Effect of the Dependent Paths in Linear Hull*

Zhenli Dai, Meiqin Wang, Yue Sun

School of Mathematics, Shandong University, Jinan, 250100, China
Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan, 250100, China
mqwang@sdu.edu.cn

Abstract. Linear Hull is a phenomenon that there are a lot of linear paths with the same data mask but different key masks for a block cipher. In 1994, Nyberg presented the effect on the key-recovery attack such as Algorithm 2 with linear hull, in which the required number of the known plaintexts can be decreased compared with that in the attack using individual linear path. In 2009, Murphy proved that Nyberg's results can only be used to give a lower bound on the data complexity and will be no use on the real linear cryptanalysis. In fact, the linear hull have this kind of positive effect in linear cryptanalysis for some keys instead of the whole key space. So the linear hull can be used to improve the traditional linear cryptanalysis for some weak keys. In the same year, Ohkuma gave the linear hull analysis on PRESENT block cipher, and pointed that there are 32% weak keys of PRESENT which make the bias of a given linear hull with multiple paths more than that of any individual linear path. However, Murphy and Ohkuma have not considered the dependency of the muti-path, and their results are based on the assumption that the linear paths are independent. Actually, most of the linear paths are dependent in the linear hull, and the dependency of the linear paths means the dependency of the equivalent key bits. In this paper, we will analyze the dependency of the linear paths in linear hull and present the real effect of linear hull with the dependent linear paths. Firstly, we give the relation between the bias of a linear hull and its linear paths in linear cryptanalysis. Secondly, we present the algorithm to compute the rate of weak keys corresponding to the expect bias of the linear hull. At last, we verify our algorithm by cryptanalyzing reduced-round of PRESENT. Compared with the rate of weak keys under the assumption of the independent linear paths, the dependency of the linear paths will greatly reduce the rate of weak keys for a given linear hull.

Keywords: Linear Hull, Dependency of Linear Paths, Weak Key, PRESENT, Block Cipher.

* Supported by 973 Project (No.2007CB807902), National Natural Science Foundation of China (Grant No.60910118), Outstanding Young Scientists Foundation Grant of Shandong Province (No. BS2009DX030), and Shandong University Initiative Scientific Research Program (2009TS087).

1 Introduction

Linear cryptanalysis [1] is a powerful method of cryptanalysis introduced by Matsui in 1993. It is a known plaintext attack in which the attacker studies the linear approximations of parity bits of the plaintext, ciphertext and the subkey. The probability p of linear approximation needs to be away from $1/2$, and the magnitude of the bias $\epsilon = p - 1/2$ represents the effectiveness of the linear approximation. Based on this idea, many variants of linear cryptanalysis appeared, such as linear cryptanalysis using multiple linear approximations with the same key mask[2], multiple linear approximations cryptanalysis[3] and linear cryptanalysis based on linear hull[4], etc.

Linear cryptanalysis using multiple approximations [2] was introduced by Kaliski and Robshaw in 1994. For a given success rate, this method reduced the data complexity by using multiple linear approximations. But their technique is limited to cases where all approximations have the same key mask. Unfortunately, this approach imposes a very strong restriction on the approximations. The concept of linear hull [4] was first announced by Nyberg in 1994, and a linear hull stands for the collection of all linear relations that have the same input mask and output mask, but involve different sets of round subkey bits in the different linear paths. The linear hull effect accounts for a clustering of linear paths, which decreased the required number of known plaintexts for a given success rate. In 2009, however, Murphy proved that there is no linear hull effect in linear cryptanalysis [5]. In the same year, Ohkuma pointed that 32% of the whole key space for PRESENT are weak keys which will produce much larger bias by the multi-path effect compared with that by the single linear path[6]. That is to say, the number of required known plaintexts can be reduced apparently for these weak keys. However, the results of Murphy and Ohkuma are all based on the assumption that all linear paths are independent. In fact, the assumption is not correct, so we need to reconsider the effect of linear hull.

Here we will analyze how the dependency of linear paths of linear hull affect the linear cryptanalysis. And then we give the relationship between the bias of linear hull and equivalent subkey values of the linear paths. Since the linear paths are dependent, we will give the method to compute the final bias of linear hull for a given key and offer a method to compute the rate of weak keys with the expected bias. In order to verify our method, we computed the bias and the corresponding weak keys for block cipher PRESENT. As a result, we found that it is deeply dependent among linear paths of the given data mask and the rate of weak keys is lower than that in [6], which has been further confirmed by our experimental data.

This paper is organized as follows. Section 2 briefly introduces the linear hull and the block cipher PRESENT. Section 3 presents the relationship between the linear bias and equivalent subkey values, then describes our computational formula of the linear bias under the dependent linear paths. In Section 4, we compute the rate of weak keys with the expected bias for reduced-round PRESENT. Section 5 concludes this paper.

2 Preliminaries

2.1 Introduction of Linear Hull

The concept of **linear hull** was first proposed by Nyberg in [4]. A linear hull stands for the collection of all linear approximations (across a certain number of rounds) that have the same input and output masks, but involves different sets of round subkeys according to different linear paths. As we know, the differential is the set of the differential characteristics, and similarly the linear hull is the set of the linear approximations. It is easy to compute the probability of the differential with multiple differential characteristics, but the bias of the linear hull is difficult to be obtained.

In [4], Nyberg also proposed the concept of **linear hull effect** which accounts for a clustering of linear paths. Because of the existence of the linear hull effect, the final bias may become significantly higher than that of any individual linear path. Denote the input mask as a and the output mask as b for a block cipher $Y = Y(X, K)$, Nyberg computed the **potential** of the corresponding linear hull as follows,

$$\mathrm{ALH}(a,b) = \sum_{c_i \in \Gamma} (\mathrm{P}(a \cdot X \oplus b \cdot Y = c_i \cdot K) - \frac{1}{2})^2 = \eta^2, \tag{1}$$

where c_i is the mask for the subkey bits, and the set $\Gamma = \{c_i \cdot K\}$. Then, key-recovery attacks such as Algorithm 2 in [1] apply with

$$N = \frac{t}{\mathrm{ALH}(a,b)} = \frac{t}{\eta^2}$$

known plaintexts, where t is a constant. An advantage to use linear hulls in key-recovery attacks, such as in Algorithm 2, is that the required number of known plaintexts can be decreased for a given success rate.

2.2 Brief Description of PRESENT

PRESENT is an Ultra-Lightweight block cipher proposed by A.Bogdanov, L.R.Knudsen and G.Leander et al.[9]. PRESENT is a 31-round SP-network with block length 64 bits and 80 bits or 128 bits key length. The round function consists of three layers: AddRoundKey, SboxLayer and pLayer. The AddRoundKey is a 64-bit exclusive OR operation with a round key. The SboxLayer is a 64-bit nonlinear transform using a single S-box 16 times in parallel. The S-box is a nonlinear bijective mapping given in Tab. 4. The pLayer is a bit-by-bit permutation given in Tab. 5. The design idea of SboxLayer and pLayer is adapted from Serpent [7] and DES block cipher [8], respectively.

