New Methods for Bounding the Length of Impossible Differentials of SPN Block Ciphers

Senpeng Wang, Dengguo Feng, Tairong Shi, Bin Hu, Jie Guan, Kai Zhang, Ting Cui

***Abstract*—How to evaluate the security of Substitution-Permutation Network (SPN) block ciphers against impossible differential (ID) cryptanalysis is a valuable problem. In this paper, a series of methods for bounding the length of IDs of SPN block ciphers are proposed. Firstly, we propose the definitions of minimal representative set and partition table. Therefore, an improved partition-first implementation strategy for bounding the length of IDs is given. Secondly, we introduce a new definition of ladder and propose the ladder-first implementation strategy for bounding the length of IDs. In order to be able to apply ladder-first implementation strategy in practice, the methods for determining ladders and integrating a ladder into searching models are given. Thirdly, a heuristic algorithm called dynamic-ladder-partition implementation strategy is proposed. According to our experimental results, dynamic-ladder-partition implementation strategy is more suitable for SPN ciphers whose number of elements in partition tables is little. Fourthly, rotation-equivalence ID sets of ciphers are explored to reduce the number of models that need to be considered. As applications, we show that 9-round PRESENT, 5-round AES, 6-round Rijndael-160, 7-round Rijndael-192, 7-round Rijndael-224 and 7-round Rijndael-256 do not have any ID under the sole assumption that the round keys are uniformly random. What's more, we obtain that 8-round GIFT-64, 12-round GIFT-128 and 14-round SKINNY-128 do not have any ID under the assumptions that GIFT and SKINNY are Markov ciphers and the round keys are uniformly random. Our methods fill crucial gaps on bounding the length of IDs with the differential properties of S-boxes considered. They enhance our confidence in the security and are valuable, especially for designers.**

***Index Terms*—Impossible differential, PRESENT, GIFT, SKINNY, Rijndael, AES.**

I. Introduction

Impossible differential (ID) cryptanalysis [1], [2] is one of the most effective cryptanalytic approaches for block ciphers. The main idea of it is to utilize IDs (differentials with probability 0) to discard wrong keys. So far, ID cryptanalysis has been used to attack lots of block ciphers, such as AES [3].

For attackers, finding ID distinguishers plays an important role in ID attacks. In [4], Kim *et al.* proposed the first automatic method for finding IDs, called $\mathcal{U}$-method. After that, many improved automatic tools are presented, such as UID-method [5], $\mathcal{WW}$-method [6], $\mathcal{U}^{\star}$-method [7], *etc.* At EUROCRYPT 2023, Hadipour *et al.* [8] unified the stage of searching ID distinguishers and the stage of key recovery to search for better ID attacks. However, all these tools treat S-boxes as ideal ones that any nonzero input difference could produce every nonzero output difference. Thus, the IDs obtained by these methods may not be the longest for real ciphers. In order to tackle this problem, Cui *et al.* [9] and Sasaki and Todo [10] independently proposed automatic tools based on Mixed Integer Linear Programming (MILP) to search IDs for block ciphers with differential details of S-box considered. With the tools based on MILP, they could identify whether a specific differential was ID. In theory, the tools based on MILP can find all IDs under the assumption that round keys are uniformly random. However, for a block cipher with n-bit block size, the number of differentials in the whole search space is about 2^{2n} which is not affordable to determine all these differentials one by one.

For designers, it is important to evaluate the security of block ciphers. To prove the security of a block cipher against ID attacks, a common way is to give an upper bound on the rounds of IDs. In [11], Cui *et al.* suggested that the differential pattern matrix of P-layer could be used to deduce all IDs for SPN block ciphers. At EUROCRYPT 2016, Sun *et al.* [12] associated a primitive index with the characteristic matrix of the linear layer. They proved that the length of IDs for some special SPN block ciphers was bounded by the primitive index of the linear layer. In order to obtain the bounds of IDs in practical time, they proved that the length of ID bound depended only on the low-weight IDs under special conditions. To overcome the limitations of the above methods, Wang and Jin [13] used linear algebra to propose a practical method that could give the upper bound on the length of IDs for any SPN block cipher when treating S-boxes as ideal ones. Since the above methods do not consider the differential details of S-box, their bounds may become invalid.

When the details of S-box are considered, the security bounds of ciphers against ID will be more convincing. The difficulty of this problem is that it needs to prove that all differentials are possible when the round number of a block cipher is not less than a certain integer. If there is no special explanation, all the contents of ID considering the details of the S-box in this paper are obtained under the assumption that round keys are uniformly random. The research progress in this field can be divided into the following three categories.

Rigorous mathematical derivation. By revealing some important properties of the S-box and linear layer used in AES, Wang and Jin [14] prove that even though the details of the S-box are considered, there do not exist ID covering more than 4 rounds for AES. However, this method is only

This work was partially supported by the National Natural Science Foundation of China under Grant No. 62102448, 62202493.

Senpeng Wang,Dengguo Feng are with the State Key Laboratory of Cryptology, Beijing, China.

Senpeng Wang, Tairong Shi, Bin Hu, Jie Guan ,Kai Zhang, Ting Cui are with Information Engineering University, Zhengzhou, China

(Corresponding author: Senpeng Wang, e-mail: wsp2110@126.com)

(Corresponding author: Tairong Shi, e-mail: tairong2018@iscas.ac.cn)

applicable to AES at present.

Bounds on partial search space. The automatic search methods based on solvers [9], [10], [15] can determine whether a concrete differential is ID. Thus, the bound on partial search space of differentials can be obtained.

Bounds on whole search space for special SPN ciphers. At SAC 2022, Hu *et al.* [16] partitioned the whole search space of difference pairs into lots of small disjoint sets. When the number of sets is reduced to a reasonable size, they can detect whether there exist ID with MILP models. Due to the limitation of huge time complexity, their method currently works only for special SPN ciphers with 64-bit block size.

A. Our Contributions

In this paper, we propose a series of methods for bounding the length of IDs of SPN block ciphers. The contributions can be classified into the following five parts.

Improved partition-first implementation strategy. The choices of representative sets and partition tables will have an important influence on the number of models that need to be solved. In order to reduce the implementation complexity, we put forward the definitions of minimal representative set and partition table and give the automatic methods for determining representative sets and partition tables of S-boxes with different sizes. Therefore, an improved partition-first implementation strategy for bounding the length of IDs is given. Compared with the partition-first implementation strategy in [16], our improved partition-first implementation strategy can use fewer or even the least number of models to obtain the ID bound.

Ladder-first implementation strategy. Based on our new definition about a special set of difference pairs, called *ladder*, we propose ladder-first implementation strategy for bounding the length of IDs. This strategy constructs a ladder for a middle component of a cipher. Thus, the sets of input and output differences can be considered separately. Then, the methods for determining ladders and integrating a ladder into searching models are put forward. It should be noted that ladder-first implementation strategy shifts the complexity calculation from multiplication to addition, which has huge impact on the improvement of efficiency. Moreover, efficiency and accuracy of improved partition-first implementation strategy and ladder-first implementation strategy are compared and analyzed.

Dynamic-ladder-partition implementation strategy. This strategy will determine ladders and partition tables dynamically. We give a heuristic algorithm for implementing dynamic-ladder-partition strategy. According to our experimental results, dynamic-ladder-partition implementation strategy is more suitable for SPN ciphers whose number of elements in partition tables is little.

Rotation-equivalence ID set. In order to reduce the number of difference pairs that need to be considered, we propose the definition of rotation-equivalence ID set. The difference pairs in the same rotation-equivalence ID set will have the same results on whether there are IDs or not. Therefore, only one difference pair for each rotation-equivalence ID set needs to be considered. In this way, we can bound the length of IDs of SPN block ciphers more effectively.

Applications to SPN block ciphers. Under the sole assumption that round keys are uniformly random, we show that 9-round PRESENT, 5-round AES, 6-round Rijndael-160, 7-round Rijndael-192, 7-round Rijndael-224 and 7-round Rijndael-256 do not have any ID. What's more, we obtain that 8-round GIFT-64, 12-round GIFT-128, 14-round SKINNY-128 do not have any ID under the assumptions that GIFT and SKINNY are Markov ciphers and the round keys are uniformly random. The results of PRESENT, GIFT-128, SKINNY-128, Rijndael-160, Rijndael-192, Rijndael-224 and Rijndael-256 are obtained for the first time. Moreover, the ID bounds of AES, Rijndael-160, Rijndael-192, Rijndael-224 and Rijndael-256 are tight. All the application results are shown in Table I.

Providing a more accurate security assessment is an important issue in the design and analysis of ciphers. The most intriguing aspect of this paper is its ability to bound the ID length for large-scale SPN ciphers with the details of S-boxes considered. This is something that couldn't be achieved before. Compared with the methods in [16], our methods have two advantages. On one hand, our methods are more efficient. For example, when determining whether there is ID for GIFT-128, the methods in [16] need to solve 2^{52} models, while our methods only need to solve $2^{25.83}$ models. On the other hand, our methods are more general which are no longer limited to special SPN ciphers with 64-bit block size. Our methods can be applied to SPN ciphers with large block size. For instance, the ID bound of SKINNY-128 is obtained for the first time. These methods can be particularly valuable, especially from a designer's standpoint.

TABLE I
THE ID RESULTS OF SOME SPN BLOCK CIPHERS

Cipher	Block size	Longest known ID	Number of models	Bound	Reference
PRESENT	64	6 [17]	-	$7^{\star}$	[17]
			$2^{24.68}$	9	Sect. VII-A
GIFT-64	64	6 [17]	-	$7^{\star}$	[18]
			2^{26}	8	[16]
			$2^{24.68}$	8	Sect. VII-B
GIFT-128	128	7 [16]	$2^{12.17}$	8^{*}	[16]
			2^{52}	$-^{\dagger}$	[16]
			$2^{25.83}$	12	Sect. VII-B
SKINNY-128	128	12.5 [7]	-	$13.5^{\ddagger}$	[7]
			$2^{26.49}$	14	Sect. VII-B
AES (Rijndael-128)	128	4 [3]	-	5	[14]
			$75 + \mathcal{O}\left(2^{32}\right)^{\blacklozenge}$	5	Sect. VIII-A
Rijndael-160	160	5 [19]	217	6	Sect. VIII-A
Rijndael-192	192	6 [20]	-	$7^{\ddagger}$	[16]
			819	7	Sect. VIII-A
Rijndael-224	224	6 [20]	2413	7	Sect. VIII-A
Rijndael-256	256	6 [19]	8925	7	Sect. VIII-A

* The security bound of the search space where there is only one active S-box for both the input and output differences.

* The security bound of the search space where there is only one active superbox for both the input and output differences.

† Because the number of models is too huge, the ID bound considering the details of S-box of GIFT128 cannot be obtained.

‡ The security bound of ID omitting the details of S-box.

$^{\blacklozenge}$ We need to verify some representatives of 32-bit superboxes in AES.

B. Outline

This paper is organized as follows: Sect. II introduces the notations, definitions and related works. In Sect. III, IV and V, we propose improve partition-first implementation strategy, ladder-first implementation strategy and dynamic-ladder-partition implementation strategy, respectively. Sect. VI proposes the rotation-equivalence ID set to further reduce the number of models that need to be solved. In Sect. VII and VIII, we apply our methods to two types of SPN block ciphers. In Sect. IX, we conclude the paper.

II. Preliminaries

A. Notations and Definitions

Some notations used in this paper are defined in Table II.

