More accurate differential properties of LED64 and Midori64

Sun, Ling; Wang, Wei; Wang, Meiqin

2018

Sun, L., Wang, W., & Wang, M. (2018). More accurate differential properties of LED64 and Midori64. IACR Transactions on Symmetric Cryptology, 2018(3), 93-123. doi:10.13154/tosc.v2018.i3.93-123

https://hdl.handle.net/10356/104634

https://doi.org/10.13154/tosc.v2018.i3.93-123

More Accurate Differential Properties of LED64 and Midori64

Ling Sun[1,2], Wei Wang[3,1] and Meiqin Wang[1](✉)

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, China
[2] School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
[3] School of Software, Shandong University, China
lingsun@mail.sdu.edu.cn, weiwangsdu@sdu.edu.cn, mqwang@sdu.edu.cn

Abstract. In differential cryptanalysis, a differential is more valuable than the single trail belonging to it in general. The traditional way to compute the probability of the differential is to sum the probabilities of all trails within it. The automatic tool for the search of differentials based on Mixed Integer Linear Programming (MILP) has been proposed and realises the task of finding multiple trails of a given differential. The problem is whether it is reliable to evaluate the probability of the differential traditionally. In this paper, we focus on two lightweight block ciphers – LED64 and Midori64 and show the more accurate estimation of differential probability considering the key schedule. Firstly, an automated tool based on Boolean Satisfiability Problem (SAT) is put forward to accomplish the automatic search of differentials for ciphers with S-boxes and is applied to LED64 and Midori64. Secondly, we provide an automatic approach to detect the right pairs following a given differential, which can be exploited to calculate the differential property. Applying this technique to the STEP function of LED64, we discover some differentials with enhanced probability. As a result, the previous attacks relying upon high probability differentials can be improved definitely. Thirdly, we present a method to compute an upper-bound of the weak-key ratio for a given differential, which is utilised to analyse 4-round differentials of Midori64. We detect two differentials whose weak-key ratios are much lower than the expected 50%. More than 78% of the keys will make these two differentials being impossible differentials. The idea of the estimation for an upper-bound of the weak-key ratio can be employed for other ciphers and allows us to launch differential attacks more reliably. Finally, we introduce how to compute the enhanced differential probability and evaluate the size of keys achieving the improved probability. Such a property may incur an efficient weak-key attack. For a 4-round differential of Midori64, we obtain an improved differential property for a portion of keys.

Keywords: Differential · Automatic search · SAT · LED64 · Midori64

1 Introduction

Differential cryptanalysis [BS90] is one of the most fundamental techniques targeting symmetric-key primitives. It investigates how an input difference propagates through the objective function. If a particular input/output difference happens in a non-random way, it can be used to construct a distinguisher or even to recover keys.

Since the introduction of differential cryptanalysis, many investigations concentrated on achieving provable security against it. Among these works, the Markov cipher theory [LMM91] is regarded as the first attempt to design block ciphers resistant against differential cryptanalysis. It has inspired many works on iterative block ciphers. Since the complexity of an attack depends on the differential probability (DP) of the differential exploited,

IACR Transactions on Symmetric Cryptology ISSN 2519-173X, Vol. 2018, No. 3, pp. 93–123
DOI:10.13154/tosc.v2018.i3.93-123

which in general relies on the value of the key, evaluating the distribution of the fixed-key probability becomes a crucial issue to realise the goal of provable security. With the hypothesis of stochastic equivalence, which claims that for almost all keys the average probability of a differential is equal to the fixed-key probability, we are allowed to construct the bound on the expected data complexity of the differential attack where the attacker uses precisely one differential. However, Daemen and Rijmen [DR07b] pointed out that for key-alternating ciphers, a more precise formulation of the distribution can be derived. They proposed an explicit expression of the distribution of the fixed-key DP for a differential concerning its expected differential probability (EDP).

Apart from the theoretical research, another strong research trend in the field of differential cryptanalysis is the construction of the automatic tool for searching differential characteristics or differentials [SHW+14, KLT15, SHY16, SGL+17, AST+17]. Nevertheless, most of these techniques pay attention to the seek of differential trails instead of differentials. The few pieces of research related to the differential searching problem mainly target ARX ciphers, and they cannot be applied to block ciphers with S-box directly, for the lack of a model on S-box. Although the tool[1] presented in [KLT15] supports primitives with S-boxes, it is based on Satisfiability Modulo Theories (SMT) instead of SAT. Many modern (lightweight) block ciphers built upon Substitution-Permutation Network (SPN) are designed to avoid the existence of the dominating trail and thus increase resistance against differential cryptanalysis.

With an automatic tool for the search of differentials, we are allowed to obtain multiple trails within a differential. The traditional way to compute the probability of the differential is to sum the probabilities of all trails within it, and this probability is adopted in the differential attack. The problem is whether it is reliable to evaluate the probability of the differential in a traditional manner. In this paper, we focus on two lightweight block ciphers - LED64 [GPPR11, GPPR12] and Midori64 [BBI+15] and show the more accurate estimation of differential probability considering the key schedule. The ideas proposed in this paper permit us to launch differential attacks more reliably.

1.1 Our Contributions

An automatic method for the search of differentials Since we target the analysis of the differential, an automatic tool for the search of differentials for ciphers with S-boxes is proposed firstly. Inspired by the previous works targeting ARX ciphers, we put forward an automated method based on SAT to realise the automatic search for ciphers with S-boxes. Owing to the invention of this method, we can search for differentials with numerous characteristics for LED64 and Midori64. We remark that the previous works related to the automatic search of differentials depending on MILP [SHW+14] and SMT [KLT15] only concentrate on the seek of multiple characteristics, and evaluate the probability of the differential as the sum of the probabilities of all characteristics within it. However, our aim is not only to obtain multiple trails of a differential but also to get a more accurate differential probability. Because the SAT solver assists the computation of the more accurate differential probability, we also use SAT to construct the automatic tool for the search of differentials for consistency.

Improved differentials of the STEP function of LED64 Since the STEP function of LED is an unkeyed permutation, the design of LED can be seen as a special case of the generalised Even-Mansour construction [EM97]. Therefore, many attacks regard the STEP function as a public mapping and stem from some attacks on the Even-Mansour construction [MRTV12, NWW15]. Some of these attacks rely on the existence of differential with high probability. In this paper, we aim to seek differentials with high probability. With plenty

[1] https://github.com/kste/cryptosmt

of trails found by the differential searching tool, we construct an automatic approach to detect the right pairs following a given differential, which can be exploited to calculate the differential property. Firstly, we derive the constraints on the right pairs following a given characteristic. Then, these constraints are converted into SAT clauses, and the solutions of these clauses are the right pairs matching that characteristic. Finally, the right pairs of a differential can be obtained by gathering all right pairs of trails within the differential. With this technique, we improve the previous results provided by Mendel et al. [MRTV12]. In particular, under the same setting as [MRTV12], we obtain an iterative differential with 66 right pairs, which updates the one with six right pairs [MRTV12]. For the non-iterative differential, the number of right pairs raises from around 2^{10} to 2^{15}. Benefiting from the increased differential probability, the previous attacks exploiting the differential property of the `STEP` function can be improved.

4-round differentials of `Midori64` with unexpectedly low weak-key ratios The distribution of fixed-key probability is an essential problem in differential cryptanalysis because it determines the validity of the differential distinguisher exploited in the attack. From the precise distribution proposed by Daemen and Rijmen [DR07b], we can infer that when the value of EDP is far from 2^{-n}, the keys, for which the fixed-key probability is higher than the expected differential probability, make up about 50% of the whole keyspace. Since the differential attacks with the fixed-keys falling into this space are more likely to succeed, we call these keys the weak-keys. For a given key schedule, the real weak-key ratio may deviate from 50%. To evaluate the deviation, we present a method to compute an upper-bound of the weak-key ratio for a given differential, which is utilised to analyse 4-round differentials of `Midori64`. We detect two differentials whose weak-key ratios are much lower than the expected 50%. More than 78% of the keys will make these two differentials being impossible differentials. If such a differential is used to launch a key-recovery attack, the attack is very likely to fail since the attacker cannot find right pairs under a vast amount of right keys. These results force us to pay more attention to the validity of the differential distinguisher. Moreover, the idea of the estimation for an upper-bound of the weak-key ratio can be employed for other ciphers.

Key subspace with enhanced differential probability In differential cryptanalysis, another interesting problem is that a differential distinguisher with much higher probability than the value of EDP under a fraction of keys can incur an efficient weak-key attack. Here, we aim at identifying such high probability differential distinguisher. After introducing the notion of partially expected differential probability ($\mathrm{EDP_P}$) to replace EDP when the upper-bound for the weak-key ratio is known, we concentrate on the problem of the maximum number of compatible characteristics in a differential. Since the compatible trails hold simultaneously under a particular set of keys, for those keys we may detect the enhancement on the differential probability. We find that this problem can be converted into a special kind of Max-PoSSo problem [AC11], which intends to find a solution that satisfies the maximum number of polynomials in a given set of polynomials $\mathcal{F}$. We put forward a technique based on SAT to solve the dedicate Max-PoSSo problem derived from the question we care about. As a result, for a 4-round differential of `Midori64` with EDP $= 2^{-23.79}$, we observe that for 2^{-12} of the keys, the differential property is increased to 2^{-16}. Note that the disadvantage of our technique lies in that it is not easy to be applied to more rounds due to the limited computational resource. Although the enhanced differential probability for the short-round differential can also be predicted under statistical method, our method is much more efficient than the random test according to the runtime.

Paper Outline The paper is organised as follows. Section 2 recalls some definitions and theories that will be useful to understand the remaining contents. Section 3 provides an automatic search method based on SAT problem to search for differentials instead of characteristics. Section 4 concentrates on the property of the weak-key space of a differential/trail. The differential analyses of LED64 and Midori64 are introduced in Section 5 and 6, respectively.

2 Preliminaries

2.1 Differential Cryptanalysis

Typically, the first step of differential cryptanalysis is to analyse the smallest nonlinear component of the function, i.e., the S-box. An S-box is called *active* if it has non-zero input difference; otherwise, we call it *passive*.

An r-round *differential characteristic/trail* $C = (C_0, C_1, \ldots, C_r)$ is a sequence of differences through various operations of the encryption. A *differential* [LMM91] over a mapping f consists of an input difference α and an output difference β and is denoted by (α, β). The *differential probability* (DP) of a differential (α, β) over an n-bit function f is computed as

$$\mathrm{DP}_f(\alpha, \beta) = \frac{\{x \in \mathbb{F}_2^n \mid f(x) \oplus f(x \oplus \alpha) = \beta\}}{2^n}.$$

For a keyed function $f(\cdot, k)$, we define differential probabilities $\mathrm{DP}_f[k](\alpha, \beta)$ and $\mathrm{DP}_f[k](C)$ for each value k of the key. Then, the *expected differential probability* (EDP) of a characteristic or a differential is the average DP of that characteristic or differential over all keys. The *weight* of a differential or a trail is the opposite number of the binary logarithm of its EDP. The *height* [MRTV12] of a possible differential or a possible trail is the binary logarithm of the number of pairs following that differential or trail, and we call these pairs the *right pairs*. The *cardinality* [DR07b] of a differential or a characteristic is the number of right pairs.

The *Markov cipher theory* [LMM91] was the first approach to design block ciphers resistant against differential cryptanalysis. The theory has inspired many pieces of research on iterative block ciphers. A *Markov cipher* is an iterative cipher for which the average differential probability over one round is independent of the input of the round function. For such ciphers, the assumption of independent round keys allows us to compute the EDP of a characteristic as the product of the probabilities of each round. The average probability of a differential can be computed as the sum of the probabilities of all characteristics sharing the same input and output differences with the differential.

Modern block ciphers are designed to resist differential cryptanalysis. In general, the release of a current block cipher is accompanied with provable security bound on the EDP of either differential or characteristic [DR02, Mat97, GPPR11, GPPR12, BBI+15]. For the classical block cipher DES [Nat77], it is well known that the EDP of a differential can be estimated by the EDP of the dominating trail. Many modern block ciphers are designed to withstand the existence of dominating trail; this estimation can no longer be adopted. The following hypothesis is often used as a replacement, and it allows one to construct proofs of security.

Hypothesis 1 (Hypothesis of stochastic equivalence [LMM91])**.** *For all differentials* (α, β)*, it holds that for most values of the key* k*,* $\mathrm{DP}_f[k](\alpha, \beta) = \mathrm{EDP}_f(\alpha, \beta)$.

