Linear Sequential Circuit Approximation of the TRIVIUM Stream Cipher

Shahram Khazaei[1] Mehdi Hassanzadeh[2]

[1] Electrical Engineering Dept., Sharif Univ. of Tech., Iran
khazaaei@yahoo.com
[2] Raymand Information and Communication Cryptographers
Tehran, Iran
hasanzadeh@raymandcrypto.ir
Sept. 2005

Abstract

TRIVIUM is the simplest ECRYPT Stream Cipher project Candidate which deals with key and IV of length 80. Using the sequential Circuit Approximation method, introduced by Golic in 94, we derive a linear function of consecutive keystream bits which is hold with correlation coefficient of about 2^{-72}. This shows that TRIVIUM is strong against linear sequential circuit approximation attack in spite of the extra simplicity of its output function and next-state function. It seems very hard to find a biased linear function of consecutive output bits which leads to a successful distinguishing attack on TRIVIUM.

1. Introduction

Golic has shown that for a binary keystream generator with M bits of memory whose initial state is chosen uniformly at random, there exists a linear function of at most M+1 consecutive output bits which is an unbalanced function of the initial state variables [5 or 6]. He also developed an effective method for the linear model determination based on linear sequential circuit approximation of autonomous finite-state machines. The linear function of consecutive output bits produces an unbalanced sequence to which one can apply the standard chi-square frequency statistical test. The test is successful if and only if the length of the sequence is chosen to be inversely proportional to the square of the correlation coefficient[1]. If the key length is k, the statistical weakness is effective if and only if the correlation coefficient is greater than $2^{-k/2}$. In this paper, using Golic's method, we extract the linear sequential circuit approximation of the TRIVIUM stream cipher [3]- the simplest ECRYPT Stream Cipher project Candidate [1]. We derive a linear function of consecutive output bits which is hold with correlation coefficient of about 2^{-72}. It seems very hard to find a linear function of consecutive output bits with correlation coefficient greater than 2^{-40} to have a successful distinguishing attack. A similar result has been mentioned in TRIVIUM specification [3] but not explained in details. However, the

[1] The correlation coefficient of the random variable x is defined as $\varepsilon = 1 - 2\Pr\{x = 1\}$.

TRIVIUM designers derived these results in a slightly different way and we were suggested to publish these results [4].
These results show that TRIVIUM is strong against linear sequential circuit approximation attack, in spite of the linearity of its output function and all of the components of its next-state function except three of them which also have very near distances from some linear functions.
The most important negative problem with the TRIVIUM is its period which is not well understood and even recently some states of period three have been found [2].
The paper is organized as follows. In sections 2 and 3 a brief descriptions of the TRIVIUM stream cipher and linear sequential circuit approximation are respectively given. We derive the linear sequential circuit approximation of the TRIVIUM in section 4 and propose the correlation coefficient analysis in section 5. The paper is concluded in section 6.

2. A Brief Description of TRIVIUM

TRIVIUM is a very simple hardware oriented synchronous stream cipher proposed as a candidate to the ECRYPT Stream Cipher Project [3]. TRIVIUM generates up to 2^{64} bits of key stream from an 80-bit secret key and an 80-bit initial value (IV). The proposed design contains a 288-bit internal state denoted by $(s_1,\ldots, s_{288})$. The key stream generation consists of an iterative process which extracts the values of 15 specific state bits and uses them both to update 3 bits of the state and to compute 1 bit of key stream z_t. The state bits are then rotated and the process repeats itself until the requested $N \le 2^{64}$ bits of key stream have been generated. A complete description is given by the following simple pseudo-code:

for t = 1 to N do
$t_1 \leftarrow s_{66} + s_{93}$
$t_2 \leftarrow s_{162} + s_{177}$
$t_3 \leftarrow s_{243} + s_{288}$
$z_t \leftarrow t_1 + t_2 + t_3$
$t_1 \leftarrow t_1 + s_{91} \,.\, s_{92} + s_{171}$
$t_2 \leftarrow t_2 + s_{175} \,.\, s_{176} + s_{264}$
$t_3 \leftarrow t_3 + s_{286} \,.\, s_{287} + s_{69}$
$(s_1, s_2,\ldots, s_{93}) \leftarrow (t_3, s_1,\ldots, s_{92})$
$(s_{94}, s_{95}, \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
$(s_{178}, s_{179}, \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$
end for

