Fault Analysis of the KTANTAN Family of Block Ciphers: A Revisited Work of Fault Analysis of the KATAN Family of Block Ciphers

Alya Geogiana Buja[1, 2], Shekh Faisal Abdul-Latip[1] and Rabiah Ahmad[1]

[1]*INSFORNET, Faculty of ICT, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, Durian Tunggal, 76100 Melaka.*

[2]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Melaka Branch (Jasin Campus), 77300 Merlimau, Melaka.*

geogiana@melaka.uitm.edu.my

***Abstract*—This paper investigates the security of the KTANTAN block cipher against differential fault analysis. This attack is considered to be first side channel analysis of KTANTAN in the literature. KTANTAN is a relative to the KATAN block cipher. Therefore, the previous fault analysis on KATAN family of block cipher is revisited. Similar to KATAN, KTANTAN has three variants namely KTANTAN32, KTANTAN48 and KTANTAN64. The inner structure of KTANTAN is similar to KATAN except the key schedule algorithms. KATAN has been practically broken by using fault analysis, employing a transient single-bit fault model, with the assumption is that the attacker is able to inject faults randomly into the internal state of the cipher. The attack is empowerd by extended cube method similarly as applied on KATAN. The complexity of this attack is 2^{74} for KTANTAN32 and 2^{76} for both KTANTAN48 and KTANTAN64. Furthermore, based on the obtained results, this paper concludes that KTANTAN is more robust against fault analysis compared to KATAN.**

***Index Terms*—Cryptanalysis; KATAN/KTANTAN; Cube Attack; Fault Analysis.**

I. Introduction

KTANTAN is a lightweight block cipher designed for small devices and usually used in embedded system. KTANTAN is designed to be an efficient hardware-oriented block cipher as proposed in KATAN/KTANTAN family [1]. There are so many research works [2][3][4][5][6] on KATAN found in the literature compared to KTANTAN. In this paper we provide the first side-channel attack on KTANTAN. Previously, KTANTAN have been attacked in standard mathematical attack model by using meet-in-the-middle technique [7][8][9][10] and has been practically broken by using related key attack [11]. Meanwhile, Abdul-Latip et al. in [2] have analyzed KATAN block cipher by using side-channel fault analysis. For KATAN32, there were 21 subkey bits have been successfully recovered from four faulty rounds by using on average 115 fault injections. For KATAN48, also from four faulty rounds, 25 subkey bits were obtained by using on average 211 fault injections. For KATAN64, as well as KATAN48, 25 subkey bits have been recovered by using on average 278 fault injections at five efficient rounds. Later, in 2013, Song and Hu [12] provided new results on fault analysis of KATAN by using single-bit fault model on the three variants of KATAN by utilizing the earlier round of the cipher and recovered the whole 80-bit secret key with 132, 44 and 52 fault injections respectively. KTANTAN has similar structure (except the key schedule) with KATAN, therefore KTANTAN may also be vulnerable to fault analysis. The same method (as implemented in [2]) with some modification is employed in this study.

Our work in this paper is motivated by the investigation of KATAN block cipher against fault analysis. Therefore, in this paper, the sibling of KATAN named KTANTAN is studied. By using the same method (with some modification) as applied in KATAN, our study shows that KTANTAN with fixed key schedule is more robust against differential fault analysis compared to KATAN.

This paper is organized such that in Section II, a brief description on innovated differential fault attack is provided. The description of KTANTAN is explained in Section III. In Section IV, methods of fault analysis on the KTANTAN family of block ciphers are presented. Section V discusses the results of the attack and finally, the conclusion is presented in Section VI.

