RESEARCH Open Access

Searching for impossible subspace trails and improved impossible differential characteristics for SIMON-like block ciphers

Xuzi Wang[1,2], Baofeng Wu[1,2*], Lin Hou[1,2] and Dongdai Lin[1,2]

Abstract

In this paper, we greatly increase the number of impossible differentials for SIMON and SIMECK by eliminating the 1-bit constraint in input/output difference, which is the precondition to ameliorate the complexity of attacks. We propose an algorithm which can greatly reduce the searching complexity to find such trails efficiently since the search space exponentially expands to find impossible differentials with multiple active bits. There is another situation leading to the contradiction in impossible differentials except for miss-in-the-middle. We show how the contradiction happens and conclude the precondition of it defined as miss-from-the-middle. It makes our results more comprehensive by applying these two approach simultaneously. This paper gives for the first time impossible differential characteristics with multiple active bits for SIMON and SIMECK, leading to a great increase in the number. The results can be verified not only by covering the state-of-art, but also by the MILP model.

Keywords: Impossible differential characteristics, Impossible subspace trails, Miss-from-the-middle, SIMON, SIMECK

Introduction

Due to the continuously growing impact of RFID tags, smart cards and FPGAs, cryptographic algorithms which are suitable for resource-constrained devices become more and more important. During the last decade, a number of lightweight block ciphers, hash functions and stream ciphers were developed by the research community.

The NSA published two lightweight block cipher families SIMON and SPECK in Beaulieu et al. (2015), which are highly optimized and have a better performance for both hardware and software platforms. Although no design rationale or cryptanalysis was given in Beaulieu et al. (2015), SIMON and SPECK draw great attention of researchers, and many cryptanalysis work have been done until now. The designers of SIMON and SPECK gave some design rationale and summarized existing cryptanalysis results in Beaulieu et al. (2017), e.g., linear cryptanalysis and differential cryptanalysis (Liu et al. 2017; AlKhzaimi and Lauridsen 2013; Abdelraheem et al. 2015; Chen and Wang 2016; Shi et al. 2014; Qiao et al. 2016), impossible differential and zero correlation cryptanalysis (Chen et al. 2015; Wang et al. 2014; Chen and Wang 2016), integral cryptanalysis (Kondo et al. 2016; Todo and Morii 2016; Wang et al. 2014; Xiang et al. 2016) and so on for SIMON.

Yang et al. proposed SIMECK in Yang et al. (2015). They use the round function of SIMON with changing the circular-shift parameter (8, 1, 2) into (0, 5, 1), and reuse the round function within the keyschedule. These lead to a better performance than SIMON. Cryptanalysis for SIMECK is similar to that of SIMON when related key is not involved. There are some comparison between them in Kölbl and Roy (2015); Kölbl et al. (2015); Qiao et al. (2016); Liu et al. (2017); Wang et al. (2018).

On the basis of NIST's lightweight cryptography project, which aims at electing cryptographic standards suitable for lightweight applications, a lot of candidates have been

*Correspondence: wubaofeng@iie.ac.cn
[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

submitted. There 32 algorithms left in Round 2. These algorithms are based on either lightweight block ciphers or lightweight hash functions, such as PRESENT (Bogdanov et al. 2007), SIMON and SPECK (Beaulieu et al. 2015), SIMECK (Yang et al. 2015), SKINNY (Beierle et al. 2016), GIFT (Banik et al. 2017), Xoodoo (Daemen et al. 2018), PHOTON (Guo et al. 2011), Spongent (Bogdanov et al. 2011) and so on. Three of the candidates ACE, SPIX and SpoC are based on the sLiSCP family permutations proposed in AlTawy et al. (2018a); AlTawy et al. (2018b) using SIMECK Sbox (AlTawy et al. 2017). The security evaluation of SIMON-like block ciphers becomes more important.

Our contributions In this paper, we further study the impossible differential characteristics for SIMON-like block ciphers. We provide impossible subspace trails for SIMON and SIMECK by searching subspace trails inversely and applying miss-in-the-middle. We also excavate another situation leading to the contradiction defining as miss-from-the-middle, and supplement impossible differentials by applying it. All impossible differentials and impossible subspace trails given in this paper can be verified by the MILP model. Our contributions are threefold.

First, we raise the concept of inverse subspace trail and give its searching algorithms for SIMON-like block ciphers. By applying miss-in-the-middle to inverse subspace trails, we can obtain impossible subspace trails for SIMON and SIMECK. One trail includes a lot of impossible differential characteristics.

Secondly, we study the contradiction condition of the left ones and define it as miss-from-the-middle, as an analog of the well-known method of miss-in-the-middle, since miss-in-the-middle approach cannot covering the state-of-art for SIMECK.

Thirdly, all the impossible differentials for SIMON and SIMECK by considering miss-in-the-middle and miss-from-the-middle can be obtained efficiently without the 1-bit constraint through our algorithm. The great increase in the number of impossible differentials is the precondition for the better attacks.

Related work Biham et al. and Knudsen independently proposed the idea of impossible differential attacks in Biham et al. (1999) and (Knudsen 1998), respectively. In such attacks, the adversary aims to pick out keys which produce differential characteristics with zero probability. All existing impossible differential characteristics of SIMON-like block ciphers are obtained with 1-bit constraint either by combining truncated differential and the miss-in-the-middle approach, or by searching automatically using MILP.

In Wang et al. (2014); Chen et al. (2015); AlKhzaimi and Lauridsen (2013); Abed et al. (2013); Kondo et al. (2016); Boura et al. (2014); Derbez and Fouque (2016); AlTawy et al. (2017); Yang et al. (2015), the impossible differential characteristics for SIMON and SIMECK are all obtained by applying miss-in-the-middle to truncated differentials. The longest impossible differentials include 11/12/13/16/19 rounds for SIMON32/48/64/96/128, respectively; and 11/13/15 for SIMECK32/48/64, respectively. While in Sadeghi and Bagheri (2018) gave 15-round and 17-round impossible differential characteristics for SIMECK48 and SIMECK64 respectively by manually finding the contradiction between two truncated differentials.

Sun et al. raised an automatic searching tool called MILP for high-probability differential and linear characteristics (Sun et al. 2014; Sun et al. 2014). Yu Sasaki and Yosuke Todo gave a new impossible differential search tool by MILP in Sasaki and Todo (2017). However, as shown in Sun et al. (2014), for SIMON-like block ciphers, due to their dependencies of their input bits to the AND operation, the trails obtained using MILP are not guaranteed to be valid. It is unadaptable for SIMON-like block ciphers searching impossible differentials by MILP until Wang et al. provided an accurate MILP model for SIMON-like block ciphers in Wang et al. (2018). They gave impossible differentials of 15-round for SIMECK48 and 17-round for SIMECK64 meeting the result in Sadeghi and Bagheri (2018), and two new 13-round for SIMON64. Leander et al. proposed invariant subspace attack for PRINTcipher in Leander et al. (2011). Later on, Grassi et al. raised the concept of subspace trail crytanalysis in Grassi et al. (2016). Leander et al. gave generic algorithms for searching subspace trails, and applied them to several ciphers including SIMON (Leander et al. 2018); specifically, they gave 6/8/12-round subspace trails for SIMON32/64/128 respectively, and the dimensions of subspaces are 30/62/126 respectively.

Preliminaries

Notations We give the description of the symbols used in this paper as following:

0: A 4-bit vector with all entries equal 0, while 0 represents only one bit.

?: A 4-bit vector with any value.

*: The value of one bit is arbitrary.

$U_i \to U_{i+1}^j$ (in Fig. 2): Inverse subspace trail from U_i to $U_{i+1}^j, j \in J = \{0, 1, ...\}$, which means there are $|J|$ possibilities for U_{i+1}; arbitrary $U_{i+1}^j \to U_i$ is an 1-round essential subspace trail.

X, Y: X is the plaintext, and Y is the ciphertext after one round. X^L, X^R represent the left and right blocks of X, respectively. $X^L[\, i]$ represent the i-th bit of X^L (0 to $n-1$

from left to right), $i \in \{0, 1, \cdots, n-1\}$. $X^L[\,i+a]$ means $X^L[\,(i+a) \bmod n]$, omitting (mod n) for simplicity.

ΔX, ΔY: ΔX is the input difference of the round function, and ΔY for the output.

