Truncated, Impossible, and Improbable Differential Analysis of Ascon

Cihangir Tezcan
Department of Mathematics, Middle East Technical University, Ankara, Turkey
Institute of Informatics, Department of Cyber Security, CYDES Laboratory, Middle East Technical University, Ankara, Turkey
Institute of Applied Mathematics, Department of Cryptography, Middle East Technical University, Ankara, Turkey
cihangir@metu.edu.tr

Keywords: Ascon, Truncated Differential, Impossible Differential, Improbable Differential, Undisturbed Bits.

Abstract: ASCON is an authenticated encryption algorithm which is recently qualified for the second-round of the Competition for Authenticated Encryption: Security, Applicability, and Robustness. So far, successful differential, differential-linear, and cube-like attacks on the reduced-round ASCON are provided. In this work, we provide the inverse of ASCON's linear layer in terms of rotations which can be used for constructing impossible differentials. We show that ASCON's S-box contains 35 undisturbed bits and we use them to construct 4 and 5-round truncated, impossible, and improbable differential distinguishers. Our results include practical 4-round truncated, impossible, and improbable differential attacks on ASCON. Our best attacks using these techniques break 5 out of 12 rounds. These are the first successful truncated, impossible, and improbable differential attacks on the reduced-round ASCON.

1 INTRODUCTION

The Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) is an ongoing cryptographic competition where authenticated encryption schemes are challenging. The first round of the competition had 56 ciphers and recently on 07.07.2015 it was announced that 29 of them qualified for the second round. It is expected that the third round candidates will be announced around June 2016 and a final portfolio will be announced at the end of 2017. However, these dates are tentative because cryptanalysis effort required to analyze candidates is unpredictable.

ASCON (Dobraunig et al., 2014) is one of the authenticated encryption schemes that made it to the second round of the CAESAR competition. Until now, this cipher is successfully analyzed against differential, differential-linear, and cube-like attacks. Currently the best key recovery attack on this scheme breaks 6 out of 12 rounds and the best forgery attack is on 4 rounds. Although the designers analyze ASCON for impossible differential attacks, they only achieve a 5-round impossible differential for the permutation. It can be used to distinguish the ASCON permutation from a random permutation but it cannot be used directly in a key recovery or forgery attack.

In this work, we first analyze ASCON's S-box and provide its undisturbed bits which can be used to construct longer truncated, impossible, or improbable differentials. Then we analyze ASCON's linear layer. We prove that its invertible and provide its inverse in terms of XOR of rotations of binary words. Then we analyze the security of ASCON against truncated, impossible, and improbable differential cryptanalysis and provide the first attacks which use these techniques. We provide truncated differential key recovery attacks on 4 and 5 rounds, impossible differential attacks on 4 rounds, and improbable differential attacks on 5 rounds of ASCON. Moreover, we provide 5 round truncated, impossible, and improbable differential distinguishers which requires much less data when compared to the impossible differential distinguisher of the designers.

This paper is organized as follows: In Sect. 2, we describe ASCON and summarize the previous cryptanalysis results on this cipher. In Sect. 3, we analyze ASCON's S-box and provide its undisturbed bits. In Sect. 4, we prove that the linear layer of ASCON is invertible and provide its inverse in terms of rotations. In Sect. 5, we provide the first truncated, impossible, and improbable differential key recovery attacks on ASCON. We conclude our paper in Sect. 6.

2 ASCON

2.1 Design

ASCON is an authenticated encryption scheme that is submitted to ongoing CAESAR competition and it qualified for the second-round. It is a substitution-permutation network and it is based on a sponge-like construction with a state size of 320 bits. ASCON's mode of operation is based on MonkeyDuplex (Daemen, 2012).

The initial design of ASCON, which is referred to as v1.0, supported two key lengths, 96 and 128 bits. However, the designers removed the 96-bit key support when tweaking for the second-round of the competition. Since 80-bit security is not suggested today, removing the 96-bit key variant is probably a good call since it may not be secure in the close future. The tweaked ASCON is referred to as v1.1 and we focus on this latest version in this paper. The tweaked version provides two recommended parameter sets referred to as ASCON-128 and ASCON-128a.

The encryption consists of four steps: Initialization, processing associated data, processing the plaintext, and finalization. The 320-bit state is represented with five 64-bit words $x_0,\dots,x_4$. The scheme uses two permutations p^a and p^b which applies the round transformation p iteratively a and b times. These steps are illustrated in Figure 1.

