A Practical Key-Recovery Attack on 805-Round Trivium

Chen-Dong Ye Tian Tian

tiantian_d@126.com

ASIACRYPT 2021

Trivium

- It is a bit oriented synchronous stream cipher designed by Christophe De Cannière, and Bart Preneel.
- It was selected as one of the eSTREAM hardware-oriented finalists and it was specified as an International Standard under ISO/IEC 29192-3:2012.
- Both the key size $|K|$ and the IV size $|IV|$ are 80 bits.
- It contains a 288-bit internal state.
- The number of initialization rounds is 1152.

Trivium

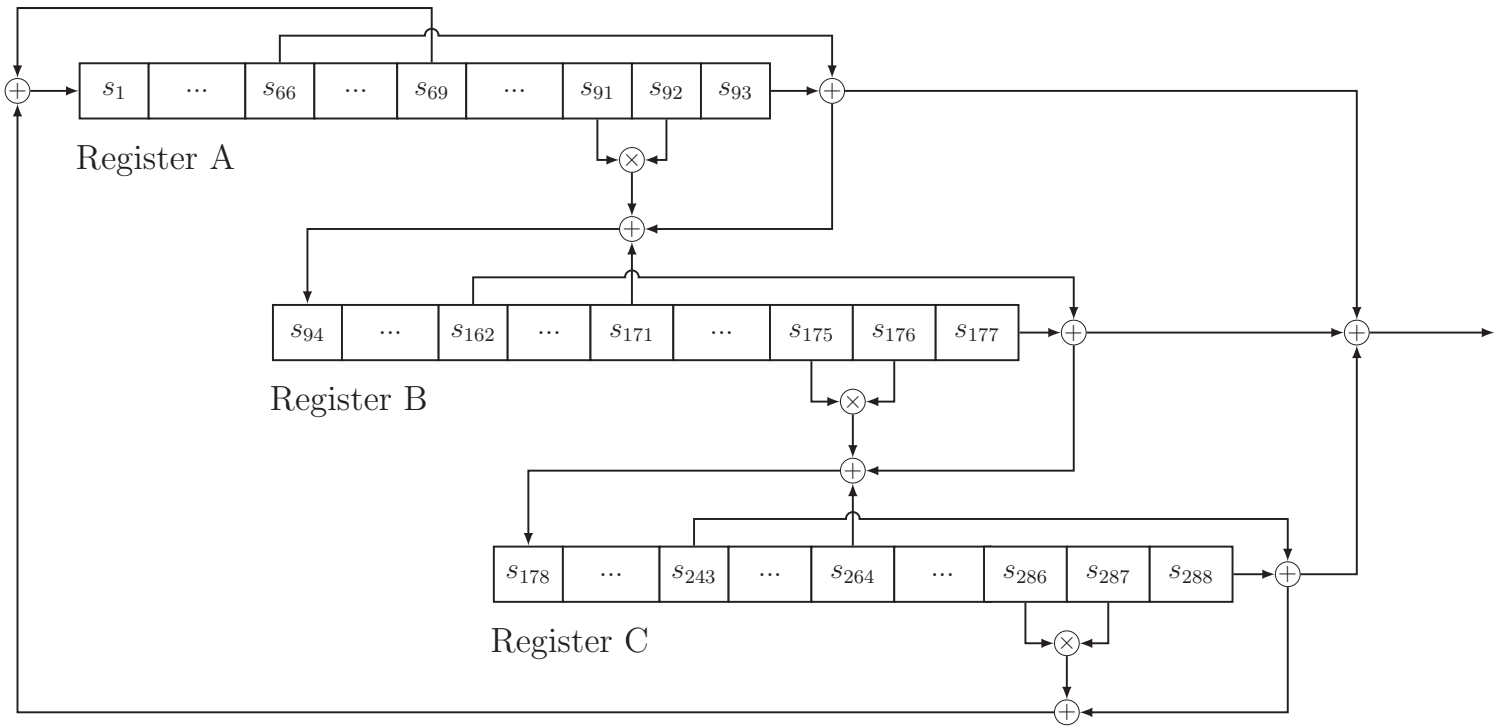


Figure: An overview of Trivium

Trivium

Algorithm 1 Key/IV initialization

$(s_1, s_2, \ldots, s_{93}) \leftarrow (k_0, k_1, \ldots, k_{79}, 0, \ldots, 0)$;
$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (v_0, v_1, \ldots, v_{79}, 0, \ldots, 0)$;
$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (0, \ldots, 0, 1, 1, 1)$;
for i from 1 to 1152 **do**
    $t_1 \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$;
    $t_2 \leftarrow s_{162} \oplus s_{175} \cdot s_{176} \oplus s_{177} \oplus s_{264}$;
    $t_3 \leftarrow s_{243} \oplus s_{286} \cdot s_{287} \oplus s_{288} \oplus s_{69}$;
    $(s_1, s_2, \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$;
    $(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$;
    $(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$;
end for

Cube Attacks

- (traditional) cube attack
  - I.Dinur and A. Shamir, Cube attacks on tweakable black box polynomials, EUROCRYPT 2009. [DS09]
  - P.A. Fouque and T. Vannet, Improving key recovery to 784 and 799 rounds of Trivium using optimized cube attacks, FSE 2013. [FV13]
  - ...
- division property based cube attack
  - Y. Todo, T. Isobe, Y.L. Hao, and W. Meier. Cube attacks on non-blackbox polynomials based on division property, CRYPTO 2017. [TIHM17a]
  - Q.J. Wang, Y.L. Hao, Y. Todo, C.Y Li, T. Isobe, and E. Meier. Improved division property based cube attacks exploiting algebraic properties of superpoly, CRYPTO 2018. [WHT+18]
  - ...
- correlation cube attack
  - M.C. Liu, J.C. Yang, W.H. Wang, D.D. Lin, Correlation cube attacks: From weak-key distinguisher to key recovery, EUROCRYPT 2018. [LYWL18]
  - ...

