Distinguishing Attack on Grain

Shahram Khazaei[⊥], Mehdi Hassanzadeh[┼], Mohammad Kiaei[⊥]

[⊥] Electrical Engineering Department
Sharif University of Technology
Tehran, Iran
{khazaaei, mohammadkiaei}@yahoo.com
[┼] Raymand Information and Communication Cryptographers
Tehran, Iran
hasanzadeh@raymandcrypto.ir
Oct. 2005

Abstract

Grain is one of the simplest ECRYPT Stream Cipher project Candidates which deals with key and IV of length 80 and 64 respectively. Using the linear sequential circuit approximation method, introduced by Golic in 1994, we derive a linear function of consecutive keystream bits which is held with correlation coefficient of about $2^{-63.7}$. Then using the concept of so-called generating function, we turn it into a linear function with correlation coefficient of 2^{-29} which shows that the output sequence of Grain can be distinguished from a purely random sequence using about $O(2^{61.4})$ bits of the output sequence with the same time complexity. A preprocessing phase for computing a trinomial multiple of a certain primitive polynomial with degree 80 is needed which can be performed using time and memory complexities of $O(2^{40})$.

Keywords. Stream Cipher, Distinguishing Attack, Linear Sequential Circuit Approximation, Grain, ECRYPT, Security Evaluation.

1. Introduction

Golic [2,3] has shown that for a binary keystream generator with M bits of memory whose initial state is uniformly chosen in a random way, there exists a linear function of at most M+1 consecutive output bits which is an unbalanced function of the initial state variables. 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

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

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 Grain stream cipher [4] - one of the simplest ECRYPT Stream Cipher project Candidates [1]. We first derive a linear function of consecutive output bits which is held with correlation coefficient of about $2^{-63.7}$. Then using the generating function concept, we turn it into a linear function with correlation coefficient of about 2^{-29}. A chi-square test could be applied to distinguish the output sequence of Grain from the output sequence of a truly binary random number generator. The required time and data complexity is $O(2^{61.4})$ for achieving the distinguishing error probability equal to 0.001.

The paper is organized as follows. In Sections 2 and 3 a brief description of the Grain stream cipher and linear sequential circuit approximation are respectively given. The linear sequential circuit approximation of Grain is derived in Section 4. The details of the attack come in Section 5 and the paper is concluded in Section 6.

2. A Brief Description of Grain

Grain [4] is a very simple hardware oriented synchronous stream cipher proposed as a candidate to the ECRYPT Stream Cipher Project [1]. Grain consists of an LFSR and a NFSR of length 80 and generates its key stream from an 80-bit secret key and a 64-bit initial value (IV). The proposed design uses an 11-input Boolean function g as the feedback function of the NFSR, and a 5-input Boolean function h to filter the contents of five fixed cells of LFSR and NFSR. The output of the feedback function is masked with the output bit of the LFSR to update the NFSR and the output of the filter function is masked with the output bit from the NFSR to produce the keystream z_t, see Figure 1.

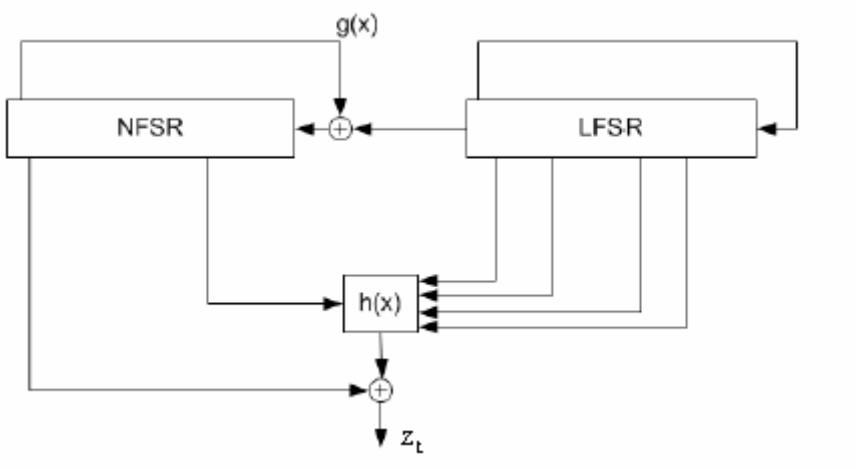

Figure 1. Schematic of the Grain stream cipher.