II. Differential Fault Analysis

Differential fault analysis (DFA) was firstly introduced by Biham and Shamir in [13]. The idea of DFA is to analyze the cipher by compromising the implementation of the cipher. DFA is a type of side channel attack. The internal state of the cipher is injected with a fault to make corruption in the internal state. By doing this, at the end of the encryption process, some information regarding the internal state can be obtained which lead to the recovery of the secret key. Abdul-Latip et al. have enriched the DFA method (single-bit fault model) with cube [14] and extended-cube methods [15] as described in [2]. The fault is transient rather than permanent. In this model, it is assumed that the attacker can cause one bit error into the internal state of a cipher during its execution without disturbing the bit position permanently. Furthermore, the attacker can choose the target round to do the fault injection. In this innovated method, the attacker can determine the position of the faulty bit in the internal state by using differential characteristics. This task is done by using cube of size 1 applying cube attack. Cube of size 1 in cube attack is equal to the standard differential in which we flip the value of a single bit of the internal state from 0 to 1 or vice versa through fault injection. By implementing cube attack, low degree polynomial (linear and quadratic) equations can be obtained. From the equations, to recover the secret key, only independent equations are chosen to be solved using, for example, Gaussian Elimination.

Table 1
Position of Chosen Bits to Enter f_a and f_b

		KTANTAN32	KTANTAN48	KTANTAN64
Chosen bit to enter f_a	x1	12	18	24
	x2	7	12	15
	x3	8	15	20
	x4	5	7	11
	x5	3	6	9
Chosen bit to enter f_b	y1	18	28	38
	y2	7	19	25
	y3	12	21	33
	y4	10	13	21
	y5	8	15	14
	y6	3	6	9

III. A Brief Description of KTANTAN

KTANTAN is a family of block ciphers that was designed to meet the requirements of small devices with limited resources [1]. There are three variants of KTANTAN named according to the block size; KTANTAN32, KTANTAN48 and KTANTAN64 with block size 32, 48 and 64 bits respectively. All variants accept 80-bit keys as the input. The differences between all variants are, first and definitely, the size of plaintext, P and the length of register L1 and L2. Besides, all the three variants can also be differentiated in terms of the position of bits that are chosen to enter nonlinear functions (as in Table 1) and the number of nonlinear functions used in each round. For KTANTAN32, the nonlinear functions, f_a and f_b, only used once while for KTATAN48, in one round of the cipher the functions f_a and f_b are applied twice. First, a pair of nonlinear functions is executed, the registers are then updated, and then the two nonlinear functions are applied again by using the same round-key bits. In KTATAN64, each round applies f_a and f_b three times. As presented in Algorithm 1, for encryption, the KTANTAN block cipher retrieves plaintext, round-key and irregular update sequence (IR) as the inputs and produces ciphertext, C as the output after 254 rounds, i.e the complete cycle of the encryption. The input plaintext is loaded into two registers, L1 and L2. Then, two nonlinear functions, f_a and f_b take place by choosing certain bit from the plaintext sequence to enter f_a and f_b. The two nonlinear functions, f_a and f_b, are shown in Equations (1) and (2).

$$f_a(L_1) = L_1[x1] + L_1[x2] + (L_1[x3] \cdot L_1[x4]) + (L_1[x5] \cdot IR) + k_a \tag{1}$$

$$f_b(L_2) = L_2[y1] + L_2[y2] + (L_2[y3] \cdot L_2[y4]) + (L_2[y5] \cdot L_2[y6]) + k_b \tag{2}$$

The position of chosen bit to enter the nonlinear functions is as presented in Table 1. As shown in Equation 1, in each round, *IR* is applied into f_a. There are 508 subkey bits used in 254 round of KTANTAN. Each round requires two subkey bits; k_a and k_b. These subkey bits are then used in f_a and f_b respectively. After completing f_a and f_b, the registers L_1 and L_2 are updated, where the *most significant bit (MSB)* falls into nonlinear function f_b and the *least significant bit (LSB)* is loaded with the output of two nonlinear functions (*LS*B of L_1 is the output of f_b and otherwise). At the end, ciphertext, C is generated. KTANTAN is the sibling of KATAN. The structure is same for both KATAN and KTANTAN (refer Figure 1 in Appendix). The only different is the key schedule. The key schedule for KTANTAN is fixed while for KATAN, the key schedule is repeatedly clocked as the *linear-feedback shift register (LFSR)* which is clocked twice after two subkey; k_a and k_b which are 2_i and 2_{i+1} are extracted.