TABLE II
SOME NOTATIONS USED IN THIS PAPER

$\mathbb{F}_2$	The finite field $\{0,1\}$
$x \in \mathbb{F}_2^n$	An n-bit vector or difference
$x \oplus y$	Bitwise XOR of x and y
$x \lll i$	Left rotation of x by i-bit position
$x \ggg i$	Right rotation of x by i-bit position
$x\|\|y$	The concatenation of x and y
$x^{n\|\|}$	The concatenation $x\|\|x\|\|\cdots\|\|x$ whose number of x is n
$\emptyset$	Empty set
A	Set is denoted as uppercase letter such as A
$\|A\|$	The number of elements in the set A
$A \cap B$	The intersection of two sets A and B
$A \cup B$	The union of two sets A and B
$A + B$	If $A \cap B = \emptyset$, we denote the union of A and B as $A + B$
$A - B$	The set $\{a \mid a \in A \text{ and } a \notin B\}$
$A \times B$	The cartesian product $\{(a,b) \mid a \in A, b \in B\}$ of sets A and B
A^n	The set $A \times A \times \cdots \times A$ whose number of A is n

Definition 1. (Expected Differential Probability [21]). *Let $f : \mathbb{F}_2^\kappa \times \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ be a keyed vectorial boolean function with κ-bit key size. Then, the expected probability of differential $(a,b) \in \mathbb{F}_2^n \times \mathbb{F}_2^m$ over f is defined as:*

$$EDP(a \xmapsto{f} b) = 2^{-\kappa} \sum_{k \in \mathbb{F}_2^\kappa} DP(a \xmapsto{f_k} b),$$

where $DP(a \xmapsto{f_k} b) = 2^{-n} \times |\{x \in \mathbb{F}_2^n | f(k,x) \oplus f(k, x \oplus a) = b\}|$ is the differential probability of (a,b) over $f(k,x)$.

If $EDP(a \xmapsto{f} b) = 0$ holds, the differential (a,b) is an ID over f, denoted as $a \overset{f}{\nrightarrow} b$. Otherwise, if $EDP(a \xmapsto{f} b) > 0$ holds, the differential (a,b) is a possible differential pattern, denoted as $a \overset{f}{\rightarrow} b$. For two sets of differences A and B, if $a \overset{f}{\rightarrow} b$ holds for all $(a,b) \in A \times B$, we denote it as $A \overset{f}{\rightarrow} B$. Otherwise we denote it as $A \overset{f}{\nrightarrow} B$. Moreover, $a \overset{f}{\rightarrow} B$ and $A \overset{f}{\rightarrow} b$ are equivalent to $\{a\} \overset{f}{\rightarrow} B$ and $A \overset{f}{\rightarrow} \{b\}$, respectively.

In this paper, we are only interested in the bit-wise XOR difference. On this condition, we introduce the following definition and theorem.

Definition 2. (Markov Cipher [22]). *Let $f(x \oplus k)$ be the round function of an iterated cipher, where k is the round key. For all choices of a and b $(a \neq 0, b \neq 0)$, if the round key is uniformly random, the value of probability*

$$P\left(f\left(x \oplus k\right) \oplus f\left(x' \oplus k\right) = b | x \oplus x' = a, x = c\right)$$

is the same for any c. This cipher is called a Markov cipher.

Theorem 1. (EDP of Markov Cipher [22]). *Let $E = f_{r-1} \circ f_{r-2} \circ \cdots \circ f_0$ be an r-round Markov cipher. Under the assumption that round keys are uniformly random, the EDP of (a_0, a_r) over E can be calculated as*

$$\begin{aligned} &EDP(a_0 \xmapsto{E} a_r) \\ &= \sum_{a_1} \sum_{a_2} \cdots \sum_{a_{r-1}} EDP(a_0 \xmapsto{f_0} a_1 \xmapsto{f_1} \cdots \xmapsto{f_{r-1}} a_r), \end{aligned} \quad (1)$$

where $EDP(a_0 \xmapsto{f_0} a_1 \xmapsto{f_1} \cdots \xmapsto{f_{r-1}} a_r) = \prod_{i=0}^{r-1} EDP(a_i \xmapsto{f_i} a_{i+1})$ is the EDP of the r-round differential trail $a_0 \longmapsto a_1 \longmapsto \cdots \longmapsto a_r$ over E.

According to Eq. (1), for an r-round Markov cipher E, if we want to prove $a_0 \overset{E}{\rightarrow} a_r$, we need to find an r-round differential trail satisfying $EDP(a_0 \xmapsto{f_0} a_1 \xmapsto{f_1} \cdots \xmapsto{f_{r-1}} a_r) > 0$. If we want to prove that there does not exist any ID for cipher E, we have to prove that $a_0 \overset{E}{\rightarrow} a_r$ holds for every concrete differential (a_0, a_r). As far as we know, most SPN block ciphers whose round keys are added to the full state (such as AES [23]) are Markov ciphers. For those SPN ciphers that are not Markov ciphers (such as SKINNY [24] and GIFT [18]), when we use the result of Theorem 1, we need to assume that those ciphers are Markov ciphers.

B. Current Automatic Methods for Finding IDs

In [25], [26], MILP based methods for searching differential distinguishers were proposed. By adding additional constraints on the input and output differences, Cui *et al.* [9] and Sasaki and Todo [10] independently proposed MILP models to search IDs for block ciphers with the details of S-boxes considered. Using MILP tools, they are able to identify whether a differential is ID or not. However, when we want to find all the IDs or to know whether there exist longer IDs for a block cipher, we have to solve about 2^{2n} models for a cipher with n-bit block size to check all input and output difference pairs. The search space far exceeds existing computing power.

In order to tackle this problem, Hu *et al.* [16]) partition the whole search space into many small disjoint sets and then exclude the sets containing no IDs. Thus, when their methods have determined that all differentials are not IDs, the provable security of ciphers against ID can be obtained. We will introduce their methods from the perspective of bounding the length of IDs which is also the main topic of this paper.

Definition 3. (Representative Set [16]). *For a function f, let A and B be the sets of input and output differences, respectively. If the following condition is satisfied,*

$$\forall a \in A, \exists b \in B \text{ satisfying } a \overset{f}{\rightarrow} b$$

we call B a representative set of A over f, denoted as $A \overset{f}{\rightarrow} \exists B$.

Definition 4. (Partition Table [16]). *For a function f , let A and B be the sets of input and output differences, respectively. For any b, we can construct a set $H[b]$ satisfying $H[b] \subseteq \{a \in A | a \xrightarrow{f} b\}$. If the following conditions are satisfied*

$$\begin{cases} H[b_1] \bigcap H[b_2] = \emptyset, \textit{ for any } b_1, b_2 \in B \textit{ satisfying } b_1 \neq b_2, \\ \bigcup\limits_{b \in B} H[b] = A, \end{cases}$$

we call $H[b], b \in B$ a partition table of A over f, denoted as $PT[A, B, H, f]$.

By dividing a large-dimension function into small parts, Hu *et al.* [16] propose a solution for obtaining a representative set and partition table of S-box layer.

Theorem 2. ([16]). *For a function S comprising of m parallel S-boxes, denoted as $S = s_{m-1} || \cdots || s_1 || s_0$, let $A = A_{m-1} \times \cdots \times A_1 \times A_0$ be the input difference set of S, where A_i is the input difference set of $s_i, i \in \{0, 1, \ldots, m-1\}$. If we obtain the partition tables $PT\left(A_i, B_i, H_i, s_i\right), i \in \{0, 1, \ldots, m-1\}$, then*

$$A = \sum_{b_{m-1} \in B_{m-1}} \cdots \sum_{b_0 \in B_0} H_{m-1}[b_{m-1}] \times \cdots \times H_1[b_1] \times H_0[b_0].$$

Thus, we obtain the partition table of A over S.

Then, Hu *et al.* [16] proposed a framework for bounding the length of IDs as showed in the following theorem (also illustrated in Fig. 1)

Theorem 3. (Bounding the Length of IDs [16]). *For a cipher $E = E_2 \circ E_1 \circ E_0$ and partition tables $PT[A_0, A_1, H_0, E_0]$ and $PT[A_3, A_2, H_2, E_2^{-1}]$, the set $A_0 \times A_3$ is the union of smaller sets as follows,*

$$A_0 \times A_3 = \bigcup_{a_1 \in A_1, a_2 \in A_2} H_0[a_1] \times H_2[a_2].$$

For each element $(a_1, a_2) \in A_1 \times A_2$, the model is built to detect whether $a_1 \xrightarrow{E_1} a_2$. If $A_1 \xrightarrow{E_1} A_2$ is satisfied, the cipher E has no ID over $A_0 \times A_3$. Thus, the ID bound of E can be obtained. Otherwise, if there exists $a_1 \overset{E_1}{\nrightarrow} a_2$, the set of difference pairs $H_0[a_1] \times H_2[a_2]$ may contain some IDs.

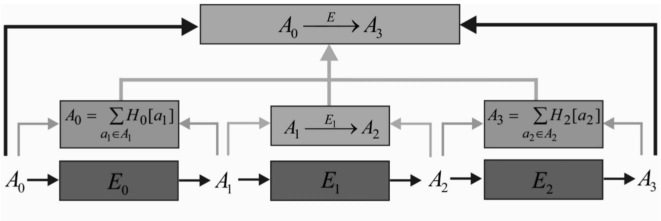

$A_0 \xrightarrow{E} A_3$
$A_0 = \sum_{a_1 \in A_1} H_0[a_1]$
$A_1 \xrightarrow{E_1} A_2$
$A_3 = \sum_{a_2 \in A_2} H_2[a_2]$
A_0
E_0
A_1
E_1
A_2
E_2
A_3

Fig. 1. The framework for bounding the length of IDs in [16]

Hu *et al.* proposed an heuristic algorithm to determine the partition table and applied Theorem 3 to some ciphers directly. We call this implementation strategy as partition-first-implementation strategy. This strategy considers the input difference set and output difference set together. In order to get the ID bound of E, at least $|A_1| \times |A_2|$ models need to be solved. Thus, the set $A_1 \times A_2$ will have an important influence on the number of models that need to be solved. When applying partition-first implementation strategy to ciphers, the number of models may not be affordable. Thus, Hu *et al.*'s methods work well only for special SPN ciphers with 64-bit block size. Following the work of [16], we proposed improved partition-fist implementation strategy in Sect. III, ladder-first implementation strategy in Sect. IV and dynamic-ladder-partition implementation strategy in Sect. V.

To facilitate the description of the strategies for bounding the length of IDs, we introduce an indicator variable $flag$ to denote the results of ID as following:

$$flag = \begin{cases} 0, & \text{if there is no ID for target cipher;} \\ 1, & \text{if cipher has at least one ID;} \\ 2, & \text{if cannot determine whether there is ID.} \end{cases}$$

When we cannot get the value of $flag$ due to the limited storage and computing capacity, we set $flag = 2$.

III. Improved Partition-First Implementation Strategy

In the paper [16], Hu *et al.* proposed an intuitive algorithm which can generated representative sets and partition tables. Just as they wrote in the paper, their algorithm was not very efficient. The choices of representative sets and partition tables will have an important influence on the number of models that need to be solved. We propose the definitions of minimal representative sets and partition tables as following.

Definition 5. (Minimal Representative Set and Partition Table). *For an S-box S, let A be the set of input differences. For a partition table $PT[A, B, H, S]$, if the size of B is minimal among all possible partition tables, we call B a minimal representative set and $PT[A, B, H, S]$ a minimal partition table of A over S.*

To help readers better understand the significance of Definition 5, we take Theorem 3 for example. The number of models that need to be solved in Theorem 3 is $|A_1| \times |A_2|$. If $PT[A_0, A_1, H_0, E_0]$ and $PT[A_3, A_2, H_2, E_2^{-1}]$ are minimal partition tables, the number of models that need to be solved will be minimum. For S-boxes of different sizes, we propose corresponding methods for determining their representative sets and partition tables as following.

A. Determining a Minimal Representative Set and Partition Table of Small-Size S-box

When the size of an S-box is small (e.g. 4-bit or 8-bit S-box), inspired by the method in [27] for modeling the differential trails of S-box, we propose an automatic method based on MILP to obtain its minimal representative set and partition table. For an S-box S, let A and B be the input and output difference sets, respectively. The overview of our algorithm is as following. Firstly, for each input difference $a \in A$, we compute the set of output differences, denoted as $R[a] = \{b \in B | a \xrightarrow{S} b\}$. Secondly, for each $a \in A$, we construct a constraint such that there must be at least 1 element of $R[a]$ belongs to representative set. Finally, we minimize the number of elements in representative set under these constraints.

Constraints. For each $b \in B$, we introduce a binary variable v_b, where $v_b = 1$ means that the output difference b is included in representative set and $v_b = 0$ means that b is not included in representative set. The constraints should ensure that each $a \in A$ has at least one representative, which can be represented by the following $|A|$ constraints

$$\sum_{b \in R[a]} v_b \geq 1, a \in A.$$

Objective Function. Our goal is to find a minimal representative set. Thus, the objective function can be expressed as

$$\text{minimize} \sum_{b \in B} v_b.$$

By solving the above MILP model, we can obtain the solutions of $v_b, b \in B$. Thus, a minimal representative set is $B' = \{b \in B | v_b = 1\}$. The whole procedure for obtaining a minimal representative set of S is demonstrated in Algorithm 1.