Afterwards, Daemen and Rijmen [DR07b] proved that this hypothesis could be discarded for *key-alternating ciphers*, which are iterative ciphers whose round keys are applied by an XOR operation in between unkeyed round functions. More specifically, they put forward

the distribution of the fixed-key probability of differential, which is characterised by the value of EDP.

Theorem 1 (Theorem 13 [DR07b]). *In a key-alternating cipher $f(\cdot, k)$, the fixed-key cardinality $N_f[k](\alpha, \beta)$ of a differential (α, β) is a stochastic variable with the following distribution:*

$$\Pr(N_f[k](\alpha, \beta) = i) \approx \text{Poisson}(i; 2^{n-1}\,\text{EDP}(\alpha, \beta)),$$

where the distribution function measures the probability over all possible values of the key and all possible choices of the key schedule.

2.2 Planar Differentials and Maps

For the differential (α, β) of the function f, denote $F_f(\alpha, \beta)$, $G_f(\alpha, \beta)$ the sets that contain the input values, respectively the output values of the right pairs following the differential, i.e.,

$$\begin{aligned} F_f(\alpha, \beta) &= \{x \mid f(x) \oplus f(x \oplus \alpha) = \beta\}, \\ G_f(\alpha, \beta) &= \{y \mid y = f(x), x \in F_f(\alpha, \beta)\}. \end{aligned}$$

A differential (α, β) is called a *planar differential* if $F_f(\alpha, \beta)$ and $G_f(\alpha, \beta)$ are affine subspaces. In that case, we can uniquely construct two linear spaces $U_f(\alpha, \beta)$ and $V_f(\alpha, \beta)$, such that

$$\begin{aligned} F_f(\alpha, \beta) &= p \oplus U_f(\alpha, \beta), \\ G_f(\alpha, \beta) &= q \oplus V_f(\alpha, \beta), \end{aligned}$$

where p is an arbitrary element in $F_f(\alpha, \beta)$, and q is an arbitrary entry in $G_f(\alpha, \beta)$. A mapping is *planar* if all differentials over it are planar.

It has been proved [DR07a] that a differential (α, β) is always planar if it has exactly two or four right pairs. When an S-box S has differential uniformity [Nyb93] of 4, i.e., $\max_{\alpha \neq 0} |N_S(\alpha, \beta)| = 4$, all the sets $F_S(\alpha, \beta)$ and $G_S(\alpha, \beta)$ are affine subspaces. Let $y = f(x)$ be a function consisting of a set of parallel functions $y_i = f_i(x_i)$ with $x = (x_0, x_1, \ldots, x_{t-1})$ and $y = (y_0, y_1, \ldots, y_{t-1})$. It can be proved [DR07a] that a differential (α, β) for f is planar if every differential (α_i, β_i) for f_i is planar. Thus, the S-layer composed of the parallel applications of S-boxes is planar when all the S-boxes have differential uniformity of 4.

2.3 SAT Problem

The boolean satisfiability problem (SAT) considers the satisfiability of a given Boolean formula, i.e., it decides whether there exists a valid assignment of boolean values to the variables such that the formula is evaluated to be `True`. If such an assignment exists, the SAT problem is said *satisfiable*. It was shown that the problem is NP-complete [Coo71]. However, modern SAT solvers based on backtracking search can solve problems of practical interest with millions of variables. Since many issues covered in this paper involve `XOR` operations and the SAT solver called Cryptominisat5[2] is specially designed to be compatible with the `XOR` operation, we adopt it to solve our problems.

For every Boolean formula, there exists an equi-satisfiable formula in Conjunctive Normal Form (CNF), expressed as the conjunction ($\wedge$) of the disjunction ($\vee$) of (possibly negated) variables. Every conjunct of the Boolean formula in CNF is called a *clause*, and

[2] https://github.com/msoos/cryptominisat

each (possibly negated) variable within a clause is called a *literal*. Because all modern SAT solvers accept problems organised in CNF as a standard input format, we need to translate the question into an equivalent one in CNF when we want to invoke SAT solvers to solve the problem we care about.

3 Automatic Search of Differentials

There are many investigations on the automatic search of differential and linear characteristics [SHW+14, KLT15, LWR16, SHY16, SGL+17, AST+17]. The principle behind these automated tools is to transform the searching problem into some mathematical problems, which can be handled by some available solvers. Very few of the previous works concentrate on the differential searching problem. The practices related to this topic [KLT15, LWR16, SHY16] mainly target ARX ciphers. Even though the tool presented in [KLT15] supports primitives with S-boxes, it is based on SMT instead of SAT. Because we rely on the SAT solver to accomplish the computation of the more accurate differential property, we also use SAT to construct the automatic tool for the search of differentials for consistency.

To realise the automatic search of differentials, we need to model the problem of finding multiple solutions under the fixed input and output differences. Besides the compatibility of `XOR` operation, another reason we choose Cryptominisat5 lies in that it supports the usage of searching multiple solutions.

In this section, we propose a method based on SAT to accomplish the automatic search of differentials. First, we model a truncated version of the differential distribution table (DDT), which only contains information of all possible differential propagations of the S-box, and we demonstrate how to transform it into formulas in CNF. Then, we illustrate this method can be adapted to generate clauses for ordinary DDT by introducing variables representing the weight of the differential propagation. After that, the objective function is created so that we can search characteristics with fixed weight. Finally, we describe how to find multiple trails within a differential.

For all automatic tools for the search of differential/linear characteristics, the most critical step is to translate the propagation rule of these trails into the language of the ultimate mathematical problem. For the reason that we mainly focus on `LED64` and `Midori64`, which adopt similar round functions, we only need to construct SAT models for the S-layer composed of the parallel applications of small-scale S-boxes and the P-layer constituted by the parallel applications of MDS matrices.

3.1 Propagation of Differences Through P-layer

For an n-bit P-layer $y = P(x)$, we can always find its primitive representation [SLR+15], which is an $n \times n$ binary matrix M_P with $P(x) = M_P \cdot x$. Thanks to the compatibility of Cryptominisat5 with the `XOR` operation, the model for the P-layer can be constructed, directly. The following n `XOR` clauses are required

$$\bigoplus_{j=0}^{n-1} (M_P)_{i,j} \cdot x_j \oplus y_i = 0, \ i = 0, 1, \ldots n-1,$$

where $(M_P)_{i,j} \in \{0, 1\}$ stands for the entry located at the i-th row and the j-th column of M_P.

3.2 Propagation of Differences Through S-layer

First of all, a truncated version of the DDT, where all the non-zero entries of the DDT are replaced by 1, which is called $*-$DDT in [AST+17], is considered. The primary method

of the generation of clauses for $*-$DDT is very similar to the logical condition modelling method adopted in MILP [SHW+14]. Then, we aim to reduce the number of acquired clauses so that we can accelerate the solution finding phase. After that, we illustrate that this method can be generalised to model ordinary DDT.

3.2.1 Modelling $*-$DDT

To describe $*$-DDT, we ought to generate a set of clauses about the variables representing the input and output differences of the S-box. All solutions of these clauses have a one-to-one correspondence with all possible differential propagations.

For a c-bit S-box, suppose that $(x_0, x_1, \ldots, x_{c-1})$ and $(y_0, y_1, \ldots, y_{c-1})$ are variables standing for the input and output differences, respectively. Denote $(a_0, a_1, \ldots, a_{c-1}) \nrightarrow (b_0, b_1, \ldots, b_{c-1})$ an impossible propagation, where $a_i, b_i \in \{0, 1\}$. Then the formula

$$\bigvee_{i=0}^{c-1}(x_i \oplus a_i) \vee \bigvee_{i=0}^{c-1}(y_i \oplus b_i) = 1$$

excludes the impossible case since this equation fails when $x_i = a_i$ and $y_i = b_i$ hold for all i's. Besides, the left hand side (LHS) of this equation is a legitimate clause, because $x_i \oplus a_i$ equals to either x_i or $\overline{x_i}$ depending on the value of a_i. If there are η impossible propagations $(a_0^{(j)}, a_1^{(j)}, \ldots, a_{c-1}^{(j)}) \nrightarrow (b_0^{(j)}, b_1^{(j)}, \ldots, b_{c-1}^{(j)})$, $j = 0, 1, \ldots, \eta - 1$, in a DDT, the following η clauses remove all the impossible differentials,

$$\bigvee_{i=0}^{c-1}(x_i \oplus a_i^{(j)}) \vee \bigvee_{i=0}^{c-1}(y_i \oplus b_i^{(j)}) = 1,\ j = 0, 1, \ldots, \eta - 1. \tag{1}$$

In other words, η clauses are enough to depict the $*-$DDT, precisely.

Due to the fact that some clauses can be integrated, the number of clauses can be reduced. As an example, suppose that $(a_0, a_1, \ldots, a_{c-1}) \nrightarrow (b_0, b_1, \ldots, b_{c-1})$ and $(\overline{a_0}, a_1, \ldots, a_{c-1}) \nrightarrow (b_0, b_1, \ldots, b_{c-1})$ are two impossible differentials, adding the clause

$$\bigvee_{i=1}^{c-1}(x_i \oplus a_i) \vee \bigvee_{i=0}^{c-1}(y_i \oplus b_i) = 1$$

enables us to remove these two cases, simultaneously. Over the course of the trial, we detect that the runtime can be slightly reduced when we use fewer clauses. Thus, we aim to reduce the number of acquired clauses further.

Note that the solution space of the η equations about x_i's and y_i's in (1) is same as that of the following function:

$$f(\boldsymbol{x}\|\boldsymbol{y}) = \bigwedge_{j=0}^{\eta-1}\left(\bigvee_{i=0}^{c-1}(x_i \oplus a_i^{(j)}) \vee \bigvee_{i=0}^{c-1}(y_i \oplus b_i^{(j)})\right) = 1,$$

where $\boldsymbol{x} = (x_0, x_1, \ldots, x_{c-1})$, $\boldsymbol{y} = (y_0, y_1, \ldots, y_{c-1})$. Equivalently, we have

$$f(\boldsymbol{x}\|\boldsymbol{y}) = \bigwedge_{\boldsymbol{a}\|\boldsymbol{b}\in\mathbb{F}_2^{2c}}\left(f(\boldsymbol{a}\|\boldsymbol{b}) \vee \bigvee_{i=1}^{c-1}(x_i \oplus a_i) \vee \bigvee_{i=0}^{c-1}(y_i \oplus b_i)\right), \tag{2}$$

where $\boldsymbol{a} = (a_0, a_1, \ldots, a_{c-1})$, $\boldsymbol{b} = (b_0, b_1, \ldots, b_{c-1})$. Equ. (2) is called the *product-of-sum representation* of f. The issue of reducing the number of clauses is turned into the problem of simplifying the product-of-sum representation of the Boolean function. Inspired by [AST+17], we know that this simplification problem is a well-studied question in the

field of Boolean algebra, and can be solved by the Quine-McCluskey (QM) algorithm [Qui55, McC56] and Espresso algorithm [BHMSV84], theoretically. Although it is also an NP-complete problem, the small-scale problem can be implemented by some off-the-shelf software, such as Logic Friday[3]. After simplification, we obtain a relatively small set of clauses, which is employed to depict the $*-$DDT.

3.2.2 Modelling Ordinary DDT

Now, we turn to set up SAT model for the ordinary DDT, and we restrict ourselves to 4-bit S-box with differential uniformity 4[4]. Thus, the entries in the DDT only take four possible values, which are 0, 2, 4, and 16, and the differential probability belongs to the set $\{0, 2^{-3}, 2^{-2}, 1\}$. To cover the information of weight in the SAT model, we need to bring in some auxiliary variables. More specifically, we introduce three variables, which are denoted as p_0, p_1 and p_2, such that $p_0 + p_1 + p_2$ equals to the weight of the corresponding differential. Let us consider an 11-bit Boolean function $f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the 4-bit input and output differences, $\boldsymbol{p} = (p_0, p_1, p_2)$. The definition of f ties to the value of $\mathrm{DP}_S(\boldsymbol{x}, \boldsymbol{y})$:

$$
\begin{aligned}
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 0: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}) = 0;\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 2^{-3}: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}) = \begin{cases} 1 & \text{if } \boldsymbol{p} = (1,1,1)\\ 0 & \text{else}\end{cases};\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 2^{-2}: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}) = \begin{cases} 1 & \text{if } \boldsymbol{p} = (0,1,1)\\ 0 & \text{else}\end{cases};\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 1: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}) = \begin{cases} 1 & \text{if } \boldsymbol{p} = (0,0,0)\\ 0 & \text{else}\end{cases}.
\end{aligned}
$$

With this definition, the product-of-sum representation of $f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p})$ can be constructed easily. The simplified set of clauses can be obtained by invoking Logic Friday. The solutions of $f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}) = 1$ (or equivalently, the set of clauses) not only describe all possible differentials but also involve messages about the corresponding weight.

