Ascon MAC, PRF, and Short-Input PRF

Lightweight, Fast, and Efficient Pseudorandom Functions

Christoph Dobraunig[1], Maria Eichlseder[2], Florian Mendel[3], and Martin Schläffer[3]

[1] Intel Labs, USA
christoph.dobraunig@intel.com
[2] Graz University of Technology, Austria
maria.eichlseder@iaik.tugraz.at
[3] Infineon Technologies AG, Germany
firstname.lastname@infineon.com

Abstract. In 2023, NIST has selected Ascon as the new standard for lightweight cryptography. The Ascon v1.2 family provides authenticated encryption, hash functions, and extendable output functions, all using the same Ascon permutation. The main use case of Ascon is to provide efficient cryptographic primitives for resource-constraint devices. While additional primitives can be built on top of the existing Ascon functions, dedicated schemes are often more efficient. In this paper, we enrich the functionality of Ascon by providing efficient Pseudorandom Functions (PRFs), Message Authentication Codes (MACs), and a fast short-input PRF for messages up to 128 bits.

Keywords: Permutation-based cryptography · Pseudorandom function · Message authentication code · Ascon

1 Introduction

The Ascon family of authenticated encryption schemes [DEMS14] was first published in the beginning of 2014 as a submission to the CAESAR competition [CAE14]. After 5 years of public scrutiny, the authenticated encryption schemes Ascon-128 and Ascon-128a (v1.2) [DEMS16] were recommended as the first choice for lightweight applications in the final portfolio of CAESAR for resource-constrained environments. Subsequently, the cipher suite Ascon (v1.2) [DEMS21b, DEMS21a], containing authenticated encryption as well as hashing functionality, has been submitted to the NIST lightweight cryptography (LWC) competition [Nat18]. After another 4 years in a public competition, Ascon has been selected in 2023 to form the new NIST lightweight cryptography standard [NIS23].

Ascon's permutation also serves as a basis of Isap [DEM+20, DEM+21], an authenticated encryption scheme designed to provide enhanced robustness

The final publication is available at Springer via https://doi.org/10.1007/978-3-031-58868-6_15

against side-channel and fault attacks on algorithmic level. Isap was a finalist in the NIST LWC standardization process. In these contexts, the cryptanalytic security and implementation properties of the Ascon permutation have been analyzed very thoroughly.

The Ascon schemes offer a full symmetric cipher suite satisfying most needs in a typical communication protocol. They can help protect authenticity and confidentiality in a keyed setting as well as integrity in an unkeyed setting. However, many applications can still profit from more efficient tailored solutions that only cover either authenticity, or confidentiality [MSP+23]. Especially MACs have traditionally been built using underlying symmetric primitives, like block ciphers or hash functions, which often results in inferior performance.

To address these needs, we define two lightweight and efficient pseudorandom functions (PRFs) in this paper; Ascon-PRF and Ascon-PRFshort. The function Ascon-PRF processes inputs of arbitrary length and produces outputs of arbitrary length. In particular, Ascon-PRF adapts the full-keyed sponge (FKS) mode [BDPV07,BDPV12,MRV15,DMV17] using a rate of 256 bits during absorption and a rate of 128 bits in the squeezing phase. In contrast, Ascon-PRFshort operates only on short inputs ($\leq$ 128 bits) producing outputs of short length ($\leq$ 128 bits).

PRFs are very versatile building blocks allowing to instantiate various cryptographic functionalities. For example, Ascon-PRF is an efficient and versatile choice for general-purpose lightweight message authentication. We define the corresponding message authentication code (MAC) Ascon-MAC based on Ascon-PRF, which produced a fixed 128-bit tag. Other use-cases of PRFs are, e.g., stream ciphers or to instantiate SIV [RS06].

Ascon-PRFshort excels whenever short data needs to be authenticated, e.g., the authentication of pointers, in challenge-response protocols, or key derivation functions (KDFs) that derive symmetric keys of entities from a master key. Since Ascon-PRFshort can very efficiently map a 128-bit input to a 128-bit output, it can potentially be used in modes of operation, which are commonly instantiated with pseudorandom permutations (block ciphers), but would work as well with a PRF.

For all proposed designs, we provide a security analysis based on the properties of the permutation and the used modes of operation. We also present a performance evaluation to show the efficiency on different software platforms.

Outline. In Section 2, we specify the new modes of operation and its parameters. In Section 3, we recall the definition of the Ascon permutation to complete the specification. In Section 4, we provide the design rationale for the new designs together with security claims and performance results on different software platforms. Finally, in Section 5 we provide preliminary cryptanalysis results of the new modes based on existing related work on Ascon.

2 Specification

In this section, we introduce the state layout and notation for our functions and specify the PRFs and the MAC. Note that these specifications are based on Ascon v1.2 while the final Ascon NIST standard may be slightly adapted (e.g. using a different endianness).

2.1 State and Notation

Our algorithms operate on a 320-bit state S which is updated using the a-round permutation p^a. The state S is divided into an outer part and an inner part, whose size is different for the absorbing and squeezing phase.

For the description and application of the round transformations (Section 3), the 320-bit state S is split into five 64-bit words x_i, as follows:

$$S = x_0 \,\|\, x_1 \,\|\, x_2 \,\|\, x_3 \,\|\, x_4\,.$$

Whenever S needs to be interpreted as a byte-array (or bitstring) used in the sponge interface, the array starts with the most significant byte (or bit) of x_0 as byte 0 and ends with the least significant byte (or bit) of x_4 as byte 39. Table 1 lists the notation.

Table 1: Notation used for Ascon's interface, mode, and permutation

K	Secret key K of k bits
M, T	Input message M, output tag T (in blocks M_i, T_i)
S	The 320-bit state S of the sponge construction
p, p^a	Permutation p^aconsisting of a update rounds p
$x \in \{0,1\}^k$	Bitstring x of length k (variable if $k = *$)
0^k	Bitstring of k bits (variable length if $k = *$), all 0
$\lvert x \rvert$	Length of the bitstring x in bits
$\lfloor x \rfloor_k$	Bitstring x truncated to the first (most significant) k bits
$\lceil x \rceil^k$	Bitstring x truncated to the last (least significant) k bits
$x \,\|\, y$	Concatenation of bitstrings x and y
$x \oplus y$	Xor of bitstrings x and y
$x \bmod y$	Remainder in integer division of x by y
$\lceil x \rceil$	Ceiling function, smallest integer larger than x
p_C, p_S, p_L	Constant-addition, substitution and linear layer of $p = p_L \circ p_S \circ p_C$
$x_0, \ldots, x_4$	The five 64-bit words of the state S
$x_{0,i}, \ldots, x_{4,i}$	Bit i, $0 \le i < 64$, of words $x_0, \ldots, x_4$, with $x_{\cdot,0}$ the rightmost bit (LSB)
$x \oplus y$	Bitwise xor of 64-bit words or bits x and y
$x \odot y$	Bitwise and of 64-bit words or bits x and y (denoted $x\,y$ in the ANF)
$x \ggg i$	Right-rotation (circular shift) by i bits of 64-bit word x

2.2 Algorithms

Pseudorandom functions. Ascon-PRF is parameterized by the key length of k bits, output rate of r_o bits, internal round number a, and maximum output length of $0 < t < 2^{32}$ bits (or $t = 0$ for unlimited output). The algorithm $\mathcal{G}_{k,r_o,a,t}$ takes as its input a secret key K with k bits, input data M of arbitrary length, and a requested output length $\ell \leq t$. It returns an output T of size ℓ bits:

$$\mathcal{G}_{k,r_o,a,t}(K, M, \ell) = T\,.$$

Ascon-PRFshort is parameterized by the key length k bits, input length $m \leq 128$ bits, internal round number a, and output size $t \leq 128$ bits. The algorithm $\mathcal{F}_{k,m,a,t}$ takes as its input a secret key K with k bits and some input data M of m bits. It produces an output T of size t bits:

$$\mathcal{F}_{k,m,a,t}(K, M) = T\,.$$

Message authentication. Ascon-MAC is parameterized by the key length k bits, output rate r_o, internal round number a, and tag length t. It specifies an authentication algorithm $\mathcal{A}_{k,r_o,a,t}$ and a verification algorithm $\mathcal{V}_{k,r_o,a,t}$, both calling $\mathcal{G}_{k,r_o,a,t}$. The authentication algorithm $\mathcal{A}_{k,r_o,a,t}$ takes as its input a secret key K with k bits and a message M of arbitrary length. It produces a tag T of length t as its output:

$$\mathcal{A}_{k,r_o,a,t}(K, M) = T\,.$$

The verification procedure $\mathcal{V}_{k,r_o,a,t}$ takes as input the key K, message M and tag T, and outputs either pass if the verification of the tag is correct or fail if it fails:

$$\mathcal{V}_{k,r_o,a,t}(K, M, T) \in \{\mathsf{pass}, \mathsf{fail}\}\,.$$

2.3 Recommended Parameter Sets

Table 2 lists our recommended instances for all PRFs and MACs and specifies their parameters. The relevant parameters include the key size k, message input size limit m, input rate (block size) r_i, tag output size limit t, output rate (block size) r_o, and the number of rounds a for the permutation p^a.

