MILP-based Cube Attack on the Reduced-Round WG-5 Lightweight Stream Cipher

Raghvendra Rohit, Riham AlTawy, & Guang Gong

Department of Electrical and Computer Engineering, University of Waterloo
Waterloo, ON, N2L 3G1, CANADA

IMACC 2017, 12-14 December 2017
St Catherines College, University of Oxford, Oxford

Outline

- Introduction
- WG stream cipher
- Cube attack on WG-5
- Comparison with Grain128a & Trivium
- Conclusions

Introduction

Cube attacks

- Proposed in 2007[1], 2009[2]
- `Basic idea:`

Let $f : \mathbb{F}_2^5 \to \mathbb{F}_2$ given by

$$f(k_0, k_1, k_2, v_0, v_1) = v_0 v_1 k_0 + v_0 v_1 k_2 + v_0 v_1 + k_0 k_1 + v_1 k_2 + k_2 + 1$$

$$\implies f(k_0, k_1, k_2, v_0, v_1) = v_0 v_1 (k_0 + k_2 + 1) + k_0 k_1 + v_1 k_2 + k_2 + 1$$

Summing f over all possible choices of v_0, v_1 gives

$$f(k_0, k_1, k_2, 0, 0) + f(k_0, k_1, k_2, 0, 1) + f(k_0, k_1, k_2, 1, 0) +$$
$$f(k_0, k_1, k_2, 1, 1) = k_0 + k_2 + 1,$$

which gives a linear relation of the two key bits k_0 and k_2.

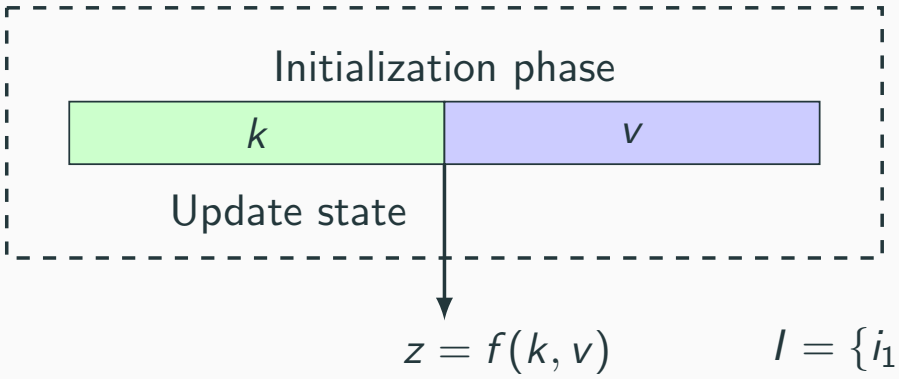
---

[1] Vielhaber, M.: Breaking ONE.FIVIUM by AIDA an algebraic IV differential attack. Cryptology ePrint Archive, Report 2007/413

[2] Dinur, I., and Shamir, A. Cube attacks on tweakable blackbox polynomials. EUROCRYPT 2009

Cube attacks (ctd.)

Mathematical description



$k = (k_0, k_1, \ldots, k_{n-1})$: secret variables

$v = (v_0, v_1, \ldots, v_{m-1})$: public variables

$z = f(k, v)$: first keystream bit

$I = \{i_1, i_2, \cdots, i_{|I|}\} \subset \{0, 1, \cdots, m-1\}$: cube indices

- $f(k, v)$ can be represented as :

$$f(k, v) = t_I \cdot p(k, v) + q(k, v),$$

where, $t_I = v_{i_1} v_{i_2} \cdots v_{i_{|I|}}$, $p(k, v)$ is a polynomial that does not contain any of the cube indices variables $(v_{i_1}, v_{i_2}, \cdots, v_{i_{|I|}})$, and $q(k, v)$ is independent of at least one variable from $(v_{i_1}, v_{i_2}, \cdots, v_{i|I|})$.

Cube attacks (ctd)

Let C_I denote the set of all the possible $2^{|I|}$ values of $(v_{i_1}, v_{i_2}, \cdots, v_{i|I|})$, and the remaining input $n + m - |I|$ variables are set to some constant values, then

$$\bigoplus_{C_I} f(k, v) = p(k, v)$$

- $p(k, v)$ is called superpoly corresponding to cube C_I.
- Simpler $p(k, v)$ leads to algebraic attacks by solving equations.

Division property

Definition: Division property[3]

Let $\mathbb{X} \subseteq \mathbb{F}_2^n$, $0 \leq k \leq n$, we say that $\mathbb{X}$ has the division property $\mathcal{D}_k^n$ if $\bigoplus_{x \in \mathbb{X}} \pi_u(x) = 0$, for all $u \in \mathbb{F}_2^n$ s.t $w_u < k$.

