MILITARY UNIVERSITY OF TECHNOLOGY

FACULTY OF CYBERNETICS

DOCTORAL THESIS

APPLICATION OF COMBINED CRYPTANALYSIS TECHNIQUES IN ATTACKS ON SELECTED BLOCK CIPHERS

Arkadiusz GĄSECKI, M. Sc. Eng.

SUPERVISOR

Professor Jerzy August GAWINECKI

Warsaw 2014

Table of Contents

1 Introduction

The most common methods to cryptanalyze block ciphers are differential and linear cryptanalysis. They use probabilistic schemes – their idea is based on prediction of how particular bits would be passed through the rounds of the algorithm. Having such knowledge, attacker tries to determine round key bits for certain rounds. These methods need plenty pairs of plaintext – ciphertext to be successfully applied. Currently block ciphers are designed in a way, which protects them against such cryptanalysis methods. This is done by increasing number of rounds and the proper construction of the algorithm.

Another cryptanalysis method, namely algebraic attack, was introduced in the last few years. This method basis on algebraic structure of the cipher. Contrary to the attacks mentioned before, this one requirements knowledge about just one or few pairs of plaintext – ciphertext. It is because of the fact, that this technique does not use any probabilistic properties. However, for such method it is hard to determine, whether it allows to perform attack successfully.

1.1 Thesis and range of the research

Solutions applied till now did not bring satisfactory results in a meaning of practical applications. The attacks known still need huge amounts of data, time needed to get solution, etc. However, in recent years new idea was introduced suggesting to join different cryptanalysis methods. Results published allow to determine thesis that **application of combined cryptanalysis techniques allows to raise effectiveness of attacks contrary to techniques used separately.**

That is why the main aim of this dissertation is either to improve or propose the methods allowing to raise effectiveness of the attacks on selected block ciphers by joining techniques mentioned above.

Dissertation contains methods of joining cryptanalysis methods, which are part of branches of computer science, like computer security or cryptology. Dependency between amount of data needed and time to find the solution was analysed. There were trials made for selected block ciphers, which compare particular methods and allow to verify additional factors like conversion methods. Until now, methods have needed either huge amount of entry data or problem was too complicated to be solved. This dissertation proves, that common application of elements coming from linear and differential cryptanalysis and algebraic attacks can benefit in more effective cryptanalysis, both in meaning of time complexity and data needed to perform the attack.

1.2 Main research problems and solution methodology

The aim of the research is comparison with methods used separately, but also verification of linear and differential characteristics used and methods of conversion of the equation sets to the form approvable by tools used to solve them.

Solutions proposed are:

- Construction of equation set based on algebraic description of the cipher
- Using more than one pair of plaintext – ciphertext, which satisfy particular linear or differential characteristics
- Adding extra equations given from differential or linear characteristics
- Conversion to Boolean satisfiability problem SAT
- Solving the set of equations with SAT-solver in regard to bits of the key

As a part of the implementation thesis given and solutions suggested, research problems introduced below were executed:

- Definition of the method allowing to join differential cryptanalysis with algebraic attack with its development
- Definition of the new method allowing to join linear cryptanalysis with algebraic attack
- Examining dependency between number of plaintexts and effectiveness of combined attacks

Details about their implementations are described in following chapters of this dissertation.

2 The DES Algorithm

This chapter describes the DES algorithm (*Data Encryption Standard*) with its substitute boxes and the method of round key generation.

2.1 Specification of the algorithm

The DES cipher, specification of which was described in (NIST, 1999) is a block cipher designed in 70s as a standard of data encryption. The length of the input block is 64 bits. Effective key length is 56 bits, additional 8 bits are used only to parity control and are not used in the algorithm of round key generation. Encipher and decipher computations consist of 16 rounds. Currently this cipher is no more considered secure, while it can be easily attacked by brute force method with CPU given with present-day technology. However it is still great example of research, because there is no low-data complexity attacks performed on this cipher. Moreover, there is 3-DES still in use, which is variation of original cipher made by multiplication of encipher and decipher computations with different keys.

The algorithm is in a form of the Feistel network. A sketch of the encipher computation is shown in Figure 2.1. Encipher computation is as follows:

- A block to be enciphered is subjected to the initial permutation IP
- Data is enciphered with 16-round complex key-dependent computation
- Output block is permuted again with permutation IP^{-1} which is inverse of the initial permutation

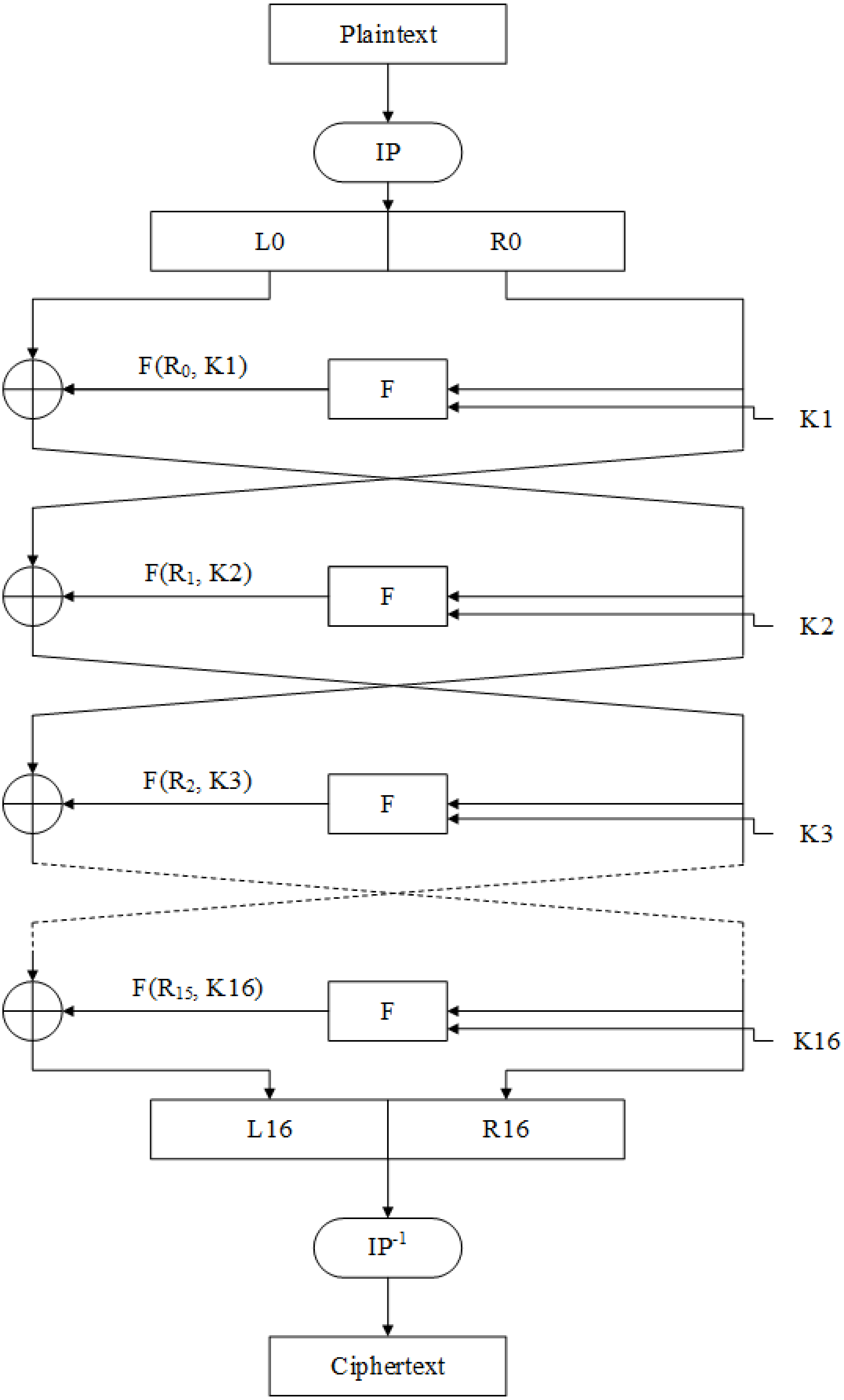


Figure 2.1 DES algorithm scheme

Such construction of the algorithm makes decipher computation similar to encipher computation. The only difference is that round keys are used in reverse order.

The cipher function, shown in Figure 2.2, takes 32 bits as an input.

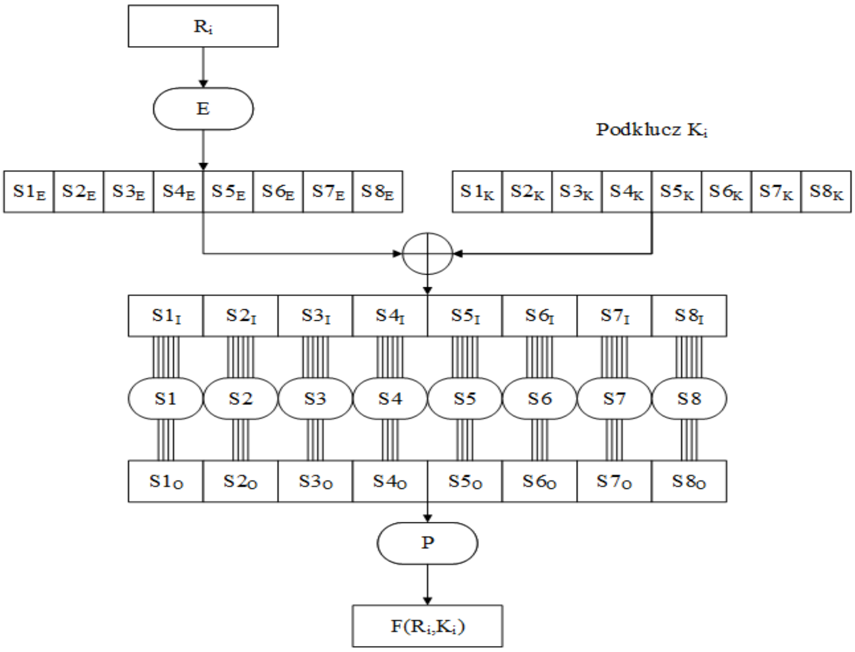

R_i
E
Podklucz K_i
$S1_E$ $S2_E$ $S3_E$ $S4_E$ $S5_E$ $S6_E$ $S7_E$ $S8_E$
$S1_K$ $S2_K$ $S3_K$ $S4_K$ $S5_K$ $S6_K$ $S7_K$ $S8_K$
$S1_I$ $S2_I$ $S3_I$ $S4_I$ $S5_I$ $S6_I$ $S7_I$ $S8_I$
S1 S2 S3 S4 S5 S6 S7 S8
$S1_O$ $S2_O$ $S3_O$ $S4_O$ $S5_O$ $S6_O$ $S7_O$ $S8_O$
P
$F(R_i,K_i)$

Figure 2.2 Cipher function of DES cipher

The function is as follows:

- Extension to 48 bits with permutation *E*
- XOR (exclusive or) operation with 48-bit round key
- Division to eight 6-bit inputs to substitutes boxes
- Output from boxes consists of 8x4 = 32 bits
- Output is subjected to permutation *P*

Then, according to the Feistel network specification, output from the cipher function is added modulo 2 with second part of the input block. After that both halves are switched. The switch operation is not executed after last round of the algorithm.

2.2 Description of substitutes boxes

The DES algorithm consists of eight substitutes boxes. Each of them takes a 6-bit block as an input and yields a 4-bit block as an output. The boxes, which are described in original specification as selection functions, are defined as a tables containing sixteen columns and four rows. The first and last bits of input represent in base 2 a number in the range 0 to 3, which is number of a row. Similarly, the middle 4 bits of input represent in base 2 a number in the range 0 to 15, which is a number of the column.

Sample substitute box is shown in Table 2.1.

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Table 2.1 First substitute box of DES cipher

Let assume, that the input block is 101000, then the output value is computed as follows:

- Row number: 2 (**1**0100**0** – bits 10 are equal 2 in hexadecimal)
- Column number: 4 (1**0100**0 – bits 0100 are equal 4 in hexadecimal)
- Returned value: 13, which in binary appears as four bits 1101

The important thing is, that rows and columns are numbered from 0 to 3 and from 0 to 15 respectively.

2.3 Key schedule

Key schedule is the algorithm to generate round keys, provided in few steps. Key schedule calculation scheme is presented in Figure 2.3.

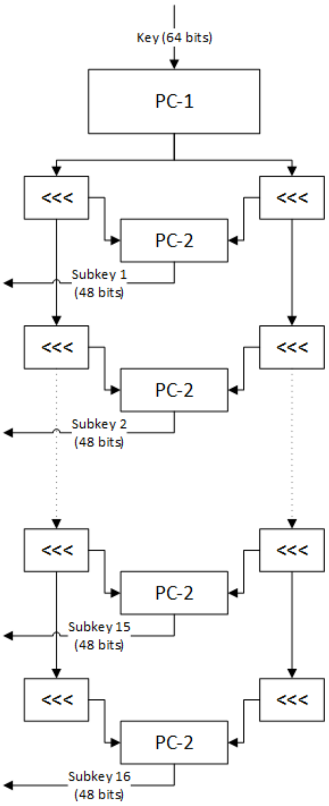


Figure 2.3 Key schedule calculation

At the beginning, 64 bits of the key are subjected to permutation *PC-1* (Table 2.2), which takes 56 bits (as mentioned before, other 8 bits could be used to parity control). Then, the result is split into two halves, 28 bits each, which are input to two separate computations, made in rounds.

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

Table 2.2 PC-1 permutation

These halves as shifted left with rotation in every round. The number of bits shifted equals 1 or 2 depending of round number (Table 2.3).

Number of rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table 2.3 Number of bits shifted with rotation

In every round, *PC-2* permutation (Table 2.4) is applied to choose 24 bits from each half. Round key, which consists of 48 bits, is given as a result.

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Table 2.4 PC-2 permutation

3 The SMS4 algorithm

This chapter describes algorithm SMS4 with its substitute boxes and round key generation.

3.1 Specification of the algorithm

The SMS4 cipher, which was first published in (SMS4, 2006) and its English translation was presented in (Diffie and Ledin, 2008), is a block cipher with 128-bit key and 128-bit input block. Encryption and decryption operations take 32 rounds of nonlinear substitutions. Both operations have the same structure, but the round key schedule for decryption is the reverse of the round key schedule for encryption.

SMS4 is an unbalanced Feistel network. One round of this cipher is shown in Figure 3.1.

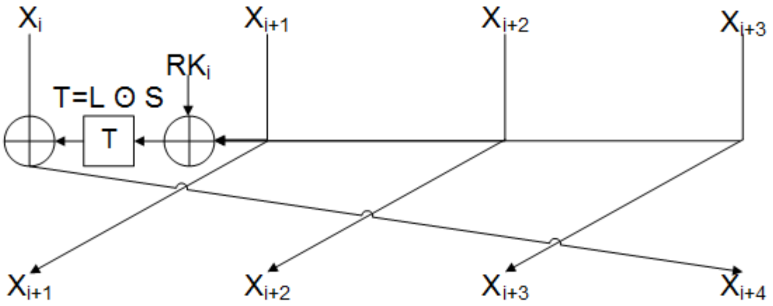


Figure 3.1 i-th round of the SMS4 algorithm

One round of encryption can be described using Equation 3.1, where T is a reversible substitution that generates 32 bits from 32 bits. It consists of two operations: non-linear substitution and linear diffusion which are further described in details.

$$X_{i+4} = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus RK_i)$$

Equation 3.1

The round function adds three last words to the round key over *GF(2)* field. The result becomes then an input to non-linear substitution *S* and linear diffusion *L*. The output is added to the first word over *GF(2)* field. Finally, words order is changed using operation $(X_i, X_{i+1}, X_{i+2}, X_{i+3}) \rightarrow (X_{i+1}, X_{i+2}, X_{i+3}, X_{i+4})$.

Description of non-linear substitution *S*

Non-linear substitution *S* is as follows:

1. 32-bit input block is divided into four 8-bit parts.
2. Every 8-bit part is an input to the *Sbox* transformation, which is performed with application of substitute box.
3. 8 bits stand as a result of every substitute box.

Let $U = (u_0, u_1, u_2, u_3) \in (Z_2^8)^4$ be the input to the S transformation and $V = (v_0, v_1, v_2, v_3) \in (Z_2^8)^4$ be its ouput. If application of substitute box is denoted as *Sbox()*, then the S transformation can be described with the Equation 3.2 given below.

$$V = (v_0, v_1, v_2, v_3) = S(U) = (Sbox(u_0), Sbox(u_1), Sbox(u_2), Sbox(u_3))$$

Equation 3.2

Linear diffusion operation *L*

Transformation *L* is a simple linear function, which takes as an input 32-bit output coming from *S* transformation. Let $V \in Z_2^{32}$ denote the input to this operation, $L(V) \in Z_2^{32}$ its output, and symbol $\lll i$ circular shift of an argument, with *i* bits shifted left. Then transformation L can be defined as it is given in Equation 3.3 below.

$$L(V) = V \oplus (V \lll 2) \oplus (V \lll 10) \oplus (V \lll 18) \oplus (V \lll 24)$$

Equation 3.3

Encryption process with application of *L* and *S* transformations is shown in Figure 3.2.

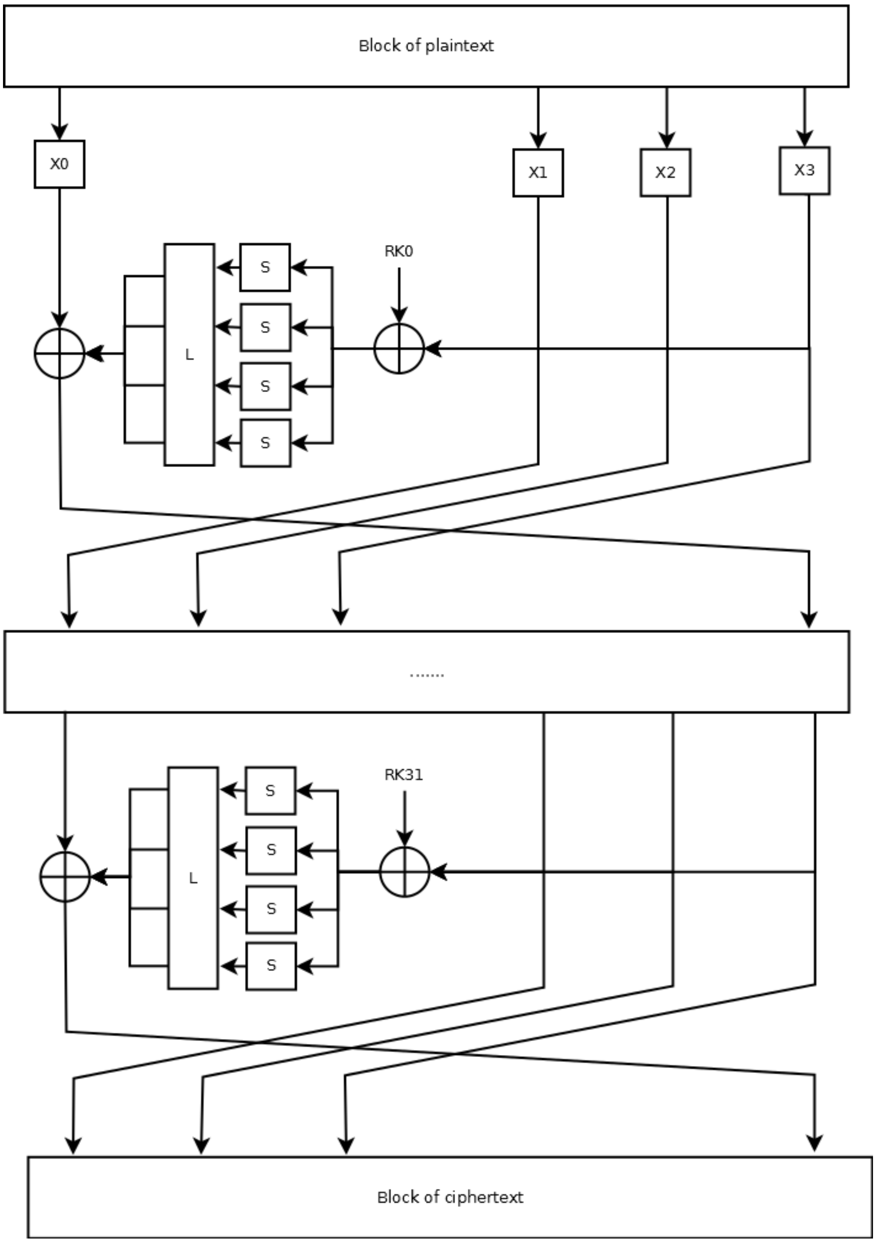


Figure 3.2 SMS4 scheme – encryption process

Every *S* transformation uses the same substitute box, which is described in details in next subchapter.

3.2 Description of the substitute box

Non-linear S transformation of the SMS4 algorithm uses the same substitute box four times. The input and the output of the S-box consist of eight bits. This transformation could be considered as a matrix with sixteen rows and columns.

Every element of this matrix contains predefined numeric value, written as hexadecimal number. The matrix, presenting S-box, is given in Table 3.1.

Row / Column number	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	d6	90	e9	fe	cc	e1	3d	b7	16	b6	14	c2	28	fb	2c	05
1	2b	67	9a	76	2a	be	04	c3	aa	44	13	26	49	86	06	99
2	9c	42	50	f4	91	ef	98	7a	33	54	0b	43	Ed	cf	ac	62
3	e4	b3	1c	a9	c9	08	e8	95	80	df	94	fa	75	8f	3f	a6
4	47	07	a7	fc	f3	73	17	ba	83	59	3c	19	e6	85	4f	a8
5	68	6b	81	b2	71	64	da	8b	f8	eb	0f	4b	70	56	9d	35
6	1e	24	0e	5e	63	58	d1	a2	25	22	7c	3b	01	21	78	87
7	d4	00	46	57	9f	d3	27	52	4c	36	02	e7	a0	c4	c8	9e
8	ea	bf	8a	d2	40	c7	38	b5	a3	f7	f2	ce	f9	61	15	a1
9	e0	ae	5d	a4	9b	34	1a	55	ad	93	32	30	f5	8c	b1	e3
A	1d	f6	e2	2e	82	66	ca	60	c0	29	23	ab	0d	53	4e	6f
B	d5	db	37	45	de	fd	8e	2f	03	ff	6a	72	6d	6c	5b	51
C	8d	1b	af	92	bb	dd	bc	7f	11	d9	5c	41	1f	10	5a	d8
D	0a	c1	31	88	a5	cd	7b	bd	2d	74	d0	12	b8	e5	b4	b0
E	89	69	97	4a	0c	96	77	7e	65	b9	f1	09	c5	6e	c6	84
F	18	f0	7d	ec	3a	dc	4d	20	79	ee	5f	3e	d7	cb	39	48

Table 3.1 Substitute box for the SMS4 algorithm

The choice of the value returned is made with usage of the eight input bits. First four bits determine number of the row, where last four bits determine number of the column (in both cases hexadecimal notation is used). For instance, if the input to the S-box consist of bits 11101111, then it could be described as *ef* in hexadecimal. This means, that value returned should come from *e*-th row and *f*-th column, which gives *Sbox(11101111)=Sbox(ef)=84.*

Substitute box could be also described as it was presented in (Liu et al., 2007) as an inversion over *GF(2^8)* field and affine transformation over *GF(2)* field. This could be written using Equation 3.4 below.

$$s(x) = I(x * A + C) * A + C,$$

Equation 3.4

Where

I() – inversion over *GF(2^8)* (with inversion of zero considered as zero)

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$C = (1,1,0,0,1,0,1,1)$$

Equation 3.5 presents irreducible polynomial used to convert from *GF(2)* to *GF(2^8)*:

$$f(x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$$

Equation 3.5

In (Erickson et al., 2009) Authors analysed and verified that way of S-box description and, based on it, they suggested alternate model, given in Equation 3.6.

$$s(x) = A_2 * I(A_1 * x + C_1) + C_2$$

Equation 3.6

The parameters for this equations are defined as follows:

$$A_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$C_1 = (1,1,0,0,1,0,1,1)^T$$

$$C_2 = (1,1,0,1,0,0,1,1)^T$$

Compared to the original model, this one uses right multiplication, *A₁* matrix was made by taking columns from *A* matrix in reverse order, and *A₂* matrix by taking rows from *A* matrix in reverse order. Vector *C₁* was made by transposition of *C* and vector *C₂* by taking elements of *C₁* in reverse order.

3.3 Key schedule

Round keys for the SMS4 algorithm are calculated based on 128-bit key *MK*. The algorithm to generate round keys is as follows:

1. Key *MK* is divided into four words, according to the Equation 3.7.

$$MK = (MK_0, MK_1, MK_2, MK_3)$$

Equation 3.7

2. Four words K_i are generated, by adding words received from previous step with constant values FK_i, given as hexadecimal numbers. Addition is made over *GF(2)* field.

$$(K_0, K_1, K_2, K_3) = (MK_0 \oplus FK_0, MK_1 \oplus FK_1, MK_2 \oplus FK_2, MK_3 \oplus FK_3)$$

Equation 3.8

$$FK_0 = (a3b1bac6), FK_1 = (56aa3350), FK_2 = (677d9197),$$
$$FK_3 = (b27022dc)$$

3. Round keys rk_i, where $i = 0,1, \dots ,31$, are calculated according to the Equation 3.9

$$rk_i = K_{i+4} = K_i \oplus L'\big(S(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)\big)$$

Equation 3.9

   a. Transformation *S* is the same as in the encipher algorithm.
   b. Linear diffusion *L'* is given in Equation 3.10.

$$L'(B) = B \oplus (B \lll 13) \oplus (B \lll 23)$$

Equation 3.10

Values CK_i, used to calculate rk_i are parameters, dependent on round number. These parameters are calculating according to the following algorithm. Let $ck_{i,j}$ denote j-th byte of CK_i ($i = 0,1,\dots,31; j = 0,1,2,3$). Then $ck_{i,j}$ are equal to: $ck_{i,j} = (4i + j) * 7\ (mod\ 256)$.

i	0	1	2	3
0	00070e15	1c232a31	383f464d	545b6269
4	70777e85	8c939aa1	a8afb6bd	c4cbd2d9
8	e0e7eef5	fc030a11	181f262d	343b4249
12	50575e65	6c737a81	888f969d	a4abb2b9
16	c0c7ced5	dce3eaf1	f8ff060d	141b2229
20	30373e45	4c535a61	686f767d	848b9299
24	a0a7aeb5	bcc3cad1	d8dfe6ed	f4fb0209
28	10171e25	2c333a41	484f565d	646b7279

Table 3.2 Values of CK$_i$ parameter used in algorithm for SMS4 key schedule

Such formula allows to calculate CK_i values, written in Table 3.2. Round number *i* in the table is calculated as sum of column and row indexes, given in first row and first column of the table.

4 Known techniques of cryptanalysis of selected block ciphers

This chapter describes these attacks on DES and SMS4, which were already performed and published and have either the best efficiency or complexity or were conducted for the highest number of rounds. The results presented were also partially used for purpose of research made within this dissertation.

4.1 Descriptions of the attacks presented

This subsection describes known cryptanalysis techniques, which were applied to perform combined attacks shown in this dissertation.

4.1.1 Linear cryptanalysis

Linear cryptanalysis was at first introduced by Matsui in (Matsui, 1993) and is one of the most popular methods for cryptanalysis, especially on block ciphers. This attack is performed in a known plaintext scenario. It uses a correlation between input and output bits of the cipher, satisfied with probability significantly different than ½. Based on it, linear approximation (also called as linear characteristic and denoted as $\Gamma_P \rightarrow \Gamma_C$) of the cipher is built, described with Equation 4.1, where *P,C,K* denote plaintext, ciphertext and key respectively. $\Gamma_P, \Gamma_C, \Gamma_K$ denote masks of plaintext *P*, ciphertext *C* and key *K*.

$$\Gamma_P \cdot P \oplus \Gamma_C \cdot C = \Gamma_K \cdot K$$

Equation 4.1

Such equation could be used in cryptanalysis, if its probability $p \neq \frac{1}{2}$. Effectiveness of the cryptanalysis is related to the value $\left|p - {}^{1}/_{2}\right|$. The higher absolute difference between probability and value ½, the more effective linear approximation based on this equation is.

4.1.2 Linear profile of substitute box

The aim of substitute boxes in the ciphers is to add nonlinearity. Linear profile of such box shows its weak points, namely parts of linearity. Linear profile for a box is shown as a table, having 2^n rows and 2^m columns, where n – number of input bits, m – number of output bits of the box. The algorithm to fill this table is as follows: every possible input to the substitute box is masked (where mask is equal to the index of the row and rows are numbered starting with zero) with bitwise operation **AND**. Similarly, output value corresponding to given input value is masked with the same operation and index of the column. Then, in both results, parity is checked, which means that if in both results number of ones is either even or odd, then final result is increased by one.

Sample linear profile is shown for simple 3-bit substitute box [7,6,0,4,2,5,1,3]. Profile is given in Table 4.1.

	Input mask							
Output mask	0	1	2	3	4	5	6	7
0	8	4	4	4	4	4	4	4
1	4	4	4	4	6	6	6	2
2	4	4	2	2	2	6	4	4
3	4	4	2	6	4	4	2	2
4	4	6	4	6	2	4	6	4
5	4	2	4	6	4	6	4	6
6	4	2	2	4	4	2	6	4
7	4	2	6	4	2	4	4	2

Table 4.1 Linear profile of substitute box given as a permutation [7,6,0,4,2,5,1,3]

Value marked with colour in table above was calculated as it is presented in Table 4.2.

