VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/v10127-012-0047-3
Tatra Mt. Math. Publ. **53** (2012), 201–212

ALGEBRAIC CRYPTANALYSIS OF PRESENT BASED ON THE METHOD OF SYLLOGISMS

Lucia Lacko-Bartošová

ABSTRACT. In this paper algebraic cryptanalysis of block cipher Present based on the method of syllogisms is presented. Different guessing strategies of the method of syllogisms were used and significantly different complexities of attack were reached.

1. Introduction

Cipher Present was first introduced on CHES '07 [3] and it belongs to the group of ciphers that were created for the need of lightweight cryptography. It is a relatively young scientific discipline that incorporates electrical engineering, cryptology, mathematics and informatics and focuses on new designs, adaptions or efficient implementations of cryptographic primitives and protocols. Lightweight cryptography is applied in limited devices with limited computation and communication resources, typically RFID tags and sensored networks. The applications of such systems are quite numerous: automated management of the supply chain, ticketing, telephone cards, micro-payments, access control, automatic tolls, public transportations, prevention of counterfeiting, pets tracking, airline luggage tracking, library management, etc., [4]. The word lightweight should not be associated with low security, but security of these ciphers should be studied.

In this paper, the algebraic cryptanalysis, as one aspect of the security of cipher Present was studied. It consists of two steps. In the first step the cipher has to be converted to the system of equations. In the second step, the equation system is solved and the secret key is obtained [2]. The second step can be performed by different methods; algorithms based on Groebner bases, SAT-solvers

2010 Mathematics Subject Classification: 94A60, 68P25.
Keywords: algebraic cryptanalysis, local reduction, method of syllogisms, Present.
Supported by Vega Grant no. 2/0206/10.

or algorithms based on local reduction can be used. Based on the solving technique, the representation of the system of equations is chosen. Further we focus on the algorithms based on local reduction, in the concrete on the method of syllogisms. Cipher has to be converted to the system of polynomial equations, usually over $GF(2)$ and then written in symbol representation.

The paper is organized as follows. In Section 2 the structure of the Present cipher and known cryptanalysis are described. In Section 3 the method of syllogisms is presented together with guessing strategies. In Section 4, goal of the experiments and experimental results are provided and paper is concluded in Section 5.

2. The Present cipher

2.1. Structure of the cipher

In this section Present cipher is described. It was first presented on CHES '07 (Workshop on Cryptographic Hardware and Embedded Systems) by the authors Bogdanov, Knudsen, Leander, Paar, Poschmann, Robshaw, Seurin and Vikkelsoe. It is SP-network that consists of 31 rounds. Block size is 64 bits and key sizes are 80 or 128 bits [3].

2.1.1. Round transformations

Every round consists of XOR operation, where to a current state, round key K_i, $1 \leq i \leq 32$ is added. In non-linear layer, 4-bit to 4-bit S-box is applied 16 times in parallel in each round. Linear layer consists of linear bitwise permutation. The cipher is described in pseudo-code in Algorithm 1.

Algorithm 1 Algorithmic description of Present [3]

```
generateRoundKeys()
for i = 1 to 31 do
    addRoundKey(STATE, K_i)
    sBoxLayer(STATE)
    pLayer(STATE)
end for
addRoundKey(STATE, K_32)
```

addRoundKey. Let us denote round key $K_i = \kappa^i_{63} \dots \kappa^i_0$, for $1 \leq i \leq 32$ and current state $b_{63} \dots b_0$. Then transformation `addRoundKey` is the operation

$$b_j \to b_j \oplus \kappa^i_j, \qquad 0 \leq j \leq 63.$$

sBoxLayer. Substitution table, S-box, is a mapping $S : \mathbb{F}_2^4 \to \mathbb{F}_2^4$. In hexadecimal notation it is given in Figure 1. For `sBoxLayer` the current state $b_{63} \dots b_0$ is considered as sixteen 4-bit words $w_{15} \dots w_0$, where $w_i = b_{4*i+3} \parallel b_{4*i+2} \parallel b_{4*i+1} \parallel b_{4*i}$ for $0 \leq i \leq 15$ and the output nibble $S[w_i]$ provides the updated state values in the obvious way [3].