3 The Bias of the Linear Hull

3.1 The General Formula of the Bias

A **linear path** is defined as a single path of linear approximations concatenated over multiple rounds [11]. Now suppose that there is a n-round linear hull with data mask (a, b) and L linear paths. The bias of the linear hull is denoted as η, and the bias of each linear path is denoted as $\epsilon_i (1 \leqslant i \leqslant L)$. In addition, $c_i \cdot K (1 \leqslant i \leqslant L)$ are subkey masks. In fact, each $c_i \cdot K$ is a key expression about the subkey bits and we name it as the equivalent subkey bit. The expressions of linear paths are defined as follows,

$$a \cdot X \oplus b \cdot Y = c_i \cdot K \text{ with probability } \frac{1}{2} + \epsilon_i,\ c_i \in \Gamma \tag{2}$$

where Γ is the space of subkey masks.

From [5] and [6], we know that η is determined by ϵ_i and $c_i \cdot K$. The bias ϵ_i may be positive or negative. Without loss of generality, we can assume that all biases are positive, as the sign can be absorbed in the equivalent subkey bits. For example, if $\epsilon_i < 0$, we have $(-1)^{c_i \cdot K} \epsilon_i = (-1)^{c_i \cdot K \oplus 1}(-\epsilon_i)$. Then we get the equivalent subkey bit $k_i = c_i \cdot K \oplus 1$. So the bias of a linear hull is given by

$$\eta = \sum_{i=1}^{L} (-1)^{k_i} \epsilon_i, \tag{3}$$

where $\epsilon_i > 0$ and k_i is the equivalent subkey bit. We confirmed equation (3) by using a 4-round linear hull of PRESENT, and our results are given in Tab. 6.

3.2 How to Compute the Bias of Linear Hull with Dependent Paths

In this part, we will discuss how the dependency of linear paths affects the bias of a linear hull. For an n-round liner hull of data mask (a, b), we suppose that it contains L linear paths. Let us denote every linear path as

$$\begin{aligned} a \cdot X \oplus b \cdot Y &= K^{(0)}[\chi_j^0] \oplus K^{(1)}[\chi_j^1] \oplus \ldots \oplus K^{(n)}[\chi_j^n] \\ &= k_j, \text{ with probability } p_j = \frac{1}{2} + \epsilon_j, 1 \leqslant j \leqslant \mathrm{L}, \end{aligned} \tag{4}$$

where k_j is a equivalent subkey bit, $\Gamma = \{k_j\}_{j=1}^L$.

Here we denote the vector of key mask c_j as $(c_{j,0}^0, \cdots, c_{j,63}^0, c_{j,0}^1, \cdots, c_{j,63}^1, \cdots, c_{j,0}^n, \cdots, c_{j,63}^n)$, where $c_{j,l}^r \in \{0,1\}$, $0 \leq r \leq n$, $0 \leq l < 63$, and $c_{j,l}^r \in \{0,1\}$ is the coefficient of the l-th bit of the r-th round subkey. According to equation (4), the dependency of linear paths means the dependency of the vector of key masks $c_j (1 \leq j \leq L)$. For example, if the first four linear paths are dependent, we have $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$. That is to say the dependency of linear paths is the dependency of vectors $c_j (1 \leq j \leq L)$. We also named $k_j = c_j \cdot K$ as equivalent subkey bit, so the dependency of linear paths also means their equivalent subkey bits are dependent.

Assume that the maximum number of linear paths whose equivalent subkey bits are independent with each other is R. Without loss of generality, we assume that $k_1, k_2, \ldots, k_R$ are independent with each other, and name them as **independent subkey bits**, which form a set Γ_1. The **dependent subkey bits**, which form a set Γ_2, are denoted by the dependent subkey expressions $k_j = c_{1,j} k_1 \oplus c_{2,j} k_2 \oplus \ldots \oplus c_{R,j} k_R$, $R < j \leqslant L$, $c_{i,j} \in \{0,1\}$, $1 \leqslant i \leqslant R$. So $\Gamma = \Gamma_1 \cup \Gamma_2$.

In order to compute the bias of linear hull, we must find out the regularity of distribution of independent subkey bits on the expressions of the dependent subkey bits. So we study the relationship between independent subkey bits and dependent subkey bits at first.

- If we do the XOR operation for two different equations in (4), we get

$$0 = K^{(0)}[\chi_i^0, \chi_j^0] \oplus \ldots \oplus K^{(n)}[\chi_i^n, \chi_j^n], \ i \neq j.$$

  Obviously, this expression is not a linear path of the linear hull. The conclusion always holds if the number of these equations is even.
- When the number of equations is three, we have

$$a \cdot P \oplus b \cdot C = K^{(0)}[\chi_u^0, \chi_v^0, \chi_w^0] \oplus \ldots \oplus K^{(n)}[\chi_u^n, \chi_v^n, \chi_w^n], \ u \neq v \neq w.$$

Obviously, the expression is a linear path of the linear hull. The conclusion always holds if the number of these equations is odd.

So we affirm that every dependent subkey bit is determined by the odd number of independent subkey bits. That is to say, the sum of coefficients $r_j = \sum_{i=1}^{R} c_{ij}$ for k_j $(k_j \in \Gamma_2)$ is odd. Let us denote the maximal sum as

$$r' = \max_{R<j\leqslant L}\{r_j\} = \max_{R<j\leqslant L}\{\sum_{i=1}^{R} c_{i,j}\}. \tag{5}$$

Now suppose that we have derived all the linear paths in a linear hull, the relationship between dependent subkey bits and independent subkey bits can be obtained. We classify all the independent equivalent subkey bits according to their values, and present the method to compute the bias of a given linear hull and the rate of weak keys satisfying a lower bound of the bias. The main idea is described as follows,

1. We study the distribution of the independent subkey bits on the expressions of the dependent subkey bits. For a given master key, every independent subkey bit has two possible values: 0 or 1, and $|\Gamma_1| = 2^R$.
   a. For a possible value of Γ_1, suppose that the number of the independent subkey bits whose values are 0 is s $(s \leqslant R)$, and the number of the independent subkey bits whose values are 1 is $(R - s)$. We classify the independent subkey bits into two groups according to their values.
   b. Consider the values of the dependent subkey bits. If there are odd number of subkey bits among the s subkey bits in the expressions of the dependent key bit k_j $(R < j \leqslant L)$, we have $k_j = 0$.
   c. In order to get the general formula, we classify the dependent subkey bits according to the number of independent subkey bits, whose values are 0, in the expressions of them.