Input of the box	Input mask	Value of input masked	Number of ones in result	Output of the box	Output mask	Value of output masked	Number of ones in result	Comparison of parity
0	4	0	0	7	3	3	2	1
1	4	0	0	6	3	2	1	0
2	4	0	0	0	3	0	0	1

3	4	0	0	4	3	0	0	1
4	4	4	1	2	3	2	1	1
5	4	4	1	5	3	1	1	1
6	4	4	1	1	3	1	1	1
7	4	4	1	3	3	3	2	0
SUM								**6**

Table 4.2 Calculation of linear profile for mask 4 → 3

The way of calculation, presented in Table 4.2 has to be performed for every possible pair of input and output mask.

4.1.3 Differential cryptanalysis

Differential cryptanalysis was formally introduced by Biham and Shamir in (Biham and Shamir, 1990). This is a chosen plaintext attack. Since its publication, this method was successfully applied on a wide range of block ciphers. Then, variants of this method were presented like truncated, higher order or impossible differentials or “square” and “boomerang” attacks. Generally, idea is based on analysis of pairs of plaintexts, which differ in some determined way and analysis of ciphertexts given from these plaintexts after encipher operation performed with the same key.

4.1.4 Differential profile of substitute box

The essence of differential cryptanalysis is the influence of substitute boxes to the difference of the ciphertexts. Other operations, like permutations, extensions and sums modulo 2 with key bits do not have any influence for this difference. Differential profile of substitute box determines number of pairs satisfying particular input and output difference. The way of construction of such profile is shown on an example using simple substitute box given as a nonlinear permutation [7,6,0,4,2,5,1,3], which has three input and three output bits. Thus, input and output differences can be equal to values in range from 0 to 7. Profile of substitute box is given in Table 4.3.

	Output difference							
Input difference	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	2	0	0	2
2	0	0	2	2	0	0	2	2
3	0	2	0	2	2	0	2	0
4	0	2	0	2	0	2	0	2
5	0	0	2	2	2	2	0	0
6	0	2	2	0	0	2	2	0
7	0	0	0	0	2	2	2	2

Table 4.3 Differential profile of substitute box given as a permutation [7,6,0,4,2,5,1,3]

In example above, input difference equal to 5 can result in output difference equal to 2, 3, 4 or 5. In the Table 4.3, difference 5 → 2 was marked. It satisfies for 2 out of 8 possible pairs, which means that its probability equals 2/8.

4.1.5 Algebraic attack

Algebraic attack, contrary to the techniques introduced before, does not base on any probabilistic properties. It is performed as an expression of the cipher as a set of equations of low degree and then solution of this set, where key bits are considered as unknown values. However, solving of such set is in general NP-hard problem, so successful application of this attack is extremely complicated.

4.2 Linear cryptanalysis of SMS4

There were plenty of attacks on SMS4 using linear cryptanalysis. In (Etrog and Robshaw, 2008) attack on 22 rounds was presented, where two-round linear approximation, shown in Figure 4.1, is considered.

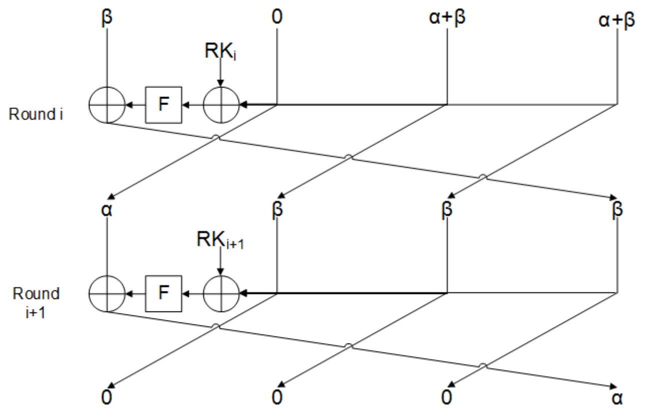


Figure 4.1 Two rounds of SMS4 with linear approximation using masks α and β

This approximation is given as $(\beta, 0, \alpha \oplus \beta, \alpha \oplus \beta) \rightarrow (\alpha, \beta, \beta, \beta) \rightarrow (0,0,0,\alpha)$, where α, β – parameters of approximation. If we assume that $\beta = \alpha$, then the approximation could be reduced to $(\alpha, 0,0,0) \rightarrow (\alpha, \alpha, \alpha, \alpha) \rightarrow (0,0,0,\alpha)$. Using such approximation and the structure of SMS4 cipher, five-round iterative linear approximation of a form $(0,0,0,\alpha) \rightarrow (0,0,\alpha,0) \rightarrow (0,\alpha,0,0) \rightarrow (\alpha,0,0,0) \rightarrow (\alpha,\alpha,\alpha,\alpha) \rightarrow (0,0,0,\alpha)$ could be built, of which only two last rounds are active. There are 24 values of parameter α found, which satisfy it with probability $2^{-10.2}$. Table 4.4 shows these values along with output values from substitute boxes, marked as S(α).

A	S(α)	A	S(α)
0x0011ffba	0x0084be2f	0x007852b3	0x00582b15
0x007905e1	0x005afbc6	0x00a1b433	0x00f1027a
0x00edca7c	0x0083ffaa	0x00fa7099	0x00d20b1d
0x05e10079	0xfbc6005a	0x11ffba00	0x84be2f00
0x3300a1b4	0x7a00f102	0x52b30078	0x2b150058
0x709900fa	0x0b1d00d2	0x7852b300	0x582b1500
0x7905e100	0x5afbc600	0x7c00edca	0xaa0083ff
0x9900fa70	0x1d00d20b	0xa1b43300	0xf1027a00
0xb3007852	0x1500582b	0xb43300a1	0x027a00f1
0xba0011ff	0x2f0084be	0xca7c00ed	0xffaa0083
0xe1007905	0xc6005afb	0xedca7c00	0x83ffaa00
0xfa709900	0xd20b1d00	0xffba0011	0xbe2f0084

Table 4.4 Values of α for five-round linear approximation

Using all elements described above, an approximation was built for 18 rounds of SMS4 cipher of a form $(0,0,0,\alpha) \xrightarrow{5\,rounds} (0,0,0,\alpha) \xrightarrow{5\,rounds} (0,0,0,\alpha)$

$\xrightarrow{5\,rounds} (0,0,0,\alpha) \rightarrow (0,0,\alpha,0) \rightarrow (0,\alpha,0,0) \rightarrow (\alpha,0,0,0)$, with probability $2^{-56.2}$. Then, attacks on cipher reduced to the number of rounds between 19 and 22 were performed. In the last case, data complexity was $2^{118.4}$ and memory complexity was 2^{112}. Summary of the attacks applied is in Table 4.5. Similarly, attack on 22 rounds of SMS4 was published in (Kim et al., 2008), where 18-round linear approximation, shown in Figure 4.2, was given.

Number of rounds	Data	Memory	Success rate(%)
19	$2^{116.4}$	2^{24}	99.5
20	$2^{117.4}$	2^{48}	99.9
21	$2^{118.4}$	2^{80}	99.9
22	$2^{118.4}$	2^{112}	99.9

Table 4.5 Summary of linear attacks on reduced SMS4 cipher

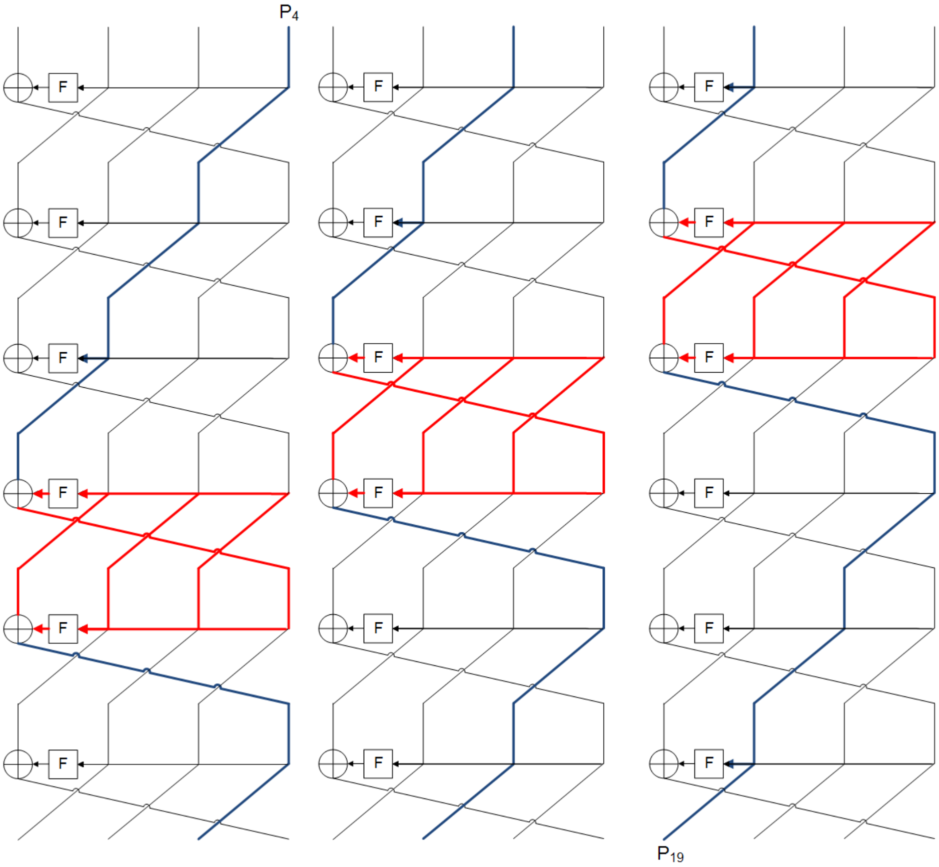


Figure 4.2 18-round linear approximation of SMS4 cipher

Based on characteristics presented, the attacks were performed on 22 rounds of SMS4, which is described in (Liu et al., 2009). Amount of data, needed to attacks shown there, was 2^{112} known plaintexts.

Complex multidimensional attack using linear cryptanalysis of 23 rounds of SMS4 was introduced in (Cho and Nyberg, 2011). To perform it, 20-round characteristic was built and it uses 18-round approximation described earlier. This characteristic starts in third round and finishes in twenty-second round. It is based on masks α, β, γ. At first, two-round characteristic for third and fourth round is constructed, and it could be given as Equation 4.2.

$$\alpha \cdot X_2 \oplus \beta \cdot (X_3 \oplus X_4 \oplus X_5 \oplus RK_2) = \alpha \cdot X_6$$

$$\gamma \cdot X_3 \oplus \alpha \cdot (X_4 \oplus X_5 \oplus X_6 \oplus RK_3) = \gamma \cdot X_7$$

Equation 4.2

Between 8th and 22nd round, 15-round characteristic given as Equation 4.3 was used.

$$\gamma \cdot X_7 \oplus \gamma \cdot X_{22} = \gamma \cdot (RK_7 \oplus RK_8 \oplus RK_{12} \oplus RK_{13} \oplus RK_{17} \oplus RK_{18})$$

Equation 4.3

Their joining results in 20-round characteristic, which could be described as an equation having left side of a form $(\alpha, \beta \oplus \gamma, \alpha \oplus \beta, \alpha \oplus \beta) \cdot P \oplus (\gamma, 0,0,0) \cdot C$, where $P = (X_2, X_3, X_4, X_5)$ and $C = (X_{22}, X_{23}, X_{24}, X_{25})$. Values of α, β, γ are taken from (Etrog and Robshaw, 2008) and given in Table 4.4.

4.3 Differential cryptanalysis of SMS4

There is a wide range of attacks on SMS4 published, using differential cryptanalysis techniques. In (Lu, 2007) rectangle attack on 14-round SMS4 and impossible differential attacks on 16-round SMS4 were presented. These attacks were further revisited and improved in (Toz and Dunkelman, 2008). In (Taehyun et al., 2008) authors introduced attack on cipher reduced to 22 rounds. It is based on 5-round iterative differential characteristic, shown in Figure 4.3, satisfied with probability 2^{-42}. It was constructed and published in (Zhang et al., 2008). There were 7905 (ca. 2^{13}) possible values of α found, which give the same input and output difference of round function with probability 2^{-21}. The first bytes of all possible α values are equal to zero and the rest bytes are all nonzero. Each nonzero byte of α holds a probability 2^{-7} through S-box.

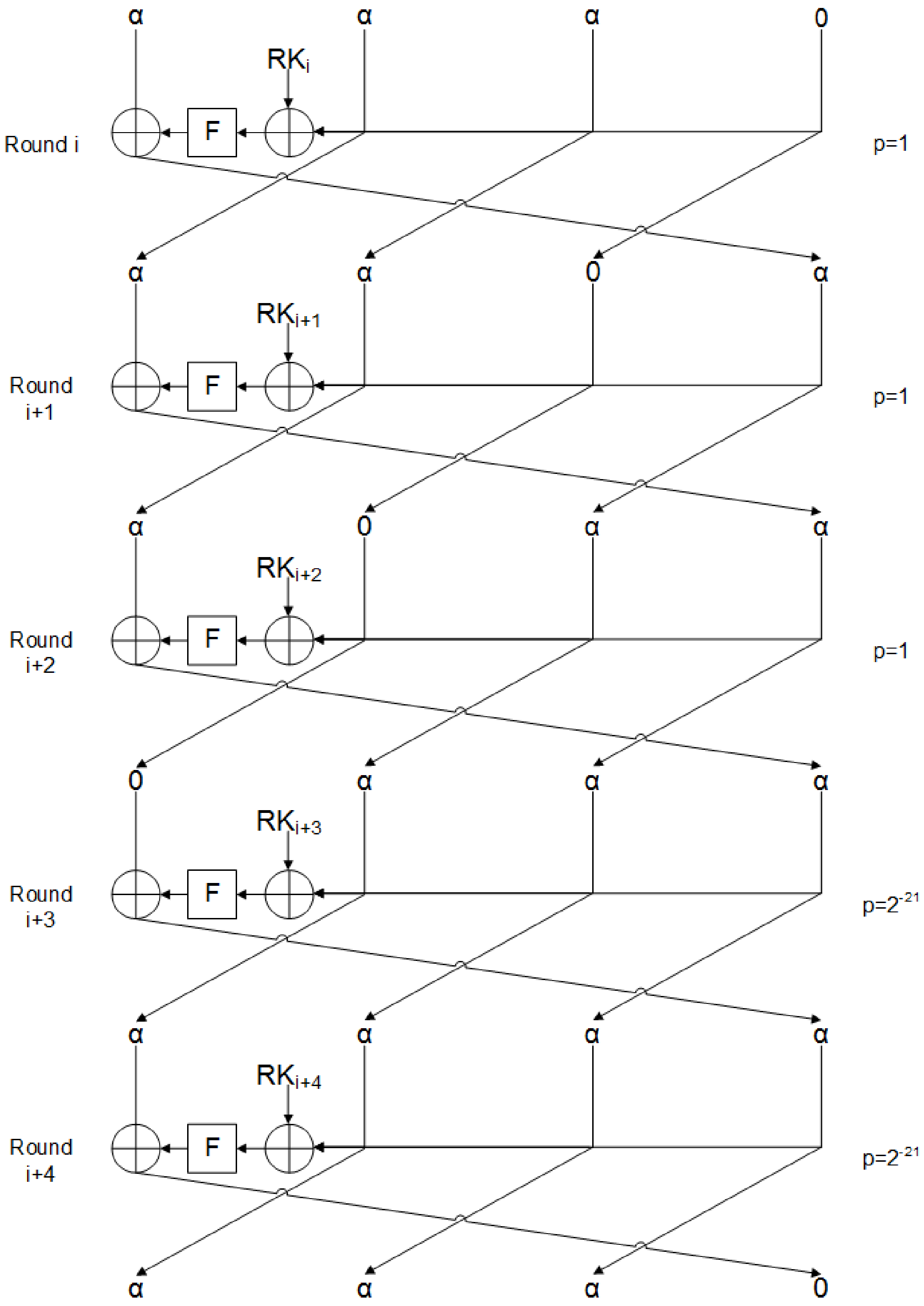


Figure 4.3 Five-round iterative differential characteristic for SMS4 cipher

Concatenation of this characteristic leads to 18-round differential characteristic with probability 2^{-126}. Its construction was described in (Zhang et al., 2008) and it is given in Table 4.6.

Round	Input	Cumulative probability
0	$(\alpha, \alpha, \alpha, 0)$	1
5	$(\alpha, \alpha, \alpha, 0)$	2^{-42}
10	$(\alpha, \alpha, \alpha, 0)$	2^{-84}
15	$(\alpha, \alpha, \alpha, 0)$	2^{-126}
16	$(\alpha, \alpha, 0, \alpha)$	2^{-126}
17	$(\alpha, 0, \alpha, \alpha)$	2^{-126}
18	$(0, \alpha, \alpha, \alpha)$	2^{-126}

Table 4.6 18-round differential characteristic

The attack on 22 rounds, described in (Taehyun et al., 2008), was performed using 4R type attack using 18-round differential characteristic mentioned above. The data complexity for this attack is 2^{118} chosen plaintexts and the memory complexity is 2^{123} bytes.

The best published attack using differential cryptanalysis was described in (Su et al., 2010). Authors showed there attack on SMS4 reduced to 23 rounds. It is also 4R type attack using 19-round differential cryptanalysis. Family of characteristics of this type is given in Table 4.7.

Round	ΔX_i	ΔX_{i+1}	ΔX_{i+2}	ΔX_{i+3}	Probability
0	$(a_0,$	$a_1,$	$a_1,$	$a_2)$	–
1	$(a_1,$	$a_1,$	$a_2,$	$a_3)$	$2^{-14}/2^{-13}/2^{-12}$
2	$(a_1,$	$a_2,$	$a_3,$	$a_1)$	1
3	$(a_2,$	$a_3,$	$a_1,$	$a_1)$	1
4	$(a_3,$	$a_1,$	$a_1,$	$a_4)$	2^{-14}
5	$(a_1,$	$a_1,$	$a_4,$	$a_5)$	2^{-14}
6	$(a_1,$	$a_4,$	$a_5,$	$a_1)$	1
7	$(a_4,$	$a_5,$	$a_1,$	$a_5)$	1
8	$(a_5,$	$a_1,$	$a_1,$	$a_5)$	2^{-14}
9	$(a_1,$	$a_1,$	$a_5,$	$a_4)$	2^{-14}
10	$(a_1,$	$a_5,$	$a_4,$	$a_1)$	1
11	$(a_5,$	$a_4,$	$a_1,$	$a_1)$	1
12	$(a_4,$	$a_1,$	$a_1,$	$a_3)$	2^{-14}
13	$(a_1,$	$a_1,$	$a_3,$	$a_2)$	2^{-14}
14	$(a_1,$	$a_3,$	$a_2,$	$a_1)$	1
15	$(a_3,$	$a_2,$	$a_1,$	$a_1)$	1
16	$(a_2,$	$a_1,$	$a_1,$	$a_6)$	2^{-14}
17	$(a_1,$	$a_1,$	$a_6,$	$a_7)$	2^{-14}

18	$(a_1,$ $a_6,$ $a_7,$ $a_1)$	1
19	$(a_6,$ $a_7,$ $a_1,$ $a_1)$	1

Table 4.7 Family of 19-round differential characteristics

Input difference (ΔX_i , ΔX_{i+1} , ΔX_{i+2} , ΔX_{i+3}) for this characteristic is considered as (a_0, a_1, a_1, a_2) and output difference as (a_6, a_7, a_1, a_1). There were about 2^{14} particular input and output differences defined, for which characteristic holds a probability 2^{-124} to 2^{-126} depending on values of parameters a_i. Data complexity for this attack on 23 rounds of SMS4 is 2^{115} chosen plaintexts. Similar attack was also published in (Z'aba et al., 2010), however in that publication modified type of SMS4 was attacked, where function *L* was replaced with function *L'* taken from key scheduling algorithm.

4.4 Algebraic attack on SMS4

The SMS4 cipher was also attacked using algebraic techniques. Theoretical deliberations on this topic were given in (Ji et al., 2007) and (Ji et al., 2009). Algebraic structure of SMS4 was also considered in (Weinmann, 2009). In (Erickson et al., 2009) in turn, authors give results for practical attack on the cipher reduced to particular number of rounds. Table 4.8 contains results given with two different tools applied to obtain the solution – Magma and MiniSAT.

	Magma		MiniSAT	
Number of rounds	Time [s]	Memory [MB]	Time [s]	Memory [MB]
4	2.37	100.20	235.58	70.74
5	7.72	207.68	>6000	-

Table 4.8 Summary of algebraic attacks on SMS4

This results show, that having small amount of input data, namely one pair plaintext – ciphertext, it is possible to attack only 5 rounds of SMS4 cipher. It is much worse than in case of linear and differential cryptanalysis however it also means much better data complexity comparing to hypothetical differential attack on 5 rounds of this cipher. Fact worth to notice is that Authors did not succeed with attack on 5 rounds using MiniSAT.

4.5 Summary of the attacks performed on SMS4

The SMS4 cipher, since its publication, became popular target for cryptanalysts all over the world. There are some publications, like (Taehyun et al., 2008), (Su et al., 2010), (Liu et al., 2009) which provide summary of the attacks made to compare with methods published. Table 4.9 contains these collations completed with other published and analysed attacks on SMS4 reduced to particular number of rounds.

Number of rounds	Data (known plaintexts)	Memory (in bytes)	Time (encipher operations)	Type of the attack	Sort of the attack
13	2^{16}	2^{20}	2^{114}	Chosen plaintext	„Integral” attack
14	$2^{121.82}$	$2^{125.82}$	$2^{116.66}$	Chosen plaintext	„Rectangle” attack
14	$2^{107.89}$		$2^{87.69}$	Chosen plaintext	„Rectangle” attack
16	2^{125}	2^{125}	2^{116}	Chosen plaintext	„Rectangle” attack
16	2^{105}	2^{109}	2^{107}	Chosen plaintext	Impossible differentials
18	2^{124}	2^{128}	$2^{112.83}$	Chosen plaintext	„Rectangle” attack
18	2^{120}	2^{123}	$2^{116.83}$	Chosen plaintext (adaptive)	„Boomerang” attack
21	2^{118}	2^{123}	$2^{126.6}$	Chosen plaintext	Differential cryptanalysis
22	2^{117}	2^{109}	$2^{109.86}$	Known plaintext	Linear cryptanalysis
22	2^{118}	2^{123}	$2^{125.71}$	Chosen plaintext	Differential cryptanalysis
22	2^{117}	No data	$2^{112.3}$	Chosen plaintext	Differential cryptanalysis
22	2^{112}	2^{122}	$2^{124.21}$	Known plaintext	Multidimensional linear cryptanalysis
23	2^{115}	No data	$2^{124.3}$	Chosen plaintext	Differential cryptanalysis
23	$2^{126.6}$	$2^{120.7}$	$2^{127.4}$	Known plaintext	Multidimensional linear cryptanalysis

Table 4.9 Summary of the attacks on SMS4

The attacks presented are of two types. The first type is attack with chosen plaintexts and attacks of this type are diverse kinds of differential cryptanalysis. The second type are attacks with known plaintexts and these are attacks using lienear cryptanalysis. Moreover table contains complexity of the attacks regard to data, memory and time needed. Results presented allow to determine some conclusions, from which the first one is that SMS4 has truly become interesting target for cryptanalysts all over the world and they have tried different methods to perform successful attack. Furthermore, much more attacks are implemented using popular techniques like differential and linear cryptanalysis and their variants. Algebraic attacks, although performed as well, are marginal. Table 4.9 shows that the largest group of the attacks are these executed using differential cryptanalysis. Moreover, cipher reduced only to 23 rounds was successfully broken, which means that original version containing 32 rounds still can be considered secure. Analysing amount of data needed to break the cipher it is easy to notice that to attack at least 14 rounds, 2^{100} known plaintexts are needed which makes these attacks impractical. Table does not contain the results of the algebraic attacks introduced in previous subsection.

4.6 Differential cryptanalysis of DES

Attack on DES using differential cryptanalysis was for the first time published in (Biham and Shamir, 1990). It is worth to notice that this method was not known before and this was the first publication describing it. Authors introduced there plenty differential characteristics and attacks on different variants of cipher, e.g. reduced to particular number of rounds, generalizations to similar systems and different modifications of DES.

Attack on the cipher, reduced to four rounds, was performed using one-round characteristic with probability 1. It is presented in Figure 4.4. With this characteristic, it is possible to find 42 bits of the round key for the last round.

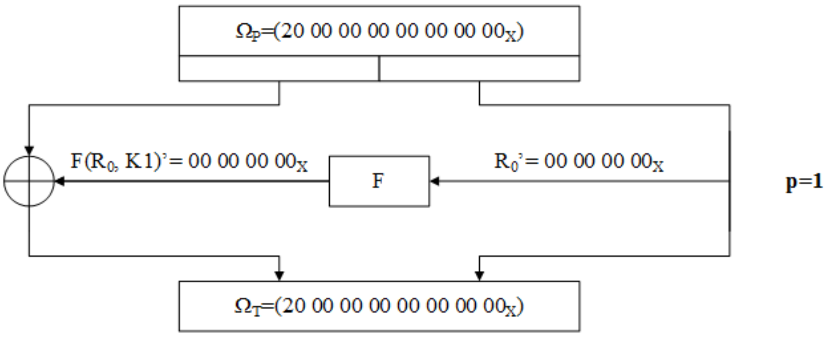

Ω_P=(20 00 00 00 00 00 00 00_X)
F(R_0, K1)'= 00 00 00 00_X
F
R_0'= 00 00 00 00_X
p=1
Ω_T=(20 00 00 00 00 00 00 00_X)

Figure 4.4 One-round characteristic with probability 1

Difference of the input data is marked as Ω_P and difference of the output data as Ω_P. R_i denotes right half of the difference in *i*-th round (where rounds are numbered starting with zero). *F* is round function.

Attack on six rounds of DES was performed using two differential characteristics with probability 1/16. They allow to implement attack of 3R type. The first characteristic, given in Figure 4.5, allows to find bits of round key for sixth round related to boxes S2, S5, S6, S7 and S8.

