Application of the new modeling technique on Grain-128AEAD

No Author Given

No Institute Given

1 Specification of the KATAN family

KATAN is a family of lightweight block ciphers proposed in [1]. The family consists of three ciphers denoted by KATANn for $n = 32, 48, 64$ indicating the block size of the cipher. All the ciphers in the KATAN family share the key schedule which accepts an 80-bit key and 254 rounds as well as the use of the same building blocks, namely two NFSRs and a small LFSR acting as a counter.

The key schedule of KATANn loads the 80-bit master key $\boldsymbol{k} = (k_0, \ldots, k_{79})$ into a linear feedback register (LFSR), and the feedback polynomial is an 80-bit primitive polynomial, i.e.,

$$k_{i+80} = k_i \oplus k_{i+19} \oplus k_{i+30} \oplus k_{i+67}, 0 \leq 427.$$

KATANn has an n-bit block size, and the n-bit internal state $\boldsymbol{s} = (s_0, s_1, \ldots, s_{n-1})$ distributed on two Registers L_1 and L_2 (of respective lengths of n_1 and n_2 bits). The n-bit plaintext $\boldsymbol{p} = (p_0, \ldots, p_{n-1})$ are load into these two registers. That is, we have

$$L_1 = (s_0, \ldots, s_{n_1-1}) \leftarrow (p_{n_2}, \ldots, p_{n-1})$$
$$L_2 = (s_{n_1}, \ldots, s_{n-1}) \leftarrow (p_0, \ldots, p_{n_2-1}).$$

In each round, L_1 and L_2 are shifted to the right, where the new computed bits are loaded in the least significant bits of L_1 and L_2. After 254 rounds of the cipher, the contents of the registers are then exported as the ciphertext. The round functions in each round t are two non-linear functions f_a and f_b which are defined as follows,

$$f_a(\boldsymbol{s}) = s_{i_1} \oplus s_{i_2} \oplus (s_{i_3} \cdot s_{i_4}) \oplus (c_t \cdot s_{i_5}) \oplus k_{2t},$$
$$f_b(\boldsymbol{s}) = s_{j_1} \oplus s_{j_2} \oplus (s_{j_3} \cdot s_{j_4}) \oplus (s_{j_5} \oplus s_{j_6}) \oplus k_{2t+1},$$

where c_t is a round constant which has the initial value $(1, 1, 1, 1, 1, 1, 1, 0)$ and is updated by a feedback polynomial of $c_{t+8} = c_t \oplus c_{t+1} \oplus c_{t+3} \oplus c_{t+5}$, $t \geq 8$. k_{2t} and k_{2t+1} are the two round subkey bits. The definitions of the indices $\{i_l\}$ and $\{j_l\}$ are different in various versions. The non-linear functions f_a and f_b are applied once, twice, and thrice for KATAN32, KATAN48, and KATAN64, using the same key bits in each round. The differences between KATAN ciphers are shown in Table 1.

Table 1: KATAN family parameters

Cipher	$\|L_1\|$	$\|L_2\|$	i_1	i_2	i_3	i_4	i_5	j_1	j_2	j_3	j_4	j_5	j_6
KATAN32	13	19	12	7	8	5	3	31	20	25	23	21	16
KATAN48	19	29	18	12	15	7	6	47	38	40	32	34	25
KATAN64	25	39	24	15	20	11	9	63	50	58	46	39	34

2 The propagation rules of 3SDP/u for basic Boolean functions

2.1 The propagation rules for three basic functions.

Proposition 1 (COPY). *Let* $\boldsymbol{x} = (x_0, x_1, ..., x_{m-1})$ *and* $\boldsymbol{y} = (x_0, x_0, x_1, ..., x_{m-1})$ *be the input and output vector of a Copy function. Let* $\mathbb{X}$ *and* $\mathbb{Y}$ *be the input and output multisets, respectively. Assuming that* $\mathbb{X}$ *has* $\mathcal{T}_{\mathbb{L}}^{1^m}$, $\mathbb{Y}$ *has* $\mathcal{T}_{\mathbb{L}'}^{1^{m+1}}$ *where* $\mathbb{L}'$ *is computed as*

$$\mathbb{L}' \leftarrow (l'_0, l''_0, l_1, \ldots, l_{m-1}), \text{ if } l'_0 \vee l''_0 = l_0,$$

for all $\boldsymbol{l} \in \mathbb{L}$. *Here* $\mathbb{L}' \leftarrow \boldsymbol{l}$ *denotes that* $\boldsymbol{l}$ *is inserted into the multiset* $\mathbb{L}'$.