So, the SAT model characterising the differential propagation through different operations of the round function is established. In the following part, we discuss the SAT model of the objective function.

3.3 Objective Function

Assuming that we intend to search an r-round differential, and there are m parallel S-boxes for one round of encryption. For the j-th S-box of the i-th round, we import three variables $p_k^{(i,j)}$, $k = 0, 1, 2$, which are introduced in Section 3.2.2, to trace the message of the weight. The ultimate goal for the search of differentials demands searching multiple trails sharing the same input and output differences with the given differential. However, in this phase, we do not conduct an irregular search. We prefer to search trails with higher probabilities foremost for the reason that the characteristic with high probability potentially on average has more contribution to the probability of the differential, which is a common sense under the hypothesis of stochastic equivalence. Therefore, we need to put a restriction on the sum of weight variables $\sum_{i,j,k} p_k^{(i,j)}$. For simplicity, we use p_ξ to represent $p_k^{(i,j)}$, where $\xi = 3mi + 3j + k$, and denote the number of weight variables $3 \cdot r \cdot m$ as μ, then $\sum_{i,j,k} p_k^{(i,j)} = \sum_{\xi=0}^{\mu-1} p_\xi$.

[3] http://sontrak.com/

[4] The S-boxes of LED64 and Midori64 satisfy this restriction.

In the automatic search of linear characteristics for ARX ciphers [LWR16], the authors mentioned that addition over integers is an unnatural operation in SAT language. They employed an inequality version of cardinality constraint as the objective function. Accordingly, in our case, it is $\sum_{\xi=0}^{\mu-1} p_\xi \leqslant w$, where $w \geqslant 1$, which requests the solver to search trails with weight less than or equal to w. By applying sequential encoding method [Sin05], this constraint can be transformed into SAT problem in CNF. To be specific, new dummy variables $u_{i,j}$ $(0 \leqslant i \leqslant \mu-2, 0 \leqslant j \leqslant w-1)$ are introduced regarding the cardinality constraint $\sum_{\xi=0}^{\mu-1} p_\xi \leqslant w$, where $w \geqslant 1$, and the following clauses will return unsatisfiable when the cardinality is larger than w,

$$\begin{cases} \overline{p_0} \vee u_{0,0} = 1 \\ \overline{u_{0,j}} = 1 \\ \overline{p_i} \vee u_{i,0} = 1 \\ \overline{u_{i-1,0}} \vee u_{i,0} = 1 \\ \overline{p_i} \vee \overline{u_{i-1,j-1}} \vee u_{i,j} = 1 \\ \overline{u_{i-1,j}} \vee u_{i,j} = 1 \\ \overline{p_i} \vee \overline{u_{i-1,w-1}} = 1 \\ \overline{p_{\mu-1}} \vee \overline{u_{\mu-2,w-1}} = 1 \end{cases}, \tag{3}$$

where $1 \leqslant i \leqslant \mu-2$, $1 \leqslant j \leqslant w-1$.

In this paper, in addition to the objective function $\sum_{\xi=0}^{\mu-1} p_\xi \leqslant w$, we take $\sum_{\xi=0}^{\mu-1} p_\xi = w$ as another candidate for the objective function. Before we explain the reason, we reveal the essence of the usage of searching multiple solutions for SAT solvers. During the searching phase of multiple solutions, after obtaining one solution, we will add one clause, which bans the acquired solution from the solution space, into the original SAT problem so that the solver will not return that solution. However, this procedure cannot repeat indefinitely, because the scale of the SAT problem becomes larger and larger with the increasing number of solutions we already obtained. If the size of the solution space is extremely huge, the solver will come to a halt after finding a reasonable amount of solutions. That is to say, the number of solutions handled by the solver is limited, which constitutes the motivation we add the equality constraint as an optional objective function.

Let us consider an extreme situation. Note that, usually, with the growing value of w, the number of trails with weight w will increase. Assuming that the amount of trails with weight no more than w' goes beyond our computation power, while the size of the set composed of the trails with weight w' is still within touch. So in that way, the objective function $\sum_{\xi=0}^{\mu-1} p_\xi \leqslant w'$ disables us from obtaining the trails with weight w', while the equality constraint enables us to do that. Thus, we take the equality constraint as a candidate, when the original objective function is out of operation. Besides, we will select $\sum_{\xi=0}^{\mu-1} p_\xi = w$ if we only target the trails with weight w. Furthermore, the amount of solutions managed by the solver is determined by the individual SAT problem. According to our experience, 2^{32} is an upper-bound.

To convert the equality constraint into a SAT problem in CNF, we first note that it is equivalent to

$$\sum_{\xi=0}^{\mu-1} p_\xi \leqslant w \text{ and } \sum_{\xi=0}^{\mu-1} p_\xi \geqslant w.$$

The first constraint is same to the previous objective function, whose model is mentioned

above. Since $p_\xi \in \{0, 1\}$, for the second constraint, we have

$$\sum_{\xi=0}^{\mu-1} p_\xi \geqslant w \Leftrightarrow \mu - \sum_{\xi=0}^{\mu-1} p_\xi \leqslant \mu - w \Leftrightarrow \sum_{\xi=0}^{\mu-1} \overline{p_\xi} \leqslant \mu - w.$$

Thus, the model of the second constraint can be set up by interchanging p_i and $\overline{p_i}$ in (3) and introducing dummy variables as usual.

3.4 From Characteristic to Differential

Let $X_i = (X_{i,0}, X_{i,1}, \ldots, X_{i,n-1})$ and $Y_i = (Y_{i,0}, Y_{i,1}, \ldots, Y_{i,n-1})$ be the input and output differences of the i-th round. After obtaining an r-round characteristic $C = (C_0, C_1, \ldots, C_r)$, where $C_i = (C_{i,0}, C_{i,1}, \ldots, C_{i,n-1})$, we delete it from the solution space of the initial SAT problem. To be specific, the following clause is appended so that the SAT solver does not find this trail again

$$\bigvee_{i=0}^{r-1} \left(\bigvee_{j=0}^{n-1} (X_{i,j} \oplus C_{i,j}) \right) \vee \bigvee_{j=0}^{n-1} (Y_{r-1,j} \oplus C_{r,j}) = 1.$$

This procedure is repeated until the solver returns unsatisfiable, which indicates that there are no more solutions.

It is important to note that the method introduced in this section can be generalised to analyse other ciphers although we only apply it to LED64 and Midori64. The potential usages are provided in Supplementary Material A.

4 Weak-key Space of a Differential

In general, in differential cryptanalysis, the EDP of the differential is utilised to compute the complexity. However, Theorem 1 claims that for key-alternating ciphers, the fixed-key differential probability is lower than the EDP for a portion of the keys. For the set of keys with $N[k](\alpha, \beta) < 2^{n-1}\mathrm{EDP}(\alpha, \beta)$, the differential attacks are more likely to fail. Because when we launch a key-recovery attack for these keys, it is relatively hard to obtain the same amount of right pairs for the right key comparing to the cases where $N[k](\alpha, \beta) \geqslant 2^{n-1}\mathrm{EDP}(\alpha, \beta)$. In this sense, we call the keys fulfilling $N[k](\alpha, \beta) \geqslant 2^{n-1}\mathrm{EDP}(\alpha, \beta)$ the *weak-keys* for the differential (α, β), and the set of weak-keys is denoted as $W_K(\alpha, \beta)$. By Theorem 1, the weak-key ratio is

$$\sum_{i=2^{n-1}\mathrm{EDP}(\alpha,\beta)}^{\infty} \mathrm{Poisson}(i; 2^{n-1}\mathrm{EDP}(\alpha, \beta)),$$

where the probability is measured over all possible values of the key and all possible choices of the key schedule.

It is well known that the normal distribution with mean λ and variance λ is an excellent approximation to the Poisson distribution with parameter λ when λ is sufficiently large (say, $\lambda > 1000$)[DS12]. Thus, when $2^{n-1}\mathrm{EDP}(\alpha, \beta) > 1000$, the weak-key ratio is about 50% measured over all possible selections of the key schedule.

Now, we restrict ourselves to key-alternating cipher taking SPN as its iterative function. We intend to derive a necessary condition for a key being a weak-key and an upper-bound of the weak-key ratio for a given differential. Moreover, we suppose that the S-layer is a planar mapping, and we remind the readers that the S-layers of LED64 and Midori64 meet this condition since they both use S-boxes with differential uniformity 4.

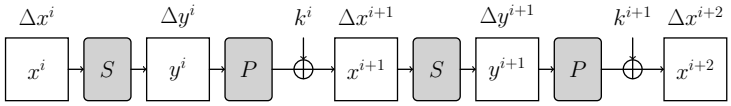

Figure 1: Two consecutive rounds of the key-alternating cipher.

Let us consider two consecutive rounds. For an n-bit cipher, denote Δx^i and Δy^i the input and output differences of the S-layer, x^i and y^i the input and output values of the S-layer, and k^i the i-th subkey. Please see Figure 1. Since we assume that the S-layer is planar, the input and output values of the differential $(\Delta x^i, \Delta y^i)$ constitute affine spaces. Thus, for the input and output spaces $F_S(\Delta x^i, \Delta y^i)$ and $G_S(\Delta x^i, \Delta y^i)$, we can construct two $l^i \times n$ matrices Mat_F^i and Mat_G^i, and two l^i-bit vectors Vec_F^i and Vec_G^i such that

$$x^i \in F_S(\Delta x^i, \Delta y^i) \text{ if and only if } \mathrm{Mat}_F^i \cdot x^i = \mathrm{Vec}_F^i,$$
$$y^i \in G_S(\Delta x^i, \Delta y^i) \text{ if and only if } \mathrm{Mat}_G^i \cdot y^i = \mathrm{Vec}_G^i.$$

Because $x^{i+1} = P \cdot y^i \oplus k^i$, we have

$$\mathrm{Mat}_F^{i+1} \cdot x^{i+1} = \mathrm{Mat}_F^{i+1} \cdot \left(P \cdot y^i \oplus k^i\right) = \mathrm{Mat}_F^{i+1} \cdot P \cdot y^i \oplus \mathrm{Mat}_F^{i+1} \cdot k^i = \mathrm{Vec}_F^{i+1}. \quad (4)$$

Then, the $2n$-bit vector $y^i \| k^i$ must satisfy the following condition:

$$\left[\begin{array}{c|c} \mathrm{Mat}_G^i & 0 \\ \hline \mathrm{Mat}_F^{i+1} \cdot P & \mathrm{Mat}_F^{i+1} \end{array}\right] \cdot \left[\begin{array}{c} y^i \\ \hline k^i \end{array}\right] = \left[\begin{array}{c} \mathrm{Vec}_G^i \\ \hline \mathrm{Vec}_F^{i+1} \end{array}\right]. \quad (5)$$

After simplifying this equation with Gaussian Elimination, $y^i \| k^i$ fulfils a linear relation of the following form

$$\left[\begin{array}{c} \mathrm{Mat}_U^i \\ \hline 0 \quad \mathrm{Mat}_K^i \end{array}\right] \cdot \left[\begin{array}{c} y^i \\ \hline k^i \end{array}\right] = \left[\begin{array}{c} \mathrm{Vec}_U^i \\ \hline \mathrm{Vec}_K^i \end{array}\right]. \quad (6)$$

Mat_K^i denotes an n-column binary matrix. For each row of Mat_U^i, there is at least one non-zero entry among the first n columns.

One necessary condition to ensure the solvability of Equ. (6) is that the equation $\mathrm{Mat}_K^i \cdot k^i = \mathrm{Vec}_K^i$ has at least one solution. That is, only the i-th subkey k^i falls into the affine space $\left\{x \middle| \mathrm{Mat}_K^i \cdot x = \mathrm{Vec}_K^i\right\}$, the corresponding 2-round differential characteristic may possess right pairs. Otherwise, it is an impossible characteristic. Thus, for an r-round trail, we can deduce $(r-1)$ spaces for the $(r-1)$ intermediate subkeys, respectively. The trail may become a possible one only if all intermediate subkeys fall into the corresponding subspaces. For an r-round differential consisting of m characteristics, if a particular key leads all m characteristics to become impossible trails, the differential under this fixed-key turns into an impossible differential. For the differential (α, β), we denote the set of these keys as $I_K(\alpha, \beta)$, which satisfies $W_K(\alpha, \beta) \subseteq \mathcal{K} - I_K(\alpha, \beta)$, where $\mathcal{K}$ represents the whole keyspace. Therefore, a key may become a weak-key only if it does not belong to $I_K(\alpha, \beta)$, which constitutes a necessary condition for a key being a weak-key.

