Cryptanalysis of Pomaranch (CJCSG)

Shahram Khazaei

Electrical engineering Department
Sharif University of Technology, Iran
khazaaei@yahoo.com

1 Oct. 2005

Abstract. Pomaranch is a synchronous stream cipher submitted to eSTREAM, the ECRYPT Stream Cipher Project. It uses 128-bit keys and IVs with different lengths. The cipher is constructed as a cascade clock control sequence generator, which is based on the notion of jump registers. Each jump register can be considered as a non-autonomous finite state machine which the input sequence is called jump control sequence. In this paper we show that a jump register with a balanced identically distributed binary jump control sequence can be modeled as a non-autonomous linear finite state machine with an additive unbalanced input sequence. Using this result we mount a correlation based key-recovery attack on Pomaranch with computational complexity around $2^{95.4}$ using about $2^{71.8}$ bits of the output sequence.

1. Introduction

Pomaranch[1] [5] is one of the ECRYPT Stream Cipher Project [3] candidates. The cipher is implemented as a binary one clock pulse cascade clock control sequence generator, and uses 128 bit keys and IVs of length between 64 and 112bits [5]. The construction is based on the notion of Jump Registers (JR) first introduced in [4] as alternative to traditional clock-controlled registers to prevent timing and power attacks. A JR is an LFSR which is able to move to a state that is more than one step ahead without having to step through all the intermediate states. The so-called Jump Control (JC) sequence controls the clocking way of the LFSR. In [2] using a weakness in the initialization procedure of Pomaranch, a chosen IV attack has been mount on it which recovers the 128 bit secret key using about 2^{65} computations. The improved version of the attack is also applicable with complexity 2^{52} [2]. In this paper we show that a JR with a balanced identically distributed binary JC sequence can be modeled as a non-autonomous linear Finite State Machine (FSM) with an additive unbalanced input sequence. This shows that

[1] The cipher is also referred in the specification document [JHK05] as Cascade Jump Controlled Sequence Generator (CJCSG).

a chi-square test could be applied to distinguish the output sequence of a single JR with balanced identically distributed binary JC sequence. Using this result we mount a key-recovery attack on Pomaranch with computational complexity around $2^{95.4}$ using about $2^{71.8}$ bits of the output sequence. *This is the first attack on the keystream generator of Pomaranch.*

2. Description of the Keystream Generator of the Pomaranch

The keystream generator of Pomaranch is depicted in Figure 1. The cipher consists of nine cascaded JR denoted by R_1 to R_9. Each JR is built on 14 memory cells which behave either as simple delay shift cells or feedback cells, depending on the value of JC sequence. At any moment, half of the cells in the registers are shift cells, while the other half are feedback cells. The initial configuration of cells is determined by the transition matrix A, and is used if the JC value is zero. If JC is one, all cells are switched to the opposite mode. This is equivalent to switching the transition matrix to $(A + I)$ [5].