2.4 Arbitrary-Length Pseudorandom Functions

The mode of operation of Ascon-PRF is based on full-state keyed sponge modes [BDPV07] such as the DonkeySponge [BDPV12] mode. The PRF is illustrated in Figure 1 and specified in Algorithm 1.

Table 2: Parameters for recommended **Pseudorandom Functions (PRF)** and **Message Authentication Codes (MAC)**. Unlimited input/output lengths ('unlim.') are implicitly limited by the security claim to $\leq 2^{64}$ blocks.

Name	Algorithms	Bit size of					Rounds
		key	data (block)		output (block)		p^a
		k	m	r_i	t	r_o	a
Ascon-MAC	$\mathcal{A}, \mathcal{V}_{128,128,12,128}$	128	unlim.	256	128	128	12
Ascon-PRF	$\mathcal{G}_{128,128,12,0}$	128	unlim.	256	unlim.	128	12
Ascon-PRFshort	$\mathcal{F}_{128,*,12,*}$	128	≤ 128	128	≤ 128	128	12

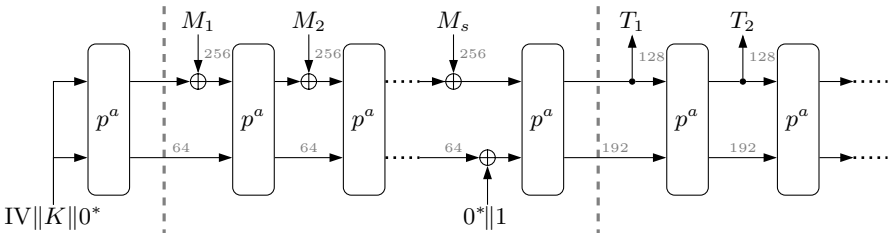


Fig. 1: Pseudorandom Function $\mathcal{G}_{k,r_\mathrm{o},a,t}$ with output length ℓ ($\ell \leq t$ or $t = 0$).

Initialization. The 320-bit initial state of Ascon-PRF is formed by the secret key K of k bits and an IV specifying the algorithm. The 64-bit IV of Ascon-PRF specifies the algorithm parameters in a similar format as for Ascon, including k and the rate r_o each written as an 8-bit integer, round number a encoded as an 8-bit integer as $2^7 + a = (1\|0^7) \oplus a = \mathtt{80} \oplus a$, followed by a zero byte and the maximum output bitsize t as a 32-bit integer, or $t = 0$ for arbitrarily long output:

$$\mathrm{IV}_{k,r_\mathrm{o},a,t} \leftarrow k \,\|\, r_\mathrm{o} \,\|((1\|0^7) \oplus a) \,\|\, 0^8 \,\|\, t$$
$$S \leftarrow \mathrm{IV}_{k,r_\mathrm{o},a,t} \,\|\, K \,\|\, 0^{256-k}$$

In the initialization, the a-round permutation p^a is applied to the initial state:

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

Absorb Message. The PRF processes the padded message M in blocks of r_i bits. The padding process appends a single 1 and the smallest number of 0s to M such that the length of the padded message is a multiple of r_i bits. The resulting padded message is split into s blocks $M_1 \,\|\, \ldots \,\|\, M_s$ of $r_\mathrm{i} = 256$ bits:

$$M_1, \ldots, M_s \leftarrow r_\mathrm{i}\text{-bit blocks of } M \,\|\, 1 \,\|\, 0^{r_\mathrm{i}-1-(|M| \bmod r_\mathrm{i})}$$

The message blocks M_i with $i = 1, \ldots, s-1$ are processed as follows. Each block M_i is XORed to the state S, followed by an application of the a-round

Algorithm 1: Pseudorandom function $\mathcal{G}_{k,r_o,a,t}$ specified for ASCON-PRF with key size $k = 128$, output rate $r_o = 128$ bits, rounds $a = 12$, and any given output size limit $t < 2^{32}$ (or $t = 0$ for unlimited output). Only $\ell \leq t$ bits of output are actually produced and returned.

Pseudorandom Function $\mathcal{G}_{k,r_o,a,t}(K, M, \ell) = T$

Input: key $K \in \{0,1\}^k$,
input $M \in \{0,1\}^*$,
output bitsize $\ell \leq t$ or ℓ arbitrary if $t = 0$
Output: output $T \in \{0,1\}^\ell$

Initialization
$S \leftarrow p^a(\mathrm{IV}_{k,r_o,a,t} \,\|\, K \,\|\, 0^{256-k})$
Absorbing
$M_1 \ldots M_s \leftarrow r_i$-bit blocks of $M\|1\|0^*$
for $i = 1, \ldots, s-1$ **do**
$S \leftarrow p^a(S \oplus (M_i \,\|\, 0^{320-r_i}))$
$S \leftarrow p^a(S \oplus (M_i \,\|\, 0^{320-r_i}) \oplus (0^{319} \,\|\, 1))$
Squeezing
$u = \lceil \ell / r_o \rceil$
for $i = 1, \ldots, u-1$ **do**
$T_i \leftarrow \lfloor S \rfloor_{r_o}$
$S \leftarrow p^a(S)$
$T_u \leftarrow \lfloor S \rfloor_{r_o}$
return $\lfloor T_1 \,\|\, \ldots \,\|\, T_u \rfloor_\ell$

---

permutation p^a to S. For the last message block M_s, a single 1 is xored to the state in addition to the message block followed by an application of the a-round permutation p^a to S:

$$S \leftarrow \begin{cases} p^a(S \oplus (M_i \,\|\, 0^{320-r_i})) & \text{if } 1 \leq i \leq s-1 \\ p^a(S \oplus (M_i \,\|\, 0^{320-r_i}) \oplus (0^{319} \,\|\, 1)) & \text{if } i = s \end{cases}$$

Squeeze Tag. Then the output is extracted from the state in r_o-bit blocks T_i until the requested output length $\ell \leq t$ (or ℓ arbitrary if $t = 0$) is completed after $u = \lceil \ell / r_o \rceil$ blocks. After each extraction (except the last one), the internal state S is transformed by the a-round permutation p^a:

$$\begin{aligned} T_i &\leftarrow \lfloor S \rfloor_{r_o} \\ S &\leftarrow p^a(S), \qquad 1 \leq i \leq u = \lceil t / r_o \rceil \end{aligned}$$

The last output block T_u is truncated to $\ell \bmod r_o$ bits and $\lfloor T_1 \,\|\, \ldots \,\|\, T_u \rfloor_\ell$ is returned.

2.5 Message authentication

Ascon-MAC calls the PRF $\mathcal{G}_{k,r_o,a,t}$ as specified in Algorithm 2. For verification, it checks if the transmitted tag T matches the computed tag T^*.

Algorithm 2: Authentication and verification procedures $\mathcal{A}_{k,r_o,a,t}(K,M,T)$, $\mathcal{V}_{k,r_o,a,t}(K,M,T)$ specified for Ascon-MAC with the same parameters as the PRF $\mathcal{G}_{k,r_o,a,t}$ in Algorithm 1, but fixing $t = 128$.