3. A Brief Description of the Linear Sequential Circuit Approximation

Keystream generators for stream cipher applications can generally be realized as autonomous finite-state machines whose initial state and possibly structure as well depend on a secret key. A binary autonomous finite-state machine is defined by

$$S_t = F(S_{t-1}) \quad t \geq 1 \tag{3-1}$$

$$z_t = f(S_t) \qquad t \geq 1 \tag{3-2}$$

where F: $GF(2)^M \rightarrow GF(2)^M$ is the next-state vector Boolean function, f: $GF(2)^M \rightarrow GF(2)$ is the output Boolean function, $S_t = (s_{t,1}, s_{t,2}, \ldots, s_{t,M})^T$ is the state vector at time t, $S_0 = (s_{0,1}, s_{0,2}, \ldots, s_{0,M})^T$ is the initial state, and $\{z_t\}$ is the output keystream sequence[2].
We just consider the case that the key merely controls the initial state, and therefore, next state function and output function are known.
Golic has shown that there exist a linear function of at most M+1 consecutive output bits $L(z_t, z_{t+1}, \ldots, z_{t+M})$ which is an unbalanced function of the initial state variables. Its probability distribution is independent of time t if the next state function is balanced. This statement has been proposed as a Theorem in [5], which has come in the following.

Theorem Let the next-state function of a binary autonomous finite state machine with M bits of memory be balanced. Then there exists a linear function L of at most M+1 consecutive output bits $L(z_t, z_{t+1}, \ldots, z_{t+M})$ which is an unbalanced function of the initial state variables for each $t \geq 1$. Moreover, the correlation coefficient between $L(z_t, z_{t+1}, \ldots, z_{t+M})$ and the constant zero function is the same for each t.
The linear function L of consecutive output bits produces an unbalanced sequence to which one can apply the standard chi-square frequency statistical test to make a distinguishing attack. The test is successful if and only if the length of the sequence is chosen to be inversely proportional to the square of the correlation coefficient. If the key length is k, the statistical weakness is effective if and only if the correlation coefficient is greater than $2^{-k/2}$.
Golic, also, has developed an efficient procedure for finding unbalanced linear functions of the output which is based on the linear sequential circuit approximation approach. To this end, he first decomposes the output Boolean function and each of the Boolean functions in the next-state function of the keystream generator into the sum of linear functions and an unbalanced Boolean function. Then, by virtue of the obtained linear approximations, he puts the basic equations (3-1) and (3-2) into the form.

$$S_t = AS_{t-1} + \Delta(S_{t-1}) \quad t \geq 1 \tag{3-3}$$

$$z_t = BS_t + \varepsilon(S_t) \qquad t \geq 1 \tag{3-4}$$

where, considering S_t as an M×1 vector, A is an M×M matrix and B is a 1×M vector, Δ is an M×1 noise vector and ε is a scalar noise component.
By using the generating function technique, Golic then solves the linear recurrence equations and thus reaches to his desire, that is, a linear function of at most m+1 consecutive output bits that is expressed as the sum of unbalanced functions of the initial state variables. He shows that the linear function corresponds to the minimal polynomial[3]

---

[2] ^{T} denotes the matrix transposition operation.

[3] The minimal polynomial of a given square matrix *A*, is the minimal degree non-zero polynomial $\varphi(x) = \sum_{k=0}^{r} \varphi_k x^k$ where $\varphi(A) = \sum_{k=0}^{r} \varphi_k A^k = 0$. Conceptually, it is assumed that $x^0 = 1$ and similarly, $A^0 = I$ where I is the identity matrix whose dimension is the same as *A*.

of A, the state transition matrix of the linear sequential circuit.
The conditions of next state function and output function independence of the secret key, and balance of the next state function is well satisfied for the TRIVIUM cipher.

