Correlation Cube Attack Revisited

Improved Cube Search and Superpoly Recovery Techniques

Jianhua Wang[1] Lu Qin[2,3] Baofeng Wu[4,5]

[1] Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

[2] China UnionPay Co., Ltd., Shanghai, China

[3] School of electronic information and electrical engineering, Shanghai Jiao Tong University, Shanghai, China

[4] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[5] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

Asiacrypt 2023

Cube attack[DS09]

An output bit of symmetric cipher could be written as a Boolean function of IV (plaintext) $\boldsymbol{x} \in \mathbb{F}_2^n$ and key $\boldsymbol{k} \in \mathbb{F}_2^m$. Given $I = \{i_0, \cdots, i_{d-1}\} \subset \{0, 1, \cdots, n-1\}$, one can write f as

$$f(\boldsymbol{x}, \boldsymbol{k}) = f_I(\boldsymbol{x}_{I^c}, \boldsymbol{k}) \cdot \boldsymbol{x}_I^{\mathbf{1}} + q_I(\boldsymbol{x}, \boldsymbol{k}).$$

Summing f over all 2^d possible values of $\boldsymbol{x}_I$, one has

$$\bigoplus_{C_I = \{\boldsymbol{x} | \boldsymbol{x}_I \in \mathbb{F}_2^d\}} f(\boldsymbol{x}, \boldsymbol{k}) = f_I(\boldsymbol{x}_{I^c}, \boldsymbol{k}).$$

Cube attack

Preprocessing phase: Recover the expressions of f_I for multiple I.

Online phase: Calculate the values of f_Is, and solve the system of equations about key.

Correlation cube attack[LYWL18]

Let $f_I(\boldsymbol{x}_J, \boldsymbol{k}) = \bigoplus_{i=1}^{r} h_i q_i$, and $Q_I = \{h_i\}_i$ is called the basis of f_I.

- **Preprocessing Phase**
  1. Obtain the basis Q_Is for f_Is.
  2. Add tuples (I, h_i, b) to Ω where $\Pr(h_i = b \mid f_I) > p$.
- **Online Phase**
  1. Randomly selects α values of $\boldsymbol{x}_J$, checks if f_I is zero constant
  2. Construct equations according to the element in Ω.

Motivation

Assume $f_I(\boldsymbol{x}_J, \boldsymbol{k}) = \bigoplus_{i=1}^{r} h_i q_i$.

The case of constructing an erroneous equation: (for a fixed key)

- $(I, h_i, 1) \in \Omega$: If $h_i = 0$, $\bigoplus_{j \neq i} h_j q_j = 1$ hold for certain values of $\boldsymbol{x}_J$.
- $(I, h_i, 0) \in \Omega$: If $h_i = 1$, $q_i = \bigoplus_{j \neq i} h_j q_j$ hold for all values of $\boldsymbol{x}_J$.

Note that the occurrence of the first case is possible only when $r > 1$.

strategies:

- Only use ”special” *ISoC* I that satisfy $f_I = hq$.
- Infer the value of h using multiple ”special” *ISoC* I_i that satisfy $f_{I_i} = hq_i$.

1. **Preprocessing phase**:
   a. Identify special *ISoC*s.
   b. For each h, let $T_h = \{I : h|f_I\}$.
   c. Let $\mathcal{T}_1 = \{T_h : \Pr(h = 0|\forall I \in T_h : f_I = 0) \leq p\}$.
   d. Let $\mathcal{T} = \{T_h : \Pr(h = 0|\forall I \in T_h : f_I = 0) > p\}$.

2. **Online phase**:
   a. Computes the value of f_I for each *ISoC* I.
   b. For every T_h in $\mathcal{T}$, make a guess on the value of h based on f_I's value for all I in T_h.
   c. For any T_h in $\mathcal{T}_1$, if $\exists I \in T_h$ satisfies $f_I = 1$, then $h = 1$. Otherwise, no guess is made for h.
   d. Store the equations $h = 1$ in to a set G_1, while store the other equations into a set G_0.
   e. Using these derived equations along with partial key guesses, we can try to obtain a candidate of the key.
      » If verifications for all partial key guesses do not yield a valid key, modify some equations from G_0 and solve again until a valid key is obtained.

