IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 4, pp. 171–191. DOI:10.13154/tosc.v2019.i4.171-191

Improved Security Evaluation of SPN Block Ciphers and its Applications in the Single-key Attack on SKINNY

Wenying Zhang[1,2] ✉, Meichun Cao[1], Jian Guo[2] and Enes Pasalic[3]

[1] School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China,
zhangwenying@sdnu.edu.cn, caomeichun@stu.sdnu.edu.cn
[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore, Singapore,
guojian@ntu.edu.sg
[3] FAMNIT, University of Primorska, Koper, Slovenia
enes.pasalic6@gmail.com

Abstract. In this paper, a new method for evaluating the integral property, truncated and impossible differentials for substitution-permutation network (SPN) block ciphers is proposed. The main assumption is an explicit description/expression of the internal state words in terms of the plaintext (ciphertext) words. By counting the number of times these words occur in the internal state expression, we can evaluate the resistance of a given block cipher to integral and impossible/truncated differential attacks more accurately than previous methods. More precisely, we explore the cryptographic consequences of uneven frequency of occurrences of plaintext (ciphertext) words appearing in the algebraic expression of the internal state words. This approach gives a new family of distinguishers employing different concepts such as the integral property, impossible/truncated differentials and the so-called zero-sum property.
We then provide algorithms to determine the maximum number of rounds of such new types of distinguishers for SPN block ciphers. The potential and efficiency of this relatively simple method is confirmed through applications. For instance, in the case of SKINNY block cipher, several 10-round integral distinguishers, all of the 11-round impossible differentials, and a 7-round truncated differential could be determined. For the last case, using a single pair of plaintexts differing in three words so that $(a = b = c) \neq (a' = b' = c')$, we are able to distinguish 7-round SKINNY from random permutations. More importantly, exploiting our distinguishers, we give the first practical attack on 11-round SKINNY-128-128 in the single-key setting (a theoretical attack reaches 16 rounds). Finally, using the same ideas, we provide a concise explanation on the existing distinguishers for round-reduced AES.

Keywords: SKINNY competition · Integral cryptanalysis · Impossible differential analysis · Truncated differential attack · Zero-Sum distinguisher

1 Introduction

Along with the development of internet of things, some new symmetric-key cryptographic schemes such as encryption algorithms, hash functions, authentication schemes and pseudo-random number generators have been proposed. The security evaluation of these schemes, against some well understood cryptanalytic techniques such as differential and linear cryptanalysis, impossible and truncated differentials, is an important task. The ability

Received: 2019-01-06, Revised: 2019-01-09, Accepted: 2019-01-11, Published: 2020-31-01

of newly proposed designs to resist these cryptanalytic methods should go through an accurate and in-depth security evaluation.

Integral cryptanalysis was originally proposed by Lars Knudsen [KW02] as a dedicated attack against the Square block cipher, and is commonly known as the Square attack. The main idea of this method is to prepare a set of plaintexts having the property that a certain portion of plaintext can attain all the possible values in the set whereas the remaining part of any plaintext is fixed to a constant value. The properties of the multiset of internal state values after encrypting several rounds are then analyzed. For example, a particular attack might use 256 different chosen plaintexts such that a single byte is variable and the remaining bytes are kept fixed. Such a set necessarily has a property that its elements add up (bitwise modulo two addition) to the all-zero vector. On the other hand, XORing the corresponding set of ciphertexts might provide a useful information about the cipher in question.

Impossible differential cryptanalysis was proposed independently by Biham, Biryukov Shamir [BBS99], and Knudsen [Knu98]. It exploits a (truncated) differential characteristic of probability exactly 0 and thus acts as a distinguisher. Then, such a distinguisher can be turned into a key-recovery attack by prepending and/or appending additional rounds, which are usually referred to as the analysis rounds. The keys involved in the analysis rounds which lead to the impossible differential are wrong keys and thus can be excluded from the key space.

In cryptography, higher-order differential cryptanalysis [Lai94] is a generalization of differential cryptanalysis, an attack used against block ciphers. While in standard differential cryptanalysis the difference between only two texts is used, higher-order differential cryptanalysis studies the propagation of a set of differences between a larger set of texts. The zero-sum property (structure) is a generalization of higher-order differentials. Namely, for a given function F, a zero-sum is a set of inputs which sum to zero, and whose images by F also sum to zero [KR07, BC10].

The related work. Commonly, when cryptanalysis of block ciphers is considered, the first step is to construct distinguishers of certain kind that cover as many rounds as possible. In this context, the most relevant security parameter is an exact estimate on the number of rounds for which different kind of distinguishers can be specified. Nevertheless, these estimates are usually cipher specific and the concept of provable security (thus being unable to find distinguishers covering more than some fixed number of rounds) for a given cipher is hard to establish. A certain progress in this direction has been made by Sun *et al.* [SLG+16] who derived some theoretical upper bounds on the number of rounds for which impossible differentials (alternatively zero-correlation linear hulls as a dual structure) may possibly exist. In particular, it was shown that there are no impossible differentials of AES covering more than $r = 5$ rounds when the properties of substitution boxes are not taken into account. This has been done by associating the so-called primitive index with the linear layers of SPN structures and showing that the length of impossible differentials of an SPN structure is upper bounded by the primitive index of the linear layers. This, however, does not exclude the existence of distinguishers that cover a larger number of rounds when the cipher specific S-boxes are taken into account.

Due to the fact that applications of our proposed methods (for the purpose of an efficient specification of distinguishers) mostly relate the SKINNY cipher, we briefly recall the main cryptanalytic advances of it. SKINNY [BJK+16] is a recently designed lightweight tweakable block cipher. In [LGS17], Liu, Ghosh, and Song investigated truncated related-tweakey differential trails of SKINNY and searched for the longest impossible and rectangle distinguishers, under the assumption of having only one active word in the input and the output. Then, Ankele *et al.* outlined a related-tweakey impossible differential attack on 21 rounds of SKINNY-64/128 and two attacks on 22 and 23 rounds of SKINNY-64/128

under the assumption that 48 bits of the tweakey are public[ABC+17]. In a related-key attack model, it is possible to cancel data differences with corresponding key differences over many rounds of SKINNY. As a consequence, in this model one can deduce differential characteristics with higher probability and therefore more rounds can be covered compared to the single key attack. However, due to a very strong assumption that the attacker can ask the encryption box to modify the unknown key in a predictable manner, this scenario is less realistic than the single-key attack model.

In order to motivate extensive public scrutiny of the cipher, the SKINNY designers launched several one-year competitions, for the details of these competitions see [SKI18]. In brief, the designers assigned the following goal: recover the secret key from a given encryption of a known book with 2^{20} randomly selected plaintext and ciphertext pairs. The suggested SKINNY instances are 4- to 20-round reduced variants of SKINNY-64-128 and SKINNY-128-128. So far, the maximum number of attacked rounds for SKINNY-64-128 is 12 [DL18], whereas for SKINNY-128-128 the number of rounds is 10 [Udo18].

Our Contributions. The main research motivation of this paper comes from the work of Sun et al [SLG+16] and the 2018-2019 SKINNY Cryptanalysis Competition. We introduce a rather simple but novel approach for checking the resistance of a cipher against distinguishers of different kind such as impossible and truncated differentials and certain integral ones. The core idea of our approach is a useful representation of the round function as a multi-variate polynomial of F_{2^b}, where b denotes the size of S-boxes. Then, counting the number of occurrences of the variable corresponding to a specific input word in each output word will provide us with a precious information related to plausible distinguishers. More precisely:

- If the considered variable does not appear in the polynomial corresponding to an output word of the round function, then it means that this output word does not depend on it. This information is useful when building probability 1 truncated trails and impossible differentials.
- The first round in which this particular variable appears in all output words is the maximum number of rounds propagating in the forward direction for an integral distinguisher.
- This framework can also be used to identify order 2 differentials that can be cancelled out in some words.

This paper concentrates on the provable security of block ciphers against integral and impossible differential cryptanalysis as well as cryptanalysis based on zero-sum distinguishers. Our main approach, based on counting the occurrences in a multi-variate model, gives a simple and intuitive description of SPN-based block ciphers when the concept of provable security against the most important cryptanalytic techniques is considered. The main achievement of this technique is the possibility of determining the properties of round functions by measuring the statistics of the above mentioned occurrences. (integral, impossible differential and truncated differential cryptanalysis) on the algebraic structure of the cipher. The algebraic structure of a given cipher, considered as a multi-variate polynomial describing the internal state, is used to evaluate its security against the above attacks.

Finally, notice that our method conceptually improves (and extends) upon the work of Sun et al. in [SLG+16] because the counting process in our case takes the number of occurrences into account and not only whether a certain word appears at the output or not. Our main results and the previous works are listed in Table 1.