Description of SIMON and SIMECK The SIMON and SIMECK families of lightweight block ciphers are both based on Feistel construction, using AND as the non-linear operation. The round function of SIMON-like block ciphers is shown in Fig. 1, while the rotation parameters are (8, 1, 2) and (0, 5, 1) for SIMON and SIMECK, respectively. We explicit the parameters for all versions of SIMON and SIMECK in Tables 1 and 2, respectively.

In this paper. we focus on the impossible differential characteristics for SIMON-like block ciphers. For simplicity, we recall the rotation invariance of impossible differentials for SIMON in Wang et al. (2014), and only exhibit the impossible subspace trails and impossible differential characteristics for SIMON and SIMECK with contradiction in the *i-th* bit, $i \in \{0, 1, ..., n-1\}$.

Rotational invariance Assume that $(0, \Delta R_i) \nrightarrow (\Delta L_j, 0)$ is a $(j-i)$-round impossible differential for SIMON where $\Delta R_i, \Delta L_j \in \mathbb{F}_2^n \backslash \{0\}$. Then for any r, where $0 \leqslant r \leqslant n-1$, one can construct a set of $(j-i)$-round impossible differentials as $(0, \Delta R_i \lll r) \nrightarrow (\Delta L_j \lll r, 0)$.

The MILP model Sun et al. proposed an automatic method to evaluate the security of block ciphers using Mixed-integer Linear Programming (MILP) technique in Sun et al. (2014). While for SIMON-like block ciphers, the dependencies of the input bits to the AND operation are not considered. The characteristics for SIMON obtained by MILP in Sun et al. (2014) is not guaranteed to be valid and need a check after solving the model. Sasaki and Todo proposed a new tool searching for impossible differentials using MILP in Sasaki and Todo (2017). Wang et al. gave the accurate MILP model for SIMON-like block ciphers in Wang et al. (2018) which can also be used to search impossible differentials. We verify all the results in this paper by using this MILP model. If the impossible differential characteristic holds, the Gurobi optimizer outputs 'Model is infeasible!'. In this section, we briefly introduce the MILP model for SIMON-like block ciphers.

Constraints imposed by ROTATION-AND Operation. According to (Wang et al. 2018), the ROTATION-AND operation from n bits to n bits can be divided into n groups with 3 input difference bits and 2 output bits in each group, $(\Delta x_i, \Delta x_{i+t}, \Delta x_{i+2t}) \rightarrow (\Delta d_{i-b}, \Delta d_{i+t-b})$. We use Δx and Δd to represent the input and output difference for ROTATION-AND operation respectively, (a, b) for rotation parameters, t=$|a-b|$, i $\in \{0, 1, ..., n-1\}$ (from left to right), omitting mod n for simplicity. Then each group should satisfy the inequalities as following,

$$\Delta x_{i+t} - \Delta x_{i+2t} - \Delta d_{i-b} + \Delta d_{i+t-b} \geq -1$$
$$\Delta x_i + \Delta x_{i+t} - \Delta d_{i-b} \geq 0$$
$$-\Delta x_i + \Delta x_{i+t} + \Delta d_{i-b} - \Delta d_{i+t-b} \geq -1$$

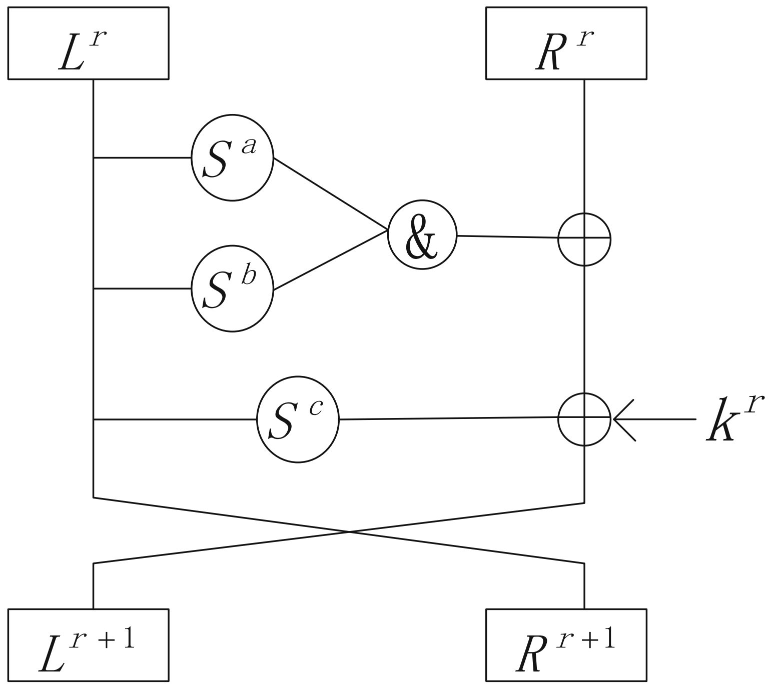

Fig. 1 The round function of SIMON-like block ciphers

Table 1 SIMON parameters

block size 2n	key size mn	rounds T
32	64	32
48	72	36
	96	36
64	96	42
	128	44
96	96	52
	144	54
128	128	68
	192	99
	256	72

$$\Delta x_{i+t} + \Delta x_{i+2t} - \Delta d_{i+t-b} \geq 0$$

Since constraints imposed by the XOR operation are universal, we do not repeat them here for simplicity. During the verification, we fix the input and output difference for the MILP model and run the Gurobi optimizer, if the model is infeasible, the characteristic with this input and output difference is impossible. We claim that all the impossible differentials exhibited in Tables 3 and 4 have been verified by the MILP. The MILP Model subsection ends.

Subspace trails Grassi et al. raised the concept of subspace trails in Grassi et al. (2016), and applied it to AES. Leander et al. gave a generic method for searching subspace trails in Leander et al. (2018). First, we recall the definition of subspace trails.

Definition 1 (Subspace Trail)**.** *Let $F : \mathbb{F}_2^n \leftarrow \mathbb{F}_2^n$. Linear subspaces $U, V \subseteq \mathbb{F}_2^n$ are called a (one round) subspace trail, if*

$$\forall a : \exists b : F(U + a) \subseteq V + b.$$

An $r + 1$-tuple of subspaces $(U_0, \cdots, U_r)$ is called a subspace trail (over r rounds), if

$$U_i \xrightarrow{F} U_{i+1}, \forall i \in \{0, \cdots, r\}.$$

Then, we present the definition of essential subspace trail.

Definition 2 (Essential Subspace Trail)**.** *Let $F : \mathbb{F}_2^n \leftarrow \mathbb{F}_2^m$ and $U \subseteq \mathbb{F}_2^n, V \subseteq \mathbb{F}_2^m$. If $U \xrightarrow{F} V$ forms a subspace trail, i.e. $F(U + a) \subseteq V + b$, and if for all subspaces U' and V' of $\mathbb{F}_2^n$ the following properties hold, we call $U \xrightarrow{F} V$ an essential subspace trail:*

$$\forall U \subset U' : U' \overset{F}{\nrightarrow} V,$$

$$\forall V' \subset V : U \overset{F}{\nrightarrow} V'.$$

Table 2 SIMECK parameters

block size 2n	key size mn	rounds T
32	64	32
48	96	36
64	128	44

Table 3 Impossible differential trails for SIMECK by applying miss-from-the-middle

	Rounds	Impossible differential trails
SIMECK48	15	(000000, 400000) ↛ (800008, 000000)
		(000000, 400000) ↛ (800000, 000000)
		(000000, 800008) ↛ (400000, 000000)
		(000000, 400000) ↛ (000008, 000000)
		(000000, 000008) ↛ (400000, 000000)
		(000000, 800000) ↛ (400000, 000000)
SIMECK64	17	(00000000, 00000002) ↛ (40000046, 00000000)
		(00000000, 40000040) ↛ (00000002, 00000000)
		(00000000, 00000002) ↛ (40000040, 00000000)
		(00000000, 40000046) ↛ (00000002, 00000000)
		(00000000, 40000004) ↛ (00000002, 00000000)
		(00000000, 40000002) ↛ (00000002, 00000000)
		(00000000, 00000002) ↛ (40000004, 00000000)
		(00000000, 00000002) ↛ (40000044, 00000000)
		(00000000, 00000002) ↛ (40000000, 00000000)
		(00000000, 00000002) ↛ (40000042, 00000000)
		(00000000, 40000006) ↛ (00000002, 00000000)
		(00000000, 40000044) ↛ (00000002, 00000000)
		(00000000, 40000042) ↛ (00000002, 00000000)
		(00000000, 00000002) ↛ (40000002, 00000000)
		(00000000, 00000002) ↛ (40000006, 00000000)
		(00000000, 40000000) ↛ (00000002, 00000000)

Truncated differential were introduced in Knudsen (1994), and generalized to subspaces of differences in Blondeau et al. (2017). Grassi et al. and Leander et al. discussed the link between subspace trails and truncated differentials in Grassi et al. (2016); Leander et al. (2018), respectively. We represent truncated differentials with subspace trails in this paper as a consequence of their close relationship.