A complete description can be given by the following pseudo-code for producing N bits of the keystream. The initial state of LFSR and NFSR denoted by (s_0, s_1,…, s_{79}) and (b_0, b_1,…, b_{79}) are determined through a certain key-IV setup procedure [4].

for $t = 1$ **to** N **do**

$t_s \leftarrow s_0 + s_{13} + s_{23} + s_{38} + s_{51} + s_{62}$

$t_b \leftarrow s_0 + g(b_{63}, b_{60}, b_{52}, b_{45}, b_{37}, b_{33}, b_{28}, b_{21}, b_{15}, b_9, b_0)$

$z_t \leftarrow b_0 + h(b_{63}, s_{64}, s_{46}, s_{25}, s_3)$

$(s_0, s_1, \ldots, s_{79}) \leftarrow (s_1, s_2, \ldots, s_{79}, t_s)$

$(b_0, b_1, \ldots, b_{79}) \leftarrow (b_1, b_2, \ldots, b_{79}, t_b)$

end for

The g and h functions come in the following.

$$h(x_4, \ldots, x_0) = x_1 + x_4 + x_0x_3 + x_2x_3 + x_3x_4 + x_0x_1x_2 + x_0x_2x_3 + x_0x_2x_4 + x_1x_2x_4 + x_2x_3x_4. \quad (2\text{-}1)$$

$$g(x_{10}, \ldots, x_0) = x_{10} + x_9 + x_8 + x_7 + x_6 + x_5 + x_4 + x_3 + x_2 + x_1 + x_0 + x_{10}x_9 + x_6x_5 + x_2x_1 + x_9x_8x_7 + x_5x_4x_3 + x_{10}x_7x_4x_1 + x_9x_8x_6x_5 + x_{10}x_9x_3x_2 + x_{10}x_9x_8x_7x_6 + x_5x_4x_3x_2x_1 + x_8x_7x_6x_5x_4x_3. \quad (2\text{-}2)$$

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 may also the structure depend on a secret key. A binary autonomous finite-state machine is defined by

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

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

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 (the superscript ^{T} denotes the matrix transposition operation).

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 exists 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 [3], which is mentioned 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 has also 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, the basic equations (3-1) and (3-2) are put into the following form.

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

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

where, considering S_t as an $M\times1$ vector, A is an $M\times M$ matrix and B is a $1\times M$ vector, Δ is an $M\times1$ noise vector and γ 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[2] of A, the state transition matrix of the linear sequential circuit.

The next state function and the output function of the Grain are independent of the secret key. The balance condition of next state function is also well satisfied. Thus, its linear sequential model can be investigated.

4. **Linear Sequential Circuit Approximation of Grain**

In this Section, we derive the linear sequential circuit approximation of the Grain stream cipher from basis. For the Grain stream cipher we have $M = 160$. Let S_t be an 160 bit binary column vector which contains the state of LFSR and NFSR of Grain at time t, that is $(s_0, s_1,\ldots, s_{79}, b_0, b_1,\ldots, b_{79})^T$ in the pseudo-code introduced in Section 2. The function g is the only nonlinear part of the next-state function. The filter function h is also nonlinear. We utilize the linear approximation $L_{g,w}(x_{10},\cdots,x_0) = w_{10}x_{10} + \cdots + w_0x_0$ for the feedback function g and linear function $L_{h,v}(x_4,\cdots,x_0) = v_4x_4 + \cdots + v_0x_0$ for the filter function h. Using these decompositions of g and h functions, the linear approximations (3-3) and (3-4) for the Grain could be written as follows

$$S_t = AS_{t-1} + H\delta_t \qquad t\geq1 \tag{4-1}$$

---

[2] The minimal polynomial of a given square matrix A, is the least 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.