Organization. The rest of the paper is organized as follows. In Section 2, we introduce some basic notation and give a brief description of AES and SKINNY block ciphers. In Section 2.4 we introduce the algebraic representation of SPN block ciphers and give a

Table 1: Summary of the Attacks on SKINNY

Target Algorithm	Attack Model	Rounds	Time Complexity	Source
SKINNY-64-128	Single-Key	12	$2^{51.95}$	[DL18]
SKINNY-128-128	Single-Key	10	2^{48}	[Udo18]
SKINNY-64-64	Related-Tweakey I D	19	$2^{63.03}$	[LGS17]
SKINNY-128-128	Related-Tweakey I D	19	$2^{124.60}$	[LGS17]
SKINNY-64-128	Related-Tweakey I D	23	$2^{125.91}$	[LGS17]
SKINNY-128-256	Related-Tweakey I D	23	$2^{251.47}$	[LGS17]
SKINNY-64-128	Related-Key I D	21	2^{86}	[ABC$^+$17]
SKINNY-64-128	Related-Key I D	†22,23	$2^{71.6}, 2^{79}$	[ABC$^+$17]
SKINNY-128-128	Single-Key	16	2^{112}	Ours

†: Attacks under the assumption that 48 bits of the tweakey are public. I D = Impossible Differential

motivation for its use. An algorithm for finding integral distinguishers is given in Section 3, along with its application to SKINNY and AES. Impossible differentials and algorithm for determining the zero-sum distinguishers are treated in Section 4. The linear combinations of the words of intermediate state on the input words is studied in Section 5, we analyze the dependence between linear combination of the output words of intermediate stage and the input words. A cryptanalysis of the reduced-round SKINNY-128-128 cipher, in a single-key model, is elaborated in Section 6. Some concluding remarks are given in Section 7.

2 Preliminaries

The substitution permutation network (SPN) structure is widely used in constructing cryptographic primitives. It iterates some SP-type round functions to achieve Shannon's concepts of confusion and diffusion. More specifically, the SP-type function $f : F_2^{b\times n} \to F_2^{b\times n}$ used in this paper is defined as follows: assume the input x is divided into n words $x = (x_0, \cdots, x_{n-1})$, where each x_i is of length b in bits (considered as a word here). Applying the nonlinear transformation S_i to x_i, we denote $y = (S_0(x_0), \cdots, S_{n-1}(x_{n-1})) \in F_2^{b\times n}$. Finally, one applies a linear transformation P to y, so that $P(y)$ is considered as the output of f. Both SKINNY and the Advanced Encryption Standard (AES) block ciphers use the design principles of iterating SP-type function several times (each iteration is called a round).

We denote by m_{ij} the (i, j) entry of the matrix M. For example m_{00} is the entry corresponding to the first row and first column, thus for convenience the indices start from 0. Other notation used throughout this article is given below.

2.1 Notations

The following notations are used throughout the rest of the paper:
-X_i: The output of the round i.
-$X_i[j]$: The j-th word of X_i, where $0 \le j < 16$.
-p_i: The i-th word of plaintext, where $0 \le i < 16$, used in the encryption.
-$P \setminus \{p_i\}$: All the bytes of plaintext, except the i-th, where $0 \le i < 16$.
-c_i: The i-th word of ciphertext, where $0 \le i < 16$, used in the decryption.
-k_j: The j-th word of the master key.
-K^i: The key of the i-th round.
-RC_i: The round constant for the i-th round.
-$A = [j_0, j_1, \cdots, j_{s-1}]$: an ordered set of integers such that $j_0 < j_1 < \cdots < j_{s-1}$.

-S: S-box.
-V: The inverse of S-box.
-b: The number of input bits of S-box.

Consider a set of plaintexts, i.e., a set of 2^b plaintexts which are equal in 15 words except for the one in the position j for a certain $0 \leq j < 16$. We call this set a δ-set.

A word is called balanced if XOR-ing its 2^b values, obtained by encrypting the δ-set, is zero.

2.2 The AES Block Cipher

The Advanced Encryption Standard(AES) is an iterated block cipher which encrypts 128-bit plaintext into 128-bit ciphertext. AES only uses sixteen identical S-boxes in each round, each S-box corresponding to the inverse function over F_{2^8}. The round function consists of four basic transformations given below (treating the internal state as 4×4 (state) matrix whose entries are bytes):

- SB is a nonlinear substitution that applies the same S-box to each byte of the internal state.
- SR is a cyclic rotation of the i-th row of the state matrix by i bytes to the left, for $i = 0, 1, 2, 3$.
- MC is a multiplication of each column with a Maximum Distance Separable (MDS) matrix over F_{2^8}.
- AK denotes XOR-ing the state matrix with the round key.

The use of an MDS matrix essentially guarantees that the sum of active bytes in the input and output of the MixColumns operation is at least 5, unless all bytes are non-active. The MDS matrices used in MC operation are shown below, both encryption and decryption direction are given. Note that $X[j]$ is the input value and $Y[j]$ is the updated value. The numbers $e, b, d, 9$ are the hexadecimal representations of finite field elements.

$$\begin{pmatrix} Y[0] \\ Y[1] \\ Y[2] \\ Y[3] \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix}, \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} e & b & d & 9 \\ 9 & e & b & d \\ d & 9 & e & b \\ b & d & 9 & e \end{pmatrix} \begin{pmatrix} Y[0] \\ Y[1] \\ Y[2] \\ Y[3] \end{pmatrix}$$

Since we do not investigate the key-recovery attacks, please refer to [DR02] for the details related to the key schedule.

2.3 The SKINNY Block Cipher

SKINNY [BJK+16] is a family of lightweight block ciphers proposed at Crypto 2016. It adopts the SPN structure just like AES. Several papers have been published that analyse SKINNY after it published. Some of the attacks such as [LGS17, ABC+17, SMB18] are in the related tweakey model, whereas in the single-tweakey model commonly impossible differentials are employed [TAY17].

SKINNY has a variable block size of 64, 128 bits where both the plaintext and the intermediate state are described by matrices of size 4×4 containing words. Each round of SKINNY is composed of four operations applied to the internal state in the order specified below:

1. SubByte: Apply the S-box of SKINNY to each nibble.

2. AddConstants and AddRoundKey(AK): XOR the state with constant and subkey.

3. ShiftRow: Shift the i-th row by i nibbles to the right, $i = 0, 1, 2, 3$.

4. MixColumn: Multiply each column by a constant 4×4 matrix M_{Skinny} over the field F_2^4 and F_2^8, respectively (depending on the block length being 64 or 128), where

$$M_{\text{Skinny}} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix}, M_{\text{Skinny}}^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

In all the cases, the words are numbered row-wise, one round of SKINNY is shown in Figure 1.

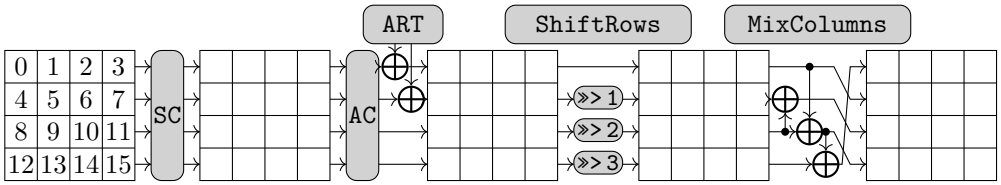

Figure 1: SKINNY Round Function.

Since XORing with constants does not influence the integral property, impossible differential and zero-sum property, we do not consider AddConstants (AC) and AddRoundKey (AK) in our analysis. For more details about SKINNY the reader can refer to [BJK+16].

2.4 The Algebraic Representation of SPN Ciphers

In this section, we will present an explicit formula for each word of the ciphertext being represented using the plaintext. We borrow some ideas from [SLG+16], thus representing the dependence of the output on the input by a binary matrix. If the input X_i to the i-th round function is viewed as a column vector $(X_i[0], \cdots, X_i[n-1])^T$, then the output X_{i+1} can also be viewed as a column vector $(X_{i+1}[0], \cdots, X_{i+1}[n-1])^T$. The latter can be computed as $X_{i+1} = MS(X_i) = M(S_0(X_i[0]), \cdots, S_{n-1}(X_i[n-1]))^T$, where M denotes the matrix of the linear transformation P.