Proposition 2 (AND). *Let* $\boldsymbol{x} = (x_1, x_2, ..., x_m)$ *and* $\boldsymbol{y} = (x_1 \cdot x_2, x_3, ..., x_m)$ *be the input and output vector of an And function. Let* $\mathbb{X}$ *and* $\mathbb{Y}$ *be the input and output multisets, respectively. Assuming that* $\mathbb{X}$ *has* $\mathcal{T}_{\mathbb{L}}^{1^m}$, $\mathbb{Y}$ *has* $\mathcal{T}_{\mathbb{L}'}^{1^{m-1}}$ *where* $\mathbb{L}'$ *is computed as*

$$\mathbb{L}' \leftarrow (l_0, l_2, \ldots, l_{m-1}), \text{ if } l_0 = l_1,$$

for all $\boldsymbol{l} \in \mathbb{L}$.

Proposition 3 (XOR). *Let* $\boldsymbol{x} = (x_1, x_2, ..., x_m)$ *and* $\boldsymbol{y} = (x_1 \oplus x_2, x_3, ..., x_m)$ *be the input and output vector of a Xor function. Let* $\mathbb{X}$ *and* $\mathbb{Y}$ *be the input and output multisets, respectively. Assuming that* $\mathbb{X}$ *has* $\mathcal{T}_{\mathbb{L}}^{1^m}$, $\mathbb{Y}$ *has* $\mathcal{T}_{\mathbb{L}'}^{1^{m-1}}$ *where* $\mathbb{L}'$ *is computed as*

$$\mathbb{L}' \leftarrow (l_0 + l_1, l_2, \ldots, l_{m-1}), \text{ if } l_0 \cdot l_1 = 0,$$

for all $\boldsymbol{l} \in \mathbb{L}$.

2.2 MILP models for three basic functions.

We introduce the MILP models for the three basic functions COPY, AND, and XOR mentioned in **??**. Note that these models are first proposed in [2,3], and describe the generalized forms of their propagation rules, which are easily deduced from their original forms. When constructing the MILP model, we usually initialize an empty model $\mathcal{M}$, then generate some MILP variables $v_1, \ldots, v_m$ by $\mathcal{M}.var \leftarrow v_1, \ldots, v_m$, which are used in the inequalities. Next, the inequality is added to the model $\mathcal{M}.con \leftarrow v_2 + \cdots + v_m \geq v_1$. Finally, we can call an off-the-shelf MILP solver to solve the model and obtain the results, including feasible or infeasible. In this paper, the MILP tool we used is the Gurobi optimizer.

Proposition 4 (MILP model for COPY). *Let a* $\xrightarrow{COPY}(b_0, b_1, \ldots, b_{m-1})$ *be a three-subset division trail of* ***COPY****. The following inequalities are sufficient to describe the propagation of the modified three-subset division property for* ***COPY****.*

$$\begin{cases} \mathcal{M}.var \leftarrow a, b_0, \ldots, b_{m-1} \text{ as binary;} \\ \mathcal{M}.con \leftarrow a = b_0 \vee b_1 \vee \cdots \vee b_{m-1}. \end{cases}$$

Note that the Gurobi optimizer supports the Or $(\vee)$ *operation.*

Proposition 5 (MILP model for AND). *Let* $(a_0, a_1, \ldots, a_{m-1}) \xrightarrow{AND} b$ *be a three-subset division trail of* ***AND****. The following inequalities are sufficient to describe the propagation of the modified three-subset division property for* ***AND****.*

$$\begin{cases} \mathcal{M}.var \leftarrow a_0, \ldots, a_{m-1}, b \text{ as binary;} \\ \mathcal{M}.con \leftarrow b = a_i, \forall i \in \{0, 1, \ldots, m-1\}. \end{cases}$$

Proposition 6 (MILP model for XOR). *Let* $(a_0, a_1, \ldots, a_{m-1}) \xrightarrow{XOR} b$ *be a three-subset division trail of* ***XOR****. The following inequalities are sufficient to describe the propagation of the modified three-subset division property for* ***XOR****.*

$$\begin{cases} \mathcal{M}.var \leftarrow a_0, \ldots, a_{m-1}, b \text{ as binary;} \\ \mathcal{M}.con \leftarrow b = a_0 + a_1 + \cdots + a_{m-1}. \end{cases}$$

