The Problem of State Recovering Attack Against Trivium

Shiyong Zhang[1, a *], Gongliang Chen[1,b] and Jianhua Li[1,c]

[1]School of Information Security Engineering,

Shanghai Jiaotong University, China

[a]poetzhangzi@sjtu.edu.cn, [b]chengl@sjtu.edu.cn, [c]lijh888@sjtu.edu.cn

Keywords: Trivium, Security, State Recovering Attack.

Abstract. Trivium is a notable light-weight synchronous stream cipher submitted to the European eSTREAM project in April 2005. State recovering attack is the best known attack to Trivium. In this paper, we study the structure of Trivium and point out the equations used in the state recovering attack are linearly dependent. The number of the equations is not enough to derive the exact solution. Then the revisional state recovering attack will be given to correct the problem of origin attack. We show that the internal state of Trivium will be recovered in time around $2^{88.8}$, and the keystream has the length of $2^{57.8}$. Therefore, the revisional attack is still faster than the exhaustive search

Introduction

Trivium is a notable light-weight synchronous stream cipher designed by Christophe De Canniere and Bart Preneel, which is submitted to the European eSTREAM project in April 2005 [1]. This algorithm is designed to be both efficient and secure. During 3 phases of eSTREAM evaluation on the stream cipher proposals, the performance of Trivium is outstanding compared with other stream ciphers such as A5. Trivium outperforms other eSTREAM candidates considered in the paper in terms of the two most important optimization criteria, minimum area and maximum throughput to area ratio, by a factor of at least two [2].

Since now, there are many works about the security of Trivium. The first known result is actually given on the eSTREAM discussion forum where the complexity to recover the internal state from given keystream is argued to be 2^{135} which is much slower than exhaustive search [3]. Raddum presents a technique to solve systems of equations associated with Trivium [4]. But his attack is very complex when applied to the full cipher and the attack complexity is 2^{164}. Borghoff presents a numerical attack [5]. However the estimated time complexity of this attack is about $2^{63.7}$ seconds. The cube attack proposed by Pierre-Alain Fouque and Thomas Vannet requires 2^{68} steps to break a variant of Trivium. However, the number of initialization rounds is reduced to 799 [6,7]. Maximov studies two attacks on Trivium , which are statistical tests and state recovering attack [8]. Statistical tests are not good since the attacks are no faster than exhaustive search. State recovering attack is the best known attack to Trivium. The internal state of the full Trivium can be recovered in time around $2^{83.5}$.

In this attack, Maximov try to recover one third of the states with 96 linear equations in the phase I and the others with 192 linear equations in the phase II. However, this attack contains some serious problems. In this paper, we point out that both the first 96 equations and the next 192 equations are linearly dependent. The number of linear equations is not enough to derive the exact solution.

The following part of the paper is organized as follows. The algorithm of Trivium will be described in section 2. Section 3 point out the problem of the state recovering attack. The revisional attack will be proposed in section 4. The conclusion will be given in section 5.

Trivium Algorithm

Trivium [1] is designed to generate up to 2^{64} bits of key stream from an 80-bit secret key Key and an 80-bit initial value IV. The process consists of two phases: first the internal state of the cipher is

initialized using Key and IV, then the state is repeatedly updated and used to generate key stream bits. There are 288 bits in the internal state, which is denoted as $\boldsymbol{s}=(s_1,s_2,\mathrm{L}\ ,s_{288})$. Trivium is designed as hardware oriented. Evaluation on implementation of Trivium for low-power application in RFID system is given by Feldhofer with comparison to AES-128 [10]. Synthesis result of Trivium is better than that of AES-128.