Authentication $\mathcal{A}_{k,r_o,a,t}(K,M)$	Verification $\mathcal{V}_{k,r_o,a,t}(K,M,T)$
Input: key $K \in \{0,1\}^k$, message $M \in \{0,1\}^*$	**Input:** key $K \in \{0,1\}^k$, message $M \in \{0,1\}^*$, tag $T \in \{0,1\}^t$
Output: tag $T \in \{0,1\}^t$	**Output:** pass or fail
$T \leftarrow \mathcal{G}_{k,r_o,a,t}(K,M,t)$ **return** T	$T^* \leftarrow \mathcal{G}_{k,r_o,a,t}(K,M,t)$ **if** $T = T^*$ **return** pass **else return** fail

2.6 Short-Input Pseudorandom Functions

Ascon-PRFshort uses a single permutation call p^a and resembles the initialization of Ascon-128 with a different IV and the nonce replaced by a single message block M of length $m \leq 128$ bits. The PRF is illustrated in Figure 2 and specified in Algorithm 3.

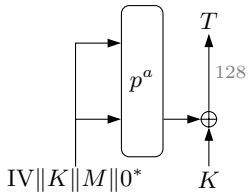


Fig. 2: PRF $\mathcal{F}_{k,m,a,t}$ for short inputs

As shown in Algorithm 3, the 320-bit input to p^a is formed by an IV specifying the algorithm, the secret key K of k bits, and the message M of m bits. The 64-bit IV of $\mathcal{F}_{k,m,a,t}$ includes the key bitsize k, the bitsize of the input block m, and the bitsize of the output block t, each written as an 8-bit integer, and the round number a encoded as an 8-bit integer as $2^6 + a = (0\|1\|0^6) \oplus a = \mathsf{40} \oplus a$:

$$\mathrm{IV}_{k,m,a,t} \leftarrow k \,\|\, m \,\|((0\|1\|0^6) \oplus a) \,\|\, t \,\|\, 0^{32}$$

Algorithm 3: Short-input PRF $\mathcal{F}_{k,m,a,t}(K, M) = T$ specified for ASCON-PRFSHORT with keysize $k = 128$ and $a = 12$ rounds. In an implementation, $m \leq 128$ and $t \leq 128$ can be inputs (instead of parameters).

Pseudorandom function $\mathcal{F}_{k,m,a,t}(K, M) = T$

Input: key $K \in \{0,1\}^k$,
input $M \in \{0,1\}^m$
Output: output $T \in \{0,1\}^t$

$S \leftarrow p^a(\mathrm{IV}_{k,m,a,t} \,\|\, K \,\|\, M \,\|\, 0^{256-k-m})$
$T \leftarrow \lceil S \rceil^t \oplus \lceil K \rceil^t$
return T

This IV is concatenated with the secret key K and the message M. Note that no padding is applied to M and hence, $\mathcal{F}_{k,m,a,t}$ only accepts M matching the length m. The permutation p^a is applied to this state:

$$S \leftarrow p^a(\mathrm{IV}_{k,m,a,t} \,\|\, K \,\|\, M \,\|\, 0^{256-k-m})$$

The XOR of the last t bits of the state with the key K are then extracted as the tag T, similarly to the authenticated encryption schemes:

$$T \leftarrow \lceil S \rceil^t \oplus \lceil K \rceil^t$$

3 Permutation

The main component of the PRFs is the 320-bit permutation p^a specified for ASCON (v1.2). We recite the specification of the permutations [DEMS21a] here for the sake of completeness. The permutation iteratively applies an SPN-based round transformation p for a rounds that in turn consists of three steps p_C, p_S, p_L:

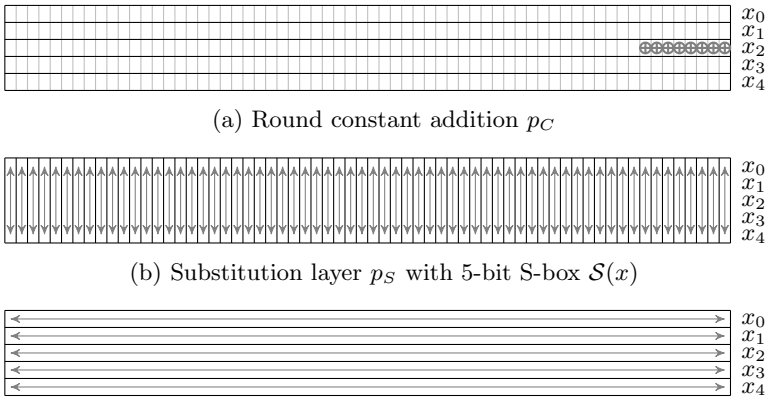
$$p = p_L \circ p_S \circ p_C \,.$$

For the description and application of the round transformations, the 320-bit state S is split into five 64-bit words x_i as follows: $S = x_0 \,\|\, x_1 \,\|\, x_2 \,\|\, x_3 \,\|\, x_4$ (see Figure 3).

3.1 Addition of Constants

The step p_C adds a round constant c_r to word x_2 of the state S in round i (see Figure 3a), where we use $r = i$ for p^a (see Table 3):

$$x_2 \leftarrow x_2 \oplus c_r \,.$$



(a) Round constant addition p_C

(b) Substitution layer p_S with 5-bit S-box $\mathcal{S}(x)$

(c) Linear layer with 64-bit diffusion functions $\Sigma_i(x_i)$

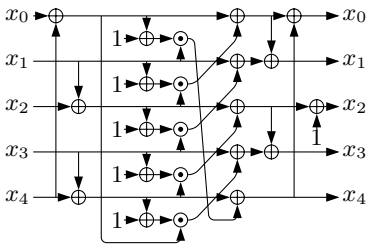
Fig. 3: The five 64-bit words of the 320-bit state S and operations $p_L \circ p_S \circ p_C$.

Table 3: The round constants c_r used in each round i of p^a.

p^{12}	Constant c_r	p^{12}	Constant c_r
0	00000000000000f0	6	0000000000000096
1	00000000000000e1	7	0000000000000087
2	00000000000000d2	8	0000000000000078
3	00000000000000c3	9	0000000000000069
4	00000000000000b4	10	000000000000005a
5	00000000000000a5	11	000000000000004b

3.2 Substitution Layer

The step p_S updates the state S with 64 parallel applications of the 5-bit S-box $\mathcal{S}(x)$ to each bit-slice of the five words $x_0, \ldots, x_4$ (Figure 3b). The operations



(a) Ascon's 5-bit S-box $\mathcal{S}(x)$

$$x_0 \leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$$
$$x_1 \leftarrow \Sigma_1(x_1) = x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$$
$$x_2 \leftarrow \Sigma_2(x_2) = x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$$
$$x_3 \leftarrow \Sigma_3(x_3) = x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$$
$$x_4 \leftarrow \Sigma_4(x_4) = x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$$

(b) Ascon's linear layer with 64-bit functions $\Sigma_i(x_i)$

Fig. 4: Ascon's substitution layer and linear diffusion layer.

specified in Figure 4a are performed bitsliced on the entire 64-bit words. For reference, the lookup table of $\mathcal{S}$ is given in Table 4, where x_0 is the MSB and x_4 the LSB.

Table 4: Ascon's 5-bit S-box $\mathcal{S}$ as a lookup table.

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
$\mathcal{S}(x)$	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

3.3 Linear Diffusion Layer

The linear diffusion layer p_L provides diffusion within each 64-bit word x_i (Figure 3c). It applies a linear function $\Sigma_i(x_i)$ defined in Figure 4b to each word x_i:

$$x_i \leftarrow \Sigma_i(x_i), \quad 0 \leq i \leq 4\,.$$

4 Design Rationale

While PRF and MAC functions can be generated from AEAD and hash functions, more efficient schemes based on the Ascon permutation can be constructed. Our proposed schemes are designed to provide a security level of 128 bits. The schemes follow the design strategy of previous Ascon v1.2 functions to maximize reusability.

4.1 Ascon-PRF

One major difference of Ascon-PRF compared to Ascon-128 and Ascon-128a is the missing keyed initialization and finalization. Not using a keyed initialization and finalization means that Ascon-PRF lacks the robustness of Ascon-128 and Ascon-128a in the case of a side-channel adversary or state recovery. We sacrificed this robustness in favor of throughput for Ascon-PRF. Due to the (almost) full-state absorption, Ascon-PRF cannot obtain the same robustness properties as Ascon-128 and Ascon-128a. If robustness is preferred, one can still use Ascon-128 or Ascon-128a instead.

An obvious difference of our construction compared to a sponge with full-state absorption [MRV15] is that we just do an almost full-state absorption. The reason for this is that it allows us to introduce domain separation between the phases of absorb and squeeze. This context switch allows us to get better security properties when increasing the rate r_o during squeezing to 128 bits. A formal security analysis of this effect is given by Bart Mennink [Men23].