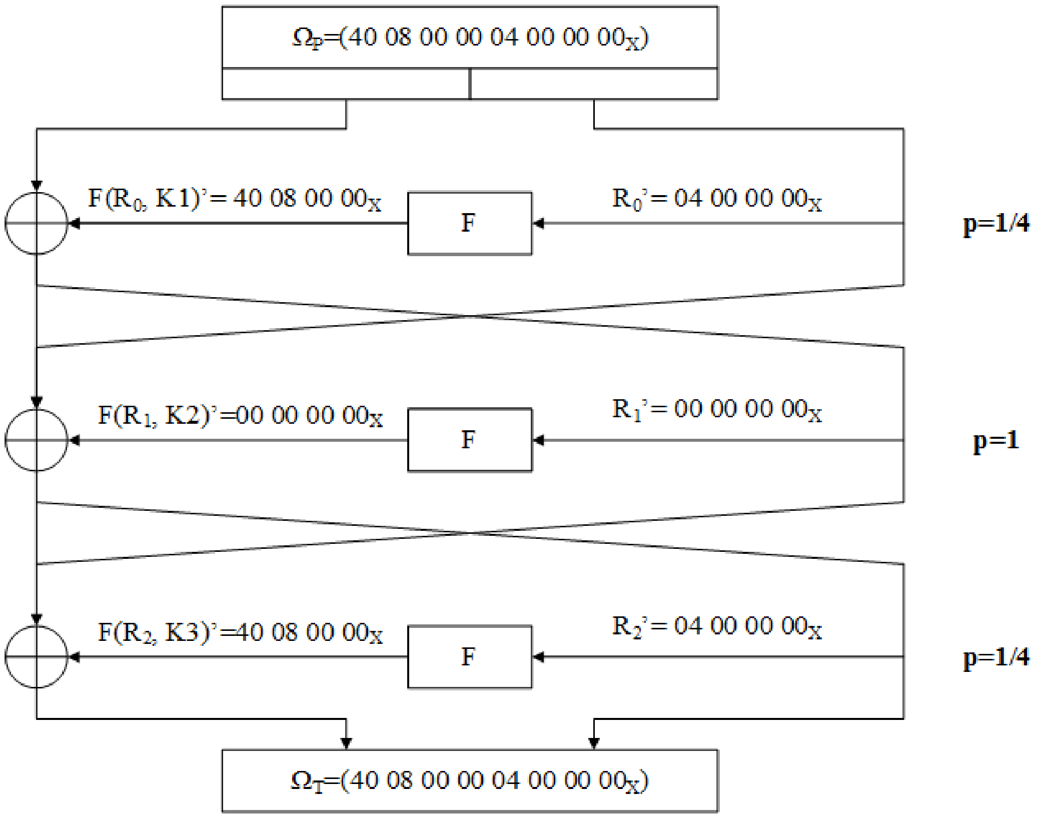
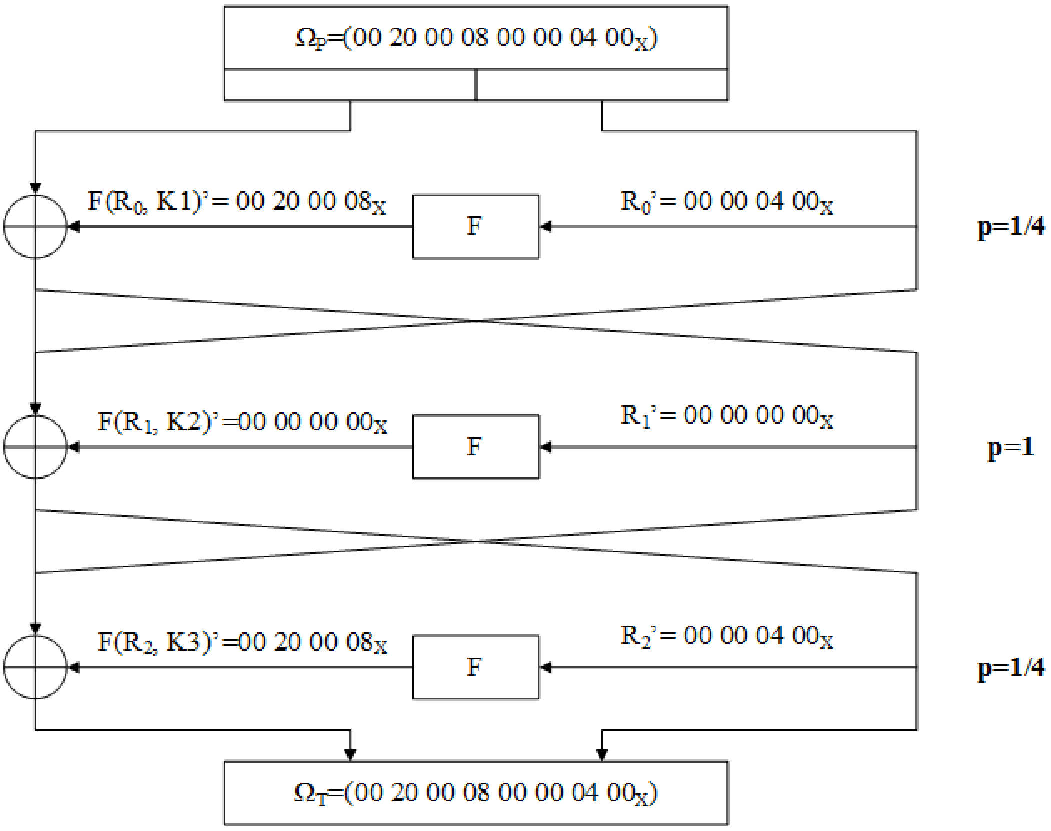

Ω_P=(40 08 00 00 04 00 00 00_X)
F(R_0, K1)'= 40 08 00 00_X
F
R_0'= 04 00 00 00_X
p=1/4
F(R_1, K2)'=00 00 00 00_X
F
R_1'= 00 00 00 00_X
p=1
F(R_2, K3)'=40 08 00 00_X
F
R_2'= 04 00 00 00_X
p=1/4
Ω_T=(40 08 00 00 04 00 00 00_X)

Figure 4.5 First three-round differential characteristic

Similarly, the second three-round characteristic, shown in Figure 4.6, could be used. It allows to find bits of last round key for boxes S1, S2, S4, S5 and S6. In cooperation, both characteristics allow to find 42 bits of the key. Last 14 bits could be then found using exhaustive search.



Ω_P=(00 20 00 08 00 00 04 00$_X$)
F(R$_0$, K1)'= 00 20 00 08$_X$
F
R$_0$'= 00 00 04 00$_X$
p=1/4
F(R$_1$, K2)'=00 00 00 00$_X$
F
R$_1$'= 00 00 00 00$_X$
p=1
F(R$_2$, K3)'=00 20 00 08$_X$
F
R$_2$'= 00 00 04 00$_X$
p=1/4
Ω_T=(00 20 00 08 00 00 04 00$_X$)

Figure 4.6 Second three-round differential characteristic

There are 120 pairs of plaintext – ciphertext needed, satisfying one of the characteristic mentioned above, to find bits of the key.

The DES cipher, reduced to eight rounds, could be broken using ca. 25000 pairs of ciphertexts, for which difference of input texts equals P'=40 5C 00 00 04 00 00 00$_X$. This method allows to find 30 bits of the round key for the last, eighth round. Probability of the characteristic used is 1/10485.76. As it was in previous case, this is attack of 3R type.

It is also possible to perform the attack for any number of rounds using two-round iterative differential characteristic, shown in Figure 4.7. Such characteristic could be concatenated with itself, however this operation lowers its probability.

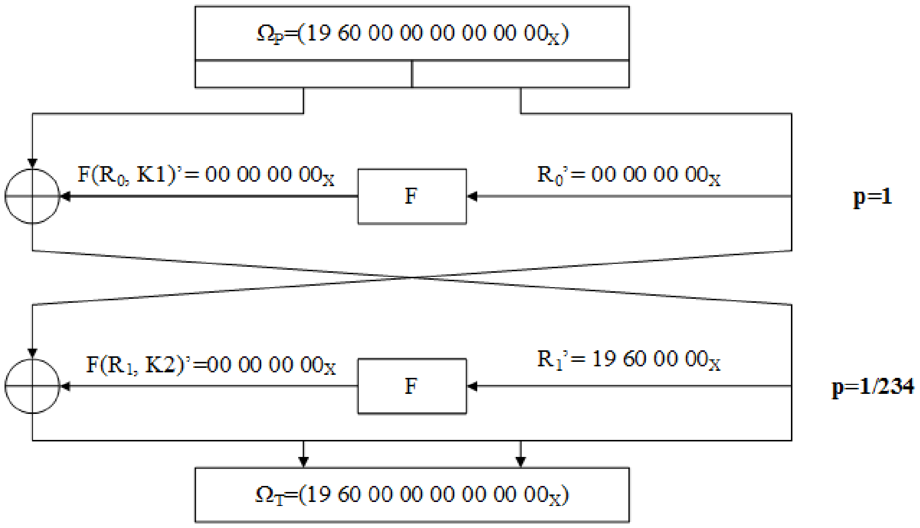

Ω_P=(19 60 00 00 00 00 00 00_X)
F(R_0, K1)'= 00 00 00 00_X
F
R_0'= 00 00 00 00_X
p=1
F(R_1, K2)'=00 00 00 00_X
F
R_1'= 19 60 00 00_X
p=1/234
Ω_T=(19 60 00 00 00 00 00 00_X)

Figure 4.7 Iterative characteristic of DES

Table 4.10 contains collation of probabilities of iterative characteristic depending on number of rounds, which are attacked.

Number of rounds	Probability
3	$1/234$
5	$1/55000$
7	$\approx 2^{-24}$
9	$\approx 2^{-32}$
11	$\approx 2^{-40}$
13	$\approx 2^{-48}$
15	$\approx 2^{-56}$

Table 4.10 Probability of iterative characteristic depending on number of rounds

Based on the table above it is easy to notice, that increasing number of rounds also increases data complexity of the attacks. Furthermore, for 15 rounds probability of the characteristic makes complexity worse than in attack using exhaustive search.

Collation of all the attacks, published in (Biham and Shamir, 1990), is given in Table 4.11.

Number of rounds	Number of pairs needed	Number of pairs used	Number of key bits found	Characteristic (rounds / probability)	Comment
4	2^3	2^3	42	1 $\quad$ 1	
6	2^7	2^7	30	3 $\quad$ $1/16$	
8	2^{15}	2^{13}	30	5 $\quad$ $1/10486$	
8	2^{17}	2^{13}	30	5 $\quad$ $1/10486$	
8	2^{20}	2^{19}	30	5 $\quad$ $1/55000$	Iterative characteristic

9	2^{25}	2^{24}	30	6 10^{-7}	
9	2^{26}	8	48	7 2^{-24}	
10	2^{34}	4	18	9 2^{-32}	
11	2^{35}	2^{11}	48	9 2^{-32}	
12	2^{42}	4	18	11 2^{-40}	
13	2^{43}	2^{19}	48	11 2^{-40}	
14	2^{50}	4	18	13 2^{-48}	
15	2^{51}	2^{27}	48	13 2^{-48}	Huge amount of memory needed
16	2^{57}	2^{5}	18	15 2^{-56}	Worse than exhaustive search

Table 4.11 Summary of differential cryptanalysis of DES cipher

Table 4.11 presents differential cryptanalysis of DES cipher, reduced to particular number of rounds. It gives complexity of input data as number of needed and used pairs. It also contains number of bits of the key, which could be found using particular characteristic. Column *Characteristic* contains number of rounds, which are described by characteristic, and its probability. The important conclusion, arisen from this summary is that those attacks would not be effective on full DES cipher. It is also easy to observe that number of pairs needed rises with number of rounds, which makes the performance of the attack tougher to execute. Contrast is also observable between amount of pairs needed and used in attacks on the cipher reduced to 9, 10, 12 and 14 rounds.

In (Biham and Shamir, 1991) in turn attack on full DES cipher, better than exhaustive search, was introduced. Summary of the attacks from this paper is given in Table 4.12.

Number of rounds	Chosen plaintexts	Analysed plaintexts	Complexity of analysis	Prior best	
				time	memory
8	2^{14}	4	2^{9}	2^{16}	2^{24}
9	2^{24}	2	2^{32}	2^{26}	2^{30}
10	2^{24}	2^{14}	2^{15}	2^{35}	-
11	2^{31}	2	2^{32}	2^{36}	-
12	2^{31}	2^{21}	2^{21}	2^{43}	-
13	2^{39}	2	2^{32}	2^{44}	2^{30}
14	2^{39}	2^{29}	2^{29}	2^{51}	-
15	2^{47}	2^{7}	2^{37}	2^{52}	2^{42}
16	2^{47}	2^{36}	2^{37}	2^{58}	-

Table 4.12 Summary of the attacks on DES using differential cryptanalysis

Table 4.12 contains information about data complexity, needed to perform the attacks on particular number of rounds and compares it with the best results from previous attacks. It is important that here for the first time possibility of the attack on full round DES, better than exhaustive search, was shown. Another interesting fact is, that for part of the attacks small amount of analysed texts is needed. This collation, contrary to the one given in Table 4.11 does not provide number of bits of the key found, which is important for the final effect of the cryptanalysis. Exhaustive search of part of the key bits is often left out in plenty publications despite the fact it has the influence on the quality and complexity of the attack.

4.7 Linear cryptanalysis of DES

Linear cryptanalysis, in context of DES cipher, was originally described in details in (Matsui, 1993). This attack is based on approximation of the cipher using linear Boolean function. In the DES case, dependencies between bits of the key, plaintext and ciphertext holding a probability significantly different than ½, were defined.

Linear cryptanalysis is an attack performed in known plaintext scenario. The attack on 16-round DES using this technique needs 2^{43} plaintexts. This attack was described in (Matsui, 1994).

Collation of the equations, allowing to approximate DES depending on number of rounds, is given in Table 4.13.

Number of rounds	Equation	Probability
3	$P_H[\alpha] \oplus P_L[15] \oplus C_H[\alpha] \oplus C_L[15] = K_1[22] \oplus K_3[22]$	$1/2 + 1.56 \times 2^{-3}$
4	$P_H[\alpha] \oplus P_L[15] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[22] \oplus K_3[22] \oplus K_4[\gamma]$	$1/2 - 1.95 \times 2^{-5}$
5	$P_H[15] \oplus P_L[\alpha,\beta] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[\gamma] \oplus K_2[22] \oplus K_4[22] \oplus K_5[\gamma]$	$1/2 + 1.22 \times 2^{-6}$
6	$P_L[\delta] \oplus C_H[\alpha] \oplus C_L[15] = L_2 \oplus K_6[22]$	$1/2 - 1.95 \times 2^{-9}$
7	$P_H[\delta] \oplus P_L[12,16] \oplus C_H[\alpha] \oplus C_L[15] = K_1[19,23] \oplus L_3 \oplus K_7[22]$	$1/2 + 1.95 \times 2^{-10}$

8	$P_H[\delta] \oplus P_L[12,16] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[19,23] \oplus L_3 \oplus K_7[22] \oplus K_8[\gamma]$	$1/2 - 1.22 \times 2^{-11}$
9	$P_H[15] \oplus P_L[\beta,\delta] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[\gamma] \oplus K_2[22] \oplus L_4 \oplus K_8[22] \oplus K_9[\gamma]$	$1/2 - 1.91 \times 2^{-14}$
10	$P_L[\alpha] \oplus C_H[\alpha] \oplus C_L[15] = L_2 \oplus L_6 \oplus K_{10}[22]$	$1/2 - 1.53 \times 2^{-15}$
11	$P_H[\alpha] \oplus P_L[15] \oplus C_H[\alpha] \oplus C_L[15] = K_1[22] \oplus L_3 \oplus L_7 \oplus K_{11}[22]$	$1/2 + 1.91 \times 2^{-16}$
12	$P_H[\alpha] \oplus P_L[15] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[22] \oplus L_3 \oplus L_7 \oplus K_{11}[22] \oplus K_{12}[\gamma]$	$1/2 - 1.19 \times 2^{-17}$
13	$P_H[15] \oplus P_L[\alpha,\beta] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[\gamma] \oplus K_2[22] \oplus L_4 \oplus L_8 \oplus K_{12}[22] \oplus K_{13}[\gamma]$	$1/2 + 1.49 \times 2^{-19}$
14	$P_L[\delta] \oplus C_H[\alpha] \oplus C_L[15] = L_2 \oplus L_6 \oplus L_{10} \oplus K_{14}[22]$	$1/2 - 1.19 \times 2^{-21}$
15	$P_H[\delta] \oplus P_L[12,16] \oplus C_H[\alpha] \oplus C_L[15] = K_1[19,23] \oplus L_3 \oplus L_7 \oplus L_{11} \oplus K_{15}[22]$	$1/2 + 1.19 \times 2^{-22}$
16	$P_H[\delta] \oplus P_L[12,16] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[19,23] \oplus L_3 \oplus L_7 \oplus L_{11} \oplus K_{15}[22] \oplus K_{16}[\gamma]$	$1/2 - 1.49 \times 2^{-24}$
17	$P_H[15] \oplus P_L[\beta,\delta] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[\gamma] \oplus K_2[22] \oplus L_4 \oplus L_8 \oplus L_{12} \oplus K_{16}[22] \oplus K_{17}[\gamma]$	$1/2 - 1.16 \times 2^{-26}$
18	$P_L[\alpha] \oplus C_H[\alpha] \oplus C_L[15] = L_2 \oplus L_6 \oplus L_{10} \oplus L_{14} \oplus K_{18}[22]$	$1/2 - 1.86 \times 2^{-28}$
19	$P_H[\alpha] \oplus P_L[15] \oplus C_H[\alpha] \oplus C_L[15] = K_1[22] \oplus L_3 \oplus L_7 \oplus L_{11} \oplus L_{15} \oplus K_{19}[22]$	$1/2 + 1.16 \times 2^{-28}$
20	$P_H[\alpha] \oplus P_L[15] \oplus C_H[15] \oplus C_L[\alpha,\beta] = K_1[22] \oplus L_3 \oplus L_7 \oplus L_{11} \oplus L_{15} \oplus K_{19}[22] \oplus K_{20}[\gamma]$	$1/2 - 1.46 \times 2^{-30}$
α: 7,18,24,29 β: 27,28,30,31 γ: 42,43,45,46 δ: 7,18,24 L_i: $K_i[22] \oplus K_{i+1}[44] \oplus K_{i+2}[22]$		

Table 4.13 Approximations of DES depending on number of rounds

Table contains best approximations for given number of rounds and their probability.

4.8 Algebraic and combined attacks on DES cipher

Algebraic attacks on DES cipher were performed by N. Courtois and G. V. Bard and described in (Courtois and Bard, 2006). To solve sets of equations, describing DES internal structure, they used statement, that this NP-hard problem is equivalent to another one problem of this type, namely satisfiability problem (SAT). Sets of equations, describing DES depending on number of rounds, were given in (Courtois, 2006). Authors were able to find the key for DES reduced to six rounds in 68 seconds assuming, that they know any 20 bits of the key. In (Courtois et al., 2012) it was certified that for algebraic attacks with small data complexity, DES reduced to eight rounds can be considered secure.

Attack on DES, joining algebraic techniques with differential cryptanalysis, was published in (Gasecki and Misztal, 2011). Using this type of attacks, four and six rounds of DES were broken. To successful performance of the attacks, there were two differential characteristics used, namely one-round characteristic with probability 1 allowing to add extra equations in attacks for four rounds and two-round iterative characteristic. Method, joining techniques mentioned, was also implemented and published in (Faugere et al., 2009), where attacks on 6, 7 and 8 rounds of DES were presented. In case of six rounds of DES, three-round differential characteristic introduced in (Biham and Shamir, 1990) was used. The attack on DES reduced to eight rounds was combined with exhaustive search of eight bits of the key.

Number of rounds	Cryptanalysis method	Number of ciphertexts	Time (in seconds)
6	Differential (Biham and Shamir, 1990)	240 (chosen)	< 1
	Differential (Knudsen, 1995)	46 (chosen)	< 10
	Algebraic attack (Courtois and Bard, 2006)	1 (known)	2^{25}
	Combined attack (Faugere et al., 2009)	32 (chosen)	3000
	Combined attack (Faugere et al., 2009)	22 (chosen)	< 36000
7	Combined attack (Faugere et al., 2009)	2000 (chosen)	10000
8	Differential (Biham and Shamir, 1990)	50000 (chosen)	100
	Linear (Matsui, 1993)	2^{20} (known)	40
	Combined attack (Faugere et al., 2009)	11500 (chosen)	2^{25}

Table 4.14 Summary of the attacks on reduced round DES

Table 4.14 (Faugere et al., 2009) contains comparison of attacks on DES, reduced to six, seven and eight rounds respectively. However, all mentioned attacks on eight rounds were combined with exhaustive search of part of the key bits.

4.9 Summary of the attacks on DES

The DES cipher has become a reason to discover and publish the most common cryptanalysis methods these days, namely differential and linear cryptanalysis. Collation of the attacks, relevant for purpose of this dissertation, is given in Table 4.15.

Number of rounds	Data (number of plaintexts)	Type of the attack	Sort of the attack
8	50000	Chosen plaintext	Differential cryptanalysis
16	2^{47}	Chosen plaintext	Differential cryptanalysis
8	2^{21}	Known plaintext	Linear cryptanalysis
16	2^{43}	Known plaintext	Linear cryptanalysis
6	2	Known plaintext	Algebraic attack (assumption: knowledge of 20 key bits)
8	11500	Chosen plaintext	Combined attack (assumption: knowledge of 8 key bits)

Table 4.15 Summary of the attacks on DES

Table 4.15 contains information about attacks on six and eight rounds of the cipher due to the fact, that attack on DES reduced to this number of rounds is introduced in this dissertation. It is easy to notice, that for algebraic and combined attacks data complexity is lower than in other attacks, however for them assumption is needed about knowledge of part of the key bits. On the other side, linear and differential attacks need huge amount of input data. This summary shows also attacks for full round DES with complexity lower than exhaustive search, however number of data needed to perform them makes them hard to implement in practice.

5 Cryptanalysis of the algorithms

The idea of mixing two cryptanalysis techniques, namely differential cryptanalysis and algebraic attack, was at first given in (Albrecht and Cid, 2008). Although most part of this work was revisited and rejected in (Wang et al., 2011), still some of the techniques shown there allow to improve the cryptanalysis. The attack performed here applies algebraic techniques combined with possibilities arisen by differential and linear cryptanalysis respectively. For the DES algorithm, the attack was implemented on the cipher reduced to six and eight rounds. In case of the SMS4 algorithm, the attack was implemented on five rounds.

5.1 Algebraic attack with differential cryptanalysis

The first step is to enhance algebraic attack by using more than one pair plaintext – ciphertext. Given two equation systems F and F^* for two plaintext – ciphertext pairs (P,C) and (P^*,C^*) under the same encryption key K, we can combine these equation systems to form a new set $\bar{F} = F \cup F^*$. Such set has twice as many equations as the original systems, however it provides many new variables. The next step is to take advantage of probabilistic dependencies given from differential cryptanalysis. This leads us to a new kind of the attack.

Let assume that attack is applied on cipher having Feistel structure where round function consists of permutations and non-linear transformations implemented as S-boxes. Moreover, assume that for such cipher we have differential characteristic for fixed number of rounds $\Omega = (\delta_0, \delta_1, \dots, \delta_r)$ where specific difference in i-th round occurs with a probability p_i. Then, probability of a characteristic Ω is equal to $p = \prod p_i$.

Each one-round difference gives rise to equations relating the input and output pairs for active S-Boxes. Let $X_{i,j}$ and $X_{i,j}^*$ denote the j-th bit of the input to the S-box layer in i-th round for the systems F and F^* respectively. Similarly, let $Y_{i,j}$ and $Y_{i,j}^*$ denote the correspondence output bits. Hence, we are given following expressions:

$$X_{i,j} + X_{i,j}^* = \Delta X_{i,j} \rightarrow \Delta Y_{i,j} = Y_{i,j} + Y_{i,j}^*$$

Values $\Delta X_{i,j}$ and $\Delta Y_{i,j}$ are given from differential characteristic. Similarly, for non-active S-boxes we have the expressions:

$$X_{i,j} + X^*_{i,j} = 0 = Y_{i,j} + Y^*_{i,j}$$

If we consider the equation system $\bar{F} = F \cup F^*$, we can combine it with additional equations given from differential characteristic. It leads to the new equation system $\bar{F}^+$ which holds with probability p. We can also perform this attack in a chosen plaintext scenario and use such pair which already satisfies differential characteristic used. Advantage of this solution against differential cryptanalysis is that in this scenario we need only one "right pair" which satisfies a set of differences at every round. We also expect system $\bar{F}^+$ to be easier to solve than original system while many linear constrains were added without adding any new variables.

A major difficulty in the differential part of our attacks is to find a "right pair" which satisfies a set of differences at every round. In order to find such a pair we first select pairs which satisfy differences at input and at the output, and expect that with high probability they also satisfy additional differences.

This makes a difference between typical differential cryptanalysis, where attacker cares only about distinguishers on observable properties, and presented attack, where we are taking advantage on all the relations arising in the middle of encryption.

This method was developed in this dissertation by adding to it the next step which should allow to increase its efficiency. It is done by concatenating sets of equations which describe more than two pairs: plaintext – ciphertext. Such sets do not share most of the state variables however they still share key variables. The result set is expected to be solved faster than both sets separately while it involves many new variables.

The attack was performed as follows. Two pairs, satisfying a particular characteristic were found. It means we had four plaintexts, P_1, P^*_1, P_2, P^*_2, and four corresponding ciphertexts, namely C_1, C^*_1, C_2, C^*_2. For every pair plaintext –

ciphertext, a set of equations, describing the way of encryption, was built. Then all sets were merged to the one set sharing the key variables. The next step was to enhance the final set with equations derived from differential characteristics. This was possible because of the fact that both pairs $(P_1, P_1^*) - (C_1, C_1^*)$ and $(P_2, P_2^*) - (C_2, C_2^*)$ were the right pairs, which means, they satisfied a specific differential characteristic as it was mentioned before. The idea of the attack is presented in Figure 5.1, where enhancement proposed in the dissertation is marked with colour.

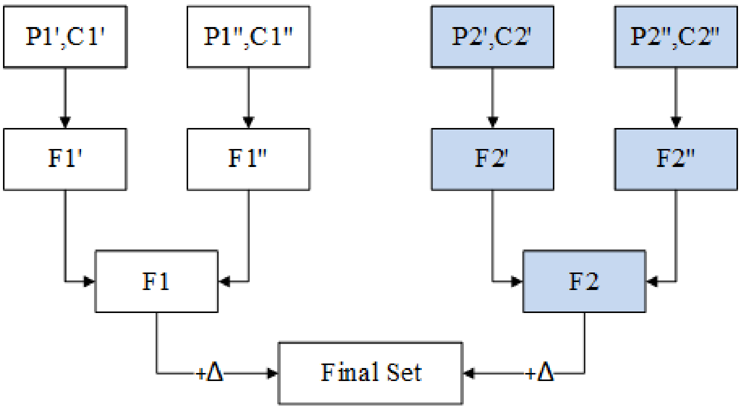


Figure 5.1 General scheme of the attack

Finally, all the equations were converted to a form possible to be solved by a SAT-solver. This technique was also applied for more than four plaintexts.

For purpose of the research made within this dissertation there was concatenation made for two, four, six and eight pairs of plaintext – ciphertext respectively when performing attack on the DES cipher. In case of the SMS4 cipher, concatenation was made for two and four pairs. The aim was to observe whether and how number of entry data impacts results obtained. Limit for the SMS4 cipher was lower because of the fact that set describing it is much bigger than in DES case. That is why it was considered pointless to concatenate more than four pairs, while such set would be too large to be solved.

5.2 Algebraic attack with linear cryptanalysis

The method described above was a contribution to introduce in this dissertation a new idea of joining different cryptanalysis methods. It is based on enhancement of algebraic attack with linear relations arisen from linear cryptanalysis. In opposite to previously described combined attack, here we do not use the correlations between pairs of plaintexts nor ciphertexts. Instead, we are just trying to add extra equations describing linear dependencies for substitute boxes (S-boxes), occurring with non-zero probability, with intention to make a set of equations easier to solve.

The implementation of the attack is as follows. At first, we are trying to find the best linear approximation for every single S-box, where by "the best" we mean approximation with the highest probability. This can be easily done by using linear profile of each S-box. Then, based on such approximations, we are adding extra equations, describing linear dependencies between input and output bits of S-boxes in every round. Probability P_L, that such set will have solution, is equal to $P_L = \prod_j \prod_i p_{ij}$ where i denotes number of S-box, j denotes number of round and p_{ij} denotes probability that linear approximation is satisfied for i-th S-box in j-th round. One thing, which has to be underlined here is, that these S-boxes are considered independently. That is the reason why probabilities are multiplied and Pilling-Up Lemma (Matsui, 1993) is not applied. The aim of the attack is to verify in experimental process two hypotheses, namely whether these extra equations can improve algebraic attack as it takes place in differential-algebraic attack and if it could be used as a filter for such plaintext – ciphertext pairs, which do not satisfy all the linear approximations given. For the second case we would expect to get the contradiction in short time when trying to solve such set of equations.

Next step which could allow to improve the attack is to use more than one pair plaintext – ciphertext to perform the attack. In this case, however, there is no correlation between different sets of equations apart from key variables. That is, why results expected should not be as good as it is in the case of differential – algebraic attack.

5.3 Performance of the attacks

Implementation of the attacks consists of several steps which have to be executed in order to proper performance of the attack.

Step 1

Creating the equations describing internal structure of the cipher. For DES cipher these equations were taken from (Courtois, 2006). For SMS4 cipher equations were prepared based on the description of the algorithm and only the algebraic description of substitute box was taken from (Erickson et al., 2009). In every case equations are given in algebraic normal form (ANF). Every equation contains only monomials of first and second degree and at most four addends. Single round of DES is described with 488 such equations, while one round of SMS4 with 7304 equations (without key scheduling). There was special notation used to create those equations, which facilitated building them and then debugging partial results, and is described in details in further part of this dissertation. Basics for this notation were also taken from (Courtois, 2006).

Step 2

Second step is to enhance equations describing particular number of rounds with equations determining values of plaintext and ciphertext. These are extra 128 equations for DES cipher and 256 for SMS4 cipher. After that, when preparing ANF set, addition is made of the equations arisen from differential characteristics or linear dependencies. Particular characteristics and dependencies used are introduced in subsection describing input data, prepared to perform the attacks. Depending on the characteristic, particular number of equations is added to the set. Total number of equations for each set is introduced in further part of this dissertation.

Step 3

Once ANF set of equations is prepared, it is possible to solve it. It is well known however that the problem of solving a multivariate simultaneous system of quadratic equations over *GF(2)* (the MQ problem) is NP-hard. Another NP-hard problem is finding a satisfying assignment for a logical expression in several variables (the SAT problem). While all NP-complete problems are polynomially equivalent, there was an investigation made in (Courtois et al., 2007) whether it is possible to apply tools, used to solve SAT problems, for MQ problem. They have found out that techniques in SAT-solvers allow to find solutions faster than exhaustive search for sparse and overdefined systems of equations. To perform experimental results they have used MiniSAT (Een and Sorensson, 2005) and this tool was also used to implement attacks within this dissertation.

5.3.1 Application of MiniSAT to perform algebraic attack

Algebraic attack is based on the description of an algorithm as a system of equations and then solving it to find bits of the key. Having system in alternative normal form (ANF) over *GF(2)* field it has to be converted to conjunctive normal form (CNF) which is applicable to solve by SAT-solver. This can be done by mathematics software with libraries implementing such conversion, like Sage (SAGE, 2008). However, as a part of research made within the dissertation, own methods were implemented and are described in details in further subsections.

5.3.2 Replacement of monomials

Logical expression of a form $(w \vee \bar{a})(x \vee \bar{a})(y \vee \bar{a})(z \vee \bar{a})(a \vee \bar{w} \vee \bar{x} \vee \bar{y} \vee \bar{z})$ is equivalent to logical expression $a \Leftrightarrow (w \wedge x \wedge y \wedge z)$ which could be also expressed as an equation $m = wxyz$ over *GF(2)* field. Based on it, every monomial of degree higher than 1 is replaced with product containing additional dummy variable. Total number of sums being a factors for this product is equal to $d + 1$ for particular monomial, where d denotes degree of the monomial. Such replacement is made only once for every monomial occurring in the equation system.