For ASCON-128, we have $a = 12$ and $b = 6$. For ASCON-128a we have $a = 12$ and $b = 8$. Both versions use 128-bit key, nonce and tag. However, data block size is 64 for ASCON-128 and 128 for ASCON-128a.

The round transformation of ASCON first adds a constant to x_2, applies a nonlinear substitution layer and then applies a linear layer. The substitution layer applies a 5-bit S-box 64 times in parallel. This S-box is affine equivalent to the Keccak (Bertoni et al., 2011) χ mapping and is provided in Table 1. The linear layer is actually XOR of right rotations of the 64-bit words $x_0,\dots,x_4$. The linear layer can be described as follows:

$$\begin{aligned}
\Sigma_0(x_0) &= x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)\\
\Sigma_1(x_1) &= x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)\\
\Sigma_2(x_2) &= x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)\\
\Sigma_3(x_3) &= x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)\\
\Sigma_4(x_4) &= x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)
\end{aligned}$$

2.2 Security

We can divide the attacks into two categories, forgery and key recovery. Forgery attacks focus on the finalization and key recovery attacks focus on the initialization phases of ASCON. When analysing ASCON, we can target either the initialization in a nonce-respecting scenario, or the processing of the plaintext in a nonce-misuse scenario.

In case of an attack on the finalization of ASCON, suitable characteristics may contain differences in stateword x_0 at the input of the permutation. The rest of the statewords have to be free of differences. For the output of the finalization, the only requirement is that there is some fixed difference pattern in x_3 and x_4. Knowledge about the expected differences in x_0, x_1, and x_2 at the output of the permutation is not required. When we focus on the initialization, differences are allowed in the nonce x_3, x_4 and the output is observed only for x_0 (i.e. output difference should be at x_0).

The first analysis of ASCON is done by the designers in the CAESAR competition submission document (Dobraunig et al., 2014). They provided collision-producing differentials and 5-round impossible differential for the permutation. In (Dobraunig et al., 2015), these observations are further improved to obtain 6-round cube-like, 5-round differential-linear key recovery attacks and 4-round differential forgery attack. They also provided linear and differential bounds and 12-round zero-sum distinguishers for the permutation that requires 2^{130} time complexity.

Moreover, Todo provided integral distinguishers for various numbers of rounds for the ASCON permutation (Todo, 2015).

Finally, Jovanovic et al. proved that ASCON's sponge mode is secure even for higher rates (Jovanovic et al., 2014).

3 ANALYSIS OF ASCON'S S-BOX

ASCON designers provide differential and linear properties of ASCON's S-box in (Dobraunig et al., 2014). The maximum differential probability of the S-box is 2^{-2} and its differential branch number is 3. The maximum linear probability of the S-box is 2^{-2} and its linear branch number is 3. The algebraic degree of the S-box is 2. A different 5×5 S-box with smaller maximum differential probability and linear probability could easily be chosen by the designers. However, this S-box was intentionally chosen because it requires very small area in hardware and performs very fast in software and hardware.

Definition 3.1. *(Tezcan, 2014) For a specific input difference of an S-box, if some bits of the output difference remain invariant, then we call such bits* undisturbed.

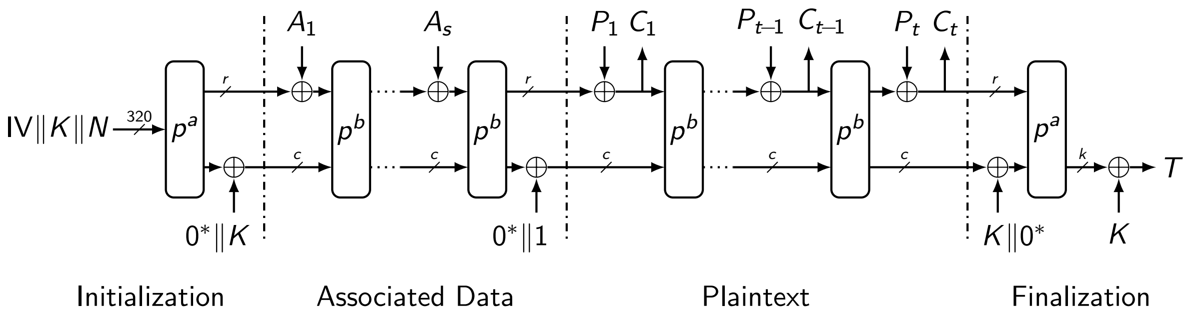

Figure 1: The encryption of ASCON. Figure is taken from the cipher's official website *http://ascon.iaik.tugraz.at/*