Challenges

1. To acquire a significant number of special *ISoC*s.
   - Introduce a ”vector numeric mapping” technique.
   - Propose an algorithm for fast search of lots of good *ISoC*s.
2. To decompose a complicated Boolean polynomial.
   - Propose ”variable substitution” technique to recover superpolys.

Related work

- Search good *ISoC*.
  1. Numeric mapping technique [Liu17]
  2. Division property + heuristic algorithms [YT21, CT22]
- Recover superpolys.
  1. Linearity tests [DS09]
  2. Degree tests [FV14]
  3. Division property [TIHM17, WHT+18, WHG+19, HLM+20, HSWW20, HST+21, HHPW22]

Vector degree

Vector Degree

$$f(\boldsymbol{x}) = \bigoplus_{\boldsymbol{u}\in\mathbb{F}_2^d} g_{\boldsymbol{u}}(\boldsymbol{x}_{I^c})\boldsymbol{x}_I^{\boldsymbol{u}}$$

$$\mathbf{vdeg}_{[I,\boldsymbol{x}]}(f) = \deg(g_{\boldsymbol{u}_0}, g_{\boldsymbol{u}_1}, \ldots, g_{\boldsymbol{u}_{2^d-1}})_{\boldsymbol{x}_{I^c}} = \left(\deg(g_{\boldsymbol{u}_0})_{\boldsymbol{x}_{I^c}}, \ldots, \deg(g_{\boldsymbol{u}_{2^d-1}})_{\boldsymbol{x}_{I^c}}\right)$$

- $\deg(f) = \max_{0\le j<2^{|I|}}\{\mathbf{vdeg}_I(f)[j] + \mathrm{wt}(j)\}$.
- $\mathbf{vdeg}_{[I,\boldsymbol{x}]}(f) \preccurlyeq \boldsymbol{v} \quad\Rightarrow\quad \deg(f) \le \max_{0\le j<2^{|I|}}\{\min\{\boldsymbol{v}[j], n-|I|\} + \mathrm{wt}(j)\}$.
- If $I_1 \subset I_2$, $\quad \mathbf{vdeg}_{I_1}(f)[j] = \max_{0\le j'<2^{|I_2|-|I_1|}}\{\mathbf{vdeg}_{I_2}(f)[j'\cdot 2^{|I_1|} + j] + \mathrm{wt}(j')\}$.

Vector degree

Example

$$f = x_0 + x_0x_2 + x_1x_2x_3 + x_0x_1$$

- $I_2 = \{0,1\}$, $f = 0 \cdot 1 + (1 + x_2) \cdot x_0 + x_2x_3 \cdot x_1 + 1 \cdot x_0x_1$

$$\mathbf{vdeg}_{[I_2, \boldsymbol{x}]}(f) = [-\infty, 1, 2, 0] \;\Rightarrow\; \deg(f) = \max\{-\infty + 0, 1 + 1, 2 + 1, 0 + 2\} = 3$$

- $I_1 = \{0\}$, $f = x_1x_2x_3 \cdot 1 + (1 + x_1 + x_2) \cdot x_0$

$$\mathbf{vdeg}_{[I_1, \boldsymbol{x}]}(f) = [3, 1] \;\Rightarrow\; \deg(f) = \max\{3 + 0, 1 + 1\} = 3$$

- $\mathbf{vdeg}_{[I_1, \boldsymbol{x}]}(f)[0] = \max\{\mathbf{vdeg}_{[I_2, \boldsymbol{x}]}[0] + 0, \mathbf{vdeg}_{[I_2, \boldsymbol{x}]}[2] + 1\} = \max\{\infty + 0, 2 + 1\} = 3$
  $\mathbf{vdeg}_{[I_1, \boldsymbol{x}]}(f)[1] = \max\{\mathbf{vdeg}_{[I_2, \boldsymbol{x}]}[1] + 0, \mathbf{vdeg}_{[I_2, \boldsymbol{x}]}[3] + 1\} = \max\{1 + 0, 0 + 1\} = 1$

