	2	1b	5	8	12	1d	3	6	1c	1e	13	7	e	0	d	11	18	10	c	1	19	16	a	f	17

Sbox
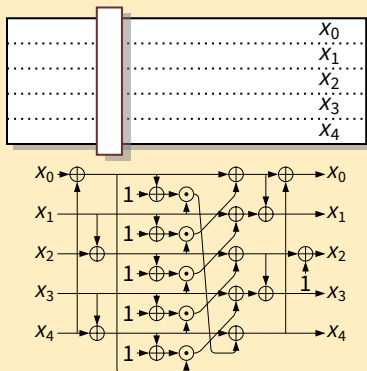
Linear Layer
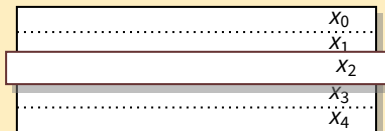
$x_0 := x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$

$x_1 := x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$

$x_2 := x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$

$x_3 := x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$

$x_4 := x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$

Outline

1. T1 - Construction

2. **T2 - Cryptanalysis**

3. T3 - Verilog

4. T4 - Automated Analysis

5. Conclusion

DDT

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	32	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	.	.	4	.	4	.	4	.	4	.	.	.	.	.	.	.	.	.	4	.	4	.	4	.	4	.
2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	.	4	.	4	.	4	.	4	.	4	.	4	.	.	4
3	.	4	.	.	.	4	.	.	.	4	.	.	.	4	.	.	.	4	.	.	.	4	.	.	.	4	.	.	.	4	.	.
4	.	.	.	.	.	.	8	.	.	.	.	.	.	.	8	.	.	.	.	.	.	.	8	.	.	.	.	.	.	.	8	.
5	.	.	.	.	.	.	.	.	.	.	4	.	4	.	4	.	.	.	.	.	.	.	.	.	4	.	4	.	4	.	4	4
6	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2
7	.	.	4	4	.	.	4	4	.	.	4	4	.	.	4	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
8	.	.	.	.	.	.	4	4	.	.	.	.	.	.	4	4	.	.	.	.	.	.	4	4	.	.	.	.	.	.	4	4
9	.	2	.	2	2	.	2	.	2	.	2	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	.
10	.	2	2	.	2	.	.	2	.	2	2	.	2	.	.	2	.	2	2	.	2	.	.	2	.	2	2	.	2	.	.	2
11	.	.	2	2	.	.	2	2	.	.	2	2	.	.	2	2	.	.	2	2	.	.	2	2	.	.	2	2	.	.	2	2
12	.	8	.	.	.	.	.	.	8	.	.	.	.	.	.	.	8	.	.	.	.	.	.	.	8	.	.	.	.	.	.	.
13	.	2	.	2	.	2	.	2	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.
14	.	4	4	.	4	.	.	4	.	.	.	.	.	.	.	.	.	4	4	.	4	.	.	4	.	.	.	.	.	.	.	.
15	.	.	.	.	.	.	.	4	4	.	4	4	.	.	.	.	.	.	.	.	.	.	.	4	4	.	4	4	.	.	.	.
16	.	.	.	.	.	.	.	.	8	.	8	.	.	.	.	.	.	.	.	.	.	.	.	.	8	.	8	.	.	.	.	.
17	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	8	.	8	.	8	.	8	.	.	.	.	.	.	.	.	.
18	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	2	.	.
19	.	.	8	.	8	.	.	.	.	8	.	8	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
20	.	.	.	4	4	4	4	.	.	.	.	.	.	.	.	.	4	4	4	4	.	.	.	.	.	.	.	.	.	.	.	.
21	.	.	.	.	4	.	4	.	4	.	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	.	.	4
22	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
23	.	.	4	.	4	.	.	.	.	4	.	4	.	.	.	.	.	.	4	.	4	.	.	.	.	.	.	4	.	4	.	.
24	.	.	.	2	2	2	2	.	.	.	.	2	2	2	2	.	.	.	.	2	2	2	2	.	.	.	.	2	2	2	2	.
25	.	.	.	4	.	4	.	.	4	.	4	.	.	.	.	.	4	.	.	.	.	.	.	.	4	.	.	.	4	.	4	.
26	.	2	2	.	.	2	2	.	2	.	.	2	2	.	.	2	.	2	2	.	2	.	.	.	.	2	2	.	.	.	.	2
27	.	.	2	2	2	2	.	.	.	2	2	2	2	.	.	.	.	2	2	2	2	.	.	.	2	2	2	2	.	.	.	.
28	.	4	.	4	.	.	.	4	.	4	.	.	.	.	.	.	.	4	.	4	.	.	.	.	.	.	4	.	4	.	.	.
29	.	.	.	4	.	4	.	.	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	.	.	.	.	4	.	4	.
30	.	.	.	.	.	.	2	2	2	2	2	2	2	2	.	.	.	.	.	.	.	.	.	.	2	2	2	2	2	2	2	2
31	.	.	4	4	4	4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	4	4	4	.	.	.	.	.	.	.	.