Considering SKINNY as an example, the output words of the first round (being inputs

of the second round) can be represented using the plaintext words as follows:

$$\begin{cases} X_1[0] &= S(p_0) + k_0 + S(p_{10}) + S(p_{13}) \\ X_1[1] &= S(p_1) + k_1 + S(p_{11}) + S(p_{14}) \\ X_1[2] &= S(p_2) + k_2 + S(p_8) + 2 + S(p_{15}) \\ X_1[3] &= S(p_3) + k_3 + S(p_9) + S(p_{12}) \\ X_1[4] &= S(p_0) + k_0 \\ X_1[5] &= S(p_1) + k_1 \\ X_1[6] &= S(p_2) + k_2 \\ X_1[7] &= S(p_3) + k_3 \\ X_1[8] &= S(p_7) + k_7 + S(p_{10}) \\ X_1[9] &= S(p_4) + k_4 + S(p_{11}) \\ X_1[10] &= S(p_5) + k_5 + S(p_8) + 2 \\ X_1[11] &= S(p_6) + k_6 + S(p_9) \\ X_1[12] &= S(p_0) + k_0 + S(p_{10}) \\ X_1[13] &= S(p_1) + k_1 + S(p_{11}) \\ X_1[14] &= S(p_2) + k_2 + S(p_8) + 2 \\ X_1[15] &= S(p_3) + k_3 + S(p_9) \end{cases} \tag{1}$$

Let M_{SK} be the matrix representation of the composition of the ShiftRow and MixColumn operations. Its precise description is given in Appendix A. Then,

$$(X_1[0], X_1[1], \cdots, X_1[n-1]) = M_{SK}(S(p_0), S(p_1), \cdots, S(p_{n-1})) + K^0 + RC_0.$$

$$(X_2[0], X_2[1], \cdots, X_2[n-1]) = M_{SK}(S(X_1[0]), S(X_1[1]), \cdots, S(X_1[n-1])) + K^1 + RC_1.$$

Since one can represent the words of the subsequent round (say $i+1$) in terms of the previous round i recursively, then it is possible to deduce an algebraic representations of SKINNY reduced to r rounds, so that output internal state is expressed in terms of plaintexts words and round subkeys. For example, the detailed expression of $X_7[8]$ (thus SKINNY reduced to 7 rounds) in terms of the plaintext words can be found in Appendix B. Where "2" is the round constant, "+2" means bitwise modulo addition with the constant 0x02. Specifying these algebraic expressions allows us to study and explore many properties of a given cipher which certainly offers some benefits in the cryptanalysis.

In [SLG+16], the authors proposed the use of the so-called **characteristic matrix** of the linear layer of a given block cipher. For $M = (m_{ij}) \in F_{2^b}^{n \times n}$, denote by Z the integer ring. The characteristic matrix of the round function is then defined as $M^* = (m_{ij}*) \in Z^{n \times n}$, where $m^*_{ij} = 0$ if $m_{ij} = 0$ and $m^*_{ij} = 1$ otherwise. A matrix $M \in Z^{n \times n}$ is non-negative if all elements of M are non-negative, and positive if all elements of M are positive. Therefore, the characteristic matrix is always non-negative.

Since SKINNY utilizes a binary matrix, the characteristic matrix M^* for SKINNY round function is equal to its linear transformation matrix M. According to the definition of characteristic matrix, $m_{i,j} = 0$ means that the i-th word of the output in the first round is independent of the j-th word of input. Generally, the characteristic matrix of r-round encryption is $(M^*)^r = (h_{ij})$, where h_{ij} denotes the frequency that p_i appears in the j-th word of the output after r rounds of encryption. Then, $h_{ij} = 0$ means that the i-th output byte of the r-round SPN cipher is independent of the j-th input byte, whereas $h_{ij} = 1$ indicates that p_i appears once in $X_r[j]$. Notice that when S is a permutation and other words of the plaintext are fixed, then $X_r[j]$ is a permutation on p_i.

Nevertheless, we want to remark that the matrix representation in [SLG+16] is not always compatible with our method. This is mainly based on the fact that the matrix-based approach only extracts and deals with a binary information, thus measuring whether words appears or not. As a consequence, more sophisticated distinguishers taking into account the exact (multiple) number of occurrences cannot be deduced from the matrix-based

approach but can be derived using Equation (1). For example, in the expression of $X_7[8]$ in Appendix B, a closer inspection reveals that p_{12} and p_{13} occurs respectively 3 and 6 times. Later in Section 3.3, we will show that utilizing the expressions in Equation (1) one can identify that $X_7[8]$ is balanced when p_{12}, p_{13} traverse over $F_{2^b} \times F_{2^b}, b = 4$ or 8. On the other hand, the matrix-based approach will wrongly identify it as an unbalanced one!

3 Algorithm Approach for Integral Distinguishers

In this section we propose an algorithmic approach for building integral distinguisher based on some simple observations. In particular, we derive an upper bound on the number of rounds which can be covered by this method.

3.1 The Integral Distinguisher

Let r_{enc} be the minimum integer identifying that after r_{enc} rounds encryption, the algebraic expression of any output word includes (depends on) the information of every word of the plaintext (full diffusion). On the other hand, at the end of $r_{enc} - 1$ or even less encryption round, there is at least one output word which does not depend on all the plaintext words. Similarly, let r_{dec} be the minimum integer so that decrypting r_{dec} rounds the algebraic representation of each word of the output becomes dependent on all the ciphertext words.

Claim. *The integral distinguisher can cover at most $r_{enc} + r_{dec}$ rounds.*

Let state_0, state_1, $\cdots$, state_4 denote the outputs of several rounds of the encryption/decryption process. More precisely, state_2 is decrypted to state_0, and encrypted to state_4. Figure 2 visualizes the diffusion of an active byte in forward (encryption) and backward (decryption) computation.

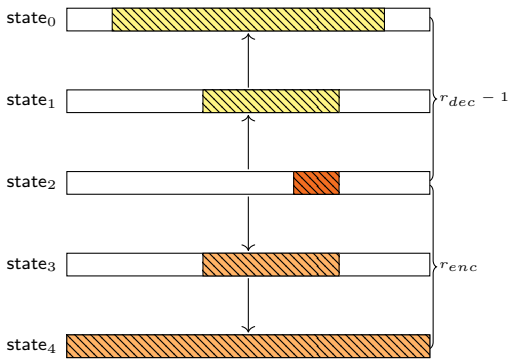

Figure 2: Building an integral distinguisher

Suppose now that q_j is an active word in

$$\mathsf{state}_2 = (q_0, q_1, \cdots, q_{n-1}),$$

which is highlighted in red in Figure 2. On one hand, the word q_j diffuses gradually in the encryption process, after r^j_{enc} rounds encryption, it appears in the expression of each word of the output. Suppose it appears once in the expression of the words of the state_4 whose coordinates are in $A = \{j_0, j_1, \cdots, j_{s-1}\}$.

On the other hand, suppose we decrypt state_2. Denote the state after $r^j_{dec} - 1$ rounds of decryption by state_0, which implies the existence of at least one word of state_0 independent of q_j. Notice that q_j is contained in all output words after r^j_{dec} rounds decryption. Suppose

that the i_0-th, i_1-th, $\cdots$, $i_{t-1}-$th, $t < n$ words of state_0 depend on q_j. When the plaintext words $p_{i_0}, p_{i_1}, \cdots, p_{i_{t-1}}$ goes through $(F_{2^b})^t$, then after $r^j_{enc} + r^j_{dec} - 1$ encryption rounds the words of the ciphertext with coordinates $\{j_0, j_1, \cdots, j_{s-1}\}$ sum to zero. Such a distinguisher has the data complexity of 2^{bt}.

Now we investigate whether the forward r^j_{enc}-round distinguisher can be extended one more round by propagating the active word in the forward direction and taking into account the linear transformation. Now we want to determine whether there is $X_{r^j_{enc}+1}[i] = \sum_{k=0}^{n-1} m_{ik} S_k(X_{r^j_{enc}}[k])$ such that q_j occurs once in each $X_{r^j_{enc}}[k]$, for any $m_{ik} \neq 0$. Where m_{ik} is the entry in the i-th row and the k-th column of the linear transformation matrix. If so, every $X_{r^j_{enc}+1}[k], m_{ik} \neq 0$ is balanced when q_j goes through F_{2^b} and the other words of X_i are fixed to a constant value. This is because each $S_k(X_{r_{enc}}[k])$ is a permutation on q_j.

For each $0 \leq j < n$, we get the corresponding number

$$r^j_{enc} + 1 + r^j_{dec} - 1 = r^j_{enc} + r^j_{dec}, \tag{2}$$

and the longest integral distinguisher may cover $max\{r^0_{enc} + r^0_{dec}, r^1_{enc} + r^1_{dec}, \cdots, r^{n-1}_{enc} + r^{n-1}_{dec}\}$ rounds.

Note that $r^j_{enc} \leq r_{enc}, r^j_{dec} \leq r_{dec}$, which proves our claim. We summarize the above discussion as follows.

Theorem 1. *For an SPN cipher, let r_{enc}, r_{dec} be the minimum number rounds needed to achieve full diffusion for encryption and decryption, respectively. Then the integral distinguisher can cover up to $r_{enc} + r_{dec}$.*

We now further analyze whether two input words $p_i \neq p_k$ $(i \neq k)$ occur once in two output words $X_{r_{enc}}[s], X_{r_{enc}}[t], s \neq t$, which consequently leads to an integral property at the end of round $r_{enc} + 1$.