4.2 Ascon-PRFshort

We decided to add a PRF called Ascon-PRFshort, where the input and output can be a maximum of 128 bits, because we see many use-cases that can profit from a dedicated lightweight PRF. Examples are key derivation functions or pointer authentication. Ascon-PRFshort is designed to follow the layout of Ascon-128's initialization.

4.3 Security Claim

Ascon-PRF, Ascon-PRFshort, and Ascon-MAC are designed to help provide 128-bit security against key recovery, and $\min(128, t)$-bit security against distinguishers and guessing T for a random key and newly queried M, all obeying the data limit of 2^{64} per key. Although the number of message blocks processed and the number of blocks output by the algorithms is limited to a total of 2^{64} blocks per key, we consider this as more than sufficient for lightweight applications in practice.

It is beneficial that a system or protocol implementing the MAC algorithms monitors and, if necessary, limits the number of tag verification failures per key. After reaching this limit, the verification rejects all tags. In applications that use a session key, for instance TLS, this limit might be small in practice, sometimes even one.

We emphasize that we do not require ideal properties for the permutation p^a. Non-random properties of the permutations p^a are known and are unlikely to affect the claimed security properties of the algorithm.

4.4 Performance

The main performance improvements of Ascon-PRF and Ascon-MAC is related to the larger input rate r_i of 256 bits, compared to the 64-bit rate of Ascon-128, Ascon-Hash and Ascon-Xof. Ascon-PRF, Ascon-MAC, Ascon-Hash and Ascon-Xof all have the same number of 12 rounds in the absorption phase, while Ascon-128 has 6 rounds in the absorption phase. Hence, absorption for Ascon-PRF and Ascon-MAC can be up to 2x faster than Ascon-128 or up to 4x faster than a KMAC based on Ascon-Hash in the ideal case.

During squeezing, the output rate r_o of Ascon-PRF and Ascon-MAC is reduced to 128 bits. Therefore, performance is typically halved compared to the absorption phase for long messages. For Ascon-128a and Ascon-Hasha we get slightly different numbers due to the different round numbers and rates. Compared to Ascon-128a, Ascon-PRF is about 1.3x faster.

The performance advantage of Ascon-PRFshort comes from the fact that only a single permutation call to p^a is used, while Ascon-128 requires at least two calls to p^a for a 128-bit message.

In Table 5, a preliminary software performance overview of Ascon-PRF, Ascon-PRFshort, and Ascon-MAC is given, in comparison with Ascon-128a, Ascon-128, Ascon-Hasha, and Ascon-Hash. These results show that

Table 5: Software performance in cycles per byte of Ascon-PRFshort, Ascon-PRF and Ascon-MAC compared to Ascon-128a, Ascon-128, Ascon-Hasha, and Ascon-Hash for different message lengths (in bytes).

(a) Ascon-PRFshort

Message Length	1	8	16
Intel® Core™ i5-6300U	185	23	12
ARM1176JZF-S (ARMv6)	1057	132	69

(b) Ascon-PRF and Ascon-MAC

Message Length	1	8	16	32	64	1536	long
Intel® Core™ i5-6300U	369	46	24	18	11.7	6.4	6.3
ARM1176JZF-S (ARMv6)	1769	223	117	85	57.5	31.9	31.6

(c) Ascon-128a

Message Length	1	8	16	32	64	1536	long
Intel® Core™ i5-6300U	365	47	31	19	13.5	8.0	7.8
ARM1176JZF-S (ARMv6)	1908	235	156	99	70.4	43.0	42.9

(d) Ascon-128

Message Length	1	8	16	32	64	1536	long
Intel® Core™ i5-6300U	367	58	35	23	17.6	11.9	11.4
ARM1176JZF-S (ARMv6)	1921	277	167	112	83.7	57.2	56.8

(e) Ascon-Hasha and Ascon-Xofa

Message Length	1	8	16	32	64	1536	long
Intel® Core™ i5-6300U	550	83	49	33	23.7	15.6	15.5
ARM1176JZF-S (ARMv6)	2390	356	211	138	100.7	65.7	65.3

(f) Ascon-Hash and Ascon-Xof

Message Length	1	8	16	32	64	1536	long
Intel® Core™ i5-6300U	747	114	69	46	34.2	23.2	23.1
ARM1176JZF-S (ARMv6)	3051	462	277	184	137.3	92.6	92.2

a dedicated Ascon-MAC can be more than 3 times faster than a KMAC alternative using Ascon-Hash or Ascon-Xof. More Ascon implementations and performance benchmarks can be found at: `https://github.com/ascon/ascon-c`.

5 Relation to Existing Cryptanalysis

Ascon and its permutation have been the subject of extensive cryptanalysis efforts by the community. In this section, we provide a preliminary discussion of how the existing results relate to Ascon-PRF, Ascon-PRFshort, and Ascon-MAC. We focus on results that are applicable in the specified modes of operation and omit pure permutation distinguishers that are not relevant in this setting, such as non-random permutation properties like inside-out zero-sum [HPTY23] or zero-sum partition [DEMS15] distinguishers. For a detailed overview of existing analysis of Ascon, including the isolated permutation, we refer to [DEMS21a] and [DEMS22].

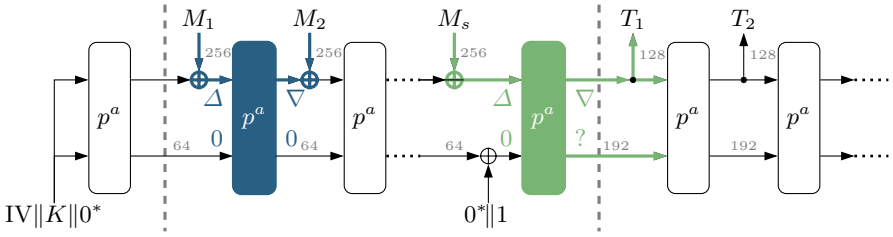
5.1 Ascon-PRF and Ascon-MAC

Initialization. As outlined in the paper describing the DonkeySponge [BDPV12], the purpose of the initialization is to diffuse the key, so that every state-bit after the initialization depends in a complex manner on the secret key K. For all permutation calls of Ascon-PRF and Ascon-MAC, we use the permutation p^a with 12 rounds as also used in the initialization of the authenticated encryption schemes Ascon-128 and Ascon-128a [DEMS16]. Full diffusion is reached after 4 rounds.

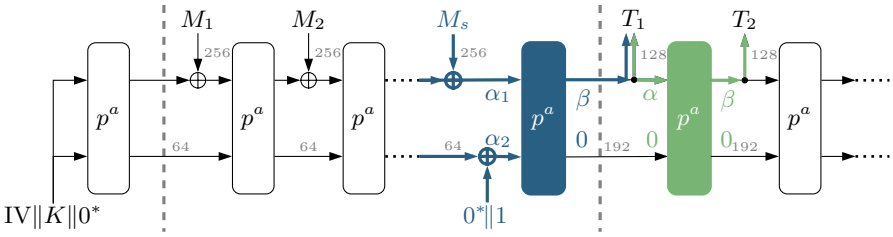
The initialization of Ascon-PRF and Ascon-MAC is invertible, unlike the doubly keyed initialization of the authenticated encryption schemes Ascon-128 and Ascon-128a. While the authenticated encryption schemes maintain resilience under misuse attacks that lead to state recovery during data processing (e.g., side-channel attacks), this is not the case for the more efficient schemes Ascon-PRF and Ascon-MAC. In applications where such robustness properties are desired for the authentication scheme, we recommend to use Ascon-128 or Ascon-128a and absorb the message as associated data.

Absorbing. In order to withstand attacks, it should be hard for an adversary to find pairs of messages M, M' that lead to the same internal state without knowing the secret key. Such a collision would allow the adversary to forge the output for M' after querying M. Additionally, an adversary could attempt to recover (parts of) the internal state by testing whether collisions occur in a conditional differential approach, and then invert the initialization to recover the key.