Impossible differential characteristics can be given by applying miss-in-the-middle to truncated differentials. For most block ciphers, existing impossible differential characteristics can be regarded as impossible subspace trails from a dim-1 subspace to a dim-1 subspace. Grassi et al. raised the concept of impossible subspace trail in Grassi et al. (2016) for the first time, and combined two-round subspaces properties of AES to find impossible subspace trails. A natural question is that, are there any impossible

Table 4 Impossible differential trails for SIMON by applying miss-from-the-middle

	Rounds	Impossible differential trails
SIMON64	13	(00000000, 08000080) ↛ (40000000, 00000000)
		(00000000, 10000000) ↛ (80000000, 00000000)
		(00000000, 40000000) ↛ (08000000, 00000000)
		(00000000, 40000000) ↛ (08000082, 00000000)
		(00000000, 40000000) ↛ (48000083, 00000000)
		(00000000, 40000000) ↛ (08000083, 00000000)
		(00000000, 40000000) ↛ (48000001, 00000000)
		(00000000, 40000000) ↛ (08000001, 00000000)
		(00000000, 48000003) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (00000002, 00000000)
		(00000000, 80000000) ↛ (10000000, 00000000)
		(00000000, 40000002) ↛ (00000002, 00000000)
		(00000000, 48000082) ↛ (40000000, 00000000)
		(00000000, 48000001) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (08000002, 00000000)
		(00000000, 08000082) ↛ (40000000, 00000000)
		(00000000, 00000002) ↛ (40000002, 00000000)
		(00000000, 08000002) ↛ (40000000, 00000000)
		(00000000, 00000001) ↛ (20000000, 00000000)
		(00000000, 00000002) ↛ (40000000, 00000000)
		(00000000, 80000000) ↛ (00000004, 00000000)
		(00000000, 08000001) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (08000003, 00000000)
		(00000000, 08000003) ↛ (40000000, 00000000)
		(00000000, 48000083) ↛ (40000000, 00000000)
		(00000000, 00000004) ↛ (80000000, 00000000)
		(00000000, 20000000) ↛ (00000001, 00000000)
		(00000000, 48000000) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (48000082, 00000000)
		(00000000, 48000081) ↛ (40000000, 00000000)
		(00000000, 08000081) ↛ (40000000, 00000000)
		(00000000, 90000000) ↛ (80000000, 00000000)
		(00000000, 48000080) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (48000080, 00000000)
		(00000000, 40000000) ↛ (48000002, 00000000)
		(00000000, 40000000) ↛ (08000080, 00000000)
		(00000000, 48000002) ↛ (40000000, 00000000)
		(00000000, 08000083) ↛ (40000000, 00000000)
		(00000000, 08000000) ↛ (40000000, 00000000)
		(00000000, 40000000) ↛ (48000081, 00000000)
		(00000000, 40000000) ↛ (48000003, 00000000)
		(00000000, 40000000) ↛ (08000081, 00000000)
		(00000000, 80000000) ↛ (90000000, 00000000)
		(00000000, 40000000) ↛ (48000000, 00000000)

subspace trails for other block ciphers? Intuitively, considering miss-in-the-middle approach, if there exist two subspace trails whose holding probabilities are both 1: $X \xrightarrow{F} Y, Z \xrightarrow{F^{-1}} W$ such that $Y \cap W = \emptyset$, then there exists an impossible subspace trail from X to Z.

Leander et al. gave a generic approach for searching subspace trails in Leander et al. (2018). For S-box layers without linear structures, i.e. word-based block ciphers, a subspace trail starting with subspace U_0 which has only one active S-box is provably optimal. For those with linear structures, i.e. bit-based block ciphers, a subspace trail starting with subspace U_0 which has only one active bit is not necessarily optimal. However, since it costs too much time ($O(2^n)$) to traverse all dim-1 subspaces, existing searching algorithms only consider that U_0 has only one active bit. This highly limits the number of impossible differential characteristics which can be found. To this end, we raise the concept of *inverse subspace trail*. Similar to subspace trails, we also need to find an essential trail $U \xrightarrow{F} V$ such that $dim(U) \leqslant dim(V)$. Their difference is that for subspace trails, it asks us to compute V given U, when V is unique; however, for inverse subspace trials, it asks us to compute U given V, when U has many possibilities. In this paper, we refer these subspace trails which are searched inversely as inverse subspace trails.

An $r+1$-tuple of subspaces $(U_0, \cdots, U_r)$ is called an inverse subspace trail (over r rounds), if

$$U_{i+1} \xrightarrow{F} U_i, dim(U_i) \geqslant dim(U_{i+1}), \forall i \in \{0, \cdots, r-1\}.$$

For two inverse subspace trails $(U_0, \cdots, U_{r_a})$ and $(V_0, \cdots, V_{r_b})$, if $U_0 \cap V_0 = \emptyset$, then we have a $(r_a + r_b - 1)$-round impossible subspace trail. Different with previous work, we consider the case where $dim(U_{r_a}), dim(U_{r_b}) \geq 1$, and under this condition, one impossible subspace trail may contain much more impossible differential characteristics, whose input and output difference may have more than one active bits. In this paper, we greatly increase the number of impossible differential characteristics for SIMON and SIMECK. In addition, we reveal another reason leading to the contradiction of impossible differential trails.

Automatic search of impossible subspace trails

For a subspace V of high dimension, there exist possibly multiple essential U, such that $dim(U) \leqslant dim(V)$ and $U \xrightarrow{F} V$. Thus, as the round increasing, branches will increase exponentially. In general, we need to traverse all branches to find the longest trail, of which the complexity is $O(2^n)$. This is why for most bit-based block ciphers, we cannot find the longest trail by searching inversely. However, for SIMON-like block ciphers, things are different. They show some special property regarding difference

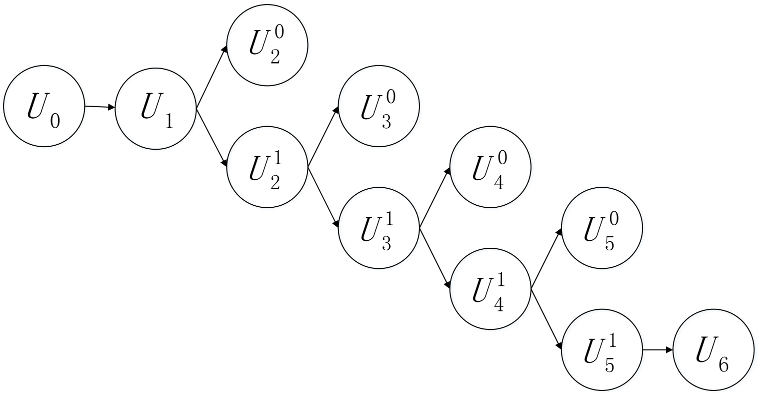

Fig. 2 Inverse subspace trails for SIMON32

(inverse) diffusion, which leads to 2^R possible branches at most for an R-round inverse subspace trail. The reason is that SIMON-like block ciphers have an special difference property

and we descript it in Theorem 2. Then it is feasible to traverse inversely all subspace trails to find the longest one, and give the longest impossible subspace trails combing the miss-in-the-middle approach. In this section, we explain the special property of SIMON-like block ciphers in detail, and present results of impossible subspace trails for SIMON and SIMECK.

Search strategy Leander et al. gave a searching algorithm for subspace trails in Leander et al. (2018). They started with a dim-1 subspace, and made the error probability negligible by using plenty of plaintext. Their method was applied to analyzing several block ciphers, and their results of subspace trails met well with existing truncated differentials. Note that for word-based block ciphers, starting searching from a dim-1 subspace with one active S-box will always lead to provable optimal subspace trails. However, it is not the case for bit-based block ciphers. For bit-based block ciphers, the complexity of traversing all dim-1 subspaces is $O(2^n)$, where n is the block size. The time cost is too high and this is why Leander et al. chose to traverse all dim-1 subspaces with one active bit.

