A Refinement of "A Key-recovery Attack on 855-round Trivium" From CRYPTO 2018

Ximing Fu[1], and Xiaoyun Wang[2,3,4], and Xiaoyang Dong[2], and Willi Meier[5], Yonglin Hao[6], and Boxin Zhao[3,4]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
fuxm07@foxmail.com
[2] Institute for Advanced Study, Tsinghua University, Beijing 100084, China
xiaoyunwang,xiaoyangdong@tsinghua.edu.cn
[3] School of Mathematics, Shandong University, Jinan 250100, China
[4] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Jinan 250100, China,
[5] FHNW, Windisch, Switzerland
willi.meier@fhnw.ch
[6] State Key Laboratory of Cryptology, Beijing, China
haoyonglin@yeah.net

Abstract. At CRYPTO 2018, we proposed a method to reduce the Boolean polynomial of 855-round Trivium [1]. By multiplying a polynomial reduction factor, the output Boolean polynomial is simplified. Based on this method, a 855-round key-recovery attack on Trivium is introduced. In addition, we also give a practical attack on 721-round Trivium to show some rationality and evidence.
However, Yonglin Hao et al. [2], find some errors in the 721-round attack recently. As a correction, we propose some new right 721-round example attacks based on our method proposed at CRYPTO 2018.

Keywords: Trivium, Nullification Technique, Polynomial Reduction, IV representation, Key-recovery attack

1 The Polynomial Reduction Technique

In CRYPTO 2018, Fu et al. [1] announced a new key-recovery attack on 855-round Trivium based on the following polynomial reduction technique.

Lemma 1. *[1] Suppose z is the output polynomial of a cipher, and*

$$z = P_1P_2 + P_3. \tag{1}$$

Then the polynomial can be reduced to a simpler one $(1 + P_1)z = (1 + P_1)P_3$ by multiplying $1 + P_1$ in both sides of Eq. (1) if $\deg(P_1P_2) > \deg((1 + P_1)P_3)$.

1. Right guess: $(1 + P_1)z = (1 + P_1)P_3$

2. Wrong guesses: $(1+P_1')z = (1+P_1')P_1P_2 + (1+P_1')P_3$

The key point is to select a proper reduction factor P_1, that could introduce a polynomial and degree reduction. There are 3 criteria to determine P_1:

1. the frequency of P_1 in high degree state terms is high;
2. the degree of P_1 is low;
3. the equivalent key guesses in P_1 are minimized.

Compute the degree of $(1+P_1)P_3$ as d, then $d+1$-dimensional cubes can serve as distinguishers. Then in the online phase, we guess the partial key bits in P_1 and compute the cube sum of $(1+P_1)z$ over $d+1$ cubes: for right guess, the result is always 0; for wrong guesses, the results are 0-1 random.

2 The Mistake in the Example Attacks on 721-round Trivium in Our CRYPTO 2018 paper

Recently, Hao et al. [2] pointed out the errors in the 721-round example attack in our paper [1].

In the wrong example attack on 721-round Trivium, we use 37 freedom variables, i.e. set $v_{2\cdot j+1} = 0$ for $j \in [0, 39]$ and $v_{58} = v_{64} = v_{72} = 0$, others are free variables. We choose s_1^{290} as P_1, write $z_{721} = s_1^{290}P_2 + P_3$. By multiplying $1+s_1^{290}$ with z_{721}, we get $(1+s_1^{290})z_{721} = (1+s_1^{290})P_3$. Then, we prove the degree of $(1+s_1^{290})z_{721}$ is lower than 32, while the degree of z_{721} is evaluated to be 36 by our degree evaluation algorithm (Algorithm 2 in [1]). So we wrongly believed that we got a proper reduction factor P_1 for 721-round attack. However, Hao et al. pointed out that the degree of z_{721} is only 29, so the 721-round example attack is against the lemma 1 and there is no polynomial or degree reduction by multiplying $(1+s_1^{290})$.

The reasons for our mistake come from three aspects:

1) The first reason is the weak diffusion of 721-round Trivium. For 855-round Trivium, we could assume that the degree could reach 75 (75 free IV bits and others are nullified) which is also proved by Hao et al.'s paper [2]. However, in 721-round Trivium, the degree is relatively low (29-degree given 37 free IV bits). So we have to do more accurate degree evaluation.

2) The second reason is that, our degree evaluation is relatively rough, the upper bound degree is relatively high, which is pointed out in Hao et al.'s paper [2].

3) The third reason is that, we forget to test the 32-dimension cubes under wrong key guessing, which leads to such mistake.

All in all, the property of degree reduction of the output bit by multiplying a $(1+P_1)$ is true as shown by our new 721-round example attacks as following. However, finding a proper P_1 is not easy.

3 The New Key-recovery Attacks on 721-round Trivium

In the new attack on 721-round Trivium, we nullified 80-29=51 IV bits and only 29 free IV bits are considered. Then we use techniques in our CRYPTO 2018 paper to find $P_1 = s_1^{221}$. In addition, we find the 29-degree term is in z_{721}, i.e., the accurate degree of z_{721} is 29. However, in $(1 + s_1^{221})z_{721}$, there is no such 29-degree term, i.e., degree of $(1 + s_1^{221})z_{721}$ is lower than 29. That means we have found a proper reduction factor $P_1 = s_1^{221}$ to simplify the output polynomial z according to Lemma 1.