Algorithm 1 Obtaining representative set of small-size S-box

```
Require: The S-box $S$, input and output difference sets $A$ and $B$
Ensure: A minimal representative set $B'$ of $A$ over $S$
Let $\mathcal{M}$ be an empty MILP model
$\mathcal{M}.Objective =$ minimize $\sum_{b \in B} v_b$    ▷ Set the objective function
for $a \in A$ do
    $\mathcal{M}.addConstr\left(\sum_{b \in R[a]} v_b \geq 1\right)$
end for
$\mathcal{M}.optimize()$    ▷ Solve the MILP model
return $B' = \{b \in B | v_b = 1\}$    ▷ Obtain a minimal representative set
```

According to Definition 4 and Definition 5, by removing the overlapping elements among sets $\{a \in A | a \xrightarrow{S} b'\}, b' \in B'$, we can get a minimal partition table $PT[A, B', H, S]$ of A over S.

B. Determining a Minimal Representative Set and Partition Table of Middle-Size S-box

When we use the method in Sect. III-A to determine a minimal representative set and partition table of middle-size S-box (e.g. 16-bit S-box), the MILP model are too large to be solved. Thus, we propose a method to solve this problem.

Theorem 4. *For an S-box S, let A and B be the input and output difference sets, respectively. Selecting a subset $A' \subseteq A$, let B' be a minimal representative set of A'. If B' is a representative set of A, then B' is a minimal representative set of A.*

Proof. Let B'' be a minimal representative set of A. Since $A' \subseteq A$, B'' is also a representative set of A'. Because B' is a minimal representative set of A', we have $|B'| \leq |B''|$. When B' is a representative set of A, according to the definition of minimal representative set, B' must be a minimal representative set of A. □

For the small subset $A' \subseteq A$, we can use Algorithm 1 to obtain a minimal representative set B' of A'. If B' is a representative of A, then we obtain a minimal representative set of A. If B' is not a representative of A, we add the elements which cannot be represented by B' into A'. That is, $A' = A' + \{a \in A | a \overset{S}{\nrightarrow} B'\}$. We will keep adding elements into A' until the corresponding B' is a minimal representative set of A. The whole procedure for obtaining a minimal representative set of A over S is demonstrated in Algorithm 2. This algorithm is use to obtain the minimal representative sets of PRESENT and GIFT-64.

According to Definition 4 and Definition 5, by removing the overlapping elements among sets $\{a \in A | a \xrightarrow{S} b'\}, b' \in B'$, we can get a minimal partition table $PT[A, B', H, S]$ of A over S.

Algorithm 2 Obtaining representative set of middle-size S-box

```
Require: The S-box $S : \mathbb{F}_2^n \to \mathbb{F}_2^n$, input and output difference sets $A$ and $B$
Ensure: A minimal representative set $B'$
Select a subset $A' \subseteq A$ and let $B' = \emptyset$
while $B'$ is not a representative set of $A$ do
    Using Algorithm 1 to obtain a minimal representative set $B'$ of $A'$
    if $B'$ is a representative of $A$ then
        return $B'$
    else
        $A' = A' + \{a \in A | a \overset{S}{\nrightarrow} B'\}$
    end if
end while
```

C. Determining a Representative Set and Partition Table of Large-Size Superbox

Previous methods in [16] cannot be applied into large-size S-box (e.g. 32-bit S-box). Most S-boxes of large size are superboxes illustrated in Fig 2, where $s_i, 0 \leq i \leq m-1$ are bijective small-size S-boxes and P is a bijective linear function. In order to construct a representative set with relatively few elements, we propose the following theorem.

Theorem 5. *For an S-box $S = (s_{m-1}||s_{m-2}|| \cdots ||s_0) \circ P \circ (s_{m-1}||s_{m-2}|| \cdots ||s_0)$, let $A = A_{m-1} \times A_{m-2} \times \cdots \times A_0$ and $B = B_{m-1} \times B_{m-2} \times \cdots \times B_0$ be the input and output difference sets, respectively. For each $0 \leq i \leq m-1$, let B_i' be a minimal representative set of A_i over s_i and $B_i'' \subseteq B_i$ be a representative of all possible differences $\{a | a \in \mathbb{F}_2^n\}$ over s_i, where n is the dimension of s_i. Then, we can use Algorithm 1 to obtain a representative set $C \subseteq B_{m-1}'' \times B_{m-2}'' \times \cdots \times B_0''$ of $B_{m-1}' \times B_{m-2}' \times \cdots \times B_0'$ over $(s_{m-1}||s_{m-2}|| \cdots ||s_0) \circ P$. Thus, C is a representative set of A.*

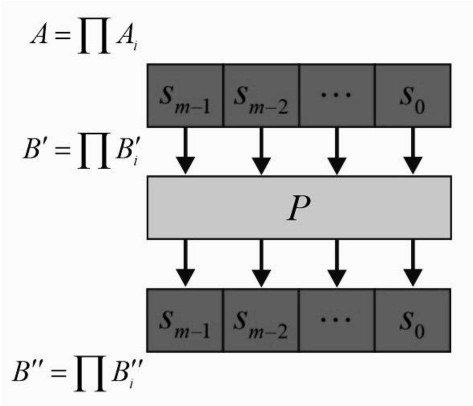


Fig. 2. Large-size superbox

Proof. Because $B''_{m-1}\times B''_{m-2}\times\cdots\times B''_0$ is a representative set of $\{a|a\in\mathbb{F}_2^{n\times m}\}$ over $(s_{m-1}||s_{m-2}||\cdots||s_0)$ and $B'_{m-1}\times B'_{m-2}\times\cdots\times B'_0 \xrightarrow{P} \exists\{a|a\in\mathbb{F}_2^{n\times m}\}$, we have $B''_{m-1}\times B''_{m-2}\times\cdots\times B''_0$ is a representative set of $B'_{m-1}\times B'_{m-2}\times\cdots\times B'_0$ over $(s_{m-1}||s_{m-2}||\cdots||s_0)\circ P$. Thus, we must be able to select a representative set $C\subseteq B''_{m-1}\times B''_{m-2}\times\cdots\times B''_0$ of $B'_{m-1}\times B'_{m-2}\times\cdots\times B'_0$ over $(s_{m-1}||s_{m-2}||\cdots||s_0)\circ P$. Because $B'_{m-1}\times B'_{m-2}\times\cdots\times B'_0$ is a representative set of $A_{m-1}\times A_{m-2}\times\cdots\times A_0$ over $(s_{m-1}||s_{m-2}||\cdots||s_0)$, C is a representative set of A over S. □

The representative set C obtained by Theorem 5 may contain redundant elements, we need to reduce C further. The whole procedure of obtaining a representative set of large-size superbox S is demonstrated in Algorithm 3. If necessary, we can repeat line 7-12 in Algorithm 3 multiple times to get a smaller representative set.

According to Definition 4 and Definition 5, by removing the overlapping elements among sets $\{a\in A|a\xrightarrow{S}c'\}, c'\in C'$, we can get the partition table $PT[A,C',H,S]$ of A over S.

D. Improved Partition-First Implementation Strategy for Bounding the Length of IDs

The differences between the strategy in [16] and our improved partition-first implementation strategy are mainly reflected in two aspects. On one hand, we propose the definitions of minimal representative set and minimal partition table and give new methods for determining representative sets and partition tables of S-boxes. Thus, our methods can solve fewer or even the least models to obtain the ID bound. On the other hand, when there are some uncertain IDs, we adopt a different enhance stage.

For a cipher $E=E_2\circ E_1\circ E_0$, we construct partition tables $PT[A_0,A_1,H_0,E_0]$ and $PT[A_3,A_2,H_2,E_2^{-1}]$, where A_0 and A_3 are the input and output difference sets of E, respectively. In the fundamental stage, if $A_1\xrightarrow{E_1}A_2$ is satisfied, according to Theorem 3, there is no ID for E over $A_0\times A_3$. If $A_1\xrightarrow{E_1}A_2$ is not satisfied, we obtain a set $I=\{(a_1,a_2)\in A_1\times A_2|a_1\overset{E_1}{\nrightarrow}a_2\}$. And we need to further determine whether $H_0[a_1]\overset{E}{\nrightarrow}H_2[a_2],(a_1,a_2)\in I$. In the enhance stage, we construct a set $I_1=\{a_1\in A_1|\,(a_1,a_2)\notin I$ holds for every $a_2\in A_2\}$. For any $a_1\in I_1$, we have $a_1\xrightarrow{E_1}A_2$. Thus, $\sum_{a_1\in I_1}H_0[a_1]\xrightarrow{E}A_3$. Therefore, for any $a_1\in A_1$, we can reduce the hash table $H_0[a_1]$ to $H'_0[a_1]=H_0[a_1]-\sum_{a\in I_1}H_0[a]$. Similarly, for any $a_2\in A_2$, we can obtain the reduced hash table $H'_2[a_2]$. Then, for any $(a_1,a_2)\in I$, we further explore whether $H'_0[a_1]\xrightarrow{E}H'_2[a_2]$ is satisfied. The whole procedure for obtaining the ID result of E over $A_0\times A_3$ is demonstrated in Algorithm 4.

Algorithm 3 Obtaining a representative set of superbox

```
Require: The S-box $S=(s_{m-1}||s_{m-2}||\cdots||s_0)\circ P\circ(s_{m-1}||s_{m-2}||\cdots||s_0)$, input and output difference sets $A=A_{m-1}\times A_{m-2}\times\cdots\times A_0$ and $B=B_{m-1}\times B_{m-2}\times\cdots\times B_0$
Ensure: A representative set of $A$ over $S$
1: for $0\le i\le m-1$ do                      ▷ Using Algorithm 1
2:     Obtain a minimal representative set $B'_i$ of $A_i$ over $s_i$
3:     Obtain a minimal representative set $B''_i$ of $\{a|a\in\mathbb{F}_2^n\}$ over $s_i$
4: end for
5: Using Algorithm 1 to obtain a representative set $C\subseteq B''_{m-1}\times B''_{m-2}\times\cdots\times B''_0$ of $B'_{m-1}\times B'_{m-2}\times\cdots\times B'_0$ over $(s_{m-1}||s_{m-2}||\cdots||s_0)\circ P$
6: Allocate $C'=\emptyset$
7: while $A\neq\emptyset$ do
8:     Select an element $a\in A$ and $c\in C$ satisfying $a\xrightarrow{S}c$
9:     $A\leftarrow A-\{a\in A|a\xrightarrow{S}c\}$    ▷ Remove the elements which have been represented
10:    $C'\leftarrow C'+\{c\}$ and $C\leftarrow C-\{c\}$
11: end while
12: return $C'$
```

IV. Ladder-First Implementation Strategy

The number of models that need to be solved in determining the up bound of IDs will greatly limit its applications. In this section, we propose a new implementation strategy which can consider the input difference set and output difference set separately. Thus, we can obtain the ID result by independently searching the input difference set and output difference set. This divide and conquer method will greatly reduce the number of models that need to be solved.

A. Ladder-First Implementation Strategy for Bounding the Length of IDs

First of all, we introduce a new definition as following.