For searching impossible differential characteristics of bit-based block ciphers, similar question exists. Whether applying miss-in-the-middle to truncated differentials or automatically searching by MILP, it will give the trails with only one active bit in the input and output differences. Intuitively, it is seemingly reasonable to search out the longest impossible differential trails on this condition considering the diffusion property of block ciphers. However, do the longest trails only exist under this condition? In this subsection, we go further into this question by searching subspace trails inversely and applying miss-in-the-middle approach.

For searching subspace trails, it takes too much time to traverse all 1-dim subspaces, so we choose to search subspace trails from the opposite direction. To be exact, we start with a subspace V of high dimension, e.g., $dim(V) = n - 1$, and as the round increasing, the dimension will decrease. Note that an essential trail from a low-dimension subspace to a high-dimension subspace over the round function is unique, but the inverse is not true. This means there may exist several trails from a high-dimension subspace to a low-dimension one. We refer readers to Fig. 2 which demonstrates the case of SIMON32 and Table 5 which exhibits the value of variables used in Fig. 2.

Straightforwardly, it takes too much time by using either strategy as aforementioned, and this is due to the XOR operation. For the equation $a \oplus b = c$, when the value of one variable is fixed, the values of the rest two variables take two possibilities, which leads to two branches.

Table 5 Symbols used in depicting Fig. 2. For simplicities, we denote 0000 by **0** and arbitrary four bits by **?**

Symbol	Description	Dimension
U_0	(1***|**?**|**?**|**?**,**?**|**?**|**?**|**?**)	31
U_1	(**?**|**?**|**?**|**?**,1***|**?**|**?**|**?**)	31
U_2^0	(1***|**?**|**?**|**?**,*00*|**?**|0***|**?**)	28
U_2^1	(0***|**?**|**?**|**?**,*01*|**?**|0***|**?**)	28
U_3^0	(*01*|**?**|0***|**?**,0*00|0***|*00*|**?**)	23
U_3^1	(*00*|**?**|0***|**?**,0*00|1***|*00*|**?**)	23
U_4^0	(0*00|1***|*00*|**?**,*000|000*|0*00|0***)	16
U_4^1	(0*00|0***|*00*|**?**,*000|001*|0*00|0***)	16
U_5^0	(*000|001*|0*00|0***,0*00|**0**|**0**|000*)	8
U_5^1	(*000|000*|0*00|0***,0*00|**0**|1000|000*)	8
U_6	(0*00|**0**|1000|000*,**0**|**0**|**0**|**0**)	2

Thus intuitively, for SIMON-like block ciphers, as the round increasing, the branches will increase exponentially. However, by our observation, for SIMON-like block ciphers, not every fixed bit will lead to branches, as shown in Theorem 2, and to be specific, only a small amount of them do. Assume that the initial subspace has only one fixed bit of difference and we denote the rounds of inverse subspace trail by r, then the branches takes at most 2^r. Hence, we prefer the width first strategy to search the inverse subspace trails for SIMON-like block ciphers, as demonstrated by Algorithm 1, then obtain impossible subspace trails by applying the miss-in-the-middle approach.

Theorem 1 (Difference property for ROTATION-AND)**.** *Let $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n$, $F = S^a(x) \odot S^b(x)$. Δx and Δd represent the input and output differences for F, respectively. If $\Delta d_i = 0$ with probability 1, then $\Delta x_{i+a} = 0$ and $\Delta x_{i+b} = 0$.*

Proof We have

$$\Delta d_i = \Delta x_{i+a} \odot x_{i+b} \oplus \Delta x_{i+b} \odot x_{i+a}$$

If $P(\Delta d_i = 0) = 1$, then the value of Δd should not be affected by any bit of the plaintext. It is easy to get that $\Delta x_{i+a} = 0$ and $\Delta x_{i+b} = 0$. □

Theorem 2 (Difference property for round function of SIMON-like block ciphers). *Let $F : \mathbb{F}_2^{2n} \rightarrow \mathbb{F}_2^{2n}$, $F\left(X^L, X^R\right) = \left(S^a(X^L) \odot S^b(X^L) \oplus S^c(X^L) \oplus X^R, X^L\right) = \left(Y^L, Y^R\right)$. ΔX and ΔY represent the input and output differences for function F, respectively. Superscript L and R represent the left and right block, respectively.*

1. If the value of $\Delta Y^L[i]$ is fixed with probability 1, then $\Delta X^L[i+a] = 0$, $\Delta X^L[i+b] = 0$ and $\Delta X^L[i+c] \oplus \Delta X^R[i] = \Delta Y^L[i]$, $i \in \{0, 1, \cdots, n-1\}$.

2. If $\Delta Y^L[i+a]$, $\Delta Y^L[i+b]$ and $\Delta Y^L[i+c]$ are all fixed with probability 1, then there exist 2 branches for the value of ΔX instead of 2^3. $\Delta X^L[i+2a, i+a+b, i+2b, i+a+c, i+b+c] = 0$, $\Delta X^R[i+a] = \Delta Y^L[i+a]$, $\Delta X^R[i+b] = \Delta Y^L[i+b]$. If $\Delta Y^L[i+c] = 0$, the two branches are $\left(\Delta X^R[i+c], \Delta X^L[i+2c]\right) = (0,0)$ and $\left(\Delta X^R[i+c], \Delta X^L[i+2c]\right) = (1,1)$, respectively; If $\Delta Y^L[i+c] = 1$, the two branches are $\left(\Delta X^R[i+c], \Delta X^L[i+2c]\right) = (0,1)$ and $\left(\Delta X^R[i+c], \Delta X^L[i+2c]\right) = (1,0)$, respectively.

Proof The first point is easy to prove according to Theorem 1. Then we prove the second point in the following. We have

$$\begin{aligned}\Delta Y^L[i+a] = \Delta X^L[i+2a] \odot X^L[i+a+b] \\ \oplus \Delta X^L[i+a+b] \odot X^L[i+2a] \\ \oplus \Delta X^L[i+a+c] \oplus \Delta X^R[i+a],\end{aligned}$$

$$\begin{aligned}\Delta Y^L[i+b] = \Delta X^L[i+a+b] \odot X^L[i+2b] \\ \oplus \Delta X^L[i+2b] \odot X^L[i+a+b] \\ \oplus \Delta X^L[i+b+c] \oplus \Delta X^R[i+b],\end{aligned}$$

$$\begin{aligned}\Delta Y^L[i+c] = \Delta X^L[i+a+c] \odot X^L[i+b+c] \\ \oplus \Delta X^L[i+b+c] \odot X^L[i+a+c] \\ \oplus \Delta X^L[i+2c] \oplus \Delta X^R[i+c].\end{aligned}$$

If $\Delta Y^L[i+a]$, $\Delta Y^L[i+b]$ and $\Delta Y^L[i+c]$ are all fixed with probability 1, it is easy to know that $\Delta X^L[i+2a, i+a+b, i+2b, i+a+c, i+b+c] = 0$.

Finally,

$$\begin{aligned}\Delta Y^L[i+a] = \Delta X^R[i+a] \\ \Delta Y^L[i+b] = \Delta X^R[i+b] \\ \Delta Y^L[i+c] = \Delta X^R[i+c] \oplus \Delta X^L[i+2c]\end{aligned}$$

□

According to Theorem 2, we find that the number of the branches of the inverse

subspace trails for SIMON-like block ciphers does not increase exponentially. It

increases with the number of rounds, reaching 2^R at most which is probably to traverse.

