# Linear (hull) Cryptanalysis of Round-reduced Versions of KATAN

Danping Shi[1,2,3], Lei Hu[1,2⋆], Siwei Sun[1,2], Ling Song[1,2]

[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
[2]Data Assurance and Communication Security Research Center, Chinese Academy of Sciences, Beijing 100093, China
[3]University of Chinese Academy of Sciences, Beijing 100093, China
{dpshi, hu, swsun, lsong}@is.ac.cn

**Abstract.** KATAN is a family of block ciphers published at CHES 2009. Based on the Mixed-integer linear programming (MILP) technique, we propose the first third-party linear cryptanalysis on KATAN. Furthermore, we evaluate the security of KATAN against the linear attack without ignoring the dependence of the input bits of the $2 \times 1$ S-box(the AND operation). Note that in previous analysis, the dependence is not considered, and therefore the previous results are not accurate. Furthermore, the mounted 131/120-round attack on KATAN32/48 respectively by our 84/90-round linear hull is the best single-key known-plaintext attack. In addition, a best 94-round linear hull attack is mounted on KATAN64 by our 76-round linear hull.

**Keywords.** KATAN, Mixed-integer linear programming, linear hull, dependence

## INTRODUCTION

Demands for lightweight ciphers used in resource-constrained devices with low cost are increasing in recent years. Many lightweight block ciphers are published in recent years, such as LBlock [1], PRESENT [2], LED [3], PRIDE [4] and SIMON [5].

**Related works**

KATAN is a family of lightweight block ciphers published at CHES 2009 [6]. After its publication, KATAN receives extensive cryptanalysis. For instance, the conditional differential cryptanalysis by Knellwolf et al. [7] on 78/70/68-round KATAN32/48/64, differential cryptanalysis by Albrecht et al. [8] on 115-round KATAN32, meet-in-the-middle attack by Isobe et al. [9] on 110/100/94-round KATAN32/48/64, and match box meet-in-the-middle cryptanalysis by Fuhr et al. [10] on 153/129/119-round KATAN32/48/64. All results are presented in Table 1.

Linear cryptanalysis is an important cryptanalysis technique on modern block ciphers [11]. It aims at finding a non-random linear expression on bits of plaintext, ciphertext, and subkey, where the expression has non-zero correlation. The extended linear hull cryptanalysis is presented by Nyberg [12] in 1995. No third-party linear cryptanalysis on KATAN has been proposed. Furthermore, the security evaluation of KATAN with respect to linear cryptanalysis proposed by the designers is not accurate owing to ignoring the dependence of the S-box, where the dependence of S-box means that different S-box share one same input.

**Our contribution**

In this paper, we first evaluate the linear security cryptanalysis on KATAN32 without ignoring the dependence of the S-box based on the Mixed-integer linear programming(MILP) technique [13, 14]. Furthermore, 84/90/76-round linear hulls on KATAN32/48/64 respectively are proposed. Moreover, 131/120/94-round attack on KATAN32/48/64 are mounted by these linear hulls. A comparison between this paper and other single-key attacks is listed in Table 1. Although, cryptanalysis provided by paper [10] can attack more rounds, their cryptanalysis is based on stricter chosen-plaintext model. As we know, the 131/120-round attacks on KATAN32/48 respectively in this paper are the best single-key known-plaintext attacks, and our 94-round attack on KATAN64 is the first linear attack on KATAN64.

The paper is organized as follows. Section 2 proposes the brief description of KATAN. Section 3 shows the searching method of linear masks. The results about the linear (hull) cryptanalysis are described in Section 4. Section 5 is the conclusion.

**Table 1.** The analysis results of KATAN based on single-key

| Version | Cryptanalysis method | Model | Rounds | Data | Time | Reference |
|---|---|---|---|---|---|---|
| KATAN32 | Differential | CP | 78 | $2^{22}$ | $2^{22}$ | [7] |
| | Differental | CP | 115 | $2^{32}$ | $2^{79}$ | [8] |
| | Match box MITM | CP | 153 | $2^{5}$ | $2^{78.5}$ | [10] |
| | MITM | KP | 110 | 138 | $2^{77}$ | [9] |
| | Match box MITM | KP | 121 | 4 | $2^{77.5}$ | [10] |
| | Linear hull | KP | 131 | $2^{28.93}$ | $2^{78.93}$ | Section 0.7 |
| KATAN48 | Differential | CP | 70 | $2^{31}$ | $2^{78}$ | [7] |
| | Match box MITM | CP | 129 | $2^{5}$ | $2^{76}$ | [10] |
| | MITM | KP | 100 | 128 | $2^{78}$ | [9] |
| | Match box MITM | KP | 110 | 4 | $2^{77.5}$ | [10] |
| | Linear hull | KP | 120 | $2^{47.22}$ | $2^{75.22}$ | Section 0.7 |
| KATAN64 | Differential | CP | 68 | $2^{32}$ | $2^{78}$ | [7] |
| | Match box MITM | CP | 119 | $2^{5}$ | $2^{78.5}$ | [10] |
| | MITM | KP | 94 | 116 | $2^{77.68}$ | [9] |
| | Match box MITM | KP | 102 | 4 | $2^{77.5}$ | [10] |
| | Linear hull | KP | 94 | $2^{57}$ | $2^{78}$ | Section 0.7 |