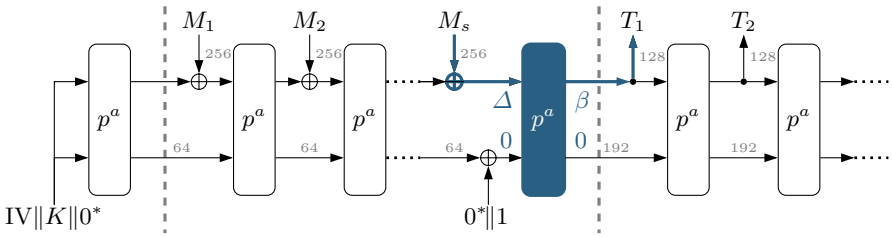
To prevent both of these, the permutation p^a should have good differential properties. Recent work by Erlacher et al. [EME22] and El Hirch et al. [EMMD22] provides provable bounds for linear and differential characteristics of the Ascon



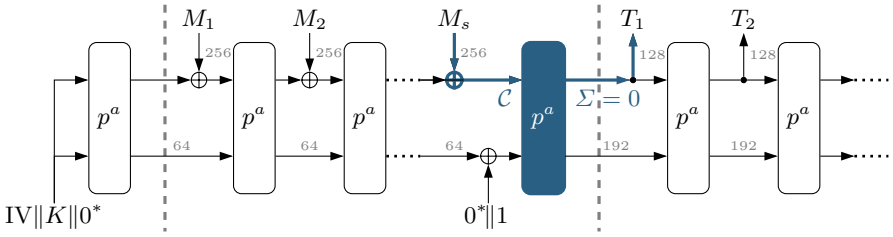
(a) Forgery (■) or MAC forgery (■) via differential cryptanalysis



(b) State recovery (■) or PRF distinguisher (■) via linear cryptanalysis



(c) Distinguisher (■) via differential-linear cryptanalysis



(d) Distinguisher or MAC forgery (■) via integral cryptanalysis

Fig. 5: Cryptanalysis approaches for Ascon-PRF and Ascon-MAC

Table 6: A 4-round truncated differential characteristic $(\Delta, 0) \to (\nabla, 0)$ with 44 active S-boxes during absorbing.

r	x_i	Differences	Differences and conditions (binary)	$\#\mathcal{S}$
0	x_0	0000000000040000	-0---0-----0--00--0--00-00--------0---00-00-0x--00--0-0-0----0--	23
	x_1	441326c02368ca84	-x---x-----x--xx--x--xx-xx--------x---xx-xx-x1--xx--x-x-x----x--	
	x_2	0000000000000000	-x---x-----x--xx--x--xx-xx--------x---xx-xx-x1--xx--x-x-x----x--	
	x_3	441326c0236cca84	-x---x-----x--xx--x--xx-xx--------x---xx-xx-xx--xx--x-x-x----x--	
	x_4	0000000000000000	-0---0-----0--00--0--00-00--------0---00-00-00--00--0-0-0----0--	
1	x_0	8040000000040000	x--------x------------------------------------x----------------	3
	x_1	0000000000000000	1--------1------------------------------------1----------------	
	x_2	0000000000000000	1--------1------------------------------------1----------------	
	x_3	8040000000040000	x--------x------------------------------------x----------------	
	x_4	0000000000000000	0--------0------------------------------------0----------------	
2	x_0	0000100004040000	----------------x---------------x-------x----------------------	3
	x_1	0000000000000000	----------------0---------------0-------0----------------------	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
3	x_0	c040100006050080	xx------x--------x---------------xx------x-x--------x-------	15
	x_1	0008980024240020	00-------0--x---x--xx-------------x--x0---x--x-0--------0-x-----	
	x_2	0000000000000000	-----------0---0--00------------0--0----0--0-----------0-----	
	x_3	0000000000000000	-----------1---0--11------------0--0----0--0-----------0-----	
	x_4	0000000000000000	----------------1---------------0-------0----------------------	
4	x_0	0000000400040040	------------------------x---------------x----------x------	
	x_1	c2040200328d8496	xx----x---------x-x--x---------xxx--x-x-x-x--xx----x--x----xx-	
	x_2	????????????????	??	
	x_3	????????????????	??	
	x_4	????????????????	??	

permutation. More specifically, any characteristic for the 6-round Ascon permutation has at least 54 differentially active S-boxes [EME22] and differential probability less than 2^{-128} [EMMD22]. Characteristics for the full 12-round permutation p^a have at least 108 active S-boxes [EME22] and differential probability less than 2^{-256} [EMMD22]. Hence, collisions during absorption for a secret key are very unlikely, even more so within the data limit of 2^{64} blocks.

Moreover, these bounds are likely not tight, particularly for constrained differentials $(\Delta, 0) \to (\nabla, 0)$ with zero difference in the $(320 - r_\mathrm{i})$-bit inner part (see Figure 5a). Not all characteristics can be transformed to be consistent with these constraints: For each S-box transition in the first round, out of 32 possible S-box output differences, 23 are easy to transform, i.e., at least one of its optimal input differences is of the form `????0` and is thus compatible. For the remaining 9 output differences, all optimal input differences are of the form `????1`, but there is a suitable difference which only decreases the probability by a factor of 2. For the S-box transitions in the last round, out of 32 possible S-box input differences, 28 are easy to transform, while 4 have no compatible output difference of the form `????0` at all. We used a heuristic search tool similar to the one used in the analysis of SHA-2 [MNS11, MNS13, EMS14] to find compatible characteristics. In Table 9, we provide a characteristic for 4 rounds with 44 active S-boxes and one for 5 rounds with 72 active S-boxes.

Switch between Absorbing and Squeezing. With the last message block, an attacker can potentially manipulate a constant incoming state and observe the

Table 7: A 3-round linear characteristic $(\alpha_1, \alpha_2) \rightarrow (\beta, 0)$ with 13 active S-boxes and bias 2^{-16} between absorbing and squeezing.

r	x_i	Masks	Masks (binary)	$\#\mathcal{S}$
0	x_0	0000022000000000	----------x---x----------	9
	x_1	0000800240020401	----------x----------x--x----------x------x---------x	
	x_2	0000800240820401	----------x----------x--x------x-----x------x---------x	
	x_3	0000022000000000	----------x---x----------	
	x_4	0000000000000000	----------	
1	x_0	0000000000000000	----------	3
	x_1	0000000000000000	----------	
	x_2	0000000000000000	----------	
	x_3	0000000040800001	----------x------x----------x	
	x_4	0000000040800001	----------x------x----------x	
2	x_0	0000000000000000	----------	1
	x_1	0000000000000000	----------	
	x_2	0000000000000000	----------	
	x_3	0000000000000000	----------	
	x_4	0000000000800000	----------x----------	
3	x_0	6925b76d24c9125b	-xx-x--x--x--x-xx-xx-xxx-xx-xx-x--x--x--xx--x--x---x--x--x-xx-xx	
	x_1	443dd4b48f9974dc	-x---x----xxxx-xxx-x-x--x-xx-x--x---xxxxx--xx--x-xxx-x--xx-xxx--	
	x_2	0000000000000000	----------	
	x_3	0000000000000000	----------	
	x_4	0000000000000000	----------	

effects of this manipulation on the first output block after an application of the permutation p^a. This is a similar scenario as we have during the initialization of the authenticated encryption schemes Ascon-128 and Ascon-128a (with empty associated data).

The attacks penetrating the highest number of rounds in case of the authenticated encryption schemes are 7 out of 12 in the case of algebraic attacks [LZWW17, LDW17, RHSS21, RS21] followed by differential-linear attacks on up to 5 out of 12 rounds [DEMS15, Tez20]. Here, the setting is slightly more difficult for the attacker, who cannot exploit known or freely controllable variables as in the IV and nonce, but only differentially influence the state. On the other hand, the attacker can differentially control a larger part of the state. The adversary can use such an integral or differential-linear distinguisher to distinguish the output stream, to facilitate forgery attacks on the first output block T_1, or to help recover the state after absorbing some fixed message $(M_1, \ldots, M_{s-1})$ and thus derive the key K. We expect that a similar number of rounds can also be attacked in the case of Ascon-PRF and Ascon-MAC.