$$z_t = BS_t + \gamma_t \qquad t \geq 1 \tag{4-2}$$

where, $H = [h_i]$ is a 160 bit binary column vector with all entries equal to zero except h_{160}, δ_t and γ_t are respectively the scalar noise terms corresponding to the linear approximation $L_{g,w}$ of g and $L_{h,v}$ of f, and A and B are as follows.

$$A = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{80} \\ e_0 + e_{13} + e_{23} + e_{38} + e_{51} + e_{62} \\ e_{81} \\ e_{82} \\ \vdots \\ e_{160} \\ w_g \end{bmatrix} \tag{4-3}$$

and

$$B = e_{80} + v_4 e_{143} + v_3 e_{64} + v_2 e_{46} + v_1 e_{25} + v_0 e_3 \tag{4-4}$$

where e_i ($0 \leq i \leq 159$) denotes the $(i+1)^{\text{th}}$ row of the 160×160 identity matrix and

$$\begin{aligned} w_g = {} & e_0 + w_{10} e_{143} + w_9 e_{140} + w_8 e_{132} + w_7 e_{125} + w_6 e_{117} \\ & + w_5 e_{113} + w_4 e_{108} + w_3 e_{101} + w_2 e_{95} + w_1 e_{89} + w_0 e_{80} \end{aligned} \tag{4-5}$$

Using the decomposition (4-1), it then follows that S_t satisfies the following relation.

$$S_t = A^t S_0 + \sum_{l=0}^{t-1} A^l H \delta_{t-l}, \quad t \geq 1 \tag{4-6}$$

Denote the minimal polynomial of A by $\varphi(x) = \sum_{k=0}^{m} \varphi_k x^k$. Since $\sum_{k=0}^{m} \varphi_k A^k = 0$, it follows that

$$\begin{aligned} \sum_{k=0}^{m} \varphi_k S_{t+k} &= \sum_{k=0}^{m} \varphi_k (A^{t+k} S_0 + \sum_{l=0}^{t+k-1} A^l H \delta_{t+k-l}) \\ &= \sum_{k=0}^{m} \varphi_k A^{t+k} S_0 + \sum_{k=0}^{m} \varphi_k \sum_{l=0}^{t+k-1} A^l H \delta_{t+k-l} \\ &= \sum_{k=0}^{m} \varphi_k \sum_{l=0}^{t+k-1} A^l H \delta_{t+k-l} \\ &= \sum_{\tau=0}^{m} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} A^r H \delta_{t+\tau} \end{aligned} \tag{4-7}$$

for $t \geq 1$.
Multiplying both the most right and the most left sides of (4-7) by B and using (4-2), we have

$$\begin{aligned}\sum_{k=0}^{m} \varphi_k (z_{t+k} + \gamma_{t+k}) &= \sum_{k=0}^{m} \varphi_k B S_{t+k} \\ &= \sum_{\tau=0}^{m} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} B A^r H \delta_{t+\tau} \\ &= \sum_{\tau=0}^{m} c_\tau \delta_{t+\tau}\end{aligned} \tag{4-8}$$

where

$$c_\tau \overset{\Delta}{=} \sum_{r=0}^{m-\tau} \varphi_{r+\tau} B A^r H \tag{4-9}$$

are scalar binary values for $\tau = 0, 1, ..., m$.
The following relation which is the same as (4-8) is what we were looking for.

$$\sum_{k=0}^{m} \varphi_k z_{t+k} = \sum_{k=0}^{m} \varphi_k \gamma_{t+k} + \sum_{\tau=0}^{m} c_\tau \delta_{t+\tau} \tag{4-10}$$

Note that the coefficients φ_k ($0 \leq k \leq m$) depend on the coefficients w_i ($0 \leq i \leq 10$) and the coefficients c_τ ($0 \leq \tau \leq m$) depend on both the coefficients w_i ($0 \leq i \leq 10$) and v_j ($0 \leq j \leq 4$).