Basics of Cube Attacks

- Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable Boolean function.
- Let $I = \{i_1, i_2, \ldots, i_d\} \subseteq \{1, 2, \ldots, n\}$ be a subset of variable indexes.
- The Boolean function f could be uniquely written as

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

where $t_I = \prod_{i \in I} x_i$ and q is the sum of terms not divisible by t_I.

- The polynomial p_I is called the superpoly of I in f.

Basics of Cube Attacks

- Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable Boolean function.
- Let $I = \{i_1, i_2, \ldots, i_d\} \subseteq \{1, 2, \ldots, n\}$ be a subset of variable indexes.
- The Boolean function f could be uniquely written as

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

where $t_I = \prod_{i \in I} x_i$ and q is the sum of terms not divisible by t_I.

- The polynomial p_I is called the superpoly of I in f.

Basics of Cube Attacks

- Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable Boolean function.
- Let $I = \{i_1, i_2, \ldots, i_d\} \subseteq \{1, 2, \ldots, n\}$ be a subset of variable indexes.
- The Boolean function f could be uniquely written as

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

where $t_I = \prod_{i \in I} x_i$ and q is the sum of terms not divisible by t_I.

- The polynomial p_I is called the superpoly of I in f.

Basics of Cube Attacks

- Let $f(x_1, x_2, \cdots, x_n)$ be an n-variable Boolean function.
- Let $I = \{i_1, i_2, \ldots, i_d\} \subseteq \{1, 2, \ldots, n\}$ be a subset of variable indexes.
- The Boolean function f could be uniquely written as

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

  where $t_I = \prod_{i \in I} x_i$ and q is the sum of terms not divisible by t_I.
- The polynomial p_I is called the superpoly of I in f.

Basics of Cube Attacks

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

- Let $d = |I|$. A d-dimensional cube C_I is a set of assignments consisting of 2^d n-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other variables remain undetermined.
- Then

$$p_I(X') = \sum_{\mathbf{x} \in C_I} f(\mathbf{x}), X' = X \backslash I.$$

  - Note that to evaluate $p_I(X')$, 2^d summations are always needed.

Basics of Cube Attacks

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

- Let $d = |I|$. A d-dimensional cube C_I is a set of assignments consisting of 2^d n-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other variables remain undetermined.
- Then

$$p_I(X') = \sum_{\mathbf{x} \in C_I} f(\mathbf{x}), X' = X \backslash I.$$

  - Note that to evaluate $p_I(X')$, 2^d summations are always needed.

Cube Attacks against Stream Ciphers

- The function $f(k_1, k_2, \ldots, k_n, v_1, v_2, \ldots, v_m)$ is the polynomial representation of the first output keystream bit on key and IV variables.
- Let $I \subseteq \{1, 2, \ldots, m\}$ be a subset of IV indexes and $d = |I|$.
- Let C_I be a set of assignments for IV variables consisting of 2^d m-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other IV variables are assigned to 0 (or fixed constants).
- Then

$$p_I(key) = \sum_{IV \in C_I} f(key, IV).$$

Cube Attacks against Stream Ciphers

- The function $f(k_1, k_2, \ldots, k_n, v_1, v_2, \ldots, v_m)$ is the polynomial representation of the first output keystream bit on key and IV variables.
- Let $I \subseteq \{1, 2, \ldots, m\}$ be a subset of IV indexes and $d = |I|$.
- Let C_I be a set of assignments for IV variables consisting of 2^d m-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other IV variables are assigned to 0 (or fixed constants).
- Then

$$p_I(key) = \sum_{IV \in C_I} f(key, IV).$$

Cube Attacks against Stream Ciphers

- The function $f(k_1, k_2, \ldots, k_n, v_1, v_2, \ldots, v_m)$ is the polynomial representation of the first output keystream bit on key and IV variables.
- Let $I \subseteq \{1, 2, \ldots, m\}$ be a subset of IV indexes and $d = |I|$.
- Let C_I be a set of assignments for IV variables consisting of 2^d m-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other IV variables are assigned to 0 (or fixed constants).
- Then

$$p_I(key) = \sum_{IV \in C_I} f(key, IV).$$

Cube Attacks against Stream Ciphers

- The function $f(k_1, k_2, \ldots, k_n, v_1, v_2, \ldots, v_m)$ is the polynomial representation of the first output keystream bit on key and IV variables.
- Let $I \subseteq \{1, 2, \ldots, m\}$ be a subset of IV indexes and $d = |I|$.
- Let C_I be a set of assignments for IV variables consisting of 2^d m-tuples in which the variables indexed by I are assigned to all the possible combinations of $0/1$ while all the other IV variables are assigned to 0 (or fixed constants).
- Then

$$p_I(key) = \sum_{IV \in C_I} f(key, IV).$$

Cube Attacks against Stream Ciphers

- Off-line phase
  - independent of the secret key
  - find some useful superpolies in secret key variables
- on-line phase
  - solve a system of equations defined by superpolies under the key

$$\begin{cases} p_{I_1}(key) = a_1 \\ p_{I_2}(key) = a_2 \\ \quad \cdots \\ p_{I_u}(key) = a_u \end{cases}$$

Basics in Cube Attacks

$$f = t_I \cdot p_I \oplus q(x_1, x_2, \cdots, x_n),$$

- Using Low-Degree tests to find simple/low-degree/balanced superpolies p_I containing secrete key information.

Low-Degree Tests

- BLR Linearity Tests: Finding linear superpolies.
- Quadraticity Tests: Finding quadratic superpolies.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Using Möbius Transformation [FV13]

- In FSE 2013, Fouque and Vannet used **Möbius Transformation** to simultaneously compute a large number of subcubes from a large cube.
- Möbius Transformation: Given the truth table of a Boolean function, compute its algebraic normal form.
  - store $\{f(key, IV), IV \in C_I\}$
  - compute $\sum_{J \subseteq I} p_J(key) \prod_{j \in J} x_j$

Remark: Much memory is needed to store the truth table for a large cube.