LAT

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	16	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	.	.	.	.	.	.	8	.	.	4	4	.	.	-4	4	.	.	.	4	4	.	.	.	4	-4	4	.	-4	.	-4	.	-4
2	.	.	.	.	.	-8	8	.	.	4	4	.	.	4	4	.	.	4	4	.	.	-4	-4	.	.	.	.	.	.	.	.	.
3	.	8	.	.	.	.	.	.	.	4	.	4	.	4	.	-4	-4	-8	.	.	.	.	.	.	4	.	4	.	4	.	-4	.
4	.	.	.	4	.	-4	.	.	.	.	4	.	.	4	-4	-4	.	4	.	.	-4	.	.	.	.	-8	.	-4	-4	.	4	-4
5	.	.	.	4	.	4	.	.	.	.	-4	.	.	.	.	-4	.	.	-4	4	.	-4	-4	4	.	-4	4	.	-8	.	.	-4
6	.	.	.	4	.	-4	.	.	.	.	.	-4	.	4	.	.	.	.	-4	-4	.	-4	-4	.	8	.	-4	-4	.	-4	4	4
7	.	.	.	-4	.	-4	.	.	.	4	4	4	.	.	-4	.	.	.	-4	.	-4	.	.	.	.	-4	.	-4	4	.	-8	4
8	.	.	.	.	.	.	.	.	.	.	4	4	.	.	-4	-4	.	.	.	.	.	.	.	.	8	-4	4	.	8	4	-4	.
9	.	.	.	.	.	.	-8	.	.	-4	.	4	.	4	.	4	.	.	4	4	.	.	-4	4	4	.	.	4	-4	.	.	4
10	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	4	4	.	.	4	4	.	8	4	-4	.	-8	4	-4	.
11	.	8	.	.	.	.	.	.	.	-4	4	.	.	-4	-4	.	8	.	.	.	.	.	.	4	.	.	-4	4	.	.	.	4
12	.	.	-8	4	-8	-4	.	.	.	.	.	4	.	-4	.	.	.	-4	.	4	.	.	.	.	.	4	.	-4	.	.	.	.
13	.	.	-4	-8	4	.	.	.	.	4	-4	-4	.	.	.	.	.	4	4	.	-4	-4	4	.	.	.	.	.	.	4	.	.
14	.	.	.	-4	8	-4	.	.	.	.	-4	.	.	-4	-4	-4	.	.	4	4	.	-4	-4	.	.	4	.	-4	.	.	.	.
15	.	.	8	-4	-8	-4	.	.	.	-4	.	.	.	.	.	-4	.	4	.	4	.	.	.	-4	.	.	.	.	.	-4	.	.
16	.	.	.	.	.	.	-8	.	.	4	.	-4	-4	.	-4	.	.	.	.	4	-4	4	4	4	.	-4	.	-4	.	-4	.	.
17	.	.	.	.	.	.	.	-8	.	-4	4	-4	-4	.	.	.	8	4	-4	-4	-4	-4	.	.	.	.	.	.	.	.	.	.
18	.	-8	.	.	.	.	.	.	.	-4	4	.	-4	.	.	-4	.	-4	4	-4	-4	.	.	4	.	4	.	4	.	-4	.	.
19	.	.	.	.	.	-8	-8	.	.	.	.	4	-4	4	4	.	.	.	-4	4	4	-4	.	.	.	.	.	.	.	.	.	.
20	.	.	.	4	.	4	.	.	.	4	4	-4	-4	4	-4	.	-4	.	.	4	.	.	.	.	.	.	.	.	.	.	.	4
21	.	.	.	4	.	-4	.	.	.	.	.	-4	4	.	-4	4	.	8	.	4	.	4	.	.	.	.	4	4	.	-4	-4	.
22	.	.	.	-4	.	-4	.	.	.	4	.	.	-4	4	4	.	8	.	.	-4	.	4	.	.	4	.	4	4	.	.	.	-4
23	.	.	.	4	.	-4	.	.	8	.	-4	.	-4	.	.	.	.	4	.	.	-4	4	-4	.	.	.	.	4	4	.	4	4
24	.	.	.	.	.	.	-8	.	.	4	4	.	-4	.	.	4	.	.	.	.	4	-4	-4	-4	.	.	.	4	4	.	-4	-4
25	.	.	.	.	.	.	.	.	.	.	4	-4	-4	4	.	-8	4	-4	-4	-4	.	.	.	.	4	4	.	.	.	-4	-4	.
26	.	8	.	.	.	.	.	.	.	-4	.	-4	-4	.	4	.	.	.	-4	4	-4	-4	.	.	-4	.	4	-4	.	.	-4	.
27	.	.	.	.	.	.	.	8	.	-4	4	-4	-4	.	.	.	.	.	-4	4	-4	4	.	.	-4	-4	.	.	.	-4	-4	.
28	.	.	8	4	.	-4	.	.	.	4	.	.	-4	4	-4	.	.	-4	.	.	-4	4	-4	4	4	.	.	.	.	4	.	.
29	.	.	.	-4	.	4	.	.	8	.	4	.	4	.	.	.	8	.	-4	.	-4	.	.	.	.	4	.	-4	.	.	.	.
30	.	.	.	4	.	4	.	.	.	4	-4	4	4	4	.	-4	8	.	.	4	.	-4	.	.	-4	.	.	.	.	-4	.	.
31	.	.	8	4	.	4	.	.	.	.	.	4	-4	.	-4	4	.	-4	.	.	4	4	-4	.	.	4	.	-4	.	.	.	.

BCT

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
1	32	·	·	·	·	·	8	·	·	4	·	4	·	4	8	4	·	4	·	4	·	·	12	·	8	·	8	·	4	4	12	4
2	32	12	4	·	4	·	·	4	8	·	·	·	·	·	·	·	8	8	4	4	4	4	·	8	·	12	·	4	·	4	·	4
3	32	4	8	·	8	4	·	·	12	8	8	4	·	·	·	4	·	·	·	4	·	·	·	12	·	8	·	4	·	·	·	·
4	32	·	·	·	4	8	12	8	·	16	·	16	4	8	12	8	·	16	·	16	4	8	12	8	16	·	16	·	4	8	12	8
5	32	·	·	·	·	8	·	·	4	·	4	·	4	8	4	·	4	·	4	·	·	12	·	8	·	8	·	4	4	12	4	·
6	32	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2
7	32	·	8	4	4	·	4	4	·	8	8	12	4	·	4	4	·	·	4	·	·	·	8	·	12	·	4	·	·	·	·	·
8	32	·	·	4	·	·	8	4	4	·	·	·	4	4	4	4	4	8	·	8	·	12	4	12	·	·	·	4	·	·	8	4
9	32	2	·	2	2	·	2	·	2	·	·	2	·	2	2	·	2	·	·	2	·	2	·	2	2	·	2	2	·	2	·	·
10	32	2	2	·	2	·	·	2	2	·	2	2	·	2	·	2	2	·	2	2	·	2	·	2	2	2	·	2	·	·	·	2
11	32	·	2	2	·	2	2	·	·	2	2	·	2	2	·	·	2	2	·	2	2	·	2	2	·	·	·	2	2	·	2	2
12	32	12	16	8	16	8	·	4	12	·	8	·	8	·	4	·	12	16	8	16	8	16	4	16	·	12	·	8	·	8	·	4
13	32	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2
14	32	12	4	·	4	·	·	4	8	·	·	·	·	·	·	·	8	8	4	4	4	4	·	8	·	12	·	4	·	4	·	4
15	32	4	8	4	8	·	·	·	8	4	12	·	12	4	·	·	4	·	·	·	·	4	8	·	4	4	4	·	·	·	·	·
16	32	4	16	4	16	8	16	8	·	12	16	12	16	8	16	8	4	·	4	·	8	·	8	·	12	·	12	·	8	·	8	·
17	32	8	·	8	·	8	16	8	16	4	·	4	·	4	16	4	16	12	·	12	·	12	16	12	·	8	·	8	·	8	16	8
18	32	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2	·	2
19	32	16	12	8	12	8	·	·	16	16	12	8	12	8	·	·	16	·	8	4	8	4	·	·	16	16	8	4	8	4	·	·
20	32	·	·	·	4	4	12	4	·	8	·	8	4	4	12	4	·	·	·	·	8	·	8	·	·	·	·	·	·	8	·	·
21	32	·	·	·	·	4	8	4	·	4	·	4	·	·	8	·	12	·	12	·	8	8	8	·	·	·	·	·	·	4	8	4
22	32	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
23	32	·	8	4	4	·	4	4	·	8	8	12	4	·	4	4	·	·	4	·	·	·	8	·	12	·	4	·	·	·	·	·
24	32	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2
25	32	·	·	4	·	·	8	4	4	·	·	·	4	4	4	4	4	8	·	8	·	12	4	12	·	·	·	4	·	·	8	4
26	32	2	2	·	2	·	·	2	2	·	2	2	·	2	·	2	2	·	2	2	·	2	·	2	2	2	·	2	·	·	·	2
27	32	·	2	2	2	2	·	·	·	2	2	2	2	·	·	·	2	2	2	2	·	·	·	·	2	2	2	2	·	·	·	·
28	32	4	8	4	8	·	·	·	8	4	12	·	12	4	·	·	4	·	·	·	·	4	8	·	4	4	4	·	·	·	·	·
29	32	8	·	4	·	4	·	·	12	·	·	·	·	·	4	12	8	·	8	·	·	8	4	8	·	8	·	4	·	4	·	·
30	32	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2	·	·	·	·	·	·	·	·	2	2	2	2	2	2	2	2
31	32	8	12	4	12	4	·	·	8	·	8	·	8	·	·	·	8	·	4	4	4	4	·	·	·	·	8	·	·	·	·	·