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

FIGURE 1. Substitution table of Present cipher [3].

pLayer. Permutation layer of Present cipher is given in Figure 2. Bit i is moved to bit position $P(i)$.

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

FIGURE 2. Permutation layer of Present cipher [3].

Round transformations are displayed on Figure 3.

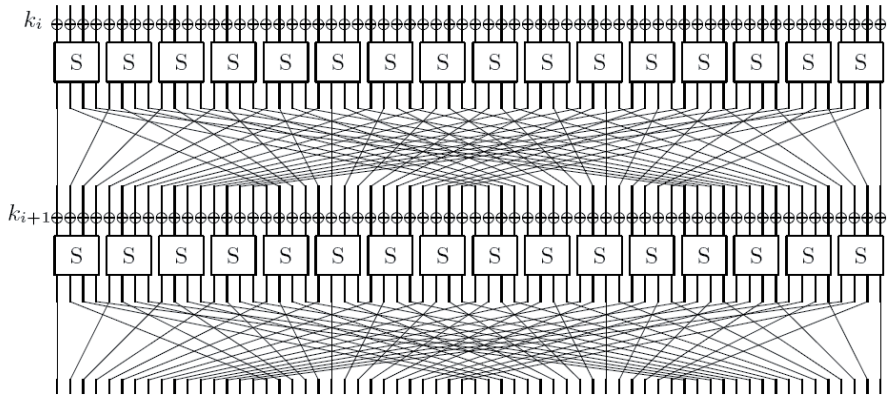

FIGURE 3. SP-network of Present cipher [3].

2.1.2. The key schedule

Present cipher can take both 80 and 128 bit keys. In this paper only 80 bit key is examined.

80 bit key. We denote the key entered by user as $k_{79}k_{78}\dots k_0$, that is stored in register K and is updated every round. At round i, 64-bit round key $K_i = \kappa_{63}\kappa_{62}\dots\kappa_0$ is derived from it

$$K_i = \kappa_{63}\kappa_{62}\dots\kappa_0 = k_{79}k_{78}\dots k_{16}.$$

After extracting the round key K_i, the key register $K = k_{79}k_{78}\dots k_0$ is updated as follows:

(1) $[k_{79}k_{78}\dots k_1k_0] = [k_{18}k_{17}\dots k_{20}k_{19}]$,
(2) $[k_{79}k_{78}k_{77}k_{76}] = S\,[k_{79}k_{78}k_{77}k_{76}]$,
(3) $[k_{19}k_{18}k_{17}k_{16}k_{15}] = [k_{19}k_{18}k_{17}k_{16}k_{15}] \oplus$ `round_counter`.

2.2. Existing cryptanalysis

Since the cipher was designed, there have been done several security evaluations of the cipher. The designers of Present [3] presented some security margins for differential, linear and algebraic cryptanalysis. They have written that any differential characteristic over 25 rounds must have at least $5 \times 10 = 50$ active S-boxes. The maximum differential probability of a Present S-box is 2^{-2} and so the probability of a single 25-round differential characteristic is bounded by 2^{-100}. For linear cryptanalysis they presented that the maximum bias of a 28--round linear approximation is bounded by 2^{-43}. So the linear cryptanalysis of the cipher would require of the order of 2^{84} known plaintexts/ciphertexts. For algebraic cryptanalysis designers started from the fact that the Present S-box can be described by 21 quadratic equations in the eight input/output-bit variables over $GF(2)$. The entire cipher can be described by $n \times 21 = 11067$ quadratic equations in $n \times 8 = 4216$ variables, where n is the number of S-boxes in the encryption algorithm and the key schedule. The designers of Present ran simulations on small-scale versions using F_4 algorithm in Magma. They were not able to get a solution in a reasonable time for a two-round version of the reduced cipher (seven S-boxes, block size of 28 bits).

In 2009 in [8] was discovered that 32 percent of Present keys (80-bit key size) are weak for linear cryptanalysis and the linear deviation can be much larger than the linear characteristic value by the multi-path effect. Also a 28-round path with a linear deviation of $2^{-39.3}$ for the weak keys was discovered, which is larger than 2^{-43} given by the designers with single-path evaluation. The linear attack was proposed for 24-rounds, where $2^{63.5}$ known texts are needed.