An adversary could also increase the success probability of forgery attempts on the first output block T_1 or recover the internal state with the help of a high-probability (truncated) differential distinguisher $(\Delta, 0) \rightarrow (\nabla, *)$ or a high-correlation linear distinguisher $(\alpha_1, \alpha_2) \rightarrow (\beta, 0)$. Based on the available bounds [EMMD22] and cryptanalytic results, we expect that these techniques are less efficient than integral attacks. The truncated differential characteristic in Table 6 yields a forgery attack on Ascon-MAC with p^a reduced to 4 rounds: When considering the truncated transition probability in the last round, the differential probability of the truncated characteristic is 2^{-112}. We provide suitable linear characteristics for 3 rounds in Table 7, as well as for 4 and 5 rounds in

Table 10. These characteristics were obtained by using a heuristic search tool similar to the one used in [DEM15]. Since all characteristics are rotationally invariant, the 3-round characteristic with bias 2^{-16} in Table 7 can be rotated to obtain 64 similar characteristics. These could be used to distinguish the output stream based on a set of messages where only the last message block M_s varies, or to minimally improve the success probability of forging the first output block T_1. Overall, we expect that 12 rounds provide a comfortable security margin.

Table 8: A 3-round linear characteristic $(\alpha, 0) \rightarrow (\beta, 0)$ with 47 active S-boxes and bias 2^{-81} during squeezing.

r	x_i	Masks	Masks (binary)	$\#\mathcal{S}$
0	x_0	0cc60086530a1c53	----xx--xx---xx--------x----xx--x-x--xx----x-x----xxx---x-x--xx	29
	x_1	a0201a0000000020	x-x-------x--------xx-x-----------------------------------x-----	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
1	x_0	0422100203400821	-----x----x---x----x---------x------xx-x---------x-----x----x	14
	x_1	0420000202400834	-----x----x-----------------x-------x--x---------x-----xx-x--	
	x_2	0022000201000811	----------x---x-------------x-------x----------x------x---x	
	x_3	0820100201000001	----x-----x--------x---------x-------x----------------------x	
	x_4	0000000000000000	--	
2	x_0	0402000002000020	-----x--------x----------------------x------------------x-----	4
	x_1	0000000002000020	---------------------------------x----------------x-----	
	x_2	0400000000000000	-----x--	
	x_3	0402000000000000	-----x--------x---	
	x_4	0000000000000000	--	
3	x_0	72dbb16fda762deb	-xxx--x-xx-xx-xxx-xx---x-xx-xxxxxx-xx-x--xxx-xx---x-xx-xxxx-x-xx	
	x_1	9ffe260e3a5807c5	x--xxxxxxxxxxx---x--xx-----xxx---xxx-x--x-xx-------xxxxx---x-x	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	

Squeezing. One relevant attack vector during squeezing is linear cryptanalysis, which could be used to detect a bias in the output blocks. For this, the attacker needs good linear approximations for constrained masks $(\alpha, 0) \rightarrow (\beta, 0)$ with no active bits in the $320 - r_o$-bit inner part. Due to the smaller output rate $r_o = 128$, these characteristics are quite constrained. We used a heuristic search tool to find such constrained characteristics and provide results for 3 rounds in Table 8, as well as for 4 and 5 rounds in Table 11. Even this 3-round characteristic already has 47 active S-boxes. Based on the available bounds [EMMD22] and results, the 12-round permutation p^a should provide a very high security margin.

5.2 Ascon-PRFshort

Ascon-PRFshort is very similar to the initialization of Ascon-128a followed by the first block of encryption in the absence of associated data. The main difference is that a different part of the state is returned: in Ascon-128a, the adversary receives the first two words of the state; in Ascon-PRFshort, the output instead

covers the last two words of the state and is additionally masked with the key. Additionally, the constant IV is different, which only has a very minor impact on the applicable attacks. Hence, we expect that similar attacks that target the initialization of Ascon and work for up to 7 out of 12 rounds [LZWW17,LDW17, RHSS21,RS21] also apply in the case of Ascon-PRFshort. In particular, this includes different variations of cube attacks for key recovery. Rohit et al. [RHSS21] proposed key recovery attacks on 7-round Ascon-128a with a time complexity of 2^{123} that respect the data limit of 2^{64} blocks per key, as well as distinguishing attacks based on 60-dimensional cubes with a time complexity of 2^{60}. While this attack details depends on the specific constant IV and the position of the output bits, it is likely that comparable attacks are possible for Ascon-PRFshort. For example, based on [RHSS21, Table 3], the complexity of the distinguisher can be reduced from 2^{60} to 2^{59} when using state word X_4 instead of X_0 or X_1. In a weak-key setting, the data complexity can be decreased further [RS21].

6 Conclusion

In this paper, we complement the existing functionality of the Ascon cipher suite with the fast and efficient pseudo-random functions Ascon-PRF and Ascon-PRFshort. PRFs are very versatile building blocks allowing to instantiate various cryptographic functionalities. We think that this additional functionality can be very useful in practice.

Ascon-PRF can be used to instantiate, e.g., schemes providing authentication or confidentiality. Examples are general-purpose lightweight and efficient message authentication, stream ciphers or to instantiate SIV. Ascon-PRFshort excels whenever short data needs to be authenticated, e.g., the authentication of pointers, in challenge-response protocols, or key derivation functions (KDFs). Since Ascon-PRFshort efficiently maps 128-bit inputs to 128-bit outputs, it may even be used as a PRF in some block cipher modes.

Acknowledgments. The authors would like to thank all researchers contributing to the design, analysis, and implementation of Ascon. In particular, we want to thank Hannes Gross and Robert Primas for all their support and various implementations of Ascon. Furthermore, we want to thank Bart Mennink for giving feedback on this document.

References

BDPV07. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. Ecrypt Hash Workshop 2007, 2007. URL: `http://sponge.noekeon.org/SpongeFunctions.pdf`.

BDPV12. Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Permutation-based encryption, authentication and authenticated encryption. DIAC 2012, 7 2012.

CAE14. CAESAR committee. CAESAR: Competition for authenticated encryption: Security, applicability, and robustness, 2014. URL: https://competitions.cr.yp.to/caesar-submissions.html.

DEM15. Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Heuristic tool for linear cryptanalysis with applications to CAESAR candidates. In *ASIACRYPT (2)*, volume 9453 of *LNCS*, pages 490–509. Springer, 2015.

DEM+20. Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0. *IACR Transactions on Symmetric Cryptology*, 2020(S1):390–416, 2020. doi:10.13154/tosc.v2020.iS1.390-416.

DEM+21. Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, Bart Mennink, Robert Primas, and Thomas Unterluggauer. Isap v2.0 (Submission to NIST). Finalist of NIST lightweight cryptography standardization process, https://csrc.nist.gov/Projects/Lightweight-Cryptography/, 2021.

DEMS14. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1. Submission to the CAESAR competition, 2014. URL: https://ascon.iaik.tugraz.at.

DEMS15. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Cryptanalysis of Ascon. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 371–387. Springer, 2015. arXiv:2015/030, doi:10.1007/978-3-319-16715-2_20.

DEMS16. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2. Submission to Round 3 of the CAESAR competition, 2016. URL: https://ascon.iaik.tugraz.at.

DEMS21a. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2: Lightweight authenticated encryption and hashing. *Journal of Cryptology*, 34(3):33, 2021. doi:10.1007/s00145-021-09398-9.

DEMS21b. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Ascon v1.2 (Submission to NIST). Finalist of NIST lightweight cryptography standardization process, https://csrc.nist.gov/Projects/Lightweight-Cryptography/, 2021.

DEMS22. Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer. Status update on Ascon v1.2. Technical Report, 2022. URL: https://csrc.nist.gov/csrc/media/Projects/lightweight-cryptography/documents/finalist-round/status-updates/ascon-update.pdf.

DMV17. Joan Daemen, Bart Mennink, and Gilles Van Assche. Full-state keyed duplex with built-in multi-user support. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017*, volume 10625 of *LNCS*, pages 606–637. Springer, 2017. arXiv:2017/498, doi:10.1007/978-3-319-70697-9_21.

EME22. Johannes Erlacher, Florian Mendel, and Maria Eichlseder. Bounds for the security of Ascon against differential and linear cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2022(1):64–87, 2022. doi:10.46586/tosc.v2022.i1.64-87.

EMMD22. Solane El Hirch, Silvia Mella, Alireza Mehrdad, and Joan Daemen. Improved differential and linear trail bounds for ASCON. *IACR Trans. Symmetric Cryptol.*, 2022(4):145–178, 2022. arXiv:2022/1377, doi:10.46586/tosc.v2022.i4.145-178.

EMS14. Maria Eichlseder, Florian Mendel, and Martin Schläffer. Branching heuristics in differential collision search with applications to SHA-512. In *FSE*, volume 8540 of *LNCS*, pages 473–488. Springer, 2014.