   Fig. 1 is useful to understand our idea.
2. Compute the bias of the linear hull for every possible value of Γ_1. Then we can calculate the rate of weak keys according to the definition of weak keys.

Let us denote the times of k_i $(1 \leqslant i \leqslant R)$ appeared in the dependent subkey bits as N_i, and the times of $k_{i_1} \oplus k_{i_2} \oplus \ldots \oplus k_{i_t}$ appeared in dependent subkey bits as $N_{i_1,i_2,\ldots,i_t}$, $(1 \leqslant i_1 < i_2 < \ldots < i_t \leqslant R,\ t \leqslant R)$. We denote $N_{i_1,i_2,\ldots i_t}$ as $N_{(t)}$ at the case of no ambiguity, and then $0 \leqslant$

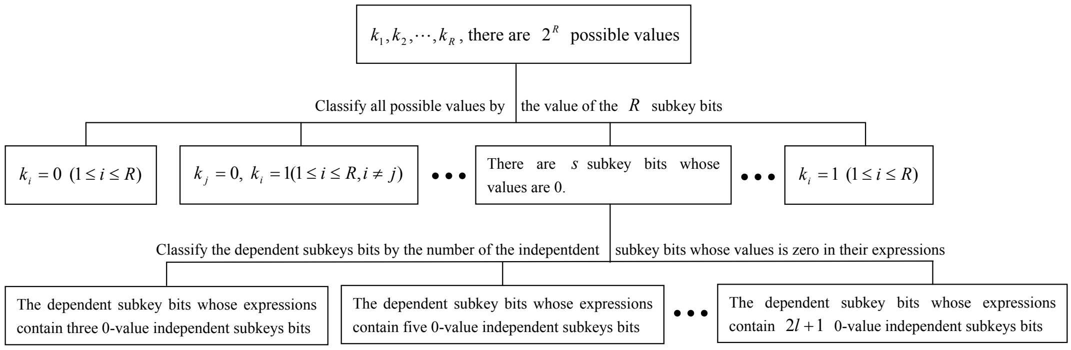

Fig. 1. Classification of Linear Paths

$N_{(t)} \leqslant L - R$. According to the definition, we have

$$N_i = \sum_{j=R+1}^{L} c_{i,j}, \qquad N_{i_1,i_2,\ldots,i_t} = \sum_{j=R+1}^{L} (\prod_{l=1}^{t} c_{i_l,j}).$$

Here we only consider the best linear paths which have the same bias $\epsilon_j = \epsilon > 0$ $(1 \leqslant j \leqslant L)$. Then the bias of linear hull is $\eta = \sum_{i=1}^{L} (-1)^{k_i} \epsilon$, Let us denote T_s^j as the number of dependent subkey bits in which the values of j independent subkey bits are zero. And we define the number of the dependent key bits with zero value as

$$T_s = \sum_{1 \leqslant j \leqslant s,\ j \text{ is odd}} T_s^j \ ,$$

If we choose the values of s independent subkey bits, we have derived equation (6) to compute T_s in App. C.

$$\begin{aligned} T_s = &\sum_{j=1}^{s} N_j - 2 \sum_{1 \leqslant i < j \leqslant s} N_{i,j} + 4 \sum N_{(3)} - 8 \sum N_{(4)} + \ldots \\ &- (2l + \binom{2l}{3} + \binom{2l}{5} + \ldots + \binom{2l}{2l-1}) \cdot \sum N_{(2l)} \\ &+ (2l + 1 + \binom{2l+1}{3} + \binom{2l+1}{5} + \ldots + \binom{2l+1}{2l+1}) \cdot \sum N_{(2l+1)} \\ &+ \ldots + (-1)^{s-1} (s + \binom{s}{3} + \binom{s}{5} + \ldots) \cdot N_{(s)}. \end{aligned} \tag{6}$$

If s independent zero subkey bits have been chosen, we can compute a value of T_s with equation (6). There are $\binom{R}{s}$ different distributions for the s independent subkey zero bits, so the number of computed T_s is $\binom{R}{s}$.

Property 1: For a given key with L subkey bits, if there are s independent zero subkey bits, $R-s$ independent non-zero subkey bits, and h dependent zero subkey bits, the bias of the linear hull corresponding to the key will be $((s+h)-((R-s)+(L-R-h)))\cdot\epsilon = (2(h+s)-L)\cdot\epsilon$.

Now in order to compute the number of possible subkey values corresponding to the different bias, we will classify T_s by their values in any distributions of s independent zero subkey bits. Considering all the distributions for the s independent zero subkey bits, we denote the total number of any s independent zero subkey bits with the bias $(2(h+s)-L)\cdot\epsilon$ as $m_h^{(s)}$. For $\binom{R}{s}$ possible values, the times of computed T_s is $\#\{T_s\} = \binom{R}{s} = \sum_{h=0}^{L-R} m_h^{(s)}$, for the different values of h, we will compute their bias corresponding to $\binom{R}{s}$ different subkey values. Then we can obtain the number of the subkey values with the expected bias.

For each value of s $0 \le s \le R$, we need to compute $\binom{R}{s}$ times of T_s. In order to reduce the computing time, we identify the following property:

Property 2: For a given key with L subkey bits, if there are s independent non-zero subkey bits and $R-s$ independent zero subkey bits, and h dependent non-zero subkey bits, the bias of the linear hull corresponding to the key will be $(((R-s)+(L-R-h))-(s+h))\cdot\epsilon = -(2(h+s)-L)\cdot\epsilon$.

From Property 1 and Property 2, the absolute bias of the two cases are equal. Therefore, we only computed the bias of $s \leqslant \lfloor R/2 \rfloor$. In equation (7), we only need to compute $N_{(t)}$, $t \leqslant \lfloor R/2 \rfloor$. If $r' < \lfloor R/2 \rfloor$ in equation (5) and $r' < l \leqslant \lfloor R/2 \rfloor$, we can obtain $N_{(l)} = 0$. Then equation (6) can be simplified to the following equation:

$$\begin{aligned}\mathrm{T}_s = &\sum_{j=1}^{s} N_j - 2\sum_{1\leqslant i<j\leqslant s} N_{i,j} + 4\sum N_{(3)} - 8\sum N_{(4)} \\ &+ \ldots + (-1)^{r'-1}\left(r' + \binom{r'}{3} + \binom{r'}{5} + \ldots + \binom{r'}{r'}\right)\cdot N_{(r')}.\end{aligned} \tag{7}$$

If $r' \geq \lfloor R/2 \rfloor$ in equation (5), we will still compute T_s with equation (6).