Division property based cube attacks

- The division property was proposed by Todo at EUROCRYPT 2015 [Tod15] as a generalization of integral property used in integral cryptanalysis against block ciphers.
- The bit-based division property as well as the three-subset division property was introduced by Todo and Morii at FSE 2016 [TM16] to search new integral distinguishers for SIMON family.

Division property based cube attacks

Division property based cube attacks were first proposed by Yosuke Todo et al. at CRYPTO 2017 [TIHM17a, TIHM17b].

- Limitation of traditional cube attacks: the cube dimension could hardly be larger than 40.
- Advantage of division property based cube attacks: large cubes could be exploited (for example, 78-dimensional cubes).

Division property based cube attacks

Definition (Bit-Based Division Property)

Let $\mathbb{X}$ be a multiset whose elements take a value of $\mathbb{F}_2^n$. Let $\mathbb{K}$ be a subset of $\mathbb{F}_2^n$. When the multiset $\mathbb{X}$ has the division property $\mathcal{D}_{\mathbb{K}}^{1^n}$, it fulfills the following conditions:

$$\bigoplus_{\mathbf{x}\in\mathbb{X}} \mathbf{x}^{\mathbf{u}} = \begin{cases} \text{unknown} & \text{if there exists } \mathbf{k} \text{ in } \mathbb{K} \text{ s.t. } \mathbf{u} \succeq \mathbf{k}, \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathbf{u} \succeq \mathbf{k}$ if $u_i \geq k_i$ for all i and $\mathbf{x}^{\mathbf{u}} = \prod_{i=1}^{n} x_i^{u_i}$.

Division property based cube attacks

- Propagation rules of division property for basic operations (AND, XOR and COPY) are studied and proved in [Todo15, TM16].
- The propagation of division property could be evaluated as

$$\{\boldsymbol{k}\} = \mathbb{K}_0 \to \mathbb{K}_1 \to \cdots \to \mathbb{K}_r,$$

  where $\mathbb{K}_i$ is the division property of the internal state after i rounds.
- For $(\boldsymbol{k}_0, \boldsymbol{k}_1, \ldots, \boldsymbol{k}_r) \in \mathbb{K}_0 \times \mathbb{K}_1 \times \cdots \mathbb{K}_r$, if $\mathbf{k}_i$ could propagate to $\mathbf{k}_{i+1}$ for all $i \in \{0, 1, \ldots, r-1\}$, then

$$\boldsymbol{k}_0 \to \boldsymbol{k}_1 \to \cdots \to \boldsymbol{k}_r$$

  is called an r-round division trail.

Division property based cube attacks

- At ASIACRYPT 2016, Xiang et al. showed that the propagation could be efficiently evaluated by using MILP (Mixed Integer Linear Program) models [XZBL16].
- An MILP model covers all division trails, and the solver evaluates the feasibility whether there are division trails from the input division property to the output one or not.

Division property based cube attacks

- At ASIACRYPT 2016, Xiang et al. showed that the propagation could be efficiently evaluated by using MILP (Mixed Integer Linear Program) models [XZBL16].
- An MILP model covers all division trails, and the solver evaluates the feasibility whether there are division trails from the input division property to the output one or not.

Degree evaluation method

Proposition ([WHT+18])

Let $f(\boldsymbol{x}, \boldsymbol{v})$ be a polynomial, where $\boldsymbol{x}$ and $\boldsymbol{v}$ denote the secret and public variables, respectively. For a set of indices $I = \{i_1, i_2, \ldots, i_{|I|}\} \subset \{1, 2, \ldots, m\}$, let C_I be a set of $2^{|I|}$ values where the variables in $\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|I|}}\}$ are taking all possible combinations of values. Let $\boldsymbol{k}_I$ be an m-dimensional bit vector such that $\boldsymbol{v}^{\boldsymbol{I}} = t_I = v_{i_1} v_{i_2} \cdots v_{i_{|I|}}$. Let $\boldsymbol{k}_\Lambda$ be an n-dimensional bit vector. If there is no division trail such that $(\boldsymbol{k}_\Lambda || \boldsymbol{k}_I) \xrightarrow{f} 1$, then the monomial $x^{\boldsymbol{k}_\Lambda}$ is not involved in the superpoly of the cube C_I.

- If there is $d \geq 0$ such that for all $\boldsymbol{k}_\Lambda$ of Hamming Weight $hw(\boldsymbol{k}_\Lambda) > d$, the division trail $x^{\boldsymbol{k}_\Lambda}$ does not exist, then it can be seen that d is an upper bound of the algebraic degree of the superpoly.

Previous result on contructing good cubes

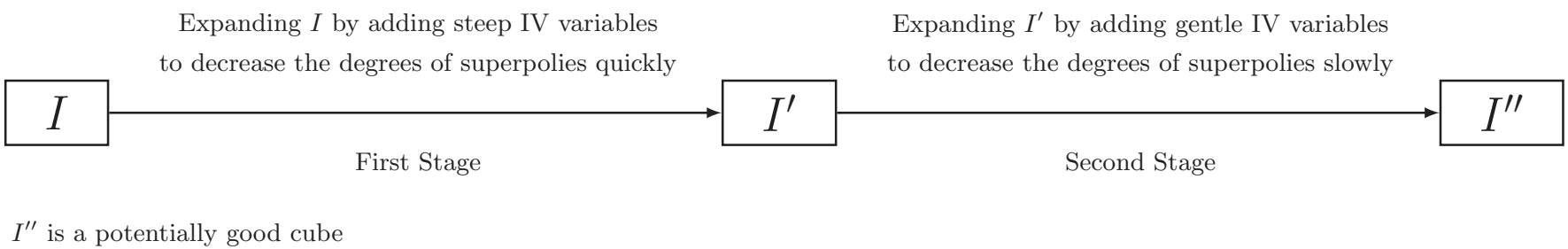
- a random walk algorithm [DS09] linear superpolies
- the union of two disjoint small cubes [FV13] linear superpolies
- cubes with no adjacent indexes [Liu17] zero-sum distinguishers
- a GreedyBitSet algorithm [SMB17, KRSM20] zero-sum distinguishers

Our aim and idea

- Our Aim: construct good cubes for recovering linear superpolies.
- Our Idea: Start from a set of cube variables indexed by I. We extend I to be a large candidate cube indexes.
  - How to determine a proper set of cube variables indexed by I?
  - How to extend I to be a large candidate cube indexes?