Definition 1. A function $\pi(x, y) = (\varphi_1(x, y), \cdots, \varphi_n(x, y))$ from $(F_{2^b})^2$ to F_{2^b} is said to have *order-2 squared property* on a set $\Omega_1 \times \Omega_2$ if $\sum_{x \in \Omega_1, y \in \Omega_2} \pi(x, y) = 0$.

If there is an $X_{r_{enc}+1}[j]$ formed as

$$X_{r_{enc}+1}[j] = S_1((f(P \setminus \{p_i\}) + p_i) + k_{r_{enc}+1,j} + S_2((g(P \setminus \{p_k\}) + p_k),$$

where p_i appears once in the first S-box and p_k appears once in the second S-box, then the active words p_i and p_k lead to an integral property at the end of round $r_{enc} + 1$. In fact, when X_i takes all the possible values in the i-th and k-th words while the remaining $n - 2$ words are kept fixed, then the j-th word of the output after $(r_{enc} + 1)$ rounds encryption sums to zero. Generally, we have the following theorem.

Theorem 2. *Let $S_1(x)$ and $S_2(x)$ be permutations on F_{2^b}. Then $S_1(x+f(y))+S_2(y+g(x))$, for $x, y \in F_{2^b}$, has order-2 squared property on $(F_{2^b})^2$.*

Proof. Since $S_1(x)$ is a permutation on F_{2^b}, for every fixed $y \in F_{2^b}$ the sum of $\sum_{x \in F_{2^b}} S_1(x + f(y))$ is always zero. The same is true for $\sum_{y \in F_{2^b}} S_2(y + g(x))$. So

$$\begin{aligned} & \sum_{(x,y) \in F_{2^b} \times F_{2^b}} (S_1(x + f(y)) + S_2(y + g(x))) \\ = & \sum_{y=0}^{2^b-1} \sum_{x=0}^{2^b-1} S_1(x + f(y)) + \sum_{x=0}^{2^b-1} \sum_{y=0}^{2^b-1} S_2(y + g(x)) = 0 + 0, \end{aligned}$$

which proves the theorem. □

3.2 A Generic Algorithm for Integral Distinguishers

In this subsection, we propose a generic algorithm for finding the longest integral distinguisher, cf. Algorithm 1. Step 2 to Step 3 of this algorithm aim to find words of X_0 appearing only once in the output words for the largest number of encryption rounds possible, which are then saved in a pool Ω. Step 4 and 5 search for the words in Ω not occurring in the output for as many decryption rounds as possible. The steps 8 - 10 check whether the distinguisher can be extended by one more round in the encryption direction.

Algorithm 1 Finding an integral distinguisher of maximum length

Input: The representation of one round of encryption $X_1[j] = \sum_{k=0}^{n-1} a_{jk} S_k(p_k)$ and one round decryption $Y_1[j] = \sum_{k=0}^{n-1} b_{jk} S_k^{-1}(c_k)$
Output: A set of integral distinguishers Γ

1: Set Γ to be an empty set
2: Encrypt X_0 by $X_{i+1}[j] = \sum_{k=0}^{n-1} a_{jk} S_k(X_i[k])$ iteratively for $i = 0, 1, \ldots$ till every word of X_0 is contained in the expression of all the words of the internal state, and there exists some $X_0[i]$ which appears exactly once in the expression of some word of the internal state.
3: Denote the number of iterated times in the step above by r^A_{enc}, and put those words which occur only once after r^A_{enc} rounds of encryption in a pool $\Omega = \{X_0[i_0], \cdots, X_0[i_t]\}$.
4: Decrypt ciphertext Y_0 and calculate the algebraic representation of the internal states by $Y_{i+1}[j] = \sum_{k=0}^{n-1} b_{jk} S_k^{-1}(Y_i[k])$, iteratively for $i = 0, 1, \ldots$ till every word of Y_0 appears in the expression of all the word of the internal states, and there exists some $Y_0[i]$ which appears exactly once in the expression of some word of the internal state. We denote the number of iterated times here by r^A_{dec}.
5: Check whether there is some word Z among $\Omega = \{X_0[i_0], \cdots, X_0[i_t]\}$ which does not appear in some word of $Y_{r^A_{dec}-1}$ — the internal state after $r^A_{dec} - 1$ rounds of decryption. If such word cannot be found, let $r^A_{dec} = r^A_{dec} - 1$ and repeat this step. Once found, record all fulfilling words Z in a set Λ.
6: Denote those words of $X_{r^A_{enc}}$, in the expression of which some element Z from Λ occurs exactly once, as $X_{r^A_{enc}}[Z_0], \cdots, X_{r^A_{enc}}[Z_k]$.
7: Encrypt $X_{r^A_{enc}}$ one more round to get $X_{r^A_{enc}+1}$.
8: **while** There is a word of $X_{1+r^A_{enc}}$ whose one round representation only includes the words among $X_{r^A_{enc}}[Z_0], \cdots, X_{r^A_{enc}}[Z_k]$ **do**
    return $r^A_{enc} + r^A_{dec}$
9: **end while**
10: **return** $r^A_{enc} + r^A_{dec} - 1$

3.3 The Integral Property for SKINNY

In the case of SKINNY cipher, for the encryption direction, each of $p_{12}, p_{13}, p_{14}, p_{15}$ occurs once in the expression of some words among the last 12 words of the output of round six. Moreover, we have $X_7[8] = S(X_6[7]) + k_7 + S(X_6[10])$. In the algebraic representation of $X_6[7]$, p_{12} occurs once and p_{13} occurs five times, in algebraic representation of $X_6[10]$, p_{12} appears twice and p_{13} appears once, respectively. This observation ensures that $S(X_6[7])$ is a permutation on p_{12} when the other fifteen words are kept fixed. Also, $S(X_6[10])$ is a permutation on p_{13}, when the remaining fifteen words are fixed. Thus, we get a 7-round integral property for all versions of SKINNY which we state now formally.

Theorem 3. *For all versions of block cipher family SKINNY, when p_{12}, p_{13} goes through $F_{2^b} \times F_{2^b}, b = 4$ or 8, and the remaining words of the plaintext are fixed, the sum of $X_7[8]$ over these plaintexts is zero.*

The correctness of Theorem 3 can be confirmed from the detailed expression of $X_7[8]$ in Appendix B. Figure 3 depicts the 7-round SKINNY integral distinguisher.

The frequency of plaintext words occurring in $X_7[8]$ are listed in Table 2. The words occurring fewer times in the expression of an output word than the others may lead to integral property.

Table 2: The frequency of plaintext bytes occur in $X_7[8]$

p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7
12	10	8	6	4	7	7	5
p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}
13	11	13	11	3	6	4	4

The 7-round distinguisher can be easily explained in the following way. Suppose that the input space X is composed of vectors taking all possible 2^{2b}, ($b = 4$ or 8) values on the twelfth and thirteenth words while the other 14 words take constants. In this case $|X| = 2^8$ or 2^{16} for SKINNY-64 and SKINNY-128 correspondingly, and X is a subspace of $(F_2^b)^{16}$. After seven rounds of encryption, the XOR sum of the eighth output byte is zero with probability 1.

The integral property can be extended by 3 rounds in backward direction by activating 15 words. In the decryption direction, c_{12}, c_{13} do not occur in the algebraic representation of the 14-th word of the output after three rounds of decryption, but after four rounds of decryption at least one of them occur in the expression of each word of the output. Consequently, 10-round integral distinguishers can be constructed. That is, by fixing the 14-th word while letting the remaining part of the plaintext goes through $(F_{2^8})^{15}$, the 8-th word sums to zero after 10 encryption rounds.

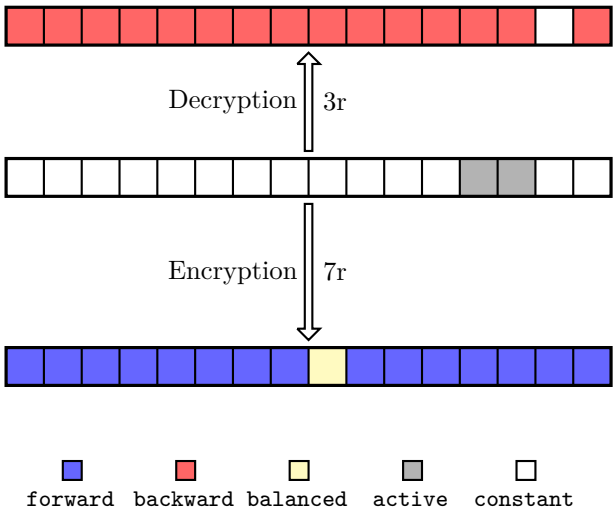

Figure 3: 10-Round integral distinguisher for SKINNY

Integral property comparison: our approach vs division property. One can also investigate the integral property by using the division property. In this context we want to emphasize that in [ZR19] the authors showed that for SKINNY-64, 10 rounds is the upper bound when searching for integral distinguishers based on division property. This analysis has been conducted taking into account the exact properties of both linear and nonlinear layer. In difference, we get a 10-round distinguisher by using simple algebraic

structure. Notice also that a major disadvantage of distinguishers based on the division property is that the time complexity for finding these (using CPLEX, SAT and STP solvers) grows super exponentially in the number of rounds.