Definition: Bit based Division property[4]

Let $\mathbb{X}$ be a multiset whose elements take a value of $\mathbb{F}_2^n$. Let $\mathbb{W}$ be a set whose elements take an n-dimensional vector of binary elements. The multiset $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{W}}^{1,n}$ if it fulfills the following conditions:

$$\bigoplus_{x \in \mathbb{X}} \pi_u(x) = \begin{cases} \textit{unknown} & \textit{if there exists } w \in \mathbb{W} \textit{ s.t } u \succeq w, \\ 0 & \textit{otherwise,} \end{cases}$$

where $u, w, x \in \mathbb{F}_2^n$, $\pi_u(x) = \prod_{i=0}^{n-1} x_i^{u_i}$ and $u \succeq w$ if $u_i \geq w_i$ for all i.

[3] Todo, Y.: Structural evaluation by generalized integral property. EUROCRYPT 2015.

[4] Todo, Y., and Morii, M. Bit-based division property and application to simon family. FSE 2016

MILP models for bit-based division property propagation

Mixed Integer Linear Programming models

$$a \xrightarrow{COPY} \{b_1, b_2, \ldots, b_m\}$$

$M.var \leftarrow a, b_1, b_2, \ldots, b_m$ `as binary`.

$M.con \leftarrow a = b_1 + b_2 + \ldots + b_m$.

$$\{a_1, a_2, \ldots, a_m\} \xrightarrow{XOR} b$$

$M.var \leftarrow a_1, a_2, \cdots a_m, b$ `as binary`.

$M.con \leftarrow a_1 + a_2 + \cdots + a_m = b$.

$$\{a_1, a_2, \ldots, a_m\} \xrightarrow{AND} b$$

$M.var \leftarrow a_1, a_2, \cdots a_m, b$ `as binary`.

$M.con \leftarrow b \geq a_i$ `for` $i = 1, 2, \cdots, m$.

Solutions of inequalities corresponds to all division trails.

Division property & Cube attacks

To check if secret variable k_j is involved in superpoly

1. For a given cube C_I, start with the initial division property $\mathcal{D}^{1,n}_{\mathbb{W}}$, where $\mathbb{W} = \{(v, e_j)\}$ and $v_i = 1$ if $i \in \{i_1, i_2, \cdots, i_{|I|}\}$, $k_j = 1$ and $v_i = 0, k_j = 0$ for all remaining indices.
2. Add the constraint $z = 1$
3. If there is no division trial s.t steps 1 & 2 are satisfied, then k_j is not involved in the superpoly of C_I [Todo et al.][5].

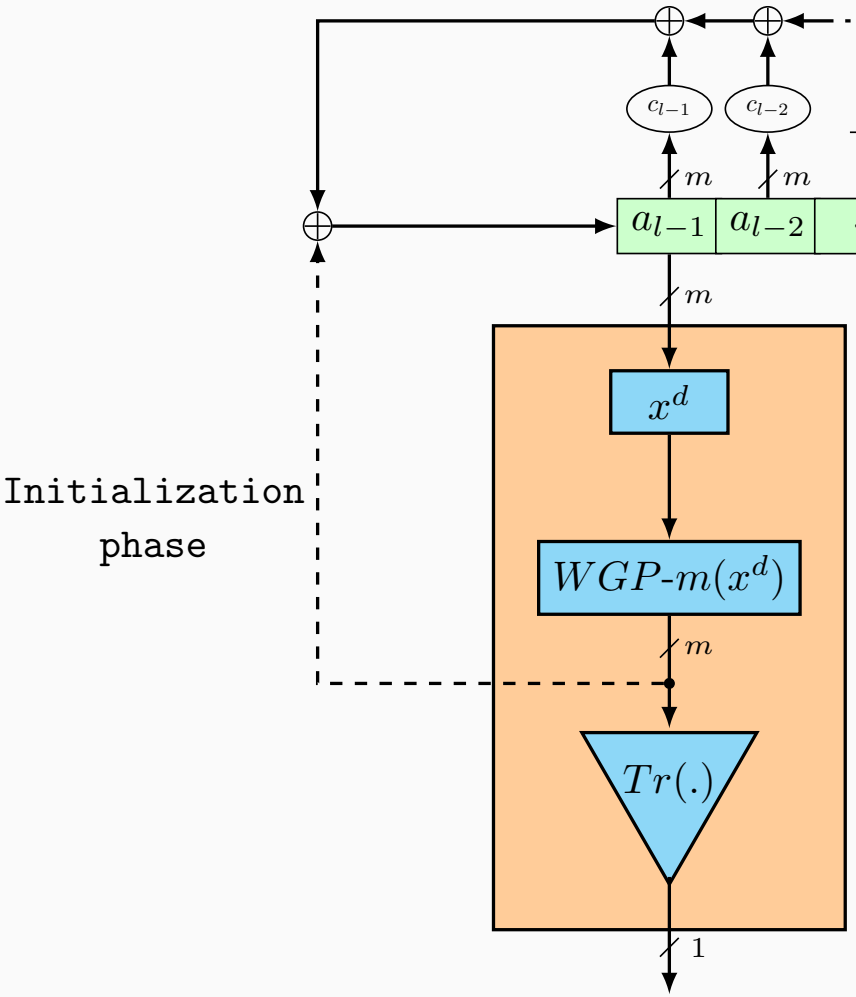
[5] Todo, Y., Isobe, T., Hao, Y., and Meier, W. Cube attacks on non-blackbox polynomials based on division property. CRYPTO 2017

Our Contributions

- We investigate the security of nonlinear intialization phase of WG-5 with respect to cube attacks.
- We present an argument to show WG-5 initialization phase is more resistant to cube attacks than that of Grain128a and Trivium.