Algorithm 1 KTANTAN Encryption

INPUT:Round-key(ka and kb), irregular update, IR and plaintext, P
OUTPUT: Ciphertext, C
1: Load plaintext, P into L1 and L2
2: For r = 0 to 253 do
3: Get ka and kb
4: Apply fa and fb
5: Update L1 and L2
6: Update round counting LFSR, T
7: End For
8: Generate ciphertext, C

The 80-bit secret key in KTANTAN are treated in the form of five words with 16 bits each. From each word, by using a MUX16to1, the same bits of *MSB* are chosen. Then, out of the five bits, only one bit is chosen. Let 80-bit key is denoted as $K = w_4||w_3||w_2||w_1||w_0$, whereby the least significant bit of w_0 is the least significant bit of *K*, and the most significant bit of w_4 is the most significant bit of *K*. Then, let denote *T* as the round-counting *LFSR* (T_7 is the most significant bit), then, let a_i = MUX16to1(w_i, $T_7T_6T_5T_4$), where MUX16to1(x, y) gives the y_{th} bit of x. The k_a and k_b of KTANTAN are as in Equations (3) and (4).

$$k_a = \sim T_3 \cdot \sim T_2 \cdot (a_0) + (T_3 \ OR \ T_2) \cdot MUX4to1(a_4a_3a_2a_1, T_1T_0 \tag{3}$$

$$k_b = \sim T_3 \cdot T_2 \cdot (a_4) + (T_3 \ OR \sim T_2) \cdot MUX4to1(a_3a_2a_1a_0, \sim T_1 \sim T_0) \tag{4}$$

As stated in [1] only one bit is used twice, 15 bits are used four times, and the remaining 64 bits are used 3 times. Most of the bits are used at least five times. Further detail of key bits used in 254 rounds of KTANTAN can be found in [1].

Table 2
Irregular Update Sequence (IR) of KTANTAN

Round	IR	Round	IR
0 – 9	1,1,1,1,1,1,1,0,0,0	130 – 139	1,0,1,0,0,1,1,1,0,0
10 – 19	1,1,0,1,0,1,0,1,0,1	140 – 149	1,1,0,1,1,0,0,0,1,0
20 – 29	1,1,1,0,1,1,0,0,1,1	150 – 159	1,1,1,0,1,1,0,1,1,1
30 – 39	0,0,1,0,1,0,0,1,0,0	160 – 169	1,0,0,1,0,1,1,0,1,1
40 – 49	0,1,0,0,0,1,1,0,0,0	170 – 179	0,1,0,1,1,1,0,0,1,0
50 – 59	1,1,1,1,0,0,0,0,1,0	180 – 189	0,1,0,0,1,1,0,1,0,0
60 – 69	0,0,0,1,0,1,0,0,0,0	190 – 199	0,1,1,1,0,0,0,1,0,0
70 – 79	0,1,1,1,1,1,0,0,1,1	200 – 209	1,1,1,1,0,1,0,0,0,0
80 – 89	1,1,1,1,0,1,0,1,0,0	210 – 219	1,1,1,0,1,0,1,1,0,0
90 – 99	0,1,0,1,0,1,0,0,1,1	220 – 229	0,0,0,1,0,1,1,0,0,1
100 – 109	0,0,0,0,1,1,0,0,1,1	230 – 239	0,0,0,0,0,0,1,1,0,1
110 – 119	1,0,1,1,1,1,1,0,1,1	240 – 249	1,1,0,0,0,0,0,0,0,1
120 – 129	1,0,1,0,0,1,0,1,0,1	250 - 253	0,0,1,0

The comparisons between previous results and our new results on KTANTAN are listed in Table 3. As presented, the previous works are based on attacks in standard and related key attack model. In our study we provide the first side-channel attack on KTANTAN.

IV. Differential Fault Analysis on KTANTAN

In this paper we apply the attack that was proposed and used by Abdul-Latip et al [2] on KTANTAN family of block ciphers. The original proposal [1] of the cipher and also the referred bit-sliced implementation [3] as used in KATAN were also applied in this work. The same fault model i.e. transient single-bit fault model is used. However we only consider our attack in abstract model. We assume that the attacker is able to inject a single bit fault into the internal state. The attacker is free to choose the target round to inject the fault. The fault is randomly injected into the internal state as the attacker is not able to hit the specific target of the internal state. By using this method, it allows us to extract enough number of independent linear and quadratic equations that are solvable. Then, to recover the 80-bit secret key of KTANTAN, the key schedule of KTANTAN is used.