Algorithm 1. *(Inverse Subspace Trail Searching Algorithm)*

Input: *a subspace list $V = [v_0, v_1, \cdots], v = (L^r[0, 1, \cdots, n], R^r[0, 1, \cdots, n])$*

Output: *longest inverse subspace trails U*

```
$U, W \leftarrow \emptyset$
$round \leftarrow 0$
while $len(V) \neq 0$ do
    $U \leftarrow V$
    for $v \in V$ do
        $W \leftarrow W \cup$ COMPUTE_SUBSPACE($v$)
    end for
    $V \leftarrow W$,
    $round \leftarrow round + 1$
end while
function COMPUTE_SUBSPACE($v$)
$V \leftarrow \emptyset$
$R^{r+1} \leftarrow L^r$
for $i \in [0, n-1]$ do
    if the value of $R^r[i]$ is determined then
        $R^{r+1}[(i+a) \mod n] \leftarrow 0$
        $R^{r+1}[(i+b) \mod n] \leftarrow 0$
    end if
end for
for $i \in [0, n-1]$ do
    if $R^r[i] = 1$ then
        if the value of $R^{r+1}[(i+c) \mod n]$ is not determined then
```

```
            $possible1.add(i)$
        **else**
            **if** $R^{r+1}[(i+c) \mod n] = 1$ **then**
                $L^{r+1}[i] \leftarrow 0$
            **end if**
            **if** $R^{r+1}[(i+c) \mod n] = 0$ **then**
                $L^{r+1}[i] \leftarrow 1$
            **end if**
        **end if**
    **end if**
    **if** $R^{r}[i] = 0$ **then**
        **if** the value of $R^{r+1}[(i+c) \mod n]$ is not determined **then**
            $possible0.add(i)$
        **else**
            **if** $R^{r+1}[(i+c) \mod n] = 1$ **then**
                $L^{r+1}[i] \leftarrow 1$
            **end if**
            **if** $R^{r+1}[(i+c) \mod n] = 0$ **then**
                $L^{r+1}[i] \leftarrow 0$
            **end if**
        **end if**
    **end if**
**end for**
$v \leftarrow (L^{r+1}, R^{r+1})$
compute all possible combinations for $possible0$ and $possible1$, denote as $item0$ and $item1$, respectively
**for** each $item0$ and each $item1$ **do**
    **for** $i0 \in item0$ **do**
        $temp\_R^{r+1}[(i0+c) \mod n] \leftarrow 1$
        $temp\_L^{r+1}[i0] \leftarrow 1$
    **end for**
    **for** $j0 \in possible0 - item0$ **do**
        $temp\_R^{r+1}[(j0+c) \mod n] \leftarrow 0$
        $temp\_L^{r+1}[j0] \leftarrow 0$
    **end for**
    **for** $i1 \in item1$ **do**
        $temp\_R^{r+1}[(i1+c) \mod n] \leftarrow 1$
        $temp\_L^{r+1}[i1] \leftarrow 0$
    **end for**
    **for** $j1 \in possible1 - item1$ **do**
        $temp\_R^{r+1}[(j1+c) \mod n] \leftarrow 0$
        $temp\_L^{r+1}[j1] \leftarrow 1$
    **end for**
    $temp\_v \leftarrow (temp\_L^{r+1}, temp\_R^{r+1})$
    **if** there is a combination in $v$ and $temp\_v$ **then**
        break
    **else**
        $V.add(v)$
    **end if**
**end for**
```

Impossible subspace trails found in SIMON and SIMECK Impossible differential characteristics obtained either by searching automatically using MILP, or by applying the miss-in-the-middle method to truncated differentials are limited to the case that both the input and output differential have only one active bit. In this part, we give impossible subspace trails, which yields impossible differential characteristics with multi active bits in input/output difference leading to exponential increase in the number.

We use Algorithm 1 and miss-in-the-middle approach to search impossible subspace trails for SIMON and SIMECK. For SIMON32/48/64/96/128, we give 11/12/13/16/19-round impossible subspace trails respectively, as shown in Table 6. Except those for SIMON64, our results cover the state-of-art and show much more impossible differential trails. For SIMON64, two impossible differential trails in Wang et al. (2018) cannot be included for present, since they do not meet the requirements of miss-in-the-middle. We will show in next section how to search these two trails and much more. Our results of SIMECK are listed in Table 7. For SIMECK32, our results cover the state-of-art (11-round) and shows much more trails as well. Sadeghi et al. and Wang et al. gave 15-round and 17-round impossible differential trails for SIMECK48 and SIMECK64 in Sadeghi and Bagheri (2018); Wang et al. (2018), respectively. We will show in next section how to search these two trails and much more. Due to the rotation invariance of SIMON-like block ciphers, in both tables, we only give impossible subspace trails whose contradiction happens in the 0^{th} bit in the middle round.

Impossible differential characteristics by applying miss-from-the-middle approach

Two 13-round trails for SIMON64, 15-round trails for SIMECK48 and 17-round trails for SIMECK64 do not meet the requirement of the miss-in-the-middle approach, so they cannot be given by our algorithm in last section. In Sadeghi and Bagheri (2018), impossible differential trails for SIMECK48 and SIMECK64 are manually deduced, and this procedure tells us how the contradiction happens. However, random two properties will not lead to this contradiction. We reveal the precondition and generalize it into *miss-from-the-middle* as an analog of the miss-in-the-middle approach, which means that the contradiction does not happen right in the middle round but it results from the middle round.

In this section, we give impossible differential characteristics by applying the miss-from-the-middle approach. Our results not only recover the state-of-art, but also give much more trails since we remove the restriction that both the input and output difference have only one active bit.

Miss-from-the-middle We recall the miss-in-the-middle approach. First, we obtain two truncated differences along the encryption and the decryption

Table 6 Impossible subspace trails for SIMON. For simplicities, we denote 0000 by **0** and arbitrary four bits by **?**

<table>
<tr><th></th><th>Rounds</th><th>Impossible subspace trails</th></tr>
<tr><td>SIMON32</td><td>11</td><td>(**0|0|0|0**, 0 ∗ 00|**0|0**|000∗) ↛ (0 ∗ 00|**0**|1000|000∗, **0|0|0|0**)
(**0|0|0|0**,0*00|**0**|1000|000*) ↛ (0*00|**0|0**|000*,**0|0|0|0**)</td></tr>
<tr><td>SIMON48</td><td>12</td><td>(**0|0|0|0|0|0**, **0|0|0|0|0**|000*) ↛ (1000|**0|0**|000*|0*00|0***, **0|0|0|0**|y**0**|000*)
(**0|0|0|0|0**|000*,1000|**0|0**|000*|0*00|0***) ↛ (**0|0|0|0|0**|000*,**0|0|0|0|0|0**)</td></tr>
<tr><td>SIMON64</td><td>13</td><td>(**0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0**|000*) ↛ (0*00|**0**|1000|000*|0*00|0***|*00*|**?**,*000|**0|0|0|0**|000*|0*00|0***)
(*000|**0|0|0|0**|000*|0*00|0***,0*00|**0**|1000|000*|0*00|0***|*00*|**?**) ↛ (0*00|**0|0|0|0|0|0**|000*,**0|0|0|0|0|0|0|0**)</td></tr>
<tr><td>SIMON96</td><td>16</td><td>(**0|0|0|0|0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0|0|0|0|0**|000*) ↛ (*010|**0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**,0*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**)
(0*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**,*010|**0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**) ↛ (0*00|**0|0|0|0|0|0|0|0|0|0**|000*,**0|0|0|0|0|0|0|0|0|0|0|0**)</td></tr>
<tr><td>SIMON128</td><td>19</td><td>(**0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0|0|0|0|0|0|0|0|0**|000*) ↛ (1*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?|?|?**,*000|**0|0|0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**)
(**0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0|0|0|0|0|0|0|0|0**|000*) ↛ (1*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?|?|?**,*010|**0|0|0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**)
(*000|**0|0|0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**, 1*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?|?|?**) ↛ (**0|0|0|0|0|0|0|0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0|0|0|0|0|0|0|0|0**|000*)
(*010|**0|0|0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?**, 1*00|**0|0|0|0|0|0**|000*|0*00|0***|*00*|**?**|0***|**?|?|?**) ↛ (**0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0**,0*00|**0|0|0|0|0|0|0|0|0|0|0|0|0|0**|000*)</td></tr>
</table>

direction respectively. Then, if there exists some contradiction between the ends of them, we obtain an impossible differential trail. In contrast, the miss-from-the-middle approach does not require such *direct* contradictions. To be exact, first, we combine two truncated differentials, then from the middle, we check down-up and up-down to find whether the combining leads to contradictions with two properties; if so, an impossible differential trail is obtained.

While two random truncated differentials will not cause this kind of contradictions, the combined middle round should satisfy some conditions. Here, we give the precondition of miss-from-the-middle approach, through which we can greatly reduce the searching space.