Definition 6. (Ladder) *For a function f, let A and B be sets of input and output differences, respectively. If the condition $A\xrightarrow{f}B$ is satisfied, we call $A\times B$ a ladder of f.*

Lemma 1. *Let f be a bijective function. If $A\times B$ is a ladder of f, then $B\times A$ is also a ladder of f^{-1}, where f^{-1} is the inverse function of f.*

Proof. Because $A\xrightarrow{f}B$, for any $(a,b)\in A\times B$, there exists x satisfying $f(x)\oplus f(x\oplus a)=b$. For the element $y=f(x)$, we have $f^{-1}(y)\oplus f^{-1}(y\oplus b)=x\oplus(x\oplus a)=a$. Thus, for any $(b,a)\in B\times A$, we have $b\xrightarrow{f^{-1}}a$. □

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3473940

Algorithm 4 Improved partition-first implementation strategy

```
Require: The cipher $E = E_2 \circ E_1 \circ E_0$, input and output difference sets $A_0$ and $A_3$
Ensure: $flag$    ▷ Return the ID result of $E$ over $A_0 \times A_3$
———————- Fundamental Stage ———————-
$PT[A_0, A_1, H_0, E_0]$ and $PT[A_3, A_2, H_2, E_2^{-1}]$    ▷ Using the methods in Sect III to obtain partition tables
Allocate $I \leftarrow \emptyset$
for $(a_1, a_2) \in A_1 \times A_2$ do
    if $a_1 \overset{E_1}{\nrightarrow} a_2$ then    ▷ Build a model to determine whether $a_1 \overset{E_1}{\rightarrow} a_2$
        $I \leftarrow I \cup \{(a_1, a_2)\}$
    end if
end for
if $I = \emptyset$ then
    return $flag = 0$    ▷ $E$ has no ID over $A_0 \times A_3$
end if
———————- Enhance Stage ———————-
$I_1 = \{a_1 \in A_1 | (a_1, a_2) \notin I$ holds for every $a_2 \in A_2\}$
$I_2 = \{a_2 \in A_2 | (a_1, a_2) \notin I$ holds for every $a_1 \in A_1\}$
$H'_0[a_1] = H_0[a_1] - \sum_{a \in I_1} H_0[a]$ for any $a_1 \in A_1$
$H'_2[a_2] = H_2[a_2] - \sum_{a \in I_2} H_2[a]$ for any $a_2 \in A_2$
for $(a_1, a_2) \in I$ do
    for $(a_0, a_3) \in H'_0[a_1] \times H'_2[a_2]$ do
        if $a_0 \overset{E}{\nrightarrow} a_3$ then    ▷ Build a model to determine whether $a_0 \overset{E}{\rightarrow} a_3$
            return $flag = 1$    ▷ $E$ has at least one ID
        end if
    end for
end for
return $flag = 0$    ▷ $E$ has no ID over $A_0 \times A_3$
```

Based on the definition of ladder, we propose ladder-first implementation strategy for bounding the length of IDs as shown in the following theorem (also illustrated in Fig. 3)

Theorem 6. *Let $E = E_4 \circ E_3 \circ E_2 \circ E_1 \circ E_0$ be a cipher, where $E_i, 0 \leq i \leq 4$ are all bijective functions. If there exist the sets of differences $A_0, A_1, A_2, A_3, A_4, A_5$ and partition tables $PT[A_0, A_1, H_0, E_0], PT[A_5, A_4, H_4, E_4^{-1}]$ satisfying*

$$\begin{cases} A_1 \overset{E_1}{\rightarrow} \exists A_2, \\ A_2 \overset{E_2}{\rightarrow} A_3, \\ A_4 \overset{E_3^{-1}}{\rightarrow} \exists A_3, \end{cases} \tag{2}$$

we have $A_0 \overset{E}{\rightarrow} A_5$. That is, the cipher E has no ID over $A_0 \times A_5$.

Proof. Because $PT[A_0, A_1, H_0, E_0]$, we have $A_0 = \sum_{a_1 \in A_1} H_0[a_1]$. For any difference $a_0 \in A_0$, there exists $a_1 \in A_1$ satisfying $a_0 \overset{E_0}{\rightarrow} a_1$. According to Definition 3, if $A_1 \overset{E_1}{\rightarrow} \exists A_2$ is satisfied, for any $a_1 \in A_1$, there exists $a_2 \in A_2$ satisfying $a_1 \overset{E_1}{\rightarrow} a_2$. Therefore, for any difference $a_0 \in A_0$, there exists $a_2 \in A_2$ satisfying

$$a_0 \xrightarrow{E_1 \circ E_0} a_2. \tag{3}$$

Similarly, for any $a_5 \in A_5$, there exists $a_3 \in A_3$ satisfying $a_5 \xrightarrow{E_3^{-1} \circ E_4^{-1}} a_3$. Because $E_3^{-1} \circ E_4^{-1}$ is a bijective function, according to Lemma 1, for any difference $a_5 \in A_5$, there exists $a_3 \in A_3$ satisfying

$$a_3 \xrightarrow{E_4 \circ E_3} a_5. \tag{4}$$

Because $A_2 \overset{E_2}{\rightarrow} A_3$ holds, we have

$$a_2 \overset{E_2}{\rightarrow} a_3. \tag{5}$$

Combining the Eq. (3), (4) and (5) together, for any $a_0 \in A_0$ and $a_5 \in A_5$, there exist $a_2 \in A_2$ and $a_3 \in A_3$ satisfying

$$a_0 \xrightarrow{E_1 \circ E_0} a_2 \overset{E_2}{\rightarrow} a_3 \xrightarrow{E_4 \circ E_3} a_5.$$

Thus, we have $A_0 \overset{E}{\rightarrow} A_5$. □

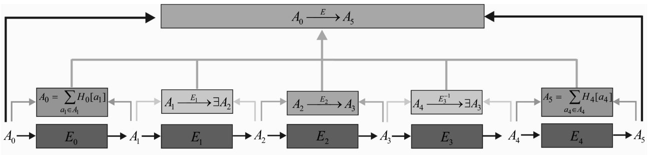

$A_0 \xrightarrow{E} A_5$
$A_0 = \sum_{a_1 \in A_1} H_0[a_1]$
$A_1 \xrightarrow{E_1} \exists A_2$
$A_2 \xrightarrow{E_2} A_3$
$A_4 \xrightarrow{E_3^{-1}} \exists A_3$
$A_5 = \sum_{a_4 \in A_4} H_4[a_4]$
A_0
E_0
A_1
E_1
A_2
E_2
A_3
E_3
A_4
E_4
A_5

Fig. 3. Ladder-first implementation strategy

According to Eq. (2), the partition tables of input difference set A_0 and output difference set A_5 can be considered separately. This will improve the efficiency of security evaluation against ID. For a cipher $E = E_4 \circ E_3 \circ E_2 \circ E_1 \circ E_0$, we construct a ladder $A_2 \overset{E_2}{\rightarrow} A_3$ and two partition tables $PT[A_0, A_1, H_0, E_0]$ and $PT[A_5, A_4, H_4, E_4^{-1}]$, where A_0 and A_5 are the input and output difference sets of E, respectively. In the fundamental stage, if $A_1 \overset{E_1}{\rightarrow} \exists A_2$ and $A_4 \overset{E_3^{-1}}{\rightarrow} \exists A_3$ are satisfied, according to Theorem 6, there is no ID for E over $A_0 \times A_5$. Otherwise, we obtain two sets $I = \{a_1 \in A_1 | a_1 \overset{E_1}{\nrightarrow} \exists A_2\}$ and $J = \{a_4 \in A_4 | a_4 \overset{E_3^{-1}}{\nrightarrow} \exists A_3\}$. In the enhance stage, similarly to improved partition-first implementation strategy in Sect. III-D, we can obtain the reduced hash tables $H'_0[a_1]$ and $H'_4[a_4]$ for any $a_1 \in A_1$ and $a_4 \in A_4$, respectively. Then, for any $a_1 \in I$ and $a_4 \in J$, we further explore whether $H'_0[a_1] \xrightarrow{E_1 \circ E_0} \exists A_2$ and $H'_4[a_4] \xrightarrow{E_3^{-1} \circ E_4^{-1}} \exists A_3$. The whole procedure for obtaining the ID result of E over $A_0 \times A_5$ is demonstrated in Algorithm 5.

B. Method for Determining Ladder

When using Algorithm 5 to evaluate the ID bound, we have to construct a ladder. To guide the selection of ladders, we propose the following theorem.

Theorem 7. *For a cipher $E = E_4 \circ E_3 \circ E_2 \circ E_1 \circ E_0$, let $A_2 \times A_3$ and $A'_2 \times A'_3$ be two ladders of E_2 satisfying $A_2 \times A_3 \subseteq A'_2 \times A'_3$. When applying Algorithm 5 to E, if we obtain the ID result $flag = 0$ when using ladder $A_2 \times A_3$, we can definitely get the ID result $flag = 0$ when using ladder $A'_2 \times A'_3$.*

Proof. According to Algorithm 5, only when $a_0 \overset{E_1 \circ E_0}{\rightarrow} \exists A_2$ and $a_5 \overset{E_3^{-1} \circ E_4^{-1}}{\rightarrow} \exists A_3$ hold for all $a_0 \in A_0, a_5 \in A_5$, the ID

Algorithm 5 Ladder-first implementation strategy

```
Require: The cipher $E = E_4 \circ \cdots \circ E_0$, input and output
    difference sets $A_0$ and $A_5$
Ensure: $flag$    ▷ Return the ID result of $E$ over $A_0 \times A_5$
——————- Fundamental Stage ——————-
$A_2 \xrightarrow{E_2} A_3$, $PT[A_0, A_1, H_0, E_0]$, $PT[A_5, A_4, H_4, E_4^{-1}]$ ▷
    Ladder and partition tables
Allocate $I \leftarrow \emptyset$ and $J \leftarrow \emptyset$
for $a_1 \in A_1$ do
    if $a_1 \overset{E_1}{\nrightarrow} \exists A_2$ then    ▷ Build a model to determine
    whether $a_1 \xrightarrow{E_1} \exists A_2$
        $I \leftarrow I \bigcup a_1$
    end if
end for
for $a_4 \in A_4$ do
    if $a_4 \overset{E_3^{-1}}{\nrightarrow} \exists A_3$ then    ▷ Build a model to determine
    whether $a_4 \xrightarrow{E_3^{-1}} \exists A_3$
        $J \leftarrow J \bigcup a_4$
    end if
end for
if $I = \emptyset$ and $J = \emptyset$ then
    return $flag = 0$    ▷ $E$ has no ID over $A_0 \times A_5$
end if
——————- Enhance Stage ——————-
$H'_0[a_1] = H_0[a_1] - \sum_{a \in A_1 - I} H_0[a]$ for any $a_1 \in A_1$
$H'_4[a_4] = H_4[a_4] - \sum_{a \in A_4 - J} H_4[a]$ for any $a_4 \in A_4$
for $a_1 \in I, a_0 \in H'_0[a_1]$ do
    if $a_0 \overset{E_1 \circ E_0}{\nrightarrow} \exists A_2$ then
        return $flag = 2$ ▷ Cannot determine whether $E$
    has ID
    end if
end for
for $a_4 \in J, a_5 \in H'_4[a_4]$ do
    if $a_5 \overset{E_3^{-1} \circ E_4^{-1}}{\nrightarrow} \exists A_3$ then
        return $flag = 2$ ▷ Cannot determine whether $E$
    has ID
    end if
end for
return $flag = 0$    ▷ $E$ has no ID over $A_0 \times A_5$
```

result $flag = 0$ can be obtained. Because $A_2 \times A_3 \subseteq A'_2 \times A'_3$, the conditions $a_0 \xrightarrow{E_1 \circ E_0} \exists A'_2$ and $a_5 \xrightarrow{E_3^{-1} \circ E_4^{-1}} \exists A'_3$ are met. Thus, we can get the ID result $flag = 0$ when using ladder $A'_2 \times A'_3$. □

The goal of this paper is to obtain the ID bounds of block ciphers. Compared with the ladder $A_2 \times A_3$, there is no doubt that $A'_2 \times A'_3$ is a better choice. Thus, we propose the following definition.

Definition 7. (Maximal Ladder). Let $A \times B$ be a ladder of function f. If there is no other ladder $A' \times B'$ of f satisfying $A \times B \subseteq A' \times B'$, we call $A \times B$ a *maximal ladder* of f.

According to Theorem 7, if a ladder $A \times B$ is not a maximal ladder, there always exists a better ladder. Thus, when applying Algorithm 5 to ciphers, only maximal ladders are used.

Theorem 8. (Maximal Ladder of S-box). *Let S be a bijective S-box. For any input difference $a \in \mathbb{F}_2^n$, we can obtain its output difference set, denoted as $DDT_S[a] = \{b \in \mathbb{F}_2^n | a \xrightarrow{S} b\}$. Thus, $A \times B$ is a maximal ladder of S if and only if the following conditions are satisfied.*

$$\begin{cases} B = \bigcap_{a \in A} DDT_S[a], \\ A = \bigcap_{b \in B} DDT_{S^{-1}}[b], \end{cases}$$

where S^{-1} is the inverse function of S.