Suppose that the cardinality of the set $I_K(\alpha, \beta)$ is $p \cdot |\mathcal{K}|$, where $|\mathcal{K}|$ is the size of $\mathcal{K}$, the possibility that a key is a weak-key is lower than $1 - p$, which is an upper-bound for the weak-key ratio.

5 Differential Analysis of the LED64 Block Cipher

LED [GPPR11, GPPR12] is a lightweight block cipher proposed by Guo et al. at CHES 2011. LED uses a block size of 64 bits and LED64 is a version with 64-bit key. A 4×4 array

represents the 64-bit plaintext, and the encryption is composed of eight STEP functions preceded by a key addition for LED64. LED64 has a simple key schedule, and the user key K is used as it is in each round.

The STEP function is an AES-like design composed of four rounds, where the addition of subkeys is replaced with the addition of constants. All the STEP functions can be seen as public permutations and differ only in the round constants they use. The design of LED can be regarded as a special case of the generalised Even-Mansour construction [EM97]. We denote the i-th STEP function as F_i, $0 \leqslant i \leqslant 7$. Each of these four rounds uses, in sequence, the operations AddConstants(AC), SubCells(SC), ShiftRows(SR), and MixColumnsSerial(MC). Please find in the Supplementary Material B.1 a detailed description of the STEP function.

Note that in the submission version to CHES 2011 [GPPR11], the constant matrix adopted in AddConstants operation only depends on the number of rounds. Later, the authors issued a new version [GPPR12] with minor modification in the constant matrix. The new round constant matrix relies on the key size as well. To distinguish these two cases, we call the version at CHES 2011 the old version of LED64 and write it as LED64-old for short. Correspondingly, the latest version is called the new version of LED64, and we denote it as LED64-new.

In this section, we aim at searching for right pairs of a given differential for the STEP function, so that some attacks targeting LED64 and counting on the differential property of the STEP function can be improved.

5.1 Previous Differential Attacks for LED64

Several distinguishing and key-recovery cryptanalyses on reduced versions of LED have been published [MRTV12, NWW15, IS12]. Some of these attacks rely on non-iterative or iterative differentials of the STEP function with high height. It was discussed in [MRTV12] that the 4-round differential characteristics of LED are not always plateau characteristic [DR07a]. Therefore, the *Two-Round Plateau Characteristic Theorem*, which is used to determine the number of right pairs following a given 2-round characteristic, cannot be applied directly. By extending this work with mega boxes [DLP+09], they proposed a specialised algorithm to find good differentials and right pairs for one step of LED. With such algorithm, they obtained a non-iterative differential with more than 2^{10} right pairs and an iterative differential with six right pairs for one step of LED64-old. However, they also claimed that both of these results might not be the best differentials regarding the probability of the STEP function of LED.

In [MRTV12], the authors put forward 3-STEP and 4-STEP related-key differential attacks for LED64. The 3-STEP attack is based on the assumption that one can detect a non-iterative differential with high probability p in F_i, $0 \leqslant i \leqslant 5$. Both the time and memory requirements of this attack are $2^{n/2} \cdot (1/p)^{1/2}$. On the other hand, the 4-STEP attack relies on an iterative differential of F_i, where $1 \leqslant i \leqslant 5$, with probability p'. Moreover, the complexity is about $2^{n/2} \cdot (1/p')^{1/2}$.

Afterwards, Nikolić et al. provided a 5-STEP chosen-key attack for LED64. The attacker managed to construct q-multicollisions with complexity less than the lower bound $q \cdot 2^{\frac{(q-2)}{(q+2)}n}$ demanded by an ideal permutation. With the help of a non-iterative differential of F_i, where $4 \leqslant i \leqslant 7$, with high probability p, this attack can be realised with complexity $(q/p)^{1/2} \cdot 2^{30.2}$.

Notice that the complexities of these attacks all tie to the probabilities of the exploited differentials. If differentials with better probability may exist and if such differentials are discovered, some attacks in [MRTV12, NWW15] will be improved immediately. Please find a detailed description of the three attacks mentioned above in Supplementary Material B.2.

5.2 Automatic Search for the Right Pairs of the STEP function

In this section, we focus on searching for differentials with better probability and propose an automatic method based on SAT to settle this problem. To begin with, we derive the constraints on the plaintexts constituting the right pairs. Then, we proceed to translate these constraints into SAT problem in CNF. After that, to get the desired right pairs, we call SAT solver and request it to return multiple solutions.

5.2.1 Constraints for the Right Pairs

Denote Δx^i and Δy^i the input and output differences of the SubCells operation in the i-th round. Let x^i and y^i be the input and output values of the i-th SubCells operation, and c^i stands for the round constant involved in the i-th round. For an illustration, please refer to Figure 2.

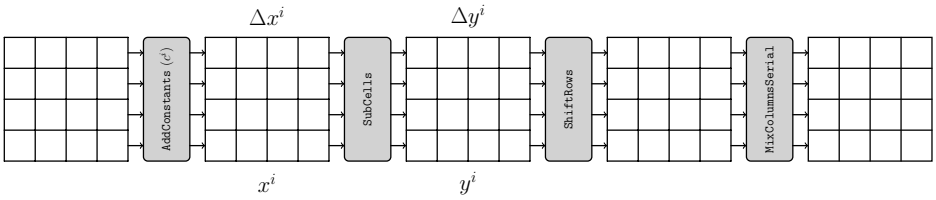

Figure 2: An illustration of the LED block cipher.

Since the STEP function does not comprise subkey, Equ. (4) is modified as

$$\mathrm{Mat}_F^{i+1} \cdot x^{i+1} = \mathrm{Mat}_F^{i+1}(P \cdot y^i \oplus c^{i+1}) = \mathrm{Mat}_F^{i+1} \cdot P \cdot y^i \oplus \mathrm{Mat}_F^{i+1} \cdot c^{i+1} = \mathrm{Vec}_F^{i+1},$$

where $P = \mathrm{MC} \circ \mathrm{SR}$ for the case of LED. Thus, the given differential characteristic only puts restrictions on the value of y^i. In other words, Equ. (5) is adapted to

$$\left[\begin{array}{c} \mathrm{Mat}_G^i \\ \hline \mathrm{Mat}_F^{i+1} \cdot P \end{array}\right] \cdot \left[\begin{array}{c} y^i \end{array}\right] = \left[\begin{array}{c} \mathrm{Vec}_G^i \\ \hline \mathrm{Vec}_F^{i+1} \oplus \mathrm{Mat}_F^{i+1} \cdot c^{i+1} \end{array}\right]. \tag{7}$$

Moreover, because the intermediate values of the right pair must follow the encryption rule, we have

$$y^i = \mathrm{SC}(x^i), \tag{8}$$
$$x^{i+1} = \mathrm{MC} \circ \mathrm{SR}(y^i) \oplus c^{i+1}. \tag{9}$$

Note that Equ. (7) - (9) entirely describe the plaintexts in the right pairs following the given trail, i.e., the solutions of these constraints have a one-to-one correspondence with the plaintexts constituting the right pairs of the given differential characteristic.

Since Equ. (8) is a non-linear constraint, we intend to transform the right pair searching problem into an equivalent SAT problem and invoke SAT solver to find out the right pairs.

5.2.2 Algorithm for the Right Pairs

Note that Equ. (7) and (9) are linear constraints, which can be expressed with matrices, and we include these constraints in the SAT problem by adding several XOR clauses. For Equ. (8), we apply the trick introduced in Section 3. We define an 8-bit Boolean function $f(\boldsymbol{x}\|\boldsymbol{y})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ refer to the input and output values of the S-box, respectively. f outputs one if and only if $\boldsymbol{y} = S(\boldsymbol{x})$. Then, the product-of-sum representation of f can be constructed, accordingly. A simplified set of clauses can be derived after calling Logic Friday. Also, we involve them in the SAT problem.

Now, all the constraints describing the right pairs are converted into clauses following CNF format. Thus, we can invoke SAT solver for the search of right pairs under a given differential characteristic. To get multiple right pairs following the given trail, after obtaining a plaintext belonging to a right pair, we ban it from the solution space of the original SAT problem. We repeat this step until the solver returns unsatisfiable.

Note that the above search is conducted under a fixed characteristic. To obtain the right pairs of a given differential, we first search for many trails within the differential. Then, we apply the automatic method to get right pairs for every characteristic, separately.

Although the choice of the constant c^i results in that the trail with high probability does not always possess right pairs, the probability of a characteristic indicates the number of its right pair on average. Thus, the search for the trails is conducted in a probability-first manner, i.e., we incline to search for trails with high probability firstly. With the growth of the weight, the number of trails may increase rapidly. However, we observe that the characteristic barely[5] has right pairs, when its weight is more than 80. Thus, for the characteristic searching phase mentioned in the remaining of this section, we only search for trails with probability higher than a certain threshold. In this sense, we do not claim that the search for the right pairs of the differentials in this paper is complete, we only ensure that the number of right pairs of a differential is at least equal to the amount we obtained.

5.3 Improved Differentials with Higher Height

5.3.1 Iterative Differentials

Note that the number of right pairs relates to the round constant matrix, which determines whether the output values of the i-th round `SubCell` operation and the input values of the $(i+1)$-th round `SubCell` operation can be connected with each other, as well as how many pairs may pass it. Since different `STEP` functions adopt different round constants, the numbers of right pairs of the same differential may vary for different `STEP` functions.

Mendel et al. [MRTV12] proposed a 4-round iterative differential with six right pairs for the first `STEP` function F_0 of `LED64-old`. To verify the correctness of our automatic algorithm, we apply it to the same setting. For the fixed input and output differences `0x6000c00070003000`, we first search for all differential characteristics with probability higher than 2^{-90}. There are 19 characteristics in total, and the probability distribution of these trails are listed in Table 1. Then, we search for the right pairs following each characteristic, the number of right pairs corresponding to the trails with a certain probability is summarised in Table 1. Moreover, we output the six plaintexts, with which six right pairs can be constructed. And the six right pairs are precisely same to those provided in [MRTV12].

Table 1: Probability distribution for the iterative differential `0x6000c00070003000`.

Probability	2^{-62}	2^{-63}	2^{-67}	2^{-68}	2^{-69}	2^{-70}	2^{-73}
#{Trails}	1	1	1	5	6	1	4
#{Right Pairs}	4	2	0	0	0	0	0

Besides, we apply the automatic method to F_0 of `LED64-new` and observe that the values of the right pairs change although the number of right pairs remains the same. The

[5] For many differentials of `LED64`, we search for all the trails with probability higher than 2^{-90}, but we cannot find any right pairs for all the trails with weight beyond 80.

three pairs are listed below:

$$\{\texttt{0x6ac6d235bedb9a2e}, \texttt{0x0ac61235cedbaa2e}\},$$
$$\{\texttt{0x00c61665c3999a43}, \texttt{0x60c6d665b399aa43}\},$$
$$\{\texttt{0x9cd618013e1da826}, \texttt{0xfcd6d8014e1d9826}\}.$$

The remaining three pairs are obtained by exchanging the order of the plaintexts in the above three pairs.

Moreover, we test many iterative differentials with the fixed differential pattern `0x*000*000*000*000` as well as unfixed patterns. The maximum numbers of right pairs for different `STEP` functions ($F_0 - F_7$) are summarised in Table 2. The concrete differentials with the maximum number of right pairs are provided in Supplementary Material B.3.

Table 2: The maximum number of right pairs for the iterative differential.

Pattern	Version	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
Fixed	$\max_{\texttt{old}}$	40	36	38	42	38	44	38	38
	$\max_{\texttt{new}}$	42	40	40	42	36	46	36	36
Unfixed	$\max_{\texttt{old}}$	66	64	62	68	58	82	90	90
	$\max_{\texttt{new}}$	62	64	76	70	54	82	68	84

Fixed: The differential pattern is fixed as `0x*000*000*000*000`.
Unfixed: The differential pattern is arbitrary.
$\max_{\texttt{old}}$: The maximum number of right pairs for `LED64-old`.
$\max_{\texttt{new}}$: The maximum number of right pairs for `LED64-new`.

5.3.2 Non-iterative Differentials

In [MRTV12], the authors found a 4-round differential with 1026 right pairs for F_0 of `LED64-old`. We apply the automatic method to search for right pairs under the same setting, and the number of right pairs is also 1026. We also test the right pairs of the same differential for `LED64-new` and observe that not only the values of right pairs vary but also the number of right pairs reduces. The differential only possesses 966 right pairs for F_0 of `LED64-new`.