Precondition of miss-from-the-middle Let $T_1 = (U_0, U_1, \cdots, U_{r_a})$ and $T_2 = (V_0, V_1, \cdots, V_{r_b})$, which are inverse subspace trails of round function F and its inverse F^{-1}, respectively. Denote $M = U_0 \cap V_0 = (L[0, 1,\cdots, n-1], R[0, 1,\cdots, n-1])$. If

1 $L[i+a, i+b] = 0$, $L[i+c]$ and $R[i]$ are fixed for some $i \in \{0, 1,\cdots, n-1\}$, there may exist contradictions between M and T_2; or

Table 7 Impossible subspace trails for SIMECK. For simplicities, we denote 0000 by **0** and arbitrary four bits by **?**

	Rounds	Impossible subspace trails
SIMECK32	11	$(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0},\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0)\not\rightarrow(0001\mid *000\mid **00\mid ***0,\mathbf{0}\mid\mathbf{0}\mid 0*00\mid 0**0)$
		$(\mathbf{0}\mid\mathbf{0}\mid 0*00\mid 0**0,0001\mid *000\mid **00\mid ***0)\not\rightarrow(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0,\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0})$
SIMECK48	13	$(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0},\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 000*\mid 000*)\not\rightarrow(\mathbf{0}\mid 1000\mid 0*00\mid 0**0\mid 0***\mid 0***,\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0\mid 00**\mid 00**)$
		$(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0\mid 00**\mid 00**,\mathbf{0}\mid 1000\mid 0*00\mid 0**0\mid 0***\mid 0***)\not\rightarrow(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 000*\mid 000*,\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0})$
SIMECK64	15	$(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0},\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 0*00)\not\rightarrow(\mathbf{0}\mid 1000\mid 0*00\mid 0**0\mid 0***\mid 0***\mid\mathbf{?}\mid\mathbf{?},\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0\mid 00**\mid 00**\mid *0**\mid\mathbf{?})$
		$(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 00*0\mid 00**\mid 00**\mid *0**\mid\mathbf{?},\mathbf{0}\mid 1000\mid 0*00\mid 0**0\mid 0***\mid 0***\mid\mathbf{?}\mid\mathbf{?})\not\rightarrow(\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid 0*00,\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0}\mid\mathbf{0})$

2 $R[i+a,i+b]=0$, $R[i+c]$ and $L[i]$ are fixed for some $i\in\{0,1,\cdots,n-1\}$, there may exist contradictions between M and T_1

Determining algorithm We introduce the searching and determining algorithms in details for impossible differential characteristics satisfying miss-from-the-middle. There are three steps in this procedure, reducing the scope, picking & rebuilding and determining, as shown in Fig. 3. The detailed procedure is in the following:

Step 1: Reducing the scope. We construct subspaces V_e and V_d satisfying the precondition of miss-from-the-middle. We take V_e and V_d as the starting points of searching subspace trails inversely along the encryption and decryption directions. Then we can obtain longest subspace trails $U_e \xrightarrow{F} V_e$ and $U_d \xrightarrow{F^{-1}} V_d$. Note that the subspace trail from U_e to U_d is not necessarily impossible. However, this step greatly reduces the searching scope, namely from 2^{2n} to $|U_e|\times|U_d|$.

Step 2: Picking & Rebuilding. We randomly pick dim-1 subspaces U_0 and U_1 of U_e and U_d respectively, and take them as the starting points to search subspace trails. We can obtain $U_0 \xrightarrow{F} V_0$ and $U_1 \xrightarrow{F^{-1}} V_1$ such that $V_0\subseteq V_e$ and $V_1\subseteq V_d$.

Step 3: Determining. We combine V_0 and V_1, then trace back along two directions to check if any contradiction exists. If so, we obtain an impossible differential characteristic.

We formalize the whole procedure into Algorithm 2.

Algorithm 2. *(Impossible Differential Characteristics Sieving Algorithm)*

Input: *Subspace trail list* T, $T=[t_0,t_1,\cdots],t_i=[TF_i,TL_i]$

Output: *Impossible differential characteristics list* C

```
$C \leftarrow \emptyset$
for $t \in T$ do
  $m = TF[-1] \cap TL[-1]$
  # m is the middle round subspace by combining TF and TL
  # $m = (mL[0,1,\cdots,n-1], mR[0,1,\cdots,n-1])$
  $flag \leftarrow firstpart\_contra(TF, m)$
  if $flag = TRUE$ then
    $C.add([TF[0], TL[0]])$
  end if
  $flag \leftarrow lastpart\_contra(TL, m)$
  if $flag = TRUE$ then
    $C.add([TF[0], TL[0]])$
  end if
end for
function firstpart_contra(TF, m):
for $r \in [1, len(TF)-1]$ do
  for $i \in [0, n-1]$ do
    if $mR[(i+a) \mod n] = 0$ and $mR[(i+b) \mod n] = 0$ then
      if $mL[i] = 0$ then
        if $mR[(i+c) \mod n] = 0$ then
          $temp\_R[i] \leftarrow 0$
        end if
        if $mR[(i+c) \mod n] = 1$ then
          $temp\_R[i] \leftarrow 1$
        end if
      end if
      if $mL[i] = 1$ then
        if $mR[(i+c) \mod n] = 0$ then
          $temp\_R[i] \leftarrow 1$
        end if
        if $mR[(i+c) \mod n] = 1$ then
          $temp\_R[i] \leftarrow 0$
        end if
      end if
    end if
  end for
```

```
    $temp \leftarrow (temp\_L, temp\_R)$
    if there is a contradiction in $TF[len(TF) - 1 - r]$ and $temp$ then
        return true
    else
        $m \leftarrow TF[len(TF) - 1 - r] \cap temp$
    end if
end for
return false
```

Impossible differential characteristics for SIMON and SIMECK. We give many impossible differential characteristics whose input and output differences have multiple active bits. Before our work, this cannot be achieved since the high complexity of $O(2^{2n})$. The impossible differentials for SIMON and SIMECK by applying miss-from-the-middle are listed in Tables 3 and 4, respectively. We have verified all the results by MILP model in Wang et al. (2018). To make the verification obviously, we show the 13-round complete impossible differential trail (00000000, 48000083) ↛ (40000000, 00000000) for SIMON64 and how the contradiction happens in Fig. 4.

Conclusion

In this paper, we make use of the diffusion property of SIMON-like block ciphers and give a specific approach for searching inverse subspace trials. In contrast to previous work, the low-dimension subspace in our work has dimension no less than one, rather than strictly one. By applying miss-in-the-middle and miss-from-the-middle, we give results of impossible differential characteristics for SIMON and SIMECK. We hope these results can provide support for cryptanalyst or help designers to make better parameter choices.

For future works, here are some interesting questions. First, whether miss-from-the-middle and miss-in-the-middle can cover all possible cases? If this can be proved, then our results turn to be provably optimal. Combining with attacks, we can easily give a security margin. Secondly, Boura derived generic complexity analysis formulas for impossible differential attacks and optimized it