With the above methods, we can only compute $m_h^{(s)}$ for $(s \leqslant \lfloor R/2 \rfloor)$. According to Property 1 and Property 2, the number of equivalent subkey values satisfying $|\eta| = |L-2(h+s)|\cdot\epsilon$ usually is $2m_h^{(s)}$. However, there is a special case, if R is an even and $s = R/2$, the number of equivalent subkey values satisfying $|\eta| = |L-2(h+s)|\cdot\epsilon$ is $m_h^{(s)}$.

When independent subkey bits take all 2^R possible values, the number of equivalent subkey values with the different biases are computed as follows,

$$
\begin{aligned}
&\#\{|\eta| = L\cdot\epsilon\} = 2,\\
&\#\{|\eta| = |L-2|\cdot\epsilon\} = 2m_0^{(1)},\\
&\#\{|\eta| = |L-4|\cdot\epsilon\} = 2m_1^{(1)} + 2m_0^{(2)},\\
&\#\{|\eta| = |L-6|\cdot\epsilon\} = 2m_2^{(1)} + 2m_1^{(2)} + 2m_0^{(3)},\\
&\#\{|\eta| = |L-8|\cdot\epsilon\} = 2m_3^{(1)} + 2m_2^{(2)} + 2m_1^{(3)} + 2m_0^{(4)},\\
&\quad\vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad \ddots,\\
&\#\{|\eta| = |L-2(L-R+1)|\cdot\epsilon\} = 2m_{L-R}^{(1)} + 2m_{L-R-1}^{(2)} + \ldots + c\cdot m_{L-R-\lfloor R/2\rfloor+1}^{(\lfloor R/2\rfloor)},\\
&\#\{|\eta| = |L-2(L-R+2)|\cdot\epsilon\} = \qquad 2m_{L-R}^{(2)} \quad + \ldots,\\
&\quad\vdots \qquad\qquad\qquad\qquad\qquad\qquad \ddots \qquad \vdots,\\
&\#\{|\eta| = |L-2(L-R+\lfloor R/2\rfloor)|\cdot\epsilon\} = \qquad\qquad c\cdot m_{L-R}^{(\lfloor R/2\rfloor)}\ .
\end{aligned}
\tag{8}
$$

where if R is even, $c = 1$, if R is odd, $c = 2$.

We classify all possible equivalent subkey values by their resulted biases of linear hull in equation (8), and we can easily compute the rate of weak keys according to this equation.

4 The Rate of Weak Keys for Reduced-Round PRESENT

4.1 Linear Paths of PRESENT

From [10] and [11], we know that there are plenty of linear hulls in PRESENT which have multiple linear paths. And each linear path exploits the linear approximations of S-boxes with only one non-zero bit for the input and output masks. The output mask of S-box with more than one non-zero bit will affect at least two S-boxes in the next round due to the permutation layer, which will result much less linear correlation in the multiple rounds of PRESENT.

Here we only focus on the linear single-bit paths with the highest bias.

Just as in [11], let $\pi(\alpha,\beta)$ denote a linear approximation of S-box S where $\alpha,\beta\in\mathbf{F}_2^4$ are the input and output masks of S respectively. The bias of $\pi(\alpha,\beta)$ is denoted by $\epsilon(\alpha,\beta)$. The S-box has the following properties [11]:

Property 3: For $\alpha,\beta\in\{2,4,8\}$, $\epsilon(\alpha,\beta) = \pm 2^{-3}$, except that $\epsilon(8,4) = 0$.
Property 4: For $\alpha\in\{1,2,4,8\}$, $\epsilon(\alpha,1) = \epsilon(1,\alpha) = 0$.

Let us define $I = \{S_5, S_6, S_7, S_9, S_{10}, S_{11}, S_{13}, S_{14}, S_{15}\}$ and $A = \{4i+1, 4i+2, 4i+3 \mid 0 \le i \le 15, S_i \in I\}$. Then, the permutation P of the pLayer has the following property [11]:

Property 5: If $x \in A$, then $\mathrm{P}(x) \in A$.

According to the above three properties, there are nine S-boxes of S which are usable for each round of a single-bit path, and there are three possible values for the mask of each S-box. Let $\mathrm{M}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ (only the i-th ($i \in A$) bit is non-zero) denote the input mask or output mask, there are no more than 27 possible mask values for each round.

For n-round linear paths, let $L_i^{(j)}(i \in A,\ 0 \leqslant j \leqslant n)$ denotes the number of linear paths in which the i-th bit of j-th round output mask (namely, (j+1)-th round input mask) is 1. When $j = 0$, the 0-th round output mask means the plaintext mask. We get the formula by Property 1 and Tab. 5 as follows:

$$\begin{array}{lll}
L_{21}^{j+1} = L_{21}^j + L_{22}^j + L_{23}^j, & L_{37}^{j+1} = L_{21}^j + L_{22}^j, & L_{53}^{j+1} = L_{21}^{j+1}, \\
L_{22}^{j+1} = L_{25}^j + L_{26}^j + L_{27}^j, & L_{38}^{j+1} = L_{25}^j + L_{26}^j, & L_{54}^{j+1} = L_{22}^{j+1}, \\
L_{23}^{j+1} = L_{29}^j + L_{30}^j + L_{31}^j, & L_{39}^{j+1} = L_{29}^j + L_{30}^j, & L_{55}^{j+1} = L_{23}^{j+1}, \\
L_{25}^{j+1} = L_{37}^j + L_{38}^j + L_{39}^j, & L_{41}^{j+1} = L_{37}^j + L_{38}^j, & L_{57}^{j+1} = L_{25}^{j+1}, \\
L_{26}^{j+1} = L_{41}^j + L_{42}^j + L_{43}^j, & L_{42}^{j+1} = L_{41}^j + L_{42}^j, & L_{58}^{j+1} = L_{26}^{j+1}, \\
L_{27}^{j+1} = L_{45}^j + L_{46}^j + L_{47}^j, & L_{43}^{j+1} = L_{45}^j + L_{46}^j, & L_{59}^{j+1} = L_{27}^{j+1}, \\
L_{29}^{j+1} = L_{53}^j + L_{54}^j + L_{55}^j, & L_{45}^{j+1} = L_{53}^j + L_{54}^j, & L_{61}^{j+1} = L_{29}^{j+1}, \\
L_{30}^{j+1} = L_{57}^j + L_{58}^j + L_{59}^j, & L_{46}^{j+1} = L_{57}^j + L_{58}^j, & L_{62}^{j+1} = L_{30}^{j+1}, \\
L_{31}^{j+1} = L_{61}^j + L_{62}^j + L_{63}^j, & L_{47}^{j+1} = L_{61}^j + L_{62}^j, & L_{63}^{j+1} = L_{31}^{j+1}.
\end{array} \tag{9}$$

For example, bypass Sboxplayer, non-zero bit in 21, 22 or 23 of the j-th round output mask all can produce the 21st non-zero bit of the output mask in (j+1)-th round, and $P(21) = 21$ according to Tab. 5. So we get $L_{21}^{j+1} = L_{21}^j + L_{22}^j + L_{23}^j$.