Besides, we detect a differential has around 2^{15} right pairs for all `STEP` functions ($F_0 - F_7$) of the two versions, and the differential is

$$\texttt{0x0780003ba0007000} \rightarrow \texttt{0x36de4c3562a87eb7}.$$

The numbers of right pairs under different settings are summarised in Table 3. We do not claim that this differential achieves the highest height for four rounds of encryption, it is just the best one we obtained. Please refer to Supplementary Material B.4 for more details.

With these new differentials, the attack results mentioned in Section 5.1 can be improved. For the 3-`STEP` related-key attack utilising a non-iterative differential, the time and memory requirements are roughly $2^{n/2} \cdot (1/p)^{1/2}$. Since p is enhanced from around 2^{-54} to 2^{-49}, the complexity decreases by a factor of about $2^{2.5}$. The same amount of improvement is achieved for the 5-`STEP` chosen-key attack, whose complexity is $(q/p)^{1/2} \cdot 2^{30.2}$. In the case of the 4-`STEP` related-key attack making use of an iterative differential, the time and memory requirements are roughly $2^{n/2} \cdot (1/p')^{1/2}$. Notice that this attack demands to put the iterative differential at the third last `STEP` function among the targeted four `STEP`

Table 3: The maximum number of right pairs for the newly acquired differential.

Version	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7
$\max_{\texttt{old}}$	32636	33270	32680	32918	32640	32736	32602	32550
$\max_{\texttt{new}}$	32722	32822	32744	32434	32562	32910	33060	32812

$\max_{\texttt{old}}$: The maximum number of right pairs for LED64-old.
$\max_{\texttt{new}}$: The maximum number of right pairs for LED64-new.

functions. From Table 2, the maximum number of right pairs for both versions of LED64 is 82, when the position i of the STEP function is restricted with $1 \leqslant i \leqslant 5$. Thus, p' grows from $6/2^{64}$ to $82/2^{64}$, which results in the decrease in the complexity by a factor of about $2^{1.89}$.

6 Differentials of Midori64 Considering Key-schedule

It is pointed in Section 4 that for the differential of a key-alternating cipher, the weak-key ratio is about 50% in theory when $2^{n-1}\mathrm{EDP}(\alpha, \beta) > 1000$. In this section, we first analyse the weak-key ratio for some 4-round differentials of Midori64 and propose two counterexamples whose weak-key ratios are much lower than 50%. Besides, we investigate how many characteristics within a differential can hold simultaneously, because it may reflect the highest fixed-key cardinality of a differential (α, β). To handle this, we realise that this problem can be converted into a *Max-PoSSo* problem [AC11]. Then, we transform this particular Max-PoSSo problem into SAT problem and invoke SAT solver to settle it.

Midori64 [BBI+15] is a lightweight block cipher with 64-bit block and uses 128-bit keys. The round function of Midori64 is iterated 16 times and consists of an S-layer SubCell, a P-layer ShuffleCell and MixColumn and a key-addition layer KeyAdd. The key-schedule is linear. A 128-bit secret-key K is split into two 64-bit keys K_0 and K_1, i.e., $K = K_0 \| K_1$. Then, the i-th round key is generated as $RK_i = K_{(i \mod 2)} \oplus \alpha_i$, where α_i's are predetermined constant matrices. Please refer to Supplementary Material C for more information.

6.1 Upper-Bound for Weak-key Ratio of Differential

6.1.1 Estimating the Cardinality of the Weak-key Space

Suppose that the r-round differential (α, β) contains m characteristics denoted as $C^{(j)} = (C_0^{(j)}, C_1^{(j)}, \ldots, C_r^{(j)})$, where $C_0^{(j)} = \alpha$, $C_r^{(j)} = \beta$, and $0 \leqslant j \leqslant m-1$. Let x^i and y^i be the input and output values of the i-th SubCell operation. Please refer to Figure 3 as an illustration, where P represents the composition of ShuffleCell and MixColumn operations.

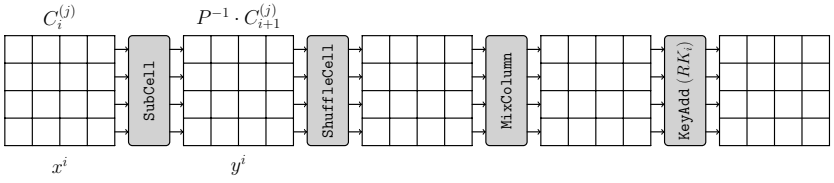

Figure 3: An illustration for the Midori64 block cipher.

We prove in Section 4 that for each trail $C^{(j)}$, $(r-1)$ affine subspaces describing the

$(r-1)$ internal subkeys can be derived. Only if all subkeys respectively fall into these subspaces, the corresponding characteristic may become a possible one. Due to the linear structure of the key schedule, for `Midori64`, these small spaces can be transformed into an affine subspace $V_K^{(j)}$ of the 128-bit keyspace $\mathcal{K} = \mathbb{F}_2^{128}$. That is, for K lies in the set $V_K^{(j)}$, the j-th trail may possess right pairs. By the definition of weak-key space $W_K(\alpha, \beta)$, it must follow $W_K(\alpha, \beta) \subseteq \bigcup_{j=0}^{m-1} V_K^{(j)}$, which indicates that

$$\Pr\{K \mid K \in W_K(\alpha,\beta)\} \leqslant \Pr\left\{K \middle| K \in \bigcup_{j=0}^{m-1} V_K^{(j)}\right\}. \tag{10}$$

Thus, $\Pr\left\{K \middle| K \in \bigcup_{j=0}^{m-1} V_K^{(j)}\right\}$ constitutes a natural upper-bound for the weak-key ratio.

To evaluate the cardinality of the set $\bigcup_{j=0}^{m-1} V_K^{(j)}$, we have to handle the union of a series of affine spaces. For small amount of affine spaces, we can make use of some techniques in the field of linear algebra. However, when the number of affine spaces is considerable, directly dealing with them is not easy because the union of the affine spaces is probably no longer an affine space. By De Morgan's laws, we know

$$\mathcal{K} - \bigcup_{j=0}^{m-1} V_K^{(j)} = \bigcap_{j=0}^{m-1} \left(\mathcal{K} - V_K^{(j)}\right).$$

Thus, estimating the quantitative characters of the set $\bigcup_{j=0}^{m-1} V_K^{(j)}$ is equivalent to evaluating those characters of the set $\bigcap_{j=0}^{m-1} \left(\mathcal{K} - V_K^{(j)}\right)$, which is the intersection of some complementary sets. And we find it is relatively convenient to deal with intersection operations.

To estimate the number of entries in $\bigcap_{j=0}^{m-1} \left(\mathcal{K} - V_K^{(j)}\right)$, we manage to convert the restrictions on the set into some clauses in CNF, and call SAT solver and ask it to return solutions satisfying the constraints. Note that translating the intersection operations is an easy task since the CNF is the conjunction of a series of clauses in nature. The remaining work is centred on how to use clauses, expressed as disjunctions of (possibly negated) variables, to describe the complement of an affine space.

For the affine space $V_K^{(j)}$, an $s(j) \times 128$ matrix $M^{(j)}$ and an $s(j)$-bit vector $V^{(i)}$ can be constructed such that

$$K = (k_0, k_1, \ldots k_{127}) \in V_K^{(j)} \text{ if and only if } M^{(j)} \cdot K = V^{(j)}. \tag{11}$$

Denote the l-th line of $M^{(j)}$ as $M_l^{(j)}$, and the l-th bit of $V^{(j)}$ as $V_l^{(j)}$. Then,

$$M^{(j)} \cdot K = V^{(j)} \text{ if and only if } \bigwedge_{l=0}^{s(j)-1} \left(1 \oplus M_l^{(j)} \cdot K \oplus V_l^{(j)}\right) = 1.$$

Equivalently, we have

$$V_K^{(j)} = \left\{K \middle| M^{(j)} \cdot K = V^{(j)}\right\} = \bigcap_{l=0}^{s(j)-1} \left\{K \middle| 1 \oplus M_l^{(j)} \cdot K \oplus V_l^{(j)} = 1\right\}.$$

By De Morgan's laws,

$$\mathcal{K} - V_K^{(j)} = \bigcup_{l=0}^{s(j)-1} \left\{ K \middle| M_l^{(j)} \cdot K \oplus V_l^{(j)} = 1 \right\}.$$

Thus,

$$K \in \mathcal{K} - V_K^{(j)} \text{ if and only if } \bigvee_{l=0}^{s(j)-1} \left(M_l^{(j)} \cdot K \oplus V_l^{(j)} \right) = 1. \tag{12}$$

Because $V_l^{(j)} \in \{0,1\}$, and $M_l^{(j)} \cdot K$ can be organised by an XOR clause about k_i's, $\bigvee_{l=0}^{s(j)-1} \left(M_l^{(j)} \cdot K \oplus V_l^{(j)} \right)$ satisfies the format of the clause. After adding m clauses matching m trails, the restrictions on the elements in $\mathcal{K} - \bigcup_{j=0}^{m-1} V_K^{(j)}$ are converted into SAT problem in CNF.

We can infer from Theorem 1 that the cardinalities of the set $\bigcup_{j=0}^{m-1} V_K^{(j)}$ and its complementary set are $\mathcal{O}(|\mathcal{K}|)$ roughly. However, SAT solvers have limited ability to search multiple solutions as we explained before. If the independent variables of the above SAT problem are set as the master key bits, say k_0, k_1, ..., k_{127}, the searching space roughly is $\varepsilon \cdot 2^{128}$, where ε is a non-negligible number. We definitely cannot afford the search.

To make the estimation feasible, we notice that some master key bits are not involved in any characteristic of a differential when the number of rounds is not so long. On the other hand, by observing Equ. (12), the composition $M_l^{(j)} \cdot K$ of the master key bits can be treated as independent variables when the number of the independent compositions is less than 128 notably, which holds when the number of rounds is small. To handle the dependencies between $M_l^{(j)} \cdot K$'s, we first combine all rows of the matrices $M^{(j)}$ into one 128-column matrix M. Then, we choose independent rows from M, which are denoted as M_0, M_1, ..., $M_{\ell-1}$, and set the independent variables of the SAT problems as x_0, x_1, ..., $x_{\ell-1}$, which respectively stand for $M_0 \cdot K$, $M_1 \cdot K$, ..., $M_{\ell-1} \cdot K$. Since the remaining vectors $M_l^{(j)}$ are linear combinations of M_0, M_1, ..., $M_{\ell-1}$, the corresponding variables $M_l^{(j)} \cdot K$ can be expressed by the XOR of x_0, x_1, ..., $x_{\ell-1}$. Please refer to Figure 4 for an illustration. Note that one solution for $x_0 \| x_1 \| \cdots \| x_{\ell-1}$ corresponds to $|\mathcal{K}|/2^\ell$ possible values for $k_0 \| k_1 \| \cdots \| k_{127}$ since a particular value of $x_0 \| x_1 \| \cdots \| x_{\ell-1}$ can be considered as putting ℓ linear limitations for the whole keyspace. For a reasonable amount of independent variables, if we can get ι solutions for $x_0 \| x_1 \| \cdots \| x_{\ell-1}$, the probability that a key falls into $\mathcal{K} - \bigcup_{i=0}^{m-1} V_K^{(i)}$ is $\iota/2^\ell$. Hence, with Equ. (10), the weak-key ratio is less than or equal to $1 - \iota/2^\ell$.

6.1.2 4-round Differentials with Weak-key Ratio Lower than 50%

We apply the above technique based on SAT to analyse some 4-round differentials of Midori64. In the following, we provide two examples. The weak-key ratio of one differential is less than 21.36%. For another one, the weak-key ratio is bounded by 3.94%, which means that for 96.06% of the key, the corresponding differential is an impossible differential although its EDP still enables us to utilise this differential to launch a key-recovery attack for the cipher in theory. These examples are meaningful since we seldom consider the probability that a distinguisher holds when we launch a differential attack. The distribution of the fixed-key DP affects the success probability of the attack. For a differential, if the

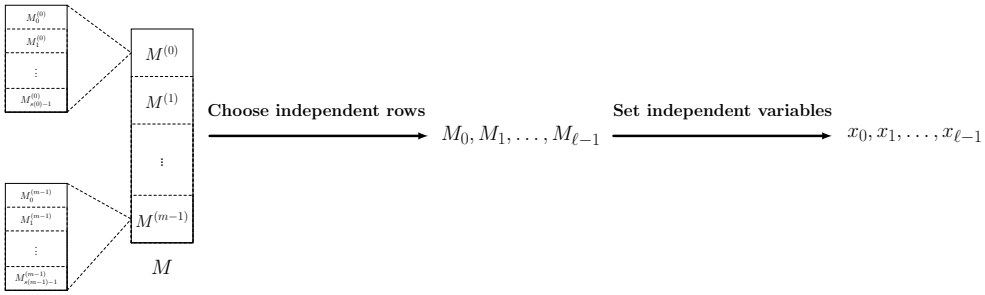

Figure 4: Independent variables selecting phase.

possibility of a key with fixed-key DP zero is very high, when we exploit this differential in an attack, we are more likely to regard the right key as a wrong key, since we cannot obtain any right pairs corresponding to it. At the same time, we will wrongfully accept a wrong key with right pairs as a right key. These examples remind us that for some lightweight block ciphers with a simple key schedule, or at least for `Midori64`, we need to pay attention to the effectiveness of the differential distinguisher itself.