3.4 The Integral Property of AES

Let us mark the states of each operation of AES round function by increasing subscripts. Then each round consists of four states. There is a basic integral distinguisher on the 3-round AES (see Figure 4 state ♯5 to state ♯16). Consider the expression of state ♯12 in terms of state ♯5 (bytewise). It can be verified that in the encryption of two AES rounds, each byte in state ♯5 occurs exactly once in the expression of every byte of state ♯12. That is each byte in state ♯12 depends on all the bytes of state ♯5.

Letting the first byte of state ♯5 to take on each of the 256 possible values exactly once while keeping the remaining 15 bytes fixed, then every byte of state ♯12 is a permutation on the first byte of state ♯5. The same property applies to the bytes in state ♯15. When the input of a permutation on F_{2^8} ranges through all possible values so does its output as well. Since any word in state ♯16 is a linear combination of bytes in state ♯15, then the XOR of its 256 values during the encryption of a δ set is zero. That is, when any byte of state ♯5 takes on all possible values (the other bytes being fixed) then each byte of the output of 3-round AES is balanced. Now we add one round on the top of the 3-round

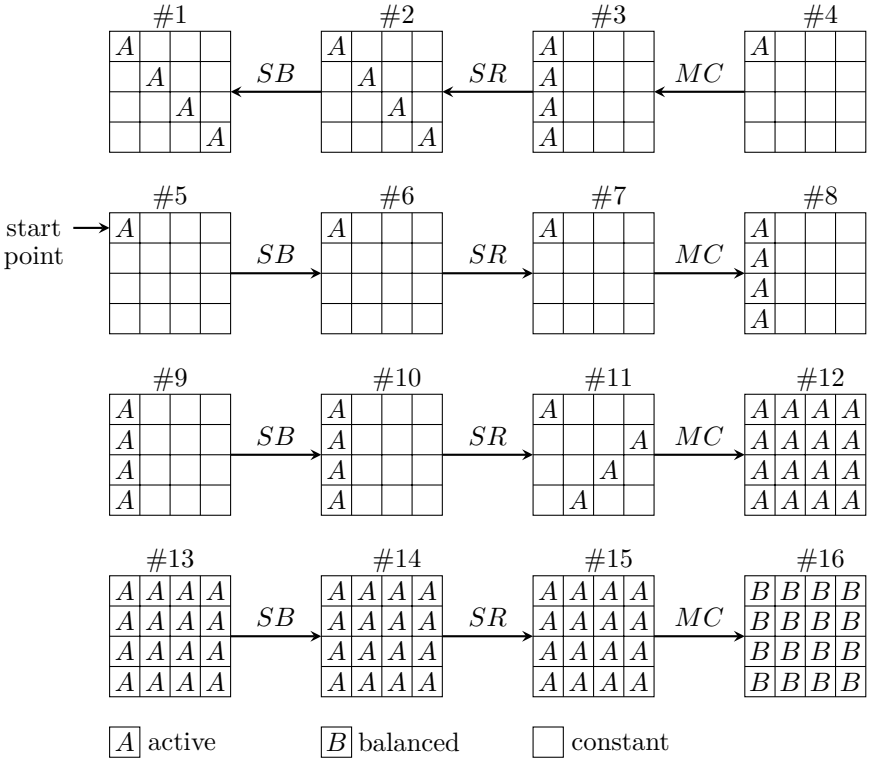

Figure 4: 4-Round integral distinguisher for AES

integral distinguisher. Denote by Δ a set corresponding to state ♯5 with the variable first and the remaining 15 bytes being fixed. When we decrypt the state ♯5 backward for one round, given Δ, it is possible to prove that the XOR-sum of state ♯1 over Δ is always equal to 0. Since in this decryption direction Δ only occurs once in the expression of each byte of state ♯1 in the diagonal, it implies that each byte in the diagonal of state ♯1 is

a permutation on Δ. On the other hand, when the four bytes in the diagonal of state $\sharp 5$ goes through $(F_{2^8})^4$, there are 2^{24} structures with the $(0,0)$ entry varying through all possibilities and the other 15 are held constants.

For any constant $Y[0], c_1, c_2, c_3 \in F_{2^8}$, the equation

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{pmatrix} = \begin{pmatrix} Y[0] \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

has exactly one solution, because M is invertible. Then, letting Y_0 goes through F_{2^8}, $M(X_0, X_1, X_2, X_3)^T = (*, c_1, c_2, c_3)^T$ has 256 solutions which describe the structure of state $\sharp 4$. When (c_1, c_2, c_3) goes through $(F_{2^8})^3$, we get 2^{24} structures of state $\sharp 4$. For each of these structures, state $\sharp 16$ sums to zero. Thus, we have a four-round integral distinguisher using four active bytes in the diagonal with the remaining bytes fixed indicating that every byte of the output of round 4 sums to zero. Figure 4 depicts the 4-round AES integral distinguisher.

Furthermore, let the 4 bytes in the diagonal of the state $\sharp 1$ takes all possible values from $(F_{2^8})^4$ and the other 12 bytes be constants, then the output of 1-round AES can be divided into 2^{24} structures of Δ. Therefore, the sum of each byte of the output of the fourth round is 0.

We can conclude that AES has no integral property of more than 4 rounds unless the details of the S-boxes are taken into account, because we can not extend the distinguisher in Figure 4 neither in the forward nor in the backward direction.

4 Impossible and Zero-Sum Differentials

In this section, we analyze impossible and second-order differentials within the framework of our algebraic representation. We also give an application of our methods to block ciphers SKINNY and AES.

To describe the search for impossible differentials we depict this process in Figure 5. The analysis starts from the encryption direction. Let $X = \mathsf{state}_0$, $X[i]$ be the i-th word of X. We first calculate the algebraic representation of the encryption operation. For each $X[i]$, we record the round number r^i_{enc} that $X[i]$ occurs in all words of $X_{r^i_{enc}}$ but not occurring in at least one word of $X_{r^i_{enc}-1}$. Let us denote $\mathsf{state}_2 = X_{r^i_{enc}}$.

Similarly, the algebraic representation of decryption operation is calculated. Let $Y = \mathsf{state}_4, Y[j]$ be the j-th word of Y.

From the encryption point of view, the active byte $X[i]$ is fully diffused to all the words of state_2, but from the decryption perspective, there is at least one word in state_2 which is independent of the active byte $Y[j]$. Therefore, having the input differential at the i-th word and the output difference at the j-th word can not hold simultaneously. Consequently, we get an impossible differential distinguisher covering $r^i_{enc} + r^j_{dec} - 1$ rounds. This is commonly named as the miss in the middle technique.

For any $0 \le i, j < n-1$, we can get the corresponding $r^i_{enc} + r^j_{dec} - 1$ so that impossible differential distinguisher covering most rounds is computed as $\max\{r^i_{enc} + r^j_{dec} - 1, 0 \le i, j < n-1\}$.

4.1 Applications to AES and SKINNY

For AES, in the encryption direction, one active byte achieves full diffusion after 2 rounds. On the other hand, considering the decryption process, one active byte propagates to four bytes after one round AES decryption and it propagates to all 16 bytes after 2 decryption rounds. Thus, there is a three-round impossible differential with two encryption and one

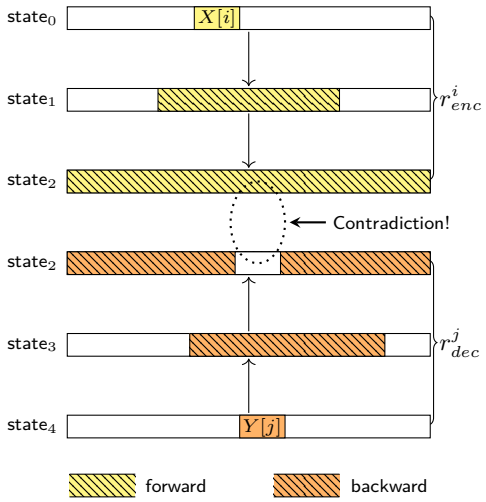

Figure 5: Building impossible differential distinguisher

decryption round. Similar to many other attacks on AES, if we require that the last round is without MixColumns, the impossible differential distinguisher can be extended to 4-round AES when the fourth round has an unique active byte. The impossible differential distinguisher is depicted in Figure 6.