An Overview of How to Extend I

We propose a two-stage method to extend I.



- Steep IV Variable: an IV variable to I which makes the degree of the superpoly decrease as fast as possible.
- Gentle IV Variable: an IV variable of I which decreases the degree of the superpoly as slowly as possible.

A Heuristic Algorithm of Constructing Cubes

- Stage I: Every time, add a steep IV variable to decrease the degree of superpoly as fast as possible.

/* The first stage */
while $ds > 1$ and $|I|$ is less than a given bound **do**
  for $v \in B$ **do**
    Estimate the upper bound of $ds(I \cup \{v\})$ using the division property based method;
  end for
  $I \leftarrow I \cup \{v\}$, where v is the first steep IV variable of I;
  $B \leftarrow B \setminus v$;
  $ds \leftarrow DS(I \cup \{v\})$, where $DS(I \cup \{v\})$ is the upper bound of $ds(I \cup \{v\})$
end while
if $ds(I) == 1$ **then**
  return I
end if

A Heuristic Algorithm of Constructing Cubes

- Stage II: add a gentle IV variable to decrease the degree of the superpoly gentlely.

/* The second stage */
if $ds(I) == 0$ **then**
  $I \leftarrow I \setminus \{v\}$, where v is the steep IV variable added in the last iteration of the first stage.
  $I \leftarrow I \cup \{v'\}$, where $DS(I \cup \{v'\})$ attains minimum except 0 in the last iteration of the first stage.
  $B \leftarrow \{v_0, v_1, \ldots, v_{m-1}\} \setminus I$;
  while $ds > 1$ and $|I|$ is less than a given bound **do**
    for $v \in B$ **do**
      Estimate the upper bound of $ds(I \cup \{v\})$ using the division property based method;
    end for
    $I \leftarrow I \cup \{v\}$, where v is the first gentle IV variable
    $B \leftarrow B \setminus v$;
    $ds \leftarrow DS(I \cup \{v\})$
  end while
end if

The start cube set

- $z = s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$
- If a cube C has linear superpoly, then it is likely only one of the six terms contributes the linear superpoly and the other five terms contribute constant 0.
- Take s_{66}^t for example:

$$s_{66}^t = s_{243}^{t-66} \oplus s_{286}^{t-66} \cdot s_{287}^{t-66} \oplus s_{288}^{t-66} \oplus s_{69}^{t-66}.$$

- Choose a set of cube variables indexed by I and search some of its subcubes to find a cube with linear superpolies either in s_{286}^{t-66} or in s_{287}^{t-66}. Such cube is a starting cube set.

The start cube set

- $z = s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$
- If a cube C has linear superpoly, then it is likely only one of the six terms contributes the linear superpoly and the other five terms contribute constant 0.
- Take s_{66}^t for example:

$$s_{66}^t = s_{243}^{t-66} \oplus s_{286}^{t-66} \cdot s_{287}^{t-66} \oplus s_{288}^{t-66} \oplus s_{69}^{t-66}.$$

- Choose a set of cube variables indexed by I and search some of its subcubes to find a cube with linear superpolies either in s_{286}^{t-66} or in s_{287}^{t-66}. Such cube is a starting cube set.

The start cube set

- $z = s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$
- If a cube C has linear superpoly, then it is likely only one of the six terms contributes the linear superpoly and the other five terms contribute constant 0.
- Take s_{66}^t for example:

$$s_{66}^t = s_{243}^{t-66} \oplus s_{286}^{t-66} \cdot s_{287}^{t-66} \oplus s_{288}^{t-66} \oplus s_{69}^{t-66}.$$

- Choose a set of cube variables indexed by I and search some of its subcubes to find a cube with linear superpolies either in s_{286}^{t-66} or in s_{287}^{t-66}. Such cube is a starting cube set.

Predict the Preference Bit

$$z^r = s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$$
$$\Rightarrow \quad p_I = p_I^{66} \oplus p_I^{93} \oplus p_I^{162} \oplus p_I^{177} \oplus p_I^{243} \oplus p_I^{288}$$

- Note that a linear superpoly usually only has one term, say $p_I = k_j$, one of the six superpolies $p_I^{66}, p_I^{93}, p_I^{162}, p_I^{177}, p_I^{243}, p_I^{288}$ is equal to k_j.
- For a given r, we would like to predict

$$k \in \{66, 93, 162, 177, 243, 288\}$$

with a large success probability such that a linear superpoly in z^r implies a linear superpoly in s_k^r(Preference Bit).

Predict the Preference Bit

$$z^r = s_{66} \oplus s_{93} \oplus s_{162} \oplus s_{177} \oplus s_{243} \oplus s_{288}$$
$$\Rightarrow \quad p_I = p_I^{66} \oplus p_I^{93} \oplus p_I^{162} \oplus p_I^{177} \oplus p_I^{243} \oplus p_I^{288}$$

- Note that a linear superpoly usually only has one term, say $p_I = k_j$, one of the six superpolies $p_I^{66}, p_I^{93}, p_I^{162}, p_I^{177}, p_I^{243}, p_I^{288}$ is equal to k_j.
- For a given r, we would like to predict

$$k \in \{66, 93, 162, 177, 243, 288\}$$

with a large success probability such that a linear superpoly in z^r implies a linear superpoly in s_k^r(Preference Bit).

Predict the Preference Bit

Our Idea

Predicting the numbers of terms in form of $t_I \cdot k_j$ (called VK-terms) in $s_{66}, s_{93}, s_{162}, s_{177}, s_{243}, s_{288}$.

- It is difficult to predict the number of called VK-terms in $s_{66}, s_{93}, s_{162}, s_{177}, s_{243}, s_{288}$ directly.

Our Solution

Exploiting the feedback function of Trivium to predict the number of called VK-terms iteratively.

