Automatic Security Evaluation of Block Ciphers with S-bP Structures against Related-key Differential Attacks

Siwei Sun, Lei Hu, Ling Song, Yonghong Xie, Peng Wang

State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
swsun@is.ac.cn, hu@is.ac.cn

Abstract. Counting the number of active S-boxes is a common way to evaluate the security of symmetric key cryptographic schemes against differential attack. Based on Mixed Integer Linear Programming (MILP), Mouha et al proposed a method to accomplish this task automatically for word-oriented symmetric-key ciphers with SPN structures. However, this method can not be applied directly to block ciphers of SPN structures with bitwise permutation diffusion layers (S-bP structures), due to its ignorance of the diffusion effect derived collaboratively by nonlinear substitution layers and bitwise permutation layers. Moreover, the MILP constrains presented in Mouha et al's method are not enough to describe the differential propagation behaviour of a linear diffusion layer constructed from a non-MDS code, even an almost MDS code. In this paper we extend Mouha et al's method for S-bP structures by introducing new representations for exclusive-or (XOR) differences to describe bit/word level differences simultaneously and by taking the collaborative diffusion effect of S-boxes and bitwise permutations into account. Our method is applied to the block cipher PRESENT-80, an international standard for lightweight symmetric key cryptography, to automatically evaluate its security against differential attacks. We obtain lower bounds on the numbers of active S-boxes in the single-key model for full 31-round PRESENT-80 and in related-key model for round-reduced PRESENT-80 up to 12 rounds, and therefore automatically prove that the full-round PRESENT-80 is secure against single-key differential attack, and the cost of related-key differential attack on the full-round PRESENT-80 is close to that of an exhaustive search: the best related-key differential characteristic for full PRESENT-80 is upper bounded by 2^{-72}.

Keywords: Block cipher, SPN structure, Differential attack, Active S-box, Mixed-integer Linear Programming

1 Introduction

Differential cryptanalysis [6] and linear cryptanalysis [21] are two of the most important attacks on symmetric-key cryptographic schemes, based on which

a whole bunch of techniques for analysing block ciphers are devised, such as related-key differential attack [4], impossible differential attack [5] and zero correlation attack [8]. Resistance against differential and linear attacks is a basic requirement for today's design of block ciphers.

After the introduction of the wide trail strategy [13] by the designers of AES, provable security against differential cryptanalysis comes from a similar argument for almost all newly designed block ciphers. That is, the designers provide a very small upper bound for the probability of the best differential characteristic of the cipher by showing a lower bound on the number of active S-boxes for any consecutive r rounds of the cipher. Therefore, how to find the minimum number of active S-boxes is of great interest.

Actually, lots of works have been done in this direction for both classes of block ciphers with substitution-permutation network (SPN) and Feistel structures. These methods can be classified into two categories.

In the first category, the lower bound is proved mathematically. In [14], the wide trail design strategy ensures that there are at least 25 active S-boxes for any 5-round AES, and the designers of PRESENT [7] proved that any 5-round differential characteristic of PRESENT-80 had a minimum of 10 active S-boxes. Results concerning block ciphers with Feistel or generalized Feistel structure can be found in [18, 24, 27, 29]. This kind of methods is tricky, and sometimes many possible cases of the differential propagation must be considered.

In the second category, algorithms are designed to count the number of active S-boxes automatically. In [3], Aoki et al used a variant of Matsui's algorithm [22] to compute a lower bound on the minimal number of active S-boxes for the block cipher Camellia, and therefore proved its security against differential attack. The minimum number of active S-boxes for generalized Feistel structure was obtained in [24] by an algorithm which searches word-based truncated differentials. Highly automatic methods employing Mixed Integer Liner Programming (MILP) were presented in [23, 26] to determine the minimum number of active S-boxes for SPN structures and Feistel structures with SPN round functions.

In this paper, we are mostly interested in the methods based on MILP since they are the most automatic methods and require less programming effort compared with other methods. Using this method, what an analyst need to do is just to write a program to generate the MILP instance with suitable objective function and constraints imposed by the differential propagation of the cipher. The remaining work for determining the bounds can be done by a highly optimized open-source or commercially available software such as CPLEX [12], SCIP [1] and Gurobi [15].