A new method for vector degree evaluation

Let $f : \mathbb{F}_2^n \to \mathbb{F}_2, \quad f = \bigoplus_{\boldsymbol{u}} a_{\boldsymbol{u}} \boldsymbol{y}^{\boldsymbol{u}}, \qquad \boldsymbol{g} : \mathbb{F}_2^m \to \mathbb{F}_2^n$

Vector numeric mapping

$$\mathtt{VDEG}_d : \quad \mathbb{B}_n \times \mathbb{Z}^{n \times 2^d} \to \mathbb{Z}^{2^d}$$
$$(f, V) \mapsto \boldsymbol{v}$$

where $v[j] = \max_{a_{\boldsymbol{u}} \neq 0} \max_{\substack{j_0, \cdots, j_{n-1} \\ 0 \le j_i \le u[i](2^d - 1) \\ j = \bigvee_{i=0}^{n-1} u[i] j_i}} \left\{ \sum_{i=0}^{n-1} u[i] V[i][j_i] \right\}$

Vector degree evaluation

$\mathbf{vdeg}_I(\boldsymbol{g}) \preccurlyeq V \quad \Rightarrow \quad \mathbf{vdeg}_I(f \circ \boldsymbol{g}) \preccurlyeq \mathtt{VDEG}_{|I|}(f, V)$

A new method for degree evaluation

Let $f(\boldsymbol{x}) = f_{r-1} \circ \boldsymbol{f}_{r-2} \circ \cdots \circ \boldsymbol{f}_0(\boldsymbol{x})$. We denoted the upper bound of the vector degree of f w.r.t. $\boldsymbol{x}$ and I by

$$\widehat{\mathbf{vdeg}}_{[I,\boldsymbol{x}]}(f) = \texttt{VDEG}(f_{r-1}, V_{r-2}),$$

where $V_i = \texttt{VDEG}(\boldsymbol{f}_i, V_{i-1})$, $0 < i \le r-2$, and $V_0 = \mathbf{vdeg}_{[I,\boldsymbol{x}]}(\boldsymbol{f}_0)$.

Mode 1. $\widehat{\mathbf{deg}}_{[I,\boldsymbol{x}]}(f) = \max_{0 \le j < 2^{|I|}} \{\min\{\widehat{\mathbf{vdeg}}_{[I,\boldsymbol{x}]}(f)[j], n - |I|\} + \mathrm{wt}(j)\}$.

Mode 2. $\widehat{\mathbf{deg}}_{[I,\boldsymbol{x}]}(f) = \widehat{\mathbf{vdeg}}_{[I,\boldsymbol{x}]}(f)[2^{|I|} - 1] + |I|$.

Mode 3. $\widehat{\mathbf{deg}}_{[I,\boldsymbol{x}]}(f) = \max_{0 \le j < 2^{|I|}} \{\widehat{\mathbf{vdeg}}_{[I,\boldsymbol{x}]}(f)[j] + \mathrm{wt}(j)\}$.

Degree evaluation [Mode 1]

$\mathbf{vdeg}(f) \preccurlyeq \widehat{\mathbf{vdeg}}_{[I,\boldsymbol{x}]}(f) \quad \Rightarrow \quad \deg(f) \le \widehat{\mathbf{deg}}_{[I,\boldsymbol{x}]}(f)$

Estimatation comparison between inclusion-based index set

$$I_1 \subset I_2 \quad \Rightarrow \quad \widehat{\mathbf{deg}}_{[I_2, \boldsymbol{x}]}(f) \le \widehat{\mathbf{deg}}_{[I_1, \boldsymbol{x}]}(f)$$