Sbox Analysis

- **Differential Branch Number** - 3
- **Differential Uniformity** - 8

DDT Frequency Analysis

$f(\Delta_i, \Delta_o)$	S
0	707
2	176
4	120
6	0
8	20

Sbox Analysis

- **Differential Branch Number** - 3
- **Differential Uniformity** - 8

- **Linear Branch Number** - 3
- **Maximum Absolute Linear Bias** - 8

DDT Frequency Analysis

$f(\Delta_i, \Delta_o)$	S
0	707
2	176
4	120
6	0
8	20

LAT Frequency Analysis

ϵ	S
-8	18
-4	174
0	647
4	162
8	22

BCT BDT

Sbox Analysis

- **Boomerang Uniformity** - 16

BCT Frequency Analysis

$f(\Delta_i, \nabla_o)$	S
0	445
2	176
4	150
8	110
12	50
16	30

BDT Frequency Analysis

$f(\Delta_i, \Delta_o, \nabla_o)$	S
0	31624
2	352
4	720
8	40
32	32

T1 - Construction
0000000000

T2 - Cryptanalysis
0000000●00000

T3 - Verilog
000000

T4 - Automated Analysis
000000

Conclusion
000

Differential and Linear Cryptanalysis

- **Linear Layer** (Σ_i) - Differential and Linear Branch Number is 4.
- The minimum number of active S-boxes after 3 rounds - (p^3)
  - **Differential Characteristics** - 15
  - **Linear Characteristics** - 13

Differential and Linear Cryptanalysis

- **Linear Layer** (Σ_i) - Differential and Linear Branch Number is 4.
- The minimum number of active S-boxes after 3 rounds - (p^3)
  - **Differential Characteristics** - 15
  - **Linear Characteristics** - 13

Rounds (R)	1	2	3	4	5
Minimum # Active Sboxes (Differential)	1	4	15	≤ 44	≤ 78
Minimum # Active Sboxes (Linear)	1	4	13	≤ 43	≤ 67

Differential and Linear Cryptanalysis

- **Linear Layer** (Σ_i) - Differential and Linear Branch Number is 4.
- The minimum number of active S-boxes after 3 rounds - (p^3)
  - **Differential Characteristics** - 15
  - **Linear Characteristics** - 13

Rounds (R)	1	2	3	4	5
Minimum # Active Sboxes (Differential)	1	4	15	≤ 44	≤ 78
Minimum # Active Sboxes (Linear)	1	4	13	≤ 43	≤ 67

Collision Producing Differentials - Forgery Attack

- Differentials with differences in the rate part (S_r) of State at the input (Δ_I^r) and output (Δ_O^r) of p^b.

$$\Delta_I^r \xrightarrow{\ p^b\ } \Delta_O^r$$

- Might be useful in Forgery attack on the AEAD scheme.

Truncated Differential over Sbox

Undisturbed Bits

For a specific input difference Δ_i of an S-box, if some bits of the output difference Δ_o^* remain invariant, then we call such bits **undisturbed**.

$$Pr\left[\Delta_i \xrightarrow{S} \Delta_o^*\right] = 1$$

Truncated Differential over Sbox

Undisturbed Bits

For a specific input difference Δ_i of an S-box, if some bits of the output difference Δ_o^* remain invariant, then we call such bits **undisturbed**.

$$Pr\left[\Delta_i \xrightarrow{S} \Delta_o^*\right] = 1$$

Δ_i	Δ_o^*
00001	*1***
00010	1***1
00011	***0*
00100	**110
00101	1****
01111	*1*0*

Δ_i	Δ_o^*
10000	*10**
10001	10**1
10011	0***0
10100	0*1**
10101	****1
10110	1****

Δ_i	Δ_o^*
00110	****1
00111	0**1*
01000	**11*
01011	***1*
01100	**00*
01110	*0***

Δ_i	Δ_o^*
10111	****0
11000	**1**
11100	**0**
11110	*1***
11111	*0***

Truncated Differential over Sbox

Undisturbed Bits

For a specific input difference Δ_i of an S-box, if some bits of the output difference Δ_o^* remain invariant, then we call such bits **undisturbed**.

$$Pr\left[\Delta_i \xrightarrow{S} \Delta_o^*\right] = 1$$

Note - Inverse S-Box only has 2 undisturbed bits.