Contribution of this paper. In this paper, we focus on how to determine the minimum number of active S-boxes in the single-key or related-key model for block ciphers of SPN structures with bitwise permutation diffusion layers (S-bP structures). We point out that Mouha et al's method is not applicable to block ciphers with bitwise permutations or non-MDS, even almost MDS diffusion layers. By extending Mouha et al's method, we propose an MILP based approach to prove the security of block ciphers of S-bP structures against

single-key or related-key differential attacks automatically. Compared with other proving methods like that presented in [7], it is highly automatic.

We have implemented our method on a personal computer: a Python module was developed to generate the MILP instances, and the Gurobi optimizer [15] was employed as the underlying MILP solver. Experimental results showed that we can automatically prove that the block cipher PRESENT-80 is secure against single-key differential attack within only 222 seconds. We have also found that there are at least 15 active S-boxes in any related-key differential characteristic for 12-round PRESENT-80, and the probability of the best related-key differential characteristic for the full 31-round PRESENT-80 is at most 2^{-72}, which leads to the conclusion that the workload of related-key differential attack on the full-round PRESENT-80 is close to that of an exhaustive search.

The paper is organized as follows. In Section 2 we recall the Mixed Integer Linear Programming and its applications to analysing word-oriented block ciphers. In Section 3 we extend Mouha et al's method to block ciphers of S-bP structures with bitwise permutation diffusion layers. We apply our method to the block cipher PRESENT-80 in Section 4. Section 5 is the discussion and conclusion.

2 Mixed-Integer Linear Prgoramming (MILP) and Mouha et al's Method

Mixed integer linear programming is an optimization method that tries to minimize or maximize a linear objective function of several variables subjected to certain linear constraints on the variables. An MILP problem can be formally stated as follows.

MILP : Given $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c_1, \cdots, c_n \in \mathbb{R}^n$, find an $x \in \mathbb{Z}^k \times \mathbb{R}^{n-k} \subseteq \mathbb{R}^n$ with $Ax \leq b$, such that the linear function $c_1x_1 + c_2x_2 + \cdots + c_nx_n$ is minimized (or maximized) with respect to the linear constraint $Ax \leq b$.

This kind of problems arises in many areas and the study of linear programming can be traced back, at least, to World War II [19]. However, it is only in recent years that MILP was applied in cryptographic research.

In [10], Borghoff at el devised a general method to transform the problem of solving a system of quadratic equations over $\mathbb{F}_2$ into a mixed-integer linear programming problem. With this method, the authors of [10] were able to recover the internal state of the stream cipher Bivium A within 4.5 hours. The same method was also employed in [2] to analyze polynomial systems with noises arising in the context of cold boot key recovery attacks [16]. In [11], Bulygin and Walter investigated the invariant coset attack on PrintCipher [20] by finding invariant projected subsets with techniques of mixed integer linear programming. A technique of MILP was also employed in optimizing the guessing strategies for algebraic attack on EPCBC [25].

Mouha et al [23] and Wu et al [26] applied MILP to automatically determine lower bounds of the numbers of active S-boxes for some word-oriented symmetric-key ciphers. In the following we give a description of Mouha et al's method introduced in [23].

Mouha et al's method uses 0-1 variables to describe the word-level differentials propagating through r rounds of the cipher. These variables are subjected to constraints imposed by the specific operations and structures of the cipher under consideration. Assume a block cipher consists of the following three operations:

1. S-box, $\mathcal{S} : \mathbb{F}_2^\omega \to \mathbb{F}_2^\omega$;
2. XOR, $\oplus : \mathbb{F}_2^\omega \times \mathbb{F}_2^\omega \to \mathbb{F}_2^\omega$; and
3. Linear transformation $L : \mathbb{F}_{2^\omega}^m \to \mathbb{F}_{2^\omega}^m$. The branch number of L is defined as

$$\mathcal{B}_L = \min_{a \neq 0} \{wt(a||L(a)) : a \in \mathbb{F}_{2^\omega}^m\}$$

where $wt(a||L(a))$ is the number of non-zero entries of the $2m$-dimensional vector $a||L(a) \in \mathbb{F}_{2^\omega}^{2m}$.

Representation of active S-boxes and objective function

For an input difference $\Delta_i \in \mathbb{F}_{2^\omega}$ of each S-box appearing in the schematic diagram of the cipher, Mouha et al introduced a new 0-1 variable A_i to describe the corresponding S-box is active or not, i.e., $A_i = 1$ or $A_i = 0$ depending on $\Delta_i \neq 0$ or $\Delta_i = 0$. Then, the total number of active S-boxes, $\sum_i A_i$, is chosen as the objective function to be minimized subjecting to constraints imposed by the operations of the cipher.

