Application of cube attack to block and stream ciphers

Janusz Szmidt
joint work with Piotr Mroczkowski
Military University of Technology
Military Telecommunication Institute
Poland

23 czerwca 2009

1. Papers

- Itai Dinur and Adi Shamir, **Cube Attacks on Tweakable Black Box Polynomials**, Eurocrypt, 2009.
- S. S. Bedi and R. Pillai, **Cube attacks on Trivium**, IACR Cryptology ePrint Archive, 2009/15.
- J-P. Aumasson, W. Meier, I. Dinur, A. Shamir, **Cube testers and key recovery attacks on reduced round MD6 and Trivium**, Fast Software Encryption, 2009.
- I. Dinur, A. Shamir, **Side channel cube attacks on block ciphers**, IACR Cryptology ePrint Archive, 2009/127.
- J. Lathrop, **Cube Attacks on Cryptographic Hash Functions**, Master's Thesis, Rochester Institte of Technology, 2009.

Paper

Miachael Vielhaber, **Breaking One.Fivium by AIDA an Algebraic IV Differential Attack**, IACR Cryptology ePrint Archive, 2007.

2. The structure of attack

- Let us consider cryptosystem described by the polynomial:

$$p(v_1, \ldots, v_m, x_1, \ldots, x_n)$$

  depending on m public variables $v_1, \ldots, v_m$ (the initial value or plaintext) and on n secret variables $x_1, \ldots, x_n$ (the key).
- The value of the polynomial represents the ciphertext bit.
- In general, the polynomial p is not explicitelly known; it can be a *black box*.
- We will consider the known plaintext attack, where at the preprocessing stage the attacker has also an access to secret variables (initial values or keys).

3. The structure of attack, cont.

1. **The preprocessing stage**. The attacker can change the values of public and secret variables. The task is to obtain a system of linear equations on secret variables.
2. **The stage *on line* of the attack**. The key is secret now. The attacker can change the values of public variables. He adds the output bits, where the inputs run over some multi-dimennsional cubes. The task is to obtain the right hand sides of linear equations. The system of linear equation can be solved giving some bits of the key.

4. Mathematical background

- For a moment we shall not distinguish the secret and public variables.
- Let p be a polynomial of n variables $x_1, \ldots, x_n$ over the field $GF(2)$.
- For a subset of indexes $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ let us take a monomial

$$t_I = x_{i_1} \ldots x_{i_k}$$

- Then we have a decomposition

$$p(x_1, \ldots, x_n) = t_I \cdot p_{S(I)} + q(x_1, \ldots, x_n)$$

where the polynomial $p_{S(I)}$ does not depend on the variables $x_{i_1}, \ldots, x_{i_k}$.

5. Example 1

- Let us consider a polynomial p of degree 3 depending on 5 variables:

$$p(x_1, x_2, x_3, x_4, x_5) = x_1x_2x_3 + x_1x_2x_4 + x_2x_4x_5 +$$

$$x_1x_2 + x_3x_5 + x_2 + x_5 = 1$$

- Let $I = \{1.2\}$ be a chosen subset of indexes.
- Then the polynomial p can be decomposed as:

$$p(x_1, x_2, x_3, x_4, x_5) = x_1x_2(x_3 + x_4 + 1) +$$

$$(x_2x_4x_5 + x_3x_5 + x_2 + x_5 + 1)$$

6. Example 1, cont.

Using the introduced above notation:

$$t_I = x_1 x_2,$$

$$p_{S(I)} = x_3 + x_4 + 1,$$

$$q(x_1, x_2, x_3, x_4, x_5) = x_2 x_4 x_5 + x_3 x_5 + x_2 + x_5 + 1$$

7. Definition 1

The maxterm of the polynomial p we call the monomial t_I, such that

$$deg(p_{S(I)}) = 1,$$

it means that the polynomial $p_{S(I)}$ corresponding to the subset of indexes I is a linear one, which is not a constant.

8. Summation over cubes

- Let $I = \{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$ be a fixed subset of k indexes.
- The set I defines the k-dimensional boolean cube C_I, where on the place of each of the indexes we put 0 or 1.
- A given vector $v \in C_I$ defines the derived polynomoal p_v depending on $n - k$ variables, where in the basic polynomoal p we put the values corresponding to the vector v.
- Summing over all vectors in the cube C_I we obtain the polynomial:

$$p_I = \sum_{v \in C_I} p_v$$

9. Theorem 1

Let p be a polynomial over the field $GF(2)$ and $I \subset \{1, \ldots, n\}$ the index subset. Then we have :

$$p_I = p_{S(I)} \mod 2$$

10. Example 2

- Let us consider a polynomial:

$$p(v_1, v_2, v_3, x_1, x_2, x_3) = v_1v_2v_3 + v_1v_2x_1 + v_1v_3x_1 + v_2v_3x_1 +$$

$$v_1v_2x_3 + v_1v_3x_2 + v_2v_3x_2 + v_1v_3x_3 + v_1x_1x_3 + v_3x_2x_3 + x_1x_2x_3 +$$

$$v_1v_2 + v_1x_3 + v_3x_1 + x_1x_2 + x_2x_3 + x_2 + v_1 + v_3 + 1$$

  of degree $d = 3$ depending on public variables v_1, v_2, v_3 and secret variables x_1, x_2, x_3.
- Substituting on public variables the values $0/1$ we get the eight derived polynomials.

11. Example 2, cont.

$$p(0,0,0,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_2x_3 + x_2 + 1$$

$$p(0,0,1,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_1 + x_2$$

$$p(0,1,0,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_2x_3 + x_2 + 1$$

$$p(0,1,1,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2$$

$$p(1,0,0,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_2 + x_3$$

$$p(1,0,1,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_1x_3 + 1$$

$$p(1,1,0,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_1x_3 + x_2x_3 + x_1 + x_2 + 1$$

$$p(1,1,1,x_1,x_2,x_3) = x_1x_2x_3 + x_1x_2 + x_1x_3 + x_2 + x_3 + 1$$

12. Example 2, cont.

- Summing the four derived polynomials with $v_1 = 0$ we get $x_1 + x_2$,
- Summing the four derived polynomials with $v_2 = 0$ we get $x_1 + x_2 + x_3$,
- Summing the four derived polynomials with $v_3 = 0$ we get $x_1 + x_3 + 1$.
- The obtained expressions lead to a system of linear equations used in the stage *on line* of the attack.

13. The preprocessing stage

1. The first task is to fix dimension of the cube and the public variables over which we will sum; they are called *the tweakable variables*, and the other public variables are put to zero. In the case we know the degree d of the basic polynomial, we put the cube dimension to $d-1$.
2. We do the summation over a fixed cube for several values of secret variables and collect the obtained values.
3. We do the linear tests for the obtained function of secret variables and store it when it is linear:

$$f(x \oplus x') = f(x) \oplus f(x') \oplus f(0),$$

where $x = (x_1, \ldots, x_n)$ are secret variables (the key).

14. The preprocessing stage, cont.

- The next task is to calculate the exact form (the coefficients) of the obtained linear function of secret variables.
- The free term of the linear function we obtan putting its all argument equal zero.
- The coefficient of the variable x_i is equal 1 if and only if the change of this variable implies the change of values of the function.
- The coefficient of the variable x_i is equal 0 if and only if the change of this variable does not imply the change of values of the function.

15. The preprocessing stage, cont.

- The task of this stage of attack is to collect possible many independent linear terms - they constitute the system of linear equations on secret variables.
- This system of linear equations will be used in the *on line* stage of attack.
- The preprocessing stage is done only once in cryptanalysis of the algorithm.

16. The stage *on line* of attack

- Now an attacker has the access only to public variables (thr plaintext for block ciphers, the initial values for stream ciphers), which he can change and calculates the corresponding bits of the ciphertext under the unknown value of secret variables.
- The task of this stage of attack is to find some bits of secret key with complexity, which would be lower than the exhuastive search in the brute force attack.
- The attacker uses the derived system of linear equations for secret variables (the unknown bits of the key), where the right hand sides of these equations are the values of bits of ciphertext obtained after summation over the same cubes as in the preprocessing stage, but now the key is not known.