5.3.3 Conversion of linear equation system to conjunctive normal form

Every polynomial is sum of the variables, which is equivalent to logical operation of exclusive disjunction operation (XOR). Such operations containing plenty of arguments are tough for SAT-solvers. In particular, sum $(a + b + c + d) = 0$ over *GF(2)* field is equivalent to following logical operation given in Equation 5.1.

$$(\bar{a} \vee b \vee c \vee d)(a \vee \bar{b} \vee c \vee d)(a \vee b \vee \bar{c} \vee d)(a \vee b \vee c \vee \bar{d})$$

$$(\bar{a} \vee \bar{b} \vee \bar{c} \vee d)(\bar{a} \vee \bar{b} \vee c \vee \bar{d})(\bar{a} \vee b \vee \bar{c} \vee \bar{d})(a \vee \bar{b} \vee \bar{c} \vee \bar{d})$$

Equation 5.1

This equation is product of logical sums of all variables occurring in the equation, with all combinations of one or three negations. This is based on the fact, that XOR operation for *n* arguments is equivalent to factor of applicable sums of particular variables. Each sum is built by negation of combination of k variables, where k takes values of all odd numbers less than or equal to n. In case of sum with l variables such operation needs 2^{l-1} factors, called also clauses.

While number of clauses depends exponentially on *l* value, it is worth to cut each sum into subsums of particular length in order to reduce number of clauses and then to join the results given. That is, why in research presented in this dissertation, all the equations were converted to a form having maximum four addends.

5.3.4 Format of CNF file

MiniSAT program, which was used here, solves equations given as a CNF file. The first line determines number of variables and clauses and is of a form *p cnf variables clauses*. Each successive line determines clause, where positive number assigned to a particular variable denotes it as a normal form and negative number denotes its negation. Each line should finish with zero. At the beginning of the file commentary lines could be put and they are denoted with letter *c* at the beginning of each line. Example CNF file is given below:

```
c Sample CNF file
p cnf 3 2
1 -3 0
2 3 -1 0
```

Example given above is a representation of logical formula $(a \vee \bar{c})(b \vee c \vee \bar{a})$ where variables *a, b, c* are denoted by numbers 1, 2 and 3 respectively.

5.3.5 Scheme of solving system of equations using MiniSAT

In case of equation systems, they are solved as follows. At first, each equation is introduced in conjunctive normal form. Then, all clauses are put into the CNF file. Let assume, that we have simple system of two equations over *GF(2)* field to solve, given as:

$$\begin{cases} a + \bar{b} + c = 0 \\ \bar{c} + d = 0 \end{cases}$$

At first we convert the equations to product of logical sums:

$$\begin{cases} (\bar{a} \vee \bar{b} \vee c)(a \vee b \vee c)(a \vee \bar{b} \vee \bar{c})(\bar{a} \vee b \vee \bar{c}) = 0 \\ (c \vee d)(\bar{c} \vee \bar{d}) = 0 \end{cases}$$

Having such form, we are able to put it into the CNF file, containing six clauses and four variables. Content of this file is given below.

```
p cnf 4 6
-1 -2 3 0
1 2 3 0
1 -2 -3 0
-1 2 -3 0
3 4 0
-3 -4 0
```

The file is put in the same location as MiniSAT program. Then, from command line and this location, following command is executed:

```
minisat [input file][output file]
```

Solution, of a form given below, will be put in the output file.

```
SAT
-1 -2 3 -4 0
```

Such result means, then variables denoted as 1,2 and 4 (in our case these are *a, b* and *d*) have to be equal to zero, while variable *c* denoted as 3 has to be equal to one. It is important to keep in mind, that in general solution given does not have to be the only one proper, however program gives the first result found. This is especially significant for equation sets with parameter.

If system of equations given is impossible to solve, output file would contain only message *UNSATISFIABLE*.

5.3.6 Execution of the attack

According to the all steps given above, attack was performed for reduced round DES and SMS4 as it was mentioned in previous subsections. Following subsections describe specification of entry data used to perform attacks, systems of equations generated for each case, detailed conversion from alternative normal form to conjunctive normal form along with a way of ANF description and all the implementations and algorithms applied to facilitate execution of the attacks. All the attacks were performed on a PC computer equipped with 1.73Ghz dual core processor and 2 gigabytes of RAM. MiniSat program was run under Linux Ubuntu operating system. In case of differential-algebraic attack on reduced round DES, results presented further are significant enhancement of the results published in (Gasecki, 2013).

5.4 Entry data used to perform combined attacks on DES

This subsection describes entry data, which were used to perform combined attacks on reduced round block cipher DES.

5.4.1 Algebraic attack with differential cryptanalysis

Data sets were generated for six different, randomly chosen keys. None of the keys is weak or half-weak.

MK1 = CA113287AADE2364$_X$

MK2 = A6F21110F26AAAAF$_X$

MK3 = 00349FF2AB3D1120$_X$

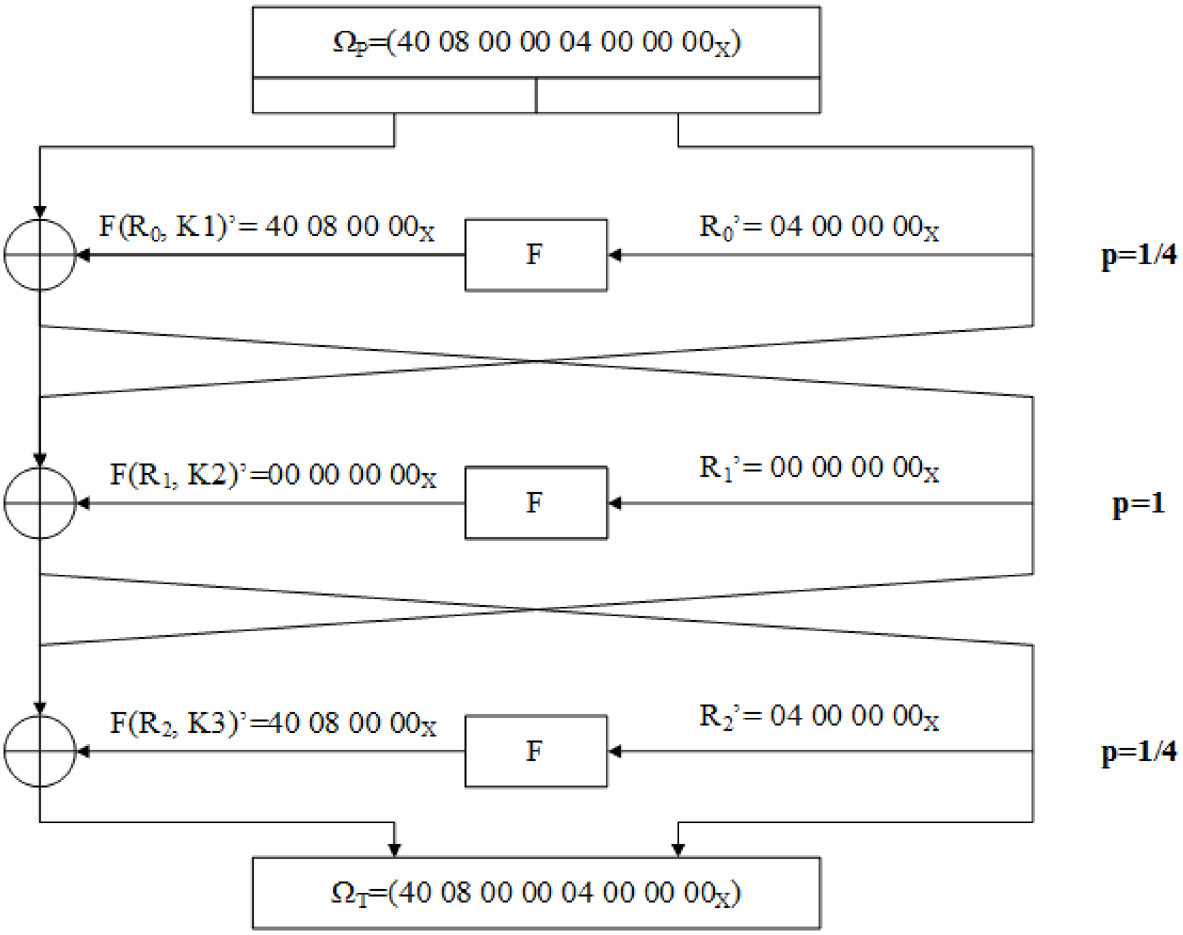
MK4 = 558AC3019FF43A87$_X$

MK5 = 10DE4912AA30874B$_X$

MK6 = CF3018B271642CF1$_X$

First set – 6 rounds

First set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data was generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.2.



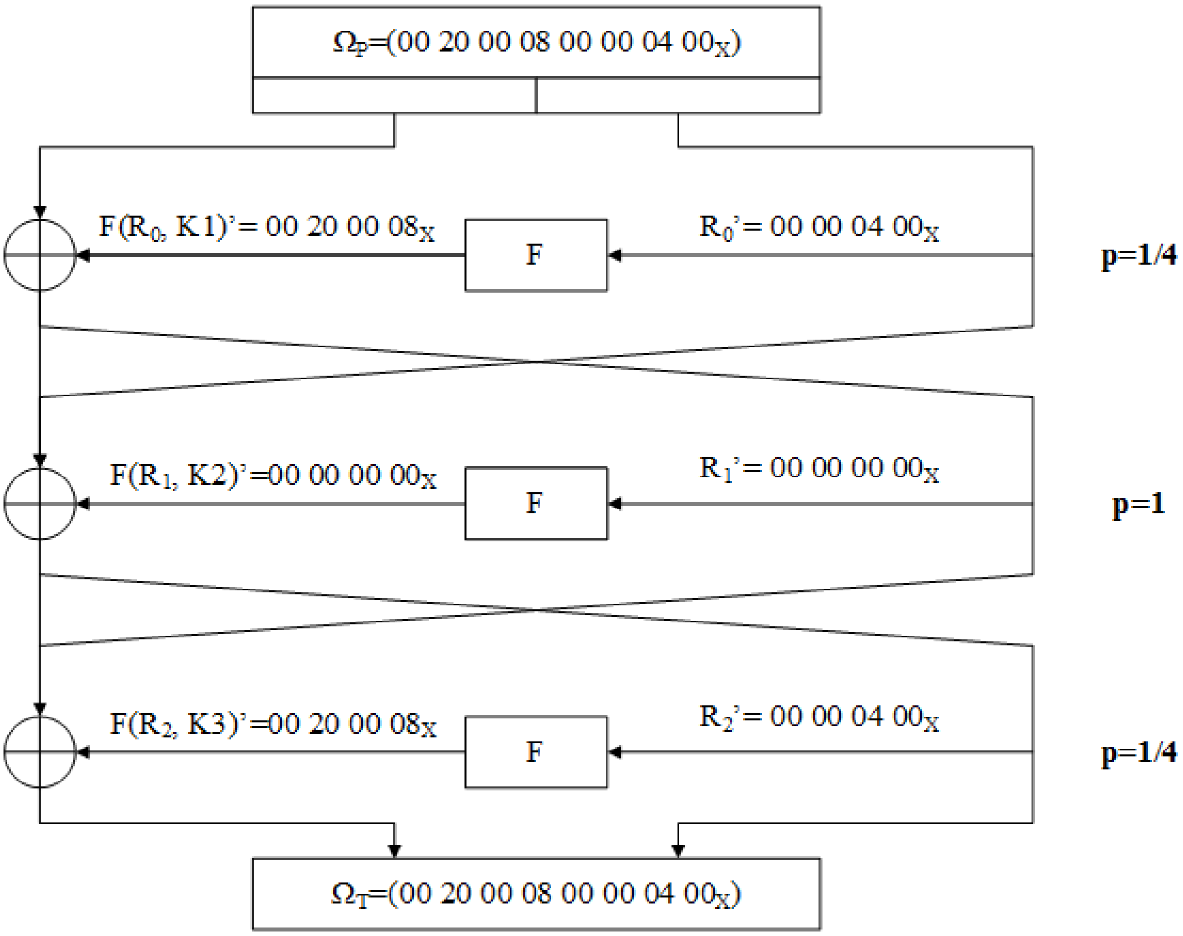
Ω_P=(40 08 00 00 04 00 00 00_X)
F(R_0, K1)'= 40 08 00 00_X
F
R_0'= 04 00 00 00_X
p=1/4
F(R_1, K2)'=00 00 00 00_X
F
R_1'= 00 00 00 00_X
p=1
F(R_2, K3)'=40 08 00 00_X
F
R_2'= 04 00 00 00_X
p=1/4
Ω_T=(40 08 00 00 04 00 00 00_X)

Figure 5.2 First three-round differential characteristic

Input difference equals $\Omega_P = 40\ 08\ 00\ 00\ 04\ 00\ 00\ 00_X$. Difference between ciphertexts is irrelevant in this case. Probability, that randomly chosen pair satisfies such characteristic, equals 1/16.

Second set – 6 rounds

Second set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data was generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.3.



Ω_P=(00 20 00 08 00 00 04 00$_X$)
F(R$_0$, K1)'= 00 20 00 08$_X$
F
R$_0$'= 00 00 04 00$_X$
p=1/4
F(R$_1$, K2)'=00 00 00 00$_X$
F
R$_1$'= 00 00 00 00$_X$
p=1
F(R$_2$, K3)'=00 20 00 08$_X$
F
R$_2$'= 00 00 04 00$_X$
p=1/4
Ω_T=(00 20 00 08 00 00 04 00$_X$)

Figure 5.3 Second three-round differential characteristic

Input difference equals $\Omega_P = 00\ 20\ 08\ 00\ 00\ 00\ 04\ 00_X$. Difference between ciphertexts is irrelevant in this case. Probability, that randomly chosen pair satisfies such characteristic, equals 1/16.

Third set – 6 rounds

Third set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data were generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.4.

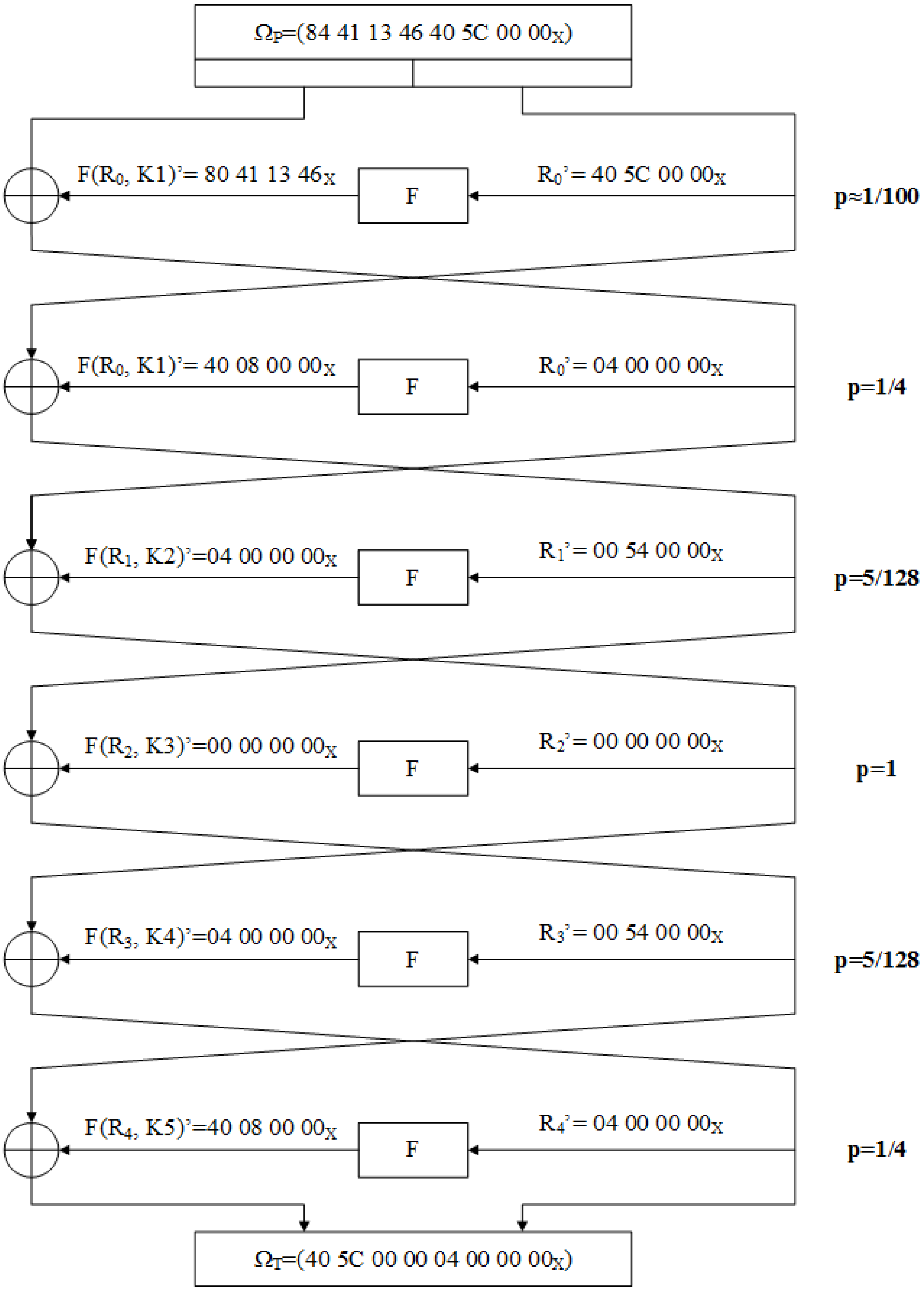


Figure 5.4 Six round differential characteristic

Input difference equals $\Omega_P = 84\ 41\ 13\ 46\ 40\ 5C\ 00\ 00_X$. Difference between ciphertexts equals $\Omega_T =\ 04\ 00\ 00\ 00\ 40\ 5C\ 00\ 00_X$. Probability, that randomly chosen pair is satisfying such characteristic, is about 1/1000000.

Fourth set – 8 rounds

Fourth set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data were generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.5.

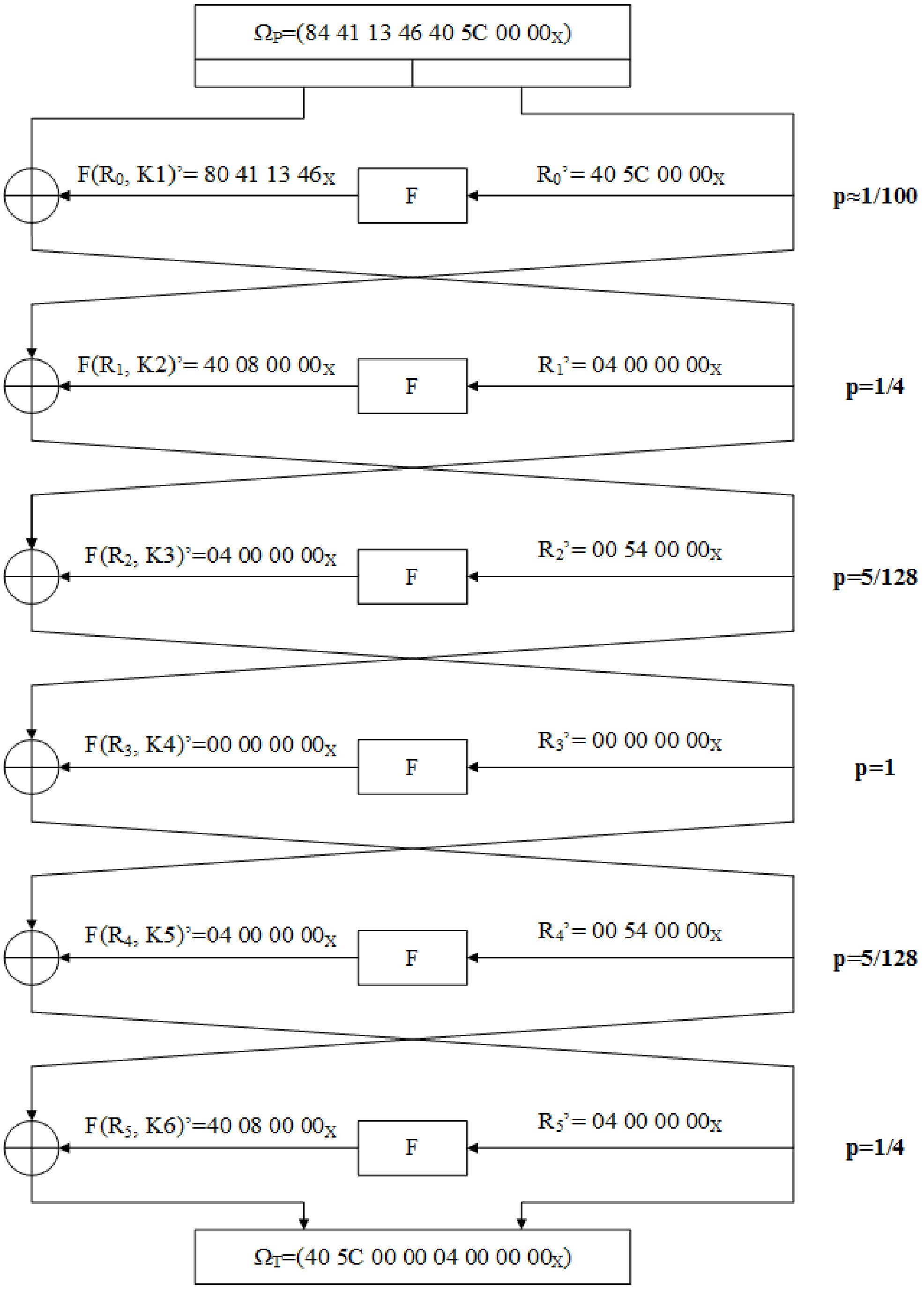


Figure 5.5 Six round differential characteristic

Input difference equals $\Omega_P = 84\ 41\ 13\ 46\ 40\ 5C\ 00\ 00_X$. Difference between ciphertexts is irrelevant in this case. Probability, that randomly chosen pair satisfies such characteristic, is about 1/1000000.

Fifth set – 8 rounds

Fifth set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data were generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.6.

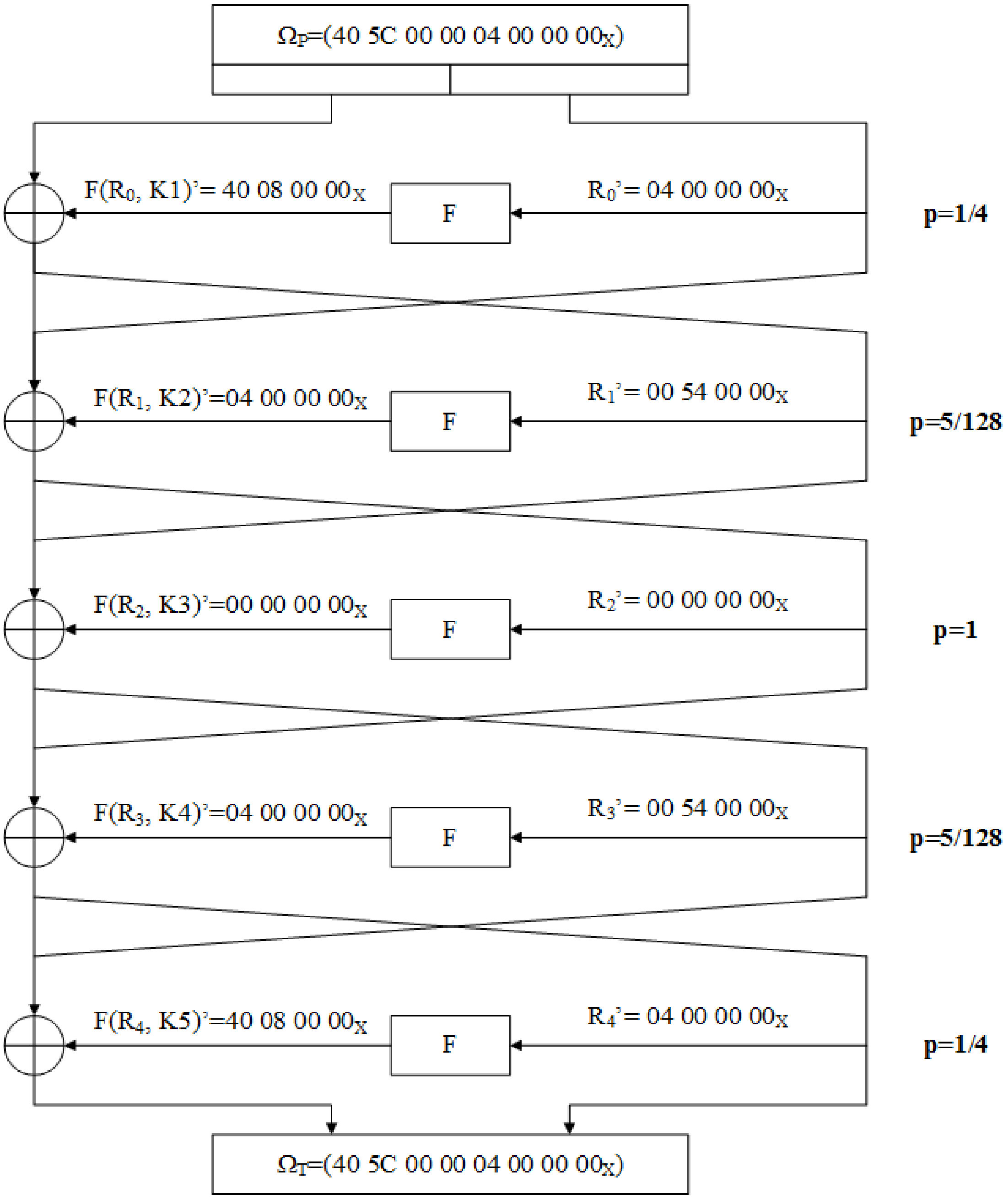


Figure 5.6 Five round differential characteristic

Input difference equals $\Omega_P = 40\ 5C\ 00\ 00\ 04\ 00\ 00\ 00_X$. Difference between ciphertexts is irrelevant in this case. Probability, that randomly chosen pair satisfies such characteristic, is about 1/10000.

Sixth set – 8 rounds

Sixth set consists of 16 plaintexts and 16 corresponding ciphertexts for every key. This data were generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristics presented in Figure 5.7.

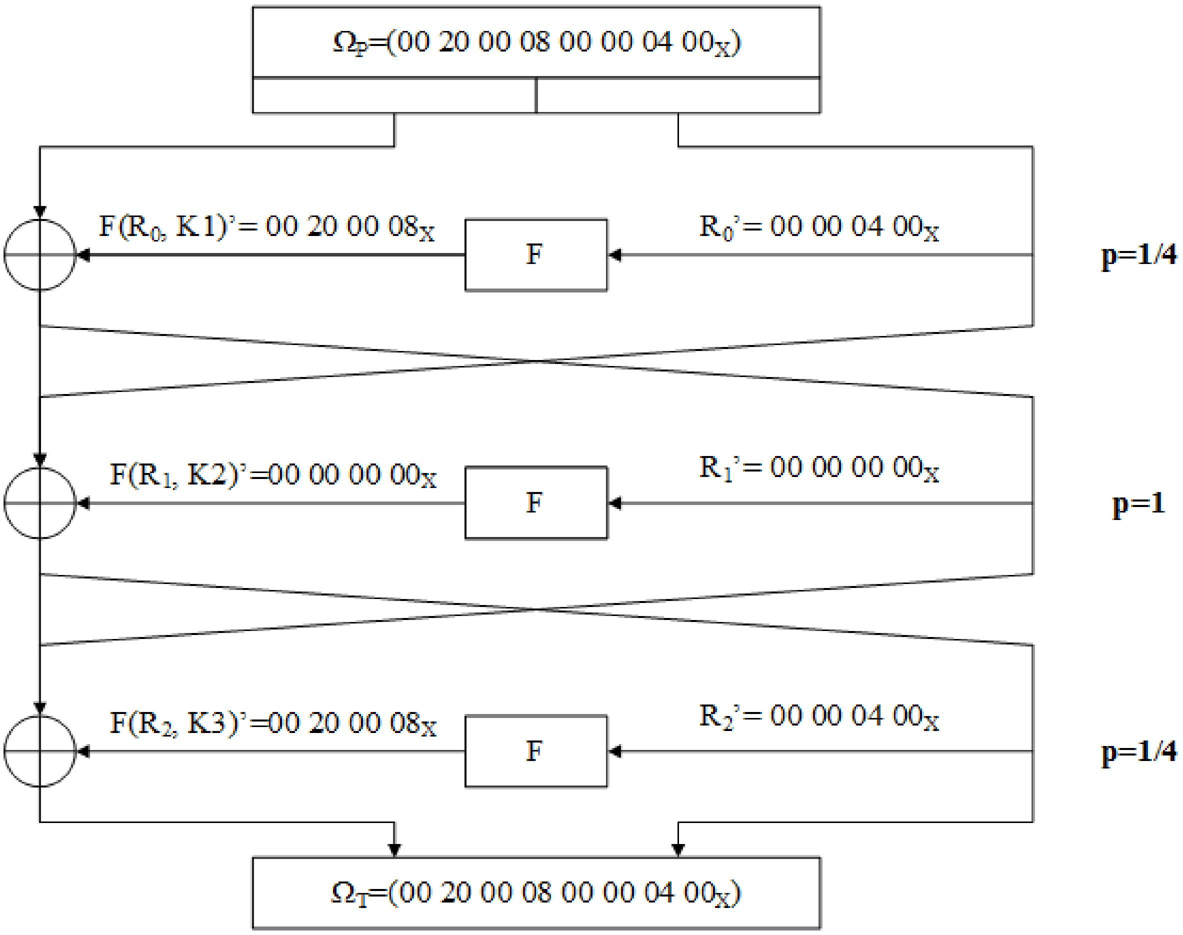


Figure 5.7 Three round differential characteristic

Input difference equals $\Omega_P = 00\ 20\ 08\ 00\ 00\ 00\ 04\ 00_X$. Difference between ciphertexts is irrelevant in this case. Probability, that randomly chosen pair satisfies such characteristic, is 1/16.