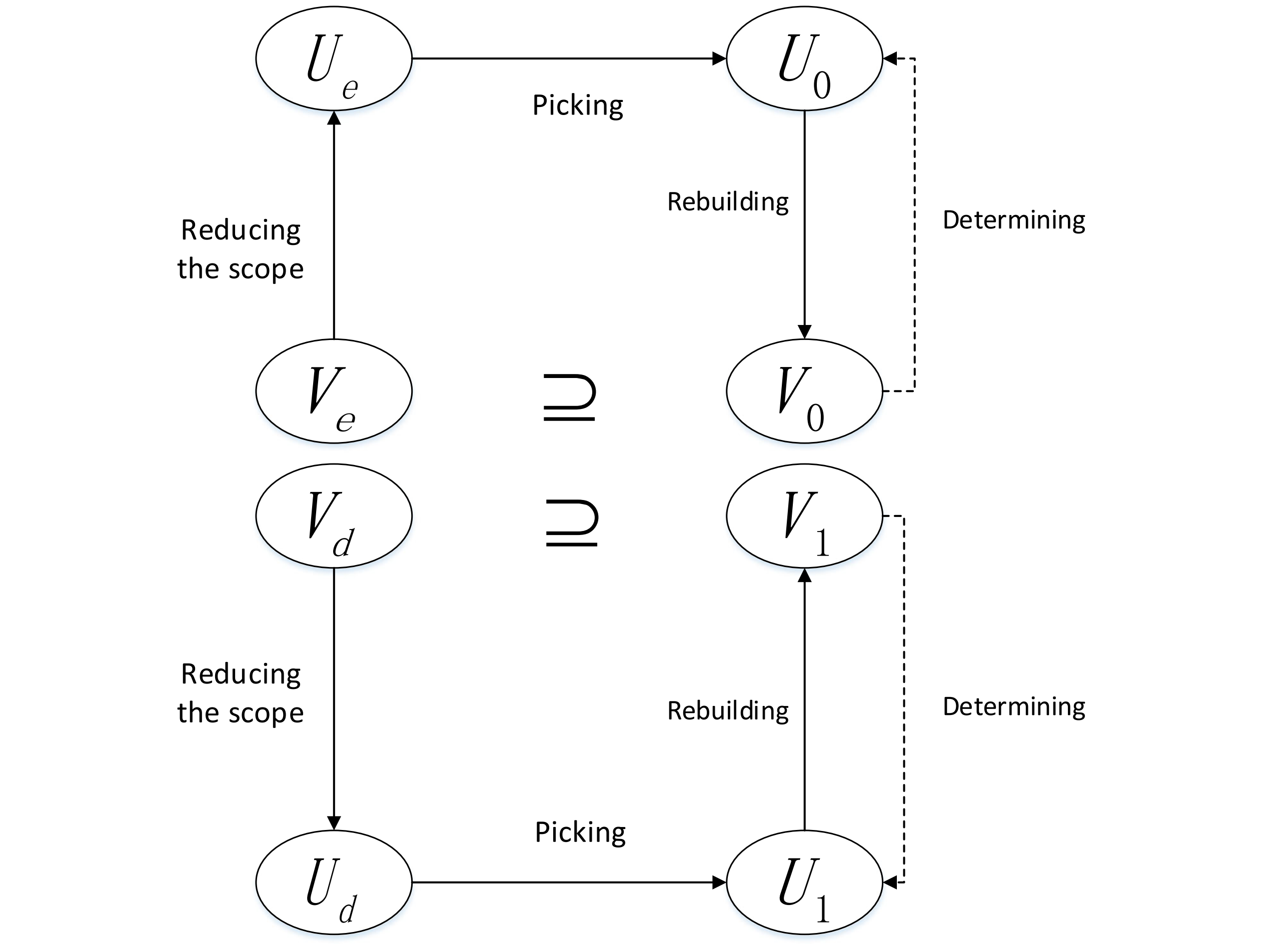

Fig. 3 The searching procedure for impossible differential characteristics

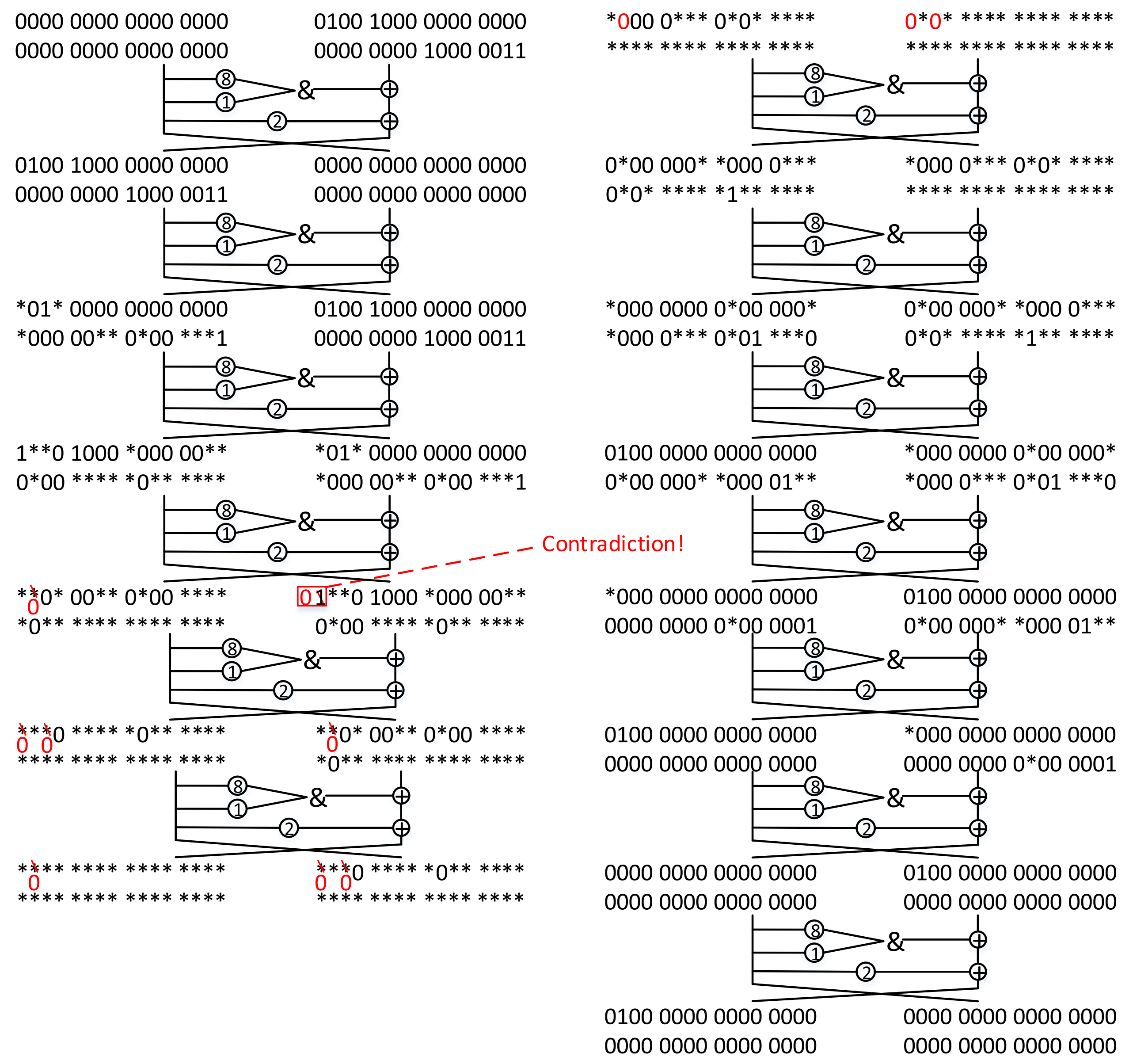

Fig. 4 Impossible differential trail (00000000, 48000083) ↛ (40000000, 00000000) for 13-round SIMON64 shows how the contradiction happens by applying miss-from-the-middle

by using multiple impossible differentials in Boura et al. (2014). However, for this analysis to be valid, the number of conditions associated to the impossible differential attack should stay the same. Since we have greatly expand the set of candidate trails, how to search those qualified trails automatically seems an attracting question. If this can be achieved, we may hopefully give better attack complexity and rounds. Lastly, we want to know whether miss-from-the-middle or inverse subspace trails can be applied to other block ciphers. A generic method for searching inverse subspace trails automatically would be much desired.

Acknowledgements
Not applicable.

Authors' contributions
Xuzi Wang proposed the impossible subspace trails searching algorithm for SIMON-like block ciphers. Xuzi Wang, Baofeng Wu and Lin Hou drafted the manuscript. Baofeng Wu proofread the theorems and algorithms in the manuscript. Dongdai Lin participated in improvements of the manuscript. All authors read and approved the final manuscript.

Funding
This research is supported by the National Natural Science Foundation of China (61972393, 61872359).

Availability of data and materials
Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 23 November 2020 Accepted: 8 February 2021
Published online: 02 June 2021

References

Abdelraheem MA, Alizadeh J, AlKhzaimi HA, Aref MR, Bagheri N, Gauravaram P (2015) Improved linear cryptanalysis of reduced-round SIMON-32 and SIMON-48. In: Biryukov A, Goyal V (eds). 16th International Conference on Cryptology in India Vol. 9462. pp 153–179. https://doi.org/10.1007/978-3-319-26617-6_9

Abed F, List E, Lucks S, Wenzel J (2013) Differential and linear cryptanalysis of reduced-round SIMON. IACR Cryptol ePrint Arch 2013:526

AlKhzaimi H, Lauridsen MM (2013) Cryptanalysis of the SIMON family of block ciphers. IACR Cryptol ePrint Arch 2013:543

AlTawy R, Rohit R, He M, Mandal K, Yang G, Gong G (2017) sLiSCP: Simeck-based permutations for lightweight sponge cryptographic primitives. In: Adams C, Camenisch J (eds). 24th International Conference on Selected Areas in Cryptography Vol. 10719. pp 129–150. https://doi.org/10.1007/978-3-319-72565-9_7

AlTawy R, Rohit R, He M, Mandal K, Yang G, Gong G (2018) Towards a cryptographic minimal design: The sLiSCP family of permutations. IEEE Trans Comput 67(9):1341–1358. https://doi.org/10.1109/TC.2018.2811467

AlTawy R, Rohit R, He M, Mandal K, Yang G, Gong G (2018) sLiSCP-light: Towards hardware optimized sponge-specific cryptographic permutations. ACM Trans. Embedded Comput Syst 17(4):1–26. https://doi.org/10.1145/3233245