When we fix the input mask α with one non-zero value in bit l and the output mask β, we have $L_l^{(0)} = 1,\ L_i^{(0)} = 0, (i \neq l,\ i, l \in A)$, then the number of linear paths of n-round linear hull with data mask $(\alpha,\ \beta)$ is

$$L(n) = \sum_{j \in A} L_j^{(n-3)},\ n \geqslant 7. \tag{10}$$

Here we omit the proof of equation (10) because of the limit space.

Tab. 1 shows our computed results for $L(n)$ corresponding to a fixed linear hull, which are same as the results of Tab. 2 in [10]. In Tab. 1, the rank denote the number of the independent linear paths or the number of the independent equivalent subkey bits in a linear hull. The rank of i ($3 \leqslant i \leqslant 13$) rounds linear hull are obtained by our computed program. We find as the round number is increased , the rank will be increased proportionally. So we compute the rank of the linear hull from 14-round to 28-round.

Table 1. Number of Linear Paths and Rank of Equivalent Subkey Bits in PRESENT for data mask $(\mathrm{IM}_{21}, \mathrm{OM}_{21})$

#round	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
#paths	1	3	9	27	72	192	512	1344	3528	9261	24255	63525	166375	435600	1140480
rank	1	3	9	27	45	63	81	99	117	135	153	171	189	207	225

#round	18	19	20	21	22	23
#paths	2,985,984	7,817,472	20,466,576	53,582,633	140,281,323	367,261,713
rank	243	261	279	297	315	333

#round	24	25	26	27	28
#paths	961,504,803	2,517,252,696	6,590,254,272	17,253,512,704	45,170,283,840
rank	351	369	387	405	423

(IM_{21} means only the 21st bit of input mask is non-zero, and OM_{21} means only the 21st bit of output mask is non-zero.)

4.2 Computing the Rate of Weak Keys

For n-round PRESENT, we only consider the linear paths with the best bias and $|\epsilon_i| = \epsilon = 2^{-2n-1}$. Denote the number of linear paths with equivalent subkey bit zero by N^0, and the number of linear paths with equivalent subkey bit 1 by N^1. Then the bias for the linear hull is approximation by $2^{-2n-1}|N^0 - N^1|$ according to equation (3).

From Tab. 1, the linear paths of PRESENT are correlative with each other as $n \geq 7$. So it is inaccurate to estimate the rate of weak keys with the assumption of the independency of the linear paths.

For 7-round PRESENT, the data mask is $(\mathrm{IM}_{21}, \mathrm{OM}_{21})$, and the number of linear paths of the linear hull is $L = 72$. The rank of equivalent subkey bits $\Gamma = \{k_i\}_{i=0}^{71}$ is $R = 45$. As in the previous section, we assume that the first 45 equivalent subkey bits are independent subkey bits.

According to the relationship of linear paths we derived, all dependent subkey bits are determined by three independent subkey bits for 7-round linear hull. So we just consider three cases according to s:

- s=1, it means a single independent subkey bit (45 possible values);
- s=2, it means the combination of two independent subkey bits ($\binom{45}{2} = 990$ possible values);
- s=3, it means the combination of three independent subkey bits ($\binom{45}{3} = 14190$ possible values).

Supposed that there are s independent subkey bits with value zero, and let $\Lambda_s = \{j_u\}_{u=1}^{s}$, where $k_{j_u} = 0$. Firstly, counting $N_j (0 \leqslant j < 45)$, $N_{j_1,j_2} (0 \leqslant j_1 < j_2 < 45)$ and $N_{j_1,j_2,j_3} (0 \leqslant j_1 < j_2 < j_3 < 45)$. And $N_{(l)} = 0$ for $l > 3$. Secondly, we can compute

$$\begin{aligned}
&\mathrm{T}_1 = N_{j_1},\\
&\mathrm{T}_2 = N_{j_1} + N_{j_2} - 2N_{j_1,j_2},\\
&\mathrm{T}_3 = N_{j_1} + N_{j_2} + N_{j_3} - 2(N_{j_1,j_2} + N_{j_1,j_3} + N_{j_2,j_3}) + 4N_{j_1,j_2,j_3},\\
&\mathrm{T}_4 = \sum_{j_u \in \Lambda_4} N_{j_u} - 2 \sum_{j_u,\, j_v \in \Lambda_4} N_{j_u,j_v} + 4 \sum_{j_u,\, j_v,\, j_w \in \Lambda_4} N_{j_u,j_v,j_w},\\
&\cdots \qquad \cdots \quad \cdots,\\
&\mathrm{T}_{22} = \sum_{j_u \in \Lambda_{22}} N_{j_u} - 2 \sum_{j_u,\, j_v \in \Lambda_{22}} N_{j_u,j_v} + 4 \sum_{j_u,\, j_v,\, j_w \in \Lambda_{22}} N_{j_u,j_v,j_w}.
\end{aligned} \tag{11}$$

The values of T_s will be different when the positions of these s independent subkey bits are changed. Then we classify T_s by s and their values and get $m_{h_1}^{(1)}, m_{h_2}^{(2)}, m_{h_3}^{(3)}, \ldots, m_{h_{22}}^{(22)} (0 \leqslant h_1, h_2, \ldots, h_{22} \leqslant 27)$, where $27 = L - R$ is the number of dependent subkey bits. Finally, we know the bias of linear hull for any equivalent subkey values.

The computed complexity increases rapidly with the growing of the number of linear paths. Here we offer another method to compute the rate of weak keys.

We define the subkey values satisfying $|\eta| = |N^0 - N^1| \cdot \epsilon \geqslant \sqrt{72}\epsilon > 8\epsilon$ as weak keys. Instead of taking all possible subkey values to compute weak keys, we chose a large number of random subkey values to compute the rate of weak keys. The testing procedure is presented as follows,

- 1. Choose $N'(< 2^{32})$ values for 45-bit independent equivalent subkey bits randomly.
- 2. For each chosen value, compute the values of other 27 dependent equivalent subkey bits by the linear paths we derived. According to the number of zero subkey bits, add the counter of the corresponding bias value.
- 3. Compute the number of weak keys satisfying $|\eta| > 8\epsilon$.

The results are shown in Tab. 2, N' means the number of equivalent subkey values we tested, r_d is the rate of weak keys computed by our method (under the dependent linear paths), and r_u is the rate of weak keys computed by Ohkuma's model (under the assumption of independency of linear paths).