The First Example The first differential is

$$\texttt{0x0022022202200202} \rightarrow \texttt{0x2220000022022022}.$$

We do not restrict the probability of the characteristic and search for all trails belonging to this differential. There are 896 characteristics in total, and the probability distribution of these trails is listed in Table 4. The EDP of this differential is $2^{-23.79}$. After analysing these trails, 16 independent variables are required to solve the estimation problem. We call SAT solver and obtain 51540 solutions for $x_0\|x_1\|\cdots\|x_{15}$. Therefore, $\Pr\left\{K\middle|K\in\mathcal{K}-\bigcup_{j=0}^{m-1}V_K^{(j)}\right\}=51540/2^{16}\approx 78.64\%$. Accordingly, the weak-key ratio for this differential is less than 21.36%, which is much lower than 50%.

To verify the validity of this probability, we test the number of right pairs for the randomly selected keys. The experimental results illustrate that the probability for a fixed-key with no right pair is about 78.66%, which is in accordance with our estimation. Moreover, if we adopt the distribution in Theorem 1 in this case, the possibility of a key with fixed-key probability zero is almost 0%, which does not match the experimental result.

Table 4: Probability distribution for the first differential of `Midori64`.

Probability	2^{-32}	2^{-36}
#{Trails}	256	640

The Second Example The second differential is

$$\texttt{0x7000000000a0000a} \rightarrow \texttt{0x5ffa05ff5faf00aa}.$$

In this case, we obtain $\Pr\left\{K\middle|K\in\mathcal{K}-\bigcup_{j=0}^{m-1}V_K^{(j)}\right\}\approx 96.06\%$. For 96.06% of the keys, the differential is an impossible one. If this differential is applied in a key-recovery attack, the attack will fail for most of the keys, since the attacker cannot find any right pair under the right key. This result is also examined with random tests.

6.2 Maximum Number of Compatible Characteristics in a Differential

In this section, we propose a method to estimate the maximum number of trails holding simultaneously. But before that, we consider how to compute the differential probability of the characteristic when every internal subkey falls into the corresponding affine space derived from the given trail.

6.2.1 Partially Expected Differential Probability

Note that the EDP of a characteristic is defined as the average differential probability of that characteristic over all keys. During the deduction of the cardinality of the weak-key space in Section 6.1, we point out that for each characteristic $C^{(j)}$, an affine subspace $V_K^{(j)}$ of the keyspace can be derived. The characteristic may possess right pairs only when K belongs to $V_K^{(j)}$. We define the *partially expected differential probability* ($\mathrm{EDP_P}(C^{(j)})$) of $C^{(j)}$ as the average differential probability of that characteristic over all keys in $V_K^{(j)}$, which is equal to

$$\mathrm{EDP_P}(C^{(j)}) = \mathrm{EDP}(C^{(j)}) \cdot \frac{\mathcal{K}}{|V_K^{(j)}|},$$

where $|V_K^{(j)}|$ stands for the size of $V_K^{(j)}$. Thus, we employ $\mathrm{EDP_P}(C^{(j)})$ as the expected differential probability of the characteristic when all intermediate keys satisfy the conditions imposed by the trail. We will see that this evaluation fits very well with the experimental result.

It is interesting to see that this definition has an association with the notion of *plateau characteristic*[DR07a], which is a particular type of characteristics whose probability for each value K of the key depends on the key and can have only two values. Nevertheless, since Midori64 takes almost-MDS matrix instead of MDS matrix as the diffusion layer, and its structure is different from that of AES, we cannot determine whether the 4-round trail is plateau characteristic.

6.2.2 Maximum Number of Compatible Characteristics & Max-PoSSo Problem

The affine spaces $V_K^{(j)}$'s derived from distinct trails may intersect with others. The keys belonging to the intersection set of some trails may result in an enhancement of the probability. The interesting problem is to investigate how many trails can hold simultaneously. Furthermore, we wonder the scale of the intersection set of $V_K^{(j)}$'s corresponding to these trails. The differential attack is more likely to succeed for the keys belonging to this intersection set comparing to the one targeting the keys out of the possible set derived from all the trails of the differential. Or rather, the differential attack can be realised with less data requirement in this case.

To begin with, we recall a closely related problem in the Cold Boot attack [AC11]. Let us consider the set $\mathcal{F} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), \ldots, f_{m-1}(\boldsymbol{x})\}$, where $f_i(\boldsymbol{x})$'s are polynomial functions over $\mathbb{F}_2^n$, $\boldsymbol{x} \in \mathbb{F}_2^n$. The *Max-PoSSo* problem is to find any $\boldsymbol{x} \in \mathbb{F}_2^n$ that satisfies the maximum number of polynomials in $\mathcal{F}$, i.e., search $\boldsymbol{x}$ such that the functions in $\mathcal{F}$ satisfy $f_i(\boldsymbol{x}) = 0$ as much as possible.

Then, we return to the problem of determining the maximum number of compatible characteristics in a differential. By the discussion in Section 6.1, the keys K in $V_K^{(j)}$ satisfies Equ. (11). That is, if $f_j(K)$ denotes $f_j(K) = M^{(j)} \cdot K \oplus V^{(j)}$, we know

$$K \in V_K^{(j)} \text{ if and only if } f_j(K) = 0.$$

Hence, the problem of determining the maximum number of compatible characteristics is equivalent to the issue of finding K under which the number of functions following $f_j(K) = 0$

is maximised. It is a Max-PoSSo problem over the set $\mathcal{F} = \{f_0(K), f_1(K), \ldots, f_{m-1}(K)\}$. There are many automatic methods based on MILP and SAT to solve this kind of problems [AC11, BCJ07]. To retain the consistency of this paper, we put forward a technique based on SAT problem to solve our Max-PoSSo problem.

The most critical step is to transform the Max-PoSSo problem into a SAT problem in CNF. To realise this goal, we introduce an auxiliary bit variable ζ_j for each of the polynomials $f_j(K)$'s. ζ_j can be regarded as the characteristic function of the set $\{K|f_j(K) = 0\}$, and it equals to 1 when $f_j(K) = 0$. Therefore, finding K maximising the number of f_j's following $f_j(K) = 0$ is equivalent to searching K maximising $\sum_{j=0}^{m-1} \zeta_j$. In other words, the objective function of this problem can be set as $\sum_{j=0}^{m-1} \zeta_j \geqslant t$, where t stands for a predetermined constant. We introduce how to convert this cardinality constraint into clauses in Section 3.3. The remaining task is to construct SAT model to depict the characteristic function ζ_j.

Following the symbols defined in Section 6.1, we have

$$\zeta_j = 1 \text{ if and only if } M_l^{(j)} \cdot K \oplus V_l^{(j)} = 0 \text{ for } l = 0, 1, \ldots, s(j) - 1.$$

We use $y_l^{(j)}$ to represent $M_l^{(j)} \cdot K \oplus V_l^{(j)}$ for short. The above condition is equivalent to restrict the value of the concatenation $y_0^{(j)} \| y_1^{(j)} \| \cdots y_{s(j)-1}^{(j)} \| \zeta_j$ of variables to either $\underbrace{0 \cdots 0}_{s(j)} 1$ or $u_0 \| u_1 \| \cdots \| u_{s(j)-1} \| 0$, where at least one u_l satisfies $u_l = 1$. Thus, all the impossible values, which are

$$\underbrace{0 \ldots 0}_{s(j)} 0,\ 1 \underbrace{* \ldots *}_{s(j)-1} 1,\ *1 \underbrace{* \ldots *}_{s(j)-2} 1,\ \ldots,\ \underbrace{* \ldots *}_{s(j)-1} 11,$$

for the concatenation are required to eliminate. To ban the $s(j)+1$ impossible values from the solutions for the concatenation, we construct the following $s(j) + 1$ clauses:

$$\begin{cases} \bigvee_{l=0}^{s(j)-1} y_l^{(j)} \vee \zeta_j = 1 \\ y_l^{(j)} \vee \overline{\zeta_j} = 1 \text{ for } l = 0, 1, \ldots, s(j) - 1 \end{cases}, \tag{13}$$

which constitute the SAT model depicting the characteristic function. Note that $y_l^{(j)}$ is just a symbolic representation of the real variables $M_l^j \cdot K$'s. Besides the variables ζ_j, we need to introduce variables for $M_l^j \cdot K$. The technique is similar to the one in Section 6.1, i.e., only independent variables among these $M_l^j \cdot K$'s are selected as the independent variables for the SAT problem and the remaining $M_l^j \cdot K$'s are represented as the `XOR` of these independent variables.

To sum up, the SAT problem for searching the maximum number of compatible characteristics is set up by adding m groups of clauses, which is in the form of Equ. (13), and involving the cardinality constraint on ζ_j. After that, we can invoke SAT solver to settle it.

6.2.3 Application

We apply this method to the first differential in Section 6.1. Firstly, we estimate the partially expected differential probability for the 896 characteristics, respectively. Among these trails, four trails have $\mathrm{EDP_P} = 2^{-22}$, 84 trails follow $\mathrm{EDP_P} = 2^{-23}$, 168 trails satisfy $\mathrm{EDP_P} = 2^{-24}$, and the $\mathrm{EDP_P}$'s of remaining 640 trails all equal to 2^{-25}.

Then, we investigate the maximum number of compatible trails. For the 896 trails, we find that the maximum number of compatible characteristics achieves 212. When fixing the objective function as $\sum_{j=0}^{m-1} \zeta_j \leqslant 212$ and searching for multiple solutions by banning the acquired solutions from the initial SAT problem, we find three groups composed of 212 trails, and the trails in each group hold simultaneously. Denote the three groups as $G^{(i)} = \left\{C_0^{(i)}, C_1^{(i)}, \ldots, C_{211}^{(i)}\right\}$, $i = 0, 1, 2$.

We analyse the three groups and observe that for all $G^{(i)}$'s, the values of $\mathrm{EDP_P}$ for 44 trails in the group are 2^{-23}, and the $\mathrm{EDP_P}$ values for the remaining 168 characteristics are 2^{-24}. Thus, for the keys falling into the intersection set of $V_K^{(j)}$'s derived from the 212 trails, which is represented as $W_K(G^{(i)})$[6], the corresponding differential probability is improved to 2^{-16}, while the EDP of the differential is $2^{-23.79}$.

Moreover, the scale of $W_K(G^{(i)})$ can be estimated. This task can be realised by selecting the augmented matrices $\left[M^{(j)}\middle|V^j\right]$'s corresponding to the trails $C_j^{(i)}$ in $G^{(i)}$ and combining these matrices into a large matrix $M(G^{(i)})$ first and then simplifying $M(G^{(i)})$ with Gaussian Elimination. Since these trails in the group are compatible trails, the linear equation system on K generated by $M(G^{(i)})$ must have solutions. Denote the rank of $M(G^{(i)})$ as $\mathrm{rank}(G^{(i)})$, the size of $W_K(G^{(i)})$ is $2^{128-\mathrm{rank}(G^{(i)})}$. For the three groups, the ranks $\mathrm{rank}(G^{(i)})$ are all equal to 15. On the other side, we must have $W_K(G^{(i)}) \cap W_K(G^{(j)}) = \emptyset$ if $i \neq j$, otherwise, $G^{(i)} \cup G^{(j)}$ will be the set with the maximum number of compatible characteristics. However, $\left|G^{(i)} \cup G^{(j)}\right| > 212$ results in a contradiction with the fact that the maximum number of compatible trails is 212.