3 The new MILP model for the KATAN family

In section 1, we introduce the specification of the KATAN family, which comprises three block ciphers with 32-, 48-, and 64-bit block sizes, denoted by KATANn. All the KATAN family ciphers share the same key schedule, master key length, encryption rounds, and NFSRs. Similar to the output of the stream ciphers, the ciphertext $\boldsymbol{c} = (c_0, \ldots, c_{n-1})$ of r-round KATANn can be represented as the vectorial Boolean function with respect to the secret key $\boldsymbol{k} \in \mathbb{F}_2^{80}$ and the plaintext $\boldsymbol{p} = (p_0, \ldots, p_{n-1}) \in \mathbb{F}_2^n$, denoted by $\boldsymbol{K}(\boldsymbol{p}, \boldsymbol{k})$.

The master key $\boldsymbol{k} = (k_0, \ldots, k_{79})$ of KATANn is loaded into an 80-bit linear feedback register, and new round keys are generated by the linear feedback relation, i.e., $k_{i+80} = k_i \oplus k_{i+19} \oplus k_{i+30} \oplus k_{i+67}, 0 \leq i \leq 427$. In each round t, KATANn uses two nonlinear functions $f_a(\cdot)$ and $f_b(\cdot)$ in each round to update state bits, which are defined as follows,

$$\begin{aligned} f_a(\boldsymbol{s}) &= s_{i_1} \oplus s_{i_2} \oplus s_{i_3} \cdot s_{i_4} \oplus c_t \cdot s_{i_4} \oplus k_{2t}, \\ f_b(\boldsymbol{s}) &= s_{j_1} \oplus s_{j_2} \oplus s_{j_3} \cdot s_{j_4} \oplus s_{j_5} \cdot s_{j_6} \oplus k_{2t+1}, \end{aligned}$$

where $\boldsymbol{s} = (s_0, \ldots, s_{n-1}) \in \mathbb{F}_2^n$ is the internal state bits, and c_t is a round constant generated by the 8-bit LFSR using the feedback polynomial. Note that the definitions of indices $\{i_l\}$ and $\{j_l\}$ are different in various versions. Therefore, the vectorial Boolean function $\boldsymbol{K}(\boldsymbol{p}, \boldsymbol{k})$ can also be decomposed into the expression as follows,

$$\boldsymbol{K}(\boldsymbol{p}, \boldsymbol{k}) = \boldsymbol{F}_r \circ \cdots \circ \boldsymbol{F}_1 \circ \boldsymbol{F}_0(\boldsymbol{p}, \boldsymbol{k}), \tag{1}$$

where $\boldsymbol{F}_i : \mathbb{F}_2^{n+80} \to \mathbb{F}_2^{n+80}$ for all $i \in \{0, \ldots, r\}$. Denote the input and output of the round function $\boldsymbol{F}_i$ by $(\boldsymbol{s}_{i-1}, \boldsymbol{k})$ and $(\boldsymbol{s}_i, \boldsymbol{k})$, respectively, where $\boldsymbol{s}_{i-1}$ and $\boldsymbol{s}_i$ are the internal state bit-vectors.

3.1 The new MILP model for KATANn

For each round t, there are only two bits updated respectively, and the subkey bits k_{2t} and k_{2t+1} are involved in the two update functions f_a and f_b. Obviously, when $t \geq 40$, k_{2t} and k_{2t+1} can be represented as the expressions of the master key $\boldsymbol{k}$. Therefore, for r-round KATANn, we can naturally regard the secret key variables $k_{80+i}, \ldots, k_{2r-1}$ as the independent secret key variables. Let $\boldsymbol{x} = (k_0, \ldots, k_{79}, k_{80}, \ldots, k_{2r-1}) \in \mathbb{F}_2^{2r}$. Thus, the round function $\boldsymbol{F}_i$ can be regarded as the function with respect to $\boldsymbol{s}_{i-1}$ and the new secret key variables $\boldsymbol{x}$, denoted by $\boldsymbol{F}'_i(\boldsymbol{s}_{i-1}, \boldsymbol{x})$. Further, the vectorial Boolean function $\boldsymbol{K}(\boldsymbol{p}, \boldsymbol{k})$ can be represented as the vectorial Boolean function with respect to $\boldsymbol{p}$ and $\boldsymbol{x}$, i.e.,

$$\boldsymbol{K}(\boldsymbol{p}, \boldsymbol{x}) = \boldsymbol{F}'_r \circ \cdots \circ \boldsymbol{F}'_1 \circ \boldsymbol{F}'_0(\boldsymbol{p}, \boldsymbol{x}). \tag{2}$$

