International Journal of Cryptology Research 2(1): 39-49 (2010)

Security Analysis of the Step (*D,K*) Generator Respect to its Parameters

[1]Mehdi M. Hassanzadeh and [2]Tor Helleseth
Department of Informatics, University of Bergen, Norway
Email: [1]Mehdi.hassanzadeh@ii.uib.no and [2]Tor.helleseth@ii.uib.no

ABSTRACT

Ciphers based on the irregularly clocked LFSR are one of the main and widely used types of stream ciphers. The simplest scheme use only two LFSRs; the first one is clocked regularly and its output controls the clocking of the second one which produce irregular output sequence. In general, the second register is clocked D or K times which is called Step(D,K) Generator. Most of the well known clock-controlled structure are a special case of the Step(D,K) Generator, e.g. Stop/Go generator is a Step(0,1) Generator. In this paper, we discuss the security of the Step(D,K) Generator respect to its parameters D and K. We will calculate the probability, P(n), of appearing n^{th} bit of the regular sequence into the output sequence. We will show that if P(n) is zero for some values of n, we can reduce the time complexity of the general attacks. In case of correlation attack based on the Levenshtein Distance, we improve the time complexity of the attack by $O(E(2M-N-E)2^L)$. Finally, some recommendations will be presented to answer "How we can choose good parameters?"

INTRODUCTION

In stream cipher design, the goal is to efficiently produce pseudorandom sequences which should be indistinguishable from truly random sequences. An important family of stream ciphers is Clock-Controlled stream cipher which has several different types. The purpose in this structure is to destroy the linearity of the LFSR sequences by applying an irregular clocking, eliminating or repeating some bits, and hence provide the resulting sequence with a large linear complexity. The simplest schemes use only two registers; Usually, the first one called *Control Register, CR*, is clocked regularly, and its output controls the clocking of the second one called *Generator Register*, *GR*, which is produced an irregular sequence [1].

There are several kinds of clock-controlled scheme, *e.g.* Stop/Go Generator [1, 2], Step1/Step2 Generator [1], Shrinking Generator [4], Self-Shrinking Generator [5], Alternative Step Generator [12], Alternative Step(*r,s*) Generator [13], and Jump Register which is proposed recently in

[6, 7, 8] and it is used in some candidates to the European ECRYPT/eSTREAM project [9], *e.g.* Pomaranch [10] and Mickey [11].

Recently, jumping clock-controlled becomes interesting in the literatures, *e.g.* Step(*D,K*) Generator [1], Alternative Step(*r,s*) Generator, Pomaranch, Mickey and *etc*. The Step(*D,K*) Generator is a general form of a clock-controlled with jumping manner proposed by Gollmann and Chambers in [1].

Our motivation is finding an efficient method to analyze the stream ciphers based on jumping register, *e.g.* Step(*D,K*) Generator, Pomaranch and Mickey. In this paper, we will discuss the security of the Step(*D,K*) Generator according to its parameters *D* and *K*.

THE STEP(*D,K*) GENERATOR DEFINITION

A Step(*D,K*) Generator composed of two registers; the first one, *CR*, is clocked normally but the clocking of the second register, *GR*, determined by the current state of the first register, *CR*. The second register clocked *D* times if the controller bit is 1 and *K* times otherwise. The controller bit can be determined with the output of a function, *f*, depending on the current state of the first register, *CR*. The simple structure of this system observed in figure 1. Most of the different clock-controlled generators can be derived from a Step(*D,K*) generator with special parameters, *e.g.* the Stop/Go Generator is a Step(0,1) Generator, the Step1/Step2 Generator is a Step(1,2) Generator, the Alternating Step Generator is composed of two Step(0,1) Generator, the Alternating Step(*r,s*) Generator is composed of Step(0,*r*) and Step(0,*s*) Generator.

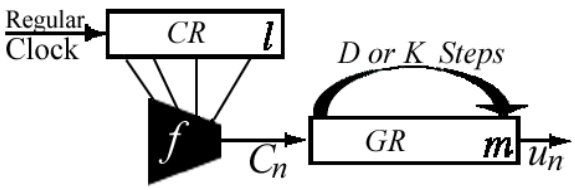

Figure 1: Step(*D,K*) Structure

Suppose $\{b_t\}=b_1,b_2\ldots b_t$ denotes the regular output sequence of *GR* and $\{u_n\}=u_1,u_2\ldots u_n$ denotes its irregular output sequence. In other words,

$\{u_n\}$ is a subsequence of the sequence $\{b_t\}$. Let $\{c_n\}$ denotes the output of the function f which determines the number of clock for GR at time n, and takes its elements from $\{D,K\}$ set. If the feedback polynomials of CR and GR register are chosen a primitive polynomial of degree L_1 and L_2 respectively, then the period, P, and the linear complexity, L, of the output sequence is [1]:

$$P = (2^{L1} - 1)(2^{L2} - 1) \tag{1}$$

$$L = L_2 * \mathrm{A} \tag{2}$$

Where, A denotes the primitive factors of $(2^{L1}-1)$, which are common with the factors of $(2^{L2}-1)$. In [1], it is showed that the output sequences of the Step(*D,K*) Generators have good statistical properties.