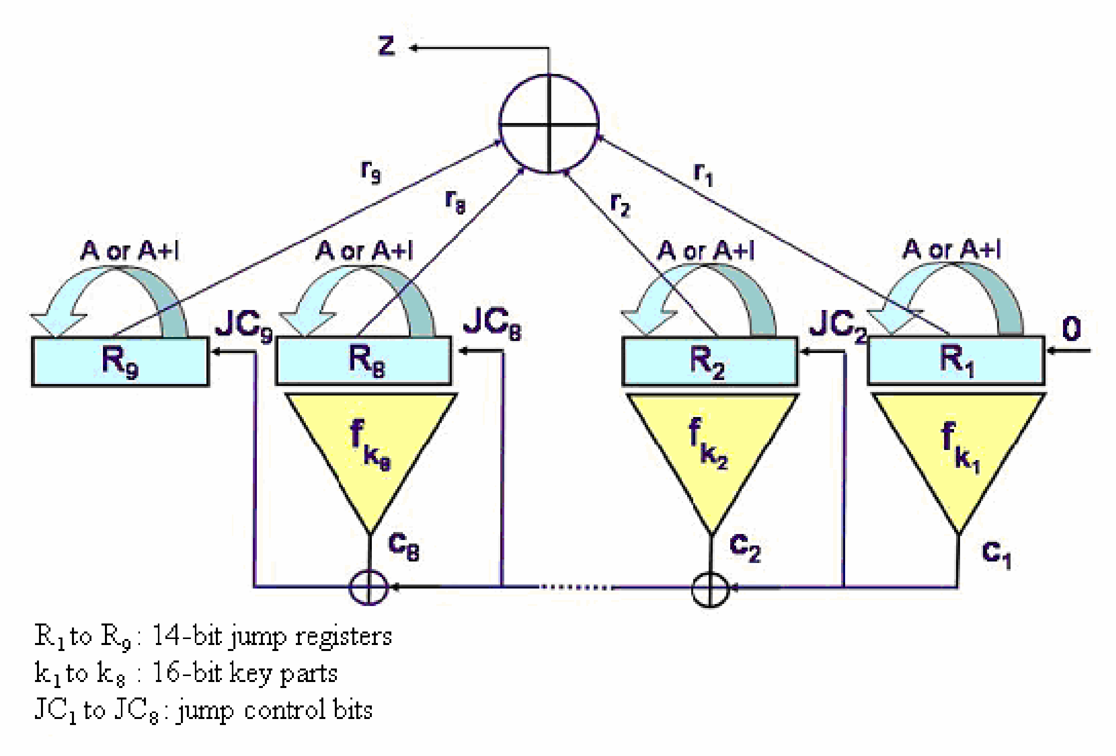

Figure 1. Schematic of the Pomaranch (adopted from [2])

The 128-bit key K is divided into eight 16-bit sub keys k_1 to k_8. At time t, the current states of the registers R_1^t to R_8^t are non-linearly filtered, using a function that involves the corresponding sub key k_i. These functions provide as output eight bits c_1^t to c_8^t, which are used to produce the jump control bits JC_1^t to JC_8^t controlling the registers R_2 to R_9 at time t, as following:

$$JC_i^t = c_1^t \oplus \cdots \oplus c_{i-1}^t \qquad (1)$$

for $2 \le i \le 9$.

The jump control bit JC_1 of register R_1 is permanently set to zero. The key stream bit z^t produced at time t is the XOR of nine bits r_1^t to r_9^t selected at second position of the registers R_1 to R_9.

3. Description of the Attack

In this section we first derive a non autonomous FSM model with unbalanced additive input for Jump Registers. Then we use this model to mount a key recovery attack on Pomaranch with time complexity much less than exhaustive key search.

3.1. The Non-Autonomous FSM Model With Unbalanced Additive Input for JR's

Consider a JR built on M bits of memory with transition matrix A. Let denote the JC and output sequence by $\{JC^t\}_{t=1}^{\infty}$ and $\{r^t\}_{t=1}^{\infty}$ respectively. If we denote the state sequence of the register by the sequence $\{R^t\}_{t=0}^{\infty}$ of M-bit column vectors, the act of JR can be described by the following non-autonomous FSM

$$R^t = (A + JC^t I)R^{t-1} \qquad (2)$$

$$r^t = BR^{t-1} \qquad (3)$$

for $t \ge 1$, where B is an M-bit row vector determined by the position of the register where the output is produced.

Since the state of the JR is linearly updated, each bit of the JR output sequence is a linear combination of the initial M-bit state of the JR. Therefore each $M + 1$ bits of the output sequence are linearly dependent where the linear dependence is determined by some bits of the JC sequence. Let concentrate on $M + 1$ consecutive bits r^t, r^{t+1}, … and r^{t+M}. In this case, for each 2^M possible values of JC^t, JC^{t+1}, … and JC^{t+M-1}, there is a non-zero binary vector[2] $w = [w_0 \ w_1 \ \cdots \ w_M]$ that $\sum_{k=0}^{M} w_k r^{t+k} = 0$.

Ideally, the uniform distribution of all 2^M vectors w among all 2^{M+1} possible values for them is desired[3]. Suppose that a vector w^* appears F times among all 2^M vectors. Assuming that $\{JC^t\}_{t=1}^{\infty}$ is a balanced identically distributed binary sequence, it turns out that the sequence $e^t = \sum_{k=0}^{M} w_k^* r^{t+k}$ is an unbalanced binary sequence with correlation coefficient[4] (or bias) $\varepsilon = F/2^M$. This could be proved as follows. Let H denote the event that the corresponding w of a purely random $\{JC^{t+k}\}_{k=0}^{M-1}$ is equal to w^*. The complement event of H is denoted by $\overline{H}$. Clearly $\Pr\{H) = F/2^M$ and $\Pr\{e_t = 1 \mid H) = 0$. The

[2] We conjecture that this vector is unique and $w_0 = w_M = 1$.