Example

Let $f = y_0 y_1$, $\boldsymbol{g} = [x_0 x_2 + x_1, x_0 x_1 + x_3]$. $\deg(f \circ \boldsymbol{g})$? $\quad (f \circ \boldsymbol{g} = x_0 x_1 + x_0 x_1 x_2 + x_0 x_2 x_3 + x_1 x_3)$

- $I_1 = \{1\}, V = \begin{bmatrix} \mathbf{vdeg}_{I_1}(g_0) \\ \mathbf{vdeg}_{I_1}(g_1) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$

  $\widehat{\mathbf{vdeg}}_{I_1}(f \circ \boldsymbol{g}) = [3, 3] \quad \Rightarrow$

  Mode 1. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 4$, Mode 2. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 4$, Mode 3. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 4$

- $I_2 = \{0, 1\}, V = \begin{bmatrix} \mathbf{vdeg}_{I_2}(g_0) \\ \mathbf{vdeg}_{I_2}(g_1) \end{bmatrix} = \begin{bmatrix} -\infty & 1 & 0 & -\infty \\ 1 & -\infty & -\infty & 0 \end{bmatrix}$

  $\widehat{\mathbf{vdeg}}_{I_2}(f \circ \boldsymbol{g}) = [-\infty, 2, 1, 1] \quad \Rightarrow$

  Mode 1. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 3$, Mode 2. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 3$, Mode 3. $\widehat{\mathbf{deg}}(f \circ \boldsymbol{g}) = 3$

Pruning technique

Theorem 5. [This work]

Let $J \subset K \subset I$. Then we have

$$\widehat{\mathbf{vdeg}}_{[J,\boldsymbol{x}_K]}(f|_{\boldsymbol{x}_{K^c}=0}) \preccurlyeq \widehat{\mathbf{vdeg}}_{[J,\boldsymbol{x}_I]}(f|_{\boldsymbol{x}_{I^c}=0}).$$

If $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_K]}(f|_{\boldsymbol{x}_{K^c}=0}) \geq d$, then $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_I]}(f|_{\boldsymbol{x}_{I^c}=0}) \geq d$ for all *ISoC*s I satisfying $K \subset I$.

- If *ISoC* I satisfies that $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_I]}(f|_{\boldsymbol{x}_{I^c}=0}) \geq d$, iteratively choose a series of *ISoC*s $I \supsetneq I_1 \supsetneq \cdots \supsetneq I_q \supset J$ such that $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_{I_i}]}(f|_{\boldsymbol{x}_{I_i^c}=0}) \geq d$ for all $1 \leq i \leq q$ and $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_{I'}]}(f|_{\boldsymbol{x}_{I'^c}=0}) < d$ for any $I' \subsetneq I_q$.
- Delete all the supersets of I_q.

Process of searching good *ISoC*s

Let J be a given index set, Ω be the set of all subsets of $[n]$ containing J and with size k, d be a threshold of degree, and a be the number of repeating times. The main steps are:

1. Prepare an empty set $\mathcal{I}$.
2. Select an element I from Ω as an *ISoC*.
3. Compute $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_I]}(f|_{\boldsymbol{x}_{I^c}=0})$;
   a. If $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}_I]}(f|_{\boldsymbol{x}_{I^c}=0}) < d$, then add I to $\mathcal{I}$ and goto Step 5;
   b. otherwise, set $count = 0$ and goto Step 4.
4. $count = count + 1$. Let $I' = I$, randomly remove an element $i \in I' \setminus J$ from I' and let $x_i = 0$. Compute $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}'_I]}(f|_{\boldsymbol{x}_{I'^c}=0})$.
   a. If $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}'_I]}(f|_{\boldsymbol{x}_{I'^c}=0}) < d$ and $count < a$, then goto Step 4;
   b. If $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}'_I]}(f|_{\boldsymbol{x}_{I'^c}=0}) < d$ and $count \geq a$, then goto Step 5;
   c. If $\widehat{\mathbf{deg}}_{[J,\boldsymbol{x}'_I]}(f|_{\boldsymbol{x}_{I'^c}=0}) \geq d$, then let $I = I'$ and goto Step 3.b;
5. Remove all the sets containing I from Ω. If $\Omega \neq \emptyset$, goto Step 2; otherwise, output $\mathcal{I}$.