The same year, the linear hull and algebraic cryptanalysis, was studied by [7]. Authors proposed a linear attack for 25 rounds of Present (128-bit key size)

with the whole code book, $2^{96.68}$ 25-round encryptions, 2^{40} blocks of memory and 0.61 success rate. An algebraic attack for 5 rounds of Present (80-bit key size) was also introduced, where half of the user key bits were recovered in less than 3 minutes.

In 2010 in [6] attack on 25-round Present was proposed that can recover the 80-bit secret key with $2^{62.4}$ data complexity. They also show that the 26-round version of Present can be attacked faster than key exhaustive search with the 2^{64} data complexity by an advanced key search technique.

3. The method of syllogisms

3.1. The system of equations

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of variables. By $X_i, 1 \leq i \leq m$ we denote subsets of X of size $l_i \leq l$, where l is a natural number. The system of Boolean equations

$$f_1(X_1) = 0, \ldots, f_m(X_m) = 0 \tag{1}$$

is considered, where f_i depend only on variables from the set X_i. We say that system (1) is l-sparse. The main goal is to find a solution to (1). In order to use the method of syllogisms, the system of equations has to be rewritten to symbol representation.

Let S be the set of solutions of a system of Boolean equations $F(x) = 0$, i.e., $S = \{x \in GF(2)^n; F(x) = 0\}$. We usually suppose that F is defined in such a way, that there is exactly one solution to the system, $|S| = 1$. Let S_i be the set of solutions of ith equation of the system, i.e., $S_i = \{x \in GF(2)^n; f_i(x) = 0\}$. If f_i depends only on a small set of variables, we can represent S_i more effectivelly by storing only the values of variables on which f_i depends. We can store them as vectors of length $|X_i|$, indexed by variables in X_i in the chosen order and call these vectors partial solutions. We will denote a set of such vectors V_i. We will call (X_i, V_i) a symbol representation of the equation $f_i(x) = 0$, and the set $\mathcal{V} = \{(X_i, V_i); i = 1, \ldots, m\}$ a symbol representation of the system as defined in [13] and introduced in [9].

3.2. Description of the method of syllogisms

The method of syllogisms was proposed by [14] and adapted for algebraic cryptanalysis purposes by [10], [11], [12]. It is one of the local reduction methods. This section is written based on [10], [11], [12].

Syllogisms are logical arguments that are based on transitive relation of logical implication. Let x_i, x_j, x_k be variables, then syllogism consists of two premises $x_i \implies x_j$, $x_j \implies x_k$ and conclusion $x_i \implies x_k$. Let (X_i, V_i) be a symbol.

Let $s \in S$ be a solution, and let s_i be its projection to X_i. Then $s_i \in V_i$. Such vectors are called true partial solutions. If $v \in V_i$ is not a projection of any $s \in S$ then v is called a false partial solution and v is in conflict with the rest of the equation system. Then we can locally reduce the system and remove false solution from the system, so we get $(X_i, V_i \setminus \{v\})$. Let $x_i, x_j \in X_i$. Let us suppose that some vector $(a, b) \in \mathbf{Z}_2^2$ is not a projection of vector $v \in V_i$. Then the following 2-clause must be true $(x_i \neq a) \vee (x_j \neq b)$. We can rewrite this clause as implications $(x_i = a) \Longrightarrow (x_j \neq b)$ and $(x_j = b) \Longrightarrow (x_i \neq a)$. The basic part of the method of syllogisms is to collect all possible 2-clauses from each equation in the system and to check partial solutions against these 2-clauses. Syllogisms are used to derive new 2-clauses from existing 2-clauses and to remove false partial solutions using newly derived 2-clauses.

Set of all 2-clauses is represented by a directed graph, specifically implication graph. Vertices correspond to unit clauses $\{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ and the arcs represent implications of unit clauses, for example arc $(\neg x_1, x_2)$ corresponds to implication $\neg x_1 \Longrightarrow x_2$. 2-clause is represented by two arcs, e.g., $x_1 \vee x_2$ corresponds to $(\neg x_1, x_2)$ and $(\neg x_2, x_1)$. Loops (x_i, x_i) can be ignored, as they correspond to tautologies $x_i \vee \neg x_i)$. Arc $(\neg x_i, x_i)$ means that $x_i = 1$ and arc $(x_i, \neg x_i)$ means that $x_i = 0$. If both arcs are present, the conflict is detected and the equation system does not have a solution. Given a set of clauses P, the maximum information which can be learned is contained in the transitive closure of the corresponding implication graph. The corresponding set of clauses is denoted by $Cl(P)$. The method of syllogisms proceeds as follows [11]:

(1) LET $P \leftarrow \emptyset$ be a set of clauses. LET $c \leftarrow \emptyset$.
(2) FOR each symbol (X_i, V_i):
   (a) APPLY: Remove all partial solutions from V_i that violate any of the clauses from P.
   (b) COLLECT: For each pair of variables from X_i find all clauses in these 2 variables that are satisfied by each partial solution in V_i and store them in set P_i.
   (c) LET $P \leftarrow Cl(P \cup P_i)$
   (d) IF a conflict is detected, STOP the algorithm with result: No solution.
(3) IF $|P| = c$ (no new clauses), STOP algorithm, and output the reduced system.
(4) LET $c \leftarrow |P|$. REPEAT from step 2.

Algorithm provided can have more solutions: no solution, when a conflict is detected, a solution of the system or a reduced system (with some redundant partial solutions). The maximum cardinality of P is finite $(4n^2)$, so the algorithm is always stop [11].

If algorithm provides a reduced system with some redundant partial solutions and does not provide a solution, process can be continued with the introduction of guesses [1]. Guessing is used to estimate a value of some variable, e.g., $x_1 = 0$. If the guess is correct, reduced version is reached, if not, a conflict is detected. In the second case, backtracking of the algorithm has to be used and different guess has to be made. If g is the number of variables guessed until a solution/conflict is found, then the complexity of the process is 2^g.

3.3. Guessing strategies

In the experiments, four guessing strategies are used that were proposed in [13]:

***rand*:** This strategy is based on random selection, it is uninformed strategy. Every symbol can be chosen with the same probability. This strategy can be used to provide an estimate for the least sophisticated attack possible.

***guess*:** Let $l_i = |X_i|$ and $k_i = |V_i|$. Then *guess* strategy chooses randomly the symbol from the set of all symbols with maximal $2^{l_i}/k_i$. Based on the fact, that this strategy chooses symbols based on the local characteristics, it is a locally informed strategy.

***impact*:** This strategy chooses the symbol with variables that influence the highest number of equations in the system. *Impact* is a globally informed strategy.

***impact2*:** Strategy is very similar to *impact*, but in addition to observation of influence of the highest number of equations it also observes the number of solutions of compared symbols. It is also globally informed strategy.

3.4. Random and correct guessing

The values of the variables can be guessed by two strategies, introduced by [13] and [12]:

- random guessing - guess of the value of the variable x_i is made randomly,
- correct guessing - if the solution of the system is known, experiment can be made and guess of the value of variable x_i can be provided by oracle, so the correct value is substituted to the variable.

3.5. Used software

For the carrying out of the experiments, this software was used:

(1) **Generator of equations for Present**: generator of equations for Present cipher, that generates equations for cipher and key schedule based on the number of rounds (© Lucia Lacko-Bartošová [1]).

[1] lucialbartosova@gmail.com

(2) **Sylog**: software that uses the combination of local reduction based on the method of syllogisms with algorithm used for guessing a part of the state of the cipher (© Pavol Zajac [2]).

4. Experimental results

In this section, the results of our experiments with the Present cipher with 80 key bits are provided.

4.1. Goal of the experiments

Our goal was to compute experimental approximations of attack complexity on reduced-round and full round Present and to compare random and correct guessing for different guessing strategies and to choose the best one for the attack. As mentioned earlier, Present cipher has key length 80-bits and block length 64-bit. Therefore two versions of Present are examined, 80 key bit and 64 key bit, where 16 bits are known.

4.2. Results of the experiments

For the representation of the results, box plot is used. The ends of the whiskers represent the minimum and maximum, the upper edge of the box indicates the 75th percentile and the lower edge indicates the 25th percentile.

4.2.1. *Rand* strategy

On Figure 4 the complexity estimation depending on the number of rounds is depicted. Random guessing and correct guessing are compared together with the information about 16 key bits. Results show that when random guessing is applied, lower amount of information is necessary to evaluate if the guessed values are correct or not. That means, that if the values we guess are not correct, we reach conflict after guessing less information that in the case where we guess correct values. Next, if we give information about 16 key bits to the system, the complexity will be lower. Resistance against presented method of algebraic cryptanalysis is obtained in average after the 6th round and the complexity is growing linearly.

