Cube Attack against 843-Round Trivium

Yao Sun

State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China.
sunyao@iie.ac.cn

Abstract. Cube attack has recently been proved as the most effective approach of attacking Trivium. So far, the attack against the highest round-reduced Trivium was given in EUROCRYPT 2020, where key-recovery attacks on 840-, 841-, and 842-round Trivium were presented. By revealing the relation between three-subset division property without unknown subset and the monomials of superpolys, Hu et al. obtained more attacks on 840-, 841-, and 842-round Trivium with lower complexities in ASIACRYPT 2020. In this short paper, we will present a key-recovery cube attack against 843-round Trivium.

Keywords: Stream cipher · Cube attack · MILP · Trivium.

1 Introduction

Dinur and Shamir proposed cube attack in EUROCRYPT 2009 [2]. Since the division property [6] [8] was used for detecting characteristics of superpolys [7], cube attack has become the most effective approach of attacking Trivium. By using three-subset division property without unknown subset, Hao et al. presented key-recovery attacks on 840-, 841-, and 842-round Trivium in [4]. From then on, 842 became the highest round of round-reduced Trivium that can be attacked with complexities strictly lower than $2^{80}$. Consequently, Hu et al. pointed out in ASIACRYPT 2020 that computing three-subset division property without unknown subset is equivalent to calculating the appearing monomials in the superpolys. They used a divide-and-conquer strategy to speed up the computations, and hence, obtained more key-recovery attacks on 840-, 841-, and 842-round Trivium with even lower complexities [5].

In this paper, we will report a cube attack against 843-round Trivium. We found a cube with 78 public variables, and this cube is related to a balance superpoly with an independent secret variable, i.e. this secret variable appears as a monomial with degree 1 in this superpoly and all other appearing monomials do not involve it. Therefore, we can recover 1 bit of information with complexity $2^{78}$, and the exhaustive searching complexity for recovering the whole key becomes $2^{79}$, which results in, as far as we know, the first key-recovery attack against 843-round Trivium.

We devised a new efficient algorithm for searching the cubes that could lead to key-recovery attacks against round-reduced Trivium. As a result, we efficiently
found hundreds of new cubes against 840-, 841-, and 842-round Trivium. More importantly, we found two cubes for 843-round Trivium, and one of them is presented in this paper. This new algorithm for searching cubes will be reported sooner.

2 Preliminaries

Cube attack: Cube attack was proposed in EUROCRYPT 2009 [2]. For a cipher with $n$ secret variables and $m$ public variables, each output bit of this cipher can be represented as a polynomial in secret and public variables. Denote $x = (x_0, x_1, \ldots, x_{n-1})$ and $v = (v_0, v_1, \ldots, v_{m-1})$ as the secret and public variables, where $x_i, v_j \in \mathbb{F}_2$ for $0 \leq i < n$ and $0 \leq j < m$. Thus, the output bit can be written as $f(x, v)$, where $f$ is a polynomial in the ring $\mathbb{F}_2[x, v]$. Let $I \subset \{0, 1, \ldots, m-1\}$ be a set of indices of public variables. A cube determined by $I$ is denoted as $C_I$, and contains all $2^{|I|}$ possible combinations of values of $v_j$’s for $j \in I$, while every value of $v_k$, where $k \in \{0, 1, \ldots, m-1\} \setminus I$, remains unchanged. Then we have the following equation:

$$\bigoplus_{C_I} f(x, v) = \bigoplus_{C_I} (t_I \cdot p(x, v) + q(x, v)) = p(x, v),$$

where $t_I$ represents the product $\prod_{i \in I} v_i$, and there is no term of $q(x, v)$ that is divisible by $t_I$. The polynomial $p(x, v)$ is called the superpoly of the cube $C_I$. In the offline phase of cube attack, attackers recover the superpoly. Next, in the online phase, attackers can get the value of this superpoly by querying the encryption oracle $2^{|I|}$ times, and then some information about the secret key can be obtained if the superpoly has a special structure.

Fig. 1. Structure of Trivium
**Trivium**: Trivium is an NFSR-based stream cipher [1]. As shown in Figure 1, Trivium has a 288-bit internal state \((s_1, s_2, \ldots, s_{288})\) which is divided into three registers. The 80-bit secret key \(K\) is loaded into the first register, and the 80-bit initialization vector \(IV\) is loaded into the second register. The other bits of the state are set to 0 except the last three bits in the third register. That is, we have

\[
(s_1, s_2, \ldots, s_{93}) := (K[0], K[1], \ldots, K[79], 0, \ldots, 0), \\
(s_{94}, s_{95}, \ldots, s_{177}) := (IV[0], IV[1], \ldots, IV[79], 0, \ldots, 0), \\
(s_{178}, s_{179}, \ldots, s_{288}) := (0, 0, \ldots, 0, 1, 1, 1).
\]