WG stream cipher

General architecture for WG ciphers



Mathematical parameters

m : Bit width of LFSR

$g(x)$: Generating polynomial for GF(2^m)

$p(x) = \sum_{i=1}^{l-1} c_i x^i + x^l$ Primitive polynomial for LFSR

l : Degree of p

Find k s.t $3k \equiv 1 \bmod m$

$r_1 = 2^k + 1$

$r_2 = 2^{2k} + 2^k + 1$

$r_3 = 2^{2k} - 2^k + 1$

$r_4 = 2^{2k} + 2^k - 1$

$WGP\text{-}m(x) = t(x+1) + 1$

$t(x) = x + x^{r_1} + x^{r_2} + x^{r_3} + x^{r_4}$, where $x \in GF(2^m)$

$\gcd(d, 2^{m-1}) = 1$

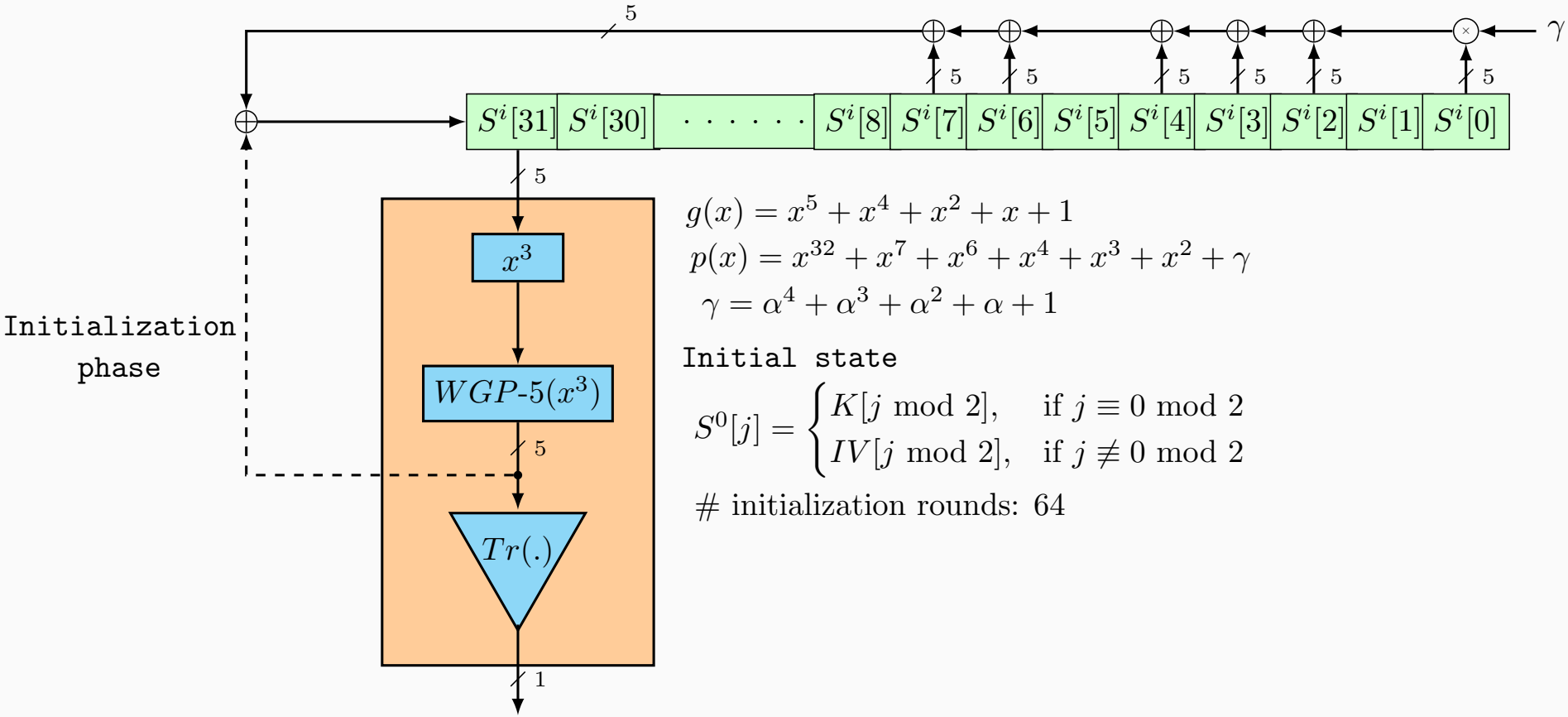
WG ciphers: Randomness properties

Randomness properties of WG keystream

- Long period: $2^{lm} - 1$
- Balanced
- Ideal 2-level autocorrelation
- Ideal t-tuple distribution