Trivium has three rounds with similar structure. Denote the intermediate variable as t_1,t_2,t_3 and the output stream as $\boldsymbol{z}=(z_1,z_2,\mathrm{L}\ ,z_N)$,with N standing for the number of output bits. The process of Trivium is shown as Algorithm 1, where $u_1<u_2<n_1<u_3<u_4<n_2<u_5<u_6<n_3$ are the parameters, which is shown in Table I:

Algorithm 1 Trivium-model Algorithm

for i=1 to N do

$t_1 \leftarrow s_{3u_1}+s_{3n_1}$

$t_2 \leftarrow s_{3u_3}+s_{3n_2}$

$t_3 \leftarrow s_{3u_5}+s_{3n_3}$

$t_1 \leftarrow t_1+s_{3n_1-2}\cdot s_{3n_1-1}+s_{3u_4}$

$t_2 \leftarrow t_2+s_{3n_2-2}\cdot s_{3n_2-1}+s_{3u_6}$

$t_3 \leftarrow t_3+s_{3n_3-2}\cdot s_{3n_3-1}+s_{3u_2}$

$(s_1,s_2,\mathrm{L}\ ,s_{3n_1}) \leftarrow (t_3,s_1,\mathrm{L}\ ,s_{3n_1-1})$

$(s_{3n_1+1},s_{3n_1+2},\mathrm{L}\ ,s_{3n_2}) \leftarrow (t_1,s_{3n_1+1},\mathrm{L}\ ,s_{3n_2-1})$

$(s_{3n_2+1},s_{3n_2+2},\mathrm{L}\ ,s_{3n_3}) \leftarrow (t_2,s_{3n_2+1},\mathrm{L}\ ,s_{3n_3-1})$

end for

u_1	u_2	n_1	u_3	u_4	n_2	u_5	u_6	n_3
22	23	31	54	57	59	81	88	96

Table I: Parameters of Trivium

Denote the internal state bits of Trivium at time t as $\boldsymbol{s}(t)=(s_1(t),s_2(t),\mathrm{L}\ ,s_{3n_3}(t))$.At time t, z_t can be expressed as:

$$\begin{aligned} z_t &= s_{3u_1}(t)+s_{3n_1}(t)+s_{3u_3}(t)+s_{3n_2}(t)+s_{3u_5}(t)+s_{3n_3}(t) \\ &= s_{66}(t)+s_{93}(t)+s_{162}(t)+s_{177}(t)+s_{243}(t)+s_{288}(t) \end{aligned} \quad (1)$$

For simplicity in further derivations we define s_i as $s_i(1),1\le i\le 3n_3$, $T_j(t)=\{s_k(t)|k\equiv j(\mathrm{mod}\,3)\}$, j=0,1,2. Then z_t can be expressed as a non-linear function of the variables s_i. Obviously the degree of equations will increase. In fact, the first three equations are linear. Then for $67\le i\le 148$ the equations have the degree of two and for $149\le i\le 214$ the degree becomes three and so on.

Problem of State Recovering Attack

In the state recovering attack, given the output stream $\boldsymbol{z}$, the target is to recover the internal state

of the cipher. Since all blocks of the cipher are divisible by 3. And the transition of the internal state from time t to time t+1 is a linear transformation of the subset $T_{t+2 \bmod 3}(t)$, plus a minor one bit disturbance from the adjacent two subsets. Therefore, in the state recovering attack, the main idea is to guess the state $T_0(t)$ at some time t in phase I, then recover the rest of the bits in phase II.

In detail, denote $\boldsymbol{z}'$ as the linear part of $\boldsymbol{z}$, then all the variables of z'_{3t} belong to $T_0(t)$. Through the guess of a set of terms and the sum of some terms, Maximov try to increase the number of linear equations to n_3. After $T_0(t)$ is guessed and derived correctly. Since a set of terms is known, some of the remaining equations are linear. Then Maximov collect $2n_3$ equations on $T_1(t)$ and $T_2(t)$ to recover the remaining 2/3 of the state.