The procedure of the attack is presented in Algorithm 2. As listed in Algorithm 2, there are six steps of differential fault attack in this study. The attack requires plaintext and ciphertext as the input. To obtain low degree polynomial equation, extended cube method is applied. As in Algorithm 3, 4 and 5, later all obtained linearly independent equations is solved by using Gaussian Elimination to recover the key bits.

Table 3
Some Results of Attack on KTANTAN

Variant	Time Complexity	Attack Model	Technique of Attack	Reference
KTANTAN32	2^{74}	Side channel	Differential fault attack	Section V in this paper
KTANTAN48	2^{76}			
KTANTAN64	2^{76}			
KTANTAN32	2^{52}	Related Key	Related key	[11]
KTANTAN48	2^{44}			
KTANTAN64	2^{42}			
KTANTAN32	$2^{75.170}$	Standard	Meet-in-the-middle attack (MITM) (3-subset MITM)	[8]
KTANTAN48	$2^{75.044}$			
KTANTAN64	$2^{75.584}$			
KTANTAN32	$2^{72.9}$		Meet-in-the-middle attack (MITM) (Improved MITM)	[9]
KTANTAN48	$2^{73.8}$			
KTANTAN64	$2^{74.4}$			
KTANTAN32	$2^{68:06}$		Meet-in-the-middle attack (MITM) (Guess then MITM)	[10]
KTANTAN48	$2^{70:92}$			
KTANTAN64	$2^{73:09}$			

The differential fault attack applied in this study uses extended cube method as described in [15] and [2]. The subkey can be recovered by solving simple independent linear and quadratic equations in GF(2) from master polynomials.

Algorithm 2 Differential Fault Attack on KTANTAN

INPUT: Plaintext, ciphertext
OUTPUT: Recovered subkey
1: Extended Cube (cube of size 1)
2: Solve all gathered linearly independent equations that contain subkey by using Gaussian Elimination
3: List all recovered subkey

The master polynomial is assumed as a black box and is shown as in Equation (5).

$$p(x_1, \ldots, x_n) = t_I \cdot pS(i) + q(x_1, \ldots, x_n) \qquad (5)$$

where:

$p(x_1, \ldots, x_n)$: Master polynomial
$x_i, \ldots, x_n$: Secret and public variables
t_I : Maxterm if superpoly in p is linear polynomial
$pS(i)$: Superpoly of t_I in p
$q(x_1, \ldots, x_n)$: Monomials that misses at least one variable from t_I

Therefore, same as using the linearization method, as we consider that the fault occurred in the earlier round and there might be a too complex polynomial exists, we used cube based method in this work. We assume that the master polynomial as the black box. Therefore, in Algorithm 3, we define all input in the register (L1, L2 and key register) as the new variables. Then, by using cube size 1, which equal to single-bit fault model, we compute the differential of the ciphertext by XORing faulty and non-faulty ciphertext bits. The generated linear and quadratic equations stored in text file.

Algorithm 3 Extracting Low Degree Polynomial Equations

INPUT: Plaintext and secret key
OUTPUT: Linear and quadratic equation
1: Define each bit in registers L1 and L2 and key bits as new variables
2: Apply cube based method
3: Collect all linear and quadratic equation

To determine the faulty-bit positions of the internal state, we use differential characteristics (refer Algorithm 4). The difference characteristic corresponding to any bit position of the internal state of a cipher is a string that was obtained by *XORing* the non-faulty *ciphertext* and the faulty *ciphertext*.

Algorithm 4 Fault Position Determination

INPUT: Non-faulty ciphertext and faulty ciphertext
OUTPUT: Difference characteristic
1: Construct a difference characteristic for each internal state bit by referring to the error propagation of faulty bit.
2: Represent difference values 0 and 1 respectively for the corresponding characteristic bits with probability 1, while the - sign represents unknown values (i.e. can be either 0 or 1). Constant 0 and constant 1 superpolys indicate values 0 and 1 in the difference characteristic bits respectively
3: Find this exact position of faulty bit

Next, as shown in Algorithm 5, is the algorithm to find efficient rounds, we determine the distribution of the linear and quadratic equations that can be obtained from non-faulty and faulty *ciphertext* differential when bits of the internal state; L_1L_2 is induced by a fault bit by bit (one bit at one time). From the distribution chart, KTANTAN same as KATAN yields high number of quadratic equations compared to linear equations.