MILP model for searching good *ISoC*s.

$$b_i = \begin{cases} 1, & i \in I \\ 0, & \text{otherwise} \end{cases}$$

- To describe that the size of each element of Ω is equal to k, we use

$$\sum_{i=0}^{n-1} b_i = k.$$

- To describe that each element of Ω includes the set J, we use

$$b_j = 1 \text{ for } \forall j \in J.$$

- To describe removing all the sets that contain I from Ω, we use

$$\sum_{i \in I} b_i < |I|.$$

callback function in Gurobi + degree evaluation

The framework of superpoly recovery[HST+21]

Let $f(\boldsymbol{x}, \boldsymbol{k}) = f_{r-1} \circ \boldsymbol{f}_{r-2} \circ \cdots \circ \boldsymbol{f}_0(\boldsymbol{x}, \boldsymbol{k})$ and denote the input and output of $\boldsymbol{f}_i$ by $\boldsymbol{y}_i$ and $\boldsymbol{y}_{i+1}$, respectively.

$$\texttt{Coe}(f, \boldsymbol{x}^{\boldsymbol{u}}) = \bigoplus_{\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m}) \in \texttt{VT}_{r_m}} \texttt{Coe}(\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m}), \boldsymbol{x}^{\boldsymbol{u}}).$$

The specific steps of recovering a superpoly requires two steps:

1. Try to obtain $\texttt{VT}_{r_m}$. If the model is solved within an acceptable time, goto Step 2.
2. For each term $\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m})$ in $\texttt{VT}_{r_m}$, compute $\texttt{Coe}(\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m}), \boldsymbol{x}^{\boldsymbol{u}})$ with our new techniques and sum them.

Let $f(\boldsymbol{x},\boldsymbol{k}) = f_{r-1} \circ \boldsymbol{f}_{r-2} \circ \cdots \circ \boldsymbol{f}_0(\boldsymbol{x},\boldsymbol{k})$ Let $\overleftarrow{\boldsymbol{f}}_{r_m}$ denote $\boldsymbol{f}_{r_m-1} \circ \cdots \circ \boldsymbol{f}_0$, i.e., $\boldsymbol{y}_{r_m} = \overleftarrow{\boldsymbol{f}}_{r_m}(\boldsymbol{x},\boldsymbol{k})$. Assume the algebraic normal form of $\overleftarrow{\boldsymbol{f}}_{r_m}$ in $\boldsymbol{x}$ is

$$\overleftarrow{\boldsymbol{f}}_{r_m} = \bigoplus_{\boldsymbol{v}\in\mathbb{F}_2^n} \boldsymbol{h}_{\boldsymbol{v}}(\boldsymbol{k})\boldsymbol{x}^{\boldsymbol{v}}.$$

Introduce new intermediates $\boldsymbol{z}$ to substitute these nonzero $\boldsymbol{h}_{\boldsymbol{v}}[j]$'s. From the ANF of $\overleftarrow{\boldsymbol{f}}_{r_m}$, it is natural to derive the new representation $\boldsymbol{g}_{r_m}$ such that $\boldsymbol{y}_{r_m} = \boldsymbol{g}_{r_m}(\boldsymbol{x},\boldsymbol{z})$, whose ANF in $\boldsymbol{x}$ and $\boldsymbol{z}$ can be written as

$$\boldsymbol{g}_{r_m}[j] = \bigoplus_{\boldsymbol{v}} a_{\boldsymbol{v},j}\boldsymbol{z}^{\boldsymbol{c}_{\boldsymbol{v},j}}\boldsymbol{x}^{\boldsymbol{v}}.$$

The process of recovering $\texttt{Coe}(\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m}),\boldsymbol{x}^{\boldsymbol{u}})$ is as follows:

1. Compute the ANF of $\boldsymbol{y}_{\boldsymbol{r_m}}$ in $\boldsymbol{x}$.
2. Substitute all different non-constant $\boldsymbol{h}_{\boldsymbol{v}}[j]$ for all $\boldsymbol{v}$ and j by new variables $\boldsymbol{z}$.
3. Recover $\texttt{Coe}(\pi_{\boldsymbol{u}_{r_m}}(\boldsymbol{y}_{r_m}),\boldsymbol{x}^{\boldsymbol{u}})$ in $\boldsymbol{z}$ by monomial prediction.