THE STEP(*D,K*) GENERATOR'S SECURITY RESPECT TO ITS PARAMETERS

In this section, the security of the Step(*D,K*) Generator depend on the parameters D and K is investigated. In this section, we suppose that D and K are nonzero positive known integer. Our main idea refers to a weakness arising from the probability of decimating a bit from the regular output sequence $\{b_t\}$ to produce the irregular output sequence $\{u_n\}$. If some bits from the sequence $\{b_t\}$ are deleted with probability close to 1, attacker can fine some information about the $\{b_t\}$ and reduce his attack's complexity. Having more confusion, all bits in the $\{b_t\}$ should have the same probability to appear in the $\{u_n\}$. It is the main idea in our investigation about the security of the Step(*D,K*) Generator depend on the parameters D and K.

In order to choose the n^{th} bits from $\{b_t\}$ to appear in $\{u_n\}$, we must to reach this bit with x jumps with D-length and y jumps with K-length. In other words, the following equation must to have at last one nonnegative answer:

$$Dx + Ky = n \qquad D, K, x, y \in \mathrm{N} \tag{3}$$

The equation (3) has an answer if and only if we have:

$$\rho = gcd(D,K) | n \tag{4}$$

In case of ρ >1, there are some values for n that they don't satisfy condition (4) and there are no answer for (3). It means that we have some bits in the $\{b_t\}$ that they don't turn up in $\{u_n\}$ with probability equal 1. Only one bit from each ρ consecutive bits from $\{b_t\}$ has good chance to turn up in the $\{u_n\}$, and (ρ -1) bits will be deleted from $\{b_t\}$ certainly. If (3) has an answer then it will have infinite answer in Z, but we want an answer in natural number set (N). It is possible that some n satisfy (4) but there is no answer in natural number set (N). Therefore, the best choice for ρ is 1 to satisfy (4) for all values for n. So, we should have:

$$gcd(D,K)=1 \tag{5}$$

Under this condition, the equation (3) has answer for any n in Z, but it is possible that there is not any nonnegative answer for x and y in N. In the rest of the paper, this problem is discussed by calculating the probability of appearing any bits from $\{b_t\}$ into $\{u_n\}$. Then, it will show that D and K must be a small number as much as possible.

Theorem 1: If P(n) denotes the probability of appearing the n^{th} bit of $\{b_t\}$ into the output sequence$\{u_n\}$ in the Step(D,K) Generator, we have:

$$\mathrm{P}(n)=\delta(n-1)+P_D\delta(n-(D+1))+\varphi(n-(D+2))\mathrm{P}(n-D)P_D+\varphi(n-(K+1))\mathrm{P}(n-K)P_K \tag{6}$$

where P_D is the probability of $c_n=D$ and $P_K=1-P_D$ is the probability of $c_n=K$. the functions $\varphi(n)$ and $\delta(n)$ are defined in (7):

$$\delta(n)=\begin{cases}0 & n\neq 0\\ 1 & n=0\end{cases} \quad \text{and} \quad \varphi(n)=\begin{cases}0 & n<0\\ 1 & n\geq 0\end{cases} \tag{7}$$

Proof: Without loosing any generality, we can suppose that $D<K$ and the first bit of $\{u_n\}$ is the first bit in $\{b_t\}$, i.e. $u_1=b_1$, Therefore we have:

$$n=1 \quad \Rightarrow \mathrm{P}(n)=1 \tag{8}$$

For n=1 in equation (6), in the right hand, all terms are zero except $\delta(n\text{-}1)$ that it is equal 1. So, equation (6) is true for n=1. We supposed that $D<K$, so no bits can turn up in the output sequence, if it is between the first bit in $\{b_t\}$ and $(D+1)^{th}$ bit. Therefore we have:

$$1<n\leq D \quad \Rightarrow P(n)=0 \tag{9}$$

For $1<n\leq D$, in the right hand of (6) all terms are zero. So, equation (6) is true in this case. The probability of applying exactly the D clocks to produce the next output bit is P_D. Therefore we have:

$$n=D+1 \quad \Rightarrow P(n)=P_D \tag{10}$$

To choose any bits between $(D+1)^{th}$ bit and K^{th} bit from $\{b_t\}$, we have to use irregular clocking only with D-jump. So, to appear n^{th} bit, $n\leq K$, from $\{b_t\}$ in the output, the $(n\text{-}D)^{th}$ bit must be chosen in the previous stage and then we have to apply D clocks to GR. Therefore we have:

$$D+1<n\leq K \quad \Rightarrow P(n)= P(n\text{-}D)P_D \tag{11}$$

Where $P(n\text{-}D)$ is the probability of choosing the $(n\text{-}D)^{th}$ bit from $\{b_n\}$ and P_D is the probability of applying D clocks to GR. For $D+1<n\leq K$, in the right hand of (6), only $\varphi(n\text{-}(D+2))$ is nonzero and it is equal 1. So, the (6) is true in this case. To choose any bits in area $n>K$, there are two ways. The first one is that we have to choose the $(n\text{-}D)^{th}$ bit in the previous stage and apply D clocks to GR. The second one is that we have to choose the $(n\text{-}K)^{th}$ bit in the previous stage and apply K clocks to GR. Therefore, we have:

$$n>K \quad \Rightarrow P(n)= P(n\text{-}D)P_D + P(n\text{-}K)P_K \tag{12}$$

For this area, $n>K$, in the right hand of (6) the $\varphi(n\text{-}(D+2))$ and $\varphi(n\text{-}(K+1))$ are nonzero and they are equal 1. So, the equation (6) is true. Therefore, from (8) to (12) in general we have:

$$P(n)=\delta(n\text{-}1)+P_D\delta(n\text{-}(D+1))+\varphi(n\text{-}(D+2))P(n\text{-}D)P_D+\varphi(n\text{-}(K+1))P(n\text{-}K)P_K \tag{13}$$

By (6), we are able to calculate and plot the $P(n)$ for any D and K respect to n easily. Figure 2 and 3 illustrate the graph of $P(n)$ for some D and K. It can be observed easily that P(n) tend to a fixed value for small parameters D and K. We can divide the curve of P(n) into two parts. The first part is the area that the amount of P(n) changes rapidly. We refer to this part by "*Transition Area*". The second part is the area that the value of P(n) is fixed. The length of transition area is independent from the length of register *GR*, but the parameters D and K. For small D and K, the length of this area is short, but it is very long for big parameters which illustrated in figure 4.

The transition area is very important, because all bits that appear in the output during this area belong to the register *GR*'s initial state. From the variety of P(n) in this area, attacker can recognize that which bits of the initial state will appear in the output with good probability. So, he can find some information about the initial state. If the length of the transition area becomes bigger and the range of variety in the curve becomes bigger, attacker can fine more information about the initial state. The result of our above discus is that the length of transition area should be short as much as possible to have a good stream cipher.

Our investigation has illustrated that for the large parameters D and K, we have a large ripple in the transition area and the length of transition area become bigger. For example, figure 4 illustrates that for K=17 and any amount of D which satisfy relation (5), some n exist such that the equation (3) doesn't have any nonnegative answer and P(n) is zero. In these examples, the length of transition area is very large. Attacker can reduce his complexity with eliminating the bits whose probability to appear in the output is close to zero. In most of the general attacks on the stream cipher, the time complexity depends to the sequences that used in the attack, e.g. correlation attacks.

Figure 5 illustrates some systems which are suitable for using in stream ciphers, because they have very short transition area and also all bits from $\{b_t\}$ has the same probability to appear in the output sequence. But, the systems illustrated in figure 4 do not have these properties. Most of the people believe that more eliminated bits from the regular output sequence can hide more information about the initial state. For example, in jump registers or ASG(r,s), designers have this idea. In [3], it is showed that the security of ASG(r,s) is not more than a original ASG. In next section, we

will show that this idea is not true, if attacker know the position of deleted bits.

IMPROVE THE ATTACKS USING OUR OBSERVATION

In this section, we explain that how we can use of our observation to improve the attacks on the Step(D,K) Generator. In the most of the attacks on the clock-controlled generators, e.g. [14-17], the time and space complexity depend on the length of the regular sequences which are used in the attack to find the initial state. Our observation can reduce the length of the regular sequences and improve the attacks.