Predict the Preference Bit

$$s_{94} \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$$

- The number of terms of the form $t_I \cdot k_j$ in s is denoted by $VK(s)$.
- The number of pure IV terms in s is denoted by $V(s)$.
- Then

$$VK(s_{94}) \leftarrow VK(s_{66}) + VK(s_{91}s_{92}) + VK(s_{93}) + VK(s_{171})$$

Predict the Preference Bit

$$s_{94} \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$$

- The number of terms of the form $t_I \cdot k_j$ in s is denoted by $VK(s)$.
- The number of pure IV terms in s is denoted by $V(s)$.
- Then

$$VK(s_{94}) \leftarrow VK(s_{66}) + VK(s_{91}s_{92}) + VK(s_{93}) + VK(s_{171})$$

Predict the Preference Bit

$$s_{94} \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$$

- The number of terms of the form $t_I \cdot k_j$ in s is denoted by $VK(s)$.
- The number of pure IV terms in s is denoted by $V(s)$.
- Then

$$VK(s_{94}) \leftarrow VK(s_{66}) + VK(s_{91}s_{92}) + VK(s_{93}) + VK(s_{171})$$

Predict the Preference Bit

$$s_{94} \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$$

- Then

$$VK(s_{91}s_{92}) \leftarrow V(s_{91}) \cdot VK(s_{92}) + V(s_{92}) \cdot VK(s_{91}).$$

Predict the Preference Bit

$$s_{94} \leftarrow s_{66} \oplus s_{91} \cdot s_{92} \oplus s_{93} \oplus s_{171}$$

- The number of terms of the form $t_I \cdot k_j$ in s is denoted by $VK(s)$.
- The number of pure IV terms in s is denoted by $V(s)$.
- Then $VK(s_{94})$ is set as

$$\begin{aligned} VK(s_{66}) \quad &+ \quad VK(s_{93}) + VK(s_{171}) \\ &+ \quad V(s_{91}) \cdot VK(s_{92}) + V(s_{92}) \cdot VK(s_{91}) \end{aligned}$$

- Base on the above method, we propose Algorithm 4 which could iteratively predict the number of VK-terms $s_{66}^{(r)}, s_{93}^{(r)}, s_{162}^{(r)}, s_{177}^{(r)}, s_{243}^{(r)}, s_{288}^{(r)}$. Thus, the preference bit could be predicted successfully.

Algorithm 4 The algorithm of predicting the preference bit

Calculate the ANFs of $s_i^{(280)}$ to initialise $NVK^{(280)}$ and $NV^{(280)}$;
for $280 \leq t \leq r-1$ **do**
  $NVK_{t_1} \leftarrow NV_{91}^{(t)} \cdot NVK_{92}^{(t)} + NV_{92}^{(t)} \cdot NVK_{91}^{(t)} + NVK_{93}^{(t)} + NVK_{66}^{(t)} + NVK_{171}^{(t)}$;
  $NV_{t_1} \leftarrow NV_{91}^{(t)} \cdot NV_{92}^{(t)} + NV_{93}^{(t)} + NV_{66}^{(t)} + NV_{171}^{(t)}$;
  $NVK_{t_2} \leftarrow NV_{175}^{(t)} \cdot NVK_{176}^{(t)} + NV_{176}^{(t)} \cdot NVK_{175}^{(t)} + NVK_{177}^{(t)} + NVK_{162}^{(t)} + NVK_{264}^{(t)}$;
  $NV_{t_2} \leftarrow NV_{175}^{(t)} \cdot NV_{176}^{(t)} + NV_{177}^{(t)} + NV_{162}^{(t)} + NV_{264}^{(t)}$;
  $NVK_{t_3} \leftarrow NV_{286}^{(t)} \cdot NVK_{287}^{(t)} + NV_{287}^{(t)} \cdot NVK_{286}^{(t)} + NVK_{288}^{(t)} + NVK_{243}^{(t)} + NVK_{69}^{(t)}$;
  $NV_{t_3} \leftarrow NV_{286}^{(t)} \cdot NV_{287}^{(t)} + NV_{288}^{(t)} + NV_{243}^{(t)} + NV_{69}^{(t)}$;
  for $288 \geq j \geq 2$ **do**
    $NVK_j^{(t)} \leftarrow NVK_{j-1}^{(t)}$;
    $NV_j^{(t)} \leftarrow NV_{j-1}^{(t)}$;
  end for
  $NV_{94}^{(t)} \leftarrow NV_{t_1}$; $NV_{178}^{(t)} \leftarrow NV_{t_2}$; $NV_1^{(t)} \leftarrow NV_{t_3}$;
  $NVK_{94}^{(t)} \leftarrow NVK_{t_1}$; $NVK_{178}^{(t)} \leftarrow NVK_{t_2}$; $NVK_1^{(t)} \leftarrow NVK_{t_3}$;
end for
Choose the bit $s_b^{(t)}$ such that

$$NVK_b^{(t)} = \max\{NVK_\lambda^{(t)} | \lambda \in \{66, 93, 162, 171, 243, 288\}\}$$

as the preference bit, where $b \in \{66, 93, 162, 171, 243, 288\}$;

Improved Möbius Transformation

- Advantage: compute a large number of subcubes from a large cube simultaneously.
- Weakness: Much memory is needed to store the truth table for a large cube.

Our Aim

Testing a large number of subcubes from a large cube simultaneously with reasonable memory.

Improved Möbius Transformation

- Let $f(x_0, \ldots, x_{n-1})$ be a Boolean function on $x_0, \ldots, x_{n-1}$.

Our Idea

Breaking the original Möbius transformation into a two-stage one to recover a part of ANF of f.

- Calculate the Möbius transformations of $f(x_0, \ldots, x_{n-q-1}, 0, \ldots, 0)$, $f(x_0, \ldots, x_{n-q-1}, 1, \ldots, 0)$, $\ldots, f(x_0, \ldots, x_{n-q-1}, 1, \ldots, 1)$, which are denoted by $g_0, g_1, \ldots, g_{2^q-1}$. Keep a part information of $g_0, g_1, \ldots, g_{2^q-1}$.
- Recover a part of ANF of f according to the kept information of $g_0, g_1, \ldots, g_{2^q-1}$.