CP: chosen-plaintext attack; KP: known-plaintext attack

# BRIEF DESCRIPTION OF KATAN

KATAN is a family of block ciphers with 32, 48, or 64-bit block length, denoted by KATAN32, KATAN48 or KATAN64 respectively. All versions share the same 80-bit master key. For each version, the plaintext is loaded in two registers $L_1$ and $L_2$, where the lengths of $L_1$ and $L_2$ for each version are listed in Table 2. In the first place, the round function for KATAN32 is illustrated. For KATAN32, the registers $L_1$ and $L_2$ are shifted to the left by 1 position, and two new computed bits by two nonlinear functions $f_a(\cdot)$ and $f_b(\cdot)$ are loaded in the least significant bits of $L_1$ and $L_2$, where the least significant(rightmost) bit for each register will be denoted as 0-th bit. The ciphertext is obtained after 254 rounds. The $f_a$ and $f_b$ are defined as follows

$$\begin{aligned} f_a(L_1) &= L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \wedge L_1[x_4]) \oplus \\ &\quad (L_1[x_5] \wedge IR) \oplus k_a, \\ f_b(L_2) &= L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \wedge L_2[y_4]) \oplus \\ &\quad (L_2[y_5] \wedge L_2[y_6]) \oplus k_b, \end{aligned}$$

where $IR$ is round constant, $k_a$ and $k_b$ are two subkey bits. The index $x_i$ and $y_i$ are listed in Table 2.

**Table 2.** The parameters for KATAN

| version | $\|L_1\|$ | $\|L_2\|$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| KATAN32 | 13 | 19 | 12 | 7 | 8 | 5 | 3 | 18 | 7 | 12 | 10 | 8 | 3 |
| KATAN48 | 19 | 29 | 18 | 12 | 15 | 7 | 6 | 28 | 19 | 21 | 13 | 15 | 6 |
| KATAN64 | 25 | 39 | 24 | 15 | 20 | 11 | 9 | 38 | 25 | 33 | 21 | 14 | 9 |

For KATAN48, the shift update of the registers and the nonlinear function $f_a$, $f_b$ are applied twice with same round subkey in each round, while the nonlinear functions and update of the register are applied three times for KATAN64.

Since we only consider single-key cryptanalysis in this paper, and therefore the key schedule is omitted here. More details on KATAN can be found in paper [6].

# THE LINEAR CRYPTANALYSIS OF KATAN

## 0.1 Notations

$L_1^r[i]$: the $i$-th bit of the register $L_1$ in $r$-th round
$L_2^r[i]$: the $i$-th bit of the register $L_2$ in $r$-th round
$k_a^r$: the $r$-th round subkey used in $f_a$
$k_b^r$: the $r$-th round subkey used in $f_b$
$\alpha_t$: the masks of the variable $t$

### 0.2 Definition of the linear cryptanalysis

Denote $f$ be a boolean function, the correlation $\epsilon_f$ of $f$ is defined by

$$Pr(f(x) = 0) - Pr(f(x) = 1).$$

The linear cryptanalysis is evaluated by the correlation.

The *potential* introduced by Nyberg[12] is used to evaluate the linear hull cryptanalysis. Give the input and output masks $\alpha$ and $\beta$ for a block cipher $C = f(P, K)$, the *potential* $ALH(\alpha, \beta)$ is defined by

$$ALH(\alpha, \beta) = \sum_{\gamma} (Pr(\alpha \cdot P + \beta \cdot C + \gamma \cdot K = 0) - 1/2)^2.$$

### 0.3 Dependence of S-box

For simplicity, each AND operation $\wedge$ is treated as a $2 \times 1$ S-box. The S-box is active if the output mask is non-zero. Since $IR$ is constant, $L_1[x_5] \wedge IR$ is a linear operation in each round, not a S-box. In this paper, the dependence of two S-box illustrates that the two S-box share one input. Owing to the fact that only few bits are registered in each round, S-box for KATAN are dependent.

Usually, the correlation of a linear characteristic for a block cipher is obtained from the correlation of the approximation of round function by pilling-up lemma. Whereas, the pilling-up lemma is not suitable for KATAN due to the dependence of the S-box.

For instance, suppose two approximations of two S-box of $L_1$ in 0-th round and 4-th round, both with zero input mask and non-zero output mask, are $L_1^0[5] \wedge L_1^0[8]$ and $L_1^4[5] \wedge L_1^4[8]$. Clearly, each approximation has the same correlation(absolute) $2^{-1}$. The correlation of XOR-ed function $L_1^0[5] \wedge L_1^0[8] + L_1^4[5] \wedge L_1^4[8]$ of the two approximations is $2^{-2}$ if applying pilling-up lemma. However, the correlation of $L_1^0[5] \wedge L_1^0[8] + L_1^4[5] \wedge L_1^4[8] = L_1^0[5] \wedge (L_1^0[8] + L_1^4[5])$ is $2^{-1}$ due to $L_1^4[8] = L_1^0[5]$.

The above example shows that the dependence of the S-box should be taken into consideration when computing the correlation. Consequently, the correlation of the linear characteristic will be computed directly instead of applying the pilling-up lemma in this paper. The computing method in the following is similar to paper[13, 14].

Obviously, the XOR-ed function of all approximations for active S-box is a quadratic function. Denote quadratic boolean function $f(t) = Q(t) + L(t)$, where $t = (t[1], t[2], \cdots, t[n]) \in \mathbb{F}_2^n$, $Q(t) = t[i_1] \wedge t[i_2] + t[i_3] \wedge t[i_4] + \cdots + t[i_{m-1}] \wedge t[i_m]$ is the sum of quadratic term $t[i] \wedge t[j]$, and $L(t) = t[j_1] + t[j_2] + \cdots + t[j_{n-1}] + t[j_n]$ is linear combination of $t[i]$. This kind of function satisfying the property that $i_1, i_2, i_3, i_4, \cdots, i_{m-1}, i_m$ are not coincident is called *the standard quadratic function* in the following. Most important, the correlation $\epsilon_f$ of the standard function can be obtained directly as follow:

- $\{j_1, j_2, \cdots, j_n\} \subseteq \{i_1, i_2, \cdots, i_m\}$: $\epsilon_f = 2^{-m/2}$.