5.4.2 Algebraic attack with linear cryptanalysis

Data sets were generated for six different, randomly chosen keys. None of the keys is weak or half-weak.

MK1 = CA113287AADE2364$_X$

MK2 = A6F21110F26AAAAF$_X$

MK3 = 00349FF2AB3D1120$_X$

MK4 = 558AC3019FF43A87$_X$

MK5 = 10DE4912AA30874B$_X$

MK6 = CF3018B271642CF1$_X$

Every pair satisfies particular linear dependencies between input and output bits of S-box. Let us denote by I_1 to I_{32} input bits to round function F. Similarly, let O_1 to O_{32} denote output bits for that function and K_1 to K_{48} bits of the round key. Then equations, which hold for each round (with probability significantly different than ½), can be described as it is given in Table 5.1.

S-box number	Equation	Probability
1	$I_1 + K_2 = O_9 + O_{17} + O_{23} + O_{31} + 1$	0,78
2	$I_4 + K_7 + I_8 + K_{11} = O_2 + O_{13} + O_{18} + 1$	0,75
3	$I_8 + K_{13} + I_{12} + K_{17} = O_6 + O_{16} + O_{24} + O_{30} + 1$	0,75
4	$I_{12} + K_{19} + I_{14} + K_{21} = O_1 + O_{10} + O_{20} + O_{26} + 1$	0,75
5	$I_{17} + K_{26} = O_3 + O_8 + O_{14} + O_{25} + 1$	0,81
6	$I_{21} + K_{32} = O_{11} + O_{19} + O_{29} + 1$	0,72
7	$I_{24} + K_{37} + I_{28} + K_{41} = O_{12} + O_{22} + O_{32} + 1$	0,75
8	$I_{29} + K_{44} = O_5 + O_{15} + O_{21} + O_{27} + 1$	0,75

Table 5.1 Linear equations for the DES substitute boxes

Illustration of equations above is also given in Figure 5.8 showing round function of DES cipher. Every dependency between input, output and key bits is marked with different colour.

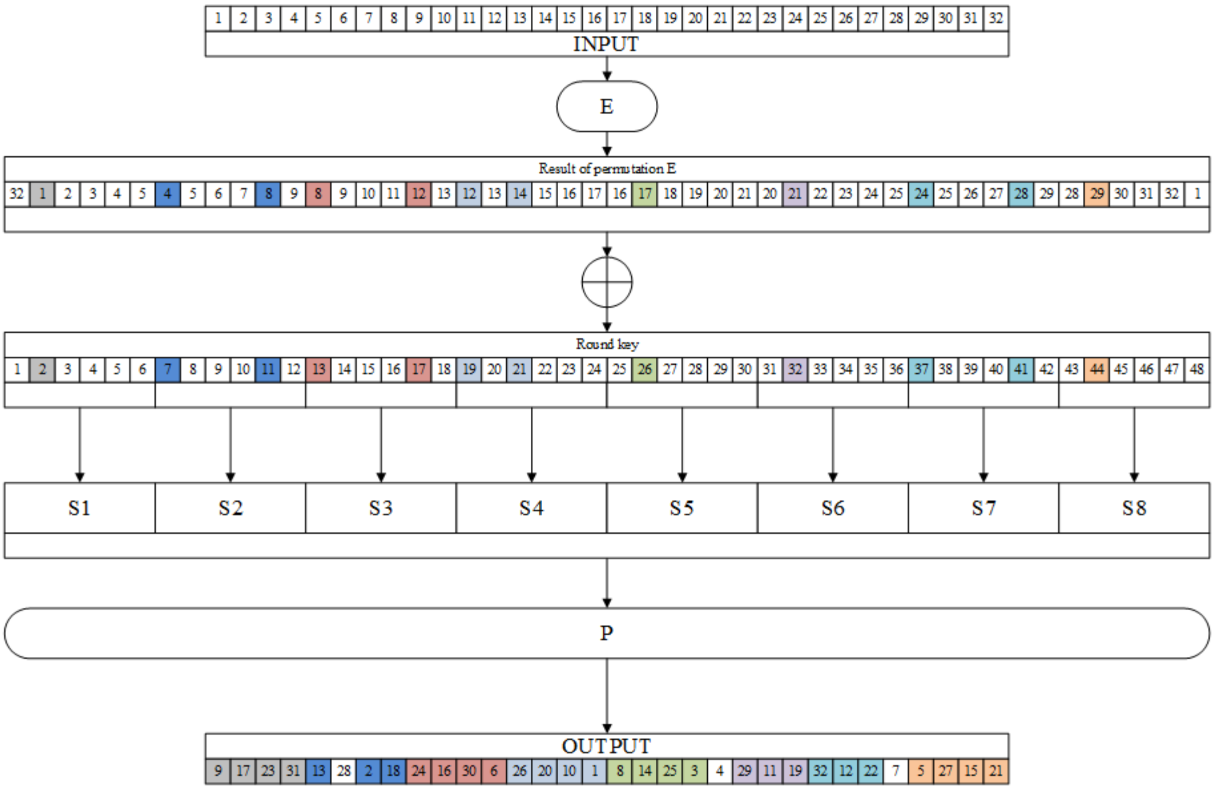


Figure 5.8 DES round function with linear dependencies marked

Probability that linear approximation is satisfied for every S-box within one round is equal to $(0{,}75)^5 * 0{,}72 * 0{,}78 * 0{,}81 = 0{,}10795 \approx 2^{-3{,}2}$. This means that for six rounds randomly chosen pair satisfies all given linear approximations with probability about 2^{-20}.

5.5 Entry data used to perform combined attacks on SMS4

5.5.1 Algebraic attack with differential cryptanalysis

The attack was performed on the cipher reduced to five rounds. Data sets were generated for ten different keys where nine of them were chosen randomly and the first one is the key used in the specification (Diffie and Ledin, 2008) of the cipher for purpose of calculation the test vector.

MK1 = $0123456789ABCDEFFEDCBA9876543210_X$

MK2 = $1B812A120E62B81D541297ED6FB76F49_X$

MK3 = $294395FC16A4E57430D1BE020020B8D6_X$

MK4 = 128B3EDF3E7ABA1301DEEFDF18A7D9E0$_X$

MK5 = 5186409736568C16244202762EFA68F2$_X$

MK6 = 11CCCB9B41E9B48B6347D1073B3183EF$_X$

MK7 = 5F607AC06926DF682945EB9F0C511A0F$_X$

MK8 = 5D76C9CB0994CF442FF0577962F08667$_X$

MK9 = 49649DAE16A08A2B156A74D73EB319DD$_X$

MK10 = 09380551028DD027223BEC243CE88A1A$_X$

First set – 5 rounds

First set consists of 8 plaintexts and 8 corresponding ciphertexts for every key. This data was generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristic which was presented in Figure 4.3.

It means that both input and output differences of such pairs are equal to $\Omega_P = \alpha\, \alpha\, \alpha\, 0$, where value of $\alpha = 00C30290_X$ was taken from (Zhang et al., 2008). Probability, that randomly chosen pair satisfies such characteristic, equals to 2^{-42}. While this value is relatively low and makes it tough to find proper data on PC computer, hence special algorithm, constructed for purpose of this dissertation, was used to generate data used to perform the attack.

At first, one pair satisfying characteristic for the particular key was found. Then, based on it, the input and output values were found for both 4th and 5th round where by input we mean value before adding round key. These values are marked on Figure 5.9 below as *R4*, *O4* (input and output for 4th round) and *R5*, *O5* (input and output for 5th round). Moreover, we know the value *Y5* which is the output coming after 5th round.

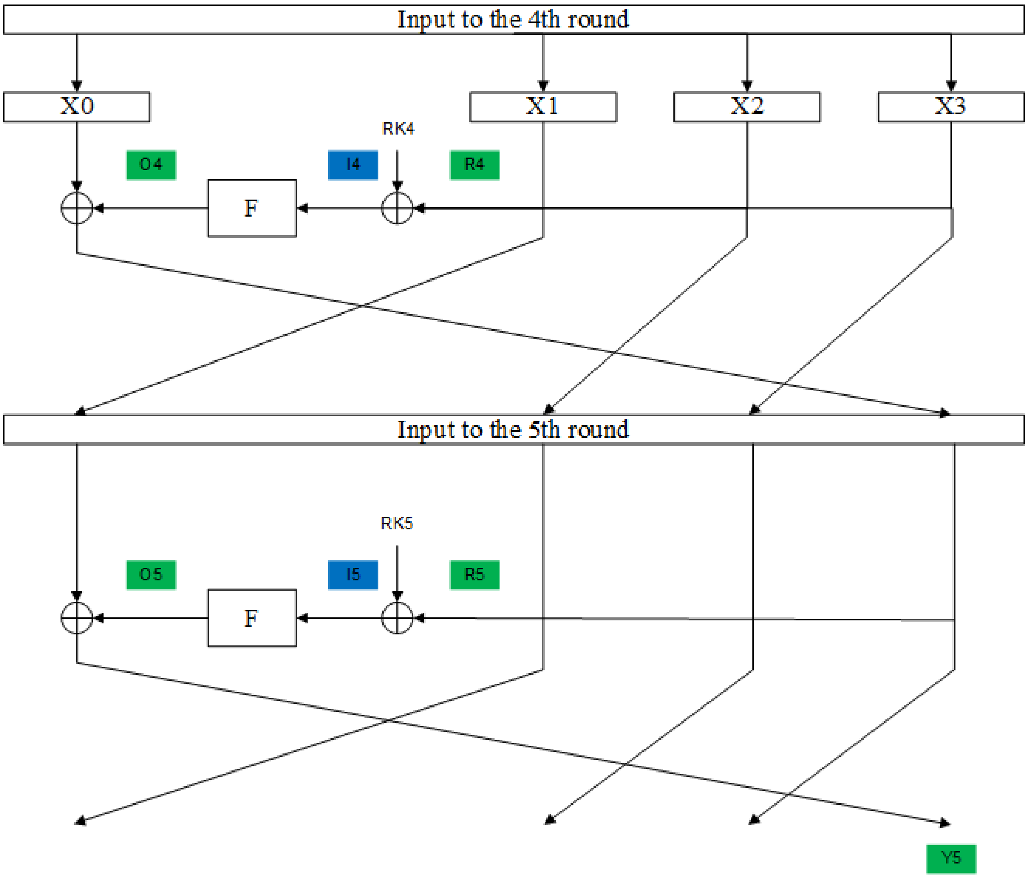


Figure 5.9 Scheme of 4th and 5th round of SMS4

Based on these values, following variables could be calculated:

$$\begin{aligned} I4 &= R4 + RK4 \\ I5 &= R5 + RK5 \\ X1 &= O5 + Y5 \\ X0 &= I4 + I5 + X1 + O4 \\ X3 &= I4 + X1 + X2 \end{aligned}$$

By *X0*, *X1*, *X2* and *X3* we denote input to the 4th round which makes data satisfy differential characteristic for 4th and 5th round. Value *X2* is a parameter, so its value can be chosen a priori and *RK4* and *RK5* are round keys for this key, which we are looking data satisfying characteristic for. Every sum above is calculated over *GF(2)*.

Having the input to the 4th round we could then decipher it using first three round keys (in reverse order) and receive input data satisfying differential characteristic.

Second set – 5 rounds

Second set consists of 8 plaintexts and 8 corresponding ciphertexts for every key. This data was also generated as pairs: two plaintexts – two ciphertexts. Each pair satisfies differential characteristic presented in Figure 5.10 below.

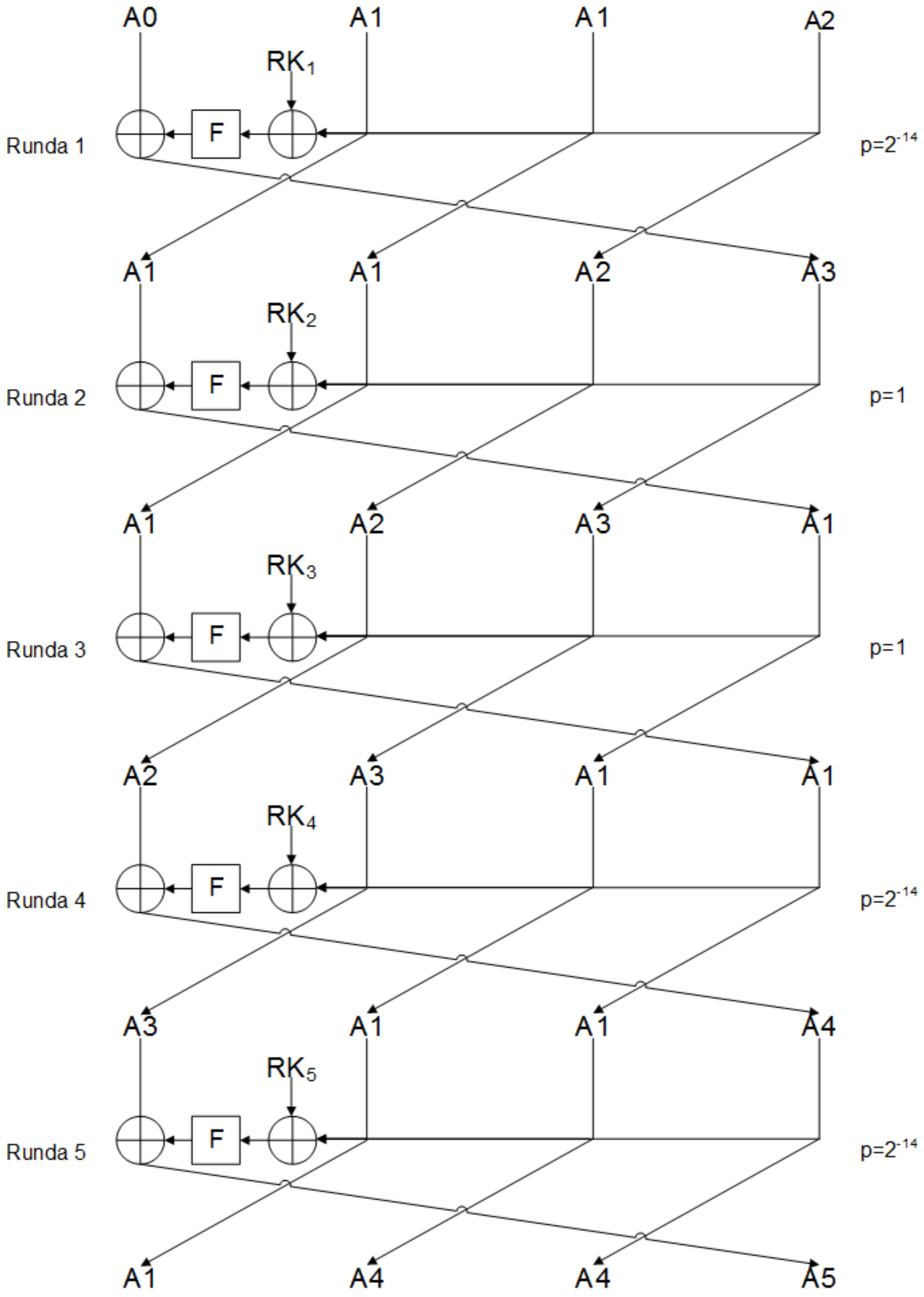


Figure 5.10 Differential characteristic of the SMS4 cipher

Constant values A0 to A4 were taken from (Su et al., 2010) and are: $A0 = E5992D8C_X$, $A1 = F3F30033_X$, $A2 = F3000030_X$, $A3 = 00F30003_X$, $A4 = 00CF0033_X$, $A5 = F33C0000_X$

While probability, that randomly chosen pair is satisfying such characteristic, equals 2^{-42} then the same algorithm to find proper data was used, which was introduced in previous section. However here slight modification had to be done, because probability for 1st round is not equal to 1. It means, that we could not just decipher input to the 4th round found by the algorithm and get proper input data. Nonetheless one value, namely *X2*, was a parameter in the algorithm. Hence, by modifying this value, we could assume that with probability 2^{-14} we will find data which, after deciphering, will satisfy input difference.

Moreover, one thing is worth to notice for generated data sets. First set could be also applied to attack on six, seven and eight rounds of SMS4 while we could concatenate iterative differential characteristic. Probability for the first three rounds is equal to 1, hence data will always satisfy input and output differences for each round. Second set could be applied to attack on six and seven rounds of SMS4, while characteristic described above is a part of 19-round differential characteristic, where differences for 6th and 7th round hold a probability equal to 1.

5.5.2 Algebraic attack with linear cryptanalysis

Data sets were generated for six different keys where five of them were chosen randomly and the first one is the key used in the specification of the cipher for purpose of calculation test vector.

$MK1 = 0123456789abcdeffedcba9876543210_X$

$MK2 = 1B812A120E62B81D541297ED6FB76F49_X$

$MK3 = 294395FC16A4E57430D1BE020020B8D6_X$

$MK4 = 128B3EDF3E7ABA1301DEEFDF18A7D9E0_X$

$MK5 = 5186409736568C16244202762EFA68F2_X$

$MK6 = 11CCCB9B41E9B48B6347D1073B3183EF_X$

Each pair satisfies linear equations describing relations between input and output values of S-box. This equation could be described as:

$$I_3 + I_4 + I_5 + I_7 + I_8 = O_3 + O_4 + O_8 + 1$$

In equation above I_k means k-th bit of input to the S-box and O_k means k-th output from the S-box and sum is calculated over *GF(2)*. Illustration of this equation is also given in Figure 5.11 showing round function of SMS4 cipher. Dependencies between input and output bits for each S-box are marked with different colour.

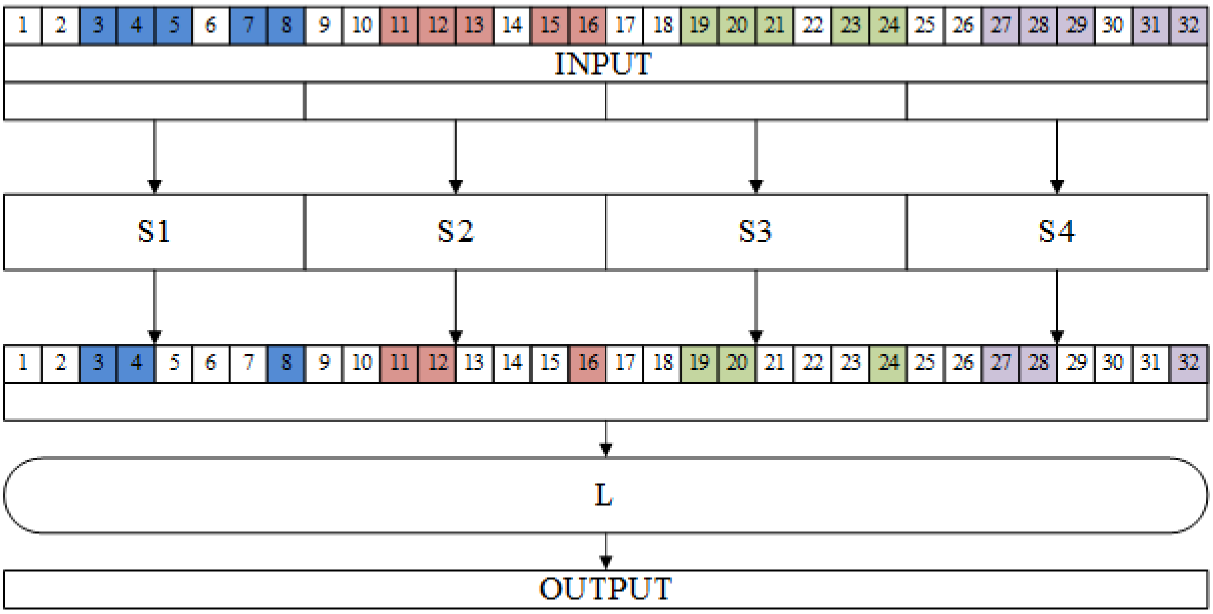


Figure 5.11 SMS4 round function with linear dependencies marked

The equation is satisfied for particular S-box with probability of $\frac{1}{2} + \frac{1}{16} = 0{,}5625$. This means, probability that such linear approximation is satisfied for every S-box within one round is equal to $(0{,}5625\,)^4 \approx 0{,}1$. For five rounds that gives us probability $0{,}00001 = 10^{-5} \approx 2^{-17}$ that randomly chosen pair satisfies all given linear approximations.

5.6 Implementation of the attack

Sets of equations for both DES and SMS4 ciphers consist of two parts. Equations based on plaintext and corresponding ciphertext are the first part. The second part consist of equations describing structure of cipher. Number of equations and variables are presented in Table 5.2, Table 5.3 and Table 5.4 below.

Table 5.2 contains information about sets describing SMS4 cipher reduced to five rounds. First column contains number of pairs plaintext – ciphertext used to construct set of equations. In following three columns it was marked whether extra equations were added, coming from linear approximation, five-round iterative differential characteristic or five-round non-iterative differential characteristic respectively. Last two columns contain information about number of equations and variables in each set.

Number of texts	LIN	ITER	DIFF	Number of equations	Number of variables
1				44623	44390
1	+			44708	44459
2				89246	88558
2			+	90014	88558
2		+		90014	88558
4				178492	176894
4			+	180028	176894
4		+		180028	176894

Table 5.2 Sets for 5 rounds of the SMS4 cipher

Table 5.3 contains information about sets describing DES cipher reduced to six rounds. First column contains number of pairs plaintext – ciphertext used to construct set of equations. In following four columns it was marked whether extra equations were added based on linear approximation, first or second three round differential characteristic or six-round differential characteristic respectively. Last two columns contain information about number of equations and variables in each set.

Number of texts	LIN	3CHAR1	3CHAR2	6CHAR	Number of equations	Number of variables
1					3056	3049
1	+				3200	3147
2					6112	6041
2	+				6400	6263
2		+			6352	6041
2			+		6352	6041
2				+	6592	6041

4					12224	12025
4		+			12704	12025
4			+		12704	12025
4				+	13184	12025
6					18336	18009
6		+			19056	18009
6			+		19056	18009
6				+	19776	18009
8					24448	23993
8		+			25408	23993
8			+		25408	23993
8				+	26368	23993

Table 5.3 Sets for six rounds of the DES cipher

Table 5.4 contains information about sets describing DES cipher reduced to eight rounds. First column contains number of pairs plaintext – ciphertext used to construct set of equations. In following two columns it was marked whether extra equations were added based on five-round differential characteristic or six-round differential characteristic respectively. Last two columns contain information about number of equations and variables in each set.

Number of texts	5CHAR	6CHAR	Number of equations	Number of variables
1			4032	4025
2			8064	7993
2	+		8464	7993
2		+	8544	7993
4			16128	15929
4	+		16928	15929
4		+	17088	15929
6			24192	23865
6	+		25392	23865
6		+	25632	23865
8			32256	31801
8	+		33856	31801
8		+	34176	31801

Table 5.4 Sets for eight rounds of the DES cipher

In every table, number of equations and variables is taken from sets converted to a form with additional variables, where every single equation contains only monomials of maximum second degree and maximum number of four components.

5.7 Description of sets of equations

In every set of equations of ANF form special way of describing variables was used. Every symbol described particular variable and the method to create those symbols allowed to precise exactly what the meaning of variable was.

For description of DES cipher, following symbols were used:

- I_*x_y_z* – marker of the input bit to the particular round function, where

    x – denotes number of the round. If the number is zero, then it symbolises left half of plaintext. If the number is equal to number of rounds plus one, then it symbolises left half of ciphertext.

    y – this is the number of pair used

    z – takes values from 1 to 32 and describes particular bit

- O_*x_y_z* – denotes output bit from round function, where

    x – denotes number of the round

    y – this is the number of pair used

    z – takes values from 1 to 32 and describes particular bit

- JS*m_x_y_z* – denotes input bit to S-BOX where

    m – symbolises number of S-BOX

    x – denotes number of the round

    y – this is the number of pair used

    z – takes values from 1 to 32 and describes particular bit

- t*xyz* – denotes additional variables, where

    x – denotes round number (with two digits)

    y – this is the number of pair used, also written with two digits

    z – three-digit ordering number, starting with 101

- k_*x* – denotes key bits, where

  x – symbolises number of bit from main key (with two digits)

For description of SMS4 cipher, following symbols were used:

- I_*x_y_z* – marker of the input bit to the particular round function, where

  x – denotes number of the round (where digit uses is equal to round number minus one)

  y – this is the number of pair used

  z – three-digit value, where first digit describes number of the word (from 1 to 4), and last two digits denotes number of bit within the word

- O_*x_y_z* – denotes output bit from particular round, where

  x – denotes number of the round

  y – this is the number of pair used

  z – three-digit value, where first digit describes number of the word (from 1 to 4), and last two digits denote number of bit within the word

- JS*m_x_y_z* – denotes input bit to S-BOX where

  m – symbolises number of S-BOX

  x – denotes number of the round

  y – this is the number of pair used

  z – takes values from 1 to 32 and describes particular bit

- t*xyz* – additional variable, where

  x – denotes round number (with two digits)

  y – denotes number of pair used, also written with two digits z – three-digit ordering number, where first digit denotes phase of the enciphering process: 1 – input to the S-box, 5 – linear transformation, 6 – post-linear transformation

- t*abcde* – additional variable, used particularly for SBOX function description, where

a – denotes number of bit within SBOX (one digit)

b – number of SBOX (one digit)

c – number of round (one digit)

d – number of pair used (one digit)

e – three-digit ordering number, starting with 101

- RKx_*y* – denotes key bits, where

x – symbolises number of bit of round key

y – symbolises number of round

Moreover, similar symbols were used to describe key schedule process (only for 5th round, while bits of round keys for first four rounds are independent and key schedule process does not have to be described for them, when creating algebraic attack). The only difference was that instead of *t* and *JS* beginning of the symbols, *tk* and *JSk* beginnings were used. Also, additional symbols was used to describe *CK* constants, having format *CKx_y*, where *x* denotes number of the constant and *y* denotes number of particular bit of the constant.

5.8 ANF to CNF conversion

Conversion to CNF file was performed two-ways. First method, called binomial conversion, looked for all monomials of second degree in set of equation and marked them at first with numeric symbol. Then, every equation was converted and if such monomial was found, marker previously assigned was taken. Table 5.5 contains first part of pseudo code describing assignment of numeric symbols to all variables existing in the set. In case of monomials of second degree, every factor is given separate symbol.

```
Declare symbols As Dictionary
Declare variable As Integer

variable := 3; /* variables 1 and 2 are assigned to values 1 and 0 respectively */

foreach (Sum in ANF)
        Sum := Replace '*' with '+' in Sum;
        Declare Sym as Array
        new Array Sym := Split Sum by '+' sign;
        foreach(symbol in Sym)
                symbols[symbol] = variable;
                variable := variable + 1;
        end foreach
end foreach
```

Table 5.5 Assignment of numeric variables to existing symbols

Table 5.6 contains pseudo code for binomial conversion. The algorithm is based on two iterations. During the first one, every monomial of second degree is converted to set of clauses acceptable by MiniSAT. Second iteration converts all equations where monomials of second degree are replaced by symbols previously assigned. Details of the method converting separate equations was shown in following table, containing code for simple conversion.

```
foreach (Sum in ANF)
        if Sum contains monomial of second degree then
                if exists symbols[monomial] then continue;
                else
                symbols[monomial] := variable;
                Array Components := Split monomial by '*' sign
                WriteClause(-variable Symbols[Components[0]] 0);
                WriteClause(-variable Symbols[Components[1]] 0);
                WriteClause(variable -Symbols[Components[0]] -Symbols[Components[1]] 0);
                variable := variable + 1;
                end if
        end if
end foreach

foreach (Sum in ANF)

        Array Components := Split Sum by '+'
        Convert to set of clauses depending on length of Components
        Put clauses in external file
end foreach
```

Table 5.6 Pseudo code for binomial conversion

The second method, called simple conversion, omits first phase of looking for second degree monomials. Instead, every single equation is converted. If it contains the monomial of second degree, then it is converted separately. For both the methods, number of clauses in CNF files will be the same, however there will be difference in their order and assignment of symbols. The algorithm for this method was presented in Table 5.7.

```
foreach (Sum in ANF)

       Array Components := Split Sum by '+' sign

       if Sum contains monomial of second degree then
              if exists symbols[monomial] then continue;
              else
              symbols[monomial] := variable;
              Array Components := Split monomial by '*' sign
              WriteClause(-variable Symbols[Components[0]] 0);
              WriteClause(-variable Symbols[Components[1]] 0);
              WriteClause(variable -Symbols[Components[0]] -Symbols[Components[1]] 0);
              variable := variable + 1;
              end if
       end if

       switch (Length of Components)
       case 0: WriteClause(-Symbols[Components[0]] 0);
                            break;

       case 1: WriteClause(-Symbols[Components[0]] Symbols[Components[1]] 0);
              WriteClause(Symbols[Components[0]] -Symbols[Components[1]] 0);
              break;

       case 2:
WriteClause(-Symbols[Components[0]] -Symbols[Components[1]] -Symbols[Components[2]] 0);
WriteClause(-Symbols[Components[0]] Symbols[Components[1]] Symbols[Components[2]] 0);
WriteClause(Symbols[Components[0]] -Symbols[Components[1]] Symbols[Components[2]] 0);
WriteClause(Symbols[Components[0]] Symbols[Components[1]] -Symbols[Components[2]] 0);
break;
       case 3:
WriteClause(-Symbols[Components[0]] -Symbols[Components[1]] -variable 0);
WriteClause(Symbols[Components[0]] Symbols[Components[1]] -variable 0);
WriteClause(Symbols[Components[0]] -Symbols[Components[1]] variable 0);
WriteClause(-Symbols[Components[0]] Symbols[Components[1]] variable 0);

WriteClause(-variable -Symbols[Components[2]] -Symbols[Components[3]] 0);
WriteClause(-variable Symbols[Components[2]] Symbols[Components[3]] 0);
WriteClause(variable -Symbols[Components[2]] Symbols[Components[3]] 0);
WriteClause(variable Symbols[Components[2]] -Symbols[Components[3]] 0);
```

```
variable := variable + 1;
break;
        end switch
end foreach
```

Table 5.7 Pseudo code for simple conversion

The algorithm above describes in details way of conversion from sum to logical conjunction. This form is based on the fact that XOR operation for *n* arguments is equal to conjunction of corresponding sums of particular arguments, which was described in details earlier in this dissertation.