Table 1: ASCON's 5-bit s-box.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S(x)	4	11	31	20	26	21	9	2	27	5	8	18	29	3	6	28
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
S(x)	30	19	7	14	0	13	17	24	16	12	1	25	22	10	15	23

Definition 3.2. *(Evertse, 1987) An $n \times m$ S-Box S is said to have a* linear structure *if there exists a nonzero vector $\overline{\alpha} \in \mathbb{F}_2^n$ together with a nonzero vector $\overline{b} \in \mathbb{F}_2^m$ such that $\overline{b} \cdot S(\overline{x}) \oplus \overline{b} \cdot S(\overline{x} \oplus \overline{\alpha})$ takes the same value $c \in \mathbb{F}_2$ for all $\overline{x} \in \mathbb{F}_2^n$.*

We further analyzed this S-box for other cryptographic properties and observed that it has 91 linear structures. 35 of them corresponds to coordinate functions, thus by (Makarim and Tezcan, 2014) they are undisturbed bits in the forward direction and they are provided in Table 2. Moreover, ASCON has 2 undisturbed bits for the inverse S-box, namely 00010 →??1? and 01000 →?1???. Although the inverse S-box is not used in the encryption or decryption process, its undisturbed bits can be used when constructing impossible differentials via the miss-in-the-middle technique.

Definition 3.3. *(Tezcan and Özbudak, 2014) Let S be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. For all $x, y \in \mathbb{F}_2^n$ that satisfy $S(x) \oplus S(y) = \mu$, if we also have $S(x \oplus \lambda) \oplus S(y \oplus \lambda) = \mu$, then we say that S has a* differential factor λ *for the output difference μ. (i.e. μ remains invariant for λ).*

Recently, a new S-box property called differential factor is introduced in (Tezcan and Özbudak, 2014) which shows that some key bits may not be captured in a differential attack or its variants. This observation may be used to reduce the time complexity of the key guess step of differential attacks. On the other hand, it increases the time complexity of exhaustive search for the remaining key bits phase. Differential factors are used in (Tezcan and Özbudak, 2014) to reduce the time complexity of differential-linear attacks on SERPENT (Biham et al., 1998). Although ASCON's S-box does not have the best cryptographic properties, surprisingly it does not contain any differential factors.

Table 2: Undisturbed Bits of ASCON's S-box

Input Difference	**Output Difference**	**Input Difference**	**Output Difference**
00001	?1???	10000	?10??
00010	1???1	10001	10??1
00011	???0?	10011	0???0
00100	??110	10100	0?1??
00101	1????	10101	????1
00110	????1	10110	1????
00111	0??1?	10111	????0
01000	??11?	11000	??1??
01011	???1?	11100	??0??
01100	??00?	11110	?1???
01110	?0???	11111	?0???
01111	?1?0?		

4 ANALYSIS OF ASCON'S LINEAR LAYER

The inverse of ASCON's linear layer is not provided in (Dobraunig et al., 2014) because ASCON is a sponge construction and the inverse of this layer is not required in the decryption process. However, in order to obtain impossible differential distinguishers using the miss-in-the-middle technique, we need the inverse permutation to check differentials in the reverse order. We will also use them as filtering conditions when we are choosing plaintext-ciphertext pairs in our truncated and improbable differential attacks.

The linear layer consists of XOR of right rotations of the 64-bit words $x_0,\ldots,x_4$. Thus, the first thing to check whether such an operation is invertible or not. The following theorem shows when XOR of rotations of binary words are invertible.

Theorem 4.1. *(Rivest, 2011) If n is a power of 2, v is an n-bit word, and r_1, r_2, ..., r_k are distinct fixed integers modulo n, then the function $R(v) = R(v;r_1,r_2,\ldots,r_k) = (v \lll r_1)\oplus(v \lll r_2)\oplus\ldots(v \lll r_k)$ is invertible if and only if k is odd, where $(v \lll r)$ denotes the n-bit word v rotated left by r positions, and where '$\oplus$' denotes the bit-wise 'exclusive-or' of n-bit words.*

Theorem 4.1 shows that the linear layer of ASCON is invertible since $k = 3$ for all of the five transformations $\Sigma_0,\ldots,\Sigma_4$. If we consider n-element vectors over the finite field $\mathbb{F}_2$, one can obtain $R(v)$ by multiplying v by an $n\times n$ circulant matrix over $\mathbb{F}_2$ having k ones per row and per column. Thus, inverse of $R(v)$ can be obtained by finding the inverse of this circulant matrix via reducing it to row-reduced echelon form by means of row operations. This way we obtained the inverse of the linear layer and the right rotations required to perform the inverse linear layer is provided in Table 3.