[2]Institute of Computer Science and Mathematics, Slovak University of Technology in Bratislava, pavol.zajac@stuba.sk.

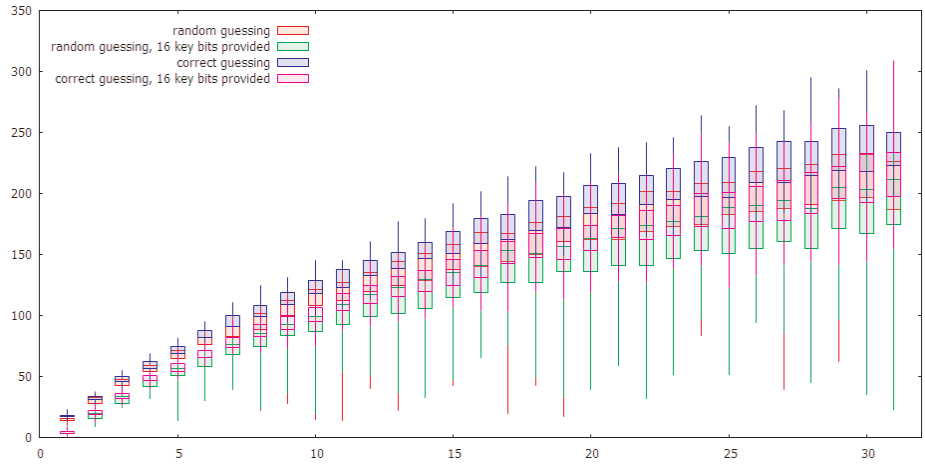

FIGURE 4. *Rand* strategy.

4.2.2. *Guess* strategy

On Figure 5 complexity estimation depending on the number of rounds is depicted. Random guessing and correct guessing are compared together with the information about 16 key bits. Results of the *guess* strategy are comparable to *rand* strategy and show that when random guessing is applied, lower amount of information is necessary to evaluate if the guessed values are correct or not. Comparable to *rand* strategy, if we give information about 16 key bits to the system, the complexity will be lower. Resistance against presented method of algebraic cryptanalysis is obtained in average after the 10th round.

4.2.3. *Impact* strategy

The complexity estimation of the *impact* strategy depending on the number of rounds is depicted on Figure 6. Random guessing, correct guessing and the information about 16 key bits is compared. Results show that when random guessing is applied, compared to *rand* and *guess* strategies, approximately the same amount of information is necessary to evaluate if the guessed values are correct or not. Figure 6 also shows that if we give information about 16 key bits to the system, the complexity will be 16 bits lower. Resistance against presented method of algebraic cryptanalysis is obtained in average after the 10th round. Complexity reaches the boundary of 80 bits, if we do not provide the values of 16 key bits, and the boundary of 64 bits, if we do provide the values of 16 key bits. That is exactly the key size.

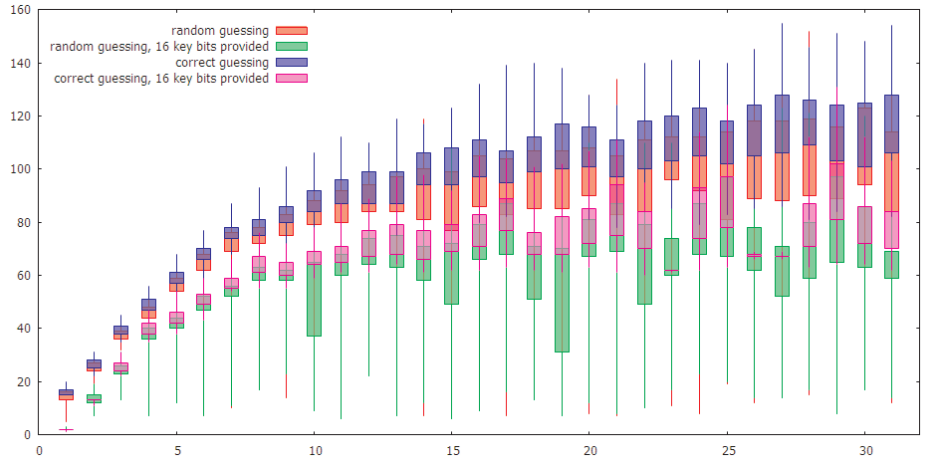

FIGURE 5. *Guess* strategy.