By operation *WriteClause* in the algorithm above we understand writing line for instance to the file. Moreover every CNF file, built for the purposes of the attacks performed, contains three extra lines at the beginning:

p cnf number_of_clauses number_of_variables
1 0
-2 0

The first line is mandatory for every file, used as an entry to MiniSAT program. The two other lines are provided to mark one and zero respectively.

5.9 Implementations

This section describes all the supporting implementations which were made for purpose of research described in this dissertation. Summary of the implementations is written in Table 5.8.

Language	Name	Short description
C#	DES Implementation	Implementation of DES cipher
C#	SMS4 Implementation	Implementation of SMS4 cipher
C#	CNF Converter	Application to create CNF files
C	SMS4 Data Generator	Code to generate data for attack on SMS4 cipher
C#	SMS4 Algebraic Attack	Code to prepare algebraic description of SMS4 cipher
C#	Log Parser	Application to parse logs from attacks performed
PHP	Attack Performer	Code to run set of CNF files with MiniSat

Table 5.8 Summary of the implementations made

Implementation of DES cipher

This is the implementation of DES cipher, written in C# programming language. It was made as a Console Application. Code allowed to perform part of the research in automatic way, namely:

- Looking for proper data satisfying either linear approximations or particular differential characteristics
- Creating of proper files with entry data used then to prepare CNF files and perform attacks

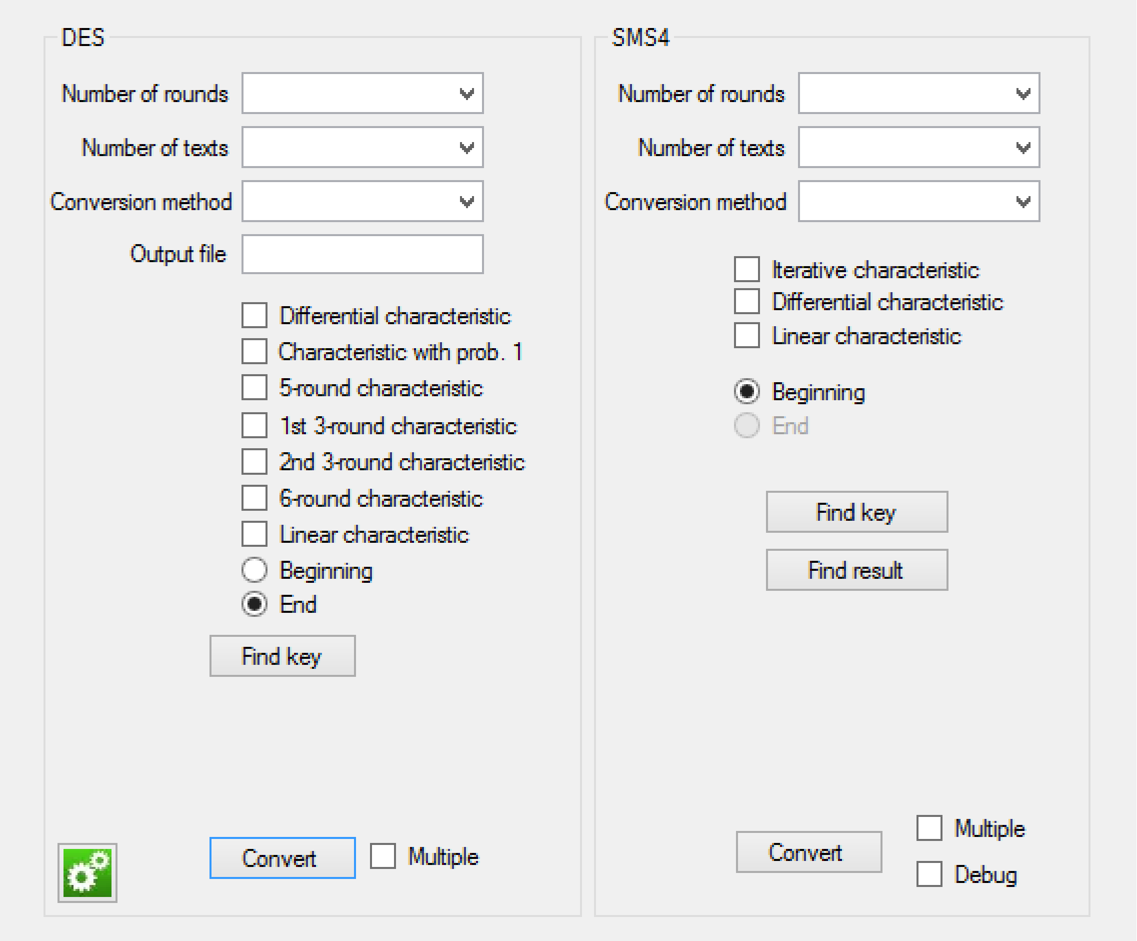
Implementation of SMS4 cipher

This implementation was also a Console Application written in C# programming language. Main purpose of creating it was to verify, whether algebraic attack on SMS4 is prepared properly. The implementation allowed to check values of data at every step of encipher procedure and compare it with values calculated by algebraic attack. Moreover, in further steps implementation allowed to:

- Generate round keys for main keys given
- Verify whether data found to perform attack indeed satisfies conditions assumed
- Prepare entry data files from data found by C implementation, used then to prepare CNF files

CNF Converter

This program was the most complex implementation made to facilitate research. It is a C# Windows Form Application, where main window is presented in Figure 5.12. Main functionality of this application was to create CNF files using files with data and files describing ciphers as sets of equations.



DES

Number of rounds

Number of texts

Conversion method

Output file

□ Differential characteristic
□ Characteristic with prob. 1
□ 5-round characteristic
□ 1st 3-round characteristic
□ 2nd 3-round characteristic
□ 6-round characteristic
□ Linear characteristic
○ Beginning
● End

□ Multiple

SMS4

Number of rounds

Number of texts

Conversion method

□ Iterative characteristic
□ Differential characteristic
□ Linear characteristic
● Beginning
○ End

□ Multiple
□ Debug

Figure 5.12 Screenshot of CNF Converter application

Application allowed to:

- Choose number of rounds
- Choose method of conversion
- Choose number of pairs used to perform attack
- Enhance sets of equations by selected additional equations
- Prepare single CNF files
- Prepare many CNF files at once in dynamically created paths
- Verifying results given (for debug purposes)

The algorithms implemented were originally created for purpose of attacks on DES cipher, however their reusability allowed quite easily to use them to build CNF files for SMS4 cipher. What more, this implementation can be extended for every cipher, which will be described according to the guidelines used for DES and SMS4.

SMS4 Data Generator

This was a set of two independent files, written in C language. Both implementations were compiled and run under GCC environment by Windows Command Console. First file, sms4_gen.c, implemented the algorithm to find data satisfying five round iterative differential characteristic for SMS4 cipher. Second file, sms4_gen_2nd_diff.c implemented similar algorithm, but for non-iterative five round differential characteristic.

SMS4 Algebraic Attack

This is set of codes implemented to prepare all elements needed to perform algebraic attack on SMS4 cipher.

Class Parsuj – class which contained methods for parse single polynomial equations describing SMS4 substitute box into a set of equations containing only sums of monomials of first and second degree. Moreover, every equation contains at most four addends. That is why new, extra variables were introduced and the task of the functions implemented was to add them to equations and to build extra equations where all these variables were summed up.

Class NoweRundy – algebraic description of SMS4 cipher was made and verified only for the first round. Description of following rounds was made by parsing 1st round description and replacing variables names with modified ones. Method from this class allowed to perform this operation automatically, where the only variable for the method was number of round. Based on this, proper replacement for every line was determined.

Class KolejnaPara – method implemented in this class allowed to prepare algebraic description for following pairs of plaintexts and ciphertexts. In general, variables names where replaced for every line, so when concatenating two or more pars in one attack, we could avoid situation, where two different variables are described with the same symbol.

Log Parser

This is a Windows Form Application written in C# programming language. It was used to scan all the result files given by MiniSAT and extract information about whether attack succeeded or not and what was the time of calculation if it had finished. Thus, there was no need to open every single log file and copy the result manually.

Attack Performer

This PHP implementation was run under Linux Ubuntu environment. For every CNF file put in path defined earlier, it was running MiniSAT program. For defined interval of time, code was checking whether implementation succeeded. In such case, next file was being run. After fixed amount of time, if there was no result, calculation was broken and the next file was taken to calculate. Thus, there was no need to run MiniSAT manually for every single CNF file.

```
Array Files := Get All files from path defined

foreach File in Files
            Run MiniSat for File
            for i:=0 to 60 by 10
               If Run Finished Then
                    Exit Loop
               Else
                    Wait 10 seconds
               End If
            end for

            If Not Run Finished Then
               Stop Current Run
            End If
end foreach
```

Table 5.9 Pseudo code for PHP implementation

Pseudo code for PHP shown in Table 5.9 above presents the idea of the algorithm. In this code the assumption was made, that after 600 seconds calculation for particular file is broken, if there is no results. Specific limits for each case are defined in the chapter describing results of the attacks.

6 Results of attacks on reduced round DES

This chapter covers results, which were obtained from combined attacks on reduced round DES.

6.1 Results of differential-algebraic attack

This subsection contains results of attacks, joining differential cryptanalysis and algebraic techniques, on DES cipher reduced to six and eight rounds. Attacks were performed using entry data described in previous chapter.

6.1.1 Attack on six-round DES

The subsection shows track record for attacks on DES cipher reduced to six rounds. There were three parts of attacks performed according to the entry data which were used.

Attack using data satisfying first three-round differential characteristic

The Table 6.1 summarizes attacks performed with data satisfying first three-round differential characteristic.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	192	144	75%	11,76	223,43
4	96	68	71%	7,99	229,38
6	48	34	71%	17,43	173,89
8	48	33	69%	12,11	193,43

Table 6.1 Results of attack on 6-round DES using first three-round differential

The first column is a number of chosen plaintexts which were used to construct CNF file. Using entry data specified a particular number of these files was generated and it is given in the second column. The third column contains the number of files, for which MiniSat successfully calculated result. There was a threshold of 600 seconds assumed. Process of calculation was interrupted if it exceeded this limit. The fourth column shows a rate of success, namely ratio of successful solutions number to the number of entry files. Column with the best result presents a least number of seconds which were needed to calculate solution among all calculations performed. Last column shows average result in seconds. The average was counted only for files successfully calculated.

Figures below show process of time needed to get the solution when sorted increasingly. The results presented on them cover attack with two, four, six and eight chosen plaintexts respectively. Additionally, results were approximated by adequate functions. Formulas used to calculate proper approximations were taken from (Antkiewicz, 2011).

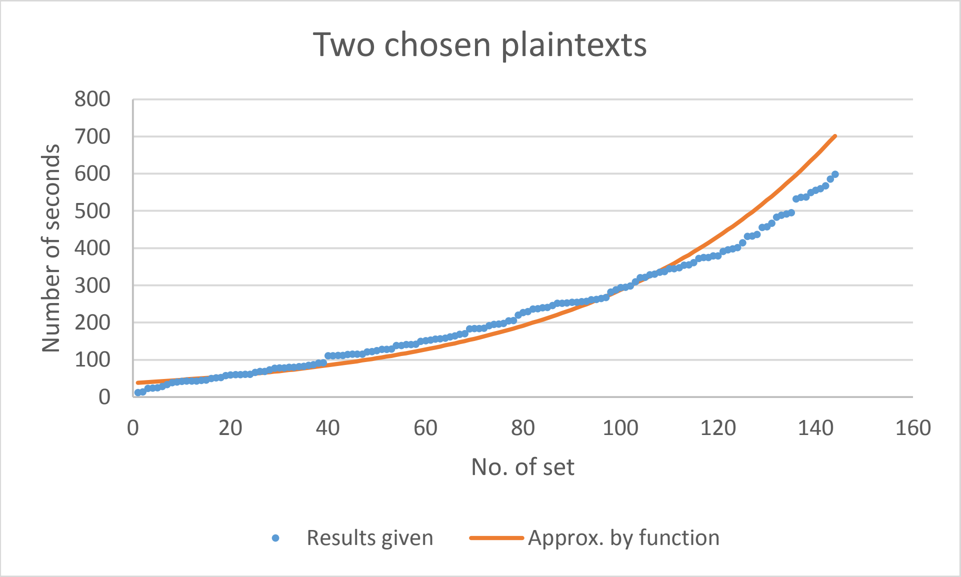


Figure 6.1

Results presented in Figure 6.1 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 37{,}80$, $\beta = 1{,}02$ and $x = \overline{1, 144}$. Estimator of variance for this approximation was equal to $S^2 = 3{,}04$.

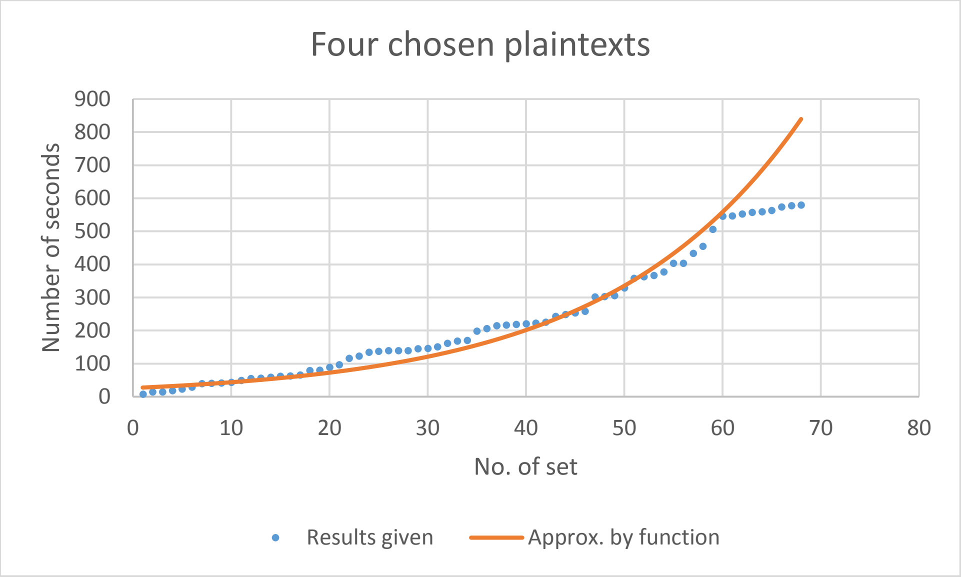


Figure 6.2

Results presented in Figure 6.2 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 26{,}19$, $\beta = 1{,}05$ and $x = \overline{1,68}$. Estimator of variance for this approximation was equal to $S^2 = 7{,}45$.

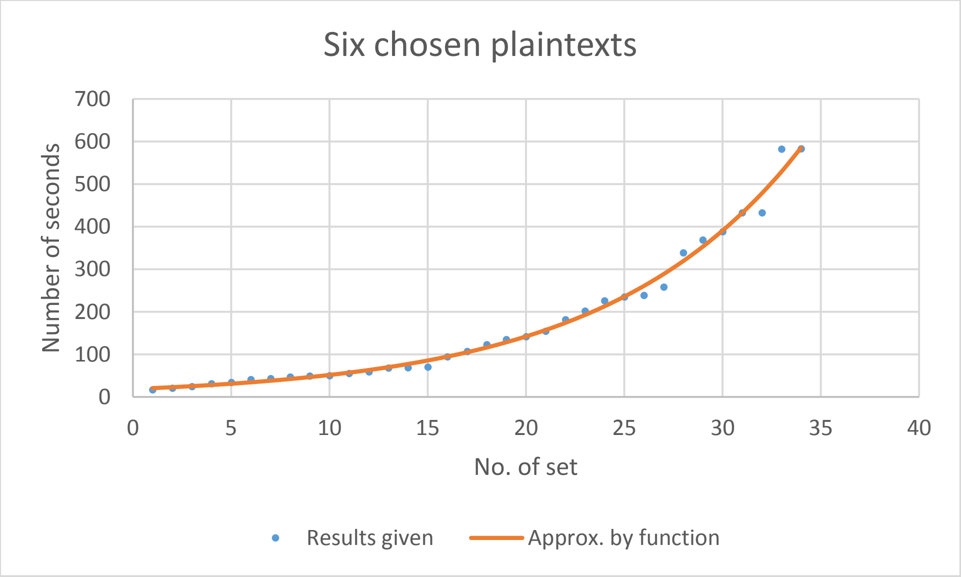


Figure 6.3

Results presented in Figure 6.3 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 18{,}86$, $\beta = 1{,}10$ and $x = \overline{1,34}$. Estimator of variance for this approximation was equal to $S^2 = 2{,}76$.

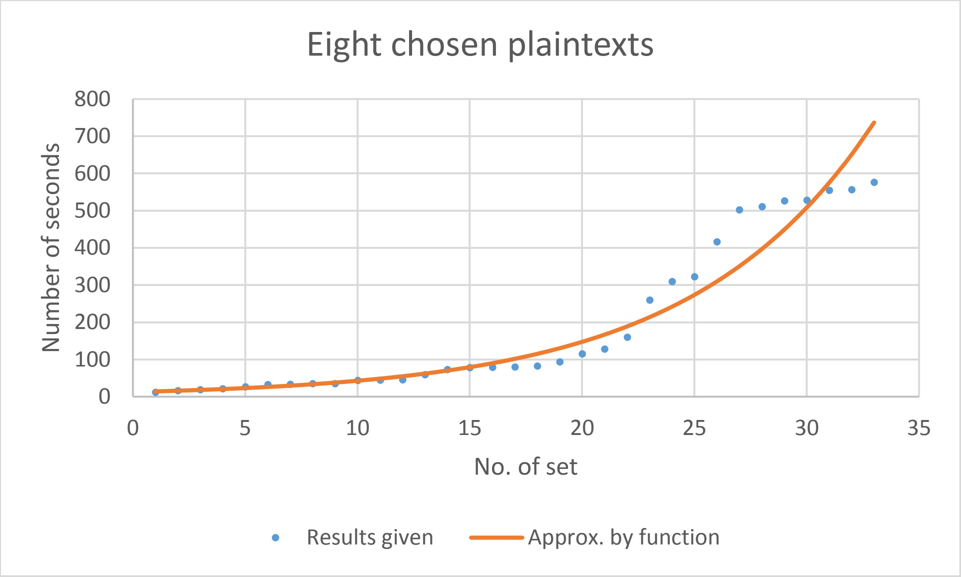


Figure 6.4

Results presented in Figure 6.4 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 12{,}33$, $\beta = 1{,}13$ and $x = \overline{1, 33}$. Estimator of variance for this approximation was equal to $S^{2} = 10{,}44$.

In each graph presented above, X-axis covers ordering of solutions while Y-axis is the time needed to get solution.

Furthermore, data complexity of attacks needs to be mentioned. It depends on two factors, namely number of chosen plaintexts and probability of differential characteristic the data satisfies. Table 6.2 summarizes the complexity of attacks, results of which were given in Table 6.1.

Number of plaintexts	Data complexity of attack
2	2^{4}
4	2^{5}
6	2^{6}
8	2^{7}

Table 6.2 Data complexity of attacks with first three-round differential characteristic

Attack using data satisfying second three-round differential characteristic

The Table 6.3 summarizes attacks performed with data satisfying second three-round differential characteristic. Columns of the table describe the result of the attack in the same way as it is in the Table 6.1.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	192	154	80%	9,42	189,59
4	96	67	70%	9,38	186,50
6	48	39	81%	5,46	210,92
8	48	40	83%	30,87	182,72

Table 6.3 Results of attack on 6-round DES using first three-round differential

Figures below show process of time needed to get the solution when sorted increasingly. The results presented on them cover attack with two, four, six and eight chosen plaintexts respectively.

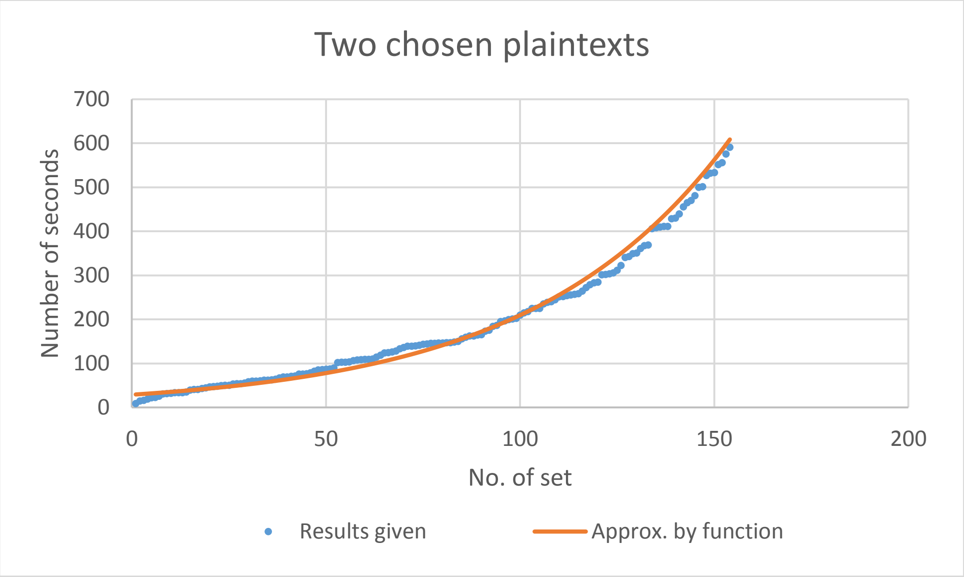


Figure 6.5

Results presented in Figure 6.5 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 29{,}22$, $\beta = 1{,}02$ and $x = \overline{1,154}$. Estimator of variance for this approximation was equal to $S^2 = 1{,}21$.

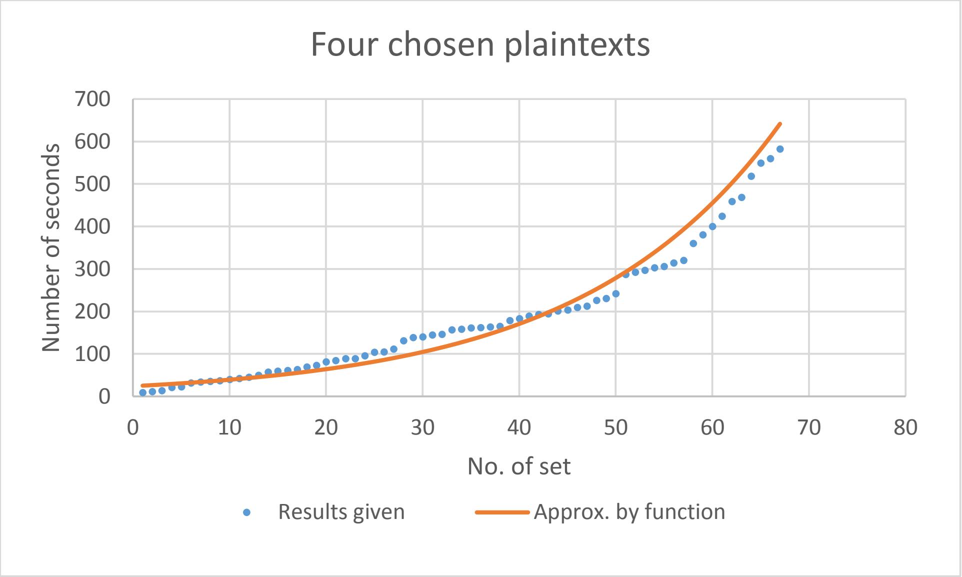


Figure 6.6

Results presented in Figure 6.6 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 23{,}98$, $\beta = 1{,}05$ and $x = \overline{1,67}$. Estimator of variance for this approximation was equal to $S^2 = 3{,}71$.

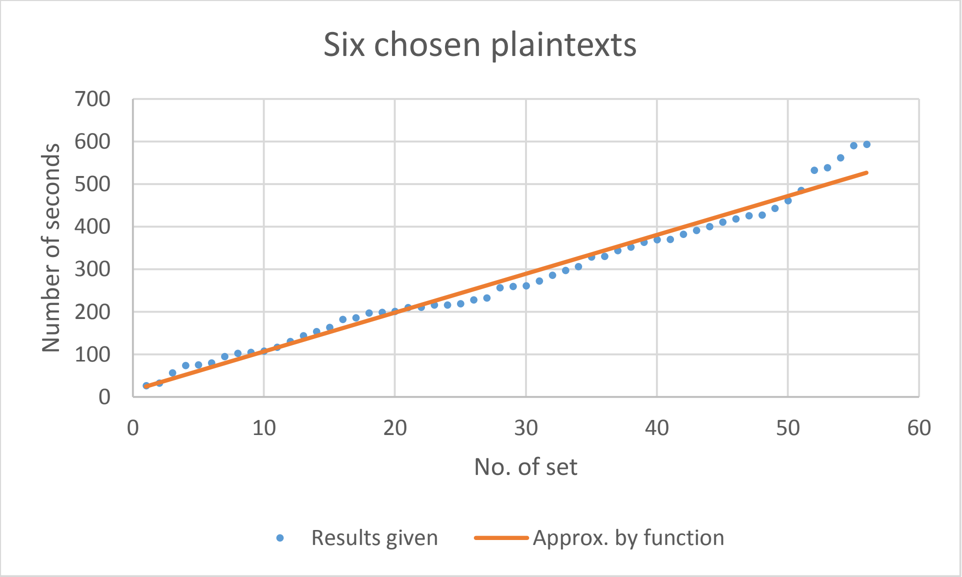


Figure 6.7

Results presented in Figure 6.7 can be approximated with function: $f(x) = \alpha x^{\beta}$. Values of parameters are equal to $\alpha = 3{,}29$, $\beta = 1{,}35$ and $x = \overline{1,39}$. Estimator of variance for this approximation was equal to $S^2 = 5{,}29$.

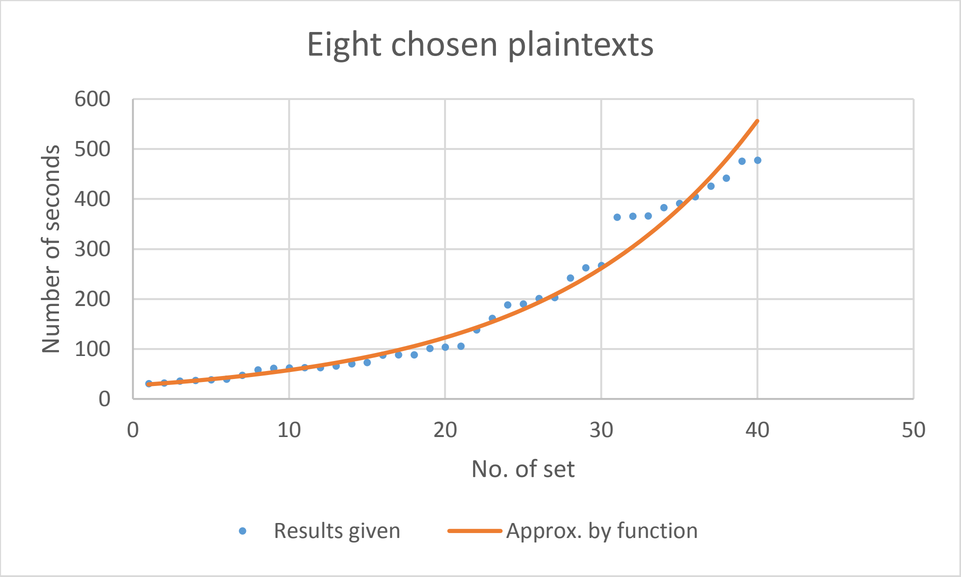


Figure 6.8

Results presented in Figure 6.8 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 27{,}1$, $\beta = 1{,}08$ and $x = \overline{1,40}$. Estimator of variance for this approximation was equal to $S^{2} = 4{,}25$.