Table 3: Linear layer of ASCON consists of XOR of rotations of binary words. Since the inverses of these operations are not required in the decryption process, they are not provided by the designers in the submission document. We provide the inverse of the linear layer which can be used for constructing impossible differentials. All of the rotations are to the right.

Permutation	Rotations	Size
Σ_0	0 19 28	3
Σ_0^{-1}	0 3 6 9 11 12 14 15 17 18 19 21 22 24 25 27 30 33 36 38 39 41 42 44 45 47 50 53 57 60 63	31
Σ_1	0 61 39	3
Σ_1^{-1}	0 1 2 3 4 8 11 13 14 16 19 21 23 24 25 27 28 29 30 35 39 43 44 45 47 48 51 53 54 55 57 60 61	33
Σ_2	0 1 6	3
Σ_2^{-1}	0 2 4 6 7 10 11 13 14 15 17 18 20 23 26 27 28 32 34 35 36 37 40 42 46 47 52 58 59 60 61 62 63	33
Σ_3	0 10 17	3
Σ_3^{-1}	1 2 4 6 7 9 12 17 18 21 22 23 24 26 27 28 29 31 32 33 35 36 37 40 42 44 47 48 49 53 58 61 63	33
Σ_4	0 7 41	3
Σ_4^{-1}	0 1 2 3 4 5 9 10 11 13 16 20 21 22 24 25 28 29 30 31 35 36 40 41 44 45 46 47 48 50 53 55 60 61 63	35

5 TRUNCATED, IMPOSSIBLE, AND IMPROBABLE DIFFERENTIAL ANALYSIS

Statistical attacks on block ciphers make use of a property of the cipher so that an event occurs with different probabilities depending on whether the correct key is used or not. We represent these probabilities with p_0 for the correct key and p for the wrong ones. For instance, differential cryptanalysis (Biham and Shamir, 1991) considers characteristics or differentials which show that a particular output difference should be obtained with a relatively high probability when a particular input difference is used. Hence, when the correct key is used, the predicted differences occur more frequently (i.e. $p_0 > p$). In a classical differential characteristic the differences are fully specified and in a truncated differential (Knudsen, 1994) only parts of the differences are specified.

On the other hand, impossible differential cryptanalysis (Biham et al., 2005) uses an impossible differential which shows that a particular difference cannot occur for the correct key (i.e. probability of this event is exactly zero). Therefore, if these differences are satisfied under a trial key, then it cannot be the correct one (i.e. $p_0 = 0$). Thus, the correct key can be obtained by eliminating all or most of the wrong keys.

Moreover, it is shown in (Tezcan, 2010) that it is also possible to obtain differentials so that the predicted differences occur less frequently for the correct key (i.e. $p_0 < p$). This new cryptanalytic technique is called the improbable differential attack and the impossible differential attack can be seen as a special case of it where $p_0 = 0$.

5.1 Truncated Differential Analysis

5.1.1 4-Round Truncated Differential Distinguisher

Undisturbed bits of ASCON's S-box allows us to obtain long truncated differentials. We first focus on probability 1 truncated differentials in order to convert them to impossible differentials via the miss-in-the-middle technique. The longest truncated differential we could find in the encryption direction with probability 1 is on 3.5-rounds of ASCON and it is provided in Table 4. By adding the permutation layer to the end, this differential can be used to distinguish 4 rounds of the permutation with only 2 chosen nonces.

Table 4: Truncated differential Δ_1 with probability 1 that covers 3.5 rounds of p in binary notation. Undisturbed bits are shown in bold. Substitution and permutation layers are denoted by S and P, respectively.