However, the number of linear equations is not enough to derive the exact solution, not only in the phase I but also in the phase II. In fact, in phase I, we have the theorem as follows:

Theorem 3.1: $(z'_1, z'_4, \ldots, z'_{3n_3-2})$ are linearly dependent. The rank of $(z'_1, z'_4, \ldots, z'_{3n_3-2})$ is n_3-3.

Proof: The characteristic polynomial of Trivium [1] can be expressed by:

$$\begin{aligned} f(x) &= x^{288}+x^{219}+x^{210}+x^{201}+x^{141}+x^{132}+x^{123}+x^{87}+x^{72}+x^{60}+x^{54}+x^{45}+x^{42}+x^{27}+x^{15}+1 \\ &= (x^3+1)^3 \cdot g(x^3) \end{aligned} \quad (2)$$

where $g(\mathrm{g})$ is a primitive polynomial.

Therefore, the characteristic polynomial of $s_i(t)$ is also $f(x)$. The characteristic polynomial of $s_i(3t+1)$ can be expressed by:

$$f'(x) = (x+1)^3 \cdot g(x) \quad (3)$$

Since z'_t is the linear combination of $s_i(t)$, the characteristic polynomial of z'_{3t+1} is also $f'(x)$. Therefore, z'_{3t+1} are linearly dependent. And the rank is $n_3-3=93$.

Another problem is in phase II. Having the $T_0(t)$ and some terms derived correctly, Maximov will collect $2n_3$ equations on $T_1(t)$ and $T_2(t)$ to recover the remaining $2n_3$ of the state. However, these $2n_3$ equations are also linearly dependent. And the rank of the equation set will be changed with the different solutions of $T_0(t)$. Obviously, if all the variables in $T_0(t)$ are zero, the rank of the equation set will be $(n_3-3)\times 2=186$. That is to say, 6 of 192 equations are useless for the recovery of the remaining state. If all the variables in $T_0(t)$ are one, through the simulation results, the rank will be changed to 188. Then 4 equations cannot be used. Therefore, in the phase II, the remaining state will not be recovered by the collected $2n_3$ linear equations. We still need to guess some state.

In the next section, the error of the origin state recovering attack will be corrected and the revisional state recovering attack will be given.

Revisional State Recovering Attack And Attack Results

In the revisional state recovering attack, given the output stream $\boldsymbol{z}$, our target is to recover the internal state of the cipher. The main idea is as same as the origin attack. We first guess the state $T_0(t)$ at some time t, then recover the rest of the bits.

In order to receive more linear equations, we consider a set of terms:

$$
\begin{aligned}
&s_{3n_1-2}(3i+1)\cdot s_{3n_1-1}(3i+1)=s_{91}(3i+1)\cdot s_{92}(3i+1), i=0,1,\mathrm{L}\ , g_a-1\\
&s_{3n_2-2}(3i+1)\cdot s_{3n_2-1}(3i+1)=s_{175}(3i+1)\cdot s_{176}(3i+1), i=0,1,\mathrm{L}\ , g_b-1\\
&s_{3n_3-2}(3i+1)\cdot s_{3n_3-1}(3i+1)=s_{286}(3i+1)\cdot s_{287}(3i+1), i=0,1,\mathrm{L}\ , g_c-1
\end{aligned}
\tag{4}
$$

where g_a, g_b, g_c are chosen parameters. If we guess these terms, then the number of linear equations will increase. For example, if we guess $s_{175}\cdot s_{176}=s_{286}\cdot s_{287}=0$, z_{67} will be a linear equation. Then the count d of linear equations can be expressed as:

$$
d=\min\left\{u_1+g_a, u_3-n_1+g_b, u_5-n_2+g_c\right\} \tag{5}
$$

The most probable guess would be that all the terms are zeros, since $P\left\{s_i \,\&\, s_j=0\right\}=0.75$. Suppose $g_a+g_b+g_c$ terms produces zeros, the probability of such an event is $p_g=0.75^{g_a+g_b+g_c}$. The keystream is required to be of length $O\left(p_g^{-1}\right)$.