Constraints imposed by XOR operations

Assume that $a,\ b \in \mathbb{F}_2^\omega$ are the input differences of the XOR operation, and $c \in \mathbb{F}_2^\omega$ is the output difference. Then we have

$$\begin{cases} a + b + c \geq 2d_\oplus \\ d_\oplus \geq a \\ d_\oplus \geq b \\ d_\oplus \geq c \end{cases} \tag{1}$$

where $d_\oplus$ is a dummy variable taking values from $\{0, 1\}$.

Constraints imposed by linear transformation

Suppose $\{i_0, \cdots, i_{m-1}\}$ and $\{j_0, \cdots, j_{m-1}\}$ are permutations of $\{0, \cdots, m-1\}$. Let x_{i_k} and y_{j_k}, $k \in \{0, 1, \ldots, m-1\}$, be 0-1 variables to denote the word-level input and output differences respectively for a linear transformation depicted in

Figure 1. Then these variables are subjected to the following constraints

$$\begin{cases} \sum_{k=0}^{m-1} (x_{i_k} + y_{j_k}) \geq \mathcal{B}_L d_L \\ d_L \geq x_{i_0} \\ \cdots \\ d_L \geq x_{i_{m-1}} \\ d_L \geq y_{j_0} \\ \cdots \\ d_L \geq y_{j_{m-1}} \end{cases} \tag{2}$$

where d_L is a dummy variable taking values in $\{0, 1\}$ and $\mathcal{B}_L$ is the branch number of the linear transformation L.

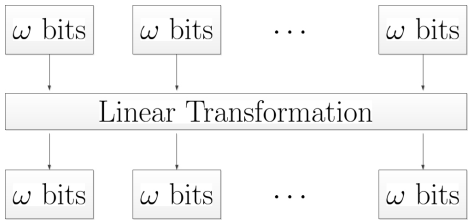

Fig. 1: Linear transformation $L : \mathbb{F}_{2^\omega}^m \to \mathbb{F}_{2^\omega}^m$

With the objective function and constraints presented as above, the problem of calculating a lower bound of the number of active S-boxes is modelled as an MILP instance which can be solved by the CPLEX [12] optimizer. The minimum numbers of active S-boxes were obtained in [23] for r-round Enocoro-128V2 ($r \leq 96$) and full-round AES. We refer the reader to [23] for more information.

These results are impressive especially for that Mouha et al's method is able to show the resistance of AES against related-key differential attacks automatically. However, Mouha et al's method is not applicable to SPN ciphers with bitwise permutation diffusion layers since it does not consider the collaborative diffusion effect of the S-box layer and bitwise permutation linear diffusion layer. In the next section, we will extend Mouha et al's method by introducing new representations linking bit-level and word-level differentials and adding new constraints concerning the diffusion effect of S-boxes to make it suitable to SPN ciphers with bitwise permutation diffusion layers.

3 Calculating the Minimum Number of Active S-boxes for S-bP Structures

In this section, we consider an r-round SP block cipher with n-bit block size, $\omega \times \omega$ S-box, and a bitwise permutation diffusion layer. We call this is a block cipher of

S-bP(n, ω, r) structure. Under this notation, PRESENT-80 is an S-bP$(64, 4, 31)$ structure, PRINTCIPHER is an S-bP$(32, 3, 48)$ structure, and EPCBC(48,96) is an S-bP$(96, 4, 32)$ structure.

Each round of an S-bP(n, ω, r) structure consists of a key addition (XOR) layer, a substitution layer where the n input bits are divided into n/ω words which will be substituted by new ones according to the underlying S-boxes, and a bitwise permutation layer that permutes the position of the output bits of the substitution layer. See Figure 5 and Figure 2 for an example of block cipher of S-bP$(64, 4, 31)$ structure.