Proof. **Sufficiency.** Because $B = \bigcap_{a \in A} DDT_S[a]$ is satisfied, we have $A \xrightarrow{S} B$ and there is no element $b' \notin B$ satisfying $A \xrightarrow{S} B \bigcup b'$. Similarly, there is no element $a' \notin A$ satisfying $B \xrightarrow{S^{-1}} A \bigcup a'$. According to Lemma 1, $B \xrightarrow{S^{-1}} A \bigcup a'$ is equivalent to $A \bigcup a' \xrightarrow{S} B$. Thus, there does not exist any $b' \notin B$ or $a' \notin A$ satisfying $A \bigcup a' \xrightarrow{S} B$ or $A \xrightarrow{S} B \bigcup b'$. Therefore, $A \times B$ is a maximal ladder of S.

Necessity. Because $A \times B$ is a ladder of S, we have $B \subseteq \bigcap_{a \in A} DDT_S[a]$. Since $A \xrightarrow{S} \bigcap_{a \in A} DDT_S[a]$ is also a ladder, the maximal ladder $A \times B$ must satisfy $B = \bigcap_{a \in A} DDT_S[a]$. According to Lemma 1, $B \times A$ is a maximal ladder of S^{-1}. Similarly, we have $A = \bigcap_{b \in B} DDT_{S^{-1}}[b]$. □

Based on the above theorem, we propose a heuristic method to obtain a maximal ladder of S. The whole procedure is demonstrated in Algorithm 6. Then, we can use the maximal ladders of small-size S-boxes to construct a maximal ladder of an S-box layer. The method is shown in Theorem 9.

Algorithm 6 Heuristic method for determining a maximal ladder

```
Require: The bijective S-box $S$, initial input difference set
    $A \neq \emptyset$
Ensure: A maximal ladder of $S$
Allocate $B \leftarrow \emptyset$
while 1 do
    $C = \bigcap_{a \in A} DDT_S[a] - B$
    Select a subset $C' \subseteq C$
    $B \leftarrow B + C'$                ▷ Expand the size of $B$
    $D = \bigcap_{b \in B} DDT_{S^{-1}}[b] - A$
    Select a subset $D' \subseteq D$
    $A \leftarrow A + D'$                ▷ Expand the size of $A$
    if $B = \bigcap_{a \in A} DDT_S[a]$ and $A = \bigcap_{b \in B} DDT_{S^{-1}}[b]$
    then
        return $A \times B$
    end if
end while
```

Theorem 9. (Maximal Ladder of an S-box Layer). *Let S be a function comprising of m parallel S-boxes, denoted as $S = s_{m-1} || s_{m-2} || \cdots || s_0$. For each $0 \leq i \leq m-1$, if $A_i \times B_i$ is a maximal ladder of s_i, then $\prod_{i=0}^{m-1} A_i \times \prod_{i=0}^{m-1} B_i$ is a maximal ladder of S.*

Proof. Because $A_i \times B_i$ is a ladder of s_i, for any $a_i \in A_i$ and $b_i \in B_i$, we have $a_i \xrightarrow{s_i} b_i$. For any $(a_{m-1}, a_{m-2}, \cdots, a_0) \in \prod_{i=0}^{m-1} A_i$ and $(b_{m-1}, b_{m-2}, \cdots, b_0) \in \prod_{i=0}^{m-1} B_i$, we have $(a_{m-1}, a_{m-2}, \cdots, a_0) \xrightarrow{S} (b_{m-1}, b_{m-2}, \cdots, b_0)$. Thus, $\prod_{i=0}^{m-1} A_i \times \prod_{i=0}^{m-1} B_i$ is a ladder of S.

If $\prod_{i=0}^{m-1} A_i \times \prod_{i=0}^{m-1} B_i$ is not a maximal ladder of S, there exists an element $(a'_{m-1}, a'_{m-2}, \ldots, a'_0) \notin \prod_{i=0}^{m-1} A_i$ or $(b'_{m-1}, b'_{m-2}, \ldots, b'_0) \notin \prod_{i=0}^{m-1} B_i$ satisfying $\left((a'_{m-1}, a'_{m-2}, \ldots, a'_0) \bigcup \prod_{i=0}^{m-1} A_i\right) \times \left(\prod_{i=0}^{m-1} B_i\right)$ or $\left(\prod_{i=0}^{m-1} A_i\right) \times \left((b'_{m-1}, b'_{m-2}, \ldots, b'_0) \bigcup \prod_{i=0}^{m-1} B_i\right)$ is also a ladder of S. Take one of the ladders $\left((a'_{m-1}, a'_{m-2}, \ldots, a'_0) \bigcup \prod_{i=0}^{m-1} A_i\right) \times \left(\prod_{i=0}^{m-1} B_i\right)$ as an example, for each $0 \leq i \leq m-1$, we have $a'_i \xrightarrow{s_i} B_i$. Because any $A_i \times B_i, 0 \leq i \leq m-1$ is a maximal ladder of s_i, we obtain that $a'_i \in A_i$. It is contradictory to $(a'_{m-1}, a'_{m-2}, \ldots, a'_0) \notin \prod_{i=0}^{m-1} A_i$. Similarly, we can also obtain the contradictory of $(b'_{m-1}, b'_{m-2}, \ldots, b'_0) \notin \prod_{i=0}^{m-1} B_i$. Therefore, $\prod_{i=0}^{m-1} A_i \times \prod_{i=0}^{m-1} B_i$ is a maximal ladder of S. □

C. Methods for Integrating a Ladder into Searching Models

After obtaining a ladder, we should integrate it into searching models (MILP or SAT). From **Line 3** and **Line 8** in Algorithm 5, we know that $|A_1| + |A_4|$ differential patterns need to be determined. For example, in **Line 4** of Algorithm 5, we need to determine whether $a_1 \xrightarrow{E_1} \exists A_2$ or not, where $A_2 \times A_3$ is a ladder of E_2. It should be noted that there is no automatic method for directly modeling this new kind of differential pattern before. For each $a_2 \in A_2$, previous automatic methods [9], [10] will build a model to determine whether $a_1 \xrightarrow{E_1} \exists a_2$. Thus, $|A_2|$ models need to be solved. This will greatly increase the complexity of Algorithm 5. In order to tackle this problem, we put forward a solution. Similar to current automatic searching models based on MILP or SAT, we introduce a sequence of variables and constraints satisfying the differential propagation rules. Take $a_1 \xrightarrow{E_1} \exists A_2$ as an example, we can construct a model $\mathcal{M}$ whose solutions are all possible differential characteristics of E_1. Let x and $y = y_{m-1}||y_{m-2}||\cdots||y_0$ be the variables representing the input and output differences of E_1.

When E_2 is a function comprising of m parallel bijective S-boxes, denoted as $E_2 = s_{m-1}||s_{m-2}||\cdots||s_0$. For any $0 \leq i \leq m-1$, we can construct a maximal ladder of s_i, denoted as $A_{2,i} \times A_{3,i}$. In order to model $a_1 \xrightarrow{E_1} \exists A_2 = A_{2,m-1} \times A_{2,m-2} \times \cdots \times A_{2,0}$, we add the following constraints into $\mathcal{M}$:

$$\mathcal{C} = \begin{cases} x = a_1, \\ y_i \neq d, \text{ where } d \in \{d \in \mathbb{F}_2^{n_i} | d \notin A_{2,i}\}, 0 \leq i \leq m-1, \end{cases}$$

where n_i is the dimension of s_i.

Then, if the whole model $\mathcal{M}+\mathcal{C}$ is feasible, we have $a_1 \xrightarrow{E_1} \exists A_2$. Otherwise, $a_1 \not\xrightarrow{E_1} \exists A_2$. Therefore, we can build only one model to determine whether $a_1 \xrightarrow{E_1} \exists A_2$ effectively.

D. Comparative Analysis of Improved Partition-First Implementation Strategy and Ladder-First Implementation Strategy

We will compare and analyze improved partition-first implementation strategy and ladder-first implementation strategy from efficiency and accuracy. Efficiency is about the number of models that need to be solved. Accuracy is about whether we can get the ID bound of a cipher or not. The enhance stages of Algorithm 4 and Algorithm 5 are greatly affected by the properties of specific ciphers and fundamental stages play a more important role in most cases. Thus, only the fundamental stages of Algorithm 4 and Algorithm 5 participate in the comparison. The comparison data of the two implementation strategies are showed in Table III.

TABLE III
THE COMPARISON OF IMPROVED PARTITION-FIRST AND LADDER-FIRST STRATEGIES

	Improved partition-first strategy (Algorithm 4)	Ladder-first strategy (Algorithm 5)
Cipher	$E = E_2 \circ E_1 \circ E_0$	$E = E'_4 \circ \cdots \circ E'_1 \circ E'_0$
Partition	$PT[A_0, A_1, H_0, E_0]$ $PT[A_3, A_2, H_2, E_2^{-1}]$	$PT[A'_0, A'_1, H'_0, E'_0]$ $PT[A'_5, A'_4, H'_4, E'^{-1}_4]$
Ladder	$A_1 \xrightarrow{E_1} A_2$	$A'_2 \xrightarrow{E'_2} A'_3$
Representative	–	$A'_1 \xrightarrow{E'_1} \exists A'_2$ $A'_4 \xrightarrow{E'^{-1}_3} \exists A'_3$
Models	$\lvert A_1\rvert \times \lvert A_2\rvert$	$\lvert A'_1\rvert + \lvert A'_4\rvert$

Under normal conditions, all input and output difference sets of the two strategies are partitioned over the same functions which means $E_0 = E'_0 = E''_0$ and $E_2 = E'_4 = E''_2$. Thus, $|A_1| = |A'_1|$ and $|A_2| = |A'_4|$.

Efficiency Comparison. From Table III, the number of models that need to be solved in Algorithm 4 is $|A_1| \times |A_2|$, while the number of models that need to be solved in Algorithm 5 is $|A'_1| + |A'_4|$. It should be noted that ladder-first implementation strategy shifts the complexity calculation from multiplication to addition, which has a huge impact on efficiency improvement. Thus, ladder-first implementation strategy is more efficient than improved partition-first implementation strategy.

Accuracy Comparison. If we obtain the result $flag = 0$ in the fundamental stage of Algorithm 5, it means that $A'_1 \xrightarrow{E'_1} \exists A'_2$ and $A'_4 \xrightarrow{E'^{-1}_3} \exists A'_3$. Because $A'_2 \times A'_3$ is a ladder of E'_2, we have $A'_1 \xrightarrow{E'_3 \circ E'_2 \circ E'_1} A'_4$ which means that Algorithm 4 will also return $flag = 0$. Thus, if Algorithm 5 can obtain the ID bound of cipher E, Algorithm 4 must also obtain the ID bound. But the opposition is not necessarily true. Therefore, improved partition-first implementation strategy is more accurate than ladder-first implementation strategy. If the time complexity is affordable, we first choose improved partition-first implementation strategy.

V. DYNAMIC-LADDER-PARTITION IMPLEMENTATION STRATEGY

This strategy will determine ladders and partition tables dynamically. For a cipher $E = E_2 \circ E_1 \circ E_0$, let A_0 and A_3 be the input and output difference sets, respectively. We will dynamically add elements into the ladder $A_1 \times A_2$ of E_1 until $A_0 \xrightarrow{E_0} \exists A_1$ and $A_3 \xrightarrow{E_2^{-1}} \exists A_2$ are satisfied or we obtain an ID. Then, we get the ID result of E over $A_0 \times A_3$. The whole procedure for obtaining the ID result of the cipher E is demonstrated in Algorithm 7.

According to **Line 4** and **Line 13** of Algorithm 7, the elements $a_0 \in A_0$ and $a_3 \in A_3$ are randomly selected. When $flag = 2$, if we want to get a more accurate result, we can call Algorithm 7 again. Because the ladders and partition tables of Algorithms 7 are determined dynamically, it is difficult for us to theoretically evaluate its efficiency and accuracy. According to our experimental results, dynamic-ladder-partition implementation strategy is more suitable for SPN ciphers whose number of elements in partition tables is little. For example, this strategy is used to prove that 5-round AES does not have any ID in Sect. VIII-A.