Algorithm 5 Finding Effective Rounds for Fault Induction

INPUT: Faulty bit
OUTPUT: Distribution number of linear and quadratic equation
1: Apply cube and extended cube methods considering cubes of size 1
2: Determine the rounds which contain a high number of quadratic and linear equation
3: Analyze the distribution of the linear and quadratic

Agreed with [2], the fault attack on KTANTAN is well done if faults are induced into the internal state within these specific effective rounds (round that contain high number of linear and quadratic equations).

V. Findings and Discussions

After applying differential fault attack on KTANTAN, we managed to obtain similar distribution of linear and quadratic equation for all variant of KTANTAN same as KATAN [2]. In addition, the differential characteristics constructed in our work are also similar with KATAN in [2]. Besides, the polynomial obtained from each efficient round as in KATAN is also similar except for the key bits. The comparison of the appeared subkey bit in the polynomial equations is as in Table 4. The same findings on differential characteristics and distribution of linear and quadratic equations of KTANTAN and KATAN are because they are composed of the same inner structure.

Table 4
Comparison of Subkey Bit Indices Appeared in the Polynomial Equations of KTANTAN and KATAN

Variants	Faulty bit	*Ciphertext* Bit Differential	KTANTAN	KATAN
KTANTAN/ KATAN32	s1	c22	s23 + s28 + k15 + s21*s24 + 0	s23 + s28 + k492 + s21s24
KTANTAN/ KATAN32	s2	c23	s24 + s29 + k31 + s22*s25	s24 + s29 + k490 + s22s25
KTANTAN/ KATAN48	s5	c34	s38 + s44 + k31 + s33*s41	s38 + s44 + k494 + s33s41
KTANTAN/ KATAN48	s8	c37	s41 + s47 + k63 + s36*s44	s41 + s47 + k492 + s36s44
KTANTAN/ KATAN64	s41	c5	s19 + s32 + k63 + s3*s8 + s15*s27	s19 + s32 + k497 + s3s8 + s15s27
KTANTAN/ KATAN64	s42	c6	s20 + s33 + k63 + s4*s9 + s16*s28	s20 + s33 + k495 + s4s9 + s16s28

At round, r = 211 until r = 253, the lowest subkey bit index was k14 and the highest is k63, while for KATAN, the lowest index is k474 and the highest is k500. All subkey bits are found began to appear in quadratic equations. The result of differential fault attack on the three variants of KTANTAN is summarized in Table 5. As shown in Table 5, only six subkey bits have been found for KTANTAN32 which requires on average of 115 fault injections at r = 231, 237, 243 and 249. For KTANTAN48 and KTANTAN64, only four subkey bits have been recovered with 211 and 278 fault injections respectively. For KTANTAN48 the subkey bits appeared at round r = 234, 238, 242, 246 and 250 and for KTANTAN64, the subkey bits recovered at r = 236, 238, 242, 246 and 250. More information about the findings obtained for KTANTAN32 can be found in Table 6 - 9 for KTANTAN32 (as in Appendix).

Table 5
Results of Differential Fault Attack on KTANTAN

Variant	Complexity	Number of Fault Injection	Faulty Round, r	Recovered Key
KTANTAN32	2^{74}	115	231, 237, 243, 249	k14, k15, k31, k47, k60, k63
KTANTAN48	2^{76}	211	234, 238, 242, 246, 250	k15, k31, k47, k63
KTANTAN64	2^{76}	278	236, 238, 242, 246, 250	k15, k31, k47, k63

VI. Conclusion

This paper presents the first differential fault attack on KTANTAN family of block cipher. By applying the same method as applied in KATAN which is transient single-bit fault model, the results obtained conclude that KTANTAN is more robust compared to KATAN against differential fault attack with less key bits can be recovered. The results by using side channel differential fault attack yields the attack complexity 2^{74} on KTANTAN32 and 2^{76} (for KTANTAN48 and 64). Meanwhile, for KATAN32/48/64, the complexity of the attack is 2^{59} for KATAN32 and 2^{55} (for KATAN48 and 64). The "burnt" key in the key schedule of KTANTAN helps in reducing the number of key bits that can be recovered by using differential fault attack. Further research and investigation on KATAN and KTANTAN key schedules are strongly recommended.