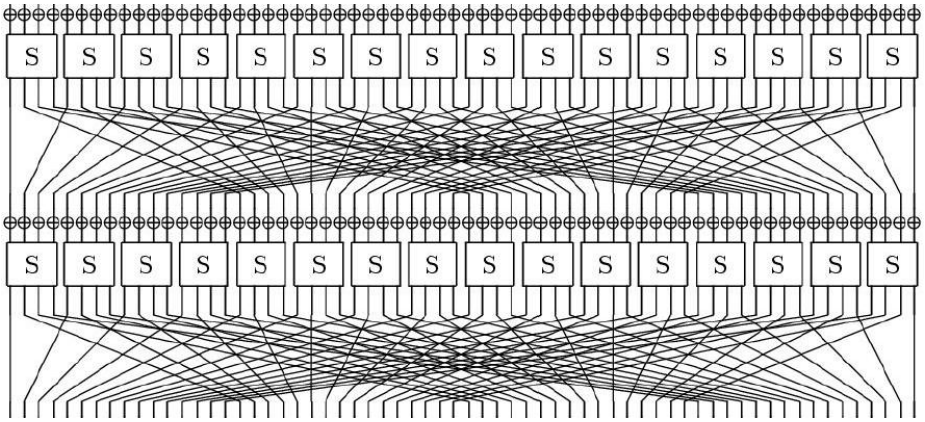

Fig. 2: Two consecutive rounds of the PRESENT-80 encryption algorithm

3.1 Representation of the Differentials

Bit-level representation. For every bit-level difference in S-bP structure, we introduce a new 0-1 variable to denote it if necessary. For differences that can be represented by variables already introduced (e.g., the r-th round input difference is the bitwise permutation of the $(r-1)$-th round output difference in single-key differential analysis), we do not introduce new variables. The reason is that we should make the number of variables as small as possible in the resulting MILP instance.

Word-level representation. For every S-box in the schematic diagram (including the encryption process and the key schedule algorithm) of the block cipher, we introduce a new 0-1 variable A_j.

3.2 Constructing the MILP instance for S-bP structure

If we follow the way of variable usage introduced in Subsection 3.1 and obey the rules of variable assignment as follows:

$$x_i = \begin{cases} 0, \text{ there is no bit level difference at this position,} \\ 1, \text{ otherwise,} \end{cases}$$

$$A_i = \begin{cases} 0, \text{ the Sbox marked by } A_i \text{ is not active,} \\ 1, \text{ otherwise,} \end{cases}$$

then it is natural to choose the objective function f as $\sum A_j$, which will be minimized to determine the lower bound of the number of active S-boxes for S-bP(n, ω, r) structure. The tricky part is to pinpoint the constraints under which the objective function f should be minimized.

Constraints imposed by XOR operations. For every XOR operation that may receive more than one nonzero input difference, we add the constraints (1) presented in Section 2, here the corresponding input and output variable should be changed to bit-level representation.

Some XOR operations may be ignored if they do not affect the output difference. To illustrate this, we analyze the round function of the block cipher MIBS [17] depicted by Figure 3. When we consider active S-boxes in the single-key model (in contrast to the related-key model), we only take care of the XORs to the right of the S-box layer, since in the single-key model, every subkey XOR operation introduces no difference or receive at most one nonzero input difference. While all XORs will be taken into account in the related-key model.

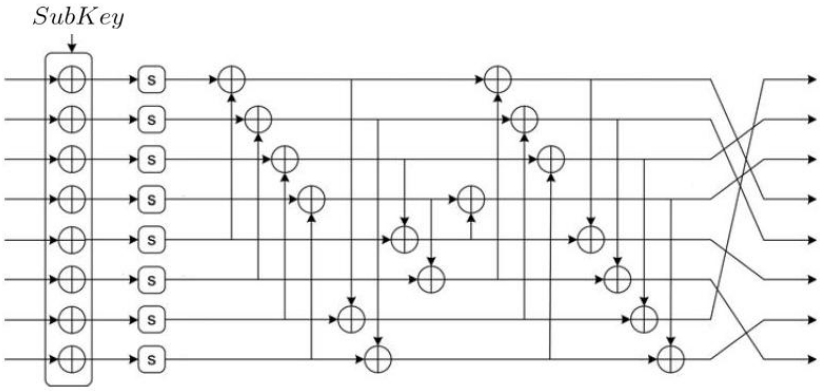

Fig. 3: XOR operations in the MIBS block cipher

Constraints describing the S-box operation. Assume $(x_{i_0}, \cdots, x_{i_{\omega-1}})$ and $(y_{j_0}, \ldots, y_{j_{\omega-1}})$ are the input and output bits of an S-box marked by A_t respectively. Firstly, to ensure $A_t = 1$ when any one of $x_{i_0}, \ldots, x_{i_{\omega-1}}$ is 1, we require

$$\begin{cases} x_{i_0} - A_t \leq 0 \\ x_{i_1} - A_t \leq 0 \\ \cdots \\ x_{i_{\omega-1}} - A_t \leq 0 \end{cases} \tag{3}$$