WG-5 Specification

- WG-5[6] is a lightweight version of eSTREAM submission WG cipher[7]



$$g(x) = x^5 + x^4 + x^2 + x + 1$$

$$p(x) = x^{32} + x^7 + x^6 + x^4 + x^3 + x^2 + \gamma$$

$$\gamma = \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1$$

Initial state

$$S^0[j] = \begin{cases} K[j \bmod 2], & \text{if } j \equiv 0 \bmod 2 \\ IV[j \bmod 2], & \text{if } j \not\equiv 0 \bmod 2 \end{cases}$$

initialization rounds: 64

[6] Aagaard, M. D., Gong, G., and Mota, R. K. Hardware implementations of the wg-5 cipher for passive rfid tags.

[7] Nawaz, Y., and Gong, G. Wg: A family of stream ciphers with designed randomness properties.

Cube attack on WG-5

Attack framework

Notations

- key: $k = (k_0, k_1, \ldots, k_{79})$, IV: $v = (v_0, v_1, \ldots, v_{79})$
- first keystream bit: $z = f(k, v)$
- superpoly: $\oplus_{C_I} f(k, v) = p(\bar{k}, \bar{v})$, where
  C_I is the cube of length $|I|$,
  $\bar{v} = \{\{v_0, v_1, \ldots, v_{79}\} - \{v_{i_1}, v_{i_2}, \ldots, v_{i_{|I|}}\}\}$,
  $\bar{k} = \{k_{j_1}, k_{j_2}, \ldots, k_{j_{|J|}}\}$, and
  $|J|$ is the number of variables in $\bar{k}$

Attack framework

The attack consists of two phases: 1) Offline phase 2) Online phase

Offline phase

Goal: To recover a superpoly that is almost balanced for a given cube C_I.

Steps:

1. Create a MILP model M that encodes the division trails for WG-5 reduced to R rounds.
2. Evaluate the secret variables $\bar{k}$ involved in the superpoly p.
3. Choose a value for $\bar{v}$ and recover $p(\bar{k}, \bar{v})$ by trying out all $2^{|I|+|J|}$ possible values. Also, store $p(\bar{k}, \bar{v})$ for all values of $\bar{k}$.

Attack framework (ctd.)

Online phase

Goal: To recover the entire secret key.

Steps:

1. Query the cube C_I to the encryption oracle to obtain the value of $p(\bar{k}, \bar{v})$ and compare to the previously stored values. This step reduces the keyspace by half.
   We use multiple cubes to reduce keyspace further.
2. Guess the remaining secret key values.

MILP model for WG-5 initialization

Algorithm 1 MILP model for the initialization of WG-5

function $\mathrm{WG5EVAL}(R)$
  Prepare empty MILP Model M
  $M.var \leftarrow S^0[j]$ for $0 \leq j \leq 31$ ▷ $S^0[j] = (s^0_{5j}, s^0_{5j+1}, s^0_{5j+2}, s^0_{5j+3}, s^0_{5j+4})$
  for $i = 1$ to R **do**
    $(M, S', a) = WGP(S^{i-1})$
    $(M, S'', b) = FBK(S', [0, 2, 3, 4, 6, 7])$
    for $j = 0$ to 30 **do**
      $S^i[j] = S''[j+1]$
    end for
    $M.con \leftarrow S''[0] = 0$
    $M.var \leftarrow S^i[31]$ as binary
    $M.con \leftarrow S^i[31] = a + b$
  end for
  $(M, S''', z) = KSG(S^R)$
  for $j = 0$ to 31 **do**
    $S'''[j] = 0$
  end for
  $M.con \leftarrow z = 1$
end function

MILP model for WG permutation (WGP-5)

```
WGP-5 = [ 0x0, 0x1, 0x1C, 0x4, 0x12, 0x10, 0x1F, 0x13,
0x1E, 0x3, 0x19, 0x15, 0x5, 0x16, 0x18, 0x8, 0xB, 0xF,
0x7, 0xE, 0x17, 0xA, 0xC, 0x6, 0xD, 0x2, 0x14, 0x1D, 0x1B,
0x11, 0x9, 0x1A ]
```

Modeling division trails of WGP-5

- Let $(x_0, x_1, x_2, x_3, x_4)$ and $(y_0, y_1, y_2, y_3, y_4)$ be the input and output of the WGP-5 Sbox, respectively.
- Reduce the #inequalities using inequality generator() function in Sage and Algorithms 1 and 2 in [XZBL][8].