4. Linear Sequential Circuit Approximation of TRIVIUM

In this section, we derive the linear sequential circuit approximation of the TRIVIUM stream cipher from basis. For ease of reference, here, we have listed all matrixes and vectors used in this section as well as their dimensions (m = 282).

vector or matrix name	dimention(size)
S_t	288×1
A	288×288
B, e_i	1×288
Δ_t	3×1
H	288×3
C_τ	1×3
C', C'', C''', Z_t	$(m+1)\times 1$
$\Delta'_t, \Delta''_t, \Delta'''_t, \phi$	$1\times (m+1)$

For the TRIVIUM stream cipher we have M = 288. Since the output function and all the components of the next-state function, except three of them, are linear for the TRIVIUM stream cipher, decomposition of these functions is performed easily. It is sufficient to just consider the best linear approximation of the 1st, 94th and 178th component of the next-state function given in the follows.

$$s_{t+1,1} = s_{t,243} + s_{t,288} + s_{t,286} \cdot s_{t,287} + s_{t,69} \qquad (4\text{-}1)$$

$$s_{t+1,94} = s_{t,66} + s_{t,93} + s_{t,91} \cdot s_{t,92} + s_{t,171} \qquad (4\text{-}2)$$

$$s_{t+1,178} = s_{t,162} + s_{t,177} + s_{t,175} \cdot s_{t,176} + s_{t,264} \qquad (4\text{-}3)$$

The best linear approximation of the above functions are achieved by eliminating the quadratic terms which is hold with probability equal to ¾. Therefore, the linear approximations (3-3) and (3-4) for the TRIVIUM could be written as follows

$$S_t = AS_{t-1} + H\Delta_t \qquad t\geq 1 \qquad (4\text{-}4)$$

$$z_t = BS_t \qquad t\geq 1 \qquad (4\text{-}5)$$

where, $H = [h_{i,j}]$ is a 288×3 matrix whose all entries are zero, except $h_{1,1}$, $h_{94,2}$ and $h_{178,3}$ which are one, $\Delta_t = [\delta'_t \quad \delta''_t \quad \delta'''_t]^T$ is the 3×1 noise vector corresponding to the 1st, 94th and 178th component of the next-state function, and A and B are as follows.[4]

$$A = \begin{bmatrix} e_{69} + e_{243} + e_{288} \\ e_1 \\ \vdots \\ e_{92} \\ e_{66} + e_{93} + e_{171} \\ e_{94} \\ \vdots \\ e_{176} \\ e_{162} + e_{177} + e_{264} \\ e_{178} \\ \vdots \\ e_{287} \end{bmatrix} \tag{4-6}$$

$$B = e_{66} + e_{93} + e_{162} + e_{177} + e_{243} + e_{288} \tag{4-7}$$

Using the decomposition (4-4), it then follows that S_t satisfies the following expressions.

$$S_t = A^t S_0 + \sum_{l=0}^{t-1} A^l H \Delta_{t-l}, \quad t \geq 1 \tag{4-8}$$

The minimal polynomial of A has degree m = 282 whose we denote by $\varphi(x) = \sum_{k=0}^{m} \varphi_k x^k$, see the appendix. Since $\sum_{k=0}^{m} \varphi_k A^k = 0$, it follows that

$$\begin{aligned} \sum_{k=0}^{m} \varphi_k S_{t+k} &= \sum_{k=0}^{m} \varphi_k (A^{t+k} S_0 + \sum_{l=0}^{t+k-1} A^l H \Delta_{t+k-l}) \\ &= \sum_{k=0}^{m} \varphi_k A^{t+k} S_0 + \sum_{k=0}^{m} \varphi_k \sum_{l=0}^{t+k-1} A^l H \Delta_{t+k-l} \\ &= \sum_{k=0}^{m} \varphi_k \sum_{l=0}^{t+k-1} A^l H \Delta_{t+k-l} \\ &= \sum_{\tau=0}^{m} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} A^r H \Delta_{t+\tau} \end{aligned} \tag{4-9}$$

for t≥1.
Multiplying both the most right and the most left sides of (4-9) by B and using (4-5), we have