– others: $\epsilon_f = 0$.

In other words, if the correlation of the standard function is non-zero, there is a negative correlation between the correlation and the amount of the variables existing in the quadratic terms. Moreover, for any quadratic function, there exists a non-singular transform $s = A \cdot t$ such that $g(s) = f(A^{-1} \cdot s) = Q(s) + L(s)$ is the standard form of $f$. What is more, the correlation of the standard function $g$ equals to that of $f$.

For instance, $f(t) = t[1] \wedge t[2] + t[1] \wedge t[3] + t[2] \wedge t[4] + t[2]$. Suppose non-singular transform $s[1] = t[1] + t[4], s[2] = t[2] + t[3], s[3] = t[3], s[4] = t[4]$, therefore the standard form $g(s) = s[1] \wedge s[2] + s[3] \wedge s[4] + s[2] + s[3]$. Hence, the correlation of $f$ is obtained from $g$ by the above method, which is $2^{-2}$, due to $\{2, 3\} \subseteq \{1, 2, 3, 4\}$.

In brief, three steps for computing the correlation of a linear characteristic are applied. Firstly, obtain the XOR-ed function of all approximations of each active S-box. Secondly, derive the stand form of the XOR-ed function. Finally, calculate the correlation of the standard form by the above method. The calculating method is also suitable for other ciphers with the similar S-box of KATAN, such as SIMON.

### 0.4 Automatic enumeration of characteristic with MILP

Similar with paper [13–15], we obtain the linear characteristic by the automatic enumeration with Mixed-integer linear Programming Modelling(MILP). The method denotes each mask bit by a 0-1 variable, then describes the propagation of the masks by linear constraints and optimizes a objective function. Constrains for linear operations are similar to paper [13–15]. Following is the MILP modelling for searching the linear characteristic, where $\alpha_t$ denotes the mask for variable $t$.

**Constraints for linear operations**

Constraints for bitwise XOR and branching structure are same with paper [13, 16] in the following.

1. For XOR operation $z = x \oplus y$, their masks satisfy $\alpha_x = \alpha_y = \alpha_z$.
2. For three branching structure $z = x = y$, their masks satisfy

$$\begin{cases} \tau \geq \alpha_x, \tau \geq \alpha_y, \tau \geq \alpha_z, \\ \alpha_x + \alpha_y + \alpha_z \geq 2\tau, \\ \alpha_x + \alpha_y + \alpha_z \leq 2, \end{cases}$$

where $\tau$ is the introduced new dummy variable.

**Constraints for S-box**

For S-box $z = x \wedge y$, their masks satisfy $2\alpha_z \geq \alpha_x + \alpha_y$.

**Constraints dealing with dependence of S-box**

In order to consider the dependence of the S-box, the $|L_1| + |L_2|$ initial variables of registers and the two new registered variables each round loaded in the

LSB of registers are treated as original variables. In this case, the XOR-ed function of approximations for each active S-box can be expressed as a quadratic function of these original 0-1 variables. Furthermore, there is a negative correlation between the correlation (non-zero) of the standard form and the number of the variables existing in the quadratic terms as shown in Section 0.3. Usually, the more variables exist in the quadratic terms of a boolean function, the more variables exist in that of its standard form. As a consequence, the amount of the variables in the quadratic terms is chosen as the preliminary measure of the correlation. On the other hand, the fact that one original variable exists in the quadratic terms is equivalent to the thing that this variable is the input of active S-box. Accordingly, the amount of all original variables as inputs of active S-box is the our preliminary measure of the correlation.

For each original variable $t$, denote a new 0-1 variable $V_t$ to indicate whether the variable $t$ is the input of one active S-box, where $V_t = 1$ if it is. In this case, $\sum_{t\in\mathbb{A}} V_t$ is our preliminary measure, where $\mathbb{A}$ denotes the set consists of all original variables.

Furthermore, each variable $t$ may be one input of several S-box (suppose $n_t$), which means these $n_t$ S-box are dependent as previous shows. For instance, the original variable $L_1^0[5]$ for KATAN32 affects 2 S-box, 0-th and 4-th round S-box of $L_1$. What is more, if all the $n_t$ S-box are not active, $V_t = 0$; otherwise $V_t = 1$. This property for each original variable can be described by following constraints:

$$n_t \cdot V_t \geq \beta_1 + \beta_2 + \cdots + \beta_{n_t},$$
$$\beta_1 + \beta_2 + \cdots + \beta_{n_t} \geq V_t,$$

where $\beta_i, i \in 1, \cdots, n_t$, are output masks of the $n_t$ S-box, and also express whether these S-box are active.

**Objective function**

As previous shows, $\sum_{t\in\mathbb{A}} V_t$ is chosen as our preliminary measure of the correlation. Usually, the more variables exist in the quadratic terms, the smaller the correlation is. Therefore the objective function is to minimize $\sum_{t\in\mathbb{A}} V_t$.

### 0.5 The computation of *potential*

For each version, we will obtain the linear hull by previous methods. In the first place, obtain a linear characteristic with high correlation by Gurobi software, with MILP modeling presented in above Section 0.4. Secondly, search again to obtain as many as possible suitable characteristics with additional constraints of fixing the input and output masks equaling to that of the obtained linear characteristic with high correlation. Finally, obtain the correlation for each characteristic by the computing method shown in previous Section 0.3, then give the *potential*.