According to Eq.(2), a new MILP model of KATANn can be built. By adding the specific constraints, we can recover superpolies at any position in the ciphertext. It is worth noting that the superpolies recovered by the new models are the polynomials with respect to the new secret key variables $\boldsymbol{x}$, which have fewer monomials than the corresponding superpolies with respect to the master key $\boldsymbol{k}$. Compared with the model built by the traditional method, the new model for KATANn does not include extra feasible solutions. The details of the new MILP model are present in Algorithm 1.

Algorithm 1: The New MILP Model for KANTANn

procedure KATANMODEL(Round R, Cube indices I, Block size n, location l)
  Prepare an empty MILP Model $\mathcal{M}$
  $\mathcal{M}.var \leftarrow \boldsymbol{s} = (s_0, \ldots, s_{n-1})$ as binary variables
  $\mathcal{M}.var \leftarrow \boldsymbol{k} = (k_0, \ldots, k_{2R-1})$ as binary variables
  $\boldsymbol{c} \leftarrow (1, 1, 1, 1, 1, 1, 1, 0)$
  $\mathcal{M}.con \leftarrow s_{n-1-i} = 1$ for $i \in I$
  $\mathcal{M}.con \leftarrow s_{n-1-i} = 0$ for $i \in \{0, \ldots, n-1\} \backslash I$
  $L_1 \leftarrow \{5, 4, 7, 24, 19, 21, 23, 28, 9, 13\}$
  $L_2 \leftarrow \{6, 3, 11, 28, 26, 34, 32, 41, 12, 19\}$
  $L_3 \leftarrow \{9, 4, 13, 38, 30, 42, 49, 54, 15, 25\}$
  $m \leftarrow 1 * (n == 32) + 2 * (n == 48) + 3 * (n == 64)$
  for r **from** 0 **to** $R - 1$ **do**
    $\mathcal{M}.var \leftarrow (tk_0, \ldots, tk_{2*m-1})$ as binary variables
    $\mathcal{M}.con \leftarrow tk_0 + tk_2 + \cdots + tk_{2*m-2} \geq k_{2*r}$
    $\mathcal{M}.con \leftarrow tk_i \leq k_{2*r}$ for $i \in \{0, 2, \ldots, 2 * m - 2\}$
    $\mathcal{M}.con \leftarrow tk_1 + tk_3 + \cdots + tk_{2*m-1} \geq k_{2*r+1}$
    $\mathcal{M}.con \leftarrow tk_i \leq k_{2*r+1}$ for $i \in \{1, 3, \ldots, 2 * m - 1\}$
    for t **from** 0 **to** $m - 1$ **do**
      $\mathcal{M}.var \leftarrow (ts_0, ts_1)$ as binary variables
      $\mathcal{M}.var \leftarrow (tv_0, \ldots, tv_{17})$ as binary variables
      for i **from** 0 **to** 8 **do**
        $\mathcal{M}.con \leftarrow tv_{2*i} + tv_{2*i+1} \geq s_{L_m[i]}$
        $\mathcal{M}.con \leftarrow tv_j \leq s_{L_m[i]}$ for $j \in \{2 * i, 2 * i + 1\}$
        $s_{L_m[i]} \leftarrow tv_{2*i}$
      $\mathcal{M}.con \leftarrow tv_j = tv_{j+2}$ for $j \in \{3, 9, 13\}$
      if $c_0 = 1$ **then**
        $\mathcal{M}.con \leftarrow tv_{16} + tv_{17} \geq s_{L_m[8]}$
        $\mathcal{M}.con \leftarrow tv_j \leq s_{L_m[8]}$ for $j \in \{16, 17\}$
        $s_{L_m[8]} \leftarrow tv_{16}$
      else
        $\mathcal{M}.con \leftarrow tv_j = 0$ for $j \in \{16, 17\}$
      $\mathcal{M}.con \leftarrow ts_0 = s_0 + tv_1 + tv_3 + tv_{17} + tk_{2*t}$
      $\mathcal{M}.con \leftarrow ts_1 = s_{L_m[9]} + tv_7 + tv_9 + tv_{13} + tk_{2*t+1}$
      $(s_0, \ldots, s_{n-1}) \leftarrow (s_1, \ldots, s_{L_m[9]-1}, ts_1, s_{L_m[9]+1}, \ldots, n-1, ts_0)$
    $(c_0, \ldots, c_7) \leftarrow (c_1, \ldots, c_7, c_0 \oplus c_1 \oplus c_3 \oplus c_5)$
  $\mathcal{M}.con \leftarrow s_l = 1$
  $\mathcal{M}.con \leftarrow s_i = 0$ for $i \in \{0, \ldots, l-1, l+1, \ldots, n-1\}$
  return $\mathcal{M}$
end procedure