HPTY23. Kai Hu, Thomas Peyrin, Quan Quan Tan, and Trevor Yap. Revisiting higher-order differential-linear attacks from an algebraic perspective. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023*, volume 14440 of *LNCS*, pages 405–435. Springer, 2023. doi:10.1007/978-981-99-8727-6_14.

LDW17. Zheng Li, Xiaoyang Dong, and Xiaoyun Wang. Conditional cube attack on round-reduced ASCON. *IACR Transactions on Symmetric Cryptology*, 2017(1):175–202, 2017. arXiv:2017/160, doi:10.13154/tosc.v2017.i1.175-202.

LZWW17. Yanbin Li, Guoyan Zhang, Wei Wang, and Meiqin Wang. Cryptanalysis of round-reduced ASCON. *SCIENCE CHINA Information Sciences*, 60(3):38102, 2017. doi:10.1007/s11432-016-0283-3.

Men23. Bart Mennink. Understanding the duplex and its security. *IACR Transactions on Symmetric Cryptology*, 2023(2):1–46, Jun. 2023. doi:10.46586/tosc.v2023.i2.1-46.

MNS11. Florian Mendel, Tomislav Nad, and Martin Schläffer. Finding SHA-2 characteristics: Searching through a minefield of contradictions. In *ASIACRYPT*, volume 7073 of *LNCS*, pages 288–307. Springer, 2011.

MNS13. Florian Mendel, Tomislav Nad, and Martin Schläffer. Improving local collisions: New attacks on reduced SHA-256. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 262–278. Springer, 2013.

MRV15. Bart Mennink, Reza Reyhanitabar, and Damian Vizár. Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015*, volume 9453 of *LNCS*, pages 465–489. Springer, 2015. doi:10.1007/978-3-662-48800-3_19.

MSP+23. John Preuß Mattsson, Göran Selander, Santeri Paavolainen, Ferhat Karakoç, Marco Tiloca, and Robert Moskowitz. Proposals for standardization of the ascon family. Sixth Lightweight Cryptography Workshop https://csrc.nist.gov/csrc/media/Events/2023/lightweight-cryptography-workshop-2023/documents/accepted-papers/03-proposals-for-standardization-of-ascon-family.pdf, 2023.

Nat18. National Institute of Standards and Technology. Submission requirements and evaluation criteria for the lightweight cryptography standardization process. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf, 2018.

NIS23. NIST Lightweight Cryptography Team. Lightweight cryptography standardization process: Nist selects ascon. https://csrc.nist.gov/News/2023/lightweight-cryptography-nist-selects-ascon, 2023.

RHSS21. Raghvendra Rohit, Kai Hu, Sumanta Sarkar, and Siwei Sun. Misuse-free key-recovery and distinguishing attacks on 7-round ascon. *IACR Transactions on Symmetric Cryptology*, 2021(1):130–155, 2021. doi:10.46586/tosc.v2021.i1.130-155.

RS06. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 373–390. Springer, 2006. doi:10.1007/11761679_23.

RS21. Raghvendra Rohit and Santanu Sarkar. Diving deep into the weak keys of round reduced ascon. *IACR Transactions on Symmetric Cryptology*, 2021(4):74–99, 2021. doi:10.46586/tosc.v2021.i4.74-99.

Tez20. Cihangir Tezcan. Analysis of Ascon, DryGASCON, and Shamash permutations. *International Journal of Information Security Science*, 9(3):172–187, 2020. arXiv:2020/1458.

Table 9: Differential characteristics for constrained differentials $(\Delta, 0) \to (\nabla, 0)$

(a) 4-round characteristic with 44 active S-boxes ($\#\mathcal{S}$)

r	x_i	Differences	Differences and conditions (binary)	$\#\mathcal{S}$
0	x_0	0000000000040000	-0---0-----0--00--0--00-00--------0---00-00-0x--00--0-0-0----0--	23
	x_1	441326c02368ca84	-x---x-----x--xx--x--xx-xx--------x---xx-xx-x1--xx--x-x-x----x--	
	x_2	0000000000000000	-x---x-----x--xx--x--xx-xx--------x---xx-xx-x1--xx--x-x-x----x--	
	x_3	441326c0236cca84	-x---x-----x--xx--x--xx-xx--------x---xx-xx-xx--xx--x-x-x----x--	
	x_4	0000000000000000	-0---0-----0--00--0--00-00--------0---00-00-00--00--0-0-0----0--	
1	x_0	8040000000040000	x-------x-------------------------------x----------------------	3
	x_1	0000000000000000	1-------1-------------------------------1----------------------	
	x_2	0000000000000000	1-------1-------------------------------1----------------------	
	x_3	8040000000040000	x-------x-------------------------------x----------------------	
	x_4	0000000000000000	0-------0-------------------------------0----------------------	
2	x_0	0000100004040000	----------------x---------------x------x----------------------	3
	x_1	0000000000000000	----------------0---------------0------0----------------------	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
3	x_0	c040100006050080	xx------x--------x---------------xx------x-x-------x------	15
	x_1	0008980024240020	00-------0--x---x--xx------------x--x0---x--x-0--------0-x-----	
	x_2	0000000000000000	-----------0---0--00------------0--0----0--0-----------0-----	
	x_3	0000000000000000	-----------1---0--11------------0--0----0--0-----------0-----	
	x_4	0000000000000000	----------------1---------------0-------0----------------------	
4	x_0	0000000400040040	------------------------x---------------x---------x------	
	x_1	c2040200328d8496	xx---x---------x-x--x---------xxx--x-x-x-x--xx----x--x----xx-	
	x_2	800cf66036a69030	x-----------xx--xxxx-xx--xx-------xx-xx-x-x--xx-x--x------xx---	
	x_3	4808aa2664281830	-x--x------x---x-x-x-x---x--xx--xx--x----x-x------xx-----xx----	
	x_4	0000000000000000	--	

(b) 5-round characteristic with 72 active S-boxes ($\#\mathcal{S}$)

r	x_i	Differences	Differences and conditions (binary)	$\#\mathcal{S}$
0	x_0	0100000000000000	-------x--	5
	x_1	8100000001400004	x------x-------------------------------x-x----------------x--	
	x_2	8000000001400004	x------1-------------------------------x-x----------------x--	
	x_3	0100000000000000	-------x--	
	x_4	0000000000000000	-------1--	
1	x_0	0000000000000000	1---0--0--------0--------------------01-0----0-------------1--	9
	x_1	0900800002420000	0---x--x--------x--------------------x0-x----x-------------0--	
	x_2	0000000000000000	----0--0--------0--------------------0--0----0----------------	
	x_3	0000000000000000	1---0--0--------0--------------------01-0----0-------------1--	
	x_4	8900800003420004	x---x--x--------x--------------------xx-x----x-------------x--	
2	x_0	0000000000000000	1----------0--00--1--------------------1------------1---0---1--	10
	x_1	8002000000000020	x----------0--x0--0--------------------1------------1---1x--1--	
	x_2	0000000000000000	1------------1---0-----	
	x_3	0000000000000000	0----------1--00--1--------------------1------------1---11--1--	
	x_4	8013200001000444	x----------x--xx--x--------------------x------------x---x---x--	
3	x_0	1088c44001000464	0--x0---x--1x---xx---x1--x--1---1------x-----------1x--1xx0-x--	23
	x_1	8088200880000020	x--1----x--0x--101x--10-----x---x------0-----------00--01x1-0--	
	x_2	8001840000000030	x--1----0---0--xx11--x------0---0-----------------------0xx----	
	x_3	9801444880000060	x--xx---0--00--x1x0--x1--x--x---x-------------------1---1xx0----	
	x_4	0011024000000890	0--10---1--x0--x10---0x--x--1---1-------------------x---x00x----	
4	x_0	0806805081044454	1--1x---1----xx-x--------x-x---0x00----x-----x---x---x---x-x-x--	25
	x_1	9096901161004810	x--x0---x--x-xx0x--x----1--x---x0xx----x---------x--x0---1-x-0--	
	x_2	1897109000000000	0--xx---x--x-xxx1--x----x--x---1-00----1---------1--1----0-1----	
	x_3	9883005120044040	x--xx---x-----xx0-------0x-x---x-1x----1-----x---x--1----x-0----	
	x_4	0801004040040000	1--1x---1-----1x0--------x-0---0-x0----1-----x---1-------0-0----	
5	x_0	198a464040003448	---xx--xx---x-x--x---xx--x--------x---------------xx-x---x--x---	
	x_1	2122100e48020344	--x----x--x---x----x--------xxx--x--x---------x-------xx-x---x--	
	x_2	54a04ea2d4021c38	-x-x-x--x-x------x--xxx-x-x---x-xx-x-x--------x----xxx----xxx---	
	x_3	940140c0a0400842	x--x-x---------x-x-------xx------x-x------x----------x----x----x-	
	x_4	0000000000000000	----1-------------------------------0----------------------------	