[4] e_i denotes the i^{th} row of the 288×288 identity matrix.

$$\sum_{k=0}^{m} \varphi_k z_{t+k} = \sum_{k=0}^{m} \varphi_k BS_{t+k}$$
$$= \sum_{\tau=0}^{m} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} BA^r H\Delta_{t+\tau} \qquad (4\text{-}10)$$
$$= \sum_{\tau=0}^{m} C_\tau \Delta_{t+\tau}$$

where

$$C_\tau \overset{\Delta}{=} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} BA^r H \qquad (4\text{-}11)$$

are 1×3 vectors for $\tau = 0, 1, ..., m$.

Let C′, C″ and C‴ denote the three columns of a (m+1)×3 matrix whose i^{th} row is C_{i-1}. Similarly, Let Δ′, Δ″ and Δ‴ denote the three rows of a 3×(m+1) matrix whose j^{th} column is Δ_{t+j-1}, that is

$$[C' \quad C'' \quad C'''] \overset{\Delta}{=} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_m \end{bmatrix} \qquad (4\text{-}12)$$

and

$$\begin{bmatrix} \Delta_t' \\ \Delta_t'' \\ \Delta_t''' \end{bmatrix} \overset{\Delta}{=} [\Delta_t \quad \Delta_{t+1} \quad \cdots \quad \Delta_{t+m}] \qquad (4\text{-}13)$$

Then, (4-10) could be rewritten as

$$\Phi Z_t = \Delta_t' C' + \Delta_t'' C'' + \Delta_t''' C''' \qquad (4\text{-}14)$$

where

$$\Phi = [\varphi_0 \quad \varphi_1 \quad ... \quad \varphi_m] \qquad (4\text{-}15)$$

and

$$Z_t = [z_t \quad z_{t+1} \quad ... \quad z_{t+m}] \qquad (4\text{-}16)$$

The relation (4-14) is what we were looking for.

5. **Correlation Coefficient Analysis**

In general the sum of unbalanced Boolean functions can be balanced. However, Golic has proved that if the functions are picked independently at random, then with high probability their sum is unbalanced with the correlation coefficient very close to the product of the individual correlation coefficients [4]. Using this fact, it can be inferred that the relation (4-14) produces an unbalanced sequence $\{e_t\} = \{\Phi Z_t\}$. The standard chi-square frequency statistical test can then be applied to $\{e_t\}$ to distinguish this sequence from a purely random binary sequence. The distinguishing error probability is less than about 10^{-3}, if the segment length is $10/\varepsilon^2$ where $\varepsilon = 1 - 2\Pr\{e_t = 1\}$ is the correlation coefficient of $\{e_t\}$.

Every component of Δ'_t, Δ''_t and Δ'''_t vectors is product of two (approximately independent random) binary terms, and therefore has correlation coefficient equal to ½. While the non-adjacent components of these vectors are (approximately) independent, it is not true for the adjacent components; because every two adjacent components of these vectors have one term in common, see the equations (4-1) to (4-3).

However, all the runs[5] in C′, C″ and C‴ have length one and thus there is no concern about the independence of the sum of noise terms in (4-14), see appendix. The total number of runs in C′, C″ and C‴ is 126 which shows that the correlation coefficient of $\{e_t\}$ is $\varepsilon = 2^{-126}$.