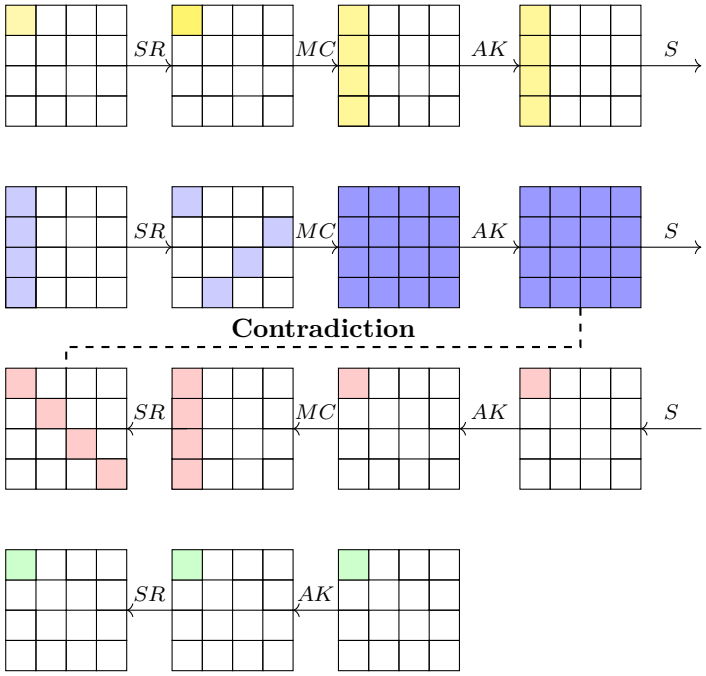

Figure 6: 4-Round Impossible Distinguisher for AES

For SKINNY, we found that only $Y[8], Y[9], Y[10], Y[11]$ do not appear in the expression of one word among the last 8 bytes of the intermediate state after five decryption rounds. On the other hand, $X[12], X[13], X[14], X[15]$ appear once in the expression of the last 8 words after six encryption rounds. Hence, we can pick up any such pair (in total having 16

possibility of combining $X[i]$ and $Y[j]$) contradicting in any of the last eight words of the intermediary state to construct impossible differential characteristic reaching 11 rounds in total.

4.2 Algorithm for Determining the Zero-Sum Differential

In the known-plaintext scenario, differential attacks are not applicable in general. Indeed, this technique depends on particular differential trails and observing such a difference in a pool of random plaintexts is very improbable. For example, in order that a specific n-bit difference is observed with high probability, it is required that the data pool has size close to $2^{\frac{n}{2}}$. Even more data is needed for higher-order differential cryptanalysis, and the same is true when the differential is probabilistic. However, a zero-sum differential has much higher chance to be observed. This probability is even higher if the random plaintext blocks have low entropy, for example, if the plaintext is a text taken from a book [Udo18]. So it is interesting to study the mathematical principles of zero-sum distinguisher via the algebraic expression of the internal states.

Similarly to what has been done for the integral property, we shall calculate the expression of the internal states as functions of the plaintext words $(p_0, p_1, \cdots, p_{15})$ in the encryption direction for r_{enc}^- rounds. We are in particular interested in those $p_i, p_j, i \neq j$, such that p_i does not occur in the expression of $X_{r_{enc}^-}[s]$ whereas p_j does not occur in the expression of $X_{r_{enc}^-}[t]$, where $0 \leq s \neq t < n-1$. But on the other hand, all the words occur in the expression of any word of $X_{r_{enc}^-+1}$.

Assume that the expression of one word of round $r_{enc}^- + 1$ is of the form

$$X_{r_{enc}^-+1}[u] = F(P) = F_1(P \setminus \{p_i\}) + F_2(P \setminus \{p_j\}), i \neq j,$$

where F is a function from $(F_{2^b})^n$ to F_{2^b} and F_1, F_2 are functions from $(F_{2^b})^{n-1}$ to F_{2^b}. Then the cipher has an $r_{enc}^- + 1$-round zero-sum distinguisher.

Assume that p_i, p_j can take on four possible values $(a,b),(a,c),(d,b),(d,c)$ whereas the remaining $n-2$ words are constants. Such a set of plaintexts is denoted by $P \setminus \{p_i, p_j\}$ while specifying $p_i = a, p_j = b$ is denoted by $(P \setminus \{p_i, p_j\}a, b)$ (similarly for other prespecified p_i and p_j). Defining

$$\begin{aligned} F_1(p_j = x) &\triangleq F_1(p_0, \cdots, p_{i-1}, p_i, p_{i+1}, \cdots, p_{j-1}, x, p_{j+1}, \cdots, p_{n-1}) \\ F_2(p_i = y) &\triangleq F_2(p_0, \cdots, p_{i-1}, y, p_{i+1}, p_{j-1}, p_j, p_{j+1}, \cdots, p_{n-1}), \end{aligned}$$

one can compute

$$\begin{aligned} & F(P \setminus \{p_i, p_j\}a, b) + F(P \setminus \{p_i, p_j\}a, c) + F(P \setminus \{p_i, p_j\}d, b) + F(P \setminus \{p_i, p_j\}d, c) \\ = \; & F_1(p_j = b) + F_2(p_i = a) + F_1(p_j = c) + F_2(p_i = a) \\ + \; & F_1(p_j = b) + F_2(p_i = d) + F_1(p_j = c) + F_2(p_i = d) = 0. \end{aligned}$$

That is to say, the cipher has the zero-sum property extending to $r_{enc}^- + 1$ rounds. The cipher can be distinguished by a quadruple of plaintexts with two words of them formed as $(a,b),(a,c),(d,b),(d,c)$ and the other words being constant.

To achieve the $r_{enc}^- + 1$ round zero-sum, the attacker only needs quadruples of plaintexts that differ in at most two (active) bytes and that sum to zero. These quadruples are then encrypted using random keys which after $r_{enc}^- + 1$ rounds ensures that the word in some predefined positions sums to zero over the four encryptions.

For SKINNY, after five encryption rounds, p_{14} does not occur in $X_5[7]$ and p_{13} does not occur in $X_5[10]$, and furthermore we have $X_6[8] = S(X_5[7]) + S(X_5[10])$. So

$$\sum_{P \setminus \{p_{13}, p_{14}\} = constant, (p_{13}, p_{14}) \in \{(a,b),(a,c),(d,b),(d,c)\}} X_6[8] = 0,$$

giving the zero-sum property of the 8-th word after 6 encryption rounds.

Remark 1. In [Udo18], the zero-sum property of SKINNY has been deduced through experiments. More precisely, by encrypting some quadruples with random keys the authors of [Udo18] observed that after exactly 6 encryption rounds, the 8-th word has the zero-sum property. Furthermore, by utilizing the 6-round truncated differential a key-recovery attack on 10 rounds SKINNY-128-128 can be mounted running in time complexity 2^{48} (guessing 6 bytes of master key in an experiment) with insignificant data complexity requiring only 9 quadruples of chosen plaintexts.

5 Using Occurrences of Linear Combinations

In the previous sections, we investigated the properties of a given cipher using the number of occurrences of an input word in the expression of the output word (considering both encryption and decryption direction). In contrast to this approach, we now measure the number of occurrences of input words in the linear combination of the output words. A motivation for this concept is that there may exist input words that influence the algebraic expressions of any ciphertext word, but the same word may not occur in a linear combination of the algebraic expressions of the ciphertext words. Thus, certain linear combinations of the ciphertext words may be independent of some input words.

In the case of SKINNY, all its versions achieve full diffusion (forward or backwards) after 6 rounds, but the XOR sum of the third and the fifteen word of the output after six encryption rounds is independent of p_{15}. In fact,

$$\begin{cases} X_6[3] = S(X_5[3]) + S(X_5[9]) + S(X_5[12]) + k_{15} \\ X_6[15] = S(X_5[3]) + S(X_5[9]) + k_{12}, \end{cases} \tag{3}$$

which gives $X_6[3] + X_6[15] = S(X_5[12]) + k_{12} + k_{15}$. Since p_{15} does not occur in the expression of $X_5[12]$, it is independent of $X_6[3] + X_6[15]$. This is a strong truncated differential, requiring a single plaintext/ciphertext pair for distinguishing SKINNY from other block ciphers. In [DL18], Derbez and Lallemand used it to mount a key recovery attack on 12-round SKINNY- 64-128.