[8] Xiang, Z., Zhang, W., Bao, Z., and Lin, D.: Applying milp method to searching integral distinguishers based on division property for 6 lightweight block ciphers. ASIACRYPT 2016

MILP model for WG permutation (WGP-5) (ctd.)

$$
\begin{cases}
2x_0 + 2x_1 + 2x_2 + 2x_3 + 6x_4 - 3y_0 - 3y_1 - 3y_2 - 3y_3 - 3y_4 \geq -1 \\
4x_3 - y_0 - y_1 - y_2 - y_3 - y_4 \geq -1 \\
4x_0 - y_0 - y_1 - y_2 - y_3 - y_4 \geq -1 \\
-x_0 - x_2 - x_3 - y_0 + 4y_1 - y_2 - y_3 - 2y_4 \geq -4 \\
-6x_0 - 3x_1 - 6x_3 - 6x_4 + 2y_0 - 4y_1 + 3y_2 - y_3 + 2y_4 \geq -19 \\
-3x_0 - x_1 - x_2 - 3x_3 - 2x_4 + 9y_0 + 7y_1 + 8y_2 + 9y_3 + 9y_4 \geq 0 \\
x_0 + x_1 + x_2 + x_3 + x_4 - 3y_0 - 3y_1 - 3y_2 - 3y_3 + 5y_4 \geq -2 \\
-x_0 - 3x_2 - 3x_3 - 2x_4 + y_0 + y_2 + y_3 - 2y_4 \geq -8 \\
-x_0 - x_1 + 2x_2 - x_3 - x_4 - y_0 - 2y_1 - 2y_2 + 3y_3 - y_4 \geq -5 \\
-x_0 - 2x_1 - 2x_2 - 2x_3 - x_4 - 2y_0 - y_1 - y_2 - y_3 + 5y_4 \geq -8 \\
-2x_0 - x_1 - 2x_2 - 2x_4 + y_0 + y_1 - y_2 + y_4 \geq -6 \\
-x_0 - x_2 - x_3 + y_0 - y_4 \geq -3.
\end{cases}
$$

MILP model for FBK and KSG

MILP model for FBK

- The feedback function is given by
  $S^i[0] \oplus S^i[2] \oplus S^i[3] \oplus S^i[4] \oplus S^i[6] \oplus S^i[7]$
- Model division property of bitwise XOR only.

MILP model for KSG

- The keystream bit at R-th round is given by
  $z = \mathtt{Tr}(\mathtt{WGP\text{-}5}(S^R[31])^3) = s^R_{155} + s^R_{156} + s^R_{157} + s^R_{158} + s^R_{159} +$
  $s^R_{155}s^R_{156} + s^R_{155}s^R_{157} + s^R_{155}s^R_{159} + s^R_{156}s^R_{158} + s^R_{156}s^R_{159} + s^R_{155}s^R_{156}s^R_{157} +$
  $s^R_{155}s^R_{157}s^R_{158} + s^R_{155}s^R_{157}s^R_{159} + s^R_{155}s^R_{158}s^R_{159} + s^R_{156}s^R_{157}s^R_{158} + s^R_{156}s^R_{158}s^R_{159}$
- Model division property of bitwise XOR and AND.

Number of MILP variables & constraints

Function	# of variables	# of constraints
WGP	15	17
FBK	65	35
KSG	79	63
R round of WG-5	160+159R + 5R	161 + 115R + 10R

MILP model to find involved secret variables in superpoly

1: **function** EXTRACTSECRETVARIABLES(MILP model M, Cube Indices I)
2: $M.var \leftarrow k_i$ as binary for $0 \leq i \leq n-1$, $\triangleright$ $k_0, k_1, \ldots, k_{n-1}$ are secret variables
3: $M.var \leftarrow v_i$ as binary for $0 \leq i \leq m-1$, $\triangleright$ $v_0, v_1, \ldots, v_{m-1}$ are public variables
4: $M.con \leftarrow v_i = 1$ for $i \in I$
5: $M.con \leftarrow v_i = 0$ for $i \in \{(0, 1, \ldots, m-1) - I\}$
6: $M.con \leftarrow \sum_{i=0}^{n-1} k_i = 1$
7: **do**
8: solve MILP model M
9: **if** M is feasible **then**
10: pick $j \in \{0, 1, \ldots, n-1\}$ s.t $k_j = 1$
11: $J = J \cup \{j\}$
12: $M.con \leftarrow k_j = 0$
13: **end if**
14: **while** M is feasible
15: return J
16: **end function**

- Step 4-6 sets the input initial division property.

Results

Rounds	Involved secret variables J	Time complexity $log_2(.)$
15	$\{k_5, k_6, \ldots, k_{54}\}$	54
16	$\{k_5, k_6, \ldots, k_{54}\}$	54
17	$\{k_5, k_6, \ldots, k_{59}\}$	59
18	$\{k_5, k_6, \ldots, k_{59}\}$	59
19	$\{k_5, k_6, \ldots, k_{64}\}$	64
20	$\{k_5, k_6, \ldots, k_{64}\}$	64
21	$\{k_5, k_6, \ldots, k_{69}\}$	69
22	$\{k_5, k_6, \ldots, k_{69}\}$	69
23	$\{k_5, k_6, \ldots, k_{74}\}$	74
24	$\{k_5, k_6, \ldots, k_{74}\}$	74