For example, we explain that how our idea can improve the correlation attack based on the Levenshtein Distance presented by J. Golic in [17]. The same idea can be used to improve other attacks. In the correlation attack based on the Levenshtein Distance, we have to guess the length of the regular sequence, M, which can produce the given irregular output. In case of Step(1,2) Generator, Golic recommended to consider M=3N/2 while N is the length of the given irregular output. The time and space complexity of finding the *Levenshtein Distance* is C_t=O(M(M-N)) and C_s=O(M-N) respectively. As you can see, both complexities depend on the length of the regular sequence, M. the time complexity of this correlation attack is equal to C=O(M(M-N)2^L) while L is the length of LFSR. To use of our observation to improve Golic's attack, we have to eliminate E bits from the original sequence that their probability to appear in the output is close to zero. Then, we have to use the remained bits, M*=M-E, as the regular sequence to find the Levenshtein Distance. If we eliminate E bits from the regular sequence, the time complexity of the algorithm that finds the Levenshtein Distance will reduce to C_{t1}=O(M*(M*-N)) which is O(E(2M-N-E)) better than Golic's algorithm:

$$C_{t1}=O(M^*(M^*-N))=O((M-E)((M-E)-N))=C_t-O(E(2M-N-E)) \qquad (14)$$

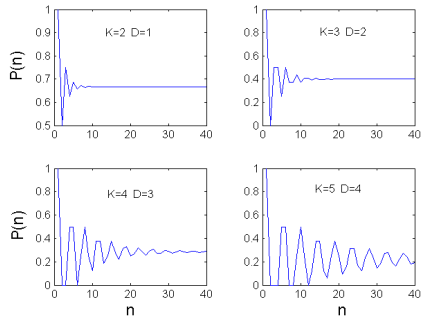

Figure 2: P(n) for some small D and K

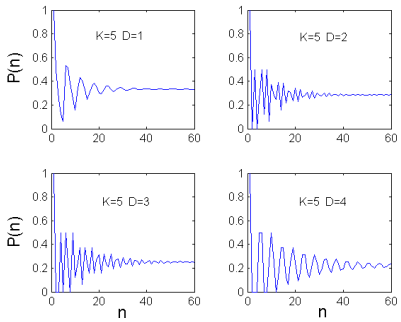

Figure 3: P(n) for K=5 and D=1,2,3,4

Therefore the time complexity of this attack will reduce to:

$$C^*=O(M^*(M^*-N)2^L)=C-O(E(2M-N-E)2^L) \quad (15)$$

It means we can improve the time complexity of the attack by $O(E(2M-N-E)2^L)$. And also for the space complexity of the algorithm that finds the Levenshtein Distance we have:

$$C_{s1}=O(M^*-N)=O((M-E)-N)=C_s-O(E) \quad (16)$$

The correlation attack based on the Levenshtein Distance [17] produce n_0= $1+(2^L-2)P_f$ possible initial state as results that we have to check them to find the correct answer. P_f is the probability of "The false alarm". It is the probability of event that a sequence is wrongly considered as the generator sequence for the given irregular output sequence. When we reduce the length of the regular sequences by using our idea, we reduce the flexibility of regular sequences to match with the given irregular output. Therefore, we reduce number of "The false alarm" and its probability, P_f. It means we reduce the number of possible solution, n_0. According to the section 2, for large parameters D and K, the value of E will be increase and we will have more improvement in the attacks complexity. If we eliminate only one bit, *i.e.* E=1, we will reduce the complexity of the attack by $O((2M-N-1)2^L)$ which is significant reduction.

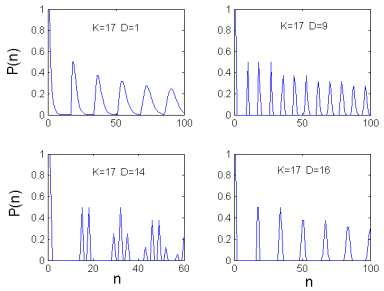

Figure 4: P(*n*) for *K*=17 and *D*=1,9,14,16

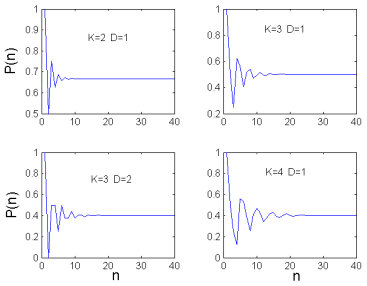

Figure 5: P(*n*) for some strong systems

CONCLUSIONS AND OPEN PROBLEMS

In this paper, we explained our observation about the Step(*D,K*) Generator and discuss the effect of the parameters *D* and *K* on the security of the structure. We support our observation with some examples and our result from implementations. We explain that the bad parameters for the Step(*D,K*) Generator can release some information about the initial state and cause to decrease the complexity of the attacks. For example, we show that how we can reduce the time complexity of the correlation attack based on the Levenshtein Distance by $O(E(2M\text{-}N\text{-}E)2^{L})$. Therefore, we have to be careful to use of clock-controlled stream ciphers. In the following, we recommend some important notes to design the Step(*D,K*) Generators:

1. The curve of P(*n*) should have a short transition area.
2. The curve of P(*n*) should be smooth in transition area.
3. The parameters *D* and *K* should be co-prime.
4. The parameters *D* and *K* should choose a small number as much as possible.
5. The security of system should be investigated by drawing the graph of P(*n*) for designed system.

To continue this work in the future, we try to use of our observation to apply an attack on the jump register which is used in some stream ciphers, *e.g.* Pomaranch and Mikey. They use of registers with large jumping, so we believe that our observation can reduce the complexity of attacks on these algorithms.

REFERENCES

[1] Gollmann, D. and Chambers, W.G. 1989. Clock-controlled shift registers: a review, *Selected Areas in Communications, IEEE Journal,* **7**(4): 525-533.

[2] Beth, T. and Piper, F. 1985. The Stop and Go Generator, *Advances in Cryptology*: *Eurocrypt* 84, Lecture Notes in Computer Science, Berlin: Springer-Verlag, **209**: 88-92.

[3] Hassanzadeh, Mehdi M. and Helleseth, Tor. 2010. Algebraic Attack on the Alternating Step(*r,s*) Generator, *IEEE International Symposium on Information Theory (ISIT2010)*, June 13-18, 2010, Austin, Texas.

[4] Coppersmith, D., Krawczyk, H. and Mansour, Y. 1993. The Shrinking. *Generator, CRYPTO*: 22-39.

[5] Meier, W. and Staffelbach, O. 1995. The self-shrinking generator, In A. De Santis, editor, *Advances in Cryptology - Eurocrypt '94*, *LNCS*, **950**: 205–214.

[6] Jansen, C.J.A. 2002. Modern stream cipher design: A new view on multiple clocking and irreducible polynomials, In: Gonz´alez, S., Mart´ınez, C. (eds.) *Actas de la VII Reuni´on Espa˜nola sobre Criptolog´ıa y Seguridad de la Informaci´on.* Volume Tomo I. Servicio de Publicaciones de la Universidad de Oviedo: 11–29.

[7] Jansen, C.J.A. 2005. Partitions of polynomials: Stream ciphers based on jumping shift registers, In: Cardinal, J., Cerf, N., Delgrange, O., Markowitch, O. (eds.) *26th Symposium on Inf. Theory in the Benelux*, Enschede, Werkgemeenschap voor Informatie- en Communicatietheorie: 277–284.

[8] Jansen, C.J.A. 2005. Stream cipher design based on jumping finite state machines, *Cryptology ePrint Archive*, Report 2005/267, http://eprint.iacr.org/2005/267/.

[9] Ecrypt Stream Cipher Project. http://www.ecrypt.eu. org/stream/.

[10] Jansen, Cees J. A., Helleseth, Tor and Alexander Kholosha. 2008. Cascade Jump Controlled Sequence Generator and Pomaranch Stream Cipher, *LNCS*, **4986**: 224-243.

[11] Steve Babbage and Matthew Dodd. 2008. The MICKEY Stream Ciphers**,** *LNCS*, **4986**: 191-209, Springer, and the ECRYPT/eSTREAM project: 224-243.

[12] Günther, C.G. 1988. Alternating Step Generators Controlled by De Bruijn Sequences, *Advances in Cryptology: Eurocrypt* 87, LNCS, Spingler-Verlag, **309**: 5-14.

[13] Kanso, A. 2002. The Alternating Step(r, s) Generator, *SECI02*, Tunis.

[14] Jiang, S. and Gong, G. 2003. On Edit Distance Attack to Alternating Step Generator, In *Other Combinatorial Structures*: 85–92.

[15] Golic, J. Dj. and Menicocci, R. 1997. Edit Distance Correlation Attack on the Alternating Step Generator, In *CRYPTO*: 499–512.

[16] Golic, J. Dj. 2005. Embedding probabilities for the Alternating Step Generator, In *IEEE Transactions on Information Theory* **51**(7): 2543–2553.

[17] Golic, J. Dj. and Mihaljevic, M. J. 1991. A Generalized Correlation Attack on a Class of Stream Ciphers Based on the Levenshtein Distance, *J. Cryptology*, **3**: 201-212.