Δ_i	Δ_o^*
00001	*1***
00010	1***1
00011	***0*
00100	**110
00101	1****
01111	*1*0*

Δ_i	Δ_o^*
10000	*10**
10001	10**1
10011	0***0
10100	0*1**
10101	****1
10110	1****

Δ_i	Δ_o^*
00110	****1
00111	0**1*
01000	**11*
01011	***1*
01100	**00*
01110	*0***

Δ_i	Δ_o^*
10111	****0
11000	**1**
11100	**0**
11110	*1***
11111	*0***

3.5 Round Truncated Differential Distinguisher

	100000000000000000000000000000000000000000000000000000000000000
	000000000000000000000000000000000000000000000000000000000000000
/	000000000000000000000000000000000000000000000000000000000000000
	100000000000000000000000000000000000000000000000000000000000000
	100000000000000000000000000000000000000000000000000000000000000

3.5 Round Truncated Differential Distinguisher

I	100000000000000000000000000000000000000000000000000000000000000
	000000000000000000000000000000000000000000000000000000000000000
	000000000000000000000000000000000000000000000000000000000000000
	000000000000000000000000000000000000000000000000000000000000000
	100000000000000000000000000000000000000000000000000000000000000
	100000000000000000000000000000000000000000000000000000000000000
S_1	000000000000000000000000000000000000000000000000000000000000000
	*00000000000000000000000000000000000000000000000000000000000000
	*00000000000000000000000000000000000000000000000000000000000000
	*00000000000000000000000000000000000000000000000000000000000000
	000000000000000000000000000000000000000000000000000000000000000

3.5 Round Truncated Differential Distinguisher

I	100000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000 100000000000000000000000000000000000000000000000000000000000000 100000000000000000000000000000000000000000000000000000000000000
S_1	000000000000000000000000000000000000000000000000000000000000000 •00000000000000000000000000000000000000000000000000000000000000 •00000000000000000000000000000000000000000000000000000000000000 •00000000000000000000000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000
P_1	000000000000000000000000000000000000000000000000000000000000000 •0000000000000000000000000000000000000•00000000000000000000•00 ••0000•0000000000000000000000000000000000000000000000000000000 •000000000•000000•00000000000000000000000000000000000000000000 000000000000000000000000000000000000000000000000000000000000000

3.5 Round Truncated Differential Distinguisher

```
      1000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000
  I   0000000000000000000000000000000000000000000000000000000000000000
      1000000000000000000000000000000000000000000000000000000000000000
      1000000000000000000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000000000000000000000
      *000000000000000000000000000000000000000000000000000000000000000
  S₁  *000000000000000000000000000000000000000000000000000000000000000
      *000000000000000000000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000

      0000000000000000000000000000000000000000000000000000000000000000
      *0000000000000000000000000000000000*0000000000000000000000000*00
  P₁  **0000*0000000000000000000000000000000000000000000000000000000000
      *000000000*000000*0000000000000000000000000000000000000000000000
      0000000000000000000000000000000000000000000000000000000000000000

      **0000*000*000000*0000000000000000*0000000000000000000000000*00
      **0000*000*000000*0000000000000000*0000000000000000000000000*00
  S₂  **0000*000*000000*0000000000000000*0000000000000000000000000*00
      **0000*000*000000*0000000000000000*0000000000000000000000000*00
      *000000000*000000*0000000000000000*0000000000000000000000000*00
```

3.5 Round Truncated Differential Distinguisher

3.5 Round Truncated Differential Distinguisher

```
    1000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
I   0000000000000000000000000000000000000000000000000000000000000000
    1000000000000000000000000000000000000000000000000000000000000000
    1000000000000000000000000000000000000000000000000000000000000000

    0000000000000000000000000000000000000000000000000000000000000000
    *000000000000000000000000000000000000000000000000000000000000000
S₁  *000000000000000000000000000000000000000000000000000000000000000
    *000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000

    0000000000000000000000000000000000000000000000000000000000000000
    *0000000000000000000000000000000000000000*0000000000000000000*00
P₁  **0000*00000000000000000000000000000000000000000000000000000000
    *000000000*000000*0000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000

    **0000*000*000000*0000000000000000000*0000000000000000000000*00
    **0000*000*000000*0000000000000000000*0000000000000000000000*00
S₂  **0000*000*000000*0000000000000000000*0000000000000000000000*00
    **0000*000*000000*0000000000000000000*0000000000000000000000*00
    *000000000*000000*0000000000000000000*0000000000000000000000*00

    **0*00*000*00000**0*0000*00**0000*0*0**00000*000000000000*00*00
    **0*00**00*00*000*00*000000000000000000*00*0000*000*000000*0*0
P₂  *000*00*00*00***000*000*0000000000000000000**0000*000000000000
    **0000**00*00*0***0*0*00*000*000000*0000*000000000*000000*0*00
    *000*00*00000**000000*00000000000000000**0*0000*0000*000000*00

    *****0**00**0*0***00**0**0000*0*0****000**0*0*0*0000*0*00**0
    *****0**00**0*0***00**0**0000*0*0****000**0*0*0*0000*0*00**0
S₃  *****0**00**0*0***00**0**0000*0*0****000**0*0*0*0000*0*00**0
    *****0**00**0*0***00**0**0000*0*0****000**0*0*0*0000*0*00**0
    **0**0**00**00*0****00***0**0000*0*0***000**0*0*0000*0*00**0
```

3.5 Round Truncated Differential Distinguisher

```
    1000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
 I  0000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000
    1000000000000000000000000000000000000000000000000000000000000000
    1000000000000000000000000000000000000000000000000000000000000000

    0000000000000000000000000000000000000000000000000000000000000000
    *000000000000000000000000000000000000000000000000000000000000000
S₁  *000000000000000000000000000000000000000000000000000000000000000
    *000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000

    0000000000000000000000000000000000000000000000000000000000000000
    *0000000000000000000000000000000000000*000000000000000000000*00
P₁  **0000*0000000000000000000000000000000000000000000000000000000
    *0000000000*000000*00000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000000

    **0000*000*000000*00000000000000000000*000000000000000000000*00
    **0000*000*000000*00000000000000000000*000000000000000000000*00
S₂  **0000*000*000000*00000000000000000000*000000000000000000000*00
    **0000*000*000000*00000000000000000000*000000000000000000000*00
    *000000000*000000*00000000000000000000*000000000000000000000*00

    **0*00*000*00000**0**0000*00**0000*0*0**000000000000*00*00
    **0*00**00*00*000*0*0000000000000000000*00**0000*000*000000*0*00*0
P₂  ***00*00**000***0000*0000000000000000***0000*000000000000*00*0
    **0000**00*00*0**0*00*000*000000*0000*0000000*000000*0000*00
    *000*00*00*00000**000000*00000000000*0*0*0000*0000*000000*00*00

    ****0*00**0*0*0****00**0**0*0000*0*0****000**00*0*0000*0*00*0
    ****0*00**0*0*0****00**0**0*0000*0*0****000**00*0*0000*0*00*0
S₃  ****0*00**0*0*0****0*00*00*00000*0*0***000**00*0*0000*0*00*0
    ****0*00**0*0*0****00**0**0*0000*0*0****000**00*0*0000*0*00*0
    **0*0*00**0*00*0****00**0**0*0000*0*0****000**00*0*0000*0*00*0

    *********0*********************************0***0*********0
    *****0***********************0*0**********0**0**0****0*****0***
P₃  *********************************00****0****0****00*****0**
    ****0***********************0***********0***0***0*0*****0***
    ********0**0****************************0**********0***0*********
```