Now we extend the distinguisher one more round in the backward direction. Suppose that we vary $X_1[15]$ while keeping $X_1[0], \cdots, X_1[14]$ fixed. From Equation (1), we have $X_1[7] = S(p_3) + k_3$ and $X_1[15] = S(p_3) + k_3 + S(p_9)$, and we need $X_1[7]$ be an fixed constant and $X_1[15]$ be active, which implies p_3 a constant and p_9 a variable. Note that $X_1[3] = S(p_3) + k_3 + S(p_9) + S(p_{12})$, in order to fix $X_1[3]$ to be a constant, $S(p_{12})$ (and hence p_{12}) must be a variable dependent on p_9. To sum up, there are three words of X_1 which depend on p_6, p_9, p_{12},

$$\begin{cases} X_1[3] &= S(p_3) + k_3 + S(p_9) + S(p_{12}) \\ X_1[11] &= S(p_6) + k_2 + S(p_9) \\ X_1[15] &= S(p_3) + k_3 + S(p_9). \end{cases} \tag{4}$$

So we can let p_6, p_9, p_{12} be varied while keeping $S(p_6) + S(p_9), S(p_9) + S(p_{12})$ fixed. That is, for different plaintexts $(p_0, p_1, \cdots, p_{15})$ and $(p'_0, p'_1, \cdots, p'_{15})$, we assume $p_i = p'_i$ for all $i \in \{0, 1, \ldots, 15\} \setminus \{6, 9, 12\}$ and $S(p_6) + S(p_9) = S(p'_6) + S(p'_9), S(p_9) + S(p_{12}) = S(p'_9) + S(p'_{12})$, which implies that the bytes of X_1 and X'_1 are equal except for the fifth byte.

Hence, we have the following result which is a strong truncated differential with one pair of plaintexts and ciphertexts for seven rounds SKINNY.

Theorem 4. *Suppose that two different plaintexts* $(p_0, p_1, \cdots, p_{15})$ *and* $(p'_0, p'_1, \cdots, p'_{15})$ *are selected so that* $p_i = p'_i$ *for all* $i \in \{0, 1, \ldots, 15\} \setminus \{6, 9, 12\}$*, for which additionally*

$S(p_6) + S(p_9) = S(p'_6) + S(p'_9)$ and $S(p_9) + S(p_{12}) = S(p'_9) + S(p'_{12})$. Then the 3rd and 15-th word of $X_7 + X'_7$ are equal. Especially, we can let $p_6 = p_9 = p_{12} \neq p'_6 = p'_9 = p'_{12}$ and the other words of P and P' be equal.

Finally, it is worth to mention that, in this paper, we only focus on the linear layer instead of the details of Sboxes, therefore the numbers of rounds of the distinguishers covered by the integeral property, impossible differential, truncated differential, and zero-sum property are the same, for SKINNY-64 and SKINNY-128.

6 Attacks on SKINNY-128-128 in Single-Key Model

In this section we propose several practical attacks in a single-key model on the SKINNY-128-128 variant. This is done by using our 7-round integral distinguisher and then appending a few more rounds (as much as possible) that can be efficiently handled. We first briefly describe a practical attack on SKINNY-128-128 reduced to 11 rounds, and then investigate the possibility of extending the number of rounds successively.

6.1 Attacking SKINNY-128-128 Reduced to 11 Rounds

Recall that for our 7-round integral distinguisher on SKINNY given in Section 3.3, when p_{12}, p_{13} go through $F_{2^8} \times F_{2^8}$ while keeping the other words of plaintext fixed, the word with coordinate 8 (denoted by $X_7[8]$) sums to zero after 7 encryption rounds. Now we append 4 rounds at the bottom of this integral distinguisher and achieve 11-round attack on SKINNY.

More precisely, for 11-round SKINNY 128-128, let $(c_0, c_1, \cdots, c_{15}) \in (F_{2^8})^{16}$ denote the ciphertext and $(k_0, k_1, \cdots, k_{15}) \in (F_{2^8})^{16}$ denote the master key. Decrypt the ciphertext at the intermediate stage of round seven, which gives the following algebraic representation of $X_7[8]$ as a function of the ciphertext words:

$$\begin{aligned} X_7[8] = V(V(V(V(c_5 + c_9 + c_{13} + k_3 + 1) + V(c_6 + c_{14} + 2) + V(c_3 + c_{15}) + k_{10}) + \\ V(V(c_6 + c_{10} + c_{14} + k_1) + V(c_0 + c_{12})) + V(V(c_6 + k_5) + V(c_1 + c_{13})) + k_1) \\ + V(V(V(c_6 + C_{10} + c_{14} + k_1) + k_{12}) + V(V(c_4 + k_6 + E) + V(c_3 + C_{15}))) + 2). \end{aligned} \quad (5)$$

Note that $X_7[8]$ only depends on 6 key bytes $k_1, k_3, k_5, k_6, k_{10}, k_{12}$. These 6 bytes of the key can be found by exhaustive search. The process of verification consists of a computation of $X_7[8]$ from ciphertexts using the equation (5) and verifying that the sum is equal to zero. In this case, the distinguisher can be seen as an 8-bit sieve.

Having a collection of 2^{16} plaintext/ciphertext pairs, one can recover 8 bits of the involved 48-bit key. Repeating this 6 times will filter out all wrong key guesses, i.e., with $6 * 2^{16}$ pairs of data, the 48-bit key can be recovered uniquely. For other 80 key bits unknown, it can be done with a shifted version ($X_7[9]$ is balanced when p_{13}, p_{14} traverse over $F_{2^8} \times F_{2^8}$ and so on) of the current distinguisher, but with exhaustive searches of less than 6-bytes. So the time complexity is 2^{48} and the data complexity is $6 * 2^{16}$.

6.2 Extending the Number of Rounds for SKINNY-128-128

Note that adding more encryption rounds at the bottom of the 7-round distinguisher slightly increases the data complexity, which is a multiple of 2^{16}, the similar amount as that for 11-round SKINNY-128-128. Concretely, on one hand, for 12-round SKINNY-128-128, the decryption of the ciphertext backs to the end of round seven, there are 8 bytes of master key involved in the expression of $X_7[8]$. On the other hand, for 13-round SKINNY-128-128, there are 14 bytes of master key involved in the expression of $X_7[8]$. Consequently, the

time complexity of these attacks are 2^{64} and 2^{112}, respectively. The data complexity of 12-round and 13-round attack are 8×2^{16} and 14×2^{16}, respectively.

Applying the same approach to SKINNY-128-128 reduced to 14 rounds, shows that all the 16 bytes of the master key are involved in the algebraic representation of $X_7[8]$ and an efficient attack is no more possible. Table 3 summarizes these key bytes appear in the 8-th, 9-th,10-th,11-th words of round 7 represented by the ciphertext words, when decrypting from the ciphertext to round 7. The number of rounds that the decryption starting from varying from 11 to 13.

Table 3: Key bytes involved in $X_7[8], X_7[9], X_7[10], X_7[11]$ in the decryption direction

Round	State word	Guessed Key Bytes
11	$X_7[8]$	$k_1; k_3; k_5; k_6; k_{10}; k_{12}$
	$X_7[9]$	$k_0; k_1; k_4; k_7; k_{11}; k_{13}$
	$X_7[10]$	$k_2; k_5; k_6; k_7; k_8; k_9$
	$X_7[11]$	$k_0; k_2; k_3; k_4; k_{14}; k_{15}$
12	$X_7[8]$	$\text{key}\backslash\{k_0; k_2; k_4; k_7; k_8; k_{14}\}$
	$X_7[9]$	$\text{key}\backslash\{k_2; k_3; k_5; k_6; k_9\}$
	$X_7[10]$	$\text{key}\backslash\{k_0; k_1; k_3; k_4; k_8; k_{12}; k_{15}\}$
	$X_7[11]$	$\text{key}\backslash\{k_1; k_5; k_6; k_7; k_{10}; k_{11}\}$
13	$X_7[8]$	$\text{key}\backslash\{k_8, k_{14}\}$
	$X_7[9]$	$\text{key}\backslash\{k_9\}$
	$X_7[10]$	$\text{key}\backslash\{k_{12}, k_{15}\}$
	$X_7[11]$	$\text{key}\backslash\{k_{10}, k_{11}\}$

Although we can not attack more than 14 rounds just by adding some rounds at the bottom of 7-round integral distinguisher, we can add one to three rounds at the top of the distinguisher and attack 14-round to 16-round SKINNY-128-128. Next, we present an attack on 16-round Skinny-128-128 by adding three rounds before the 7-round integral distinguisher.

6.3 A Theoretical Attack on 16-Round SKINNY-128-128

For 16-round SKINNY-128-128, based on the preceding 13-round attack, we add three more rounds at the top and add 6 rounds at the bottom of the 7-round integral distinguisher, thus considering in total 16 rounds.