The state of Trivium is updated in the following way:

\[
\begin{align*}
t_1 &\leftarrow s_{66} \oplus s_{93}, \\
t_2 &\leftarrow s_{162} \oplus s_{177}, \\
t_3 &\leftarrow s_{243} \oplus s_{288}, \\
z &\leftarrow t_1 \oplus t_2 \oplus t_3, \\
t_1 &\leftarrow t_1 \oplus s_{91} \cdot s_{92} \oplus s_{171}, \\
t_2 &\leftarrow t_2 \oplus s_{175} \cdot s_{176} \oplus s_{264}, \\
t_3 &\leftarrow t_3 \oplus s_{286} \cdot s_{287} \oplus s_{69}, \\
(s_1, s_2, \ldots, s_{93}) &\leftarrow (t_3, s_1, \ldots, s_{92}), \\
(s_{94}, s_{95}, \ldots, s_{177}) &\leftarrow (t_1, s_{94}, \ldots, s_{176}), \\
(s_{178}, s_{179}, \ldots, s_{288}) &\leftarrow (t_2, s_{178}, \ldots, s_{287}),
\end{align*}
\]

where \(z\) denotes the 1-bit key stream. First, the state is updated 1152 times without producing an output. After the key initialization is done, one bit key stream is produced by every update function.

### 3 A cube attack against 843-round Trivium

Taking the cube indices \(I = \{0, 1, \ldots, 29, 31, \ldots, 75, 77, 78, 79\}\) with

\[
IV[30] = IV[76] = 0,
\]

we can recover a balance superpoly of \(I\) with an independent variable \(K[2]\), i.e. \(K[2]\) appears as a monomial with degree 1 in this superpoly and all other appearing monomials do not involve it. The details about the superpoly can be found in Appendix.

Therefore, we can get the value of this superpoly by summing up all \(2^{78}\) possible values in the cube. Since the value of \(K[2]\) can be deduced directly from the other secret bits by using this superpoly, we can recover the whole 80-bit key by doing \(2^{79}\) exhaustive searches. The overall complexity is lower than \(2^{80}\).
For recovering the superpoly of $I$, we built an MILP model similar to that in [5], and proposed a new divide-and-conquer algorithm to solve this model using Gurobi [3]. Some new techniques are used and the computations are sped up significantly. We will report these techniques sooner. Finally, we got 1,085,554,019 solutions of the MILP model. These solutions are related to 140,096 raw monomials, and the distribution of solutions is quiet unbalanced. For instance, there are 396,911,938 solutions corresponding to the constant monomial “1”. By removing the raw monomials whose solution numbers are even, we obtained 16,561 monomials that really appear in the superpoly of $I$. Detailed results can be found in https://github.com/ysun0102/trivium. We provide a program based on the codes of [5] for verification. This program will output the set of all monomials containing $K[2]$ in the superpoly, which is supposed to be $\{K[2]\}$. One can also verify other parts of this superpoly by this program.

References

Appendix

The superspoly recovered for 843-round 8TIVIUM is listed below, where the secret variables are represented as $k = (k_0, k_1, \ldots, k_79)$:

$$p_r(k) = k_{90}k_{37} + k_{90}k_{1}k_{55}k_{58} + k_{90}k_{31}k_{60} + k_{90}k_{23} + k_{90}k_{22}k_{10} + k_{90}k_{23}k_{70}k_{71} + k_{90}k_{23} + k_{90}k_{145} + k_{90}k_{23} + k_{90}k_{12} + k_{90}k_{5} + k_{90}k_{5}k_{5}$$

Cube Attack against 843-Round TRIVIUM
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round TRIVIUM
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium

+k23k29k35k36k38 + k23k29k35k36k60 + k23k29k55k60k70k71
+k23k29k35k36k72 + k23k29k35k36k73 + k23k29k35k36k54 + k23k29k35k36k60 + k23k29k35k36k59
+k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54
+k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54
+k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54
+k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54
+k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54 + k23k29k35k36k54
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
+k_39k_{46}k_7k_{87}k_70 + k_30k_{47}k_{64} + k_30k_{52}k_8 + k_30k_{51}k_5 + k_30k_{52}k_{55}k_58
+k_39k_{52}k_3k_{60} + k_30k_{52}k_{53}k_7 + k_30k_{52}k_{53}k_{72} + k_30k_{52}k_{60}
+k_39k_{53} + k_30k_{54}k_5 + k_30k_{53}k_3k_8 + k_30k_{53}k_4k_9 + k_30k_{53}k_4k_9
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_5 + k_30k_{53}k_5 + k_30k_{53}k_5 + k_30k_{53}k_6
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
+k_39k_{53}k_{57}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_7 + k_30k_{53}k_8
Cube Attack against 834-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium
Cube Attack against 843-Round Trivium