Remark If there were some runs in C′, C″ or C‴ with length $n \geq 2$, we must have grouped the noise functions into suitable categories such that the required independence assumption was satisfied. In other words, we must have included the total effect of the noise terms corresponding to each run as one independent noise term. The some of n adjacent noise terms could be expressed by the Bent function $x_1x_2 + x_2x_3 + \cdots + x_nx_{n+1}$ which has correlation coefficient equal to $2^{-\lfloor (n+1)/2 \rfloor}$. Therefore, if we denote the total number of runs with length n ($n \geq 1$) in C′, C″ or C‴ by k_n, the correlation coefficient of $\{e_t\}$ is $\varepsilon = \prod_{n\geq 1} 2^{-k_n \lfloor (n+1)/2 \rfloor}$ in general.

Since every linear function of a given sequence can be defined as polynomial in the generating function domain, it follows that linear equations with correlation coefficients greater than 2^{-126} may be found by using the generating function concept. Let $\{a_t\}$ denote a given sequence. In generating function domain, the linear function $\sum_{k=0}^{r} u_k a_{t+k}$ is denoted by $U(D)a_t$ where $U(D) = \sum_{k=0}^{r} u_k D^k$.

[5] A consecutive subsequence of ones in a sequences (or vector) which are followed immediately after and before by a zero (if there are such bits) is called a run. For example the vector [1 0 1 1 0 0 111 0 1 0 0 11 0] has two runs of length one, two runs of length two and one run of length three.

Let corresponding to a given vector $U = [u_0 \quad u_1 \quad ... \quad u_r]$ define the polynomial $U(D) = \sum_{k=0}^{r} u_k D^k$. Using this notation, the relation (4-14) can be expressed in generating function domain as follows.

$$\Phi(D)z_t = C'(D)\delta'_t + C''(D)\delta''_t + C'''(D)\delta'''_t \qquad (4\text{-}17)$$

The polynomials $\Phi(D)$, $C'(D)$, $C''(D)$ and $C'''(D)$ are available in the appendix.
It is easy to show that multiplying both sides of (4-11) in any non-zero polynomial gives another unbalanced linear equation. In order to find equations with correlation coefficients greater than 2^{-126}, we must multiply both sides of (4-11) by some suitable polynomial P(D). We did thorough search over all polynomials P(D) with non-zero constant term and degree up to 24. The maximum correlation coefficient, among all those polynomials, is achieved by the following two independent choices for P(D) which is equal to 2^{-72}.

$$P_1(D) = 1 + D^6 \qquad (4\text{-}18)$$

$$P_2(D) = (1+D)(1 + D^6) \qquad (4\text{-}19)$$

For $P_1(D)$, all of the vector representation of the polynomials $P_1(D)C'(D)$, $P_1(D)C''(D)$ and $P_1(D)C'''(D)$ have exactly just 24 runs of length one, while in case of $P_2(D)$ all of them have exactly just 24 runs of length two. According to the above remark, both of them are corresponding to correlation coefficient equal to 2^{-72}.
Looking into the polynomials $C'(D)$, $C''(D)$ and $C'''(D)$, it is obvious that all of the polynomials $C'(D)$, $C''(D)$ and $C'''(D)$ are multiplication of some polynomial in D^3 and the simple polynomial D ($\Phi(D)$ is also a polynomial in D^3!!). One may think that linear functions with greater correlation coefficients could be found by considering P(D) as a polynomial in D^3. We also did thorough search over all polynomials $P(D) = K(D^3)$ which K(D) had non-zero constant term and degree up to 24. In this case, the maximum correlation coefficient among all those polynomials is again 2^{-72} achieved by $K(D) = 1 + D^2$ which is in accordance with $P_1(D) = 1 + D^6$.
The value 2^{-72} of the beast correlation coefficient which we found shows that the time complexity for distinguishing the output sequence of the TRIVIUM form a truly random generator is $O(2^{144})$. It seems impossible to find a linear function of consecutive output bits with correlation coefficient more than 2^{-40} to have a successful distinguishing attack.