Secondly, when $A_t = 1$, one of $x_{i_0}, \ldots, x_{i_{\omega-1}}$ must be 1:

$$x_{i_0} + x_{i_1} + \cdots + x_{i_{\omega-1}} - A_t \geq 0 \tag{4}$$

Thirdly, input difference must result in output difference and vice versa:

$$\begin{cases} \omega y_{j_0} + \omega y_{j_1} + \cdots + \omega y_{j_{\omega-1}} - (x_{i_0} + x_{i_1} + \cdots + x_{i_{\omega-1}}) \geq 0 \\ \omega x_{i_0} + \omega x_{i_1} + \cdots + \omega x_{i_{\omega-1}} - (y_{j_0} + y_{j_1} + \cdots + y_{j_{\omega-1}}) \geq 0 \end{cases} \tag{5}$$

Here we stress that similar constraints must be added for invertible linear transformation $L : \mathbb{F}_{2^\omega}^m \to \mathbb{F}_{2^\omega}^m$ with branch number $\mathcal{B}_L < \omega + 1$. For example, the block cipher PRINCE in [9] applies an almost-MDS linear diffusion layer L with $\mathcal{B}_L = \omega$.

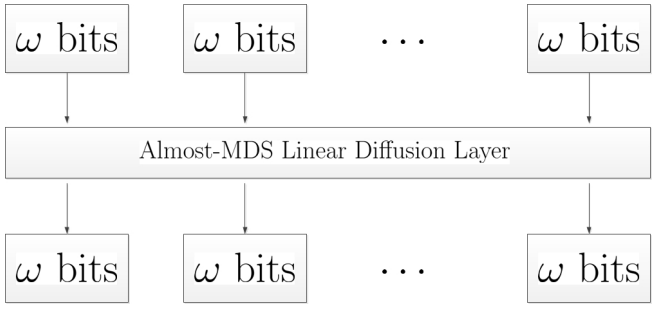

Fig. 4: Almost MDS linear diffusion layer

In Mouha et al's method, the variables representing input and output differences of a linear diffusion transformation are subjected to (2). It is easy to check that the following assignment

$$\begin{cases} d_L = 1 \\ x_{i_0} = 1 \\ \vdots \\ x_{i_{m-1}} = 1 \\ y_{j_0} = 0 \\ \vdots \\ y_{j_{m-1}} = 0 \end{cases}$$

does not violate (2) if $\mathcal{B}_L < \omega + 1$. However, this contradicts the invertibility of L since a nonzero input difference must result in a nonzero output difference. This defect can be remedied by adding (5) as additional constraints.

Finally, since a single active S-box may lead to more than one active S-box in the next round in S-bP structure, the collaborative diffusion effect of the S-boxes and bitwise permutations can not be ignored.

Definition 1. *The branch number $\mathcal{B}_\mathcal{S}$ of an $\omega \times \omega$ S-box $\mathcal{S} : \mathbb{F}_2^\omega \to \mathbb{F}_2^\omega$ is defined as follows*

$$\mathcal{B}_\mathcal{S} = \min_{a \neq b} \{wt((a \oplus b)||(\mathcal{S}(a) \oplus \mathcal{S}(b)) : a, b \in \mathbb{F}_2^\omega\}$$

where $wt(\cdot)$ is the Hamming weight of a 2ω-bit word.

Similarly to the constraints describing the diffusion effect of linear transformations in Mouha et al's method, we have

$$\begin{cases} \sum_{k=0}^{\omega-1} (x_{i_k} + x_{j_k}) \geq \mathcal{B}_\mathcal{S} d_t \\ d_t \geq x_{i_0} \\ \cdots \\ d_t \geq x_{i_{\omega-1}} \\ d_t \geq y_{j_0} \\ \cdots \\ d_t \geq y_{j_{\omega-1}} \end{cases} \tag{6}$$

Additional constraints. Add an extra constraint to ensure nonzero input difference to rule out the trivial result where zero input difference results in 0 active S-box. Let $(x_1, \cdots, x_n)$ be the input difference, we require a constraint that $x_1 + \cdots + x_n \geq 1$.