Table 2. The Rate of Weak Keys for 7-Round PRESENT

N'	r_d	r_u
2^{15}	28.05%	29.13%
2^{16}	28.07%	29.06%
2^{17}	28.04%	28.94%
2^{18}	28.09%	28.92%
2^{19}	28.13%	28.91%
2^{20}	28.12%	28.91%
2^{21}	28.13%	28.89%

If the 72 subkey bits are independent, each bit takes zero with the probability $\frac{1}{2}$. So the rate can be computed by the following equation:

$$1 - \frac{2\binom{72}{32} + 2\binom{72}{33} + 2\binom{72}{34} + 2\binom{72}{35} + \binom{72}{36}}{2^{72}} = 0.28878, \qquad (12)$$

which approaches to 28.89% in Tab. 2. So we believe that it is reasonable to use the above random test, and there are 28.13% weak key in 7-round linear hull of PRESENT, which is lower than the case under the assumption of independent linear paths.

We confirm that the rate of weak keys under dependent linear paths is less than that under independent case. That is to say, the dependency reduces the number of weak keys for the given linear hull.

In order to further verify our method, we compute the rate of weak keys of linear hull for more rounds PRESENT. First of all, we use different number of samples to count weak keys of i-round linear hull ($7 \leqslant i \leqslant 13$). And then we focus on the size of sample N' where the rate of weak keys is steady (that is more sample don't change the rate obviously). Finally, we randomly choose 100 groups sample whose size are N' to compute the rate of weak key. The results are listed in Tab. 3. Here n is the round of linear hull, L is the number of linear paths, R is the number of independent linear paths of all L paths, N' stands for the number of equivalent subkey values we used, r_d is the rate of weak keys computed by our method (under the dependent linear paths), and is called the computed rate, r_u is the rate of weak keys computed by Ohkuma's model (under the assume of independent linear paths), whose calculation method is similar to equation (12), and is called the predicted rate. $\triangle r$ is defined as

$$\triangle r = \frac{|r_u - r_d|}{r_u},$$

we call it **reduced rate**.

Table 3. The Rate of Weak Key for Reduced-Round PRESENT

n	L	R	N'	r_d	r_u	$\triangle r$
7	72	45	2^{21}	28.13%	28.88%	2.60%
8	192	63	2^{21}	32.65%	34.82%	6.23%
9	512	81	2^{21}	27.86%	30.94%	9.95%
10	1344	99	2^{22}	27.30%	31.28%	12.72%
11	3528	117	2^{22}	27.10%	32.05%	15.44%
12	9261	135	2^{22}	26.05%	31.85%	18.21%
13	24255	153	2^{22}	25.15%	31.65%	20.54%

At last, we compares the computed rate r_d and the predicted rate r_u in Fig. 2. From Fig. 2, the difference between the computed rate and the predicted rate will increase as the round number increases, which is caused by the dependency of the linear paths. Therefore, as the round number increases, the rate of weak keys will be reduced to very low value even approach to zero.

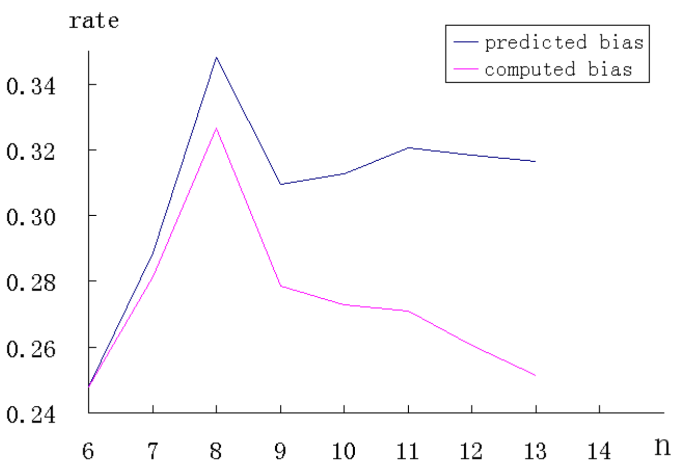

Fig. 2. Difference of Rate of Weak Keys

5 Conclusion

Linear cryptanalysis has been an important cryptanalytic method for block cipher. However, if there are linear hulls in the block cipher, the linear cryptanalysis may be strengthened or weaken which is decided by the used key. In fact, the linear cryptanalysis with linear hull is the cryptanalytic method under the assumption of the special weak keys. The previous attack with linear hull assumed the linear paths are independent. But the assumption is not true, so the previous attack is inaccurate.

In this paper, we assume the round subkeys are independent between each other and consider all kinds of the dependency in the linear paths, and derive the method to compute the number of the weak keys satisfying the expected bias for the linear hull. We verified our method by analyzing the reduced-round PRESENT block cipher. Compared with the rate of weak keys under the assumption of the independent linear paths, the dependency of the linear paths will greatly reduce the rate of weak keys for a given linear hull. As the round number increases, the rate of weak keys will be reduced to very low value. Therefore the linear cryptanalysis with linear hull has to be proceeded considering the dependency of the linear paths. However, as the round number increases, how to identify the dependency of all the linear paths in a linear hull needs huge computations, so it is difficult to proceed. In all, the linear cryptanalysis with linear hull will be difficult to proceed.

If we only consider the best paths and the number of linear paths L is an odd, the bias of linear hull is bias of a single path at least. We say that linear cryptanalysis is still effective for any master key in this case.

If we only consider the best paths and L is an even, the bias of linear hull may be zero for some master key. In this case we can not use linear cryptanalysis to attack the cipher.

However, if we consider all paths, the cases will be very complicated and it is difficult to decide the effectiveness of the linear cryptanalysis with the linear hull.

References

1. Matsui, M.: Linear Cryptoanalysis Method for DES Cipher. Advances in Cryptology, Eurocrypt 1993, Springer, LNCS 765, pp. 386-397 (1994).
2. Kaliski, B.S. , Robshaw, M.J.B.: Linear Cryptanalysis Using Multiple Approximations. Advances in Cryptology, Crypto 1994, Springer, LNCS 839, pp. 26-39 (1994).
3. Alex, B., Christophe, D.C, Michel Q.: On Multiple Linear Approximations. CRYPTO 2004, Springer, LNCS 3152, pp. 1-22 (2004).