In each graph presented above, X-axis covers ordering of solutions while Y-axis is the time needed to get solution.

The data complexity of this attack is the same as the previous one, while this attack uses characteristic which exactly the same probability.

Attack using data satisfying six-round differential characteristic

The Table 6.4 summarizes attacks performed with data satisfying six-round differential characteristic.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	192	56	29%	26,99	275,88
4	96	40	42%	3,70	169,82
6	48	35	73%	4,76	162,37
8	48	44	92%	2,38	15,03

Table 6.4 Results of attack on 6-round DES using six-round differential

Columns of the table describe the result of the attack in the same way as it's in the Table 6.1.

Figures below show process of time needed to get the solution when sorted increasingly. The results presented on them cover attack with two, four, six and eight chosen plaintexts respectively.

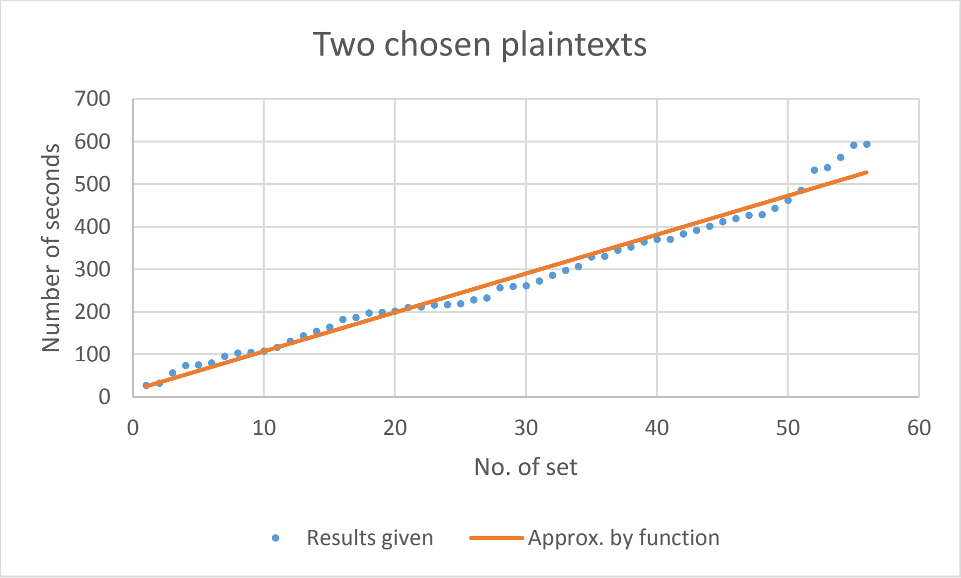


Figure 6.9

Results presented in Figure 6.9 can be approximated with function: $f(x) = \alpha + \beta x$. Values of parameters are equal to $\alpha = 15{,}35$, $\beta = 9{,}14$ and $x = \overline{1,56}$. Estimator of variance for this approximation was equal to $S^2 = 3{,}14$.

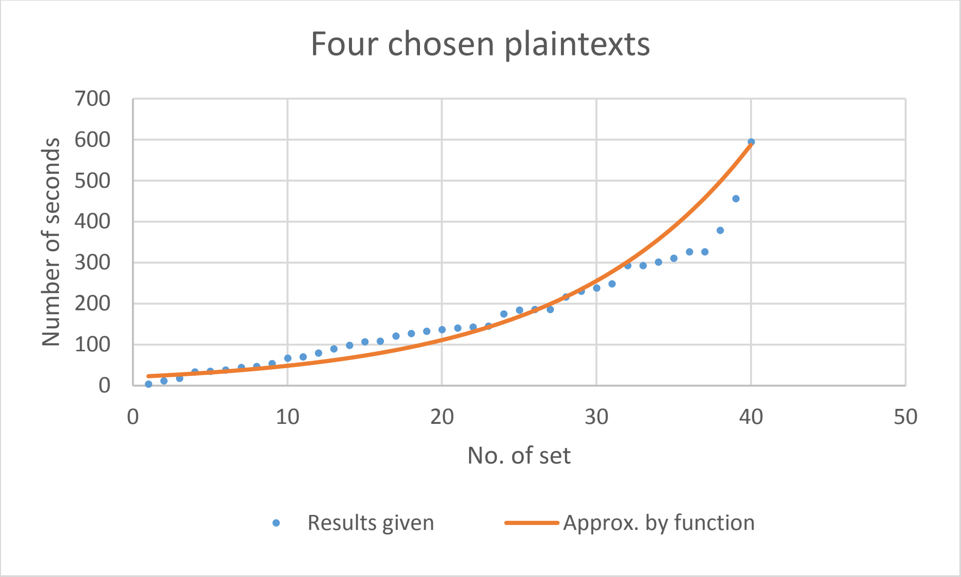


Figure 6.10

Results presented in Figure 6.10 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 20{,}94$, $\beta = 1{,}09$ and $x = \overline{1,40}$. Estimator of variance for this approximation was equal to $S^2 = 6{,}93$.

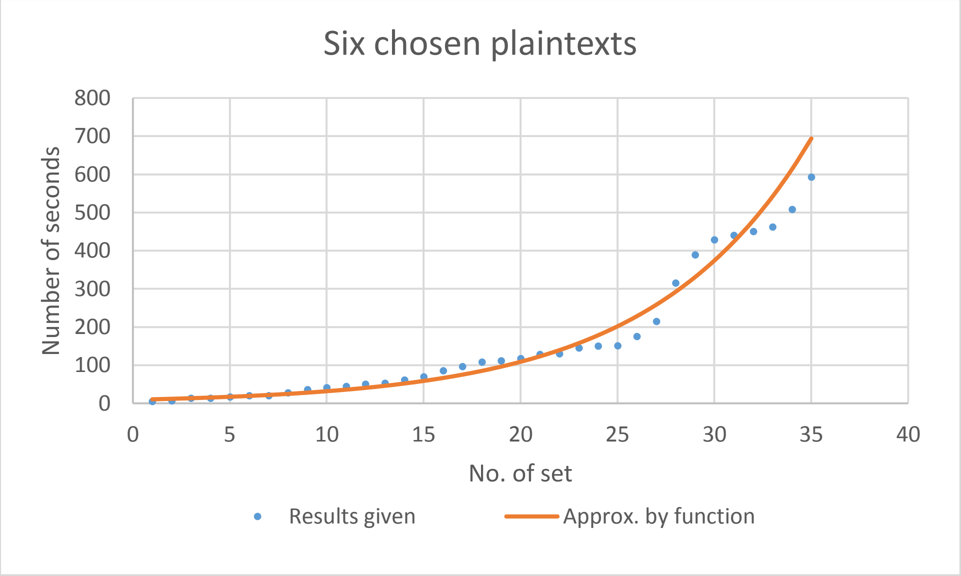


Figure 6.11

Results presented in Figure 6.11 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 9{,}19$, $\beta = 1{,}13$ and $x = \overline{1,35}$. Estimator of variance for this approximation was equal to $S^2 = 6{,}52$.

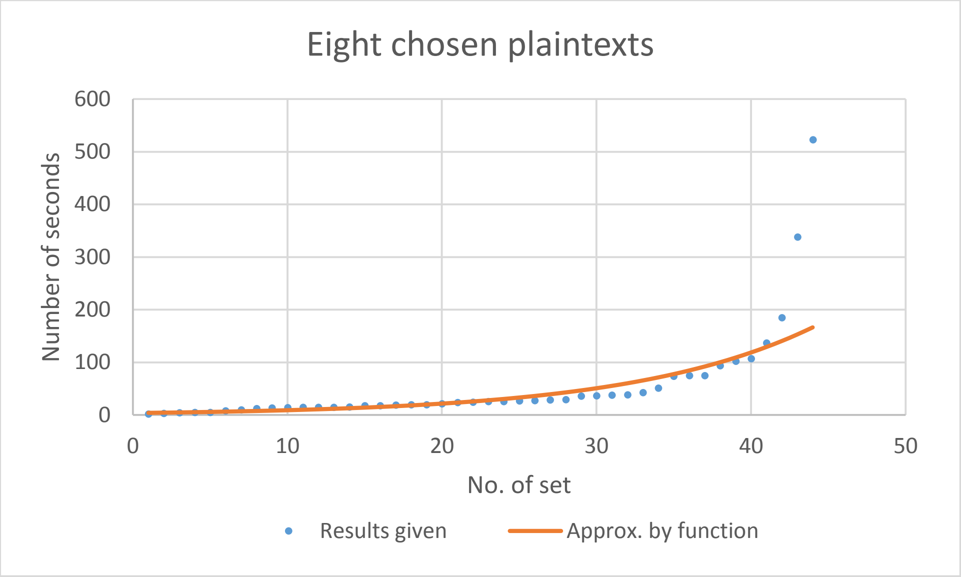


Figure 6.12

Results presented in Figure 6.12 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 4{,}06$, $\beta = 1{,}09$ and $x = \overline{1,44}$. Estimator of variance for this approximation was equal to $S^2 = 9{,}72$.

In each graph presented above, X-axis covers ordering of solutions while Y-axis is the time needed to get solution.

Data complexity of this attack differs from previous ones. The reason is that the differential characteristic, which needed to be satisfied by entry data, has probability around 1/10000. This leads to data complexity described in Table 6.5.

Number of plaintexts	Data complexity of attack
2	$\approx 2^{14}$
4	$\approx 2^{15}$
6	$\approx 2^{16}$
8	$\approx 2^{17}$

Table 6.5 Data complexity of attacks with six-round differential characteristic

6.1.2 Attack on eight-round DES

This subsection shows track record for attacks on DES cipher reduced to eight rounds. There were two parts of attacks performed with data satisfying either five-round or six-round differential characteristic.

Attack using data satisfying five-round differential characteristic

The Table 6.6 summarizes attacks performed with data satisfying five-round differential characteristic.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	192	6	3%	95,60	388,23
4	96	6	6%	14,99	225,96
6	48	4	8%	25,04	270,64
8	48	3	6%	51,81	206,57

Table 6.6 Results of attack on 8-round DES using five-round differential

Columns of this table are the same as in tables describing results of attacks for DES reduced to six rounds. Worth to remember is that average result is calculated only for successful results. Threshold assumed was also set to 600 seconds and calculations were interrupted after exceeding this limit. Due to low level of success rate, no charts were prepared to present results as a graphical visualization.

Probability of characteristic used to perform this attack is about 1/10000, which gives data complexity at level of 2^{14} for two chosen plaintexts and it rises in proportion to number of plaintexts used.

Attack using data satisfying six-round differential characteristic

The Table 6.7 summarizes attacks performed with data satisfying six-round differential characteristic.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	192	186	97%	0,85	17,03
4	96	96	100%	0,91	6,99
6	48	48	100%	0,67	5,02
8	48	48	100%	0,96	5,81

Table 6.7 Results of attack on 8-round DES using six-round differential

Columns of the table describe the result of the attack in the same way as it's in the Table 6.6. Probability of characteristic used to perform this attack is about 1/1000000, which gives data complexity at level of 2^{22} for two chosen plaintexts and it rises in proportion to number of plaintext used. For attack with eight plaintexts it is about 2^{25}.

Figures below show process of time needed to get the solution when sorted increasingly. The results presented on them covered attack with two, four, six and eight chosen plaintexts respectively. Due to the nature of the results, there was no approximation made as it was done for attack on six rounds of DES.

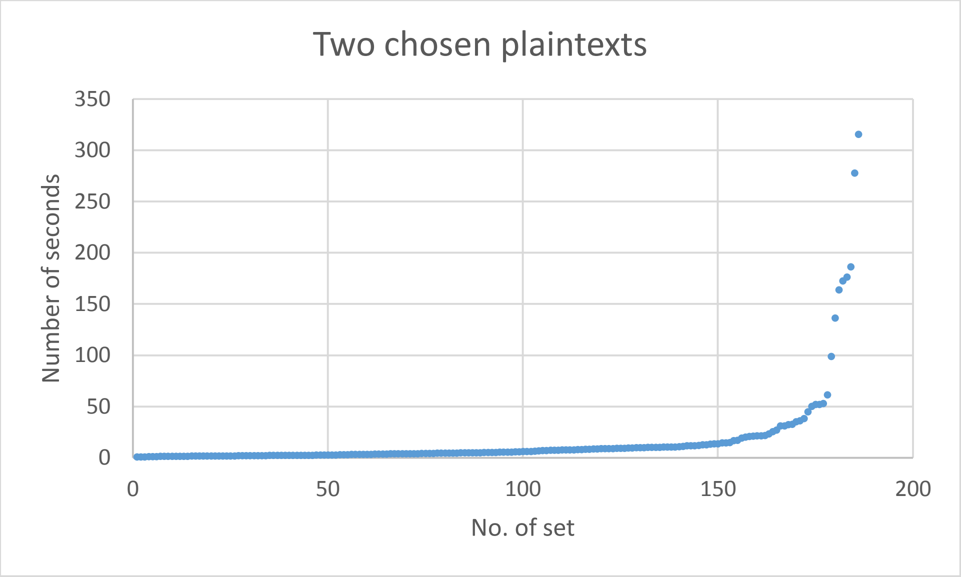


Figure 6.13

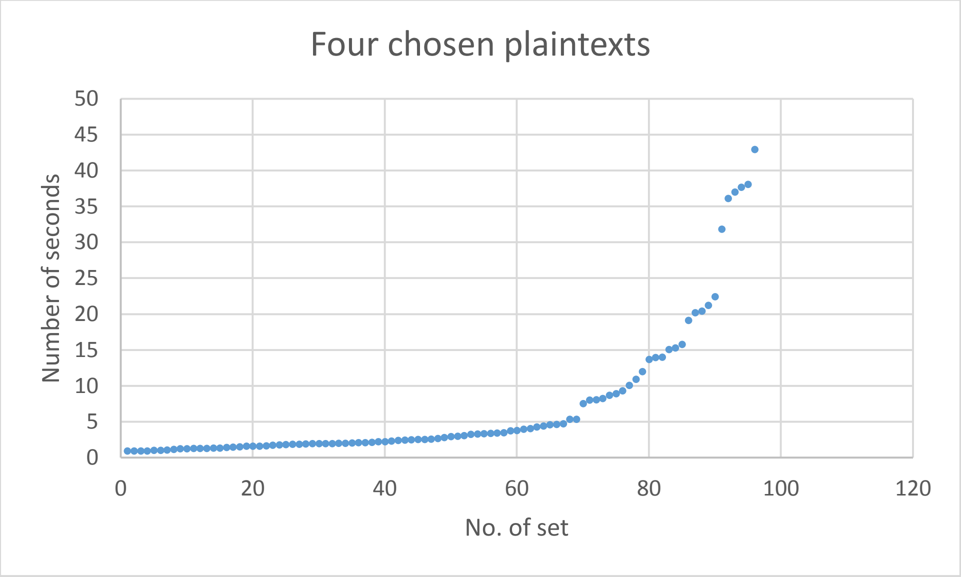


Figure 6.14

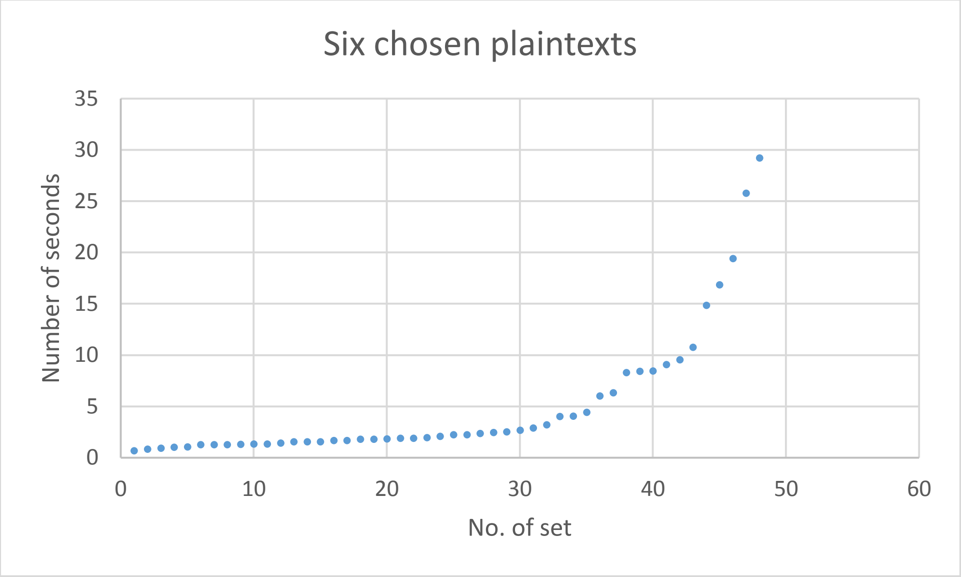


Figure 6.15

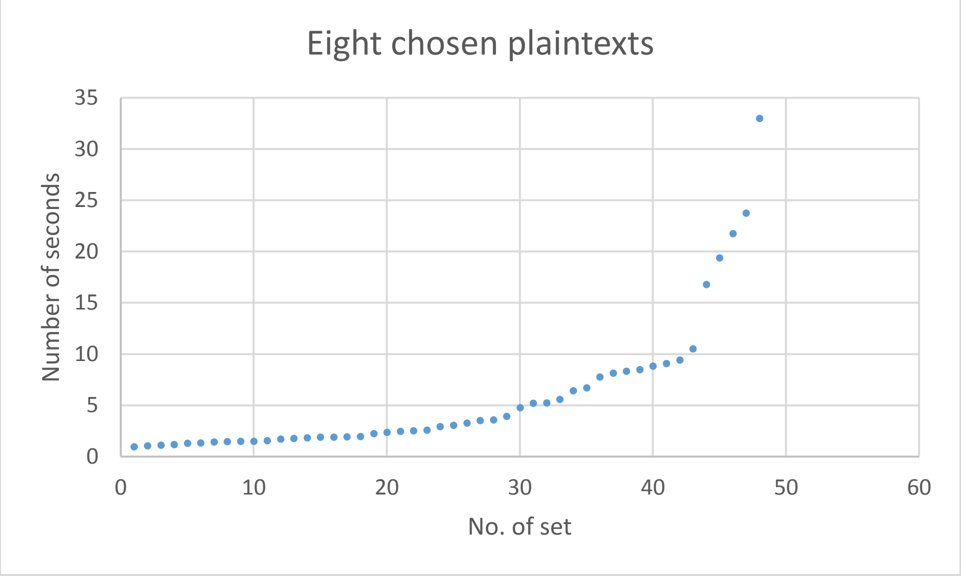


Figure 6.16

Similarly at it was previously, X-axis covers ordering of solutions while Y-axis is the time needed to get solution. Furthermore, for results of this attack no function was found which could approximate them relatively well (which was examined by the value of the estimator of variance).

6.2 Results of linear-algebraic attack

This subsection contains results of attacks, joining linear dependencies and algebraic techniques, on DES cipher reduced to six and eight rounds. Attacks were performed using entry data described in the previous chapter.

6.2.1 Attack on six-round DES

The subsection shows track record for attack on DES cipher reduced to six rounds.

The Table 6.8 summarizes attacks performed with data satisfying particular linear dependencies.

Number of CP	Number of CNF files generated	Number of successful results	Result gained [s]
1	96	1	814,18
2	48	1	324,08

Table 6.8 Results of attack on 6-round DES with application of linear dependencies

The first column gives the number of chosen plaintexts which were used to construct CNF file. Using entry data specified a particular number of these files was generated, which is shown in the second column. Third column contains the number of files, for which MiniSat successfully calculated result. There was a threshold of 3600 seconds assumed, which is 6 times larger than for differential-algebraic attack and was the reason of reducing number of entry sets used to perform the attack. Similarly, process of calculation was interrupted if it exceeded this limit. “Result gained” shows number of seconds which were needed to calculate solution in case it succeeded. Average result and success ratio are not presented while there was only one file in both cases, which led to successful result. Nevertheless, based on these results, we could say, that success ratio for this type of attack is at level of 1-2%.

7 Results of attacks on reduced round SMS4

This chapter covers results which were obtained from combined attacks on reduced round SMS4 cipher. For the test data, used to perform these attacks, also pure algebraic attack was performed. It however failed in all cases, which means, the calculation did not finish in certain amount of time, the same which was given for combined attacks.

7.1 Results of differential-algebraic attack

This subsection contains results of differential-algebraic attacks on SMS4 cipher reduced to five rounds. Attacks were performed using entry data described in previous chapter. There were two parts of attacks according to the two differential characteristics which were used.

Attack using data satisfying five-round iterative differential characteristic

The Table 7.1 summarizes attacks performed with data satisfying five-round iterative differential characteristic. CNF files were generated using so-called binomial conversion.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	40	25	63%	313,46	2103,07
4	20	8	40%	2174,04	4590,63

Table 7.1 Results of attack on 5-round SMS4 using iterative differential characteristic and binomial conversion

The Table 7.2 also summarizes attacks performed with data satisfying five-round iterative differential characteristic. In this case however CNF files were generated using simple conversion.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	40	40	100%	356,62	978,98
4	20	19	95%	1479,81	3582,95

Table 7.2 Results of attack on 5-round SMS4 using iterative differential characteristic and simple conversion

Both tables contain similar information, the only difference is, that results come from calculations performed on CNF files, obtained with different ANF to CNF conversion methods. The first column gives the number of chosen plaintexts which were used to construct CNF file. Using entry data specified a particular number of these files was generated, which is presented in the second column. The third column contains the number of files, for which MiniSat successfully calculated result. There was a threshold of 7200 seconds assumed. Process of calculation was interrupted if it exceeded this limit. The fourth column shows a rate of success, namely ratio of successful solutions number to the number of entry files. Column with the best result presents a least number of seconds which were needed to calculate solution among all calculations performed. Last column shows average result in seconds. The average was counted only for files successfully calculated.

Figures below show process of time needed to get the solution when sorted increasingly. The results presented on them cover attacks with two and four chosen plaintexts and binomial and simple conversion.

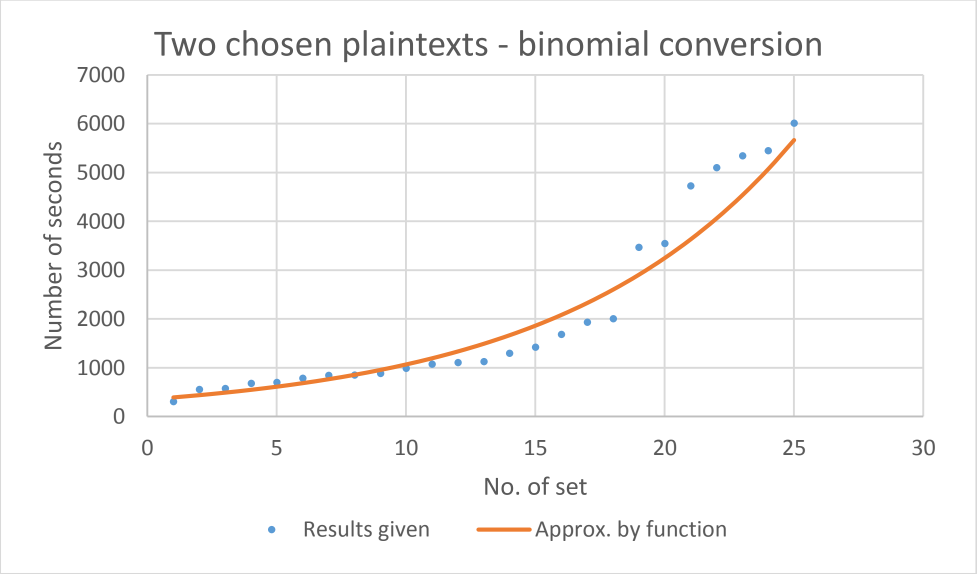


Figure 7.1

Results presented in Figure 7.1 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 351{,}02$, $\beta = 1{,}12$ and $x = \overline{1,25}$. Estimator of variance for this approximation was equal to $S^2 = 96{,}27$.

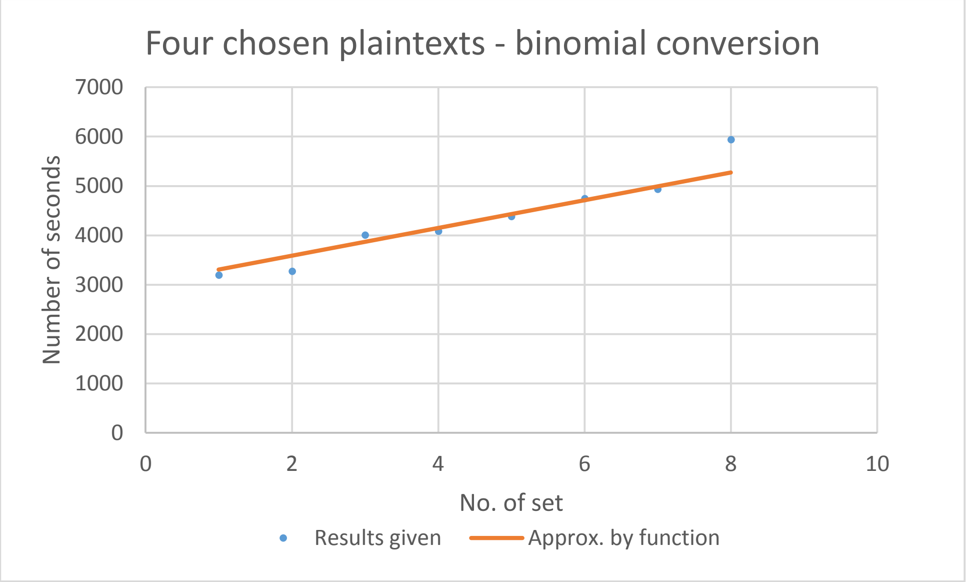


Figure 7.2

Results presented in Figure 7.2 can be approximated with function: $f(x) = \alpha + \beta x$. Values of parameters are equal to $\alpha = 1750{,}96$, $\beta = 631{,}04$ and $x = \overline{1,8}$. Estimator of variance for this approximation was equal to $S^2 = 128{,}49$.

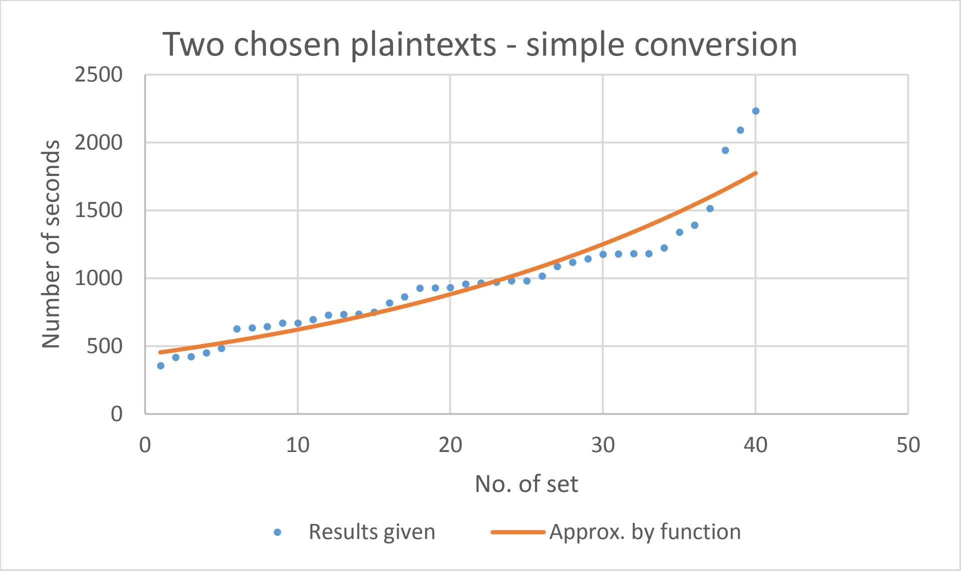


Figure 7.3

Results presented in Figure 7.3 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 438{,}76$, $\beta = 1{,}035$ and $x = \overline{1,40}$. Estimator of variance for this approximation was equal to $S^2 = 22{,}39$.

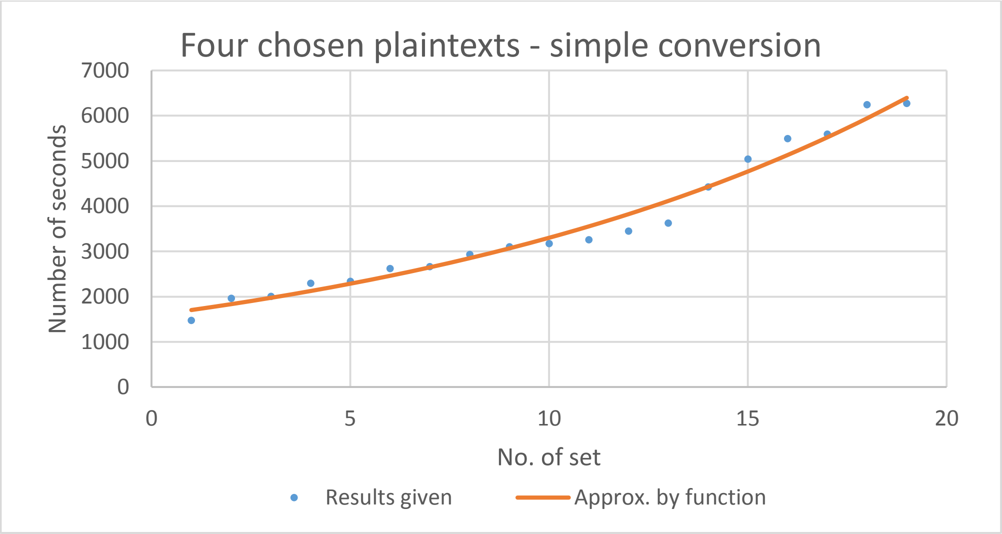


Figure 7.4

Results presented in Figure 7.4 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 1583{,}23$, $\beta = 1{,}076$ and $x = \overline{1,19}$. Estimator of variance for this approximation was equal to $S^2 = 25{,}49$.