4 The results of verification experiments on the KATAN family.

We recover the superpolies by solving different MILP models for the KATAN family of block ciphers. In addition to the total number of the three-subset division trails, we also record the number of the trails corresponding to the 1-

constant. For KATAN32, the target rounds are from 97 to 102, and the cubes we used are 31-dimension. For KATAN48, the target rounds are from 79 to 85, and the cubes we used are 47-dimension. Similarly, for KATAN64, the target rounds are from 70 to 73, and the cubes we used are 63-dimension. All the cubes used in these experiments are constructed by a heuristic method. Table 2 shows the statistics of the results. Since this table is a little different from the previous ones, we make a simple explanation. For example, the data "453928(34162), 52522(34162), 1688, 43, 15476, 266, 3, 3, 31" in line 3 of Table 2 means that the superpoly we recovered is from the ciphertext bit c_{31} of 102-round KATAN32. It took 1688 seconds to solve the traditional model, and the number of solutions is 453928, among which there are 34162 three-subset division trails corresponding to the monomial 1; in addition, the recovered superpoly has 15476 monomials, and its algebraic degree is 3. Similarly, It took 43 seconds to solve the new MILP model, and the number of solutions is 52522, among which there are 34162 trails corresponding to the monomial 1; in addition, the recovered superpoly has 266 monomials, and its degree is 3. As shown in Table 2, the numbers of feasible solutions in the new model is far less than those in the traditional models, and the more solutions contained in the traditional model, the better the effect of our new technique, which turns out that our new modeling technique is practically effective for the KATAN family. It is remarkable that, for these block ciphers, we recover the superpolies with a higher number of rounds than the previous best recovered superpolies.

It is easy to observe two interesting phenomena from Table 2. The one is that the degree and the number of three-subset division trails corresponding to the monomial 1 are the same for both the traditional model and the new model, which are caused by the fact that the key expressions replaced with the new variables in the new technique are linear. The other is that for the models built by our new technique, there are usually more than half of the feasible solutions related to the constant 1. Suppose we can propose a method to determine whether the 1-constant appears in the superpoly quickly. In that case, we can require such three-subset division trails to be forbidden to exist in the new model, which can further accelerate the solving of the new model.