Table 1: Involved secret variables in superpoly for cube indices $I \in \{I_1, I_2, I_3, I_4, I_5\}$

$I_1 = \{0, 1, 2, 3\}, I_2 = \{0, 1, 2, 4\}, I_3 = \{0, 1, 3, 4\},$
$I_4 = \{0, 2, 3, 4\}, I_5 = \{1, 2, 3, 4\}$

Key recovery for 24 rounds

Key recovery procedre

1. Choose a value in the constant part of the IV and vary all $2^4 \times 2^{70}$ values to recover $p(k_5, k_6, \ldots, k_{74}, \bar{v})$ where $\bar{v} = (\{v_0, v_1, \ldots, v_{79}\} - \{v_j | \ j \in I_i\})$ for $1 \leq i \leq 5$ and $R = 24$.
2. Store 2^{70} values of $p(\bar{k}, \bar{v})$.
3. Query the cube C_{I_i} to the encryption oracle and compute the sum $\oplus_{C_{I_i}} f(k, v)$.
4. Compare above sum with values of p stored in the offline phase and discard the values of $\{k_5, k_6, \ldots, k_{74}\}$ for which the sum is different.

Data complexity: $5 \times 2^4 \approx 2^{6.32}$

Time complexity: $5 \times 2^{74} + 2^{75} \approx 2^{76.81}$

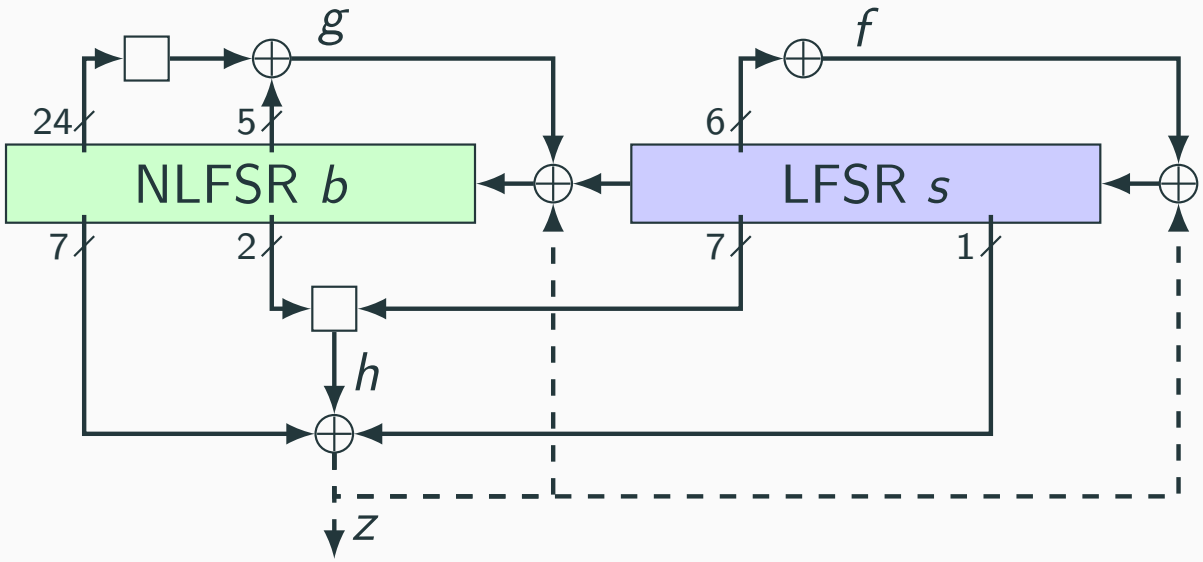
Attack comparison with algebraic attacks

- Existing algebraic attack[9] on WG-5 require data and time complexity 2^{15} and 2^{33}, resp.
- Not applicable if WGP-5 is feedback into the state during KSG phase.
- Our attack remains unaffected by feedback of WGP-5 during KSG phase.

[9] Ronjom, S. Improving algebraic attacks on stream ciphers based on linear feedback shift register over F_{2^K}. DCC 2017.

Comparison with Grain128a & Trivium

Grain128a



Key : 128-bit, IV : 96 bit, #initialization rounds : 256

Initial state : $(b_0, b_1, \ldots, b_{127}) = (k_0, k_1, \ldots, k_{127})$,
$(s_0, s_1, \ldots, s_{127}) = (iv_0, iv_1, \ldots, iv_{95}, 1, \ldots, 1, 0)$.