In each graph presented above, X-axis covers ordering of solutions while Y-axis is the time needed to get solution.

Attack using data satisfying five-round non-iterative differential characteristic

The Table 7.3 summarizes attacks performed with data satisfying five-round non-iterative differential characteristic. Columns of the table describe the result of the attack in the same way as it is in the Table 7.1.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	40	24	60%	317,66	2141,75
4	20	14	70%	3191,23	5135,05

Table 7.3 Results of attack on 5-round SMS4 using non-iterative differential characteristic and binomial conversion

Results given in Table 7.3 were calculated for CNF files prepared with binomial conversion. Table 7.4 below gives results for CNF files obtained from simple conversion.

Number of CP	Number of CNF files generated	Number of successful results	Success rate	Best result [s]	Average result [s]
2	40	39	98%	296,79	1142,90
4	20	17	85%	2814,75	5212,99

Table 7.4 Results of attack on 5-round SMS4 using non-iterative differential characteristic and simple conversion

Similarly as it was done for previous attacks, below figures are presented, which show process of time needed to get the solution when sorted increasingly. The results, presented on them, covered attacks with two and four chosen plaintexts and binomial and simple conversion.

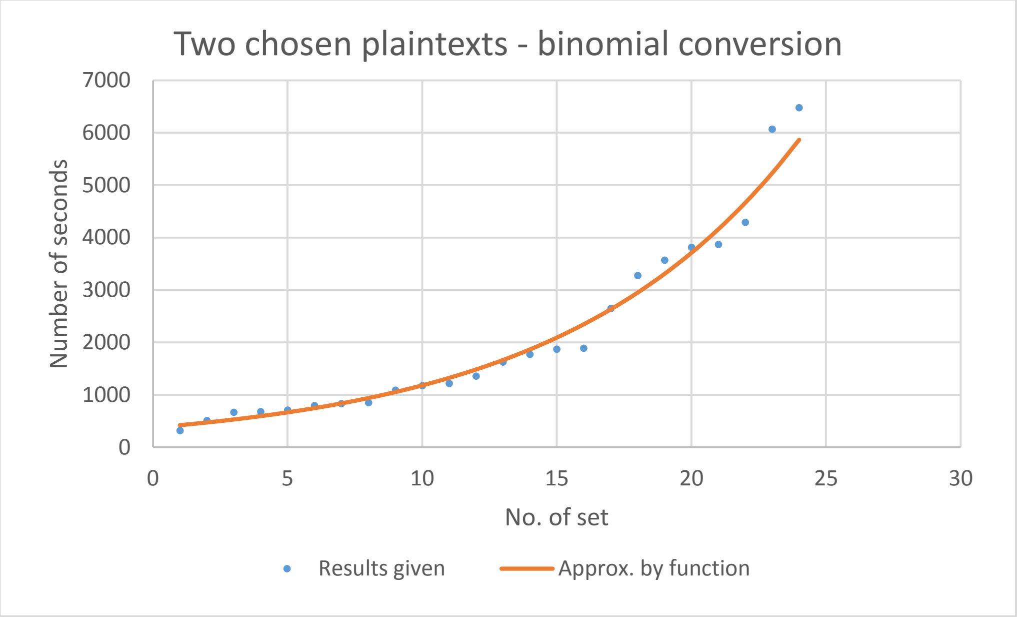


Figure 7.5

Results presented in Figure 7.5 can be approximated with function: $f(x) = \alpha\beta^{x}$. Values of parameters are equal to $\alpha = 374{,}61$, $\beta = 1{,}12$ and $x = \overline{1,24}$. Estimator of variance for this approximation was equal to $S^2 = 61{,}62$.

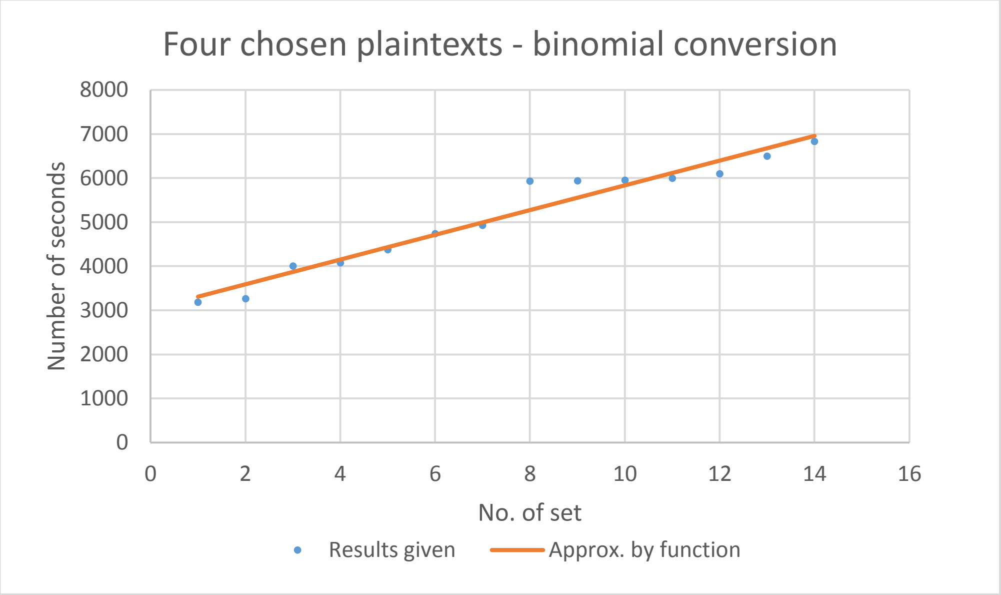


Figure 7.6

Results presented in Figure 7.6 can be approximated with function: $f(x) = \alpha + \beta x$. Values of parameters are equal to $\alpha = 3028{,}98$, $\beta = 280{,}81$ and $x = \overline{1,15}$. Estimator of variance for this approximation was equal to $S^2 = 78{,}81$.

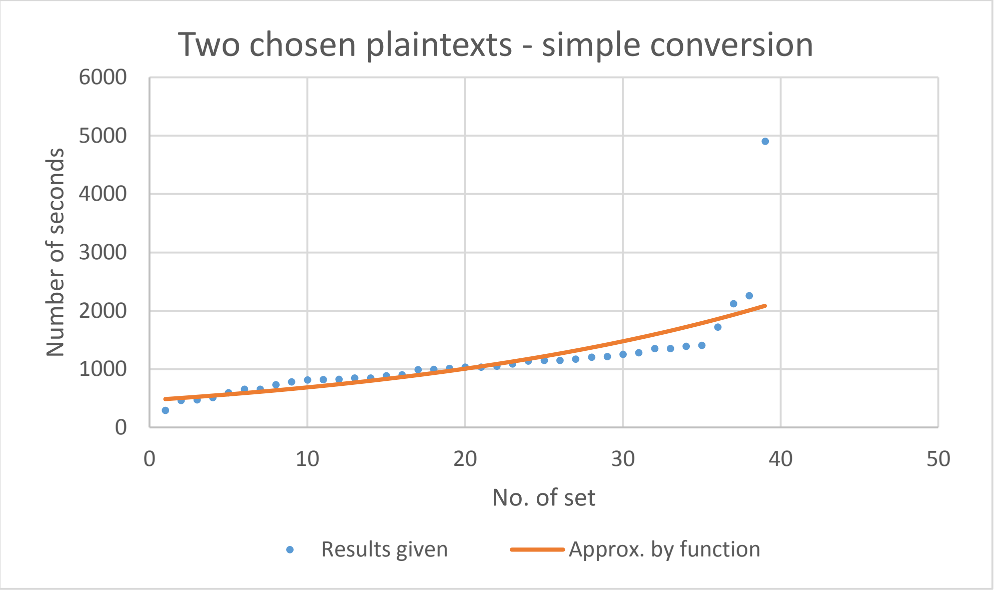


Figure 7.7

Results presented in Figure 7.7 can be approximated with function: $f(x) = \alpha\beta^x$. Values of parameters are equal to $\alpha = 469{,}29$, $\beta = 1{,}04$ and $x = \overline{1,39}$. Estimator of variance for this approximation was equal to $S^2 = 80{,}39$.

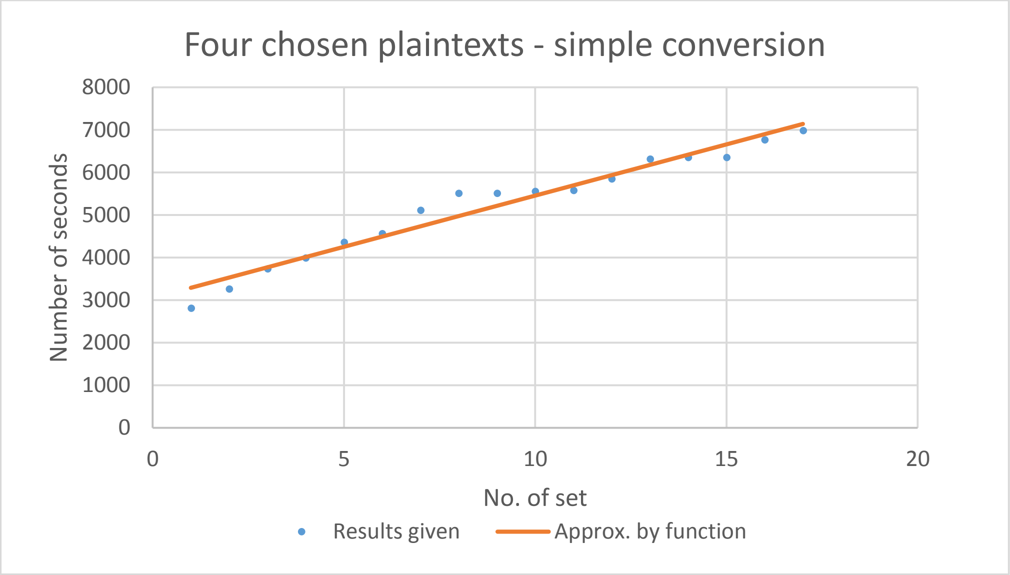


Figure 7.8

Results presented in Figure 7.8 can be approximated with function: $f(x) = \alpha + \beta x$. Values of parameters are equal to $\alpha = 3048{,}83$, $\beta = 240{,}46$ and $x = \overline{1,17}$. Estimator of variance for this approximation was equal to $S^2 = 67{,}55$.

For both attacks using different characteristics data complexity is the same and in scenario with two chosen plaintexts is equal to 2^{42} while for four chosen plaintexts it increases to 2^{43}.

7.2 Results of linear-algebraic attack

This subsection contains results of attacks, joining algebraic techniques with linear dependencies, on SMS4 cipher reduced to five rounds. Attacks were performed using entry data described in previous chapter.

The Table 6.8 summarizes attacks performed both without any additional dependencies and with data satisfying particular linear dependencies.

Number of CP	Additional equations	Number of CNF files generated	Number of successful results
1	No	72	0
1	Yes	72	0

Table 7.5 Results of attack on 5-round SMS4 with application of linear dependencies

The first column contains the number of chosen plaintexts which were used to construct CNF file. The second column indicates whether the attack was performed with additional equations given from linear dependencies or it was pure algebraic attack. Using entry data specified a particular number of these files was generated, which is given in the third column. What is worth to notice, that all CNF files were generated using both binomial and simple conversion. While no impact was observed, thus this information is not given separately. The fourth column contains the number of files, for which MiniSat successfully calculated result. There was a threshold of 7200 seconds assumed, which is 2 times larger than for linear-algebraic attack on reduced DES cipher and was the reason of reducing number of entry sets used to implement attack. Process of calculation was interrupted if it exceeded this limit. As it is easy to observe, in both cases there was no successful result given.

8 Summation

This chapter summarizes the results obtained and presented in previous chapters. It also contains analysis of the attacks performed.

8.1 Differential-algebraic attack on DES

Results of combined differential and algebraic attack on reduced round DES can be analysed from different points of view. However, they will be presented separately for DES reduced to six and eight rounds.

6 rounds

The most important conclusion is that combined attack on reduced round DES gives better results than attacks applied separately. It can be successfully performed with only 16 chosen plaintexts, while differential cryptanalysis needs 180 chosen plaintexts. Moreover it does not need any knowledge about key bits, which is obligatory for pure algebraic attacks.

Results of the attacks show that increasing number of plaintext used to perform does not have impact on the results, when using data satisfying three-round differential characteristics. The impact can be only noticed for the third of attacks.

In all the cases, time needed to find solutions was in the best case less than 30 seconds. While security of the cipher has to consider the worst-case scenario, then these results show clearly that time complexity of attack on six round of DES is significantly small. Similarly, data complexity at a level of 2^4 chosen plaintexts proves, that six rounds of DES cannot be considered secure.

When trying to choose the best result among all three attacks, it has to be analysed from different points of view. Time complexity is the best for attack with 8 chosen plaintexts satisfying six-round differential characteristic. Both best time and average time are the lowest among all results obtained. However, this attack is at once the one with the worst data complexity at level of 2^{14} chosen plaintexts.

Moreover, success rate and average time for this attack lowers when reducing number of data used to perform it.

Looking overall, the best results are given for attack with second three-round differential characteristic. While data complexity is the same as for the first one at a level between 2^4 and 2^7 CP, it has a better success rate and average time. Interesting fact is that for this attack it is harder to find influence between results and entry data, which can be easily observed for the first attack.

In case of the attack with first three-round differential characteristic, results show that there is relation between amount of entry data used, success rate and average time. More plaintext used makes success rate lower, but can also decrease average time.

8 rounds

The main conclusion, when looking at the results of attacks performed, is that eight round of DES can be broken without knowledge about any bits of the key. Moreover, the data complexity of this attack is better than pure differential cryptanalysis.

Attack applying five-round differential cryptanalysis shows, that it is possible to break eight round of DES with 2^{14} chosen plaintexts. However, results obtained are not satisfactory as success rate is very low. It is also possible to observe impact of amount of data used to the time needed to find the solution. Likewise, there is such impact to success rate. However it is hard to find direct correlation between these factors.

More interesting is, that modifying attack by using six-round differential characteristic changes results a lot. Although extra equations covered only one round more, when comparing to previous attack, both success rate and time have arisen significantly. In almost every case cipher was broken within seconds or even less. Results show also, that using more pairs of data, according to the development introduced in this dissertation, gives better effect, however on such level there is no need of improvement. The only one thing that makes it worse, is the data complexity, which is about 2^{22} chosen plaintexts. Nonetheless, it is still better than typical differential cryptanalysis and still there is no need to have knowledge about bits of the key.

Summarizing, all attacks performed confirm thesis that application of combined methods of cryptanalysis gives better results than using those techniques separately. Results obtained are satisfactory, however still do not close the path to enhance these attacks, but rather encourage to evolve such techniques.

8.2 Linear-algebraic attack on DES

Results obtained from application of newly introduced linear-algebraic attack clearly show that enhancement of pure algebraic attack with relations arising from linear cryptanalysis has no such positive impact on performance of this attack as it is for differential-algebraic attack.

Nevertheless, some successful results allow us to say that such attack is possible to apply with positive feedback. To perform it, about 2^{20} CP files were needed and it succeeded in 1% of the cases, so based on it we could say that overall complexity of this attack is at level of $2^{26.5}$ CP. It means, that there is still way to improvement, especially that such type of attack was never applied successfully before.

Modification of this certain type of attack to apply it to eight rounds would give us basic complexity at level of $2^{3.2\times 8} \approx 2^{26}$ CP. Assuming the same success ratio as it was for six rounds it would give us complexity exceeding 2^{30} CP. That is, why attack on DES reduced on eight rounds was not performed as results expected would be much worse than it is for attacks already performed.

One thing which has to be underlined as well is that single experiments were also executed for modified types of such combined attack. These experiments relied on removal of parts of equations, describing internal structure of S-boxes, and replacement with their linear approximations. This however did not succeed or led to sets of equations having plenty proper solutions (equation systems with parameters).

Nevertheless, research based on joining linear cryptanalysis with algebraic attack is still open while there are many variants which could be verified. These variants are, but not limited to:

- Replacement of S-boxes non-linear description with their different approximations occurred with particular probability
- Searching of maximum possible replacement as described above and keeping set of equations consistent
- Setting some bits of the key fixed when performing combined attack

This gives us plenty of possibilities to improve the attack which however were impossible to perform in researches made for this thesis due to their variety.

8.3 Differential-algebraic attack on SMS4

Results provided from the attack on reduced SMS4, joining both differential cryptanalysis and algebraic attack, strictly confirm thesis determined. While pure algebraic attack failed in all cases, combined method allowed to get the solution in relatively short time. Moreover, having only one pair satisfying characteristic was enough to perform the attack successfully. There are however some other interesting conclusions based on the results obtained.

There are three aspects which allow to examine the performance of the attack. The first one is the method of conversion. While for attacks on DES cipher, results were similar independently on how the equations were converted, here we are able to observe much better success rate when so-called simple conversion was used. This is the proof that way of conversion from ANF to CNF can make a difference on how much time is needed to get the solution. The second conclusion is, that using more pairs to perform attack does not improve it. Not only time needed to get the solution was in general longer, but also data complexity is slightly bigger while more right pairs are needed to perform such attack. The probable reason is that number of equations describing SMS4 is large, especially in collation to DES cipher, so increasing it cannot facilitate calculations made by SAT solver. The last thing is differential characteristic used in the attack. In this case it is hard to distinguish whether one of the characteristics had better influence on the performance than the other. Such conclusion was also expected, while both characteristics give the same data complexity and provide the same number of additional equations to the set.

There is one more interesting fact worth to notice. Similarly, as it was for results of attacks on DES, when we sort successful result by time ascending and put in the chart, we are given shape of function. In almost all cases it could be quite well approximated with exponential function (there are however some exceptions, especially for results related to SMS4). Such behaviour is probably the consequence of the fact, that we handle here with NP-hard problem, so all the algorithms allowing to solve it have non-polynomial complexity and the results just provide its graphical presentation.

8.4 Linear-algebraic attack on SMS4

Results obtained from application of linear-algebraic attack clearly show that enhancement of pure algebraic attack with relations arising from linear cryptanalysis has no impact at all on performance of this attack. Such results were however expected due to the fact, that additional equations given by linear approximations do not enhance original set of equations in a way it happens for attack on the DES cipher. Moreover, due to the algebraic structure of SMS4, these extra equations did not involve either bits of the round key or outputs from round function directly, as it was in DES case. The only enhancement was to improve relations between some of the input and output bits of substitution box.

Still there are other topics to consider, mentioned in subsection summarizing similar attacks on reduced-round DES cipher. Moreover, one more thing has to mentioned, namely looking for a contradiction. While for attack on DES cipher, time of solving was approximately the same independently on the fact whether pair used was satisfying linear relations or not, here finding contradiction happens within a second. This means, that such attack could be used as a filter for “right pairs” –

these satisfying all linear approximations given. This could be done by assuming, that if no solution is found for a certain amount of time, then such pair can be considered as a “right pair”. Otherwise, we should be given information, that set of equations is impossible to be solved, which gives clear statement that such pair does not satisfy all relations. Sample information, received from trial of solving improper pair, is given below.

```
==================================[MINISAT]===================================
| Conflicts |     ORIGINAL     |              LEARNT              | Progress |
|           | Clauses Literals |   Limit Clauses Literals  Lit/Cl |          |
==============================================================================
|         0 |  233654   593302 |   77884        0        0     nan |  0.000 % |
|       100 |  233664   593302 |   85672       90     1698    18.9 |  1.549 % |
|       250 |  233676   593302 |   94239      228     3058    13.4 |  1.549 % |
|       475 |  233681   593302 |  103663      447     6218    13.9 |  1.559 % |
|       814 |  233709   593302 |  114029      755    11798    15.6 |  1.676 % |
|      1320 |  233741   593302 |  125432     1228    20194    16.4 |  1.678 % |
==============================================================================
restarts              : 6
conflicts             : 1727           (2127 /sec)
decisions             : 4778           (5884 /sec)
propagations          : 1381682        (1701579 /sec)
conflict literals     : 28928          (41.57 % deleted)
Memory used           : 278.00 MB
CPU time              : 0.812 s

UNSATISFIABLE
```

It is easy to notice, that it took MiniSAT less than a second to define, that set of equations given contains a contradiction making it unable to solve. This information allows us to certify, that particular pair plaintext – ciphertext, used to construct this set, does not satisfy additional equations describing linear dependencies.

8.5 Conclusions

Results obtained from attacks on reduced round DES and SMS4 ciphers clearly indicate that thesis, saying that application of combined cryptanalysis techniques allows to raise effectiveness of attacks contrary to techniques used separately, is true. Not only such combined attacks provide better effects, like reduction of data complexity or time needed to get the solution, but in some cases are the best implemented attacks, as it was in the case of DES cipher, reduced to six and eight rounds respectively. Moreover, dependency was found between number of plaintexts and conversion type and time required to successful accomplishment of the attack.

References

1. (NIST, 1999) U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, *Data Encryption Standard (DES)*, FIPS PUB 46-3, Reaffirmed 1999
2. (SMS4, 2006) Beijing Data Security Technology Co. Ltd, *Specification of SMS4, Block Cipher for WLAN Products - SMS4 (in Chinese),* http://www.oscca.gov.cn/UpFile/200621016423197990.pdf.
3. (Diffie and Ledin, 2008) W. Diffie and G. Ledin (translators), *SMS4 Encryption Algorithm for Wireless Networks*, Cryptology ePrint Archive, report 2008/329, received 29 Jul 2008, http://eprint.iacr.org/.
4. (Liu et al., 2007) Liu, F., Ji, W., Hu, L., Ding, J., Lv, S., Pyshkin, A., Weinmann, R.P., *Analysis of the sms4 block cipher,* In Pieprzyk, J., Ghodosi, H., Dawson, E., eds.: ACISP. Volume 4586 of Lecture Notes in Computer Science., Springer (2007) 158–170
5. (Erickson et al., 2009) Erickson J., Ding J., Christensen Ch., *Algebraic cryptanalysis of SMS4: gröbner basis attack and SAT attack compared*, Proceedings of the 12th international conference on Information security and cryptology, 2009, Pages 73-86
6. (Biham and Shamir, 1990) Biham E., Shamir A., *Differential Cryptanalysis of DES-like Cryptosystems*, Advances in Cryptology - Proceedings of CRYPTO 1990, LNCS 537, pp. 2-21, Springer-Verlag, 1990.
7. (Matsui, 1993) Matsui M., *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology - Eurocrypt 1993, Lecture Notes in Computer Science, vol. 765, Springer–Verlag, pp. 386–397, 1993
8. (Matsui, 1994) Matsui M., *The First Experimental Cryptanalysis of the Data Encryption Standard*, Advances in Cryptology - CRYPTO ’94 (Lecture Notes in Computer Science no. 839), Springer-Verlag, pp. 1-11, 1994
9. (Liu et al., 2009) Liu Z., Gu D., Zhang J., *Multiple Linear Cryptanalysis of Reduced-Round SMS4 Block Cipher*, Cryptology ePrint Archive: Report 2009/256, 2009, http://eprint.iacr.org/.

10. (Cho and Nyberg, 2011) Cho J.Y., Nyberg K., *Improved Linear Cryptanalysis of SMS4 Block Cipher*, Proceedings of Symmetric Key Encryption Workshop, SKEW 2011
11. (Kim et al., 2008) Kim T., Kim J., Hong S., Sung J., *Linear and differential cryptanalysis of reduced SMS4 block cipher*, Cryptology ePrint Archive, Report 2008/281, 2008, http://eprint.iacr.org/.
12. (Etrog and Robshaw, 2008) Etrog J., Robshaw M., *The cryptanalysis of reduced-round sms4*, Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers, vol. 5381, 2008, pp. 51–65.
13. (Taehyun et al., 2008) Taehyun K., Jongsung K., Seokhie H., Jaechul S., *Linear and Differential Cryptanalysis of Reduced SMS4 Block Cipher*, IACR Cryptology ePrint Archive, Report 2008/281, http://eprint.iacr.org
14. (Zhang et al., 2008) Zhang L., Zhang W., Wu W., *Cryptanalysis of Reduced-Round SMS4 Block cipher*, Proceedings of ACISP'08, 2008.
15. (Su et al., 2010) Su B., Wu W., Zhang W., *Differential Cryptanalysis of SMS4 Block Cipher*, IACR Cryptology ePrint Archive, Report 2010/062, http://eprint.iacr.org
16. (Ji et al., 2007) Ji W., Hu L., *New description of sms4 by an embedding over GF(2^8)*, In Srinathan, K., Rangan, C.P., Yung, M., eds.: INDOCRYPT. Volume 4859 of Lecture Notes in Computer Science., Springer (2007) 238–251
17. (Ji et al., 2009) Ji W., Hu L., Ou H., *Algebraic attack to sms4 and the comparison with AES*, Information Assurance and Security, International Symposium on 1 (2009) 662–665
18. (Biham and Shamir, 1991) Biham E., Shamir A., *Differential Cryptanalysis of the Full 16-Round DES*, CS 708, Proceedings of CRYPTO '92, Volume 740 of Lecture Notes in Computer Science, December 1991
19. (Courtois and Bard, 2006) Courtois N.T., Bard G.V., *Algebraic Cryptanalysis of the Data Encryption Standard*, IACR Cryptology ePrint Archive, Report 2006/402

20. (Courtois, 2006) Courtois N.T., *Examples of equations generated for experiments with algebraic cryptanalysis of DES,* http://www.cryptosystem.net/aes/toyciphers.html
21. (Courtois et al., 2012) Courtois N.T., Gawinecki J.A., Song G., *Contradiction Immunity and Guess-Then-Determine Attacks On GOST,* Tatra Mountains Mathematic Publications, Vol. 53 no. 3 (2012), pp. 65-79
22. (Gasecki and Misztal, 2011) Gąsecki A., Misztal M., *Zastosowanie technik algebraicznych w kryptoanalizie różnicowej na przykładzie szyfru blokowego DES,* Biuletyn WAT, Vol. LX, Nr 3, 2011, pp. 379-390
23. (Faugere et al., 2009) Faugere J.-C., Perret L., Spaenlehauer P.-J., *Algebraic-Differential Cryptanalysis of DES*, Western European Workshop on Research in Cryptology - WEWoRC 2009, pp. 1–5 (2009)
24. (Knudsen, 1995) Knudsen L.R., *Truncated and Higher Order Differentials,* Fast Software Encryption (FSE 1995), Lecture Notes in Computer Science, vol. 1008, Springer–Verlag, pp. 196–211, 1995
25. (Z'aba et al., 2010) Z'aba M. R., Simpson L., Dawson E., Wong K., *Linearity within the SMS4 Block Cipher*, In: Lecture Notes in Computer Science: Information Security and Cryptology (ICISC 2009), 12-15 December, 2009, Beijing, China.
26. (Albrecht and Cid, 2008) Albrecht M., Cid C., *Algebraic Techniques in Differential Cryptanalysis,* IACR Cryptology ePrint Archive, Report 2008/177
27. (Wang et al., 2011) Wang M., Sun Y., Mouha N., Preneel B., *Algebraic Techniques in Differential Cryptanalysis Revisited*, Lecture Notes in Computer Science, Volume 6812, 2011, pp. 120-141
28. (Weinmann, 2009) Weinmann R.-P., *Algebraic Methods in Block Cipher Cryptanalysis*, Dissertation zur Erlangung des Grades Dr. rer. nat. (rerum naturalium), Darmstadt, 2009
29. (Lu, 2007) Lu J., *Attacking Reduced-Round Versions of the SMS4 Block Cipher in the Chinese WAPI Standard*, Proceedings of ICICS 2007, Lecture Notes in Computer Science 4861, pp. 306–318, Springer-Verlag, 2007

30. (Toz and Dunkelman, 2008), Toz D., Dunkelman O., *Analysis of two Attacks on Reduced-Round Versions of the SMS4*, Lecture Notes in Computer Science 5308, 2008, pp. 141-156
31. (Antkiewicz, 2011) Antkiewicz R., *Lectures "Elementy teorii prognozy" at Doctoral Studies in Computer Science*, Military Univeristy of Technology, 2011
32. (Courtois et al., 2007) Courtois N.T., Bard G.V., Jefferson J., *Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers*, IACR Cryptology ePrint Archive, Report 2007/024
33. (Een and Sorensson, 2005) Een N., Sorensson N., *MiniSat - A SAT Solver with Conflict-Clause Minimization*, Proc. Theory and Applications of Satisfiability Testing (SAT'05), 2005
34. (SAGE, 2008) The SAGE Group, *SAGE Mathematics Software (Version 3.3)*, 2008. Available at http://www.sagemath.org.
35. (Gasecki, 2013) Gąsecki A.: *Low data complexity differential-algebraic attack on reduced round DES*, Tatra Mt. Math. Publ. 57 (2013), 35–43