\[ k_{67}k_{71}k_{72} + k_{67}k_{71}k_{72}k_{73} + k_{67}k_{71}k_{72}k_{75} + k_{67}k_{72}k_{76} + k_{67}k_{72}k_{78}k_{79} + k_{67}k_{73} + k_{67}k_{73}k_{75} + k_{67}k_{74} + k_{67}k_{75} + k_{68} + k_{68}k_{69}k_{70}k_{71}k_{72} + k_{68}k_{69}k_{71}k_{73}k_{74} + k_{68}k_{69}k_{72}k_{75} + k_{68}k_{69}k_{74} + k_{68}k_{69}k_{75} + k_{68}k_{69}k_{76}k_{77} + k_{68}k_{70}k_{71} + k_{68}k_{70}k_{72} + k_{68}k_{70}k_{73} + k_{68}k_{70}k_{74} + k_{69}k_{70}k_{71}k_{72} + k_{69}k_{70}k_{73} + k_{69}k_{70}k_{74} + k_{69}k_{70}k_{75} + k_{69}k_{71} + k_{69}k_{72} + k_{69}k_{72}k_{73} + k_{69}k_{72}k_{74} + k_{69}k_{72}k_{75} + k_{69}k_{72}k_{76} + k_{69}k_{72}k_{77} + k_{69}k_{72}k_{78} + k_{69}k_{72}k_{79} + k_{70} + k_{70}k_{71}k_{72} + k_{70}k_{71}k_{73} + k_{70}k_{71}k_{74} + k_{70}k_{71}k_{75} + k_{70}k_{71}k_{76} + k_{70}k_{71}k_{77} + k_{70}k_{71}k_{78} + k_{70}k_{71}k_{79} + k_{70}k_{72}k_{73} + k_{70}k_{72}k_{74} + k_{70}k_{72}k_{75} + k_{70}k_{72}k_{76} + k_{70}k_{72}k_{77} + k_{70}k_{72}k_{78} + k_{70}k_{72}k_{79} + k_{70}k_{73}k_{74} + k_{70}k_{73}k_{75} + k_{70}k_{73}k_{76} + k_{70}k_{73}k_{77} + k_{70}k_{73}k_{78} + k_{70}k_{73}k_{79} + k_{70}k_{74}k_{75} + k_{70}k_{74}k_{76} + k_{70}k_{74}k_{77} + k_{70}k_{74}k_{78} + k_{70}k_{74}k_{79} + k_{70}k_{75}k_{76} + k_{70}k_{75}k_{77} + k_{70}k_{75}k_{78} + k_{70}k_{75}k_{79} + k_{70}k_{76}k_{77} + k_{70}k_{76}k_{78} + k_{70}k_{76}k_{79} + k_{70}k_{77}k_{78} + k_{70}k_{77}k_{79} + k_{70}k_{78}k_{79} + k_{70}k_{79} + k_{71}k_{72} + k_{71}k_{73}k_{74} + k_{71}k_{73}k_{75} + k_{71}k_{73}k_{76} + k_{71}k_{73}k_{77} + k_{71}k_{73}k_{78} + k_{71}k_{73}k_{79} + k_{71}k_{74}k_{75} + k_{71}k_{74}k_{76} + k_{71}k_{74}k_{77} + k_{71}k_{74}k_{78} + k_{71}k_{74}k_{79} + k_{71}k_{75}k_{76} + k_{71}k_{75}k_{77} + k_{71}k_{75}k_{78} + k_{71}k_{75}k_{79} + k_{71}k_{76}k_{77} + k_{71}k_{76}k_{78} + k_{71}k_{76}k_{79} + k_{71}k_{77}k_{78} + k_{71}k_{77}k_{79} + k_{71}k_{78}k_{79} + k_{71}k_{79} + k_{72}k_{73} + k_{72}k_{74} + k_{72}k_{75}k_{76} + k_{72}k_{75}k_{77} + k_{72}k_{75}k_{78} + k_{72}k_{75}k_{79} + k_{72}k_{76}k_{77} + k_{72}k_{76}k_{78} + k_{72}k_{76}k_{79} + k_{72}k_{77}k_{78} + k_{72}k_{77}k_{79} + k_{72}k_{78}k_{79} + k_{72}k_{79} + k_{73}k_{74} + k_{73}k_{75} + k_{73}k_{76}k_{77} + k_{73}k_{76}k_{78} + k_{73}k_{76}k_{79} + k_{73}k_{77}k_{78} + k_{73}k_{77}k_{79} + k_{73}k_{78}k_{79} + k_{73}k_{79} + k_{74}k_{75}k_{76} + k_{74}k_{75}k_{77} + k_{74}k_{75}k_{78} + k_{74}k_{75}k_{79} + k_{74}k_{76}k_{77} + k_{74}k_{76}k_{78} + k_{74}k_{76}k_{79} + k_{74}k_{77}k_{78} + k_{74}k_{77}k_{79} + k_{74}k_{78}k_{79} + k_{75}k_{76} + k_{75}k_{77}k_{78} + k_{75}k_{77}k_{79} + k_{75}k_{78}k_{79} + k_{75}k_{79} + k_{76}k_{77} + k_{76}k_{78}k_{79} + k_{76}k_{78}k_{79} + k_{76}k_{79} + k_{77}k_{78} + k_{77}k_{79} + k_{78} + k_{79} + k_{9} \]