We wonder whether the phenomenon, where the expected differential probability is improved to 2^{-16}, occurs for other groups containing less than 212 trails. Thus, we gradually decrease the value in the objective function. For the objective function $\sum_{j=0}^{m-1} \zeta_j \leqslant 64$, we obtain 16 groups in total. The information of these groups is summarised in Table 5. We exam the eight subspaces derived from the eight groups with no less than 208 compatible trails and find that arbitrary two of the subspaces do not result in an intersection. Hence, we obtain eight subspaces, on which the expected differential probability is improved to 2^{-16}. For a randomly drawn key, the possibility that the EDP of the differential under this key is no less than 2^{-16} is at least $2^{-15} \times 8 = 2^{-12}$, which means that on average for one of the 4096 keys, the EDP of the differential under this key is enhanced from $2^{-23.79}$ to 2^{-16}. On the contrary, by Theorem 1, it is almost impossible for a key with fixed-key probability 2^{-16} under the same setting.

Table 5: The information of groups with no less than 64 trails.

#{Trails}	212	211	208	128
#{Groups}	3	4	1	8
Rank	15	15	15	16
$\mathrm{EDP_P}$	2^{-16}	2^{-16}	2^{-16}	2^{-18}

To exam the correctness of this conclusion, we search for right pairs for 9280 randomly generated keys. Among the 9280 keys, the differential under 1994 keys has right pairs, and the distribution of the number of right pairs for the 1994 keys is illustrated in Figure 5. We find that the number of right pairs under two keys achieves about 2^{16}, and the probability

[6] We reuse the notation of weak-key space since these keys belong to weak-key space.

is about $2/9280 \approx 2^{-12.18}$, which is very close to the theoretical prediction 2^{-12}. Thus, the rightmost singular point in Figure 5 can be explained.

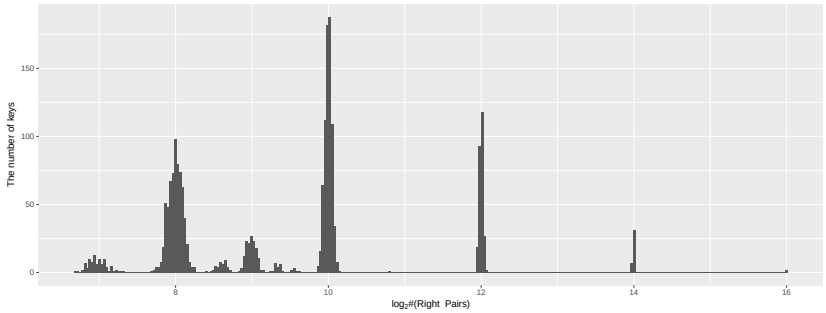

Figure 5: Distribution of the number of right pairs.

We remark that the techniques exploited to derive the theoretical conclusions are meaningful, although it seems that all the theoretical conclusions can be verified with random tests. On the one hand, the experimental results hold with a probability, and we cannot entirely rely on them to draw conclusions. Besides, as we mentioned before, the occurrences of some phenomena depend on a vast amount of tests. If we are not convinced of their existences, we do not know how many times we need to repeat until we observe them. On the other hand, the runtimes consumed by the theoretical deduction and the random tests are different. For the examples in this section, the complexity is around 2^{40}, and it takes us a few days to obtain the results. However, the theoretical conclusions are discovered within a few minutes, and they hold for sure.

Acknowledgements

The authors would like to thank Thomas Peyrin and the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. The research leading to these results has received funding from National Natural Science Foundation of China (Grant No. 61572293), Science and Technology on Communication Security Laboratory of China (Grant No. 9140c110207150c11050), Key Science Technology Project of Shandong Province (Grant No. 2015GGX101046), and Chinese Major Program of National Cryptography Development Foundation (Grant No. MMJJ20170102). Wei Wang is partially supported by the Open Research Fund from Shandong provincial Key Laboratory of Computer Network (Grant No. SDKLCN-2017-04).

References

[AC11] Martin R. Albrecht and Carlos Cid. Cold boot key recovery by solving polynomial systems with noise. In *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, pages 57–72, 2011.

[AST+17] Ahmed Abdelkhalek, Yu Sasaki, Yosuke Todo, Mohamed Tolba, and Amr M. Youssef. MILP modeling for (large) S-boxes to optimize probability of differential characteristics. *IACR Trans. Symmetric Cryptol.*, 2017(4):99–129, 2017.

[BBI+15] Subhadeep Banik, Andrey Bogdanov, Takanori Isobe, Kyoji Shibutani, Harunaga Hiwatari, Toru Akishita, and Francesco Regazzoni. Midori: A

block cipher for low energy. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 411–436, 2015.

[BCJ07] Gregory V. Bard, Nicolas Courtois, and Chris Jefferson. Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via sat-solvers. *IACR Cryptology ePrint Archive*, 2007:24, 2007.

[BHMSV84] Robert K Brayton, Gary D Hachtel, Curt McMullen, and Alberto Sangiovanni-Vincentelli. *Logic minimization algorithms for VLSI synthesis*, volume 2. Springer Science & Business Media, 1984.

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, pages 450–466, 2007.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and related-key attack on the full AES-256. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 231–249, 2009.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, pages 2–21, 1990.

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[DLP+09] Joan Daemen, Mario Lamberger, Norbert Pramstaller, Vincent Rijmen, and Frederik Vercauteren. Computational aspects of the expected differential probability of 4-round AES and AES-like ciphers. *Computing*, 85(1-2):85–104, 2009.

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

[DR07a] Joan Daemen and Vincent Rijmen. Plateau characteristics. *IET Information Security*, 1(1):11–17, 2007.

[DR07b] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *J. Mathematical Cryptology*, 1(3):221–242, 2007.

[DS12] Morris H DeGroot and Mark J Schervish. *Probability and statistics*. Pearson Education, 2012.

[EM97] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *J. Cryptology*, 10(3):151–162, 1997.

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pages 326–341, 2011.

[GPPR12] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. *IACR Cryptology ePrint Archive*, 2012:600, 2012.

[IS12] Takanori Isobe and Kyoji Shibutani. Security analysis of the lightweight block ciphers XTEA, LED and Piccolo. In *Information Security and Privacy - 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9-11, 2012. Proceedings*, pages 71–86, 2012.

[KLT15] Stefan Kölbl, Gregor Leander, and Tyge Tiessen. Observations on the SIMON block cipher family. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 161–185, 2015.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 17–38, 1991.

[LWR16] Yunwen Liu, Qingju Wang, and Vincent Rijmen. Automatic search of linear trails in ARX with applications to SPECK and Chaskey. In *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, pages 485–499, 2016.

[Mat97] Mitsuru Matsui. New block encryption algorithm MISTY. In *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, pages 54–68, 1997.

[McC56] Edward J McCluskey. Minimization of boolean functions. *Bell Labs Technical Journal*, 35(6):1417–1444, 1956.

[MRTV12] Florian Mendel, Vincent Rijmen, Deniz Toz, and Kerem Varici. Differential analysis of the LED block cipher. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 190–207, 2012.

[Nat77] National Bureau of Standards. Data encryption standard. *Federal information processing standards publication 46, US Department of Commerce*, 4, 1977.

[NWW15] Ivica Nikolic, Lei Wang, and Shuang Wu. Cryptanalysis of round-reduced LED. *IACR Cryptology ePrint Archive*, 2015:429, 2015.

[Nyb93] Kaisa Nyberg. Differentially uniform mappings for cryptography. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, pages 55–64, 1993.

[Qui55] Willard V Quine. A way to simplify truth functions. *The American Mathematical Monthly*, 62(9):627–631, 1955.

[SGL+17] Siwei Sun, David Gerault, Pascal Lafourcade, Qianqian Yang, Yosuke Todo, Kexin Qiao, and Lei Hu. Analysis of AES, SKINNY, and others with constraint programming. *IACR Trans. Symmetric Cryptol.*, 2017(1):281–306, 2017.

[SHW+14] Siwei Sun, Lei Hu, Peng Wang, Kexin Qiao, Xiaoshuang Ma, and Ling Song. Automatic security evaluation and (related-key) differential characteristic search: Application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers. In *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*, pages 158–178, 2014.

[SHY16] Ling Song, Zhangjie Huang, and Qianqian Yang. Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. In *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*, pages 379–394, 2016.

[Sin05] Carsten Sinz. Towards an optimal CNF encoding of boolean cardinality constraints. In *Principles and Practice of Constraint Programming - CP 2005, 11th International Conference, CP 2005, Sitges, Spain, October 1-5, 2005, Proceedings*, pages 827–831, 2005.

[SLR+15] Bing Sun, Zhiqiang Liu, Vincent Rijmen, Ruilin Li, Lei Cheng, Qingju Wang, Hoda AlKhzaimi, and Chao Li. Links among impossible differential, integral and zero correlation linear cryptanalysis. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 95–115, 2015.

A Potential Usages for General 4-bit and 8-bit S-boxes

Even though we put some limitations on the S-box in Section 3.2.2, we remark that the method can be generalised to the more general cases.

For 4-bit S-boxes with differential uniformity larger than 4, whose weight is not limited to being an integer, some tricks can be applied to address this problem. For example, we may import new variables as a symbolic representation for the decimal part of the weight. Suppose that $\mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 6/16$, and the corresponding weight is 1.415. For the integer part, we introduce variables p_i's as usual. Additionally, we use variable s to stand for the existence of 0.415. The Boolean function describing the DDT is modified as:

$$
\begin{aligned}
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 0: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}\|\boldsymbol{s}) = 0;\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 2^{-3}: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}\|\boldsymbol{s}) = \begin{cases} 1 & \text{if } \boldsymbol{p}\|\boldsymbol{s} = (1,1,1,0)\\ 0 & \text{else}\end{cases};\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 2^{-2}: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}\|\boldsymbol{s}) = \begin{cases} 1 & \text{if } \boldsymbol{p}\|\boldsymbol{s} = (0,1,1,0)\\ 0 & \text{else}\end{cases};\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 2^{-1.415}: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}\|\boldsymbol{s}) = \begin{cases} 1 & \text{if } \boldsymbol{p}\|\boldsymbol{s} = (0,0,1,1)\\ 0 & \text{else}\end{cases};\\
&\text{for } \mathrm{DP}_S(\boldsymbol{x},\boldsymbol{y}) = 1: && f(\boldsymbol{x}\|\boldsymbol{y}\|\boldsymbol{p}\|\boldsymbol{s}) = \begin{cases} 1 & \text{if } \boldsymbol{p}\|\boldsymbol{s} = (0,0,0,0)\\ 0 & \text{else}\end{cases}.
\end{aligned}
$$

The objective function remains only related to $\sum_{i,j,k} p_k^{(i,j)}$, the solution for s is as output in addition to the trail. The total weight, which equals to $\sum_{i,j,k} p_k^{(i,j)} + 0.415 \cdot \left(\sum_{i,j} s^{(i,j)}\right)$, is computed outside the SAT problem.

For 8-bit S-boxes, whose DDT cannot be handled by Logic Friday directly, we may borrow the method of converting DDT into some *pb*-DDT's introduced in [AST+17]. As to the weights being non-integer values, the trick stated above can be applied.

B Supplementary Materials for LED

B.1 STEP Function of LED

The four operations consisted in the round function is defined as follows.

AddConstants At each round, the 6-bit round constant $(\mathrm{rc}_5, \mathrm{rc}_4, \mathrm{rc}_3, \mathrm{rc}_2, \mathrm{rc}_1, \mathrm{rc}_0)$ is arranged into an array and is combined with the state using bitwise exclusive-or. In the submission version to CHES 2011, the constant matrix is defined as

$$
\begin{bmatrix}
0 & (\mathrm{rc}_5\|\mathrm{rc}_4\|\mathrm{rc}_3) & 0 & 0\\
1 & (\mathrm{rc}_2\|\mathrm{rc}_1\|\mathrm{rc}_0) & 0 & 0\\
2 & (\mathrm{rc}_5\|\mathrm{rc}_4\|\mathrm{rc}_3) & 0 & 0\\
3 & (\mathrm{rc}_2\|\mathrm{rc}_1\|\mathrm{rc}_0) & 0 & 0
\end{bmatrix}. \tag{14}
$$

Later, the authors issued a new version [GPPR12] with minor modification in the round constant matrix. The new round constant matrix depending on the key size as well is

$$
\begin{bmatrix}
0 \oplus (\mathrm{ks}_7\|\mathrm{ks}_6\|\mathrm{ks}_5\|\mathrm{ks}_4) & (\mathrm{rc}_5\|\mathrm{rc}_4\|\mathrm{rc}_3) & 0 & 0\\
1 \oplus (\mathrm{ks}_7\|\mathrm{ks}_6\|\mathrm{ks}_5\|\mathrm{ks}_4) & (\mathrm{rc}_2\|\mathrm{rc}_1\|\mathrm{rc}_0) & 0 & 0\\
2 \oplus (\mathrm{ks}_3\|\mathrm{ks}_2\|\mathrm{ks}_1\|\mathrm{ks}_0) & (\mathrm{rc}_5\|\mathrm{rc}_4\|\mathrm{rc}_3) & 0 & 0\\
3 \oplus (\mathrm{ks}_3\|\mathrm{ks}_2\|\mathrm{ks}_1\|\mathrm{ks}_0) & (\mathrm{rc}_2\|\mathrm{rc}_1\|\mathrm{rc}_0) & 0 & 0
\end{bmatrix}, \tag{15}
$$

where $(\mathrm{ks}_7, \mathrm{ks}_6, \ldots, \mathrm{ks}_0)$ stands for the 8-bit representation of the key size.