3.5 Round Truncated Differential Distinguisher

Impossible Differential

5 Round Differential

x_0	0000000000000000
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	8000000000000000
	$\not\to$
x_0	0100000000100002
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	0000000000000000

Impossible Differential

5 Round Impossible Truncated Differential
3.5 Truncated Differential Distinguisher

5 Round Differential

x_0	0000000000000000
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	8000000000000000
$\not\rightarrow$	
x_0	0100000000100002
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	0000000000000000

Impossible Differential

5 Round Impossible Truncated Differential		
	3.5 Truncated Differential Distinguisher	
S_4	***0*********	
	0******	
	0******	
	0******	
	0******	
1.5 Round Impossible Backward Differential		

5 Round Differential

x_0	0000000000000000
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	8000000000000000
	$\not\to$
x_0	0100000000100002
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	0000000000000000

Impossible Differential

5 Round Differential

x_0	0000000000000000
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	8000000000000000
	$\nrightarrow$
x_0	0100000000100002
x_1	0000000000000000
x_2	0000000000000000
x_3	0000000000000000
x_4	0000000000000000

5 Round Impossible Truncated Differential	
3.5 Truncated Differential Distinguisher	
S_4	**0*********
	0*******
	0*******
	0*******
	0*******
1.5 Round Impossible Backward Differential	
S_4	**
	**
	111000010110001011001111101111101010100101000110100011010100101
	**
P_4	0**0*0**0*00*0000*00****0****0***0***0**00*0*0*00***000*0000*00*0*
	0**0*0**0*00*0000*00****0****0***0***0**00*0*0*00***000*0000*00*0*
	0**0*0**0*00*0000*00****0****0***0***0**00*0*0*00***000*0000*00*0*
	0110101101001000011001111011110111011100101010011100010000100101
	0**0*0**0*00*0000*00****0****0***0***0**00*0*0*00***000*0000*00*0*
S_5	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0110101101001000011001111011110111011100101010011100010000100101
	0000000000000000000000000000000000000000000000000000000000000000
P_5	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000

T1 - Construction
○○○○○○○○○○

T2 - Cryptanalysis
○○○○○○○○○○○●○

T3 - Verilog
○○○○○○

T4 - Automated Analysis
○○○○○○

Conclusion
○○○

Zero Sum Distinguisher

ASCON Sbox - χ Keccak Mapping

- χ has branch number 2, Fix Point at 0, Outputs depend on 3 Inputs
- ASCON S-box is designed by adding lightweight affine transformations to the input and output of .

Zero Sum Distinguisher

ASCON Sbox - χ Keccak Mapping

- χ has branch number 2, Fix Point at 0, Outputs depend on 3 Inputs
- ASCON S-box is designed by adding lightweight affine transformations to the input and output of .

Algebraic Degree

- ASCON S-box (S) $\rightarrow$ ASCON Permutation (p)

$$d(S) = 2 \rightarrow d(p) \leq 2 \rightarrow d(p^r) \leq 2^r$$

- ASCON S-Box Inverse (S^{-1}) $\rightarrow$ ASCON Permutation Inverse (p^{-1})

$$d(S^{-1}) = 3 \rightarrow d(p^{-1}) \leq 3 \rightarrow d\left((p^{-1})^r\right) = 3^r$$

T1 - Construction
○○○○○○○○○○

T2 - Cryptanalysis
○○○○○○○○○○○○○●

T3 - Verilog
○○○○○○

T4 - Automated Analysis
○○○○○○

Conclusion
○○○

12 Round (p^a) - Zero Sum Distinguisher (2^d)

Backward Rounds
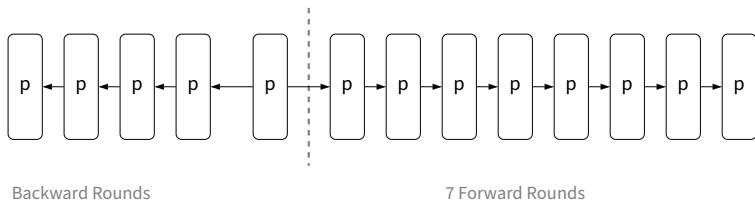
7 Forward Rounds

Basic Distinguisher (2^{244})

- 5 Backward Rounds - (3^5)
- 7 Forward Rounds - (2^7)

$d = \max\{3^5, 2^7\} + 1 = 243 + 1 = 244$

12 Round (p^a) - Zero Sum Distinguisher (2^d)

Backward Rounds   Free Middle Round   7 Forward Rounds
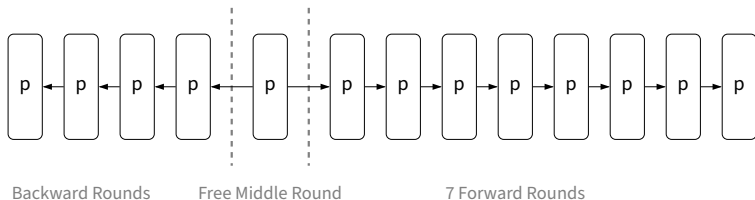
Basic Distinguisher (2^{244})

- 5 Backward Rounds - (3^5)
- 7 Forward Rounds - (2^7)

$d = \max\{3^5, 2^7\} + 1 = 243 + 1 = 244$

Free Middle Round (2^{130})

- d is a multiple of 5-Bit Sbox
- All d variables positioned in S-box
- S-Box Output also contains d variable and $320 - d$ constant bits
- 4 Backward, 7 Forward Rounds