17. The stage *on line* of attack, cont.

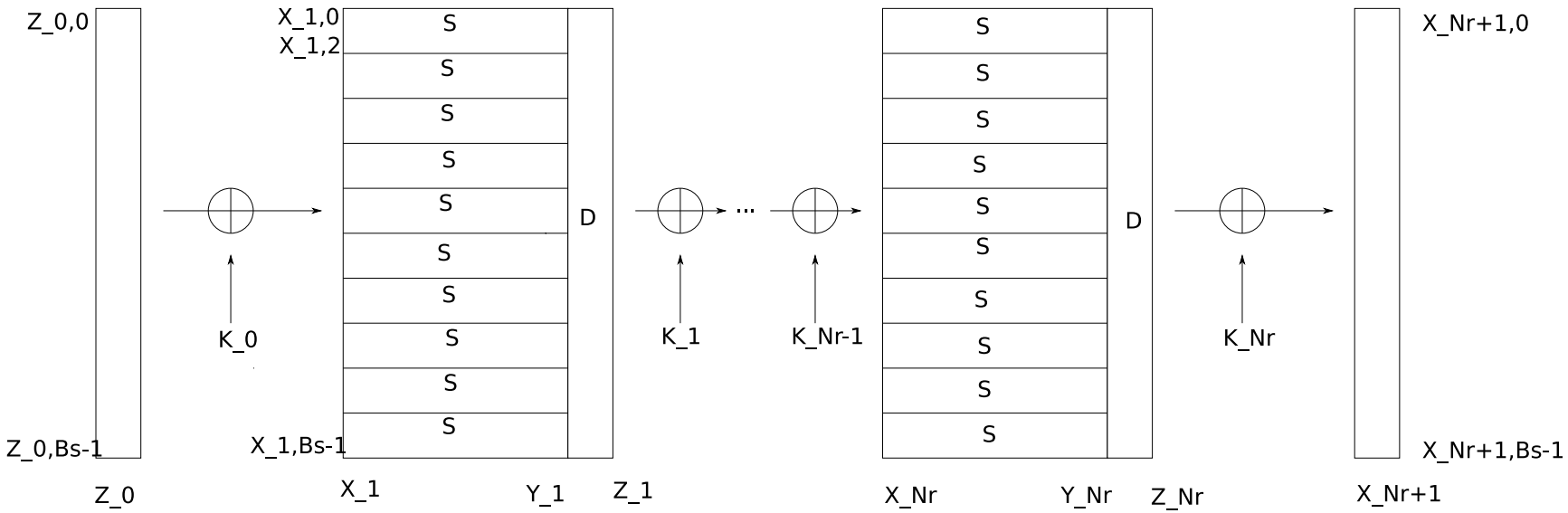
- The cube attach is applicable to symmetric ciphers for which the polynomials descibing the system have relatively low degree.
- Then one can eventually find some bits of unknown key. The remaining bits of the key may be found by brute force search.
- After successful preprocwssing stage, the stage *on line* of attack can be done many times for different unknown keys
- The cube attack is applicable, in general, to cryptosystems without knowing their inner structure. The attacker must have the possibility to realize the preprocessing stage and in the *on line* stage an access to the implementation of the algorithm (to perform the summation over cubes under unknown key).

18. CTC - Courtois Toy Cipher

Specification

- CTC was designed by Nicolais Courtois to apply and test the methods of algebraic analysis.
- It is a SPN network which is scalable in the number of rounds, the block and key size.
- Each round performs the same operation on the input data, except that a different round key is added each time. The number of rounds is denoted by N_r. The output of round $i-1$ is the input to round i.
- Each round consists of parallel applications of B S-boxes (S), the application of the linear diffusion layer (D), and a final key addition of the round key. The round key K_0 is added to the plaitext block before the first round.

19. CTC overviev for B = 10



20. CTC - Courtois Toy Cipher

Specyfication, cont.

- The plaintext bits $p_0 \ldots p_{Bs-1}$ are identified with $Z_{0,0} \ldots Z_{0,Bs-1}$ and the ciphertext bits $c_0 \ldots c_{Bs-1}$ are identified with $X_{N_r+1,0} \ldots X_{N_r+1,Bs-1}$ to have an uniform notation.
- The S-box was chosen as the permutation

$$[7, 6, 0, 4, 2, 5, 1, 3]$$

It has $2^3 = 8$ inputs and 8 outputs. The output bits are quadratic boolean functions of the input bits. The explicit form of these functions is not used in cube attack.

21. CTC - Courtois Toy Cipher

Specyfication, cont.

- The diffusion layer (D) is defined as

$$Z_{i,257 modBs} = Y_{i,0}$$

for all $i = 1 \ldots N_r$,

- $$Z_{i,(1987j+257) modBs} = Y_{i,j} + Y_{i,(j+137) modBs}$$

for $j \neq 0$ and all i, where $Y_{i,j}$ represents input bots and $Z_{i,j}$ represents output bits.

22. CTC - Courtois Toy Cipher

Specyfication, cont.

- The key schedule is a simple permutation of bits:

$$K_{i,j} = K_{0,(i+j) mod Bs}$$

  for all i and j, where K_0 is the main key.
- Key addition is performed bit-wise:

$$X_{i+1,j} = Z_{i,j} + K_{i,j}$$

  for all $i = 1 \ldots N_r$ and $j = 1 \ldots Bs - 1$, where $Z_{i,j}$ represents output bits of the previous diffusion layer, $X_{i+1,j}$ the input bits of the next round, and $K_{i,j}$ the bits of the current round key.