[3] If the conjecture $w_0 = w_M = 1$is true 2^{M+1} should be changed to 2^{M-1} (see footnote 2).

[4] The correlation coefficient of the random variable x is defined as ε = 1 – 2Pr{x = 1}.

probability $\Pr\{e_t = 1 \mid \overline{H})$ can be considered equal to one half because in this case $w \neq w^*$ and in half percent of the times e^t is equal to one. Therefore,

$$\begin{aligned} \Pr\{e^t = 1\} &= \Pr\{e^t = 1 \mid H\}\Pr\{H\} + \Pr\{e^t = 1 \mid \overline{H}\}\Pr\{\overline{H}\} \\ &= 0 \times F/2^M + 1/2 \times (1 - F/2^M) \\ &= 1/2(1 - F/2^M) \end{aligned} \tag{4}$$

which is what we were going to prove.
Having a large bias makes a JR unsuitable. The distribution of vectors w in general must be more investigated.
Now let go on with Pomaranch. The transition matrix A and vector B of JR's of Pomaranch are given in the following.

$$A = \begin{bmatrix} 0 & & & & & & & & & & & & & 1 \\ 1 & 1 & & & & & & & & & & & & 0 \\ & 1 & 0 & & & & & & & & & & & 0 \\ & & 1 & 1 & & & & & 0 & & & & & 0 \\ & & & 1 & 0 & & & & & & & & & 0 \\ & & & & 1 & 1 & & & & & & & & 0 \\ & & & & & 1 & 1 & & & & & & & 0 \\ & & & & & & 1 & 1 & & & & & & 0 \\ & & & & & & & 1 & 0 & & & & & 1 \\ & & & 0 & & & & & 1 & 0 & & & & 0 \\ & & & & & & & & & 1 & 0 & & & 0 \\ & & & & & & & & & & 1 & 1 & & 0 \\ & & & & & & & & & & & 1 & 0 & 0 \\ & & & & & & & & & & & & 1 & 1 \end{bmatrix}_{14 \times 14} \tag{5}$$

$$B = [0\ 1\ 0\ \cdots\ 0]_{1\times 14} \tag{6}$$

In this case, we have computed and tabulated all the vectors w for all 2^{14} possible values of JC^t, JC^{t+1}, … and JC^{t+13}. We realized that only 334 different vectors of w have non-zero frequency of appearance which differs from 2 to 840. The maximum frequency of appearance, that is 840, belongs to these two vectors.

$$w^* = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1] \tag{7}$$

$$w^{**} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 1] \tag{8}$$

Let just consider the vector w^*. Assuming that the JC sequences of JR R_2 to R_9 are i.i.d., it follows that the sequences $e_i^t = r_i^t + r_i^{t+8} + r_i^{t+14}$, $2 \le i \le 9$, are unbalanced with correlation coefficient $\varepsilon = 840/2^{14}$. This was verified by computer simulation.

3.2. Recovery of the Initial State of the Registers and the Secret Key

Let first introduce the following sequence for $1 \le i \le 8$.

$$u_i^t = e_{10-i}^t \oplus \cdots \oplus e_9^t \quad (9)$$

Using this notation the output sequence of the Pomaranch satisfies the following relation

$$z^t \oplus z^{t+8} \oplus z^{t+14} = r_1^t \oplus r_1^{t+8} \oplus r_1^{t+14} \oplus u_8^t \quad (10)$$

Under the independence assumption of e_i^t sequences, $2 \le i \le 9$, the correlation coefficient of the sequences u_i^t is equal to $\varepsilon^i = (840/2^{14})^i$ for $1 \le i \le 8$.

The equation (10) can be used in a correlation attack for finding the correct initial state of R_1. The aim of correlation attack, first introduced by Siegenthaler [6], is to find the initial state of a binary LFSR of length L, given the noisy output sequence of a BMSC[5] when the output sequence of the LFSR is applied to the input of this channel. As the first *N* bits of the output sequence of an LFSR is a codeword of the corresponding truncated cyclic linear code of given LFSR, the problem is essentially a decoding problem. Siegenthaler solves this problem using ML[6] decoding and computes the minimum required output length of the given noisy sequence, denoted by N_0, for successfully finding the initial state of the LFSR by considering a Hypotheses Testing problem. However, it is easier to express, N_0, using the capacity of the corresponding channel [1]. The ML decoding is performed by searching over all 2^L possible initial states for the LFSR and choosing one that its corresponding output sequence has the least (most) hamming distance from the given noisy sequence if the channel error probability is less (more) than one half. However, the correlation attack is not limited to LFSR's and can be applied to any generator with M different equally probable initial states provided that the output sequences of different initial states have good statistical properties. The channel capacity argument shows that the minimum required output length of the given noisy sequence of the generator is determined by

$$N_0 = \log_2(M)/C(p) \quad (11)$$

where *p* is the error probability and *C*(*p*) is the Channel Capacity[7] of the corresponding BMSC. Since the ML decoding is performed by exhaustive search over all possible initial states, he required computational complexity is $O(MN_0)$.