5. Details of the Attack

5.1. Generating Function Concept

Every linear function of a given sequence can be defined as a polynomial in the generating function domain. Let $\{a_t\}$ be an arbitrary binary sequence, and let $\{b_t\}$ be a linear function of $\{a_t\}$ defined by $b_t = \sum_{k=0}^{r} p_k a_{t+k}$. In generating function domain, the linear function $b_t = \sum_{k=0}^{r} p_k a_{t+k}$ is denoted by $b_t = p(D)a_t$ where $P(D) = \sum_{k=0}^{r} p_k D^k$. It can be easily shown that for an arbitrary polynomial $k(x)$ we have $k(D)b_t = k(D)\,p(D)a_t$.

5.2. 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 [3]. Using this fact, it can be inferred that the relation (4-10) produces an unbalanced sequence $u_t = \sum_{k=0}^{m} \varphi_k z_{t+k}$ if the errors of both linear approximation $L_{g,w}$ and $L_{h,v}$ of g and h have non-zero correlation coefficients.
In the generating function domain, introduced in Section 5.1, the relation (4-10) can be rewritten in the following way.

$$u_t = \varphi(D) z_t = \varphi(D)\gamma_t + c(D)\delta_t \tag{5-1}$$

where $\varphi(x) = \sum_{k=0}^{m} \varphi_k x^k$ and $c(x) = \sum_{\tau=0}^{m} c_\tau x^\tau$.
The weight of a given polynomial $k(x)$, denoted by $wh(k)$, is defined as the number of its non-zero coefficients. Let $\varepsilon_{g,w}$ and $\varepsilon_{h,v}$ denote the correlation coefficients of δ_t and γ_t - the noise terms corresponding to the linear approximation $L_{g,w}$ of g and $L_{h,v}$ of f. Under the independence assumption of the noise terms in (5-1), the correlation coefficient of u_t denoted by $\varepsilon_{w,v}$ is equal to $\varepsilon_{w,v} = \varepsilon_{g,w}^{wh(c)} \cdot \varepsilon_{h,v}^{wh(\varphi)}$.
We carried out exhaustive search over all of the $2^{11} \times 2^5$ possible choices for w and v to find the one with the greatest correlation coefficient. The greatest correlation coefficient is achieved by the following choice for w and v,

$$w = [w_{10} \cdots w_0] = [0 \quad \cdots \quad 0 \quad 1] \tag{5-2}$$

$$v = [v_4 \cdots v_0] = [0 \quad 1 \quad 0 \quad 1 \quad 0] \tag{5-3}$$

in accordance with the linear approximations $L_{g,w}(x_{10}, \ldots, x_0) = x_0$ and $L_{h,v}(x_4, \ldots, x_0) = x_3 + x_1$ for g and h respectively. The correlation coefficient of noise terms corresponding to these linear approximations are $\varepsilon_{g,w} = 5/256$ and $\varepsilon_{h,v} = 1/4$. The corresponding $\varphi(x)$ and $c(x)$ are as follows.

$$\begin{aligned} \varphi(x) = 1 &+ x^{13} + x^{23} + x^{38} + x^{51} + x^{62} \\ &+ x^{93} + x^{103} + x^{118} + x^{131} + x^{142} + x^{160} \end{aligned} \tag{5-4}$$

$$c(x) = x + x^{14} + x^{24} + x^{39} + x^{52} + x^{63} + x^{81} \tag{5-5}$$

Since $wh(\varphi) = 12$ and $wh(c) = 7$, the corresponding correlation coefficient of u_t is equal to $\varepsilon_{w,v} = (1/4)^{12}(5/256)^7 \approx 2^{-63.7}$. The standard chi-square frequency statistical test can then be applied to $\{u_t\}$ 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 about $10/\varepsilon_{w,v}^2 \approx 2^{130.8}$. The computational complexity of processing this amount of keystream is $O(2^{130.8})$ which is much higher than the exhaustive key search $O(2^{80})$. In the next Section we explain how to achieve a sequence with correlation coefficient greater than 2^{-40}.