	3.5-Round Truncated Differential
	1000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
I	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	?000000000000000000000000000000000000000000000000000000000000000
S_1	?000000000000000000000000000000000000000000000000000000000000000
	?000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	?00000000000000000000000000000000000000?000000000000000000000?00
P_1	??0000?000000000000000000000000000000000000000000000000000000000
	?000000000?000000?0000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	??0000?000?000000?000000000000000000000?000000000000000000000?00
	??0000?000?000000?000000000000000000000?000000000000000000000?00
S_2	??0000?000?000000?000000000000000000000?000000000000000000000?00
	??0000?000?000000?000000000000000000000?000000000000000000000?00
	?**0**0000**0**000?000000?000000000000000000000?000000000000000000000?00
	??0?00?000?00000??0??0000?00??0000?0?0??00000?000000000000?00?00
	??0?00??00?000?00?000000000000000000?00??0000?000?000000?0?00??0
P_2	????00??00???000???0000?000000000000000??0000?000000000000000??0
	??0000??00??00?0???0?00?000?000000?0000?000000000?000000?0000?00
	?000?00?00?00000??000000?0000000000000??0?0000?0000?000000?00?00
	?????0??00???0?0?????00???0???0000?0?0????000??00?0?0000?0?00??0
	?????0??00???0?0?????00???0???0000?0?0????000??00?0?0000?0?00??0
S_3	?????0??00???0?0???**0**?00??**0**0?**00**0000?0?0????000??00?0?0000?0?00??0
	?????0??00???0?0?????00???0???0000?0?0????000??00?0?0000?0?00??0
	??**0**??0??00??**0**0?0?????00???0???0000?0?0????000??00?0?0000?0?00??0
	????????0???0???0????????0
	?????0????????????????????????0?0???????????0??0????0?0?????0???
P_3	???????????????????????????????00??????????0????0????00?????0???
	?????0??????????????????????????0??????????0???0????0?0?????0???
	?????????0??0????????????????????????0????????????0???0?????????
	??0?????????
	??0?????????
S_4	??0?????????
	??0?????????
	??0?????????

5.1.2 4-Round Truncated Differential Attack

We cannot use our 3.5-round truncated differential Δ_1 in a key recovery attack because we can only provide input differences at the words x_3 and x_4. We observe that if we provide the input difference 3_x to a single S-box, then the output difference is 1_x with probability 2^{-3}. Then with probability 1, we have 54^{th} S-box with 0_x output difference at the end of substitution layer of round 4. After the permutation layer we focus on the word x_0 because this is the only word that we can work on in an attack to the initialization phase. Thus, output differences that have the difference 0 corresponding to the most significant bit of the 54^{th} after the application of the inverse permutation provided in Table 3 are the right pairs for our attack. Since half of the output differences make that bit have 0 difference, this filtering condition has probability 1/2. Details of this differential are provided in Table 5. Since this is a probability 1 differential distinguisher, complementing the output differences provides a 4-round impossible differential distinguisher.

For a wrong key, this differential holds with probability $p = 1/2$. However, it holds with probability $p_0 = 1$ for the correct key. If we think ASCON as a block cipher where the plaintext is XORed with the key, then we can capture 2 bits of the key corresponding to the active S-box with 2^{11} data complexity and negligible time and memory complexity. Due to the symmetry of the cipher, remaining key bits can be captured by applying the same attack with shifting the input difference. However, key is not XORed with the plaintext in ASCON and the S-box input difference 3_x gives the output difference 1_x when the corresponding two key bits are 1. Hence, this attack can be used with the symmetry of the cipher to check if the two key bits corresponding to the active S-boxes are 1. Approximately 16 of them would be 1 and thus the attack should work for them. And the remaining 48 of them can be found via exhaustive search which would require 3^{48} 4-round ASCON encryptions.

Table 5: 4-round truncated differential attack. Substitution and permutation layers are denoted by S and P, respectively.

	4-Round Truncated Differential Attack
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
I	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
S_1	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
2^{-3}	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
P_1	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	1000000100000000000000000000000000000000010000000000000000000000
	?000000?000000000000000000000000000000000?0000000000000000000000
	1000000100000000000000000000000000000000010000000000000000000000
S_2	?000000?000000000000000000000000000000000?0000000000000000000000
	?000000?000000000000000000000000000000000?0000000000000000000000
	?000000?000000000000000000000000000000000?0000000000000000000000
	?0000?0?00000000000?000000?0?000000?00000?000000000000000000?000
	0000100000000000100000000000000000000011000000100000000000000100
P_2	??0000???0000?000000000000000000000000000??0000?0000000000000000
	?000000?00?000000?000000?0000000000000000?000000000?000000?00000
	?000000?000000?000?0000000000000000000000?000000?000000000000000
	??00?????0?00??0????0000?0?0?000000?00??0??000???00?000000?0??00
	??00?????0?00??0????0000?0?0?000000?00??0??000???00?000000?0??00
S_3	??0010???0?00??01??00000?0000000000000110??0001??00?000000?00100
	??001????0?00??01???0000?0?0?000000?00110??0001??00?000000?0?100
	?000??0?00?000?0????0000?0?0?000000?00??0?0000?0?00?000000?0??00
	?????????0???????????0????????00????????0????????00??0?0???0??0?
	?????????0??0???????0??????0?000??0??0????????????0???0????0????
P_3	???101???????????????01???0000?0000000011???110???0????00???0010
	??00?????0??0????????1??????????01???0?00???0?0??10??0010?????01
	??0???0?00???0??????0?0????0?00?0?0?0???0??00??0?00?0?0?0?????00
	??0?????????????
	??0?????????????
S_4	??0?????????????
	??0?????????????
	??0?????????????

