Countering the Correlation Attack on Pomaranch

Cees J.A. Jansen[1] and Alexander Kholosha[2]

[1] Banksys NV
Haachtsesteenweg 1442
1130 Brussels, Belgium
[2] The Selmer Center
Department of Informatics, University of Bergen
P.O. Box 7800, N-5020 Bergen, Norway
cja@iae.nl; Alexander.Kholosha@ii.uib.no

Abstract. A recent key-recovery attack on Pomaranch stream cipher was built due to the spotted biases in the distribution of certain linear relations in the output sequence of a Jump Register Section. All the relations and corresponding biases were found by computer experiments. The suggested attack has the complexity $O(2^{95.4})$ and requires $2^{71.8}$ bits of the key-stream. In this paper we give theoretical reasons explaining the bias and provide the means for its evaluation. We also propose a minor change to the Pomaranch jump register configuration that allows to reduce the maximal bias to a level that effectively counters the attack increasing its complexity to $O(2^{133.4})$ with $2^{116.9}$ bits of the key-stream required.

1 Introduction

The Pomaranch stream cipher algorithm [1] uses a cascade construction of so called jump registers [2] being essentially linear finite state machines with a special transition matrix. The cascade consists of nine identical registers each built on 14 memory cells. The transition matrix of the jump register has been chosen with side channel resistance in mind. Moreover, the characteristic polynomial of the transition matrix was made to be primitive and satisfying additional constraints that arise from the need to use the register in a cascade jump control setup. In particular this means that it must be a member of a primitive S_6 set (see [2]), i.e., its jump index J as well as $J-1$ must be coprime with its period. The total number of polynomials of degree 14 belonging to primitive S_6 sets is 228 or when counting a polynomial and its dual as one (since both of them have similar properties) then this number goes down to 114 pairs. Then we looked for the polynomials that correspond to jump registers having a minimal number of feedback taps (namely, one). A polynomial chosen this way was implemented in the submitted design.

It was noted in [3] that some linear relations in any 15 consecutive bits of a Jump Register Section output have a biased distribution. This result was obtained by simulating the jump register operation during 14 cycles using all possible 2^{14} different Jump Control (JC) sequences of length 14 and then finding, listing and counting arising linear relations on corresponding 15 output bits. It thus turned out that there were two linear relations, which resulted from 840 different 14-bit sections of the JC sequence, and a total of 334 linear relations occurred. In [3] the author described a correlation key-recovery attack having the complexity $O(2^{95.4})$ that was based on the nonuniform distribution of the linear relation having the highest bias of $840/2^{14}$.

2 Linear Equivalences in Jump Registers

Depending on the value of the jump control bit, the transition matrix applied to a jump register state is either A or $A+I$, where I denotes the identity matrix. A has a primitive characteristic polynomial $C(x)$ (obviously, the characteristic polynomial of $A+I$ is the dual $C^{\perp}(x)$ which is primitive as well). Let R_i denote the state of the register at time i. Then $R_{i+1} = (A + JC_{i+1}I)R_i$, where JC_i denotes the jump control bit at time i. The matrix A is similar to the companion matrix of $C(x)$ and, hence, can be seen as a primitive element of $\mathrm{GF}(2^L)$, were L is the degree of $C(x)$. Clocking the register that is implemented by multiplying the state by the transition matrix, then is equivalent to multiplication by x or $x+1$.

Let $Z = \{z_i\}_{i=0}^{\infty}$ denote the output sequence of a jump register. Starting from some register state R_i, the first output bit z_i is not affected by the jump control bits from $(JC_{i+1}, \ldots, JC_{i+L})$, the second output bit z_{i+1} is defined by JC_{i+1}, the third z_{i+2} is defined by (JC_{i+1}, JC_{i+2}) and so on. Assume that a linear relation holds on $L+1$ consecutive bits of Z at the shift position i independent of the initial state of the register. This means that for some set of binary coefficients $(\ell_0, \ell_1, \ldots, \ell_L)$ and any initial state we have $\ell_0 z_i + \ell_1 z_{i+1} + \ldots + \ell_L z_{i+L} = 0$. This is equivalent to

$$\ell_0 + \sum_{j=1}^{L} \ell_j \prod_{k=1}^{j} (x + JC_{i+k}) = \sum_{j=0}^{L} \ell_j x^{j-k_j} (x+1)^{k_j} = C(x) \ , \tag{1}$$