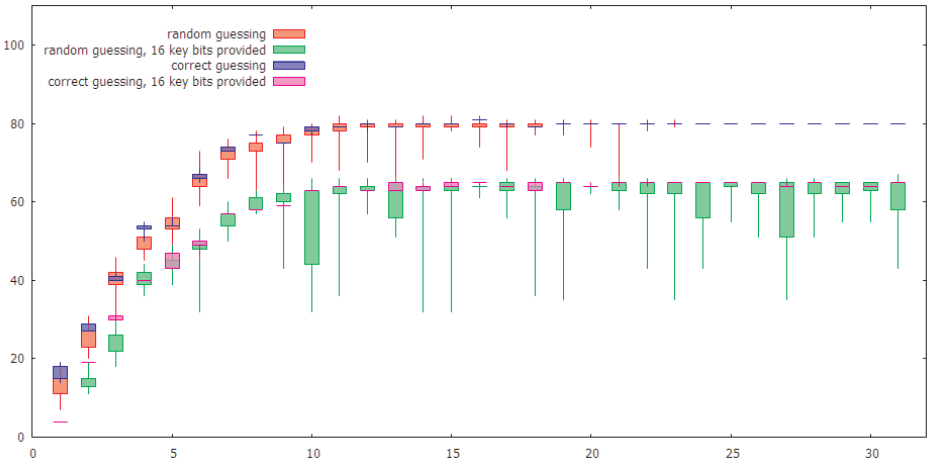

FIGURE 6. *Impact* strategy.

4.2.4. *Impact2* strategy

The complexity estimation of the *impact2* strategy depending on the number of rounds is depicted on Figure 7. Random guessing, correct guessing and the information about 16 key bits is compared. Results are comparable to *impact* strategy and show that when random guessing is applied, approximately the same amount of information is necessary to evaluate if the guessed values are correct or not. That means, if the values we guess are not correct, we reach conflict after guessing the same amount of information as in the case,

where we guess correct values. Figure 7 also shows that if we give information about 16 key bits to the system, the complexity will be 16 bits lower. Resistance against presented method of algebraic cryptanalysis is obtained in average after the 10th round. Complexity reaches the boundary of 80 bits, if we do not provide the values of 16 key bits, and the boundary of 64 bits, if we do provide the values of 16 key bits.

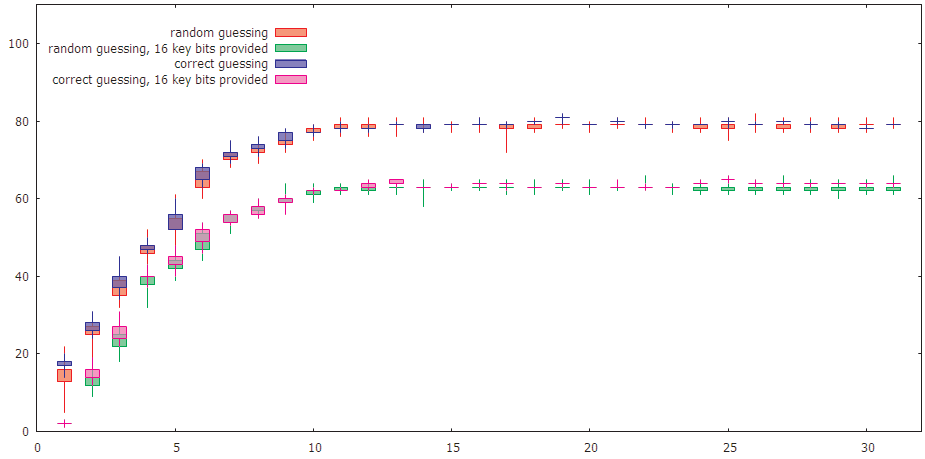

FIGURE 7. *Impact2* strategy.