For the remaining nonlinear equations, the linear part consists of the bits from $T_0(t)$, and the nonlinear part is the sum of w terms, for some small w. Since the outcome of each of them is biased, then their sum is biased as well. Denote p_w as the probability that the sum of w terms is zero, then we have $p_w=\sum_{i=0}^{\left\lfloor \frac{w}{2} \right\rfloor}\binom{w}{2i}0.75^{w-2i}0.25^{2i}$ [8]. Denote l_w be the number of nonlinear equations with the sum of w terms. Then the time complexity to recover l_w bits is $p_w^{l_w}$, the keystream length is increased by the ratio $p_w^{-l_w}$. The total probability of such an event is $p_l=\prod_{w=1}^{\infty}p_w^{l_w}$. Since only the first $n_3-3=93$ variables of the linear part of $\boldsymbol{z}$ are linearly independent, if the number of the linear equations are greater than 93, only the first 93 equations are valuable. The time of phase I can be expressed as:

$$
time_I=2^{\max\left\{n_3-d-\sum_{w=1}^{\infty}l_w,3\right\}}\cdot p_g^{-1}\cdot p_l^{-1} \tag{6}
$$

After $T_0(t)$ is guessed and derived correctly. To recover the remaining 2/3 of the state we need to collect a number of equations on $T_1(t)$ and $T_2(t)$, enough to derive the exact solution. In fact when (10) is known, some of the remaining equations are linear. Since $2n_3$ equations are linearly dependent, we should find the lower bound of the rank. However, it is difficult to traverse all the solutions to confirm the minimum linearly independent subset. Through the extensive experimental data tests, we make a conjecture as follow:

Conjecture 4.1: Traversing all the solutions of $T_0(t)$, the minimum rank of $2n_3$ equations is $2n_3-6$.

If we choose the suitable parameters g_a, g_b, g_c and guess 6 variables of the state, we can recover all the initial state of Trivium. When all the parameters are fixed, a particular scenario can be described. The breaking complexity of the attack can be expressed as:

$$
\begin{aligned}
time &= time_I\cdot 2^6\\
&= 2^{\max\left\{n_3-d-\sum_{w=1}^{\infty}l_w,3\right\}+6}\cdot p_g^{-1}\cdot p_l^{-1}
\end{aligned}
\tag{7}
$$

The algorithm of the state recovering attack on Trivium is given in Algorithm 2.

Algorithm 2 Revisional State Recovering Attack on Trivium

for t=0 to $\lceil time \rceil$ do

Assume $s_{3n_1-2}(3i+1)\cdot s_{3n_1-1}(3i+1)=s_{3n_2-2}(3i+1)\cdot s_{3n_2-1}(3i+1)=s_{3n_3-2}(3i+1)\cdot s_{3n_3-1}(3i+1)=0$, $0\le i<g_a, 0\le j<g_b, 0\le k<g_c$.

Collect d linear equations on $T_0(t)$ with probability 1 and $\sum_{w=1}^{\infty} l_w$ more linear equations with the total probability p_l.

Guess s_{93}, s_{177}, s_{288}.

for each guess of the remaining $n_3-d-\sum_{w=1}^{\infty} l_w-3$ bits in $T_0(t)$ do

Derive the state of $T_0(t)$ using the linear equations collected in the last step.

Collect $2n_3$ linear equations on $T_1(t)$ and $T_2(t)$.

for each guess of the fixed 6 values of the remaining state do

Recover the state of $T_1(t)$ and $T_2(t)$ by the linear equations collected in the last step, and verify the solution in time $O(1)$.

end for

end for

end for

The parameters g_a, g_b, g_c cannot be too small in order to derive enough linear equations on $T_1(t)$ and $T_2(t)$. However, the larger the parameters, the greater the keystream. On the other hand, the increase of l_w will generate more linear equations, but when $w>4$, p_w is very close to 0.5. Therefore, it is important to search suitable parameters. We choose parameters (g_a, g_b, g_c)=(43,34,39) and (l_1, l_2, l_3, l_4)=(5,5,4,1). With these parameters, Trivium will be broken in time around $2^{88.8}$ and the keystream has the length of $2^{57.8}$. The result is shown in Table II.