4. Nyberg, K.: Linear Approximation of Block Ciphers. Advances in Cryptology, Eurocrypt 1994, Springer, LNCS 950, pp. 439-444 (1994).
5. Murphy, S.: The Effectiveness of the Linear Hull Effect. Technical Report, RHUL-MA-2009-19, http://www.isg.rhul.ac.uk/∼sean/Linear_Hull.pdf (2009).
6. Ohkuma, K.: Weak keys of Reduced-Round PRESENT for Linear Cryptanalysis. SAC 2009, Springer, LNCS 5867, pp. 249-265 (2009).
7. Anderson, R., Biham, E., Knudsen, L.: Serpent: A proposal for the Advanced Encryption Standard. First Advanced Encryption Standard (AES) conference (1998).
8. National Bureau of Standards: FIPS PUB 46-3, Data Encryption Standard (DES), National Institute for Standards and Technology (1977).
9. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an Ultra-Lightweight Block Cipher. CHES 2007, Springer, LNCS 4727, pp. 450C466 (2007)
10. Nakahara, J., Sepehrdad, P., Zhang, B., Wang M.: Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. CANS 2009, Springer, LNCS 5888, pp. 58-75 (2009).
11. Joo, Y.C.: Linear Cryptanalysis of Reduced-Round PRESENT. CT-RSA 2010, Springer, LNCS 5985, pp. 302-317 (2010).

A The S-box and Permutation Tables of PRESENT

The S-box and the permutation tables of PRESENT are given in Tab. 4 and Tab. 5, respectively.

Table 4. S-box Table in Hexadecimal Notation

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

Table 5. Permutation Table

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

B Bias of 4-Round Linear Hull of PRESENT

For block cipher PRESENT, the data mask we used is $(00000000_x||00300000_x, 00400000_x||00400040_x)$, denoted by active bits form as $(\mathrm{P}[20, 21], \mathrm{C}[6, 22, 54])$. There are three linear paths in all:

$$\mathrm{P}[20,21] \oplus \mathrm{C}[6,22,54] = K_0[20,21] \oplus K_1[21] \oplus K_2[37] \oplus K_3[25] \oplus K_4[6,22,54],$$
$$\mathrm{P}[20,21] \oplus \mathrm{C}[6,22,54] = K_0[20,21] \oplus K_1[37] \oplus K_2[41] \oplus K_3[26] \oplus K_4[6,22,54],$$
$$\mathrm{P}[20,21] \oplus \mathrm{C}[6,22,54] = K_0[20,21] \oplus K_1[21,53] \oplus K_2[37,45] \oplus K_3[25,27] \oplus K_4[6,22,54].$$

The biases separately are $\epsilon_1 = -2^{-8}$, $\epsilon_2 = 2^{-8}$ and $\epsilon_3 = 2^{-11}$. Then we have

$$k_1 = K_0[20,21] \oplus K_1[21] \oplus K_2[37] \oplus K_3[25] \oplus K_4[6,22,54],$$
$$k_2 = K_0[20,21] \oplus K_1[37] \oplus K_2[41] \oplus K_3[26] \oplus K_4[6,22,54],$$
$$k_3 = K_0[20,21] \oplus K_1[21,53] \oplus K_2[37,45] \oplus K_3[25,27] \oplus K_4[6,22,54].$$

We do computer simulations with random plaintexts for three different initial 80-bit key. Here we denote the 80-bit master key as κ, and $\kappa[j]$

means the j-th ($0 \leqslant j < 80$) bit of κ, $\kappa[19] = 1$ means only the 19-th bit of κ is 1 and others are 0. Similarly, $\kappa[17] = \kappa[19] = 1$ means the 17-th and the 19-th bit are 1 and the rest are all 0.

Table 6. Bias of 4-Round Linear Hull

Initial Key	Equivalent Keys			Number of Plaintexts	Bias Computed by Equation (3)	Experimental Bias
	k_1	k_2	k_3			
$\kappa[j] = 0,\ 0 \leqslant j < 80$	1	0	1	2^{18}	$2^{-7.0931}$	$2^{-7.0960}$
$\kappa[19] = 1$	1	1	0	2^{25}	2^{-11}	$2^{-10.8513}$
$\kappa[17] = \kappa[19] = 1$	1	0	0	2^{18}	$2^{-6.9125}$	$2^{-6.8961}$

C Computing T_s in Sect. 4

Symbols:
Γ_1: The set of independent subkey bits;
Γ_s: The subset of Γ_1, whose elements have zero value;
T_s^j: The number of dependent subkey bits in which j elements of Γ_s appear, and the number of dependent subkey bits with value zero is

$$T_s = \sum_{1 \leqslant j \leqslant s,\ j \text{ is odd}} T_s^j \ .$$

All biases are computed according to equation (3) in what follows.

1. Let us first consider the case of $s = 1$. It means that there is only one independent subkey bit with value zero, and we denote it as k_j, j is one value of (1,2,...,R).
   If $N_j = 0$, it means that k_j does not appear in any dependent subkey expressions. The number of dependent subkey bits with value zero is $T_1 = 0$, then $\eta = -(L-2)\epsilon$.
   If $N_j = 1$, it means that k_j only appears once in all dependent subkey bits. The number of dependent subkey bits with value zero is $T_1 = 1$, then $\eta = -(L - 2(T_1 + 1))\epsilon = -(L-4)\epsilon$.
   In the similar way, if $N_j = t$, it means that k_j appears t times in all dependent subkey bits. The number of dependent subkey bits with value zero is $T_1 = N_j = t$, then $\eta = -(L-2(T_1+1))\epsilon = -(L-2(N_j+1))\epsilon = -(L-2(t+1))\epsilon$.
   If $s = R - 1$, it means that there is only one independent subkey bit with value one. We also denote it as k_j, j is one value of (1,2,...,R),

that is because $k_j = 0,\ k_l = 1\ (1 \leqslant l \leqslant R,\ l \neq j)$ and $k_j = 1,\ k_l = 0\ (1 \leqslant l \leqslant R,\ l \neq j)$ for j are one to one correspondence. Similar with above process, we know that $T_{R-1} = N_l = t$ and the bias $\eta = (L - 2(T_{R-1} + 1))\epsilon = (L - 2(N_l + 1))\epsilon = (L - 2(t + 1))\epsilon$.

Here we classify T_1 by their value. Let $m_h^{(1)}$ denote the number of possible equivalent subkeys with only one zero independent subkey bit appearing h times in all dependent subkey bits, that is $m_h^{(1)} = \#\{1 \leqslant j \leqslant R \mid T_1 = N_j = h\}$, $0 \leqslant h \leqslant L - R$. *Then* $m^{(1)} = \sum_{h=0}^{L-R} m_h^{(1)}$ *means the total number of* T_1*, and* $T_1 = N_j$ *has* R *possible values.* So $m^{(1)} = \binom{R}{1} = R$. Therefore, the number of equivalent subkeys satisfying $\eta = -(L - 2(h + 1))\epsilon$ is $m_h^{(1)}$. According to symmetry, the number of equivalent subkeys satisfying $\eta = (L - 2(h + 1))\epsilon$ is $m_h^{(1)}$ too.