0-1 Variables vs mixed-integer linear programming. If we restrict all variables appearing in the objective function and constraints to be 0-1, the resulting instance is a pure integer programming problem. In practice, as suggested in [10], we only require all variables representing differences of plaintexts and all dummy variables to be 0-1 whilst other variables are only required to be real numbers, this may lead to a faster solving process.

4 Applications to the Block Cipher PRESENT-80

The increasing popularity of small computing devices with restrictive cost, power and size makes it a crucial task to design lightweight block ciphers. However, designing a secure lightweight block cipher suitable for extremely constrained devices is still a challenging goal.

Some designers employ the well understood SPN structure to meet the lightweight requirement with smaller S-boxes and bitwise permutation diffusion layers, both of which can be implemented in hardware with very low cost. For example, PRESENT [7] and EPCBC [28] use 4×4 S-boxes, and PrintCipher [20] uses 3×3 S-boxes. All these schemes have bitwise permutation diffusion layers. It is remarkable that the PRESENT cipher has become an international standard for lightweight cryptography. Hence, it is of great importance to evaluate the security of S-bP structures.

4.1 Description of the PRESENT-80 Cipher

PRESENT-80 is an SPN block cipher with 31 rounds. The block size and key length are 64 bits and 80 bits respectively. Its top-level algorithmic description is depicted in Figure 5. The S-box and permutation table in the sBoxLayer and pLayer are given in Tables 1 and 2 respectively.

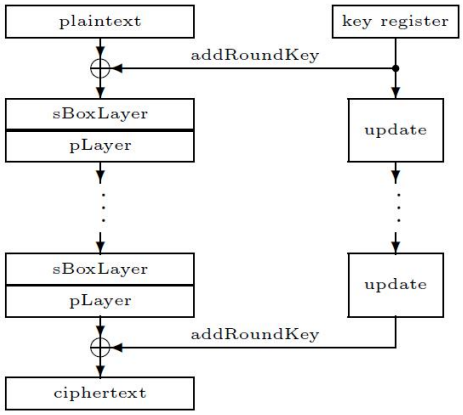

Fig. 5: A top-level description of PRESENT-80

Table 1: The S-box of PRESENT-80

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

The 64-bit subkeys are extracted from a 80-bit key register. Initially, the key register is filled with the 80-bit secret key of PRESENT-80, and then updated as depicted in Figure 6. We refer the reader to [7] for more information of the block cipher PRESENT-80.

Table 2: The bitwise permutation diffusion layer of PRESENT-80

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

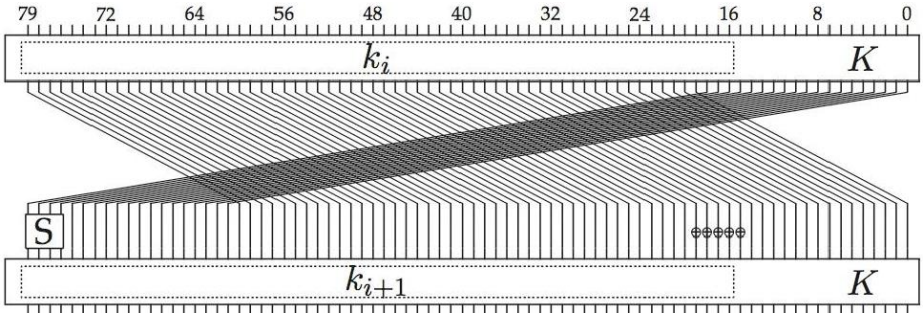

Fig. 6: The key schedule algorithm of PRESENT-80

4.2 Experimental Results for PRESENT-80

The numbers of differentially active S-boxes in the single-key and related-key model are summarized in Tables 3 and 4 respectively. The MILP instances were generated by a Python script and solved by the Gurobi5.5 optimizer running on a PC with Intel(R) Core(TM) Quad CPU (2.83GHz, 3.25GB RAM), and to make full use of the CPU, all computations were performed parallelly with four threads.

From Table 3 we know that the MILP instance corresponding to the full-round PRESENT-80 in the single-key model consists of 1056 0-1 variables, 1984 continuous variables, and 7937 constraints. This instance can be solved within 222 seconds and the number of active S-boxes is 62. Since the S-box of PRESENT-80 achieves a maximum probability of differentials 2^{-2}, the maximum probability for differentials of the PRESENT-80 cipher is roughly $(2^{-2})^{62} = 2^{-124}$, which is less than 2^{-80}, the probability of success for an exhaustive search, thus, we have proved that PRESENT-80 is secure against single-key differential attack.