5.3. Linear Equation with Greater Correlation Coefficient

Given a linear equation of consecutive output bits of the form (5-1), linear equations with greater correlation coefficients may be found using the generating function concept. To this end, we must multiply both sides of (5-1) by an appropriate polynomial $k(D)$ to obtain

$$\begin{aligned} u_t^* \overset{\Delta}{=} k(D)u_t &= k(D)\varphi(D)z_t \\ &= k(D)\varphi(D)\gamma_t + k(D)c(D)\delta_t \end{aligned}$$

such that the correlation coefficient of $\{u_t^*\}$ is greater than that of $\{u_t\}$. The less $wt(k\varphi)$ and $wt(kc)$ are, the greater the correlation coefficient of $\{u_t^*\}$ will be. In general, it is not easy to manage to keep both $wt(k\varphi)$ and $wt(kc)$ low. However, for the aforementioned values of w and v in (5-2) and (5-3), the corresponding polynomials $\varphi(x)$ and $c(x)$ in (5-4) and (5-5) have very special forms and can be factorized in the following way.

$$\varphi(x) = (1 + x^{80})(1 + x^{13} + x^{23} + x^{38} + x^{51} + x^{62} + x^{80}) \qquad (5\text{-}6)$$

$$c(x) = x(1 + x^{13} + x^{23} + x^{38} + x^{51} + x^{62} + x^{80}) \qquad (5\text{-}7)$$

Therefore, trying to find $k(x)$ is much easier in this case. Suppose that $p^*(x) = 1 + x^b + x^t,\ (1 \le b \le t),$ is a trinomial multiple of $p(x)$ where

$$p(x) = 1 + x^{13} + x^{23} + x^{38} + x^{51} + x^{62} + x^{80} \qquad (5\text{-}8)$$

Then choosing $k(x) = p^*(x) \big/ p(x)$ leads to

$$\begin{aligned} u_t^* \overset{\Delta}{=} k(D)u_t &= (D^{80} + 1)p^*(D)z_t \\ &= (D^{80} + 1)p^*(D)\gamma_t + Dp^*(D)\delta_t \end{aligned}$$

If $b = 80$ then $wh((x^{80} + 1)p^*(x)) = 4$, otherwise $wh((x^{80} + 1)p^*(x)) = 6$. In the worst case, that is $b \neq 80$, the correlation coefficient of $\{u_t^*\}$ is equal to $\varepsilon = (1/4)^6 (5/256)^3 \approx 2^{-29}$. Therefore, the required output length and computational time complexity for distinguishing the Grain output sequence from a purely random sequence with error probability less than 10^{-3} is about $10/\varepsilon^2 \approx 2^{61.4}$.

Remark. The problem of finding a low weight multiple of a randomly chosen irreducible polynomial of degree n has been well considered in [5] and [6]. In sort, a trinomial multiple of degree about $2^{n/2}$ could be found using $O(2^{n/2})$ time and space. Therefore, we expect that the required trinomial multiple $p^*(x)$ of the primitive polynomial $p(x)$ be found using time and memory complexities of $O(2^{40})$.

6. Conclusion

In this paper using the *forgotten* linear sequential circuit approximation method, we mounted a distinguishing attack on Grain which needs about $O(2^{61.4})$ bits of the keystream. A preprocessing phase for computing a trinomial multiple of a certain primitive polynomial with degree 80 is also needed which can be performed using $O(2^{40})$ time and space.

References

1. eSTREAM, the ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/
2. 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.
3. J. Dj. Golic, "Linear models for keystream generators," IEEE Trans. Comput., vol. C-45, pp. 41-49, Jan. 1996.
4. M. Hell, T. Johansson and W. Meier, "Grain - A Stream Cipher for Constrained Environments," ECRYPT Stream Cipher Project Report 2005/010, 2005, available at http://www.ecrypt.eu.org/stream/
5. W.T. Penzhorn, G.J. Kühn, "Computation of Low-Weight Parity Checks for Correlation Attacks on Stream Ciphers," Cryptography and Coding, LNCS 1024, Springer, pp.74{83, 1995.
6. D. Wagner, "A generalized birthday problem," Advances in Cryptology CRYPTO 2002, LNCS 2442, pp.288-304, Springer-Verlag, 2002. the extended abstract is available at: http://www.cs.berkeley.edu/~daw/papers/genbday.html