Improved Möbius Transformation

Algorithm 5 An Improved Möbius Transformation

Require: A Boolean function f, the parameter q, the bound ω

/* the first stage */

for $(c_0, c_1, \ldots, c_{q-1})$ from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$ **do**
  $S \leftarrow$ the truth table of $f(x_0, x_1, \ldots, x_{n-q-1}, c_0, c_1, \ldots, c_{q-1})$;
  Call Algorithm 2 to do Möbius transformation on S;
  $t \leftarrow 0$, $j \leftarrow \sum_{l=0}^{q-1} 2^l c_l$;
  for i from 0 to $2^{n-q} - 1$ **do**
    $tmp \leftarrow (b_0, b_1, \ldots, b_{n-q-1})$, where $i = \sum_{l=0}^{n-q-1} b_l \cdot 2^l$;
    if $wt(tmp) \geq \omega$ **then**
      $FS[j][t] \leftarrow S[i]$;
      $t \leftarrow t + 1$;
    end if
  end for
end for

/* the second stage */

for i from 1 to q **do**
  $Sz \leftarrow 2^i$, $Pos \leftarrow 1$;
  while $Pos < 2^q$ **do**
    for b from 0 to $Sz - 1$ **do**
      for a from 0 to $t - 1$ **do**
        $FS[Pos + Sz + b][a] \leftarrow FS[Pos + Sz + b][a] \oplus FS[Pos + b][a]$;
      end for
    end for
    $Pos \leftarrow Pos + 2 \times Sz$;
  end while
end for

Memory Complexity of Improved Möbius Transformation

- Memory Complexity of Algorithm 5.
  - The size of S is 2^{n-q}, and so it costs 2^{n-q} bits memory.
  - For each j, the size of $FS[j]$ is t, and so it requires $2^q \times t$ bits memory totally.

- Memory Complexity Comparison
  - Improved Möbius transformatio: $2^q \times t + 2^{n-q}$.
  - Original Möbius transformatio: 2^n.

A Practical Key-Recovery Attack on 805-Round Trivium

- The preference bit to find a linear superpoly is $s_{66}^{(805)}$.
- $s_{66}^{(805)} = s_{286}^{(739)} \cdot_{287}^{(739)} \oplus s_{243}^{(739)} \oplus s_{288}^{(739)} \oplus s_{69}^{(739)}$
- Choose cubes of sizes 22 and use the Moebius transformation to find a proper cube whose superpoly in $s_{286}^{(739)}$ is linear.
- Start from

$$I_1 = \{v_4, v_6, v_{10}, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, v_{25}, v_{29}, v_{32},$$

$$v_{34}, v_{36}, v_{39}, v_{41}, v_{43}, v_{50}\}$$

A Practical Key-Recovery Attack on 805-Round Trivium

- The preference bit to find a linear superpoly is $s_{66}^{(805)}$.
- $s_{66}^{(805)} = s_{286}^{(739)} \cdot_{287}^{(739)} \oplus s_{243}^{(739)} \oplus s_{288}^{(739)} \oplus s_{69}^{(739)}$
- Choose cubes of sizes 22 and use the Moebius transformation to find a proper cube whose superpoly in $s_{286}^{(739)}$ is linear.
- Start from

$$I_1 = \{v_4, v_6, v_{10}, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, v_{25}, v_{29}, v_{32},$$

$$v_{34}, v_{36}, v_{39}, v_{41}, v_{43}, v_{50}\}$$

A Practical Key-Recovery Attack on 805-Round Trivium

- First stage: We extend I_1 of size 17 to I_2 of size 34 as follows:

$$I_2 = \{v_4, v_6, v_{10}, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, v_{25}, v_{29}, v_{32}, v_{34}, v_{36}, v_{39}, v_{41}, v_{43}, v_{50},$$
$$v_2, v_{69}, v_{79}, v_8, v_{27}, v_0, v_1, v_{28}, v_{71}, v_{13}, v_{45}, v_{23}, v_{26}, v_{38}, v_{76}, v_{47}, v_{56}\}.$$

- Second stage: We start with

$$I_3 = \{v_4, v_6, v_{10}, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, v_{25}, v_{29}, v_{32}, v_{34}, v_{36}, v_{39}, v_{41}, v_{43}, v_{50},$$
$$v_2, v_{69}, v_{79}, v_8, v_{27}, v_0, v_1, v_{28}, v_{71}, v_{13}, v_{45}, v_{23}, v_{26}, v_{38}, v_{76}, v_{47}, v_{52}\},$$

Table: The chosen cube variables in the last iteration

Chosen cube	$I_3 \cup \{v_{48}\}$	$I_3 \cup \{v_{59}\}$	$I_3 \cup \{v_{58}\}$	$I_3 \cup \{v_{63}\}$
degree of superpolies	1	1	2	3

A Practical Key-Recovery Attack on 805-Round Trivium

- Finally, I_4 of size 40 is a candidate large cube, which is given by

$$I_4 = \{v_4, v_6, v_{10}, v_{11}, v_{15}, v_{17}, v_{19}, v_{21}, v_{25}, v_{29}, v_{32}, v_{34}, v_{36}, v_{39},$$
$$v_{41}, v_{43}, v_{50}, v_2, v_{69}, v_{79}, v_8, v_{27}, v_0, v_1, v_{28}, v_{71}, v_{13},$$
$$v_{45}, v_{23}, v_{26}, v_{38}, v_{76}, v_{47}, v_{52}, v_{57}, v_{42}, v_{48}, v_{58}, v_{59}, v_{63}\}.$$

A Practical Key-Recovery Attack on 805-Round Trivium

- Together with some other candidate cubes, we find more than 1000 cubes with linear superpolies in the output of 805-round Trivium.
- There are 38 independent linear superpolies.
- We also find 16 linear superpolies for 806-round Trivium.