# RESULTS

The linear cryptanalysis shown by the designers [6] did not consider the dependence of the S-box. The 126-round linear characteristic eliminated by 42-round linear approximation is not accurate, due to the dependence of the S-box. With taking the dependence of S-box into consideration, we obtain some new results for the security cryptanalysis. Furthermore, some linear hulls with high *potential* are obtained by the MILP modelling shown in Section 0.4. What is more, we mount some best attacks by these linear hulls.

## 0.6 Results for linear characteristic

For KATAN32, the correlation for the best 42-round linear approximation is $2^{-5}$ according to the designers [6], with ignoring the effect of the dependence of S-box. In the case of taking the dependence of S-box into consideration, we evaluate the security of the linear cryptanalysis again. The correlation of the best 42-round linear approximation is still $2^{-5}$ by our method. At the same time, a best 84-round linear characteristic with correlation $2^{-15}$ is presented in Appendix, while the previous 84-round linear characteristic is directly eliminated as no more than $2^{-5*2}$ by pilling-up lemma according to the designers. The obtained 84-round linear characteristic demonstrates that KATAN32 is secure against linear cryptanalysis based on one linear characteristic, however we can mount linear hull attack on KATAN in the following.

For KATAN48/64, a best 43/37-round respectively linear characteristic with correlation $2^{-8}/2^{-10}$ is also obtained in this paper under the consideration of the dependence of the S-box, while the previous 43/37-round linear characteristic for KATAN48/64 provided by designers has correlation $2^{-9}/2^{-10}$. Due to the limitation of computing resources, the more accurate security analysis for KATAN48/64 are not obtained. The masks are listed in Appendix.

## 0.7 Results for linear hull

By setting the input and output masks presented in Table 3, linear hulls for some versions are obtained with some additional constraints owing to the limitation of computing resource. Moreover, some best attacks are mounted by these linear hulls.

For KATAN32, a 84-round linear hull consisting of 98264 linear characteristics with *potential* $2^{-27.93}$ is obtained with additional constraints $\sum_{t\in\mathbb{A}} V_t \leq 44$. In addition, a 131-round attack with 21-round forward and 26-round backward is mounted by this linear hull. In the attack process, 50-bit subkey require to be guessed, while another 10-bit subkey with linear effect to the linear approximation do not. The involved subkey bits in the attack process are listed in Table 4, and the involved bits of the registers are listed in Table 5.

For KATAN48, a 90-round linear hull consisting of 99434 linear characteristics with *potential* $2^{-46.22}$ is obtained with additional constraints $\sum_{t\in\mathbb{A}} V_t \leq 65$. Furthermore, we mount a 120-round attack with 16-round forward and 14-round

**Table 3.** The input and output masks for linear hull

| version | Input masks of register $L_1$ | Input masks of register $L_2$ |
|---|---|---|
| KATAN32 | 1000010001000 | 0000000000000010000 |
| KATAN48 | 0000000000000000000 | 00000000000000000100000000100 |
| KATAN64 | 0100000000100000100000000 | 000000000000000000100000000100010000000 |
| version | output masks of register $L_1$ | output masks of register $L_2$ |
| KATAN32 | 0010000000000 | 1000000000000000000 |
| KATAN48 | 1000000001000000001 | 00000000100000000000000000000 |
| KATAN64 | 1000001001001000100000000 | 100010000100000100101000100100000100100 |

backward by this linear hull. In the attack process, 28-bit subkey require to be guessed, while another 8-bit subkey do not. The involved subkey bits in the attack process are listed in Table 4, and the involved bits of the registers are listed in Table 6.

For KATAN64, a 76-round linear hull consisting of 82908 linear characteristics with *potential* $2^{-57}$ is obtained with additional constraints $\sum_{t\in\mathbb{A}} V_t \leq 77$. What is more, we mount a 94-round attack with 12-round forward and 6-round backward by this linear hull. In the attack process, 20-bit subkey require to be guessed, while another 13-bit subkey do not. The involved subkey bits in the attack process are listed in Table 4, and the involved bits of the registers are listed in Table 7.

The constraints $\sum_{t\in\mathbb{A}} V_t \leq 44/65/77$ for KATAN32/48/64 respectively is added due to the limitation of computing resource. The data complexity $N$ of the linear hull attack is set by $2 * ALH^{-1}$. Suppose the length of the guessed-key is $l_k$, thus the time complexity is $N * 2^{l_k}$. The complexity is summarized in Table 1.

**Table 4.** The involved subkey bits in the attack process

| version | round of guessed-key for $k_a$ |
|---|---|
| KATAN32 | 13, 9, 8, 6, 4, 3, 2, 1, 0, 111, 114, 115, 116, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130 |
| KATAN48 | 7, 4, 3, 2, 1, 0, 107, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, |
| KATAN64 | 5, 4, 3, 2, 1, 0, 89, 90, 91, 92,93 |
| version | round of guessed-key for $k_b$ |
| KATAN32 | 10, 7, 5, 4, 3, 2, 1, 0, 111, 113, 115, 117, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130 |
| KATAN48 | 10, 6, 3, 2, 1, 0, 113, 116, 117, 118 |
| KATAN64 | 5, 4, 2, 1, 0, 89, 91, 92, 93 |
| version | non-guessed key |
| KATAN32 | $k_a^5, k_a^{16}, k_a^{107}, k_a^{112}, k_a^{117}, k_b^8, k_b^{13}, k_b^{17}, k_b^{105}, k_b^{116}$ |
| KATAN48 | $k_a^{10}, k_a^{14}, k_a^{106}, k_b^5, k_b^8, k_b^{110}, k_b^{114}, k_b^{119}$ |
| KATAN64 | $k_a^5, k_a^8, k_a^9, k_a^{88}, k_a^{90}, k_a^{91}, k_a^{93}, k_b^2, k_b^7, k_b^9, k_b^{88}, k_b^{89}, k_b^{92}$ |

