The cube attack on stream cipher Trivium and quadraticity tests

Piotr Mroczkowski Janusz Szmidt
Military Communication Institute
Poland

17 sierpnia 2010

Cube Attack - Papers and Preprints

- Itai Dinur and Adi Shamir. **"Cube Attacks on Tweakable Black Box Polynomials"**, Eurocrypt, 2009
- Michael Vielhaber. **"Breaking One. Fivium by AIDA an Algebraic IV Differential Attack"**, IACR Cryptology ePrint Archive, 2007.
- 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.
- P. Mroczkowski, J. Szmidt. **The Cube Attack on Courtois Toy Cipher**, IACR Cryptology ePrinf Archive, 2009/497.

1. **The preprocessing stage**
   - The attacker can change the values of public and secret variables.
   - The task is to obtain a system of quadratic and 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.
   - The task is to obtain the right hand sides of equations.
   - The system of equation can be solved giving some bits of the key.

Boolean functions

- During the *preprocessing stage* there are analysed Boolean functions $f(x_0, x_1 \dots, x_{n-1})$ depending on n secret variables (bits of the key) appearing in the process of summation over k-dimensional cubes in public variables; $0 < k < n - 1$.
- The task is to detect the cases where these functions are affine ones:

$$f(x_0, \dots, x_{n-1}) = \bigoplus_{0 \leqslant i \leqslant n-1} a_i x_i \oplus c$$

where $a_0, \dots, a_{n-1}, c$ are binary coefficients.

Boolean functions, cont.

- And to detect other cases where these functions are quadratic ones:

$$f(x_0,\ldots,x_{n-1}) = \bigoplus_{0\leqslant i<j\leqslant n-1} a_{ij}x_ix_j \oplus \bigoplus_{0\leqslant i\leqslant n-1} a_ix_i \oplus c$$

  where a_{ij}, a_i, c are binary coefficients.

- Affine functions are recognized by applying the lenearity tests:

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

  for chosen values of collections of secret variables: $x=(x_0,\ldots,x_{n-1}), x'=(x'_0,\ldots,x'_{n-1})$.

Boolean functions, cont.

- And to recognize quadratic functions we apply the quadraticity tests:

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

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

  for chosen values of collections of secret variables: $x = (x_0, \ldots, x_{n-1}), x' = (x'_0, \ldots, x'_{n-1}), x'' = (x''_0, \ldots, x''_{n-1})$.
- The binary coefficients in Algebraic Normal Forms of Boolean functions are calculated by summing over suitable cubes.

Trivium stream cipher, cont.

We applied the above process to Trivium stream cipher with reduced number ($740 \div 752$) of initialization rounds.
Here there are sample examples of obtained quadratic equations for bits of secret key:
745, $\{2,3,5,6,11,13,16,18,20,22,24,26,27,28,33,34,35,36,42,45,50,52,55,59,62,63,64,69,70,73\}$, $x8+x35+x9x10 = 1$
746, $\{3,4,6,7,12,14,17,19,21,23,25,27,28,29,34,35,36,37,43,46,51,53,56,60,63,64,65,70,71,74\}$, $x9+x36+x10x11 = 1$

Trivium stream cipher, cont.

747, {4,5,7,8,13,15,18,20,22,24,26,28,29,30,35,36,37,38,44,
47,52,54,57,61,64,65,66,71,72,75}, $x10+x37+x11x12 = 1$
748, {5,6,8,9,14,16,19,21,23,25,27,29,30,31,36,37,38,39,45,
48,53,55,58,62,65,66,67,72,73,76}, $x11+x38+x12x13 = 1$
749, {6,7,9,10,15,17,20,22,24,26,28,30,31,32,37,38,39,40,46,
49,54,56,59,63,66,67,68,73,74,77}, $x12+x39+x13x14 = 1$
750, {7,8,10,11,16,18,21,23,25,27,29,31,32,33,38,39,40,41,47,
50,55,57,60,64,67,68,69,74,75,78}, $x13+x40+x14x15 = 1$
751, {8,9,11,12,17,19,22,24,26,28,30,32,33,34,39,40,41,42,48,
51,56,58,61,65,68,69,70,75,76,79}, $x14+x41+x15x16 = 1$
742, {0,9,10,11,14,23,24,26,27,30,34,36,39,40,42,44,45,47,48,
49,51,54,63,64,65,66,67,69,74,77}, $x16+x43+x17x18 = 1$
743, {1,10,11,12,15,24,25,27,28,31,35,37,40,41,43,45,46,48,
49,50,52,55,64,65,66,67,68,70,75,78}, $x17+x44+x18x19 = 0$

Trivium stream cipher, cont.

740, {1,5,7,8,10,13,14,20,22,34,38,39,40,45,46,48,52,56,57,58, 60,62,63,64,65,66,69,75,78,79}, x18x23 $=$ 0
744, {1,2,4,6,11,12,18,26,34,36,38,48,50,53,54,55,56,57,58,59, 60,61,62,64,67,68,71,73,76,77}, x17$+$x59$+$x60x61 $=$ 1
752, {0,2,5,7,14,21,23,25,28,29,32,37,39,40,43,44,46,48,56,58, 59,60,63,67,69,70,75,76,77,79}, x0$+$x27$+$x1x2 $=$ 1
We used fast implementation of Trivium in Python - 128 independent key streams.
Paul Crowley, *Trivium, SSE2, CorePy, and the cube attack*. Published on http://www.lshift.net/blog/

Thank you