For PRESENT-80 in the related-key differential attack, we are only able to obtain the results for its round-reduced version up to 12 rounds within a reasonable timing, and the results are listed in Table 4. For example, the probability of the best related-key differential characteristics for 7-round and 12-round PRESENT-80 are upper bounded by $(2^{-2})^6$, and $(2^{-2})^{15}$ respectively.

From these results, the probability of the best related-key differential characteristic for full 31-round PRESENT-80 is upper bounded by $(2^{-2})^{15+15+6} = 2^{-72}$. Although this is slightly larger than the probability of success for an exhaustive search, we conjecture that the actual minimum number of active S-boxes is greater than 40. How to reduce the gap and completely prove the security of the full-round PRESENT-80 against related-key differential attack is still an open question.

Table 3: Results for the single-key differential analysis

Rounds	#Variables	#Constraints	#Active S-boxes	Timing (in seconds)
1	$96 + 64$	257	1	1
2	$128 + 128$	513	2	1
3	$160 + 192$	769	4	1
4	$192 + 256$	1025	6	1
5	$224 + 320$	1281	10	1
6	$256 + 384$	1537	12	1
7	$288 + 448$	1739	14	2
8	$320 + 512$	2049	16	5
9	$352 + 576$	2305	18	3
10	$384 + 640$	2561	20	6
11	$416 + 704$	2817	22	14
12	$448 + 768$	3073	24	13
13	$480 + 832$	3329	26	14
14	$512 + 896$	3585	28	17
15	$544 + 960$	3841	30	22
16	$576 + 1024$	4097	32	27
17	$608 + 1088$	4353	34	35
18	$640 + 1152$	4609	36	33
19	$672 + 1216$	4865	38	46
20	$704 + 1280$	5121	40	39
21	$736 + 1344$	5377	42	43
22	$768 + 1408$	5633	44	82
23	$800 + 1472$	5889	46	69
24	$832 + 1536$	6145	48	88
25	$864 + 1600$	6401	50	107
26	$896 + 1664$	6657	52	105
27	$928 + 1728$	6913	54	116
28	$960 + 1792$	7169	56	140
29	$992 + 1856$	7425	58	165
30	$1024 + 1920$	7681	60	262
31	$1056 + 1984$	7937	62	222

Table 4: Results for related-key differential analysis

Rounds	#Variables	#Constraints	#Active S-boxes	Timing (in seconds)
1	97+277	530	0	1
2	130+474	1058	0	1
3	163+671	1586	1	1
4	196+868	2114	2	1
5	229+1065	2642	3	3
6	262+1262	3170	4	10
7	295+1459	3698	6	26
8	328+1656	4226	8	111
9	361+1853	4754	9	171
10	394+2050	5282	12	1540
11	427+2247	5810	13	8136
12	460+2444	6338	15	18102
13	493+2641	8192	–	> 5 days

5 Conclusion and discussion

In this paper, we extended Mouha et al's method and propose an approach for automatically computing a lower bound on the number of active S-boxes for block ciphers with S-bP structures based on mixed-integer linear programming (MILP). We applied this method to the PRESENT-80 block cipher and successfully obtained the minimal numbers of active S-boxes in any single-key differential characteristic for the full-round PRESENT-80, and any related-key differential characteristic for its round reduced versions. We proved that PRESENT-80 is secure against the single-key differential attack, and that the cost of related-key differential attack against the full-round PRESENT-80 is close to the cost of an exhaustive search.

Finally, we would like to mention some related topics that deserve further investigation:

1. Completely prove the security of the full-round PRESENT-80 with respect to the related-key differential attack. A direct approach is to solve the MILP instance generated from the 31-round PRESENT-80 in the related-key model. However, according to our experiment, we are not even able to solve the MILP instance corresponding to 13-round PRESENT-80 within 5 days.
2. The MILP instances generated from cryptographic problems are in general very hard to solve compared to usual MILP instances coming from other fields. To practically solve the MILP instances derived from the full-round PRESENT-80 against related-key differential attack, it is an interesting research topic to develop methods to utilize specific structures in the MILP instances generated from cryptography and speed up the solving process.