## CONCLUSION

We first propose a third-party linear cryptanalysis on KATAN in this paper. What is more, we first take the dependence of the S-box into the analysis for KATAN. At the same time, we evaluate the security analysis on KATAN32 in the

**Table 5.** The involved bits of the registers for KATAN32

| Round | bits of register $L_1$ | bits of register $L_2$ |
|---|---|---|
| 0 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 1 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 16, 17, 18 |
| 2 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18 |
| 3 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18 |
| 4 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 17, 18 |
| 5 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15, 16, 18 |
| 6 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16, 17 |
| 7 | 1, 2, 3, 4, 5, 6, 7, 10, 11 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 17, 18 |
| 8 | 0, 2, 3, 4, 5, 6, 7, 8, 11, 12 | 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, 16, 18 |
| 9 | 0, 1, 3, 4, 5, 7, 8, 12 | 0, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 17 |
| 10 | 1, 2, 4, 5, 6, 9 | 0, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15, 18 |
| 11 | 0, 2, 3, 5, 6, 7, 10 | 16, 1, 2, 4, 5, 6, 8, 10, 12 |
| 12 | 1, 3, 4, 6, 7, 8, 11 | 17, 2, 3, 5, 6, 7, 9, 11, 13 |
| 13 | 2, 4, 5, 7, 8, 9, 12 | 18, 3, 4, 6, 7, 8, 10, 12, 14 |
| 14 | 0, 10, 3, 5, 6 | 0, 4, 5, 7, 9, 15 |
| 15 | 1, 11, 4, 6, 7 | 16, 1, 5, 6, 8, 10 |
| 16 | 8, 2, 12, 5, 7 | 17, 2, 6, 7, 9, 11 |
| 17 | 8, 3 | 0, 18, 3, 7, 8, 10, 12 |
| 18 | 0, 9, 4 | 1 |
| 19 | 1, 10, 5 | 2 |
| 20 | 2, 11, 6 | 3 |
| 21 | 3, 12, 7 | 4 |
| 105 | 10 | 18 |
| 106 | 0, 11 | 8, 9, 11, 4, 13 |
| 107 | 1, 12 | 9, 10, 12, 5, 14 |
| 108 | 8, 9, 2, 6 | 0, 6, 10, 11, 13, 15 |
| 109 | 9, 10, 3, 7 | 16, 1, 7, 11, 12, 14 |
| 110 | 8, 10, 11, 4 | 17, 2, 8, 12, 13, 15 |
| 111 | 9, 11, 12, 5 | 16, 18, 3, 9, 13, 14 |
| 112 | 0, 6, 8, 9, 10, 12 | 0, 4, 8, 9, 10, 11, 13, 14, 15, 17 |
| 113 | 1, 4, 6, 7, 8, 9, 10, 11 | 0, 1, 5, 9, 10, 11, 12, 14, 15, 16, 18 |
| 114 | 0, 2, 5, 7, 8, 9, 10, 11, 12 | 1, 2, 4, 6, 8, 9, 10, 11, 12, 13, 15, 16, 17 |
| 115 | 1, 3, 4, 6, 8, 9, 10, 11, 12 | 0, 2, 3, 5, 7, 9, 10, 11, 12, 13, 14, 16, 17, 18 |
| 116 | 0, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18 |
| 117 | 0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18 |
| 118 | 0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 |
| 119 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 120 | 0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 121 | 0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 122 | 0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 123 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 124 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 125 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 126 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 127 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 128 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 129 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 130 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 131 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |

**Table 6.** The involved bits of the registers for KATAN48

| Round | bits of register $L_1$ |
|---|---|
| 0 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 1 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18 |
| 2 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 |
| 3 | 1, 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 17, 18 |
| 4 | 0, 1, 3, 5, 6, 9, 11, 12, 14, 17 |
| 5 | 2, 3, 5, 8, 11, 14 |
| 6 | 16, 1, 4, 5, 7, 10, 13 |
| 7 | 0, 1, 3, 6, 7, 9, 12, 15, 18 |
| 8 | 2, 3, 5, 8, 11, 14 |
| 9 | 16, 1, 4, 5, 7, 10, 13 |
| 10 | 18, 3, 6, 7, 9, 12, 15 |
| 11 | 8, 0, 11, 5 |
| 12 | 10, 7, 2, 13 |
| 13 | 4, 15, 12, 9 |
| 14 | 17, 11, 6, 14 |
| 15 | - |
| 16 | - |
| 106 | 0, 9, 18 |
| 107 | 17, 2, 11, 14, 9 |
| 108 | 16, 8, 11, 4, 13 |
| 109 | 18, 13, 10, 6, 15 |
| 110 | 17, 8, 9, 12, 14, 15 |
| 111 | 1, 7, 8, 10, 11, 13, 14, 16, 17 |
| 112 | 3, 8, 9, 10, 12, 13, 15, 16, 18 |
| 113 | 5, 8, 9, 10, 11, 12, 14, 15, 17, 18 |
| 114 | 0, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17 |
| 115 | 0, 2, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18 |
| 116 | 2, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 117 | 0, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 118 | 0, 1, 2, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 119 | 1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| 120 | 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 |
| Round | bits of register $L_2$ |
| 0 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 27, 28 |
| 1 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 26, 28 |
| 2 | 0, 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 28 |
| 3 | 0, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 24, 27, 28 |
| 4 | 0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 23, 24, 26 |
| 5 | 0, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 21, 22, 25, 26, 28 |
| 6 | 2, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 24, 27, 28 |
| 7 | 4, 6, 8, 11, 12, 13, 14, 17, 19, 21, 26 |
| 8 | 1, 6, 8, 10, 13, 14, 15, 16, 19, 21, 23, 28 |
| 9 | 16, 18, 3, 25, 10, 12 |
| 10 | 18, 20, 5, 27, 12, 14 |
| 11 | 1 |
| 12 | 3 |
| 13 | 5 |
| 14 | 7 |
| 15 | 0, 9 |
| 16 | 2, 11 |
| 106 | 20 |
| 107 | 1, 22 |
| 108 | 0, 24, 3 |
| 109 | 2, 26, 5 |
| 110 | 1, 4, 28, 7 |
| 111 | 0, 17, 3, 21, 6, 23, 8, 9, 15 |
| 112 | 0, 2, 5, 8, 10, 11, 17, 19, 23, 25 |
| 113 | 1, 2, 4, 7, 10, 12, 13, 19, 21, 25, 27 |
| 114 | 0, 1, 3, 4, 6, 7, 9, 12, 14, 15, 16, 20, 21, 22, 23, 27 |
| 115 | 0, 2, 3, 5, 6, 7, 8, 9, 11, 14, 16, 17, 18, 20, 22, 23, 24, 25 |
| 116 | 1, 2, 4, 5, 7, 8, 9, 10, 11, 13, 16, 18, 19, 20, 22, 24, 25, 26, 27 |
| 117 | 0, 1, 3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 21, 22, 24, 26, 27, 28 |
| 118 | 0, 1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 26, 28 |
| 119 | 0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28 |
| 120 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28 |

**Table 7.** The involved bits of the registers $L_1$ for KATAN64

| Round | bits of register $L_1$ |
|---|---|
| 0 | 0, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 18, 19, 22, 23 |
| 1 | 0, 1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 21, 22 |
| 2 | 0, 2, 3, 4, 5, 6, 9, 10, 11, 13, 14, 15, 18, 19, 20, 24 |
| 3 | 0, 1, 3, 5, 6, 7, 8, 9, 12, 13, 14, 16, 17, 18, 21, 22 |
| 4 | 3, 4, 6, 8, 9, 10, 11, 12, 15, 17, 19, 20, 21, 24 |
| 5 | 2, 6, 7, 9, 11, 13, 15, 18, 20, 22, 24 |
| 6 | 1, 18, 5, 9, 10, 14 |
| 7 | 17, 4, 21, 8, 12, 13 |
| 8 | 16, 2, 20, 7, 24, 11, 15 |
| 9 | 10, 19, 5, 14, 23 |
| 10 | 8, 17, 2 |
| 11 | 11, 20, 5 |
| 12 | 8, 14, 23 |
| 88 | 8, 24, 18, 12, 15 |
| 89 | 2, 21, 23, 11, 18, 14, 15 |
| 90 | 0, 1, 5, 13, 14, 17, 18, 21, 22, 24 |
| 91 | 3, 4, 8, 12, 14, 16, 17, 18, 20, 21, 23, 24 |
| 92 | 1, 2, 6, 7, 11, 12, 13, 14, 15, 17, 18, 19, 20, 21, 22, 23, 24 |
| 93 | 0, 1, 4, 5, 9, 10, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24 |
| 94 | 0, 2, 3, 4, 7, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 |
| Round | bits of register $L_2$ |
| 0 | 0, 1, 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 30, 31, 32, 33, 35, 36, 37, 38 |
| 1 | 0, 1, 3, 4, 5, 7, 8, 9, 10, 12, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 28, 29, 31, 33, 34, 36, 38 |
| 2 | 0, 3, 4, 6, 7, 8, 10, 11, 12, 13, 15, 17, 18, 19, 20, 23, 24, 27, 28, 31, 32, 36, 37 |
| 3 | 2, 3, 6, 7, 9, 11, 13, 14, 15, 18, 35, 20, 21, 22, 26, 30, 31 |
| 4 | 0, 5, 6, 9, 10, 12, 14, 16, 17, 18, 21, 23, 24, 25, 29, 33, 34, 38 |
| 5 | 32, 2, 3, 37, 8, 9, 13, 15, 19, 20, 21, 24, 26, 27 |
| 6 | 0, 2, 35, 5, 6, 11, 12, 16, 18, 22, 24, 29, 30 |
| 7 | 32, 33, 3, 5, 38, 8, 9, 14, 15, 19, 21, 25, 27 |
| 8 | 18, 35, 22, 6, 8, 11, 30 |
| 9 | 33, 2, 38, 9, 11, 14, 21, 25 |
| 10 | 1, 5, 14 |
| 11 | 8, 17, 4 |
| 12 | 11, 20, 7 |
| 88 | 2, 5, 38, 11, 34, 14, 18, 20, 23, 29 |
| 89 | 32, 2, 36, 37, 8, 12, 14, 17, 21, 23, 24, 26, 28, 5 |
| 90 | 1, 34, 35, 5, 8, 10, 11, 15, 16, 17, 20, 22, 23, 24, 26, 27, 29, 31 |
| 91 | 0, 32, 34, 4, 37, 38, 8, 11, 13, 14, 18, 19, 20, 23, 2, 25, 26, 27, 29, 30 |
| 92 | 1, 2, 3, 5, 7, 11, 12, 14, 16, 17, 21, 22, 23, 24, 26, 27, 28, 29, 30, 32, 33, 35, 36, 37 |
| 93 | 0, 1, 2, 4, 5, 6, 8, 10, 11, 14, 15, 16, 17, 19, 20, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 38 |
| 94 | 0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 32, 33, 34, 35, 36, 37, 38 |

case of taking the dependence of the S-box into consideration. Furthermore, the 131/120-round attack mounted by our linear hull on KATAN32/48 respectively is the best single-key known-plaintext attack for KATAN.