State update function:

$$
\begin{aligned}
g \leftarrow\ & b_0 + b_{26} + b_{56} + b_{91} + b_{96} + b_3 b_{67} + b_{11} b_{13} \\
& + b_{17} b_{18} + b_{27} b_{59} + b_{40} b_{48} + b_{61} b_{65} + b_{68} b_{84} \\
& + b_{88} b_{92} b_{93} b_{95} + b_{22} b_{24} b_{25} + b_{70} b_{78} b_{82}
\end{aligned}
$$

$$f \leftarrow s_0 + s_7 + s_{38} + s_{70} + s_{81} + s_{96}$$

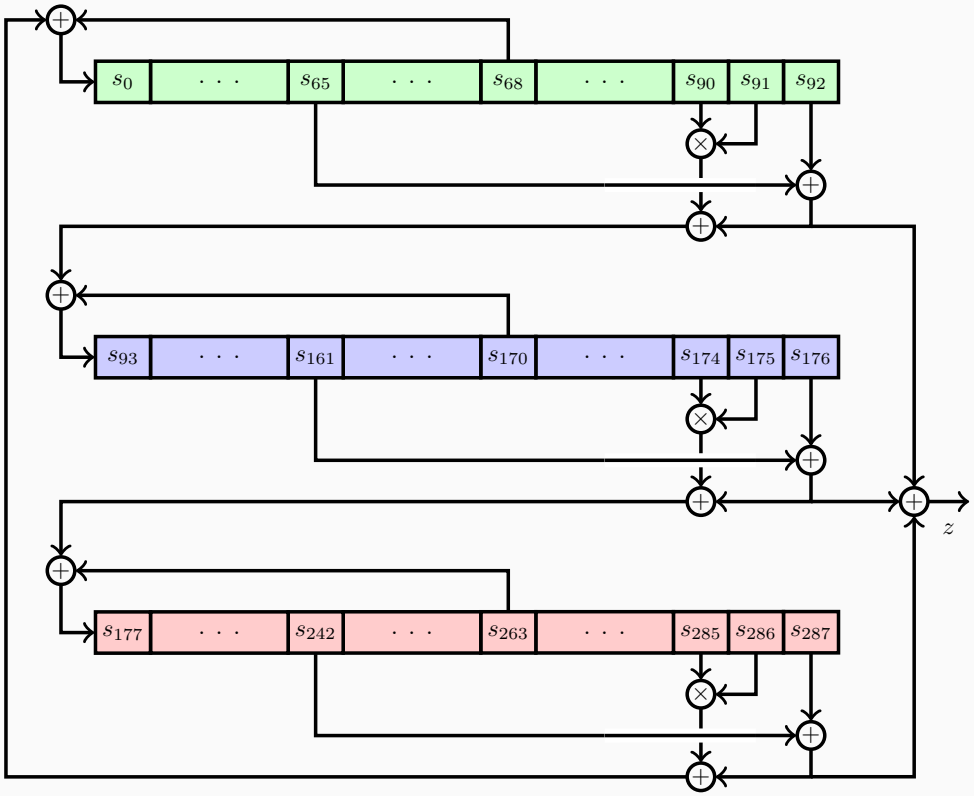
$$h \leftarrow b_{12} s_8 + s_{13} s_{20} + b_{95} s_{42} + s_{60} s_{79} + b_{12} b_{95} s_{94}$$

$$z \leftarrow h + s_{93} + b_2 + b_{15} + b_{36} + b_{45} + b_{64} + b_{73} + b_{89}$$

$$(b_0, b_1, \ldots, b_{127}) \leftarrow (b_1, b_2, \ldots, b_{127}, g + s_0 + z)$$

$$(s_0, s_1, \ldots, s_{127}) \leftarrow (s_1, s_2, \ldots, s_{127}, f + z)$$

Trivium



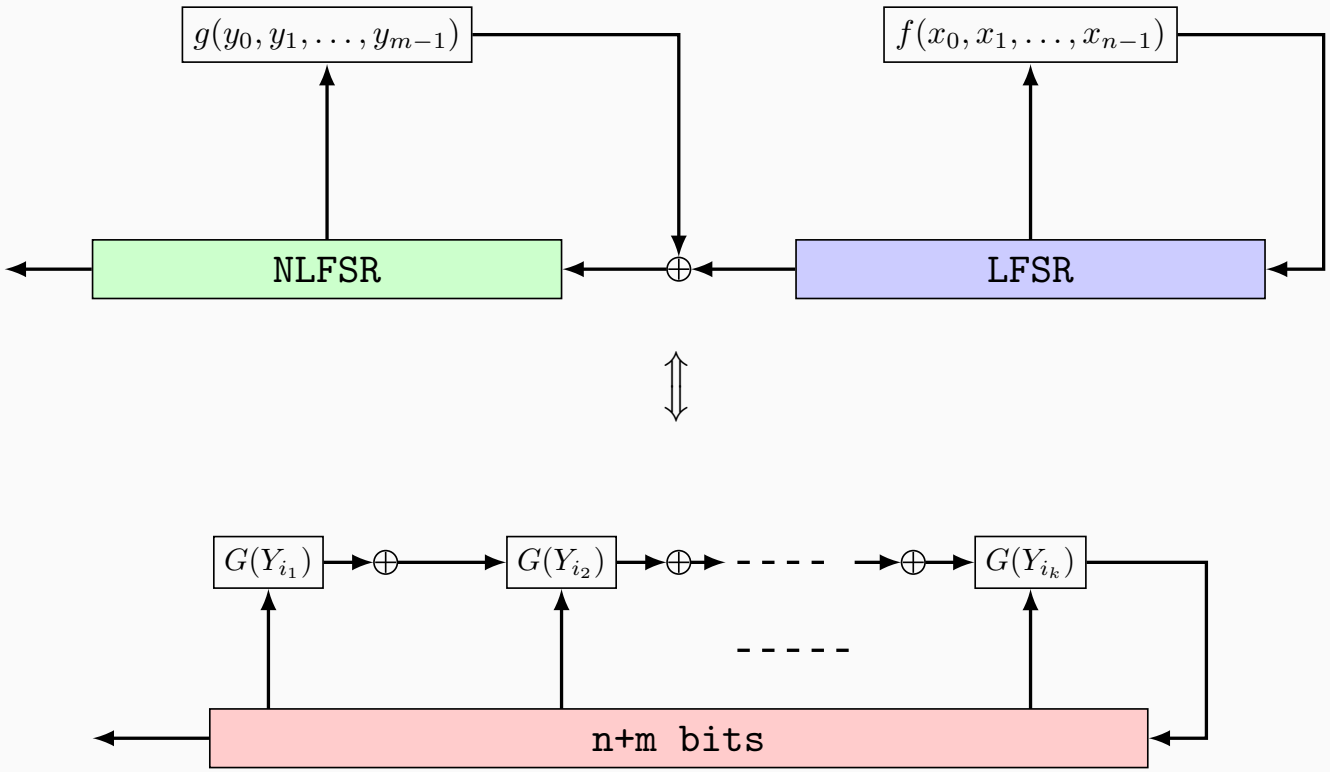
Key: 80 bit
IV: 80 bit
#initialization rounds: 1152
Initial state:

$$(s_0, s_1, \ldots, s_{92}) = (k_0, k_1, \ldots, k_{79}, 0, \ldots, 0)$$
$$(s_{93}, s_{94}, \ldots, s_{176}) = (iv_0, iv_1, \ldots, iv_{79}, 0, \ldots, 0)$$
$$(s_{177}, s_{178}, \ldots, s_{287}) = (0, 0, \ldots, 0, 1, 1, 1)$$

State update function:

$$t_1 \leftarrow s_{65} + s_{92}$$
$$t_2 \leftarrow s_{161} + s_{176}$$
$$t_3 \leftarrow s_{242} + s_{287}$$
$$z \leftarrow t_1 + t_2 + t_3$$
$$t_1 \leftarrow t_1 + s_{90}s_{91} + s_{170}$$
$$t_2 \leftarrow t_2 + s_{174}s_{175} + s_{263}$$
$$t_3 \leftarrow t_3 + s_{285}s_{286} + s_{68}$$
$$(s_0, s_1, \ldots, s_{92}) \leftarrow (t_3, s_0, \ldots, s_{91})$$
$$(s_{93}, s_1, \ldots, s_{176}) \leftarrow (t_1, s_{93}, \ldots, s_{175})$$
$$(s_{177}, s_1, \ldots, s_{287}) \leftarrow (t_2, s_{177}, \ldots, s_{286}).$$

Comparison of initialization phases



Comparison of initialization phases (cont.)

Observations on keystream bit

- For Trivium, the degree of z is 3 after 81 rounds.
- For Grain128a, the degree of z is 6 after 32 rounds.
- For WG-5 the degree of z is 6 after 1 round.
- Degree of WG-5 grows much faster than Grain128a and Trivium.

Comparison of initialization phases (cont.)

More observations

- For WG-5, 5 bits processed by WGP-5 at the i-th round are used to generate the keystream bit at round $(i+1)$ along with $5 \times 6 = 30$ new bits from the feedback function.
- For Grain128a, updated bits b_{127} and s_{127} in i-th round are used in keystream bit at $i+32$ and $i+33$, respectively.
- For Trivium, the values of t_1, t_2 and t_3 at i-th round are used in keystream bit at $i+90, i+81$ and $i+108$ rounds, respectively.

- Cube attack can cover more than half number of rounds for Grain128a (183/256) and Trivium (832/1152) ([Todo et al.][10]) compared to WG-5 (24/64).

[10] Todo, Y., Isobe, T., Hao, Y., and Meier, W. Cube attacks on non-blackbox polynomials based on division property. CRYPTO 2017

Conclusions

Conclusions

In this paper:

- we investigated the security of reduced-round WG-5 with respect to cube attacks.
- the attack require **data complexity:** $5 \times 2^4 \approx 2^{6.32}$ and **time complexity:** $5 \times 2^{74} + 2^{75} \approx 2^{76.81}$ for 24 rounds.
- we compared WG-5 initialization phase with that of Grain128a and Trivium and showed that WG-5 is more resistant to cube attacks.

Full paper can be found at:
http://cacr.uwaterloo.ca/techreports/2017/cacr2017-06.pdf

Thank you for your attention!

Communication Security (ComSec) Lab
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, N2L 3G1, CANADA
www.comsec.uwaterloo.ca