23. Cube attack on CTC

- We have applied the cube attack to the version of CTC with four rounds and B $= 40$ of S-boxes, i.e. the block and the key sizes being 120 bits.
- In the preprocessing stage we have done the summation over 50000 randomly chosen four dimensional cubes taken from the plaintext (other bits of the plaintext are put to zero). Then 757 boxes lead to linear expresions (maxterms) for bits of the key. For derivation of each linear expresion we used 5000 linear tests. We have chosen 120 linearly independent maxterms. They give the solvable system of linear equation for bits of the key.

24. The linear equations

The equation	Cube indexes
1+x66+x68 = c66	{4,5,22,52}
x27+x28 = c105	{1,60,62,90}
1+x58 = c18	{16,17,60,110}
1+x14 = c80	{29,38,61,106}
1+x54+x56 = c36	{41,55,64,115}
1+x115+x116 = c66	{5,10,22,89}
1+x43 = c11	{73,74,75,118}
x69+x70 = c28	{48,68,77,110}
1+x25 = c70	{73,76,93,104}
x78+x79 = c114	{10,39,70,118}

There are together 120 linearly independent such equations for the bits $x_0, \ldots, x_{119}$ of the key; $c_0, \ldots, c_{119}$ are output bits.

25. Cube attack on CTC, cont.

- In the *on line* stage of the attack we need to sum over 120 chosen cubes (the key is unknown) to find the right hand sides of the linear equations.
- The complexity of the attack here is about $2^7 \cdot 2^4 = 2^{11}$ encryptions of the four round CTC to recover the full 120-bit key.
- In fact, this reduced round cipher is described by a system of linear equations.
- Probably, for biger number of rounds, it is impossible to find such a description.

26. The stream cipher Trivium

The specification of algorithm

- Algorithm Trivium (the authors: C. de Canniere and B. Preneel) is one of the finalists of eSTREAM competitions.
- The basic parameters are the 80-bit key and the 80-bit initial value.
- The inner state of Trivium are 288 bits loaded to three nonlinear registers of different lengths.
- In each round of the algorithm the registers are shifted on one bit.
- The feedback in each register is given by a nonlinear function.

27. The specification of Trivium

$$(s_1, s_2, \ldots, s_{93}) \leftarrow (k_1, k_2, \ldots, k_{80}, 0, \ldots, 0)$$

$$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (IV_1, IV_2, \ldots, IV_{80}, 0, \ldots, 0)$$

$$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (0, 0, \ldots, 0, 1, 1, 1)$$

for i $= 1$ to 1152

$$t_1 \leftarrow s_{66} + s_{93}$$

$$t_2 \leftarrow s_{162} + s_{177}$$

$$t_3 \leftarrow s_{243} + s_{288}$$

$$t_1 \leftarrow t_1 + s_{91} \cdot s_{92} + s_{171}$$

$$t_2 \leftarrow t_2 + s_{175} \cdot s_{176} + s_{264}$$

$$t_3 \leftarrow t_3 + s_{286} \cdot s_{287} + s_{69}$$

28. The specification of Trivium, cont.

$$(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$$

$$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$$

$$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$$

end for

29. The specification of Trivium, cont.

- The generation of the ouput btstring (z_i) of the maximal length up to $N = 2^{64}$ bits, can be represented as:
  for i=1 to N

$$t_1 \leftarrow s_{66} + s_{93}$$

$$t_2 \leftarrow s_{162} + s_{177}$$

$$t_3 \leftarrow s_{243} + s_{288}$$

$$z_i \leftarrow t_1 + t_2 + t_3$$

$$t_1 \leftarrow t_1 + s_{91} \cdot s_{92} + s_{171}$$

$$t_2 \leftarrow t_2 + s_{175} \cdot s_{176} + s_{264}$$

$$t_3 \leftarrow t_3 + s_{286} \cdot s_{287} + s_{69}$$

30. The specification of Trivium, cont.

$$(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$$

$$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$$

$$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$$

end for

31. The cube attack on Trivium

- Dinur and Shamir investigated the reduced version of Trivium which contains 672 (instead of 1152) initialization rounds.
- During the preprocessing stage they obtained 63 linearly independent maxterms corresponding to 12-dimensional cubes and output bits of the indices from 672 to 685.

32. The cube attack on Trivium, cont.

- In the *on line* stage the attacker must find the values of the maxterms summing over 63 12-dimensional cubes.
- After solving the system of linear equations the attacker obtains 63 bits of the key and the remaining 17 bits of the key are found by brute force search.
- The complexity of the attack (in the on line stage) is ca. 2^{19} evaluatioins of the investigated, reduced algorithm. It is smaller than the complexity 2^{55} in the previous attacks on this version of Trivium.

Thank you