Parameter	Trivium
$g_a : g_b : g_c$	43:34:39
d	56
p_g	$2^{-48.1}$
$l_1 : l_2 : l_3 : l_4$	5:5:4:1
p_l	$2^{-9.7}$
Keystream	$2^{57.8}$
Breaking Complexity	$2^{88.8}$

Table II: Trivium Under Revisional State Recovering Attack

From the result, the breaking complexity of the revisional state recovering attack is $2^{88.8}$, while the complexity of the origin attack is $2^{83.5}$. However, the origin attack exists a series of problems. And the keystream reduces from $2^{61.5}$ to $2^{57.8}$. On the other hand, the exhaustive search requires about 2^{90} [8]. Therefore, the revisional state recovering attack is still a little faster than the exhaustive search.

Conclusion

In this paper, we study the internal structure of Trivium and state recovering attack which is the best known attack to Trivium. We point out that the equations used in the state recovering attack are linearly dependent. Therefore, the number of the equations is not enough to derive the exact solution. The revisional state recovering attack is proposed in section 4.

We show that the internal state of Trivium will be recovered in time around $2^{88.8}$, and the keystream has the length of $2^{57.8}$. Therefore, the revisional attack is still faster than the exhaustive search.

Acknowledgment

This work was supported in part by International Researcher Exchange Project of National Science Foundation of China and Centre national de la recherche scientifique de France (NSFC-CNRS) under Grant No. 61211130104 and National Science Foundation of China under Grants No. 61271220.

References

[1] C. De Cannire and B. Preneel. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project (http://www.ecrypt.eu.org/stream), Report 2005/030, April 2005.

[2] K. Gaj, G. Southern and R. Bachimanchi, "Comparison of hardware performance of selected phase II eSTREAM candidates", http://www.ecrypt.eu.org/stream/papersdir/200 7/026.pdf, 2007.

[3] eSTREAM Discussion Forum. "A reformulation of trivium". created on 02/24/06 12:52PM, 2005. http://www.ecrypt.eu.org/stream/phorum/read.php?1,448.

[4] H. Raddum, "Cryptanalytic results on Trivium", http://www.ecrypt.eu.org/stream/papersdir/2006/039.ps, 2007.

[5] J. Borghoff, L. R. Knudsen and M. Stolpe, "Bivium as a mixed-integer linear programming problem", in LNCS vol.5921, M. G. Parker Eds. Heidelberg: Springer, 2009, pp. 133-152, 2009.

[6] Dinur,I.,Shamir,A. "Cube Attacks on weakable Black Box Polynomials." in Joux,A. (ed.) EUROCRYPT 2009. LNCS, Springer, Heidelberg, vol. 5479, pp. 278-299, 2009.

[7] Fouque, P.A., Vannet, T, "Improving Key Recovery to 784 and 799 rounds of Trivium using Optimized Cube Attacks." inMoriai, S. (ed.) FSE 2013. LNCS, Springer, Heidelberg, vol. 8424, pp. 502-517, 2014.

[8] A. Maximov, A. Biryukov. "Two trivial attacks on TRIVIUM", in SASC2007: The State of the Art of Stream Ciphers, pp. 1-16, 2007.

[9] M. Feldhofer and J. Wolkerstorfer. "Hardware Implementation of Symmetric Algorithms for RFID Security". in RFID Security: Techniques, Protocols and System-on-Chip Design, pp. 373-415. Springer, September 2008.

[10]M. Feldhofer. "Comparison of Low-Power Implementations of Trivium and Grain". \textit{Workshop on The State of the Art of Stream Ciphers (SASC2007)} pp 236-246, 2007.