Algorithm 7 Dynamic-ladder-partition implementation strategy

```
Require: The cipher $E = E_2 \circ E_1 \circ E_0$, input and output difference sets $A_0$ and $A_3$
Ensure: $flag$    ▷ Return the ID result of $E$ over $A_0 \times A_3$
 1: Allocate $A_1 \leftarrow \emptyset$,$A_2 \leftarrow \emptyset$
 2: while $A_0 \neq \emptyset$ or $A_3 \neq \emptyset$ do
 3:     if $A_0 \neq \emptyset$ then
 4:         Randomly select an element $a_0 \in A_0$
 5:         if exists $a_1$ satisfying $a_0 \xrightarrow{E_0} a_1$ and $A_1 \cup a_1 \xrightarrow{E_1} A_2$ then
 6:             $A_0 \leftarrow A_0 - \{a_0 \in A_0 | a_0 \xrightarrow{E_0} a_1\}$    ▷ Remove elements represented by $a_1$
 7:             $A_1 \rightarrow A_1 \bigcup a_1$ ▷ Add element into the set $A_1$
 8:         else
 9:             return $flag = 2$
10:         end if
11:     end if
12:     if $A_3 \neq \emptyset$ then
13:         Randomly select an element $a_3 \in A_3$
14:         if there exists $a_2$ satisfying $a_3 \xrightarrow{E_2^{-1}} a_2$ and $A_1 \xrightarrow{E_1} A_2 \cup a_2$ then
15:             $A_3 \leftarrow A_3 - \{a_3 \in A_3 | a_3 \xrightarrow{E_2^{-1}} a_2\}$    ▷ Remove elements represented by $a_2$
16:             $A_2 \rightarrow A_2 \bigcup a_2$ ▷ Add element into the set $A_2$
17:         else
18:             return $flag = 2$
19:         end if
20:     end if
21:     if $A_0 = \emptyset$ and $A_3 = \emptyset$ then
22:         return $flag = 0$    ▷ $E$ has no ID over $A_0 \times A_3$
23:     end if
24: end while
```

VI. Exploring Rotation-Equivalence ID Set

In [28], Erlacher *et al.* exploited the rotational symmetry of ASCON and reduced the number of differential patterns that need to be considered. Inspired by their work, we propose the rotation-equivalence ID set defined as following.

Definition 8. (Rotation-Equivalence ID Set). *For a cipher E, let $A^m \subseteq \{a | a \in \mathbb{F}_2^{m\times n}\}$ and $B^m \subseteq \{b | b \in \mathbb{F}_2^{m\times n}\}$ be the input and output difference sets, respectively, where n is the dimension of the elements in A and B. $A^m \times B^m$ is called the rotation-equivalence ID set, if it satisfies the following conditions. For any $a \in A^m$, if there exists an output difference $b \in B^m$ satisfying $a \overset{E}{\nrightarrow} b$, then for each $1 \leq l \leq m-1$, there exists an output difference $b_l \in B^m$ satisfying $(a \lll l \times n) \overset{E}{\nrightarrow} b_l$.*

For the rotation-equivalence ID set $A^m \times B^m$ of E, we can divide the input difference set A^m into many disjoint subsets as following

$$A^m = \sum_{r\in R} \Omega_r, \tag{6}$$

where $R \subseteq A^m$ and $\Omega_r = \{r \lll l \times n | 0 \leq l \leq m-1\}$. According to Definition 8, all elements in Ω_r have the same result of determining whether E has ID. Thus, for each Ω_r, we only need to consider one element. This will reduce the number of differentials that need to be considered. In combinatorics terminology, the subset Ω_r in Eq. (6) is called $|A|$-ary **necklaces** of length m. According to Redfield-Pólya theorem [29], [30], the number of k-ary necklaces of length m is

$$N_k(m) = \frac{1}{m}\sum_{d|m} \varphi(d) \cdot k^{\frac{m}{d}}, \tag{7}$$

where φ is the Euler totient function and d is the divisor of m. For example, the number of 3-ary necklaces of length 4 is

$$\begin{aligned} &N_3(4) \\ &= \frac{1}{4}\left(\varphi(1)\cdot 3^{\frac{4}{1}} + \varphi(2)\cdot 3^{\frac{4}{2}} + \varphi(4)\cdot 3^{\frac{4}{4}}\right) \\ &= \frac{1}{4}\left(3^4 + 3^2 + 2\times 3\right) = 24. \end{aligned}$$

For $A^m \times B^m$ of E, there are $|A|^m \times |B|^m$ differential. If $A^m \times B^m$ is rotation-equivalence ID set of E, the number of disjoint subsets Ω_r in Eq. (6) is $|R| = N_{|A|}(m)$. Thus, when we evaluate the ID bound of E, only $N_{|A|}(m) \times |B|^m$ differentials need to be considered. Moreover, there is algorithm which can generating necklaces in constant amortized time, see [31].

VII. Applications to SPN Ciphers with Bit-Permutation Linear Layer

In order to improve the hardware efficiency, lightweight block ciphers often use bit-permutation linear layer. The representative algorithms are PRESENT [32] and GIFT [18].

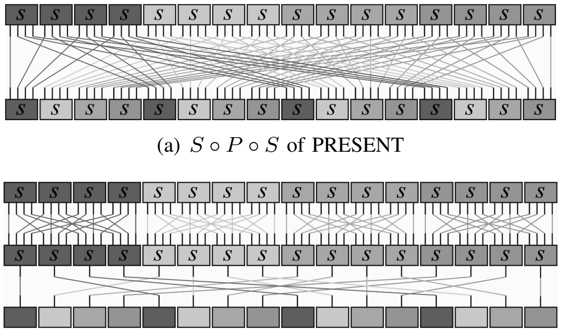
A. Application to PRESENT

PRESENT [32] is an important lightweight cipher. It adopts SPN structure with 64-bit block size through 31 rounds. Each round has three operations: AddRoundKey (XORed with a 64-bit round key), SubBox (16 parallel applications of the same 4-bit S-box, denoted by $S = s^{16||}$), BitPermutation (a bit-wise permutation of 64 bits, denoted as P). PRESENT is a Markov cipher. Under the assumption that the round keys are uniformly random, the AddRoundKey operation can be omitted. Therefore, the round function of PRESENT can be denoted as $R = P \circ S$. An illustration for $S \circ P \circ S$ is shown in Fig. 4(a). By introducing a bit oriented permutation $P_1 = [0,4,8,12,1,5,9,13,2,6,10,14,3,7,11,15]$ and a nibble oriented permutation $P_2 = [0,4,8,12,1,5,9,13,2,6,10,14,3,7,11,15]$, we can get an equivalent representation of $S \circ P \circ S$ as shown in Fig. 4(b). Then,

$$S \circ P \circ S = P_2 \circ S \circ (P_1||P_1||P_1||P_1) \circ S.$$

For $(r+4)$-round PRESENT R^{r+4}, because $P \circ P_2$ is a linear permutation, we omit $P \circ P_2$ in the last round. This will not affect the result of ID bound. Thus,

$$R^{r+4} = \underbrace{S \circ (P_1||P_1||P_1||P_1) \circ S}_{E_2} \circ \underbrace{R^r \circ P \circ P_2}_{E_1} \circ \underbrace{S \circ (P_1||P_1||P_1||P_1) \circ S}_{E_0}.$$



(a) $S \circ P \circ S$ of PRESENT

(b) $P_2 \circ S \circ (P_1||P_1||P_1||P_1) \circ S$ of PRESENT

Fig. 4. The functions of PRESENT

Next, we use Algorithm 2 to determine a minimal representative sets of $s^{4||} \circ P_1 \circ s^{4||}$ and $s^{-4||} \circ P_1^{-1} \circ s^{-4||}$, where $s^{-4||} = s^{-1}||s^{-1}||s^{-1}||s^{-1}$. From Table IV, we know that the number of elements in the minimal representative sets of $s^{4||} \circ P_1 \circ s^{4||}$ and $s^{-4||} \circ P_1^{-1} \circ s^{-4||}$ are 8 and 9, respectively. When applying Algorithm 4 to PRESENT, the number of models that need to be built in fundamental stage is $\left(8^4-1\right) \times \left(9^4-1\right) = 26863200 \approx 2^{24.68}$. After the fundamental stage of Algorithm 4, for 7-round and 8-round PRESENT, there are too many differentials which need to be further determined in enhance stage. Due to the limited storage and computing capacity, we cannot determine whether there exist IDs for 7-round and 8-round PRESENT. Then, we prove that 9-round PRESENT does not exist any ID under the sole condition that round keys are uniformly random.

TABLE IV
MINIMAL REPRESENTATIVE SETS FOR PRESENT

S-box	Minimal representative sets (hexadecimal)
$s^{4\|\|} \circ P_1 \circ s^{4\|\|}$	{0, 766, d33, 5060, 7000, 9779, ccee, 0300}
$s^{-4\|\|} \circ P_1^{-1} \circ s^{-4\|\|}$	{0, 700, 97a, bb0, 9000, ae55, b0d0, dddd, e7a7}

B. Applications to GIFT

As an improved version of PRESENT, GIFT [18] is composed of two version: GIFT-64 with 64-bit block size and GIFT-128 with 128-bit block size. It should be noted that the full state is not XORed with the round keys, but only half the round keys are XORed. When we assume that GIFT is a Markov cipher, it means that the half-state XOR of round key is replace with a full-state XOR. Similar to PRESENT, we omit the linear function $P \circ P_2$ in the last round. The $(r+4)$-round GIFT-64 can be written as

$$R^{r+4} = \underbrace{S \circ (P_1||P_1||P_1||P_1) \circ S}_{E_2} \circ \underbrace{R^r \circ P \circ P_2}_{E_1} \circ \underbrace{S \circ (P_1||P_1||P_1||P_1) \circ S}_{E_0}.$$

where $P_1 = [0,5,10,15,12,1,6,11,8,13,2,7,4,9,14,3]$ is a bit oriented permutation and $P_2 = [0,4,8,12,1,5,9,13,2,6,10,14,3,7,11,15]$ is a nibble oriented permutation. Then, we use Algorithm 2 to determine minimal representative sets of $s^{4||} \circ P_1 \circ s^{4||}$ and $s^{-4||} \circ P_1^{-1} \circ s^{-4||}$ shown in Table V. When applying Algorithm 4 to GIFT-64, the number of models that need to be built in fundamental stage is $\left(9^4-1\right) \times \left(8^4-1\right) = 26863200 \approx 2^{24.68}$. After the fundamental stage of Algorithm 4, for 7-round GIFT64, there are too many differentials which need to be further determined in enhance stage. Due to the limited storage and computing capacity, we cannot determine whether there exist IDs for 7-round GIFT64. Then, we prove that 8-round GIFT-64 does not exist any ID under the assumptions that GIFT-64 is a Markov cipher and the round keys are uniformly random.

TABLE V
MINIMAL REPRESENTATIVE SETS FOR GIFT-64 AND GIFT-128

S-box	Minimal representative set (hexadecimal)
$s^{4\|\|} \circ P_1 \circ s^{4\|\|}$	{0, 505, 55f, f35, 350f, 50f7, 5f09, 9d9d, b750}
$s^{-4\|\|} \circ P_1^{-1} \circ s^{-4\|\|}$	{0, d, f9, d00, 7dda, 9b00, cf9c, fccd}

For GIFT-128, if we apply Algorithm 4 to it, the number of models that need to be built in the fundamental stage is about $(9^8-1) \times (8^8-1) \approx 2^{49.36}$ which is not affordable. Thus, we will use Algorithm 5 to evaluate its ID bound. For GIFT-128, when we omit the linear function $P \circ P_2$ in the last round, (r_1+r_2+5)-round GIFT-128 can be written as

$$R^{r_1+r_2+5} = \underbrace{S \circ P_1^{8||} \circ S}_{E_4} \circ \underbrace{R^{r_2} \circ P}_{E_3} \circ \underbrace{S}_{E_2} \circ \underbrace{R^{r_1} \circ P \circ P_2}_{E_1} \circ \underbrace{S \circ P_1^{8||} \circ S}_{E_0},$$

where $P_1 = [0,5,10,15,12,1,6,11,8,13,2,7,4,9,14,3]$ is a bit oriented permutation (same with that in GIFT-64) and $P_2 = [0,8,16,24,1,9,17,25,2,10,18,26,3,11,19,27,4,12,20,28,5,13,21,29,6,14,22,30,7,15,23,31]$ is a nibble oriented permutation. Then, we use Algorithm 6 to find a maximal ladder of the 4-bit S-box used in GIFT-128. When we apply Algorithm 5 to $(r_1 + r_2 + 5)$-round GIFT-128, the number of models that need to be built in fundamental stage is $\left(9^8 - 1\right) + \left(8^8 - 1\right) = 59823935 \approx 2^{25.83}$. By setting $r_1 = 4$ and $r_2 = 3$, we prove that 12-round GIFT-128 does not exist any ID under the assumptions that GIFT-128 is a Markov cipher and the round keys are uniformly random.