$d = \max\{3^4, 2^7\} + 1 = 128 + 1 = 129 \rightarrow 130$

Outline
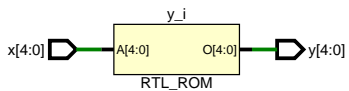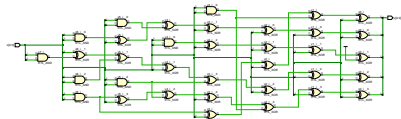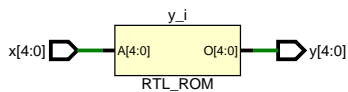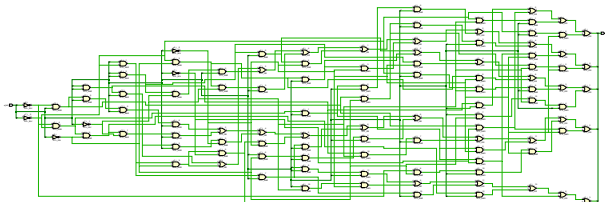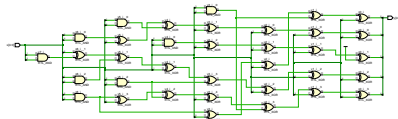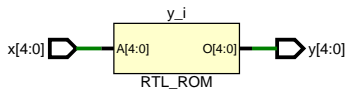
Sbox

Sbox

Sbox

Sbox
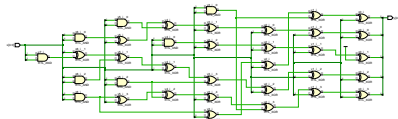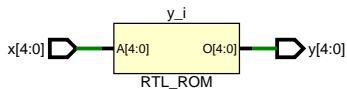
LuT	ANF	K-MAP
43.920	42.480	39.240

Component	Slice LuT (Logic)	OBUF	IBUF	LUT5	LUT4
Number Used	3	5	5	3	2
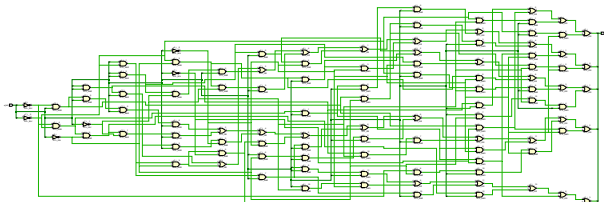
Linear Diffusion - 1958.400 GE
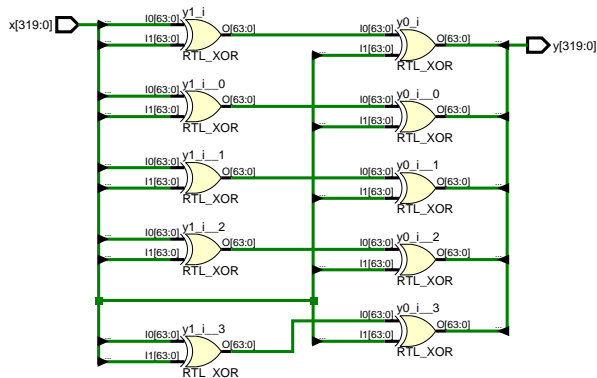
Linear Diffusion - 1958.400 GE
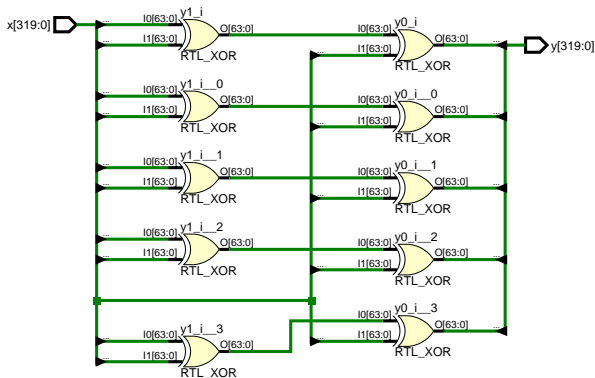
Components	Number Used
Slice LuT (Logic)	169
OBUF	320
LUT3	320
IBUF	320

Round Constant - 321.840 GE
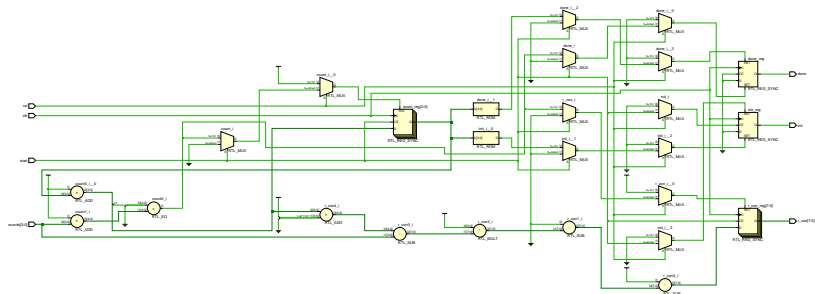
Round Constant - 321.840 GE
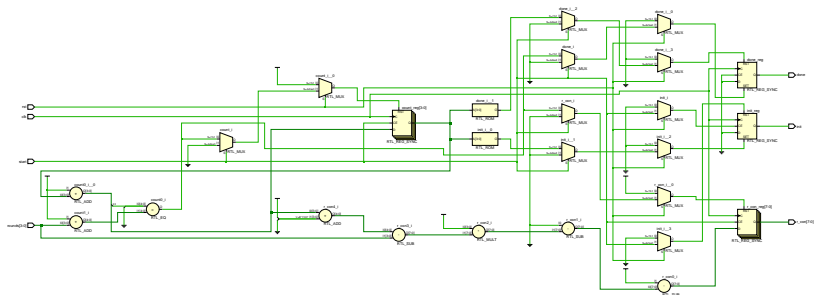
Components	Slice LuT (Logic)	Slice Register (Flip Flop)	FDRE	FDRE	OBUF	LUT1	LUT6	IBUF	LUT4	LUT2	LUT3	FDSE	LUT5	BUFG (Clock)
Number Used	29	14	12	10	9	7	7	7	6	5	3	2	1	1

Permutation

Permutation

- LuT Sbox - 9148.320 GE

Components	Slice LuT (Logic)	Slice Registers (Flip Flop)	LUT3	FDRE	IBUF	OBUF	LUT5	LUT4	LUT6	LUT	1LUT2	BUFG (Clock)
Number Used	740	334	862	334	326	320	196	110	99	9	5	1

- KMAP Sbox - 8848.800 GE

Components	Slice LuT (Logic)	Slice Registers (Flip Flop)	LUT3	FDRE	IBUF	OBUF	LUT5	LUT6	LUT4	LUT1	LUT2	BUFG (Clock)
Number Used	777	334	955	334	326	320	196	134	13	9	5	1

- ANF Sbox - 9056.160 GE

Components	Slice LuT (Logic)	Slice Registers (Flip Flop)	LUT3	FDRE	IBUF	OBUF	LUT6	LUT5	LUT4	LUT1	LUT2	BUFG (Clock)
Number Used	883	334	955	334	326	320	246	140	13	9	5	1