Table 10: Linear characteristics for $(\alpha_1, \alpha_2) \to (\beta, 0)$

(a) 4-round characteristic with $\#\mathcal{S} = 44$ active S-boxes and bias 2^{-75}

r	x_i	Masks	Masks (binary)	$\#\mathcal{S}$
0	x_0	d000044000900009	xx-x---------------x---x----------x--x----------x--x	14
	x_1	0000000021000230	----------------------------x---x----------x--xx----	
	x_2	0000000021000230	----------------------------x---x----------x--xx----	
	x_3	0000000000000000	--	
	x_4	5000044000100000	-x-x---------------x---x----------------x----------	
1	x_0	0000000000000000	--	3
	x_1	8000000000800008	x-------------------------------x----------------x---	
	x_2	8000000000800008	x-------------------------------x----------------x---	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
2	x_0	0000000000000000	--	9
	x_1	0000000000000000	--	
	x_2	80000000000218a7	x-------------------------------------x---xx---x-x--xxx	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
3	x_0	0200000020100806	------x-------------------------x--------x--------x--------xx-	18
	x_1	0000000000000000	--	
	x_2	8000000000000802	x---x---------x-	
	x_3	0000000000000000	--	
	x_4	c881031100040045	xx--x---x------x-----xx---x---x-----------x----------x---x-x	
4	x_0	27b583bf0cce6dd4	--x--xxxx-xx-x-xx----xxx-xxxxxx----xx--xx--xxx--xx-xx-xxx-x-x--	
	x_1	40db8037ba8483e6	-x------xx-xx-xxx--------xx-xxxx-xxx-x-x----x--x-----xxxxx--xx-	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	

(b) 5-round characteristic with $\#\mathcal{S} = 78$ active S-boxes and bias 2^{-136}

r	x_i	Differences	Differences and conditions (binary)	$\#\mathcal{S}$
0	x_0	2800008480120201	--x-x---------------x---x--x---------x--x------x--------x	15
	x_1	0000010804200042	---------------x---x--------x----x----------x----x-	
	x_2	0000010804200042	---------------x---x--------x----x----------x----x-	
	x_3	0000000000000000	--	
	x_4	2800008080020200	--x-x---------------x------x----------x------x--------	
1	x_0	0000000000000000	--	3
	x_1	0000000400100001	----------------------------x----------x----------------x	
	x_2	0000000400100001	----------------------------x----------x----------------x	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
2	x_0	0000000000000000	--	3
	x_1	0000000000000000	--	
	x_2	0000000000000000	--	
	x_3	0000000000020401	--x------x--------x	
	x_4	0000000000000000	--	
3	x_0	6d5b69224494d848	-xx-xx-x-x-xx-xx-xx-x--x---x---x---x---x-x-x-x--xx-xx----x--x---	28
	x_1	0000000000000000	--	
	x_2	0000000000000000	--	
	x_3	0000000000000001	--x	
	x_4	0000000000000000	--	
4	x_0	48c94331c1c24239	-x--x---xx--x--x-x----xx--xx---xxx-----xxx----x--x----x---xxx--x	29
	x_1	2009201000008008	--x---------x--x--x--------x----------------x----------x----	
	x_2	28c16331c0024801	--x-x---xx-----x-xx---xx--xx---xxx----------x--x--x---------x	
	x_3	2040002040000000	--x------x----------x------x----------------------------------	
	x_4	2000204000008810	--x----------x------x----------------------x--x-----x----	
5	x_0	b90a3b90d45fa1c1	x-xxx--x----x-x---xxx-xxx--x----xx-x-x---x-xxxxxx-x----xxx-----x	
	x_1	82f23012ad744a2a	x-----x-xxxx--x---xx-------x--x-x-x-xx-x-xxx-x---x--x-x---x-x-x-	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	

Table 11: Linear characteristics for $(\alpha, 0) \to (\beta, 0)$

(a) 4-round characteristic with $\#\mathcal{S} = 68$ active S-boxes and bias 2^{-105}

r	x_i	Masks	Masks (binary)	$\#\mathcal{S}$
0	x_0	741d408c41873d05	-xxx-x-----xxx-x-x------x---xx---x-----xx----xxx--xxxx-x-----x-x	40
	x_1	00609d063064008b	--------xx-----x--xxx-x-----xx---xx-----xx--x----------x---x-xx	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
1	x_0	0200c80220811808	------x--------xx--x--------x---x----x-----x---xx------x---	22
	x_1	0462c00000010c0b	-----x---xx---x-xx----------------------------x----xx------x-xx	
	x_2	0463400220811402	-----x---xx---xx-x----------x---x----x-----x---x-x-------x-	
	x_3	0240880600001803	------x--x------x---x-------xx----------------xx--------xx	
	x_4	000000162000084b	----------------------x-xx---x---------------x----x--x-xx	
2	x_0	0001000020000000	------------x---------------x----------------------------	5
	x_1	0001001000000040	------------x----------x----------------------------x------	
	x_2	0001000000000000	------------x--	
	x_3	0201001000000040	------x-------x----------x----------------------------x------	
	x_4	0200001020000040	------x----------------x-----x----------------------x------	
3	x_0	0000000020000000	----------------------------x----------------------------	1
	x_1	0000000000000000	--	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000020000000	----------------------------x----------------------------	
4	x_0	0000000000000000	--	
	x_1	0f752d23e65d3711	----xxxx-xxx-x-x--x-xx-x--x---xxxxx--xx--x-xxx-x--xx-xxx---x---x	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	

(b) 5-round characteristic with $\#\mathcal{S} = 93$ active S-boxes and bias 2^{-145}

r	x_i	Differences	Differences and conditions (binary)	$\#\mathcal{S}$
0	x_0	57dc09af5052a314	-x-x-xxxxx-xxx------x--xx-x-xxxx-x-x-----x-x--x-x-x---xx---x-x--	44
	x_1	24212a480309d822	--x--x----x----x--x-x-x--x--x--------xx----x--xxx-xx-----x---x-	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	
1	x_0	254182a618460120	--x--x-x-x-----xx-----x-x-x--xx----xx----x---xx--------x--x-----	36
	x_1	21688230f8094962	--x----x-xx-x---x-----x---xx----xxxxx-------x--x-x--x--x-xx---x-	
	x_2	046b8232f1474943	-----x---xx-x-xxx-----x---xx--x-xxxx---x-x---xxx-x--x--x-x----xx	
	x_3	21408232304c0022	--x----x-x------x-----x---xx--x---xx------x--xx-----------x--x-	
	x_4	256392b2e0484de0	--x--x-x-xx---xxx--x--x-x-xx--x-xxx------x--x----x--xx-xxxx-----	
2	x_0	0000000040000001	----------------------------x-----------------------------x	9
	x_1	0000800240020401	---------------x------------x--x-----------x------x---------x	
	x_2	0000800240820401	---------------x------------x--x------x-----x------x---------x	
	x_3	0000022000020001	--------------------x---x-----------------x---------------x	
	x_4	0000022000020001	--------------------x---x-----------------x---------------x	
3	x_0	0000000000000000	--	3
	x_1	0000000000000000	--	
	x_2	0000000000000000	--	
	x_3	0000000040800001	----------------------------x------x-----------------------x	
	x_4	0000000040800001	----------------------------x------x-----------------------x	
4	x_0	0000000000000000	--	1
	x_1	0000000000000000	--	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000800000	------------------------------------x---------------------------	
5	x_0	6925b76d24c9125b	-xx-x--x--x--x-xx-xx-xxx-xx-xx-x--x--x--xx--x--x---x--x--x-xx-xx	
	x_1	443dd4b48f9974dc	-x---x----xxxx-xxx-x-x--x-xx-x--x-----xxxxx--xx--x-xxx-x--xx-xxx--	
	x_2	0000000000000000	--	
	x_3	0000000000000000	--	
	x_4	0000000000000000	--	