where $0 \le k_j \le j$ are defined by the control bits $JC_{i+1}, \ldots, JC_{i+L}$, namely, $k_0 = 0$ and k_j is equal to the binary weight of vector $(JC_{i+1}, \ldots, JC_{i+j})$. Thus, if assuming the jump control sequence is purely random, then the values of k_j are binomially distributed. Since the degree of $C(x)$ is L and $C(0) = 1$ then the coefficients at the highest-order and the constant term of the polynomial standing on the left hand side of (1) should be nonzero, i.e., $\ell_0 = \ell_L = 1$ for any linear relation in the jump register output. Given an arbitrary jump control sequence (that provides the values of k_j) the solution of (1) for the unknowns ℓ_j can be found applying a simplified version of Gaussian elimination with the complexity linear in L. Such a solution always exists since every bit of the output is a linear combination of L bits from the initial state R_0 and thus any $L+1$ bits of the output sequence are linearly dependent. The same can be also easily seen from the matrix of the system that is nonsingular triangular.

Take a set of coefficients $\ell_0, \dots, \ell_N$ that satisfy (1) for some fixed jump control sequence and assume that the term at the coefficient $\ell_j = 1$ has the form $x^a(x+1)^b$ and for the nearest $t > j$ with $\ell_t = 1$ the term is $x^c(x+1)^d$ (obviously, $a+b = j$, $c+d = t$, $a \leq c$ and $b \leq d$). Then the number of possible $(t-j)$-long sections of the jump control sequence leading from $x^a(x+1)^b$ to $x^c(x+1)^d$ is equal to $\binom{c+d-a-b}{d-b}$ (these are exactly the sequences with the binary weight of $(JC_{i+j+1}, \dots, JC_{i+t})$ equal to $d-b$ since $t-j = c+d-a-b$). In a similar manner, starting from the constant term $x^0(x+1)^0$ at $\ell_0 = 1$ and proceeding till the highest-order term at $\ell_L = 1$ we can find the total number of jump control sequences that correspond to the given polynomial $\sum_{j=0}^{L} \ell_j x^{j-k_j}(x+1)^{k_j}$ and this number is obtained as a product of the relevant binomial coefficients for all $\ell_j \neq 0$ and $j > 0$.

As can be seen from (1), the set of possible linear relations that correspond to different control sequences, and the number of their occurrences only depend on the characteristic polynomial of the jump register. As the linear relation occurring most often plays an essential role in the aforementioned attack, we will call this number the *Linear Equivalence Bias* (LEB) of the polynomial. All occurrence numbers together form a *Linear Equivalence Spectrum* (LES) of the polynomial. It can easily be seen by interchanging the roles of x and $x+1$ that $C(x)$ and $C^{\perp}(x)$ have the same LES.

The following *Doubling Rule* holds

$$x^a(x+1)^b = \begin{cases} x^{a-1}(x+1)^b + x^{a-1}(x+1)^{b+1}, \\ x^a(x+1)^{b-1} + x^{a+1}(x+1)^{b-1} \end{cases} . \qquad (2)$$

This doubling rule can be applied to different terms in (1) that correspond to any given linear relation. Other relations can be found this way and partial contributions to their occurrences can be calculated. The most obvious example is doubling the highest-order term corresponding to $\ell_L = 1$, which gives rise to $\ell_{L-1} = 1$ and $\ell_L = 1$. Due to the binomial identity $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$ the partial contribution number computed for latter linear relation will be the same and this, in particular, implies that all values in the LES are even. Applying the doubling rule to other terms can result in a new relation having higher or lower partial contribution number. This feature will be illustrated in the following sections.

Note that using the presented technique we can evaluate a partial contribution to the total number of occurrences for some linear relations of length $L+1$ in the output sequence of a jump control register. In some cases this value is equal to the LEB of a polynomial meaning that we have found a relation that belongs to the ones occurring most often. We can not currently provide the algorithm for evaluating the LEB with the complexity lower than $O(L \cdot 2^L)$ (the number of JC sequences of length L multiplied by the complexity of a simple version of Gaussian elimination of length L). Finding a less complex algorithm remains an interesting open problem.

3 The LEB of Pomaranch

The characteristic polynomial chosen for Pomaranch is given by the following equation

$$C(x) = 1 + x^{\frac{L}{2}+k-n}(x+1)^{\frac{L}{2}-k} + x^{\frac{L}{2}}(x+1)^{\frac{L}{2}} \tag{3}$$

for $L = 14$, $n = 6$ and $k = 2$. So the linear relation $z_i + z_{i+L-n} + z_{i+L} = 0$ is immediately evident. Applying the doubling rule to the senior term in (3) we can get another relation $z_i + z_{i+L-n} + z_{i+L-1} + z_{i+L} = 0$. Both linear relations have the same occurrence number, given by

$$\binom{L-n}{\frac{L}{2}-k} \cdot \binom{n}{k} \quad . \tag{4}$$

For the values chosen for Pomaranch this results in $\binom{8}{5} \cdot \binom{6}{2} = 840$. The value of 840 also turns out to be the LEB of the characteristic polynomial used.