A Practical Key-Recovery Attack on 805-Round Trivium

Table: Linear superpolies of 805-round Trivium (incomplete)

cube indies	superpolies
0,1,2,4,6,8,11,13,15,17,19,21,23,26,27,28,29,32,34, 36,38,39,41,42,45,47,48,50,52,53,57,69,71,75,76,79	$1 \oplus k_2 \oplus k_{65}$
0,1,2,4,6,8,10,11,12,13,15,16,19,21,23,25,26,27,29, 31,34,36,38,39,40,43,45,47,49,62,64,70,74,77,79	$1 \oplus k_3$
0,1,2,4,6,8,10,11,13,15,17,19,21,23,26,27,29,31, 34,36,38,39,40,41,43,45,47,49,58,62,64,77,79	$k_4 \oplus k_{19} \oplus k_{34}$
0,1,2,4,6,8,10,13,15,17,19,21,23,25,26,27,28,29,32,34, 36,38,39,41,42,43,47,48,50,52,57,59,69,71,75,76,79	k_{14}
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,29,32,34, 36,38,39,41,42,43,47,48,50,52,53,57,59,69,71,76,79	k_{15}
0,1,2,4,6,8,10,13,15,17,19,21,23,25,26,27,28,29,32, 34,36,38,39,41,42,43,47,48,50,52,59,69,71,75,76,79	$1 \oplus k_{16}$
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,28,29,32, 34,36,38,39,41,42,43,45,47,48,50,53,57,69,71,75,76,79	$1 \oplus k_{17}$
0,1,2,4,6,8,10,11,12,13,15,16,19,21,23,25,27,28, 29,34,36,38,40,41,43,45,47,49,50,64,70,74,77,79	k_{18}
0,1,2,4,6,8,10,11,12,13,15,16,19,23,25,27,28,31,34, 36,38,39,40,41,43,45,47,49,50,58,62,64,74,77,79	$1 \oplus k_{19} \oplus k_{34} \oplus k_{51}$

A Practical Key-Recovery Attack on 805-Round Trivium

Table: Linear superpolies of 805-round Trivium (incomplete)

cube indies	superpolies
0,2,4,6,8,10,12,13,15,17,19,21,23,25,26,27,28,29,31, 34,38,39,40,41,43,45,47,49,50,58,62,64,70,74,77,79	k_{21}
1,2,4,6,8,10,11,12,13,15,17,19,21,23,25,26,27,28,29, 31,34,36,38,39,40,41,43,47,49,50,58,62,70,74,77,79	$1 \oplus k_{29}$
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,29,32,34, 36,38,39,42,43,45,47,48,50,52,53,57,59,69,71,75,76,79	$k_{31} \oplus k_{46} \oplus k_{56}$
0,1,2,4,6,8,10,13,15,17,19,21,23,25,26,28,29,32,34, 36,38,39,41,42,45,47,48,50,52,57,59,69,71,75,76,79	$k_{17} \oplus k_{32}$
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,29,32,34, 36,38,39,42,43,45,47,48,50,52,53,57,59,69,71,76,79	$1 \oplus k_{33}$
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,29,32, 34,36,39,41,42,43,45,47,48,50,52,57,59,69,71,76,79	k_{34}
0,1,2,4,6,8,10,11,13,15,17,19,21,23,25,26,27,29,32, 34,36,38,39,41,42,43,45,47,50,52,53,57,69,71,75,79	k_{36}
0,1,2,4,6,8,10,12,13,15,17,19,21,23,25,26,27,28, 29,31,34,36,39,40,41,43,47,49,50,62,64,70,77,79	k_{40}
0,1,2,4,6,8,10,11,13,15,17,19,21,23,26,27,28,31, 34,36,38,40,41,43,45,47,49,50,58,62,64,70,77,79	k_{42}

A Practical Key-Recovery Attack on 805-Round Trivium

Table: Linear superpolies of 806-round Trivium (incomplete)

cube indices	superpoly
0,1,3,5,7,9,10,11,12,14,15,18,20,22,24,27,28,30, 33,35,37,39,40,42,44,46,48,49,57,61,63,73,76,78	$k_{14} \oplus k_{44}$
0,1,3,5,7,9,10,11,12,14,15,18,20,22,24,26,28,30, 33,35,37,39,40,42,44,46,48,49,57,61,63,73,76,78	k_{15}
0,1,3,5,7,9,10,11,12,14,15,18,20,22,24,26,28,30, 33,35,37,39,40,42,44,46,48,49,57,61,63,76,78	$1 \oplus k_{17}$
0,1,3,5,7,9,10,11,12,14,16,18,20,22,24,25,26,27,28, 30,33,35,37,38,39,40,42,46,48,49,57,61,69,73,76,78	$1 \oplus k_{28}$
0,1,3,5,7,9,10,11,12,14,15,16,18,20,22,24,25,26,27, 28,30,33,35,37,38,39,40,42,46,48,49,57,61,63,76,78	k_{32}
0,1,3,5,7,9,10,11,12,14,15,18,20,22,24,26,27,28, 33,35,37,39,40,42,44,46,48,49,57,61,63,73,76,78	k_{33}
0,3,5,7,9,11,14,15,18,20,22,24,25,26,27,30,33,35, 37,39,40,42,44,46,48,49,57,61,63,69,73,76,78	k_{41}
0,1,3,5,7,9,10,11,12,14,15,18,20,22,24,26,27,28, 30,33,35,37,40,42,44,46,48,49,57,61,63,73,76,78	k_{42}
1,3,5,7,9,10,11,12,14,16,18,20,22,24,25,26,27,28, 30,33,35,37,38,39,40,42,46,48,49,57,61,63,73,76,78	k_{44}