5.1.3 5-Round Truncated Differential Attack

We can perform a 5-round attack on ASCON by giving 3_x input difference to 35 S-boxes and check if all of the output differences are 1_x. Thus, we need to guess

$2 \cdot 35 = 70$ bits of the key. To the bottom of these differences, we add a 4-round truncated differential that holds with probability 2^{-3} which is provided in Table 6. For a wrong key, this differential holds with probability $p = 1/2$. However, it holds with probability $p_0 = 1/2 + 1/8$ for the correct key.

If ASCON were a block cipher where the plaintext is XORed with the key, then we could perform a key recovery attack where knowledge of 2^{110} data would be enough to distinguish 70 bits of the key from the wrong ones and around 2^{101} 5-round ASCON encryptions would be required. However, key is not XORed with the plaintext in ASCON and the S-box input difference 3_x gives the output difference 1_x when the corresponding two key bits are 1. So the attack works when the key bits corresponding to the 35 active S-boxes are all 1. So the attack works for a weak key space of size $2^{128-2\cdot35} = 2^{58}$. The weak key space becomes around 2^{64} when we use the symmetry of the cipher but it is still very small compared to 2^{128}. Therefore, if the attacked key is in the weak key space, then we capture its 70 bits with negligible time complexity and recover the remaining bits with exhaustive search that requires 2^{58} 5-round ASCON encryptions. However, if the key is not in this weak key space, then the attack only becomes slightly faster than the exhaustive search, namely $2^{128} - 2^{64}$ 5-round ASCON encryptions.

Note that the whole differential provided in Table 6 can be seen as a 5-round truncated differential distinguisher with probability 2^{-107}. Hence, we can use it with 2^{109} data to distinguish the 5-round ASCON from a random permutation. Complementing the output differences provides a 5-round improbable differential distinguisher that works similar to this 5-round truncated differential.

Table 6: 5-round truncated differential attack. Substitution and permutation layers are denoted by S and P, respectively.

	5-Round Truncated Differential Attack
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
I	0000000000000000000000000000000000000000000000000000000000000000
	1111110001110100100011101100111100011000110011111010010100001101
	1111110001110100100011101100111100011000110011111010010100001101
S_1	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
2^{-105}	0000000000000000000000000000000000000000000000000000000000000000
	1111110001110100100011101100111100011000110011111010010100001101
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
P_1	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
S_2	1000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
2^{-3}	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	1000000000000000000100000000100000000000000000000000000000000000
	1000000000000000000000000000000000000001000000000000000000000100
P_2	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	0000000000000000000000000000000000000000000000000000000000000000
	?000000000000000000?00000000?0000000000?000000000000000000000?00
	?00000000000000000010000000010000000000?000000000000000000000?00
S_3	1000000000000000000000000000000000000001000000000000000000000100
	?000000000000000000?00000000?00000000001000000000000000000000100
	?000000000000000000?00000000?0000000000?000000000000000000000?00
	?00?000000000000?00?00000?00?000000000??0000000?00000000?0?00?00
	?0010000000000?010000000010000000000?00?000000000000000000?00?00
P_3	0101001000000000000000000000000000000000100001000000000000000010
	?000000100?000100?0?00000000??000000?0?000000?000100000010000000
	?000??0?00000000?00?000000?0?000000?00??0?0000?0000000000000??00
	??0?????00?000?0??0?00000??0??00000??0????000???01000000?0?0???0
	??0?????00?000?0??0?00000??0??00000??0????000???0?000000?0?0???0
S_4	?10???1?00?000?0??0?00000??0??00000??0??1?000??00?000000?0?0??10
	?10???1?00?000?0??0?00000??0??00000??0??1?000???0?000000?0?0??10
	?00???0?00?000?0??0?00000??0??00000??0??0?000???01000000?0?0??00
	????????0???0?????0??0?????0??0??????0?????0?????0000???????????
	????????00??0?????0????????0??0???0?????????????0?000?0????????0
P_4	????????????1????????0??0?????????0????????0????0????00????????1
	?0??????1???0?????1????1??????0?0????0??0???0?????1???0?????????
	????????00??????????0??????0?00???0?????0????????00????????0??0?
S_5	??????????????????????????????0?????????????????????????????????
	??????????????????????????????0?????????????????????????????????
	??????????????????????????????0?????????????????????????????????
2^{-1}	??????????????????????????????0?????????????????????????????????
	??????????????????????????????0?????????????????????????????????