## References

1. Wenling Wu and Lei Zhang. Lblock: A lightweight block cipher. In Javier Lopez and Gene Tsudik, editors, *Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*, pages 327–344. Springer Berlin Heidelberg, 2011.
2. A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer Berlin Heidelberg, 2007.
3. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems ?CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer Berlin Heidelberg, 2011.
4. Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin. Block ciphers - focus on the linear layer (feat. PRIDE). In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 57–76, 2014.
5. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *IACR Cryptology ePrint Archive*, 2013:404, 2013.
6. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, pages 272–288, 2009.
7. Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional differential cryptanalysis of nlfsr-based cryptosystems. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 130–145, 2010.
8. Martin R. Albrecht and Gregor Leander. An all-in-one approach to differential cryptanalysis for small block ciphers. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, pages 1–15, 2012.
9. Takanori Isobe and Kyoji Shibutani. All subkeys recovery attack on block ciphers: Extending meet-in-the-middle approach. In *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, pages 202–221, 2012.
10. Thomas Fuhr and Brice Minaud. Match box meet-in-the-middle attack against KATAN. In *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*, pages 61–81, 2014.
11. Mitsuru Matsui. Linear cryptanalysis method for des cipher. In Tor Helleseth, editor, *Advances in Cryptology ?EUROCRYPT ?3*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer Berlin Heidelberg, 1994.
12. Kaisa Nyberg. Linear approximation of block ciphers. In *Advances in CryptologyEUROCRYPT'94*, pages 439–444. Springer, 1995.

13. Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. *IACR Cryptology ePrint Archive*, 2014:747, 2014.
14. Danping Shi, Lei Hu, Siwei Sun, Ling Song, Kexin Qiao, and Xiaoshuang Ma. Improved linear (hull) cryptanalysis of round-reduced versions of SIMON. *IACR Cryptology ePrint Archive*, 2014:973, 2014.
15. 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 Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology ?ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*, pages 158–178. Springer Berlin Heidelberg, 2014.
16. Andrey Bogdanov and Vincent Rijmen. Linear hulls with correlation zero and linear cryptanalysis of block ciphers. *Designs, Codes and Cryptography*, 70(3):369–383, 2014.

# APPENDIX

**Table 8.** The 84-round linear characteristic for KATAN32

| Round | input masks of register $L_1$ | input masks of register $L_2$ |
|---|---|---|
| 0 | 1000010001000 | 0000000000000010000 |
| 1 | 0000000000000 | 0000000000000100001 |
| 2 | 0000000000000 | 0000000000001000010 |
| 3 | 0000000000000 | 0000000000010000100 |
| 4 | 0000000000000 | 0000000000100001000 |
| 5 | 0000000000000 | 0000000001000010000 |
| 6 | 0000000000000 | 0000000010000100000 |
| 7 | 0000000000000 | 0000000100001000000 |
| 8 | 0000000000000 | 0000001000010000000 |
| 9 | 0000000000000 | 0000010000100000000 |
| 10 | 0000000000000 | 0000100001000000000 |
| 11 | 0000000000000 | 0001000010000000000 |
| 12 | 0000000000000 | 0010000100000000000 |
| 13 | 0000000000000 | 0100001000000000000 |
| 14 | 0000000000000 | 1000010000000000000 |
| 15 | 0000000000001 | 0000100000100000000 |
| 16 | 0000000000010 | 0001000001000000000 |
| 17 | 0000000000100 | 0010000010000000000 |
| 18 | 0000000001000 | 0100000100000000000 |
| 19 | 0000000010000 | 1000001000000000000 |
| 20 | 0000000100001 | 0000010000100000000 |
| 21 | 0000001000010 | 0000100001000000000 |
| 22 | 0000010000100 | 0001000010000000000 |
| 23 | 0000100001000 | 0010000100000000000 |
| 24 | 0001000010000 | 0100001000000000000 |
| 25 | 0010000100000 | 1000010000000000000 |
| 26 | 0100001000001 | 0000100000100000000 |
| 27 | 1000010000010 | 0001000001000000000 |
| 28 | 0000000000100 | 0010000010000000001 |
| 29 | 0000000001000 | 0100000100000000010 |
| 30 | 0000000010000 | 1000001000000000100 |
| 31 | 0000000100001 | 0000000000100001000 |
| 32 | 0000001000010 | 0000000001000010000 |
| 33 | 0000010000100 | 0000000010000100000 |
| 34 | 0000100001000 | 0000000100001000000 |
| 35 | 0001000010000 | 0000001000010000000 |
| 36 | 0010000100000 | 0000010000100000000 |
| 37 | 0100001000000 | 0000100001000000000 |
| 38 | 1000010000000 | 0001000010000000000 |
| 39 | 0000000000000 | 0010000100000000001 |
| 40 | 0000000000000 | 0100001000000000010 |
| 41 | 0000000000000 | 1000010000000000100 |
| 42 | 0000000000001 | 0000100000100001000 |
| 43 | 0000000000010 | 0001000001000010000 |
| 44 | 0000000000100 | 0010000010000100000 |
| 45 | 0000000001000 | 0100000100001000000 |
| 46 | 0000000010000 | 1000001000010000000 |
| 47 | 0000000100001 | 0000000000000000000 |
| 48 | 0000001000010 | 0000000000000000000 |
| 49 | 0000010000100 | 0000000000000000000 |
| 50 | 0000100001000 | 0000000000000000000 |
| 51 | 0001000010000 | 0000000000000000000 |
| 52 | 0010000100000 | 0000000000000000000 |
| 53 | 0100001000000 | 0000000000000000000 |
| 54 | 1000010000000 | 0000000000000000000 |
| 55 | 0000000000000 | 0000000000000000001 |
| 56 | 0000000000000 | 0000000000000000010 |
| 57 | 0000000000000 | 0000000000000000100 |
| 58 | 0000000000000 | 0000000000000001000 |
| 59 | 0000000000000 | 0000000000000010000 |
| 60 | 0000000000000 | 0000000000000100000 |
| 61 | 0000000000000 | 0000000000001000000 |
| 62 | 0000000000000 | 0000000000010000000 |
| 63 | 0000000000000 | 0000000000100000000 |
| 64 | 0000000000000 | 0000000001000000000 |
| 65 | 0000000000000 | 0000000010000000000 |
| 66 | 0000000000000 | 0000000100000000000 |
| 67 | 0000000000000 | 0000001000000000000 |
| 68 | 0000000000000 | 0000010000000000000 |
| 69 | 0000000000000 | 0000100000000000000 |
| 70 | 0000000000000 | 0001000000000000000 |
| 71 | 0000000000000 | 0010000000000000000 |
| 72 | 0000000000000 | 0100000000000000000 |
| 73 | 0000000000000 | 1000000000000000000 |
| 74 | 0000000000001 | 0000000000100000000 |
| 75 | 0000000000010 | 0000000001000000000 |
| 76 | 0000000000100 | 0000000010000000000 |
| 77 | 0000000001000 | 0000000100000000000 |
| 78 | 0000000010000 | 0000001000000000000 |
| 79 | 0000000100000 | 0000010000000000000 |
| 80 | 0000001000000 | 0000100000000000000 |
| 81 | 0000010000000 | 0001000000000000000 |
| 82 | 0000100000000 | 0010000000000000000 |
| 83 | 0001000000000 | 0100000000000000000 |
| 84 | 0010000000000 | 1000000000000000000 |