Banik S, Pandey SK, Peyrin T, Sasaki Y, Sim SM, Todo Y (2017) GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer W, Homma N (eds). 19th International Conference on Cryptographic Hardware and Embedded Systems Vol. 10529. pp 321–345. https://doi.org/10.1007/978-3-319-66787-4_16

Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2015) The SIMON and SPECK families of lightweight block ciphers. In: Proceedings of the 52nd Annual Design Automation Conference. pp 1–6

Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L (2017) Notes on the design and analysis of SIMON and SPECK. IACR Cryptol ePrint Arch 2017:560

Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, Sasaki Y, Sasdrich P, Sim SM (2016) The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw M, Katz J (eds). 36th Annual International Cryptology Conference Vol. 9815. pp 123–153. https://doi.org/10.1007/978-3-662-53008-5_5

Biham E, Biryukov A, Shamir A (1999) Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Stern J (ed). International Conference on the Theory and Application of Cryptographic Techniques Vol. 1592. pp 12–23. https://doi.org/10.1007/3-540-48910-X_2

Blondeau C, Leander G, Nyberg K (2017) Differential-linear cryptanalysis revisited. J Cryptol 30(3):859–888. https://doi.org/10.1007/s00145-016-9237-5

Bogdanov A, Knezevic M, Leander G, Toz D, Varici K, Verbauwhede I (2011) spongent: A lightweight hash function. In: Preneel B, Takagi T (eds). 13th International Workshop on Cryptographic Hardware and Embedded Systems Vol. 6917. pp 312–325. https://doi.org/10.1007/978-3-642-23951-9_21

Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, Seurin Y, Vikkelsoe C (2007) PRESENT: an ultra-lightweight block cipher. In: Paillier P, Verbauwhede I (eds). 9th International Workshop on Cryptographic Hardware and Embedded Systems Vol. 4727. pp 450–466. https://doi.org/10.1007/978-3-540-74735-2_31

Boura C, Naya-Plasencia M, Suder V (2014) Scrutinizing and improving impossible differential attacks: applications to CLEFIA, Camellia, LBlock and Simon. In: Sarkar P, Iwata T (eds). 20th International Conference on the Theory and Application of Cryptology and Information Security Vol. 8873. pp 179–199. https://doi.org/10.1007/978-3-662-45611-8_10

Chen H, Wang X (2016) Improved linear hull attack on round-reduced Simon with dynamic key-guessing techniques. In: Peyrin T (ed). 23rd International Conference on Fast Software Encryption Vol. 9783. pp 428–449. https://doi.org/10.1007/978-3-662-52993-5_22

Chen Z, Wang N, Wang X (2015) Impossible differential cryptanalysis of reduced round SIMON. IACR Cryptol. ePrint Arch 2015:286

Daemen J, Hoffert S, Assche GV, Keer RV (2018) The design of Xoodoo and Xoofff. IACR Trans Symmetric Cryptol 2018(4):1–38. https://doi.org/10.13154/tosc.v2018.i4.1-38

Derbez P, Fouque P (2016) Automatic search of meet-in-the-middle and impossible differential attacks. In: Robshaw M, Katz J (eds). 36th Annual International Cryptology Conference Vol. 9815. pp 157–184. https://doi.org/10.1007/978-3-662-53008-5_6

Grassi L, Rechberger C, Rønjom S (2016) Subspace trail cryptanalysis and its applications to AES. IACR Trans Symmetric Cryptol 2016(2):192–225. https://doi.org/10.13154/tosc.v2016.i2.192-225

Guo J, Peyrin T, Poschmann A (2011) The PHOTON family of lightweight hash functions. In: Rogaway P (ed). 31st Annual Cryptology Conference Vol. 6841. pp 222–239. https://doi.org/10.1007/978-3-642-22792-9_13

Knudsen LR (1994) Truncated and higher order differentials. In: Preneel B (ed). Second International Workshop on Fast Software Encryption Vol. 1008. pp 196–211. https://doi.org/10.1007/3-540-60590-8_16

Knudsen L (1998) DEAL-A 128-bit block cipher. Complexity 258(2):216

Kölbl S, Leander G, Tiessen T (2015) Observations on the SIMON block cipher family. In: Gennaro R, Robshaw M (eds). 35th Annual Cryptology Conference Vol. 9215. pp 161–185. https://doi.org/10.1007/978-3-662-47989-6_8

Kölbl S, Roy A (2015) A brief comparison of simon and simeck. IACR Cryptol ePrint Arch 2015:706

Kondo K, Sasaki Y, Iwata T (2016) On the design rationale of SIMON block cipher: Integral attacks and impossible differential attacks against SIMON variants. In: Manulis M, Sadeghi A, Schneider SA (eds). 14th International Conference on Applied Cryptography and Network Security Vol. 9696. pp 518–536. https://doi.org/10.1007/978-3-319-39555-5_28

Leander G, Abdelraheem MA, AlKhzaimi H, Zenner E (2011) A cryptanalysis of printcipher: The invariant subspace attack. In: Rogaway P (ed). 31st Annual Cryptology Conference Vol. 6841. pp 206–221. https://doi.org/10.1007/978-3-642-22792-9_12

Leander G, Tezcan C, Wiemer F (2018) Searching for subspace trails and truncated differentials. IACR Trans Symmetric Cryptol 2018(1):74–100. https://doi.org/10.13154/tosc.v2018.i1.74-100

Liu Z, Li Y, Wang M (2017) Optimal differential trails in SIMON-like ciphers. IACR Trans Symmetric Cryptol 2017(1):358–379. https://doi.org/10.13154/tosc.v2017.i1.358-379

Qiao K, Hu L, Sun S (2016) Differential analysis on SIMECK and SIMON with dynamic key-guessing techniques. In: Camp O, Furnell S, Mori P (eds). Second International Conference Information Systems Security and Privacy Vol. 691. pp 64–85. https://doi.org/10.1007/978-3-319-54433-5_5

Sadeghi S, Bagheri N (2018) Improved zero-correlation and impossible differential cryptanalysis of reduced-round SIMECK block cipher. IET Inf Secur 12(4):314–325. https://doi.org/10.1049/iet-ifs.2016.0590

Sasaki Y, Todo Y (2017) New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers. In: Coron J, Nielsen JB (eds). 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques Vol. 10212. pp 185–215. https://doi.org/10.1007/978-3-319-56617-7_7

Shi D, Hu L, Sun S, Song L, Qiao K, Ma X (2014) Improved linear (hull) cryptanalysis of round-reduced versions of SIMON. IACR Cryptol ePrint Arch 2014:973

Sun S, Hu L, Wang P, Qiao K, Ma X, Song L (2014) Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers. In: Sarkar P, Iwata T (eds). 20th International Conference on the Theory and Application of Cryptology and Information Security Vol. 8873. pp 158–178. https://doi.org/10.1007/978-3-662-45611-8_9

Sun S, Hu L, Wang M, Wang P, Qiao K, Ma X, Shi D, Song L (2014) Automatic enumeration of (related-key) differential and linear characteristics with predefined properties and its applications. IACR Cryptol ePrint Arch 2014:747

Todo Y, Morii M (2016) Bit-based division property and application to Simon family. In: Peyrin T (ed). 23rd International Conference on Fast Software Encryption Vol. 9783. pp 357–377. https://doi.org/10.1007/978-3-662-52993-5_18

Wang Q, Liu Z, Varici K, Sasaki Y, Rijmen V, Todo Y (2014) Cryptanalysis of reduced-round SIMON32 and SIMON48. In: Meier W, Mukhopadhyay D (eds). 15th International Conference on Cryptology in India Vol. 8885. pp 143–160. https://doi.org/10.1007/978-3-319-13039-2_9

Wang X, Wu B, Hou L, Lin D (2018) Automatic search for related-key differential trails in SIMON-like block ciphers based on MILP. In: Chen L, Manulis M, Schneider SA (eds). 21st International Conference on Information Security Vol. 11060. pp 116–131. https://doi.org/10.1007/978-3-319-99136-8_7

Xiang Z, Zhang W, Bao Z, Lin D (2016) Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Cheon JH, Takagi T (eds). 22nd International Conference on the Theory and Application of Cryptology and Information Security Vol. 10031. pp 648–678. https://doi.org/10.1007/978-3-662-53887-6_24

Yang G, Zhu B, Suder V, Aagaard MD, Gong G (2015) The Simeck family of lightweight block ciphers. In: Güneysu T, Handschuh H (eds). 17th International Workshop on Cryptographic Hardware and Embedded Systems Vol. 9293. pp 307–329. https://doi.org/10.1007/978-3-662-48324-4_16

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.