Table 7: Impossible differential of (Dobraunig et al., 2014) that covers 5 rounds of p in hexadecimal notation. It holds with probability $p = 2^{-320}$ for a random permutation.

	Input difference		**Output Difference**
x_0	0000000000000000		0000000000100000
x_1	0000000000000000		0000000000000000
x_2	0000000000000000	$\nrightarrow$	0000000000000000
x_3	0000000000000000		0000000000000000
x_4	8000000000000000		0000000000000000

5.2 Impossible Differentials

ASCON's security against impossible differential attacks is discussed in (Dobraunig et al., 2014) by the designers and they obtained a 5-round impossible differential via computer search. This differential can be used to distinguish the permutation p and it is provided in Table 7. However, for a random permutation this impossible differential holds with probability $p = 2^{-320}$. Thus, one needs to use the whole codebook to use it as a distinguisher. Moreover, since the output differences are fully specified, it cannot be used in a key recovery or forgery attack.

We consider truncated differentials in the decryption direction to obtain impossible differentials by combining them with our 3.5-round truncated differential Δ_1. We cannot find such long truncated differentials in the decryption direction because a single bit difference to the permutation provides differences at more than 30 bits because of the inverse linear transformations. Moreover, the inverse of ASCON's S-box has only two undisturbed bits. The longest truncated differentials we could find covers 1.5 rounds in the decryption direction. Thus, we can use them to obtain 5-round impossible differentials using the miss-in-the-middle technique. An example of such an impossible differential is provided in Table 8. The differences are fully specified in this impossible differential, too. However, note that since the contradiction is obtained at a single bit, half of the differences given only to x_3 or x_1 at P5 still make it miss in the middle due to the undisturbed bits. Since we can give 2^{63} different differences to the x_3 or x_1, we have $p = 2^{-256}$ for this bundle of impossible differentials instead of $p = 2^{-320}$.

Table 8: An impossible differential that covers 5 rounds of p in binary notation. Substitution and permutation layers are denoted by S and P, respectively. The miss-in-the-middle is obtained by combining the 3.5-round Δ_1 in the forward direction with the 1.5-round differential in the backward direction that is provided below.

	5-Round Impossible Differential
	3.5-round truncated differential Δ_1
S_4	??0????????? ??0????????? ??0????????? ??**0**????????? ??0?????????
	Impossible
S_4	?? ?? ?? 111000010110001011001111101111110101010010100011010001**1**010100101 ??
P_4	0??0?0??0?00?0000??00????0????0???0???00?0?0?00???000?0000?00?0? 0??0?0??0?00?0000??00????0????0???0???00?0?0?00???000?0000?00?0? 0??0?0??0?00?0000??00????0????0???0???00?0?0?00???000?0000?00?0? 0110101101001000011001111011110111011100101010011100010000100101 0??0?0??0?00?0000??00????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

6 CONCLUSIONS

ASCON's S-box contains many undisturbed bits and in this study we used them to construct truncated, impossible, and improbable differentials. We provide the results of our distinguishers in Table 9. Our best attacks break 5 out of 12 rounds of ASCON and they are provided in Table 10. These attacks can be prevented by replacing ASCON's S-box with a cryptographically more secure one. However, ASCON's S-box is deliberately chosen this way mainly because of its bit-sliced implementation with few, well pipelined instructions.

Our attacks show that further analysis may provide truncated, impossible or improbable differential distinguishers or attacks on 6 or more rounds of ASCON. However, the full scheme looks resistant to these type of attacks. Thus, we conclude that the security/performance trade-off due to the choice of the S-box is well justified and the full cipher is secure against truncated, impossible, and improbable differential attacks. However, our analysis and differentials can be used to obtain better attacks when combined with other cryptanalysis techniques.