VIII. Applications to SPN Ciphers with Non-Bit-Permutation Linear Layer

A. Applications to Rijndael

Rijndael [23] was designed by Daemen and Rijmen in 1998. According to block size, Rijndael can be divided into Rijndael-128, Rijndael-160, Rijndael-192, Rijndael-224 and Rijndael-256. The 128-bit block size version Rijndael-128 was selected as AES. For Rijndael-$32n$, $n \in \{4, 5, 6, 7, 8\}$, the state is viewed as $4 \times n$ rectangle array of 8-bit words. The round function of Rijndael-$32n$ consists of the following four operations: SubBox ($4 \times n$ parallel applications of the same 8-bit Sbox, denoted as $S = s^{4 \times n||}$), ShiftRow (a byte transposition that cyclically shifts the rows of the state over different offsets, denoted as SR), MixColumn (a linear matrix M is multiplied to each column of the state, denoted as MC), AddRoundKey (XORed with a $32n$-bit round key). All versions of Rijndael are Markov ciphers. When the round keys are uniformly random, we do not need to consider the AddRoundKey operation. Therefore, the round function of Rijndael-$32n$ can be denoted as $R = MC \circ SR \circ S$. Because SR and MC are linear operations, we omit SR operation of the first round and the $MC \circ SR$ operation of the last round. This will not affect the result of ID bound. For $(r + 4)$-round Rijndael-$32n$, we have

$$R^{r+4} = \underbrace{S \circ MC \circ S}_{E_2} \circ \underbrace{SR \circ R^r \circ MC \circ SR}_{E_1} \circ \underbrace{S \circ MC \circ S}_{E_0}. \tag{8}$$

The functions E_0 and E_2^{-1} of Rijndael-$32n$ can be seen as n parallel 32-bit superboxes $s^{4||} \circ M \circ s^{4||}$ and $s^{-4||} \circ M^{-1} \circ s^{-4||}$, respectively. Next, we use Algorithm 3 to determine representative sets of $s^{4||} \circ M \circ s^{4||}$ and $s^{-4||} \circ M^{-1} \circ s^{-4||}$. From Table VI, we know that both the numbers of elements in the representative sets of $s^{4||} \circ MC \circ s^{4||}$ and $s^{-4||} \circ M^{-1} \circ s^{-4||}$ are 2. Then, we explore the rotation-equivalence ID sets of Rijndael-$32n$ shown in Theorem 10.

Theorem 10. *For Rijndael-$32n$, let a_1 and a_2 be the input and output differences of E_1, respectively. If $a_1 \overset{E_1}{\nrightarrow} a_2$ is satisfied, then $SR_i\left(a_1\right) \overset{E_1}{\nrightarrow} SR_i\left(a_2\right)$ holds for all $i \in \{1, 2, \ldots, n-1\}$, where SR_i means cyclically shifting every row of the state over i bytes.*

Proof. According to the definitions of SR, MC and S, we have the following equations

$$\begin{cases} SR \circ SR_i = SR_i \circ SR \\ MC \circ SR_i = SR_i \circ MC \\ S \circ SR_i = SR_i \circ S \end{cases}$$

Thus, $a_1 \overset{E_1}{\nrightarrow} a_2$ is equivalent to $SR_i\left(a_1\right) \overset{E_1}{\nrightarrow} SR_i\left(a_2\right), i \in \{1, 2, \ldots, n-1\}$. □

TABLE VI
Representative sets of Rijndael-$32n$

S-box	Representative sets (hexadecimal)
$s^{4\|\|} \circ M \circ s^{4\|\|}$	{0, f8f9f9f9}
$s^{-4\|\|} \circ M^{-1} \circ s^{-4\|\|}$	{0, f8faf8f8}

We apply Algorithm 4 to Rijndael-$32n$. According to Sect. VI, the number of models that need to be built in fundamental stage is $\left(N_2\left(n\right) - 1\right) \times \left(2^n - 1\right)$. Then, we prove that 6-round AES (Rijndael-128), 6-round Rijndael-160, 7-round Rijndael-192, 7-round Rijndael-224, 7-round Rijndael-256 do not have any ID under the sole assumption that round keys are uniformly random.

Because the longest known ID of AES (Rijndael-128) is 4 rounds, the security bound obtained by us has room for improvement. Therefore, we apply Algorithm 7 to AES. The specific process is as following. Similarly to the above analysis, 5-round AES can be written as,

$$R^5 = \underbrace{S \circ MC \circ S}_{E_2} \circ \underbrace{SR \circ MC \circ SR \circ S \circ MC \circ SR}_{E_1} \circ \underbrace{S \circ MC \circ S}_{E_0}.$$

Let $A_0 = A_{0,3} \times A_{0,2} \times A_{0,1} \times A_{0,0}$ and $A_3 = A_{3,3} \times A_{3,2} \times A_{3,1} \times A_{3,0}$ be the sets of all nonzero input and output differences of AES, respectively. Thus, the whole search space $A_0 \times A_3$ can be divided into the following $15 \times 15 = 225$ disjoint subsets.

$$\begin{aligned} & A_0 \times A_3 = \\ & \textstyle\sum_{(i_0,i_1,i_2,i_3)\in\mathbb{F}_2^{4*},(j_0,j_1,j_2,j_3)\in\mathbb{F}_2^{4*}} [A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0} \\ & \times [A_{3,3}]^{j_3} \times \cdots \times [A_{3,0}]^{j_0} \end{aligned}$$

where $\mathbb{F}_2^{4*} = \{a \in \mathbb{F}_2^4 | a \neq 0\}$ is the set of all nonzero 4-bit vectors. For any $i \in \{0, 3\}$ and $m \in \{0, 1, 2, 3\}$, let $[A_{i,m}]^0 = \{0 \in \mathbb{F}_2^{32}\}$ be the set of only 32-bit zero difference and $[A_{i,m}]^1 = \{a \in \mathbb{F}_2^{32} | a \neq 0\}$ be the set of all nonzero 32-bit differences. According to Theorem 10, we only need to consider $\left(N_2\left(4\right) - 1\right) \times \left(2^4 - 1\right) = 75$ disjoint subsets.

For any one of the above subsets, we select $a_0 = (a_{0,3}, a_{0,2}, a_{0,1}, a_{0,0}) \in [A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0}$ and $a_3 = (a_{3,3}, a_{3,2}, a_{3,1}, a_{3,0}) \in [A_{3,3}]^{j_3} \times \cdots \times [A_{3,0}]^{j_0}$ and build a model to obtain $a_1 = (a_{1,3}, a_{1,2}, a_{1,1}, a_{1,0})$ and $a_2 = (a_{2,3}, a_{2,2}, a_{2,1}, a_{2,0})$ satisfying $a_0 \overset{E_0}{\to} a_1$, $a_1 \overset{E_1}{\to} a_2$ and $a_3 \overset{E_2^{-1}}{\to} a_2$. If $[A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0} \overset{E_0}{\to} a_1$ and $[A_{3,3}]^{j_3} \times \cdots \times [A_{3,0}]^{j_0} \overset{E_2^{-1}}{\to} a_2$ are satisfied, all the differentials in subset $[A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0} \times [A_{3,3}]^{j_3} \times \cdots \times [A_{3,0}]^{j_0}$ over E are possible.

The method for verifying $[A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0} \overset{E_0}{\to} a_1$ and $[A_{3,3}]^{j_3} \times \cdots \times [A_{3,0}]^{j_0} \overset{E_2^{-1}}{\to} a_2$ is as following. Take $[A_{0,3}]^{i_3} \times \cdots \times [A_{0,0}]^{i_0} \overset{E_0}{\to} a_1$ as an example, we just need to verify whether $[A_{0,m}]^{i_m} \overset{s^{4||} \circ M \circ s^{4||}}{\to} a_{1,m}$ holds for all $m = 0, 1, 2, 3$. For any i_m, if $i_m = 0$ is satisfied, we only need to verify 1 difference. If $i_m = 1$ is satisfied, we have to verify $2^{32} - 1$ input differences in $[A_{0,m}]^{i_m}$. In order to improve the success rate, if $i_m = 1$ is satisfied, we add a constrain to $a_{1,m}$ that every byte of $a_{1,m}$ is nonzero. After verifying all the disjoint subsets, we prove that 5-round AES does not have any ID under the sole assumption that round keys are uniformly random.

B. Application to SKINNY-128

SKINNY family ciphers were proposed at CRYPTO 2016 [24]. As the ISO standard block ciphers, SKINNY family ciphers have 64-bit and 128-bit block versions, denoted as SKINNY-64 and SKINNY-128 respectively. It should be noted that the full state is not XORed with the round keys, but only half the round keys are XORed. When we assume that SKINNY is a Markov cipher, it means that the half-state XOR of round key is replace with a full-state XOR. Because the methods in [16] cannot obtain the ID bound of SKINNY-128, we apply our methods into SKINNY-128. For SKINNY-128, the internal state is viewed as a 4×4 square array of cells, where each cell is a byte. Each round function consists of the following five operations: SubCells (the same 8-bit Sbox S is applied to every cell of the cipher internal state, denoted as $SC = S^{16||}$), AddConstants (XORed with round constant), AddRoundTweakey (XORed with round key), ShiftRow (the rows of the cipher state cell array are rotated, denoted as SR), MixColumn (each column of the cipher internal state array is multiplied by a binary matrix M, denoted as MC). Under the assumptions that SKINNY-128 is a Markov cipher and the round keys are uniformly random, we do not need to consider the AddConstants and AddRoundTweakey operations. Therefore, the round function of SKINNY-128 can be denoted as $R = MC \circ SR \circ S$. Because SR and MC are linear operations, we omit SR operation of the first round and the $MC \circ SR$ operation of the last round. This will not affect the result of ID bound. For $(r_1 + r_2 + 5)$-round SKINNY-128, we have

$$\begin{aligned} &R^{r_1+r_2+5} = \\ &\underbrace{SC \circ MC \circ SC}_{E_4} \circ \underbrace{SR \circ R^{r_2} \circ MC \circ SR}_{E_3} \circ \underbrace{SC}_{E_2} \\ &\circ \underbrace{R^{r_1} \circ MC \circ SR}_{E_1} \circ \underbrace{SC \circ MC \circ SC}_{E_0}. \end{aligned}$$

The functions E_0 and E_4^{-1} of SKINNY-128 can be seen as 4 parallel 32-bit S-boxes $S^{4||} \circ M \circ S^{4||}$ and $S^{-4||} \circ M^{-1} \circ S^{-4||}$, respectively. Next, we use Algorithm 3 to determine representative sets of $S^{4||} \circ M \circ S^{4||}$ and $S^{-4||} \circ M^{-1} \circ S^{-4||}$. Thus, the number of elements in the representative sets of $S^{4||} \circ M \circ S^{4||}$ and $S^{-4||} \circ M^{-1} \circ S^{-4||}$ are 86 and 134. Then, we use Algorithm 6 to find a maximal ladder of the 8-bit S-box used in SKINNY-128. Because SKINNY-128 has a round structure similar to Rijndael, SKINNY-128 also has rotation-equivalence ID sets. When we apply Algorithm 5 to $(r_1 + r_2 + 5)$-round SKINNY-128, the number of models that need to be built in fundamental stage is $(N_{86}(4) - 1) + (N_{134}(4) - 1) = 94286134 \approx 2^{26.49}$. By setting $r_1 = 5$ and $r_2 = 4$, we prove that 14-round SKINNY-128 does not exist any ID under the assumptions that SKINNY-128 is a Markov cipher and the round keys are uniformly random.

IX. CONCLUSION

In this paper, a series of methods for bounding the length of IDs of SPN block ciphers are proposed. Our methods are widely applicable. This is of great significance for evaluating the security of SPN block ciphers against ID attack. However, our ID bounds are obtained under some assumptions, how to obtain the ID bounds of SPN ciphers under no assumption is an open problem that needs to be tackled.

Acknowledgement. The authors would like to thank the anonymous reviewers for their detailed comments and suggestions. This work was supported by the National Natural Science Foundation of China [Grant No.62102448, 62202493].