References

1. Achterberg, T.: Scip–a framework to integrate constraint and mixed integer programming, report 04-19, zuse institute berlin, 2004
2. Albrecht, M., Cid, C.: Cold boot key recovery by solving polynomial systems with noise. In: Applied Cryptography and Network Security. pp. 57–72. Springer (2011)
3. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platformsdesign andanalysis. In: Selected Areas in Cryptography. pp. 39–56. Springer (2001)
4. Biham, E.: New types of cryptanalytic attacks using related keys. Journal of Cryptology 7(4), 229–246 (1994)
5. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Advances in CryptologyEurocrypt99. pp. 12–23. Springer (1999)
6. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. Journal of CRYPTOLOGY 4(1), 3–72 (1991)
7. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y., Vikkelsoe, C.: Present: An ultra-lightweight block cipher. In: Cryptographic Hardware and Embedded Systems-CHES 2007, pp. 450–466. Springer (2007)
8. Bogdanov, A., Rijmen, V.: Zero correlation linear cryptanalysis of block ciphers. IACR Eprint Archive Report 123 (2011)
9. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., et al.: Prince–a low-latency block cipher for pervasive computing applications. In: Advances in Cryptology–ASIACRYPT 2012, pp. 208–225. Springer (2012)
10. Borghoff, J., Knudsen, L.R., Stolpe, M.: Bivium as a mixed-integer linear programming problem. In: Cryptography and Coding, pp. 133–152. Springer (2009)
11. Bulygin, S., Walter, M.: Study of the invariant coset attack on printcipher: more weak keys with practical key recovery. Tech. rep., Cryptology eprint Archive, Report 2012/85 (2012)
12. CPLEX, I.I.: Ibm software group. User-Manual CPLEX 12 (2011)
13. Daemen, J., Rijmen, V.: The wide trail strategy. In: The Design of Rijndael, pp. 123–147. Springer (2002)
14. Daemen, J., Rijmen, V., Proposal, A.: Rijndael. In: Proceedings from the First Advanced Encryption Standard Candidate Conference, National Institute of Standards and Technology (NIST) (1998)
15. Gurobi: Gurobi optimizer reference manual. URL: http://www. gurobi. com (2012)
16. Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: cold-boot attacks on encryption keys. Communications of the ACM 52(5), 91–98 (2009)
17. Izadi, M., Sadeghiyan, B., Sadeghian, S.S., Khanooki, H.A.: Mibs: a new lightweight block cipher. In: Cryptology and Network Security, pp. 334–348. Springer (2009)
18. Kanda, M.: Practical security evaluation against differential and linear cryptanalyses for feistel ciphers with spn round function. In: Selected Areas in Cryptography. pp. 324–338. Springer (2001)
19. Kantorovich, L.V.: A new method of solving some classes of extremal problems. In: Doklady Akad Sci USSR. vol. 28, pp. 211–214 (1940)

20. Knudsen, L., Leander, G., Poschmann, A., Robshaw, M.J.: Printcipher: a block cipher for ic-printing. In: Cryptographic Hardware and Embedded Systems, CHES 2010, pp. 16–32. Springer (2010)
21. Matsui, M.: Linear cryptanalysis method for des cipher. In: Advances in CryptologyEUROCRYPT93. pp. 386–397. Springer (1994)
22. Matsui, M.: Differential path search of the block cipher e2. ISEC99-19 pp. 57–64 (1999)
23. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Information Security and Cryptology. pp. 57–76. Springer (2012)
24. Shibutani, K.: On the diffusion of generalized feistel structures regarding differential and linear cryptanalysis. In: Selected Areas in Cryptography. pp. 211–228. Springer (2011)
25. Walter, M., Bulygin, S., Buchmann, J.: Optimizing guessing strategies for algebraic cryptanalysis with applications to epcbc. In: The 8th China International Conference on Information Security and Cryptology (Inscrypt 2012). Springer (2012)
26. Wu, S., Wang, M.: Security evaluation against differential cryptanalysis for block cipher structures. Tech. rep., Cryptology ePrint Archive, Report 2011/551 (2011)
27. Wu, W., Zhang, W., Lin, D.: On the security of generalized feistel scheme with sp round function. International Journal of Network Security 3(3), 215–224 (2006)
28. Yap, H., Khoo, K., Poschmann, A., Henricksen, M.: Epcbc-a block cipher suitable for electronic product code encryption. In: Cryptology and Network Security, pp. 76–97. Springer (2011)
29. Zhang, M., Liu, J., Wang, X.: The upper bounds on differential characteristics in block cipher sms4