Table 9: Summary of impossible, improbable, and truncated differential distinguishers on ASCON

Rounds	**Data**	**Method**	**Source**
5/12	2^{109}	Improbable Diff.	Sect. 5.1.3
5/12	2^{109}	Truncated Diff.	Sect. 5.1.3
5/12	2^{256}	Impossible Diff.	Sect. 5.2
5/12	2^{320}	Impossible Diff.	(Dobraunig et al., 2014)
4/12	2^{2}	Impossible Diff.	Sect. 5.1.1
4/12	2^{2}	Truncated Diff.	Sect. 5.1.1

ACKNOWLEDGEMENTS

This work was supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under the grant 115E447 titled "Quasi-Differential Factors and Time Complexity of Block Cipher Attacks".

Table 10: Summary of attacks on ASCON

Type	Rounds	Time	Method	Source
Key Recovery	6/12	2^{66}	Cube-like	(Dobraunig et al., 2015)
Key Recovery	5/12	2^{35}	Cube-like	(Dobraunig et al., 2015)
Key Recovery	5/12	2^{36}	Differential-Linear	(Dobraunig et al., 2015)
Key Recovery	5/12	2^{58} or $2^{127.99}$	Truncated/Improbable	Sect. 5.1.3
Key Recovery	4/12	2^{18}	Differential-Linear	(Dobraunig et al., 2015)
Key Recovery	4/12	3^{48}	Truncated/Impossible	Sect. 5.1.2
Forgery	4/12	2^{101}	Differential	(Dobraunig et al., 2015)
Forgery	3/12	2^{33}	Differential	(Dobraunig et al., 2015)

REFERENCES

Bertoni, G., Daemen, J., Peeters, M., and Assche, G. V. (2011). The Keccak SHA-3 submission. Submission to NIST (Round 3).

Biham, E., Anderson, R. J., and Knudsen, L. R. (1998). Serpent: A new block cipher proposal. In Vaudenay, S., editor, *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings*, volume 1372 of *Lecture Notes in Computer Science*, pages 222–238. Springer.

Biham, E., Biryukov, A., and Shamir, A. (2005). Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. *J. Cryptology*, 18(4):291–311.

Biham, E. and Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *J. Cryptology*, 4(1):3–72.

Daemen, J. (2012). Permutation-based encryption, authentication and authenticated encryption. DIAC - Directions in Authenticated Ciphers.

Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2014). ASCON v1, submission to the CAESAR competition.

Dobraunig, C., Eichlseder, M., Mendel, F., and Schläffer, M. (2015). Cryptanalysis of Ascon. In Nyberg, K., editor, *Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, April 20-24, 2015. Proceedings*, volume 9048 of *Lecture Notes in Computer Science*, pages 371–387. Springer.

Eisenbarth, T. and Öztürk, E., editors (2015). *Lightweight Cryptography for Security and Privacy - Third International Workshop, LightSec 2014, Istanbul, Turkey, September 1-2, 2014, Revised Selected Papers*, volume 8898 of *Lecture Notes in Computer Science*. Springer.

Evertse, J.-H. (1987). Linear Structures in Blockciphers. In Chaum, D. and Price, W. L., editors, *EUROCRYPT*, volume 304 of *Lecture Notes in Computer Science*, pages 249–266. Springer.

Jovanovic, P., Luykx, A., and Mennink, B. (2014). Beyond 2 c/2 security in sponge-based authenticated encryption modes. In Sarkar, P. and Iwata, T., editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 85–104. Springer.

Knudsen, L. R. (1994). Truncated and higher order differentials. In Preneel, B., editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer.

Makarim, R. H. and Tezcan, C. (2014). Relating undisturbed bits to other properties of substitution boxes. In (Eisenbarth and Öztürk, 2015), pages 109–125.

Rivest, R. L. (2011). The invertibility of the XOR of rotations of a binary word. *Int. J. Comput. Math.*, 88(2):281–284.

Tezcan, C. (2010). The improbable differential attack: Cryptanalysis of reduced round CLEFIA. In Gong, G. and Gupta, K. C., editors, *Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings*, volume 6498 of *Lecture Notes in Computer Science*, pages 197–209. Springer.

Tezcan, C. (2014). Improbable differential attacks on Present using undisturbed bits. *J. Computational Applied Mathematics*, 259:503–511.

Tezcan, C. and Özbudak, F. (2014). Differential factors: Improved attacks on SERPENT. In (Eisenbarth and Öztürk, 2015), pages 69–84.

Todo, Y. (2015). Structural evaluation by generalized integral property. In Oswald, E. and Fischlin, M., editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314. Springer.