**Table 9.** The 43-round linear characteristic for KATAN48

| Round | input masks of register $L_1$ | input masks of register $L_2$ |
|---|---|---|
| 0 | 0000000001000001001 | 00000100000000100000000000000 |
| 1 | 0000000100000100100 | 00010000000010000000000000000 |
| 2 | 0000010000010010000 | 01000000001000000000000000000 |
| 3 | 0001000001001000001 | 00000000000000000000000000000 |
| 4 | 0100000100100000100 | 00000000000000000000000000000 |
| 5 | 0000000010010010000 | 00000000000000000000000000001 |
| 6 | 0000001001001000000 | 00000000000000000000000000100 |
| 7 | 0000100100100000000 | 00000000000000000000000010000 |
| 8 | 0010010010000000000 | 00000000000000000000001000000 |
| 9 | 1001001000000000000 | 00000000000000000000100000000 |
| 10 | 0000000000000000000 | 00000000000000000010000000010 |
| 11 | 0000000000000000000 | 00000000000000001000000001000 |
| 12 | 0000000000000000000 | 00000000000000100000000100000 |
| 13 | 0000000000000000000 | 00000000000010000000010000000 |
| 14 | 0000000000000000000 | 00000000001000000001000000000 |
| 15 | 0000000000000000000 | 00000000100000000100000000000 |
| 16 | 0000000000000000000 | 00000010000000010000000000000 |
| 17 | 0000000000000000000 | 00001000000001000000000000000 |
| 18 | 0000000000000000000 | 00100000000100000000000000000 |
| 19 | 0000000000000000000 | 10000000010000000000000000000 |
| 20 | 0000000000000000010 | 00000000000000000000000000000 |
| 21 | 0000000000000001000 | 00000000000000000000000000000 |
| 22 | 0000000000000100000 | 00000000000000000000000000000 |
| 23 | 0000000000010000000 | 00000000000000000000000000000 |
| 24 | 0000000001000000000 | 00000000000000000000000000000 |
| 25 | 0000000100000000000 | 00000000000000000000000000000 |
| 26 | 0000010000000000000 | 00000000000000000000000000000 |
| 27 | 0001000000000000000 | 00000000000000000000000000000 |
| 28 | 0100000000000000000 | 00000000000000000000000000000 |
| 29 | 0000010000010000000 | 00000000000000000000000000001 |
| 30 | 0001000001000000000 | 00000000000000000000000000100 |
| 31 | 0100000100000000000 | 00000000000000000000000010000 |
| 32 | 0000000000000000000 | 00000000000000000000001000001 |
| 33 | 0000000000000000000 | 00000000000000000000100000100 |
| 34 | 0000000000000000000 | 00000000000000000010000010000 |
| 35 | 0000000000000000000 | 00000000000000001000001000000 |
| 36 | 0000000000000000000 | 00000000000000100000100000000 |
| 37 | 0000000000000000000 | 00000000000010000010000000000 |
| 38 | 0000000000000000000 | 00000000001000001000000000000 |
| 39 | 0000000000000000000 | 00000000100000100000000000000 |
| 40 | 0000000000000000000 | 00000010000010000000000000000 |
| 41 | 0000000000000000000 | 00001000001000000000000000000 |
| 42 | 0000000000000000000 | 00100000100000000000000000000 |
| 43 | 0000000000000000000 | 10000010000000000000000000000 |