Moreover, when given the wrong key guessing in s_1^{221}, the 29-degree term also appears, which means under the wrong key guessing the output polynomial z is not reduced. So this 29-dimension cube could be served as key-recovery distinguisher. Finally, we find more than 17 such 29-dimension cubes. Since, the number of involved key bits in s_1^{221} is 17, whose indexes are $\{8, 9, 46, 71, 72, 73, 59, 60, 52, 10, 17, 18, 19, 1, 26, 27, 28\}$, we only list 17 such cubes in Table 1. The source code for this test is in `https://github.com/dongxiaoyang/721R_Trivium_Test`.

Table 1. New Example 29-Dimensional Cubes in 721-round Attack

0,4,10,16,20,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,2,4,8,14,16,20,24,28,30,32,34,36,38,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,4,6,10,14,16,20,24,28,30,32,34,36,38,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,4,8,10,14,16,20,24,28,30,32,34,36,38,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,4,10,16,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,52,54,56,60,62,66,68,70,74,76
0,2,4,8,16,20,22,24,28,30,32,34,36,38,40,42,44,46,48,50,54,56,60,62,66,68,70,74,76
0,4,8,14,16,20,22,24,28,30,32,34,36,38,40,42,44,46,48,52,54,56,60,62,66,68,70,74,76
0,4,8,10,14,20,24,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,4,10,14,20,22,24,26,28,30,32,34,36,38,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,2,4,8,14,16,20,22,24,28,30,32,34,36,38,42,44,46,48,50,54,56,60,62,66,68,70,74,76
0,4,10,14,16,20,24,26,28,30,32,34,36,38,40,42,44,46,48,52,54,56,60,62,66,68,70,74,76
10,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
4,10,16,18,20,22,24,26,28,30,32,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,2,4,8,14,16,20,24,28,30,32,34,36,38,40,42,44,46,48,50,54,56,60,62,66,68,70,74,76
0,4,10,14,16,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,2,4,16,20,22,24,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76
0,2,4,8,16,20,24,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,60,62,66,68,70,74,76

In addition, for each 29-dimension cube in Table 1, we test the cube sums for 72 random keys. For the wrong key, we just XOR a random 17-bit number to the corresponding 17 positions of the right key, which are involved in the s_1^{221}. As show in Table 2, we explain the first cube and the others act in a similar way.

1. For cube sums of z_{721}, 29 cube sums out of 72 are 1.
2. For cube sums of $(1 + s_1^{221})z_{721}$ under right key guessing, the 72 cube sums are always 0.

3. For cube sums of $(1 + s_1^{221'})z_{721}$ under wrong key guessing, 29 cube sums out of 72 are 1.

Table 2. Random Test of the New 721-round Attack

Cube index	z_{721}	$(1 + s_1^{221})z_{721}$	$(1 + s_1^{221'})z_{721}$
1	29	0	29
2	28	0	1
3	39	0	32
4	28	0	20
5	33	0	33
6	24	0	24
7	43	0	43
8	32	0	8
9	40	0	8
10	41	0	12
11	44	0	21
12	23	0	4
13	20	0	4
14	45	0	12
15	31	0	20
16	36	0	36
17	29	0	29

In order to prove the effectiveness of our method, we find another new P_1 for 721 round attack using different nullification scheme and different degree. For example, we find a new $P_1 = s_0^{164}$, and a 31-dimension cube, whose indexes are $\{0, 4, 14, 16, 20, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76\}$, other bits are nullified. The degree of z_{721} is 31, the degree of $(1+P_1)z_{721}$ is lower than 31. Moreover, if we multiply a wrong $(1+P_1')$, the degree of $(1 + P_1')z_{721}$ is still 31.

We give 960 random tests for the new cube:

1. Without multiplying $1 + P_1$ with z_{721}, the number of 1-cube-sums of z_{721} is 519 out of 960 tests.
2. Multiplying $1 + P_1$ with z_{721}:
   (a) Under the right key guessing in P_1, the number of 1-cube-sums of $(1 + P_1)z_{721}$ is 0 out of 960 tests;
   (b) Under the wrong key guessing in P_1, the number of 1-cube-sums of $(1 + P_1')z_{721}$ is 249 out of 960 tests;

4 Conclusion

In this paper, we give some correct 721 example attacks which support the basic ideas of Fu et al.'s CRYPTO 2018 paper [1]. We restate the idea as follows:

suppose the output polynomial of Trivium is $z = P_1P_2 + P_3$, by finding a proper reduction factor P_1 and multiplying $(1+P_1)$ with z, the output polynomial could be simplified and the degree could be reduced.

Acknowledgement

We would like to thank Qingju Wang for the fruitful discussion on this short paper.

References

1. Fu, X., Wang, X., Dong, X., Meier, W.: A key-recovery attack on 855-round trivium. In: Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II. pp. 160–184 (2018), https://doi.org/10.1007/978-3-319-96881-0_6
2. Hao, Y., Jiao, L., Li, C., Meier, W., Todo, Y., Wang, Q.: Observations on the dynamic cube attack of 855-round trivium from crypto'18. Cryptology ePrint Archive, Report 2018/972 (2018), https://eprint.iacr.org/2018/972