The aforementioned discussion on correlation attack shows that the initial state of R_1 of Pomaranch can successfully be found using $N_0 = 14/C(0.5(1-\varepsilon^8)) \approx 2^{72.8}$ bits of the output sequence with computational complexity $2^{14}N_0 \approx 2^{86.8}$. Note that $C(0.5(1-\varepsilon)) \approx \varepsilon^2/(2\ln 2)$ when $|\varepsilon| << 1$.

After finding the initial state of the R_1, we can eliminate theportion of r_1^t from the output sequence of Pomaranch. Let define the sequence z_1^t as the XOR of z^t and r_1^t which is now available. Then similar to (10) we have

$$z_1^t \oplus z_1^{t+8} \oplus z_1^{t+14} = r_2^t \oplus r_2^{t+8} \oplus r_2^{t+14} \oplus u_7^t \quad (12)$$

---

[5] Binary Memoryless Symmetric Channel

[6] Maximum Likelihood

[7] $C(p) = 1 - p\log_2(p) - (1-p)\log_2(1-p)$

Since r_2^t depends on both the 14-bit initial state of R_2 and 16-bit sub-key k_1, it can be considered as output of a generator with 2^{30} possible output sequences. Similarly, (12) can be used in a correlation attack to find the correct value of initial state of R_2 and sub-key k_1. Since the correlation coefficient of u_7^t is ε^7, the required output length and computational complexity are $N_0 = 30 / C(0.5(1-\varepsilon^7)) \approx 2^{65.4}$ and $2^{30} N_0 \approx 2^{95.4}$ respectively. The initial state of the other registers and other sub-keys can be found similarly with much lower computational complexity.

In this attack we have only used the vector w^* introduced in (7). Using both w^* and w^{**}, relations (7) and (8), halves the required output length. To summarize the results, we showed that the secret key of Pomaranch can be found using $2^{71.8}$ bits of the output sequence with computational complexity $2^{95.4}$.

4. Conclusion and Open Problem

In this paper we showed that an M-bit JR with a balanced identically distributed binary JC sequence can be modeled as a non-autonomous linear Finite State Machine (FSM) with an additive unbalanced input sequence. In order to find the best model, that is one whose unbalanced input sequence has the largest correlation coefficient, we made exhaustive search over all 2^M possible M consecutive bits of the JC sequence which has computational complexity $O(M^3 2^M)$. Since the JR's of Pomaranch are very short, M = 14, we could easily derive the model. Now, the open problem is the existence of any reduced complexity procedure. If there is any, it would be a new distinguisher for traditional clock-controlled LFSR's.

Using this model we applied a correlation based key-recovery attack on Pomaranch using $2^{71.8}$ bits of the output sequence with computational complexity $2^{95.4}$.

References

1. V. Chepyzhov and B. Smeets, "On a fast correlation attack on certain stream ciphers", Advances in Cryptology, EUROCRYPT 91, Lecture Notes in Computer Science, vol. 547, Springer-Verlag, 1991, pp. 176-185
2. C. Cid, H. Gilbert and T. Johansson, "Cryptanalysis of Pomaranch", ECRYPT Stream Cipher Project Report 2005/060, 2005, available at http://www.ecrypt.eu.org/stream/
3. eSTREAM, the ECRYPT Stream Cipher Project. http://www.ecrypt.eu.org/stream/.
4. C. J. Jansen, " Streamcipher Design: Make your LFSRs jump!", In SASC, Workshop Record, ECRYPT Network of Excellence in Cryptology, pages 94_108, 2004.
5. C. J. Jansen, T. Helleseth, and A. Kholosha. "Cascade Jump Controlled Sequence Generator (CJCSG)". ECRYPT Stream Cipher Project Report 2005/022, 2005, available at http://www.ecrypt.eu.org/stream/
6. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only", IEEE Trans. on Computers,vol . 34, 1985, pp. 81-85.