Going in the backward direction, from X_3 to the plaintext, one can verify that all bytes of the plaintext except for the 14-th byte depend on $X_3[12]$ or $X_3[13]$. So when $p_0, \cdots, p_{13}, p_{15}$ goes through $(F_{2^8})^{15}$ and p_{14} is fixed to any constant value, there are 2^{112} such structures where each structure leads to $X_3[12], X_3[13]$ being active and the other bytes being constants.

To calculate $X_{10}[8]$ using the ciphertext bytes all of the k_i apart from k_1 and k_4 are needed. These facts give rise to the following attack that can be mounted against 16-round SKINNY-128-128.

1. Encrypt 2^{120} plaintexts with the 14th byte being fixed to a constant value and all the remaining bytes being active for 16 rounds.

2. Guess the key bytes $\{k_0, \ldots, k_{15}\} \setminus \{k_1, k_4\}$ and decrypt the ciphertext to get X_{10}.

3. Sum X_{10} up and check whether the sum is zero.

The data and time complexity of this attack are 14×2^{120} and 2^{112}, respectively.

7 Conclusions

An application of our method to SKINNY gives many 10-round integral distinguishers, all of the 11-round impossible differentials and a strong 7-round truncated differential which can be efficiently used to distinguish 7-round SKINNY from random permutations. Moreover, key recovery attacks on 11- to 16-round SKINNY-128-128 in a single-key model are presented. The attacks are chosen plaintext-ciphertext attacks and are in the single-key model. To the best of our knowledge a key recovery attack on 16 rounds of SKINNY-128-128 is currently the largest number of rounds cryptanalyzed in the single-key model.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions. The first author is supported by the National Natural Science Foundation of China (Grants No. 61672330 and 61602287) and the State Scholarship Fund no.201808370069 from China Scholarship Council. The third author is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Strategic Capability Research Centres Funding Initiative, Nanyang Technological University under grant M4082123, and Singapore's Ministry of Education under grants M4012049, M4012153, and M4020466.

References

[ABC+17] Ralph Ankele, Subhadeep Banik, Avik Chakraborti, Eik List, Florian Mendel, Siang Meng Sim, and Gaoli Wang. Related-key impossible-differential attack on reduced-round skinny. In Dieter Gollmann, Atsuko Miyaji, and Hiroaki Kikuchi, editors, *Applied Cryptography and Network Security - 15th International Conference, ACNS 2017, Kanazawa, Japan, July 10-12, 2017, Proceedings*, volume 10355 of *Lecture Notes in Computer Science*, pages 208–228. Springer, 2017.

[BBS99] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999.

[BC10] Christina Boura and Anne Canteaut. Zero-sum distinguishers for iterated permutations and application to keccak-f and hamsi-256. In *Selected Areas in Cryptography - 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers*, pages 1–17, 2010.

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[DL18] Patrick Derbez and Virginie Lallemand. Submission to the skinny 2018-2019 competition cryptanalytic report. 2018.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[Knu98] Lars Knudsen. A 128-bit block cipher. nist aes proposal. 258(2):216, 1998.

[KR07] Lars R. Knudsen and Vincent Rijmen. Known-key distinguishers for some block ciphers. In *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, pages 315–324, 2007.

[KW02] Lars R. Knudsen and David A. Wagner. Integral cryptanalysis. In *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, pages 112–127, 2002.

[Lai94] Xuejia Lai. Higher order derivatives and differential cryptanalysis. In *Communications and Cryptography*, pages 227–233. Springer, 1994.

[LGS17] Guozhen Liu, Mohona Ghosh, and Ling Song. Security analysis of SKINNY under related-tweakey settings (long paper). *IACR Trans. Symmetric Cryptol.*, 2017(3):37–72, 2017.

[SKI18] https://sites.google.com/site/skinnycipher/cryptanalysis-competition/2018-2019-competition. 2018.

[SLG+16] Bing Sun, Meicheng Liu, Jian Guo, Vincent Rijmen, and Ruilin Li. Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 196–213. Springer, 2016.

[SMB18] Sadegh Sadeghi, Tahereh Mohammadi, and Nasour Bagheri. Cryptanalysis of reduced round SKINNY block cipher. *IACR Trans. Symmetric Cryptol.*, 2018(3):124–162, 2018.

[TAY17] Mohamed Tolba, Ahmed Abdelkhalek, and Amr M. Youssef. Impossible differential cryptanalysis of reduced-round SKINNY. In *Progress in Cryptology - AFRICACRYPT 2017 - 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings*, pages 117–134, 2017.

[Udo18] Aleksei Udovenko. Cryptanalysis of 10-round skinny-128-128 for the skinny 2018-2019 competition. 2018.

[ZR19] Wenying Zhang and Vincent Rijmen. Division cryptanalysis of block ciphers with a binary diffusion layer. *IET Information Security*, 13(2):87–95, 2019.

A The SKINNY linear transformation matrix

$$M_{SK} = \begin{pmatrix}
1&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&1&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0
\end{pmatrix}.$$

B The expression of $X_7[8]$

$S(S(S(S(S(S(p_3)+k_3+S(p_9)+S(p_{12}))+k_{13}+S(S(p_4)+k_4+S(p_{11}))+S(S(p_0)+k_0+S(p_{10})))+k_5+S(S(S(p_0)+k_0)+k_{10}+S(S(p_6)+k_6+S(p_9)))+S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9+S(S(p_5)+k_5+S(p_8)+2)))+k_{14}+S(S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9)+k_2+S(S(S(p_2)+k_2)+k_{12}+S(S(p_4)+k4+S(p_{11}))))+S(S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9+S(S(p_5)+k_5+S(p_8)+2)+S(S(p_1)+k_1+S(p_{11})))+k1+S(S(S(p_1)+k_1)+k_{14}+S(S(p_7)+k_7+S(p_{10}))+2)))+k_6+S(S(S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9+S(S(p_5)+k_5+S(p_8)+2)+S(S(p_1)+k_1+S(p_{11})))+k_1)+k_8+S(S(S(S(p_2)+k_2+S(p_8)+2+S(p_{15}))+k_8)+k_4+S(S(S(p_0)+k_0)+k_{10}+S(S(p_6)+k_6+S(p_9)))))+S(S(S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9+S(S(p_5)+k5+S(p_8)+2)+S(S(p_1)+k_1+S(p_{11})))+k_1+S(S(S(p_1)+k_1)+k_{14}+S(S(p_7)+k_7+S(p_{10}))+2)+S(S(S(p_1)+k_1+S(p_{11})+S(p_{14}))+k_{15}+S(S(p_6)+k_6+S(p_9))))+k_{15}+S(S(S(S(p_1)+k_1+S(p_{11})+S(p_{14}))+k_{15})+k_6+S(S(S(p_3)+k_3)+k_{11}+S(S(p_5)+k_5+S(p_8)+2))+2)))+k_{12})+k_6+S(S(S(S(S(S(p_1)+k_1+S(p_{11})+S(p_{14}))+k_{15}+S(S(p_6)+k_6+S(p_9))+S(S(p_2)+k_2+S(p_8)+2))+k_7+S(S(S(p_2)+k_2)+k_{12}+S(S(p_4)+k_4+S(p_{11})))+S(S(S(p_2)+k_2+S(p_8)+2+S(p_{15}))+k_8+S(S(p_7)+k_7+S(p_{10}))+2))+k_{11}+S(S(S(S(p_2)+k_2+S(p_8)+2+S(p_{15}))+k_8)+k_4+S(S(S(p_0)+k_0)+k_{10}+S(S(p_6)+k_6+S(p_9))))+S(S(S(S(p_2)+k_2+S(p_8)+2+S(p_{15}))+k_8+S(S(p_7)+k_7+S(p_{10}))+2+S(S(p_3)+k_3+S(p_9)))+k_0+S(S(S(p_3)+k_3)+k_{11}+S(S(p_5)+k_5+S(p_8)+2))+2))+k_3)+k_{10}+S(S(S(S(S(p_3)+k_3+S(p_9)+S(p_{12}))+k_{13}+S(S(p_4)+k_4+S(p_{11}))+S(S(p_0)+k_0+S(p_{10})))+k_5+S(S(S(p_0)+k_0)+k_{10}+S(S(p_6)+k_6+S(p_9)))+S(S(S(p_0)+k_0+S(p_{10})+S(p_{13}))+k_9+S(S(p_5)+k_5+S(p_8)+2)))+k_{14})+k_5+S(S(S(S(S(p_1)+k_1+S(p_{11})+S(p_{14}))+k_{15}+S(S(p_6)+k_6+S(p_9))+S(S(p_2)+k_2+S(p_8)+2))+k_7)+k_{12}+S(S(S(S(p_3)+k_3+S(p_9)+S(p_{12}))+k_{13})+k_3+S(S(S(p_1)+k_1)+k_{14}+S(S(p_7)+k_7+S(p_{10}))+2))+2))+2)$