6. Conclusion

In this paper, we extracted the linear sequential circuit approximation of the TRIVIUM stream cipher. We derive a linear function of consecutive output bits which is hold with correlation coefficient of about 2^{-72}. It seems very hard to find a linear function of consecutive output bits with correlation coefficient greater than 2^{-40} to have a successful distinguishing attack. These results show that TRIVIUM is strong against linear sequential circuit approximation attack, in spite of the linearity of its output function and all of the

components of its next-state function except three of them which also have very near distances from some linear functions.

References

1. eSTREAM, the ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/
2. J. Hong, "Some Trivial States of TRIVIUM", ECRYPT Discussion Forum, http://www.ecrypt.eu.org/stream/phorum/
3. C. D. Canniere and B. Preneel, "TRIVIUM Specifications", ECRYPT Stream Cipher Project Report 2005/030, 2005, available at http://www.ecrypt.eu.org/stream/
4. C. D. Canniere, Personal Communication.
5. J. Dj. Golic, "Intrinsic statistical weakness of keystream generators," Advances in Cryptology - ASIACRYPT '94, Lecture Notes in Computer Science, vol. 917, pp. 91-103, 1995.
6. J. Dj. Golic, "Linear models for keystream generators," IEEE Trans. Comput., vol. C-45, pp. 41-49, Jan. 1996

Appendix

$$\Phi(x) = 1 + x^{6} + x^{12} + x^{15} + x^{18} + x^{21} + x^{24} + x^{30} + x^{36} + x^{45} + x^{51} + x^{54} + x^{57} + x^{63} + x^{69} + x^{72} + x^{75} + x^{78} + x^{81} + x^{84} + x^{90} + x^{96} + x^{102} + x^{108} + x^{114} + x^{120} + x^{123} + x^{126} + x^{129} + x^{135} + x^{201} + x^{207} + x^{210} + x^{213} + x^{216} + x^{222} + x^{228} + x^{234} + x^{240} + x^{246} + x^{252} + x^{258} + x^{264} + x^{270} + x^{276} + x^{282}$$

$$C'(x) = x + x^{7} + x^{13} + x^{16} + x^{19} + x^{22} + x^{31} + x^{37} + x^{40} + x^{52} + x^{61} + x^{73} + x^{79} + x^{85} + x^{88} + x^{91} + x^{94} + x^{97} + x^{100} + x^{103} + x^{106} + x^{118} + x^{124} + x^{127} + x^{130} + x^{133} + x^{145} + x^{151} + x^{154} + x^{157} + x^{160} + x^{163} + x^{166} + x^{169172} + x^{175} + x^{178} + x^{181} + x^{184} + x^{187} + x^{190} + x^{193} + x^{199} + x^{205} + x^{211} + x^{217}$$

$$C''(x) = x + x^{7} + x^{13} + x^{16} + x^{19} + x^{22} + x^{40} + x^{43} + x^{61} + x^{64} + x^{67} + x^{85} + x^{88} + x^{91} + x^{97} + x^{103} + x^{118} + x^{124} + x^{130} + x^{133} + x^{151} + x^{154} + x^{157} + x^{160} + x^{163} + x^{166} + x^{169} + x^{172} + x^{175} + x^{178} + x^{181} + x^{184} + x^{187} + x^{190} + x^{193} + x^{196} + x^{199} + x^{202} + x^{208} + x^{214}$$

$$C'''(x) = x + x^{16} + x^{22} + x^{31} + x^{34} + x^{37} + x^{40} + x^{52} + x^{58} + x^{61} + x^{64} + x^{67} + x^{70} + x^{79} + x^{88} + x^{94} + x^{109} + x^{112} + x^{115} + x^{118} + x^{121} + x^{124} + x^{127} + x^{133} + x^{139} + x^{154} + x^{157} + x^{160} + x^{163} + x^{166} + x^{169} + x^{172} + x^{175} + x^{181} + x^{187} + x^{193} + x^{199} + x^{205} + x^{211} + x^{217}$$