A Summery of Key-Recovery Attacks on Trivium

Attack type	# of rounds	Off-line phase		On-line phase	Total time	ref.
		cube size	# of key bits			
Practical	672	12	63	2^{17}	$2^{18.56}$	[DS09]
	709	22-23	79	< 2	$2^{29.14}$	[MS11]
	767	28-31	35	2^{45}	$2^{45.00}$	[DS09]
	784	30-33	42	2^{38}	2^{39}	[FV13]
	805	32-38	42	2^{38}	$2^{41.40}$	this report
Not practical	799	32-37	18	2^{62}	$2^{62.00}$	[FV13]
	802	34-37	8	2^{72}	$2^{72.00}$	[YT18]
	805	28	7	2^{73}	$2^{73.00}$	[LYWL18]
	806	34-37	16	2^{64}	2^{64}	this report
	835	35	5	2^{75}	$2^{75.00}$	[LYWL18]
	832	72	1	2^{79}	$2^{79.01}$	[WHG+19, TIHM17a, TIHM18]
	832	72	> 1	2^{79}	$< 2^{79.01}$	[YT20]
	840	78	1	2^{79}	$2^{79.58}$	[HLM+20b]
	840	75	3	2^{77}	$2^{77.32}$	[HSWW20]
	841	78	1	2^{79}	$2^{79.58}$	[HLM+20b]
	841	76	2	2^{78}	$2^{78.58}$	[HSWW20]
	842	78	1	2^{79}	$2^{79.58}$	[HLM+20a]
	842	76	2	2^{79}	$2^{78.58}$	[HSWW20]

A candidate cube for 810-Round Trivium

- We construct a candidate cube I_9 for 810-round Trivium of size 43, and we find two subcubes of size 42 with linear superpolies.

$$I_9 = \{v_2, v_6, v_8, v_{10}, v_{11}, v_{15}, v_{19}, v_{21}, v_{25}, v_{29}, v_{30}, v_{32}, v_{34}, v_{36},$$

$$v_{39}, v_{41}, v_{43}, v_{45}, v_{50}, v_0, v_{75}, v_{12}, v_4, v_{14}, v_{20}, v_{22}, v_{16}, v_{27}, v_{23},$$

$$v_{72}, v_{52}, v_{55}, v_{60}, v_{37}, v_{79}, v_{62}, v_{64}, v_{47}, v_{54}, v_{69}, v_{71}, v_{18}, v_{53}\}$$

Conclusions

- A new algorithm to construct good cubes and its application to 805-round Trivium.
  - A Heuristic Algorithm to Construct Candidate Cubes.
  - The Preference Bit and an Algorithm to Predict It.
  - The Improved Möbius Transformation.
- The new algorithm could also be applied to Trivium-like ciphers.

Thanks for Your Attention!

References I

Itai Dinur and Adi Shamir.

Cube attacks on tweakable black box polynomials.

In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, pages 278–299, 2009.

Pierre-Alain Fouque and Thomas Vannet.

Improving key recovery to 784 and 799 rounds of Trivium using optimized cube attacks.

In *Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, pages 502–517, 2013.

References II

Yonglin Hao, Gregor Leander, Willi Meier, Yosuke Todo, and Qingju Wang.

Modeling for three-subset division property without unknown subset.

IACR Cryptol. ePrint Arch., 2020:441, 2020.

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.

In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 466–495. Springer, 2020.

References III

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.

In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 446–476. Springer, 2020.

Abhishek Kesarwani, Dibyendu Roy, Santanu Sarkar, and Willi Meier.

New cube distinguishers on nfsr-based stream ciphers.

Des. Codes Cryptogr., 88(1):173–199, 2020.

References IV

Meicheng Liu.

Degree evaluation of NFSR-Based cryptosystems.

In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 227–249, 2017.

Meicheng Liu, Jingchun Yang, Wenhao Wang, and Dongdai Lin.

Correlation cube attacks: From weak-key distinguisher to key recovery.

In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 715–744, 2018.

References V

Piotr Mroczkowski and Janusz Szmidt.

Corrigendum to: The cube attack on stream cipher Trivium and quadraticity tests.

IACR Cryptology ePrint Archive, 2011:32, 2011.

Santanu Sarkar, Subhamoy Maitra, and Anubhab Baksi.

Observing biases in the state: case studies with trivium and trivia-sc.

Des. Codes Cryptogr., 82(1-2):351–375, 2017.

References VI

Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier.

Cube attacks on non-blackbox polynomials based on division property.

In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, pages 250–279, 2017.

Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier.

Cube attacks on non-blackbox polynomials based on division property (full version).

Cryptology ePrint Archive, Report 2017/306, 2017.

https://eprint.iacr.org/2017/306.

References VII

Yosuke Todo, Takanori Isobe, Yonglin Hao, and Willi Meier.

Cube attacks on non-blackbox polynomials based on division property.

IEEE Trans. Computers, 67(12):1720–1736, 2018.

Yosuke Todo and Masakatu Morii.

Bit-based division property and application to simon family.

In *Fast Software Encryption - 23rd International Conference, FSE 2016, Bochum, Germany, March 20-23, 2016, Revised Selected Papers*, pages 357–377, 2016.

References VIII

Yosuke Todo.

Structural evaluation by generalized integral property.

In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 287–314. Springer, 2015.

References IX

Senpeng Wang, Bin Hu, Jie Guan, Kai Zhang, and Tairong Shi.

Milp-aided method of searching division property using three subsets and applications.

In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 398–427. Springer, 2019.

References X

Qingju Wang, Yonglin Hao, Yosuke Todo, Chaoyun Li, Takanori Isobe, and Willi Meier.

Improved division property based cube attacks exploiting algebraic properties of superpoly.

In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, pages 275–305, 2018.

Zejun Xiang, Wentao Zhang, Zhenzhen Bao, and Dongdai Lin.

Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers.

In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, pages 648–678, 2016.

References XI

Chen-Dong Ye and Tian Tian.

A new framework for finding nonlinear superpolies in cube attacks against trivium-like ciphers.

In Willy Susilo and Guomin Yang, editors, *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Wollongong, NSW, Australia, July 11-13, 2018, Proceedings*, volume 10946 of *Lecture Notes in Computer Science*, pages 172–187. Springer, 2018.

Chen-Dong Ye and Tian Tian.

Algebraic method to recover superpolies in cube attacks.

IET Information Security, 14(4):430–441, 2020.