REFERENCES

[1] L. R. Knudsen, Deal - a 128-bit block cipher, Technical report, Department of Informatics, University of Bergen, Norway (1998).

[2] E. Biham, A. Biryukov, A. Shamir, Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials, in: J. Stern (Ed.), Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding, Vol. 1592 of Lecture Notes in Computer Science, Springer, 1999, pp. 12–23. doi:10.1007/3-540-48910-X_2.
URL https://doi.org/10.1007/3-540-48910-X_2

[3] H. Mala, M. Dakhilalian, V. Rijmen, M. Modarres-Hashemi, Improved impossible differential cryptanalysis of 7-round AES-128, in: G. Gong, K. C. Gupta (Eds.), Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings, Vol. 6498 of Lecture Notes in Computer Science, Springer, 2010, pp. 282–291. doi:10.1007/978-3-642-17401-8_20.
URL https://doi.org/10.1007/978-3-642-17401-8_20

[4] J. Kim, S. Hong, J. Sung, C. Lee, S. Lee, Impossible differential cryptanalysis for block cipher structures, in: T. Johansson, S. Maitra (Eds.), Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings, Vol. 2904 of Lecture Notes in Computer Science, Springer, 2003, pp. 82–96. doi:10.1007/978-3-540-24582-7_6.
URL https://doi.org/10.1007/978-3-540-24582-7_6

[5] Y. Luo, X. Lai, Z. Wu, G. Gong, A unified method for finding impossible differentials of block cipher structures, Inf. Sci. 263 (2014) 211–220. doi:10.1016/j.ins.2013.08.051.
URL https://doi.org/10.1016/j.ins.2013.08.051

[6] S. Wu, M. Wang, Automatic search of truncated impossible differentials for word-oriented block ciphers, in: S. D. Galbraith, M. Nandi (Eds.), Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings, Vol. 7668 of Lecture Notes in Computer Science, Springer, 2012, pp. 283–302. doi:10.1007/978-3-642-34931-7_17.
URL https://doi.org/10.1007/978-3-642-34931-7_17

[7] L. Sun, D. Gérault, W. Wang, M. Wang, On the usage of deterministic (related-key) truncated differentials and multidimensional linear approximations for SPN ciphers, IACR Trans. Symmetric Cryptol. 2020 (3) (2020) 262–287. doi:10.13154/tosc.v2020.i3.262-287.
URL https://doi.org/10.13154/tosc.v2020.i3.262-287

[8] H. Hadipour, S. Sadeghi, M. Eichlseder, Finding the impossible: Automated search for full impossible-differential, zero-correlation, and integral attacks, in: C. Hazay, M. Stam (Eds.), Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV, Vol. 14007 of Lecture Notes in Computer Science, Springer, 2023, pp. 128–157. doi:10.1007/978-3-031-30634-1_5.
URL https://doi.org/10.1007/978-3-031-30634-1_5

[9] T. Cui, K. Jia, K. Fu, S. Chen, M. Wang, New automatic search tool for impossible differentials and zero-correlation linear approximations, IACR Cryptol. ePrint Arch. (2016) 689.
URL http://eprint.iacr.org/2016/689

[10] Y. Sasaki, Y. Todo, New impossible differential search tool from design and cryptanalysis aspects - revealing structural properties of several ciphers, in: J. Coron, J. B. Nielsen (Eds.), Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III, Vol. 10212 of Lecture Notes in Computer Science, 2017, pp. 185–215. doi:10.1007/978-3-319-56617-7_7.
URL https://doi.org/10.1007/978-3-319-56617-7_7

[11] T. Cui, C. Jin, B. Zhang, Z. Chen, G. Zhang, Searching all truncated impossible differentials in SPN, IET Inf. Secur. 11 (2) (2017) 89–96. doi:10.1049/iet-ifs.2015.0052.
URL https://doi.org/10.1049/iet-ifs.2015.0052

[12] B. Sun, M. Liu, J. Guo, V. Rijmen, R. Li, Provable security evaluation of structures against impossible differential and zero correlation linear cryptanalysis, in: M. Fischlin, J. Coron (Eds.), 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, Vol. 9665 of Lecture Notes in Computer Science, Springer, 2016, pp. 196–213. doi:10.1007/978-3-662-49890-3_8.
URL https://doi.org/10.1007/978-3-662-49890-3_8

[13] Q. Wang, C. Jin, Bounding the length of impossible differentials for SPN block ciphers, Des. Codes Cryptogr. 89 (11) (2021) 2477–2493. doi:10.1007/s10623-021-00932-1.
URL https://doi.org/10.1007/s10623-021-00932-1

[14] Q. Wang, C. Jin, More accurate results on the provable security of AES against impossible differential cryptanalysis, Des. Codes Cryptogr. 87 (12) (2019) 3001–3018. doi:10.1007/s10623-019-00660-7.
URL https://doi.org/10.1007/s10623-019-00660-7

[15] C. Boura, D. Coggia, Efficient MILP modelings for sboxes and linear layers of SPN ciphers, IACR Trans. Symmetric Cryptol. 2020 (3) (2020) 327–361. doi:10.13154/tosc.v2020.i3.327-361.
URL https://doi.org/10.13154/tosc.v2020.i3.327-361

[16] K. Hu, T. Peyrin, M. Wang, Finding all impossible differentials when considering the DDT, IACR Cryptol. ePrint Arch. (2022) 1034.
URL https://eprint.iacr.org/2022/1034

[17] X. Hu, Y. Li, L. Jiao, S. Tian, M. Wang, Mind the propagation of states - new automatic search tool for impossible differentials and impossible polytopic transitions, in: S. Moriai, H. Wang (Eds.), Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I, Vol. 12491 of Lecture Notes in Computer Science, Springer, 2020, pp. 415–445. doi:10.1007/978-3-030-64837-4_14.
URL https://doi.org/10.1007/978-3-030-64837-4_14

[18] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, Y. Todo, GIFT: A small present - towards reaching the limit of lightweight encryption, in: W. Fischer, N. Homma (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, Vol. 10529 of Lecture Notes in Computer Science, Springer, 2017, pp. 321–345. doi:10.1007/978-3-319-66787-4_16.
URL https://doi.org/10.1007/978-3-319-66787-4_16

[19] L. Zhang, W. Wu, J. H. Park, B. Koo, Y. Yeom, Improved impossible differential attacks on large-block rijndael, in: T. Wu, C. Lei, V. Rijmen, D. Lee (Eds.), Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings, Vol. 5222 of Lecture Notes in Computer Science, Springer, 2008, pp. 298–315. doi:10.1007/978-3-540-85886-7_21.
URL https://doi.org/10.1007/978-3-540-85886-7_21

[20] J. N. Jr., I. C. Pavão, Impossible-differential attacks on large-block rijndael, in: J. A. Garay, A. K. Lenstra, M. Mambo, R. Peralta (Eds.), Information Security, 10th International Conference, ISC 2007, Valparaíso, Chile, October 9-12, 2007, Proceedings, Vol. 4779 of Lecture Notes in Computer Science, Springer, 2007, pp. 104–117. doi:10.1007/978-3-540-75496-1_7.
URL https://doi.org/10.1007/978-3-540-75496-1_7

[21] J. Daemen, V. Rijmen, Probability distributions of correlation and differentials in block ciphers, J. Math. Cryptol. 1 (3) (2007) 221–242. doi:10.1515/JMC.2007.011.
URL https://doi.org/10.1515/JMC.2007.011

[22] X. Lai, J. L. Massey, S. Murphy, Markov ciphers and differential cryptanalysis, in: D. W. Davies (Ed.), Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings, Vol. 547 of Lecture Notes in Computer Science, Springer, 1991, pp. 17–38. doi:10.1007/3-540-46416-6_2.
URL https://doi.org/10.1007/3-540-46416-6_2

[23] J. Daemen, V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Information Security and Cryptography, Springer, 2002. doi:10.1007/978-3-662-04722-4.
URL https://doi.org/10.1007/978-3-662-04722-4

[24] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, S. M. Sim, The SKINNY family of block ciphers and its low-latency variant MANTIS, in: M. Robshaw, J. Katz (Eds.), Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II, Vol. 9815 of Lecture Notes in Computer Science, Springer, 2016, pp. 123–153. doi:10.1007/978-3-662-53008-5_5.
URL https://doi.org/10.1007/978-3-662-53008-5_5

[25] N. Mouha, Q. Wang, D. Gu, B. Preneel, Differential and linear cryptanalysis using mixed-integer linear programming, in: C. Wu, M. Yung, D. Lin (Eds.), Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers, Vol. 7537 of Lecture Notes in Computer Science, Springer, 2011, pp. 57–76. doi:10.1007/978-3-642-34704-7_5.
URL https://doi.org/10.1007/978-3-642-34704-7_5

[26] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, L. Song, Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers, in: P. Sarkar, T. Iwata (Eds.), Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I, Vol. 8873 of Lecture Notes in Computer Science, Springer, 2014, pp. 158–178. doi:10.1007/978-3-662-45611-8_9.
URL https://doi.org/10.1007/978-3-662-45611-8_9

[27] Y. Sasaki, Y. Todo, New algorithm for modeling s-box in MILP based differential and division trail search, in: P. Farshim, E. Simion (Eds.), Innovative Security Solutions for Information Technology and Communications - 10th International Conference, SecITC 2017, Bucharest, Romania, June 8-9, 2017, Revised Selected Papers, Vol. 10543 of Lecture Notes in Computer Science, Springer, 2017, pp. 150–165. doi:10.1007/978-3-319-69284-5_11.
URL https://doi.org/10.1007/978-3-319-69284-5_11

[28] J. Erlacher, F. Mendel, M. Eichlseder, Bounds for the security of ascon against differential and linear cryptanalysis, IACR Trans. Symmetric Cryptol. 2022 (1) (2022) 64–87. doi:10.46586/tosc.v2022.i1.64-87.
URL https://doi.org/10.46586/tosc.v2022.i1.64-87

[29] J. H. Redfield, The theory of group-reduced distributions, American Journal of Mathematics 49 (3) (1927) 433–455.
URL http://www.jstor.org/stable/2370675

[30] G. Pólya, Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen, Acta Mathematica 68 (none) (1937) 145 – 254. doi:10.1007/BF02546665.
URL https://doi.org/10.1007/BF02546665

[31] K. Cattell, F. Ruskey, J. Sawada, M. Serra, C. R. Miers, Fast algorithms to generate necklaces, unlabeled necklaces, and irreducible polynomials over GF(2), J. Algorithms 37 (2) (2000) 267–282. doi:10.1006/jagm.2000.1108.
URL https://doi.org/10.1006/jagm.2000.1108

[32] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: an ultra-lightweight block cipher, in: P. Paillier, I. Verbauwhede (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, Vol. 4727 of Lecture Notes in Computer Science, Springer, 2007, pp. 450–466. doi:10.1007/978-3-540-74735-2_31.
URL https://doi.org/10.1007/978-3-540-74735-2_31

Senpeng Wang He received the B.E. degree in 2014, M.S. degree in 2017, and Ph.D. in 2020 from Information and Engineering University. His research interests include information security and cryptology.

Dengguo Feng He is a professor of State Key Laboratory of Cryptology. His research interests include symmetric cryptography and quantum cryptanalysis.

Tairong Shi She received the Ph.D. degree from Information and Engineering University, Zhengzhou, China in 2021. Her research interests include symmetric cryptography and quantum cryptanalysis.

Bin Hu He is a professor of the Information Engineering University, China. His main subject interests and his main teaching are Boolean function, information security and cryptology. He received Ph.D degree in cryptography from Information Engineering University in 2008.

Jie Guan She is a professor of the Information Engineering University, China. Her main subject interest is cryptography and her main teaching lies in the areas of information systems, the theory of cryptography and quantum computation. She received Ph.D. degree in cryptography from Information Engineering University in 2004.

Kai Zhang He is an instructor of the Information Engineering University, China. He received the M.S. and Ph.D. degree in cryptology from this university in 2013 and 2016. He was a postdoctoral fellow at Shanghai Jiao Tong University. His main research interests lie in cryptography and cryptanalysis. His works have been published in several refereed journals and he has been serving as a referee for several international journals in the area of information security and cryptology.

Ting Cui He is a professor of Information Engineering University. His research interests include symmetric cryptography and quantum cryptanalysis.