Appendix

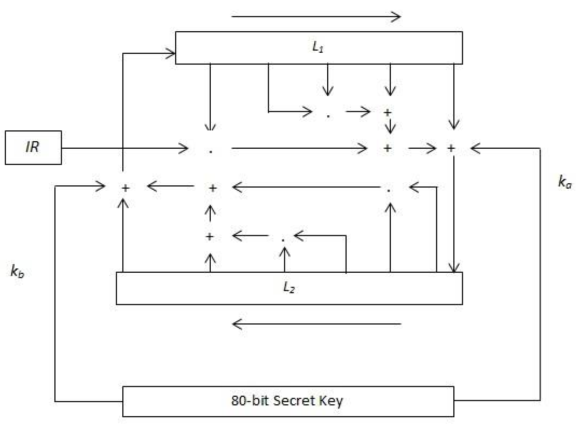

Figure 1: Structure of KATAN/KTANTAN

Tables 6 - 9 show the polynomial equations obtained at round, r = 231, 237, 243, 249 for KTANTAN32.

Table 6
Findings at r = 231

Faulty Bit in Internal State	Ciphertext Bit Differential	Polynomial Equation
s8	c7	s10
S9	c8	s11
S10	c9	s12
S11	c17	s9
S19	c28	s22
S20	c29	s23
S21	c30	s24
S22	c31	s25

Table 7
Findings at r = 237

Faulty Bit in Internal State	Ciphertext Bit Differential	Polynomial Equation
s1	c28	s19 + s23 + s28 + k15 + s21s24
	c24	s22 + s26 + s31 + k63 + s24s27
	c6	s6
	c4	s4 + s15 + k47 + s0s5 + s7s9
s2	c29	s20 + S24 + s29 + k31 + s22s25
	c27	s4
	c25	s0
	c5	s5 + s16 + k63 + s1s6 + s8s10
s3	c30	s25 + s30 + k31 + s23s26
	c28	s5
	c26	s1
	c12	s8
	c6	s6 + s17 + k63 + s2s7 + s9s11
s4	c27	s2
	c7	s7 + s18 + k31 + s3s8 + s10s12
s5	c30	s7
	c28	s3
	c21	s21 + s26 + k60 + s19s22
	c8	s19
s9	c2	s11
s10	c12	s12
s11	c7	s9
s12	c12	s10
s19	c22	s22
s20	c23	s23
s21	c24	s24
s22	c25	s25
s23	c26	s26
	c12	s20
s24	c27	s27
	c20	s7 + s18 + s22 + s27 + k14 + k31 + s3s8 + s10s12 + s20s23 + 1
	c13	s21

Table 8
Findings at r = 243

Faulty Bit in Internal State	Ciphertext Bit Differential	Polynomial Equation
s0	c21	s22 + s27 + k14 + s20s23
s1	c27	s6
	c22	s23 + s28 + k15 + s21s24
	c20	s3
s2	c28	s7
	c23	s24 + s29 + k31 + s22s25
	c21	s4
	c19	s0
s3	c24	s10
	c22	s5
	c20	s1
	c0	s6 + s17 + k63 + s2s7 + s9s11
s4	c25	s26 + s31 + k63 + s24s27
	c21	s2
	c1	s7 + s18 + k31 + s3s8 + s10s12
s18	c4	s21
	c1	s4 + s15 + k47 + s0s5 + s7s9
s19	c5	s22
	c2	s5 + s16 + k63 + s1s6 + s8s10
s21	c7	s24
s22	c8	s25
	c5	s19
s23	c9	s26
	c6	s20
s24	c10	s27
	c7	s21
s25	c20	s21 + s23 + s31 + k60 + k63 + s19s22 + s24s27 + 1
	c9	s23