Test Bench - p^{12}, p^8, p^6

Test Bench - p^{12}, p^8, p^6
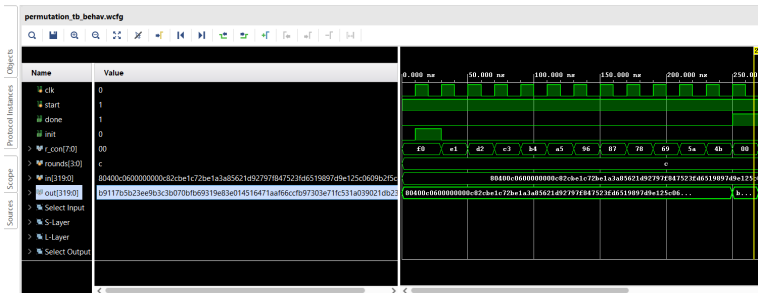
Word	Input	12 Rounds Output	8 Rounds Output	6 Rounds Output
x_0	80400c0600000000	b9117b5b23ee9b3c	7e23b301941a4a62	25c593c800b08021
x_1	c82cbe1c72be1a3a	3b070bfb69319e83	9b907c66bad8b7ed	3cb7732ae5f81096
x_2	85621d92797f8475	e014516471aaf66c	332e8d3ba929ce17	e8edf67b79fb4c3e
x_3	23fd6519897d9e12	cfb97303e71fc531	efc6c3a1c24a7b4e	2d1d39a69214abf8
x_4	5c0609b2f5ca3aaa	a039021db23a484c	45a7f92724354d6a	9e9ce6a15408bc5d

Outline

1. T1 - Construction

2. T2 - Cryptanalysis

3. T3 - Verilog

4. T4 - Automated Analysis

5. Conclusion

MILP

Number of Active Sboxes - Basic

Variables

MILP

Number of Active Sboxes - Basic

Variables

- $x_{r,w,b} \in \{0,1\}$ - S-box input Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

MILP

Number of Active Sboxes - Basic

Variables

- $x_{r,w,b} \in \{0,1\}$ - S-box input Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

- $y_{r,w,b} \in \{0,1\}$ - S-box output Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

MILP

Number of Active Sboxes - Basic

Variables

- $x_{r,w,b} \in \{0,1\}$ - S-box input Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

- $y_{r,w,b} \in \{0,1\}$ - S-box output Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

- $d_{r,b} \in \{0,1\}$ - b^{th} ($0 \leq b \leq 63$) S-box of round r is active

MILP

Number of Active Sboxes - Basic

Variables

- $x_{r,w,b} \in \{0,1\}$ - S-box input Bit b ($0 \le b \le 63$) of word X_w ($0 \le w \le 4$) of round r active or not.

- $y_{r,w,b} \in \{0,1\}$ - S-box output Bit b ($0 \le b \le 63$) of word X_w ($0 \le w \le 4$) of round r active or not.

- $d_{r,b} \in \{0,1\}$ - b^{th} ($0 \le b \le 63$) S-box of round r is active

- $u_{r,w,b} \in \{0,1\}$ - Linear layer model in word x_w ($0 \le w \le 4$) of round r.

MILP

Number of Active Sboxes - Basic

Variables

- $x_{r,w,b} \in \{0, 1\}$ - S-box input Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

- $y_{r,w,b} \in \{0, 1\}$ - S-box output Bit b ($0 \leq b \leq 63$) of word X_w ($0 \leq w \leq 4$) of round r active or not.

- $d_{r,b} \in \{0, 1\}$ - b^{th} ($0 \leq b \leq 63$) S-box of round r is active

- $u_{r,w,b} \in \{0, 1\}$ - Linear layer model in word x_w ($0 \leq w \leq 4$) of round r.

Objective Function

- Minimize the Active S-boxes

$$\min \sum_{r=1}^{R} \sum_{b=0}^{63} d_{r,b}$$

MILP

Number of Active Sboxes - Basic

Constraints

- **Non-Triviality** - At least 1 Active Input Bit at Start $\sum_{w=0}^{4} \sum_{b=0}^{63} x_{0,w,b} \geq 1$

MILP

Number of Active Sboxes - Basic

Constraints

- **Non-Triviality** - At least 1 Active Input Bit at Start $\sum_{w=0}^{4} \sum_{b=0}^{63} x_{0,w,b} \geq 1$
- **Active S-box** - Minimum 1 and Maximum 5 Active Input and Output Bit(s).

$$d_{r,b} \leq \sum_{w=0}^{4} x_{r,w,b} \leq 5d_{r,b} \qquad\qquad d_{r,b} \leq \sum_{w=0}^{4} y_{r,w,b} \leq 5d_{r,b}$$

MILP
Number of Active Sboxes - Basic

Constraints

- **Non-Triviality** - At least 1 Active Input Bit at Start $\sum_{w=0}^{4} \sum_{b=0}^{63} x_{0,w,b} \geq 1$
- **Active S-box** - Minimum 1 and Maximum 5 Active Input and Output Bit(s).

$$d_{r,b} \leq \sum_{w=0}^{4} x_{r,w,b} \leq 5d_{r,b} \qquad\qquad d_{r,b} \leq \sum_{w=0}^{4} y_{r,w,b} \leq 5d_{r,b}$$

- **S-box Branch Number** - 3. $\sum_{w=0}^{4} (x_{r,w,b} + y_{r,w,b}) \geq 3d_{r,b}$

MILP
Number of Active Sboxes - Basic

Constraints

- **Non-Triviality** - At least 1 Active Input Bit at Start $\sum_{w=0}^{4} \sum_{b=0}^{63} x_{0,w,b} \geq 1$
- **Active S-box** - Minimum 1 and Maximum 5 Active Input and Output Bit(s).

$$d_{r,b} \leq \sum_{w=0}^{4} x_{r,w,b} \leq 5d_{r,b} \qquad\qquad d_{r,b} \leq \sum_{w=0}^{4} y_{r,w,b} \leq 5d_{r,b}$$