2. Considering the case of $s = 2$, it is a subset of any two zero or non-zero independent subkey bits $k_u \oplus k_v$ $(1 \leqslant u < v \leqslant R)$.
   As we know, k_u appears N_u times in dependent subkey expressions, and k_v appears N_v times, $k_u \oplus k_v$ appears $N_{u,v}$ times, then the number of dependent subkey expressions which are dependent on k_u but independent on k_v is $N'_u = N_u - N_{u,v}$, similarly the number of dependent subkey expressions which are dependent on k_v but independent on k_u is $N'_v = N_v - N_{u,v}$. Since $k_u = k_v = 0$, the value of dependent subkey bits which is only dependent on one of k_u and k_v is 0, and the number of dependent subkey bits with value zero is $T_2 = N'_u + N'_v = N_u + N_v - 2N_{(u,v)}$. So $\eta = -(L - 2(T_2 + 2))\epsilon$.
   With the method in case 1, let us classify T_2 by its value, and define $m_h^{(2)} = \#\{1 \leqslant u < v \leqslant R \mid T_2 = h\}$, $0 \leqslant h \leqslant L - R$. Then $m^{(2)} = \sum_{h=0}^{L-R} m_h^{(2)} = \binom{R}{2}$ means the total number of T_2.
   So the number of equivalent subkeys satisfying $\eta = -(L - 2(h + 2))\epsilon$ is $m_h^{(2)}$. By symmetry, the number of equivalent subkeys satisfying $\eta = (L - 2(h + 2))\epsilon$ is $m_h^{(2)}$ too.
3. For the general case, we consider the subset of any s independent subkey bits $\Gamma_s = \{k_1, k_2, \ldots, k_s\}$. We need to compute the number of dependent subkey expressions in which odd elements of Γ_s appear.
   The number of dependent subkey expressions in which $k_1 \oplus k_2 \oplus \ldots \oplus k_{s-1}$ appear but k_s does not appear (call the $s - 1$ elements of Γ_s appear independently) is $N'_{1,2,\ldots,(s-1)} = N_{1,2,\ldots,(s-1)} - N_{1,2,\ldots,s}$. The number of dependent subkey expressions in which $k_1 \oplus k_2 \oplus \ldots \oplus k_{s-2}$ appear independently is $N'_{1,2,\ldots,(s-2)} = N_{1,2,\ldots(s-2)} - N'_{1,2,\ldots,(s-2),(s-1)} - N'_{1,2,\ldots,(s-2),s} - N_{1,2,\ldots,s} = N_{1,2,\ldots,(s-2)} - (N_{1,2,\ldots,(s-2),(s-1)} + N_{1,2,\ldots,(s-2),s}) +$

$N_{1,2,\dots,s}$. We can also compute the number of dependent subkey expressions in which any $s-2$ elements of Γ_s appear independently. Then we compute the number of dependent subkey expressions in which any $s-3$ elements of Γ_s appear independently. According to mathematical induction, we get the number of dependent subkey expressions in which one element of Γ_s appears independently

$$N_i' = N_i - \sum_{1 \leqslant j \leqslant s,\ j \neq i} N_{i,j} + \sum_{1 \leqslant j < k \leqslant s,\ j \neq i,\ k \neq i} N_{i,j,k} + \ldots + (-1)^{s-1} N_{(s)}.$$

We know that $N_{(l)}$ is just a symbol and it stands for $\binom{s}{l}$ different values. $N_{(l)}$ in $N_i'(1 \leqslant i \leqslant s)$ are always related with k_i, then the number of $N_{(l)}$ in N_i' is $\binom{s-1}{l-1}$. Therefore, the number of all $N_{(l)}$ in $\sum_{i=1}^{s} N_i'$ is $s \cdot \binom{s-1}{l-1}$. By symmetry, the times of every $N_{(l)}$ appeared in $\sum_{i=1}^{s} N_i'$ is equal. So the coefficient of $\sum N_{(l)}$ in $\sum_{i=1}^{s} N_i'$ is $\frac{s \cdot \binom{s-1}{l-1}}{\binom{s}{l}} = l$. Hence,

$$\begin{aligned} T_s^1 = \sum_{i=1}^{s} N_i' = & \sum_{j=1}^{s} N_j - 2 \sum_{1 \leqslant i < j \leqslant s} N_{i,j} + 3 \sum N_{(3)} \\ & - 4 \sum N_{(4)} + \dots + (-1)^{l-1} \cdot l \cdot \sum N_{(l)} \\ & + \dots + (-1)^{s-1} \cdot s \cdot N_{(s)}. \end{aligned}$$

Similarly, we consider the number of $N_{(l)}$ appeared in $N_{(u)}'(u < l, u$ is an odd). $N_{(l)}$ in $N_{(u)}'$ always related with u elements of Γ_s, then the number of $N_{(l)}$ in $N_{(u)}'$ is $\binom{s-u}{l-u}$. Therefore, the number of all $N_{(l)}$ in $\sum N_{(u)}'$ is $\binom{s}{u} \cdot \binom{s-u}{l-u}$. By symmetry, the times of every $N_{(l)}$ appeared in $\sum N_{(u)}'$ is equal. So the coefficient of $\sum N_{(l)}$ in $\sum N_{(u)}'$ is $\frac{\binom{s}{u} \cdot \binom{s-1}{l-u}}{\binom{s}{l}} = \binom{l}{u}$. Hence,

$$\begin{aligned} T_s^u = \sum N_{(u)}' = \sum N_{(u)} - \binom{u+1}{u} \cdot \sum N_{(u+1)} + \ldots \\ + (-1)^{l-1} \cdot \binom{l}{u} \sum N_{(l)} + \dots + (-1)^{s-1} \cdot \binom{s}{u} \cdot N_{(s)}. \end{aligned}$$

Finally, we get the equation (6)

$$
\begin{aligned}
T_s = & \sum_{1\leqslant j\leqslant s, j \text{ is an odd}} T_s^j \\
= & \sum_{j=1}^{s} N_j' + \sum N_{(3)}' + \sum N_{(5)}' + \dots \\
= & \sum_{j=1}^{s} N_j - 2 \sum_{1\leqslant i<j\leqslant s} N_{i,j} + 4 \sum N_{(3)} - 8 \sum N_{(4)} + \dots \\
& - (2l + \binom{2l}{3} + \binom{2l}{5} + \dots + \binom{2l}{2l-1}) \cdot \sum N_{(2l)} \\
& + (2l+1 + \binom{2l+1}{3} + \binom{2l+1}{5} + \dots + \binom{2l+1}{2l+1}) \cdot \sum N_{(2l+1)} \\
& + \dots + (-1)^{s-1}(s + \binom{s}{3} + \binom{s}{5} + \dots) \cdot N_{(s)}.
\end{aligned}
$$