Table 2: statistics of the results for different MILP models of KATANn

n	R	I	# of trails		time (s)		# of monomials degree		cipher		
			$\mathcal{M}_T$	$\mathcal{M}_{new}$	$\mathcal{M}_T$	$\mathcal{M}_N$	$\boldsymbol{k}$	$\boldsymbol{x}$	$\boldsymbol{k}$	$\boldsymbol{x}$	bit
32	102	$\{0,\ldots,31\}\backslash\{13\}$	9306(1718)	2292(1718)	9	2	286	22	2	2	31
	101	$\{0,\ldots,31\}\backslash\{14\}$	9706(2084)	2743(2084)	12	8	286	27	2	2	31
	102	$\{0,\ldots,31\}\backslash\{14\}$	453928(34162)	52522(34162)	1688	43	15476	266	3	3	31
	100	$\{0,\ldots,31\}\backslash\{15\}$	9390(2122)	2781(2122)	18	4	284	23	2	2	31
	99	$\{0,\ldots,31\}\backslash\{16\}$	7684(1947)	2554(1947)	21	3	192	22	2	2	31
	100	$\{0,\ldots,31\}\backslash\{16\}$	358457(35641)	53918(35641)	1403	104	17177	296	3	3	31
	99	$\{0,\ldots,31\}\backslash\{17\}$	332554(38373)	57796(38373)	1067	66	15074	282	2	2	31
	97	$\{0,\ldots,31\}\backslash\{16\}$	27(7)	9(7)	1	1	17	3	1	1	31
	97	$\{0,\ldots,31\}\backslash\{17\}$	179(122)	128(122)	2	1	17	2	1	1	31
	98	$\{0,\ldots,31\}\backslash\{17\}$	12653(3632)	4670(3632)	19	14	489	32	2	2	31
	97	$\{0,\ldots,31\}\backslash\{18\}$	31133(7424)	9710(7424)	40	19	1475	62	3	3	31
	98	$\{0,\ldots,31\}\backslash\{18\}$	630639(80615)	122553(80615)	4392	191	18195	191	3	3	31
48	85	$\{0,\ldots,47\}\backslash\{19\}$	261093(42346)	81651(42346)	494	60	4321	22	2	2	47
	84	$\{0,\ldots,47\}\backslash\{20\}$	154(101)	121(101)	4	6	6	3	1	1	47
	84	$\{0,\ldots,47\}\backslash\{21\}$	437391(112938)	197174(112938)	2043	444	3207	268	4	4	47
	83	$\{0,\ldots,47\}\backslash\{22\}$	257(165)	197(165)	4	4	9	3	1	1	47
	83	$\{0,\ldots,47\}\backslash\{23\}$	477066(136217)	235146(136217)	2389	559	2300	238	4	4	47
	82	$\{0,\ldots,47\}\backslash\{24\}$	165(116)	141(116)	3	4	9	3	2	2	47
	82	$\{0,\ldots,47\}\backslash\{25\}$	320231(85463)	156698(85463)	1012	269	1957	226	4	4	47
	81	$\{0,\ldots,47\}\backslash\{27\}$	212533(68297)	126283(68297)	489	194	1135	221	4	4	47
	80	$\{0,\ldots,47\}\backslash\{29\}$	130629(45848)	89055(45848)	202	129	541	191	4	4	47
	79	$\{0,\ldots,47\}\backslash\{30\}$	168(130)	162(130)	4	3	4	4	1	1	47
	79	$\{0,\ldots,47\}\backslash\{31\}$	174900(72917)	130314(72917)	356	213	760	266	4	4	47
64	73	$\{0,\ldots,63\}\backslash\{27\}$	1050(680)	1050(680)	20	7	12	12	2	2	63
	73	$\{0,\ldots,63\}\backslash\{28\}$	63715(32223)	58285(32223)	146	107	139	121	3	3	63
	72	$\{0,\ldots,63\}\backslash\{30\}$	6678(3839)	6666(3839)	9	31	48	48	3	3	63
	72	$\{0,\ldots,63\}\backslash\{31\}$	94097(54275)	93932(54275)	127	125	155	148	3	3	63
	71	$\{0,\ldots,63\}\backslash\{33\}$	390(319)	390(319)	25	24	10	10	2	2	63
	71	$\{0,\ldots,63\}\backslash\{34\}$	197670(119865)	192108(119865)	339	319	352	296	4	4	63
	70	$\{0,\ldots,63\}\backslash\{38\}$	584(488)	584(488)	29	30	8	8	2	2	63
	70	$\{0,\ldots,63\}\backslash\{39\}$	31300(20136)	31300(20136)	103	80	112	112	3	3	63

References

1. Cannière, C.D., Dunkelman, O., Knezevic, M.: KATAN and KTANTAN - A family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings. Lecture Notes in Computer Science, vol. 5747, pp. 272–288. Springer, Berlin, Heidelberg (2009), https://doi.org/10.1007/978-3-642-04138-9_20
2. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for three-subset division property without unknown subset - improved cube attacks against Trivium and Grain-128AEAD. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptol-

ogy - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 466–495. Springer, Cham (2020), https://doi.org/10.1007/978-3-030-45721-1_17

3. Hao, Y., Leander, G., Meier, W., Todo, Y., Wang, Q.: Modeling for three-subset division property without unknown subset. J. Cryptol. **34**(3), 22 (2021). https://doi.org/10.1007/s00145-021-09383-2, https://doi.org/10.1007/s00145-021-09383-2