Applying the doubling rule, other LES values can be calculated. For example, applying the doubling rule to the 8th order term in $z_i + z_{i+8} + z_{i+14} = 0$ we obtain a new linear relation $z_i + z_{i+7} + z_{i+8} + z_{i+14} = 0$ having the occurrence number $\binom{7}{5} \cdot \binom{6}{1} + \binom{7}{4} \cdot \binom{6}{3} = 826$, the second largest LES value of this polynomial.

For $L = 14$ and using one feedback tap only there are only 5 choices for (n, k) and the dual $(n, n-k)$ resulting in characteristic polynomials satisfying all the conditions, viz. $(6, 2)$, $(7, 2)$, $(7, 3)$, $(8, 3)$ and $(11, 5)$. The corresponding LEBs are 840, 567, 1225, 840 and 1386 respectively. The conclusion is that with one feedback tap the resulting characteristic polynomials all have too high LEB to counter the attack suggested in [3].

4 A Modified Jump Register for Pomaranch

In order to find a characteristic polynomial with a sufficiently low LEB, the Pomaranch jump register is changed to have two feedback taps. There is one tap, the rightmost, at position n_1 with k_1 feedback cells among cells 1 to n_1. The other tap is at position $n_2 > n_1$, with k_2 feedback cells among cells $n_1 + 1$ to n_2. The modified characteristic polynomial now becomes

$$C(x) = 1+x^{\frac{L}{2}+k_1+k_2-n_2}(x+1)^{\frac{L}{2}-k_1-k_2}+x^{\frac{L}{2}+k_1-n_1}(x+1)^{\frac{L}{2}-k_1}+x^{\frac{L}{2}}(x+1)^{\frac{L}{2}} \tag{5}$$

for $L = 14$. The LES of this polynomial contains the obvious relation $z_i + z_{i+L-n_2} + z_{i+L-n_1} + z_{i+L} = 0$.

Searching through all relevant (n_1, n_2, k_1, k_2) quadruplets results in a set of 16 primitive S_6-set polynomials, amongst which are the five polynomials already obtained for one tap. The polynomial with the least LEB in this set is $x^{14} + x^{13} + x^{12} + x^{11} + x^9 + x^7 + x^5 + x^4 + x^2 + x + 1$ is obtained for $n_1 = 4$, $n_2 = 8$, $k_1 = k_2 = 1$ and has a LEB equal to 124 and a LES consisting of 1088 relations. The linear relation $z_i + z_{i+6} + z_{i+10} + z_{i+14} = 0$ occurs $\binom{6}{1} \cdot \binom{4}{3} \cdot \binom{4}{3} = 96$ times. Performing a doubling operation on the 6th order term yields a relation which occurs 124 times that is equal to the LEB value.

Plugging in the bias of $124/2^{14}$ of the jump register in the equations of [3] now results in the attack complexity of $O(2^{133.4})$ with $2^{116.9}$ bits of the key-stream required. This complexity exceeds the one of the exhaustive search over the key space containing 2^{128}.

5 Conclusion

We introduced a minor change in the configuration of the Jump Register Section in Pomaranch. Namely, feedback now is computed by taking the bits from the tap positions 4, 8 and 14. The positions of the F- and S-cells in the register are FFSFFFFSSFSSFSS. This new configuration brings the complexity of the key-recovery attack in [3] to $O(2^{133.4})$ with $2^{116.9}$ bits of the key-stream required.

References

1. Jansen, C.J.A., Helleseth, T., Kholosha, A.: Cascade jump controlled sequence generator (CJCSG). In: Symmetric Key Encryption Workshop, Workshop Record, ECRYPT Network of Excellence in Cryptology (2005) http://www.ecrypt.eu.org/stream/ciphers/pomaranch/pomaranch.pdf.
2. Jansen, C.J.A.: Stream cipher design based on jumping finite state machines. Cryptology ePrint Archive, Report 2005/267 (2005) http://eprint.iacr.org/2005/267/.
3. Khazaei, S.: Cryptanalysis of pomaranch (CJCSG). eSTREAM, ECRYPT Stream Cipher Project, Report 2005/065 (2005) http://www.ecrypt.eu.org/stream/papersdir/065.pdf.