`SubCells` Each nibble is replaced by the nibble generated after using the PRESENT [BKL+07] S-box.

`ShiftRows` Row i of the array is rotated i cell positions to the left for $i = 0, 1, 2, 3$.

`MixColumnsSerial` Each column of the array is viewed as a column vector and replaced by the column vector that results after post-multiplying the vector by a matrix M.

B.2 Differential Attacks for LED64

In this section, we recall the three differential attacks of LED64 for the integrality of the paper.

B.2.1 3-STEP and 4-STEP Related-key Attacks for LED64

The attack is based on the assumption that one can find a good differential $\Delta^* \to \Delta$ with high probability p in F_i, see Figure 6.

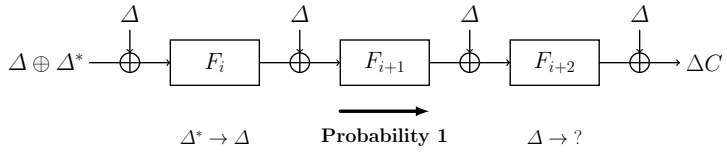

Figure 6: Attack on LED64 with 3-STEP

The attacker first constructs a list L composed of $2^{n/2} \cdot (1/p)^{1/2}$ pairs $(\Delta F_{i+2}(a), a)$, where $\Delta F_{i+2}(a) = F_{i+2}(a) \oplus F_{i+2}(a \oplus \Delta)$. Then, he randomly chooses P and $P' = P \oplus \Delta \oplus \Delta^*$ and asks for the ciphertexts C and C'. If $\Delta C = C \oplus C'$ is in the list L, a candidate for K is obtained as $K = F_{i+2}(a) \oplus C$. After repeating the randomly chosen procedure, the expected number of matches in the list L is $1/p$. Since the differential in F_i holds with probability p, one of these matches will satisfy $\Delta F_i = \Delta$. In sum, the time and memory requirements of this attack depending on p are $2^{n/2} \cdot (1/p)^{1/2}$.

This attack can be extended to four steps of LED64. The main observation is that if we can find a good iterative differential $\Delta \to \Delta$ for F_{i+1} with probability p, then a differential covering four steps can be constructed and the attack described above can be applied with $2^{n/2} \cdot (1/p)^{1/2}$ time and memory requirements. Please refer to Figure 7 for an illustration.

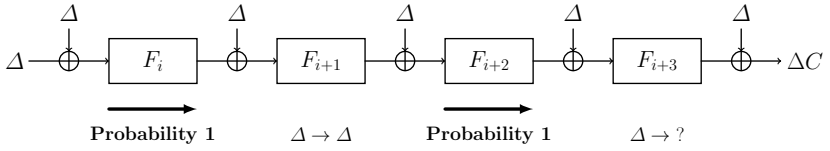

Figure 7: Attack on LED64 with 4-STEP

B.2.2 5-STEP Chosen-key Attack for LED64

The attack is designed to construct q-multicollisions, which is defined below.

Definition 1 ([BKN09])**.** A *differential q-multicollisions* for the block cipher $E_K(\cdot)$ is defined as a set of two differences ΔP and ΔK and q key-plaintext pairs (K_1, P_1), (K_2, P_2), $\ldots$, (K_q, P_q) that satisfy the relation:

$$E_{K_1}(P_1) \oplus E_{K_1 \oplus \Delta K}(P_1 \oplus \Delta P)$$

$$\begin{aligned} &= E_{K_2}(P_2) \oplus E_{K_2 \oplus \Delta K}(P_2 \oplus \Delta P) \\ &= \cdots \\ &= E_{K_q}(P_q) \oplus E_{K_q \oplus \Delta K}(P_q \oplus \Delta P). \end{aligned}$$

Bogdanov et al. [BKN09] proved that it takes at least $q \cdot 2^{\frac{(q-2)}{(q+2)}n}$ queries to produce a differential q-multicollisions for an ideal n-bit permutation. Thus, an attacker can distinguish the dedicated cipher from an ideal permutation in the chosen-key model if he can find a differential q-multicollisions on the cipher with a complexity less than the lower bound $q \cdot 2^{\frac{(q-2)}{(q+2)}n}$.

The attack based on the differential path shown in Figure 8. and aims to construct a differential q-multicollisions. The differential on F_{i+4} is fixed to be a good differential $\Delta \to \Delta^*$ with probability p, and then set $\Delta P = \Delta$, $\Delta K = \Delta$, and $\Delta C = \Delta \oplus \Delta^*$.

The attacker starts with launching a meet-in-the-middle attack between F_{i+1} and F_{i+2}. He randomly selects 2^s values for the input X of F_{i+1} as well as the output Y of F_{i+2}, and matchs between $F_{i+1}(X) \oplus F_{i+1}(X \oplus \Delta)$ and $F_{i+2}^{-1}(Y) \oplus F_{i+2}^{-1}(Y \oplus \Delta) \oplus \Delta$. On average, there are 2^{2s-64} matches. For each match, the two values $F_{i+1}(X) \oplus F_{i+2}^{-1}(Y)$ and $F_{i+1}(X) \oplus F_{i+2}^{-1}(Y \oplus \Delta)$ are selected as key K. For each candidate key, he computes C and C' from the pairs (K, Y) and $(K \oplus \Delta, Y \oplus \Delta)$. If ΔC equals to $\Delta \oplus \Delta^*$, he stores the corresponding pair (P, K). On average, $2^{2s-63} \cdot p$ values of (P, K) are stored.

To produce a differential q-multicollisions, we set $2^{2s-63} \cdot p = q$, which implies $s = \frac{1}{2}\log_2 q - \frac{1}{2}\log_2 p + 31.5$. Since the time complexity is dominated by the meet-in-the-middle attack, which is $2^{s-1.3}$. To make the distinguishing attack success, we demand $2^{s-1.3} < q \cdot 2^{\frac{(q-2)}{(q+2)}n}$.

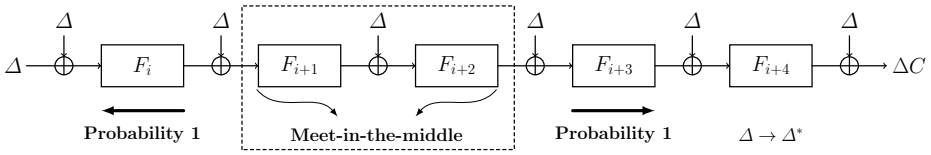

Figure 8: Attack on LED64 with 5-STEP

B.3 Iterative Differentials with High Height for LED64

We fix the number of active S-boxes to 25 for four rounds of LED and invoke SAT solver to exhaustively search for the 4-round iterative differential with input and output differences following the pattern 0x*000*000*000*000, i.e., only the first column is active. The experimental results indicate that 829 differentials satisfy this pattern, while it was mentioned in [MRTV12] that 240 iterative characteristics were available. For each of the 829 differentials, we search for the corresponding right pairs for different STEP functions $(F_0 - F_7)$ of LED64-old as well as LED64-new, and the maximum number of right pairs under different cases can be found in Table 2. The concrete differentials with the maximum number of right pairs are listed in Table 6.

Since we wonder whether there are iterative differentials with higher height following other patterns, we remove the restriction on the pattern and aim to search for iterative differential characteristic with maximum probability since the higher probability potentially indicates more right pairs. We observe that the probability for a 4-round iterative characteristic is less than or equal to 2^{-60}. There are in total 80 trails with probability 2^{-60}. Then, we search for right pairs for the 80 differentials containing the 80 trails. The maximum number of right pairs under different settings can be found in Table 2. Please refer to Table 7 for the dedicate differentials.

Table 6: Iterative differentials with pattern 0x*000*000*000*000.

STEP	LED64-old Input/Output Difference	#{RPs}	LED64-new Input/Output Difference	#{RPs}
F_0	0x00000000d0008000	40	0x0000000000003000	42
F_1	0x0000000000003000 0x000000003000a000	36	0x4000e000c0000000	40
F_2	0x000000008000f000	38	0x00000000d0008000	40
F_3	0x0000000000003000	42	0x00000000d0008000	40
F_4	0x0000000000003000	38	0x00000000d0008000	36
F_5	0x0000000000003000	44	0x000000008000f000	46
F_6	0x0000000000003000	38	0x7000e00000000000 0x000000003000a000	36
F_7	0x0000000000003000	38	0x0000000000003000	36

#{RPs}: The maximum number of right pairs.

Table 7: Iterative differentials with unfixed pattern.

STEP	LED64-old Input/Output Difference	#{RPs}	LED64-new Input/Output Difference	#{RPs}
F_0	0x0089000016000fd0	66	0xd004d00000a0009d	62
F_1	0x0089000016000fd0	64	0x010006f000d80009	64
F_2	0x00e0c0f0d0009030	62	0x010006f000d80009	76
F_3	0x00d3020302500975	68	0x00d3020302500975	70
F_4	0x0000070008c000b0	58	0x010006f000d80009 0x00e0c0f0d0009030	54
F_5	0x0001700630088000	82	0x0001700630088000	82
F_6	0x00d3020302500975	90	0x0001700630088000	68
F_7	0x00d3020302500975	90	0x0001700630088000	84

#{RPs}: The maximum number of right pairs.

B.4 Details for the Non-iterative Differentials

The authors [MRTV12] provided a 4-round differential with 1026 right pairs for the first `STEP` function of `LED64-old`,

$$\texttt{0x002280ff00091b30} \rightarrow \texttt{0xbb0b800098050701}.$$

We apply the automatic method to search for the right pairs of this differential, and first search for all characteristics with probability greater than 2^{-80}. There are 1039 trails, and the probability distribution of these trails is shown in Table 8. Note that only the characteristic with EDP 2^{-54} has right pairs, and the number of right pairs is 1026. We also test the right pairs of the same differential for `LED64-new` and observe that not only the values of right pairs vary but also the number of right pairs reduces. The differential only possesses 966 right pairs for the first `STEP` function of `LED64-new`.

Table 8: Probability distribution for the non-iterative differential in [MRTV12].

Probability	2^{-54}	2^{-73}	2^{-74}	2^{-75}	2^{-76}	2^{-77}	2^{-78}	2^{-79}
#{Trails}	1	4	22	67	142	178	261	364
#{Right Pairs}	1026	0	0	0	0	0	0	0

Theoretically, the probability of 4-round differential characteristics may go up to 2^{-50} (25 active S-boxes and each with probability 2^{-2}). Thus, we feel that the height of the 4-round differential can be improved and target at finding differentials with more right pairs. Firstly, by fixing the weight of the trail, we search for 4-round characteristics with EDP 2^{-50} by using the method introduced in Section 3. We can obtain more than five millions of characteristics. Then, we randomly select 100 trails and search for the right pairs of the 100 differentials corresponding to these trails. In the characteristic-search phase, we restrict ourselves to search for trails with probability more significant than 2^{-80}.

Among these differentials, one differential has around 2^{15} right pairs for all `STEP` functions ($F_0 - F_7$) of the two versions, and the differential is

$$\texttt{0x0780003ba0007000} \rightarrow \texttt{0x36de4c3562a87eb7}.$$

C A Brief Introduction of `Midori64`

`Midori64` takes the following 4×4 state as a data expression:

$$\begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix},$$

where the size of each cell is 4 bits.

Each layer of the round function updates the state S as follows.

`SubCell` A 4-bit S-box is applied to every cell of the state.

`ShuffleCell` The cells of the state are permuted as follows:

$$(s_0, s_1, \ldots, s_{15}) \leftarrow (s_0, s_{10}, s_5, s_{15}, s_{14}, s_4, s_{11}, s_1, s_9, s_3, s_{12}, s_6, s_7, s_{13}, s_2, s_8).$$

`MixColumn` The matrix M is applied to every column of the state S.

`KeyAdd` The i-th round key RK_i is `XOR`ed to the state S.