Table 9
Findings at r = 249

Faulty Bit in Internal State	Ciphertext Bit Differential	Polynomial Equation
s4	c19	s22 + s26 + s31 + k63 + s24s27
s5	c20	s0
s21	c1	s24
s23	c3	s26
	c0	s20
s25	c2	s22

ACKNOWLEDGMENT

This work was supported by Universiti Teknologi MARA (UiTM) Malaysia under SLAB Scholarship and Fundamental Research Grant Scheme of UTeM FRGS/1/2015/ICT05/FTMK/02/F00293 funded by Ministry of Higher Education, Malaysia.

REFERENCES

[1] C. De Canni`ere, O. Dunkelman, M. Knezevic, *KATAN and KTANTAN – A family of small and efficient hardware-oriented block ciphers*. In: C. Clavier, K. Gaj (eds.) CHES. Lecture Notes in Computer Science, vol. 5747, pp. 272–288. Springer

[2] S.F. Abdul-Latip, M.R. Reyhanitabar, W. Susilo, J. Seberry, *Fault analysis of the KATAN family of block ciphers*. In: M.D. Ryan, B. Smyth, G. Wang (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 319–336. Springer, Heidelberg

[3] J-P. Aumasson, M. Knezevic, O. Dunkelman,O, *Bit-sliced reference code of KATAN and KTANTAN*. Available from http://www.cs.technion.ac.il/˜orrd/KATAN/katan.c

[4] T. Fuhr, B. Minaud, *Match box meet-in-the-middle attack against KATAN*, FSE 2014, LNCS, vol. 8540, pp. 61-81, Springer, 2015.

[5] T. Isobe, K. Shibutani, *Improved all-subkeys recovery attacks on FOX, KATAN and SHACAL-2 block ciphers*, FSE 2014, LNCS, vol. 8540, pp. 104-126, Springer, 2015.

[6] S. Rasoolzadeh, H. Raddum, Improved multi-dimensional meet-in-the-middle cryptanalysis of KATAN. IACR Cryptology ePrint Archive 2016

[7] A. Bogdanov, C. Rechberger. A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In Selected Areas in Cryptography, pages 229–240, 2010.

[8] A. Bogdanov, C. Rechberger, Generalized meet-in-the-middle attacks: cryptanalysis of the lightweight block cipher KTANTAN. Preproceedings of SAC 2010

[9] L. Wei, C. Rechberger, J. Guo, H. Wu, H., Wang, S. Ling, *Improved meet-in-the-middle cryptanalysis of KTANTAN*. Cryptology ePrint Archive, Report 2011/201 (2011) http://eprint.iacr.org/.

[10] B. Zhu, G. Gong, *Guess-then-meet-in-the-middle attacks on the KTANTAN family of block ciphers*. Cryptology ePrint Archive, Report 2011/619, pp. 1–14, 2011

[11] M. Agren, Some instant- and practical-time related-key attacks on KTANTAN32/48/64. http://eprint.iacr.org/2011/140
[12] L. Song, L. Hu, *Improved algebraic and differential fault attacks on the KATAN block cipher*, In R. H. Deng and T. Feng, Information Security Practice and Experience, ISPEC 2013, LNCS, vol. 7863, pp 372-386, Springer, 2013.
[13] E. Biham, A. Shamir, *Differential fault analysis of secret key cryptosystems.* In: B.S. Kaliski (Ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg
[14] I. Dinur, A. Shamir, *Cube Attacks on Tweakable Black Box Polynomials*. In: S.F. Abdul-Latip, M.R. Reyhanitabar, W. Susilo, J. Seberry, *Fault analysis of the KATAN family of block ciphers*. In: M.D. Ryan, B. Smyth, G. Wang (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 319–336. Springer, Heidelberg
[15] S.F. Abdul-Latip, M.R. Reyhanitabar, W. Susilo, J. Seberry, *Extended cubes: enhancing the cube attack by extracting low-degree non-linear equations*. In: B. Cheung et al. (Eds.) ASIACCS 2011. ACM, pp. 296–305.