5. Conclusion

Four different guessing strategies of the method of syllogisms were used to evaluate the security of Present cipher. *Rand* strategy gave the worst complexity of an attack, then followed the *guess* strategy and the best were *impact* and *impact2*. For the *rand* and *guess* strategy, when random guessing is applied, lower amount of information is necessary to evaluate if the guessed values are correct or not. In comparison, for *impact* and *impact2* strategy, if the values we guess are not correct, we reach conflict after guessing the same amount of information as in the case where we guess correct values. For all strategies, if we give information about 16 key bits to the system, the complexity will be lower (for *impact* and *impact2* strategy, 16 bits lower). Resistance against presented method of algebraic cryptanalysis is obtained in average after the 6th round for *rand* strategy, after the 10th round for *guess*, *impact* and *impact2* strategy. When compared to an attack conducted by [7], where 5 rounds were attacked by algebraic cryptanalysis, we can conclude that an attack was successfull for number of rounds, where a cipher has not yet developed a resistance against these types of an attack. Security of the Present cipher can be further studied also in terms of randomness of its output, e.g., using a new matrix test [5].

REFERENCES

[1] ADAMČEK, P. — LODERER, M. — ZAJAC, P.: *A comparison of local reduction and SAT–solver based algebraic cryptanalysis of JH and KECCAK*, 2012 (preprint).
[2] BARD, G. V.: *Algebraic Cryptanalysis*. Springer, Dordrecht, 2009.
[3] BOGDANOV, A.—KNUDSEN, L. R.—LEANDER, G.—PAAR, C.—POSCHMANN, A. – –ROBSHAW, M. J. B.—SEURIN, Y.—VIKKELSOE, C.: *PRESENT: An ultra-light- -weight block cipher*, in: Proc. Cryptographic Hardware and Embedded Systems– –CHES '07, The 9th International Workshop, Vienna, Austria, September 10–13, 2007, Lecture Notes in Comput. Sci., Vol. 4727, 2007, pp. 450–466.
[4] ECRYPT II, EUROPEAN NETWORK OF EXCELLENCE IN CRYPTOLOGY II: *Report on "Lightweight Cryptographic Algorithms"*, 2010, `www.ecrypt.eu.org/documents/D.SYM.5.pdf`.
[5] GROŠEK, O. — VOJVODA, M. — KRCHNAVÝ, R.: *A new matrix test for randomness*, Computing **85** (2009), 21–36.
[6] CHO, J. Y.: *Linear cryptanalysis of reduced-round PRESENT* CT-RSA, Lecture Notes in Comput. Sci., Vol. 5985, Springer, New York, 2010, pp. 302–317.
[7] NAKAHARA, J. — SEPEHRDAD, P. — ZHANG, B. — WANG, M.: *Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT* Proceedings of The 8th International Conference on Cryptology and Network Security, CANS '09, Springer, New York, 2009, pp. 58–75.
[8] OHKUMA, K.: *Weak keys of reduced-round PRESENT for linear cryptanalysis*, Selected Areas in Cryptography, Lecture Notes in Comput. Sci., Vol. 5867, Springer, New York, 2009, pp. 249–265.
[9] RADDUM, H.—SEMAEV, I.: *New technique for solving sparse equation systems*, Technical Report, Cryptology ePrint Archive, 2006.
[10] ZAJAC, P.: *On the use of the method of syllogisms in algebraic cryptanalysis*, in: Proc. of The 1st Plenary Conference of the NIL-I-004, Bergen 2009, pp. 21-30.
[11] ZAJAC, P.: *Solving trivium-based Boolean equations using the method of syllogisms*, Fund. Inform. **114** (2012), 2012, 359–373.
[12] ZAJAC, P.: *Use of the local reduction in experimental evaluation of the block cipher security* 2012 (preprint).
[13] ZAJAC, P. — ČAGALA, R.: *Local reduction and the algebraic cryptanalysis of the block cipher GOST*, Period. Math. Hungar. **65** (2012), 239–255.
[14] ZAKREVSKIJ, A. — VASILKOVA, I.: *Reducing large systems of Boolean equations*, The 4th International Workshop on Boolean Problems, September 21-22, 2000, Freiberg, Germany, pp. 21–28.

Received December 12, 2012

Mathematical Institute
Slovak Academy of Sciences
Štefánikova 49
SK–814-73 Bratislava
SLOVAKIA
E-mail: lucia.lacko-bartosova@mat.savba.sk