An example to illustrate the process

Example

Assume $\boldsymbol{y}_{r_m} = \overleftarrow{\boldsymbol{f}}_{r_m}(\boldsymbol{x}, \boldsymbol{k}) = [(k_0k_1 \oplus k_2k_5 \oplus k_9 + k_{10})x_0x_2 \oplus (k_3 \oplus k_6)x_5, (k_2k_7 \oplus k_8)x_3 \oplus x_6x_7]$.

Variable substitution: $k_0k_1 \oplus k_2k_5 \oplus k_9 + k_{10} \to z_0, \quad k_3 \oplus k_6 \to z_1, \quad k_2k_7 \oplus k_8 \to z_2$

$\Rightarrow \boldsymbol{y}_{r_m} = \boldsymbol{g}_{r_m}(\boldsymbol{x}, \boldsymbol{z}) = [z_0x_0x_2 \oplus z_1x_5, z_2x_3 \oplus x_6x_7]$.

- To compute $\texttt{Coe}(y_{r_m}[0]y_{r_m}[1], x_0x_2x_3)$, at least $4 * 2 = 8$ monomial trails $\boldsymbol{k}^{\boldsymbol{w}}x_0x_2x_3 \rightsquigarrow y_{r_m}[0]y_{r_m}[1]$ to form $(k_0k_1 \oplus k_2k_5 \oplus k_9 + k_{10})(k_2k_7 \oplus k_8)x_0x_2x_3$.
- After variable substitution, there remains only one trail $z_0z_2x_0x_2x_3$, which means we have consolidated 8 monomial trails into a single one.

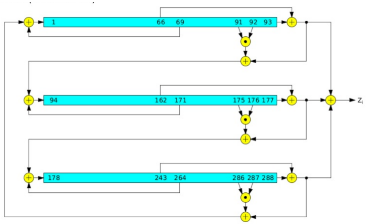
- **Reduce the number of monomial trails.**
- **Make the superpoly more concise and easy to factorize.**

Trivium stream cipher [De 06]

Padding:
$$(s_0, s_1, \cdots, s_{92}) \leftarrow (k_0, k_1, \cdots, k_{79}, 0, \cdots, 0)$$
$$(s_{93}, s_{94}, \cdots, s_{176}) \leftarrow (v_0, v_1, \cdots, v_{79}, 0, \cdots, 0)$$
$$(s_{177}, s_{178}, \cdots, s_{287}) \leftarrow (0, 0, \cdots, 0, 1, 1, 1).$$

Update:
$$s_{92} \leftarrow s_{65} \oplus s_{90} \cdot s_{91} \oplus s_{92} \oplus s_{170}$$
$$s_{176} \leftarrow s_{161} \oplus s_{174} \cdot s_{175} \oplus s_{176} \oplus s_{263}$$
$$s_{287} \leftarrow s_{242} \oplus s_{285} \cdot s_{286} \oplus s_{287} \oplus s_{68}$$

Output: $z = s_{65} \oplus s_{92} \oplus s_{161} \oplus s_{176} \oplus s_{242} \oplus s_{287}$



Structure diagram of Trivium stream cipher

Practical Key Recovery Attacks against 820-/825-/830- round Trivium

Parameter settings:

- Search *ISoC*s: Mode = 2;
  1. 820 rounds: $J = \{0, 1, 2, i, i+1\}$, where $3 \leq i \leq 26$; $\Omega = \{I \supset J : |I| = 38\}$; $d = 41$.
  2. 825 rounds: $J = \{0, 1, \cdots, 10\} \setminus \{j_0, j_1, j_2\}$, where $j_0 > 2, j_1 > j_0 + 1$ and $j_1 + 1 < j_2 < 11$; $\Omega = \{I \supset J : |I| = 41\}$; $d = 44$.
  3. 830 rounds: $J = \{0, 1, \cdots, 10\} \setminus \{j_0, j_1, j_2\}$, where $j_0 > 2, j_1 > j_0 + 1$ and $j_1 + 1 < j_2 < 11$; $\Omega = \{I \supset J : |I| = 41\}$; $d = 45$.
- Recover superpolys: $r_m = 200$.
- New correlation cube attack: $p = 0.77$

# of Rounds	size of *ISoC*	# of *ISoC*s	Total time	# of keys	Ref.
820	38	2^{13}	2^{52}	$2^{79.2}$	This work
820	38	2^{13}	2^{60}	$2^{79.8}$	This work
825	41	2^{12}	2^{54}	$2^{79.3}$	This work
825	41	2^{12}	2^{60}	$2^{79.7}$	This work
830	41	2^{13}	2^{55}	$2^{78.9}$	This work
830	41	2^{13}	2^{60}	$2^{79.4}$	This work

Conclusion

- We give a generalized definition of degree of Boolean function and give out a degree evaluation method with the vector numeric mapping technique.
- We introduce a pruning technique to fast filter the *ISoC*s and describe it into an MILP model to search automatically.
- Propose a variable substitution technique for cube attacks, which makes great improvement to the computational complexity of superpoly recovery and can provide more concise expression in new variables.
- We perform practical key recovery attacks on 820-, 825- and 830-round Trivium cipher, promoting up to 10 more rounds than previous best practical attacks as we know.

Thanks for your attention!

Reference I

Cheng Che and Tian Tian.
An experimentally verified attack on 820-round trivium.
In Yi Deng and Moti Yung, editors, *Information Security and Cryptology - 18th International Conference, Inscrypt 2022, Beijing, China, December 11-13, 2022, Revised Selected Papers*, volume 13837 of *Lecture Notes in Computer Science*, pages 357–369. Springer, 2022.

Christophe De Cannière.
Trivium: A stream cipher construction inspired by block cipher design principles.
pages 171–186, 2006.

Itai Dinur and Adi Shamir.
Cube attacks on tweakable black box polynomials.
pages 278–299, 2009.

Reference II

Pierre-Alain Fouque and Thomas Vannet.
Improving key recovery to 784 and 799 rounds of Trivium using optimized cube attacks.
pages 502–517, 2014.

Jiahui He, Kai Hu, Bart Preneel, and Meiqin Wang.
Stretching cube attacks: Improved methods to recover massive superpolies.
In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 537–566, Cham, 2022. Springer Nature Switzerland.

Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang.
Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128AEAD.
pages 466–495, 2020.

Reference III

Kai Hu, Siwei Sun, Yosuke Todo, Meiqin Wang, and Qingju Wang.
Massive superpoly recovery with nested monomial predictions.
pages 392–421, 2021.

Kai Hu, Siwei Sun, Meiqin Wang, and Qingju Wang.
An algebraic formulation of the division property: Revisiting degree evaluations, cube attacks, and key-independent sums.
pages 446–476, 2020.

Meicheng Liu.
Degree evaluation of NFSR-based cryptosystems.
pages 227–249, 2017.

Meicheng Liu, Jingchun Yang, Wenhao Wang, and Dongdai Lin.
Correlation cube attacks: From weak-key distinguisher to key recovery.
pages 715–744, 2018.

Reference IV

Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier.
Cube attacks on non-blackbox polynomials based on division property.
In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 250–279. Springer, Heidelberg, August 2017.

SenPeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi.
MILP-aided method of searching division property using three subsets and applications.
pages 398–427, 2019.

Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier.
Improved division property based cube attacks exploiting algebraic properties of superpoly.
pages 275–305, 2018.

Chen-Dong Ye and Tian Tian.
A practical key-recovery attack on 805-round trivium.
pages 187–213, 2021.