- **S-box Branch Number** - 3. $\sum_{w=0}^{4}(x_{r,w,b} + y_{r,w,b}) \geq 3d_{r,b}$
- **Linear Layer - XOR Operation Model**

$$y_{r,0,b} + y_{r,0,b-19} + y_{r,0,b-28} + x_{r+1,0,b} = 2u_{r,0,b} \qquad y_{r,2,b} + y_{r,2,b-1} + y_{r,2,b-6} + x_{r+1,2,b} = 2u_{r,2,b}$$

$$y_{r,1,b} + y_{r,1,b-61} + y_{r,1,b-39} + x_{r+1,1,b} = 2u_{r,1,b} \qquad y_{r,3,b} + y_{r,3,b-10} + y_{r,3,b-17} + x_{r+1,3,b} = 2u_{r,3,b}$$

$$y_{r,4,b} + y_{r,4,b-7} + y_{r,4,b-41} + x_{r+1,4,b} = 2u_{r,4,b}$$

Logical Conditional Modelling
MILP

$$x_{r,0,b}, x_{r,1,b}, x_{r,2,b}, x_{r,3,b}, x_{r,4,b}) = (\delta_0, \delta_1, \delta_2, \delta_3, \delta_4) \Rightarrow y_{r,w,b} = \delta$$

$$\sum_{w'=0}^{4} (-1)^{\delta_i} x_{r,w',b} + (-1)^{\delta+1} y_{r,w,b} - \delta + \sum_{w'=0}^{4} \delta_i \geq 0$$

Logical Conditional Modelling
MILP

$$x_{r,0,b}, x_{r,1,b}, x_{r,2,b}, x_{r,3,b}, x_{r,4,b}) = (\delta_0, \delta_1, \delta_2, \delta_3, \delta_4) \Rightarrow y_{r,w,b} = \delta$$

$$\sum_{w'=0}^{4} (-1)^{\delta_i} x_{r,w',b} + (-1)^{\delta+1} y_{r,w,b} - \delta + \sum_{w'=0}^{4} \delta_i \geq 0$$

Recall - Undisturbed Bits

Δ_i	Δ_o^*
00001	*1***
00011	***0*
00101	1****
11111	*0***
01011	***1*

Δ_i	Δ_o^*
10101	****1
10110	1****
11110	*1***
01110	*0***

Δ_i	Δ_o^*
10111	****0
11000	**1**
11100	**0**
00110	****1

Gurobi Implementation

Rounds	Active S-Boxes	Variables (Real)	Variables (Binary)	Inequalities	Time Taken
1	1	960	384	2561	0.03 s
2	4	1920	448	5121	0.51 s
3	12	2880	512	7681	2 min 52.47 s

Table: Basic MILP

Rounds	Active S-Boxes	Variables (Real)	Variables (Binary)	Inequalities	Time Taken
1	1	965	384	3393	0.02 s
2	4	1925	448	6785	1.32 s
3	12	2885	512	10177	3 min 28.54 s

Table: Logical Conditional Modeling

Convex Hull

MILP

```
ascon_hull = list(Polyhedron(vertices=generate_vertices(Ascon)).inequality_generator())

ascon_hull

[An inequality (-1, 0, 0, 0, 0, 0, 0, 0, 0, 0) x + 1 >= 0,
 An inequality (0, -1, 0, 0, 0, 0, 0, 0, 0, 0) x + 1 >= 0,
 An inequality (0, 0, -1, 0, 0, 0, 0, 0, 0, 0) x + 1 >= 0,
 An inequality (0, 0, 0, -1, 0, 0, 0, 0, 0, 0) x + 1 >= 0,
 An inequality (-1, 0, 0, -1, -1, 0, 0, -1, -1, 0) x + 4 >= 0,
 An inequality (0, -1, 0, -1, -1, 0, 1, 1, 1, 0) x + 2 >= 0,
 An inequality (0, 0, 0, -1, 0, 0, 0, 0, 0, 0) x + 1 >= 0,
 An inequality (1, -1, -1, -1, -1, 0, 1, 0, 0, 0) x + 3 >= 0,
 An inequality (0, 0, 0, 0, 0, 0, 1, 0, 0, 0) x + 0 >= 0,
 An inequality (-1, 0, 0, -1, -1, 0, 0, 1, 1, 0) x + 2 >= 0,
 An inequality (0, 0, -1, -1, -1, 0, 1, 1, 1, 0) x + 2 >= 0,
 An inequality (-3, -2, -3, -4, -6, -1, -1, -3, -3, -1) x + 23 >= 0,
 An inequality (-1, -1, -1, -1, -1, 0, -1, 0, 0, 0) x + 5 >= 0,
 An inequality (-1, -1, -1, -2, -2, 0, -1, -1, -1, 0) x + 8 >= 0,
 An inequality (-1, -1, -1, -1, 1, 0, 1, 0, 0, 0) x + 3 >= 0,
 An inequality (0, -1, -1, -1, -1, 0, 0, -1, 0) x + 5 >= 0,
 An inequality (-2, -1, -1, -1, 1, 0, 2, -1, -1, 0) x + 7 >= 0,
 An inequality (-1, -1, -1, -2, -1, 0, 1, -2, -2, 0) x + 8 >= 0,
 An inequality (-1, -1, -1, -1, 0, 1, 2, 2, 0) x + 4 >= 0,
 An inequality (-1, -1, -1, -3, -2, 0, 2, 3, 3, 0) x + 5 >= 0,
 An inequality (-1, -1, -1, -1, 0, 1, 1, 1, 0) x + 3 >= 0,
 An inequality (1, -1, -1, -2, -2, 0, 2, 1, 1, 0) x + 4 >= 0,
 An inequality (0, 0, 0, -1, -1, -1, 1, 1, 1, -1) x + 3 >= 0,
 An inequality (0, 0, 0, 0, 0, 0, 0, 0, -1) x + 1 >= 0,
 An inequality (0, 0, -1, -1, -1, 0, -1, -1, 0) x + 5 >= 0,
```

Figure: Number of Inequalities generated - 2415

T1 - Construction
0000000000

T2 - Cryptanalysis
0000000000000

T3 - Verilog
000000

T4 - Automated Analysis
000000

Conclusion
●○○

Outline

1. T1 - Construction

2. T2 - Cryptanalysis

3. T3 - Verilog

4. T4 - Automated Analysis

5. Conclusion

References

- https://ascon.iaik.tugraz.at
- **[DEMS15]** *Cryptanalysis of Ascon* - Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. **IACR** - 2015/030 (pp. 28, 31–33, 35).
- **[Tez16]** *Truncated, Impossible, and Improbable Differential Analysis of Ascon* - Cihangir Tezcan. **IACR** - 2016/490 (pp. 28, 33).

T1 - Construction
○○○○○○○○○○

T2 - Cryptanalysis
○○○○○○○○○○○○○○

T3 - Verilog
○○○○○○

T4 - Automated Analysis
○○○○○○

Conclusion
○○●

Thanks

Brownie Points - LaTeX-Tikz Block Diagrams

- State Design
- Authenticated Encryption
- Zero Sum Distinguisher

Implementation Info

- Github Link: https://github.com/highgroundmaster/ASCON