Studying the relationship between linguistic variables and the degrees of primitive polynomials used in pseudo-random number generator based on fuzzy logic

I.V Anikin[1], K. Alnajjar[1]

[1]Information Security Systems Department, Kazan National Research Technical University named after A.N. Tupolev-KAI, 420111, Kazan, Russia.

Abstract. In this paper we study the relation between linguistic variables and the degree of characteristic primitive polynomial of used LFSR in previously suggested fuzzy pseudorandom number generator FRNG, keeping the out pseudorandom series secure. This means that we should take into account two important properties: the randomness of the output and the security against correlation attacks. The first property mainly depends on selection of primitive polynomials used in constructing FRNG (this was discussed deeply in [1, 2]). The second property as described in [2] can be realized by making the output series balanced in sense that the probability of appearing the bits of each used LFSR approximately equal to others in the output stream.

Keywords: pseudo-random numbers generator, fuzzy logic, linguistic variable, primitive polynomial.

1. Introduction

As described in [1] the structure of the suggested FRNG consists of number of LFSRs (to simplify the study we will use only two). The outputs of LFSRs go through 32 bits sized buffers where an estimation of two fuzzy linguistic variables is done - the first involves in evaluating number of ones (f_0) in the buffer, and the second ($|f_1$-$f_2|$ or f_{12}) in estimating the difference between number of blocks (0110) of two consecutive ones (f_1) and the number of gaps (1001) consist of two consecutive zeros (f_2) in the considered buffer for every bit. Then a group of fuzzy If-Then rules plays main role in deciding which one of the used LFSRs is best at every moment and pass it's out bit to be a bit of the pseudorandom series generated by FRNG then a new estimation of the linguistic variables associated with every buffer begins (after shifting the continent of the buffers one bit to the right and inserting a new bit from the related LFSR) to select the next bit and pass it to the output of the system and so on. ‘Figure 1’ illustrates the general structure of the proposed FRNG. It's very important here to describe the linguistic variables f_0, f_{12} in details, because they are the main keys of this study. As seen in ‘figure 1’ these two linguistic variables play essential role with the If-Then rules in deciding which LFSR's bit will be passed to the output of the FRNG depending on the results of comparing the output of combining these two fuzzy variables [3] for each LFSR' buffer and then selecting the best one at every bit.

As a result of studying the parameters of FRNG in [2], we found that every linguistic variable has three membership functions (MFs); (Low, Medium, High) for the first variable f_0 and (Excellent, Good, Bad) for the second f_{12}.

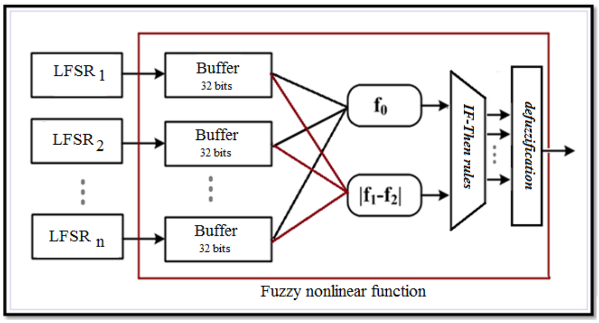

Figure 1. General structure of proposed FRNG.

Then a group of fuzzy rules shown in ‘table 1’ helps in estimating the statistical situation of the considered buffer at the moment through combining the two variables, then in the last step we compare the obtained results for all used LFSRs and select the best one of them and pass its bit to be the output of the FRNG, then the process continues in this way repeatedly for every bit.

Table 1. Fuzzy If-Then rules FRNG.

f_0	f_{12}	*result*
Low	Excellent	Bad
Low	Good	Bad
Low	Bad	Bad
Good	Excellent	Best
Good	Good	Good
Good	Bad	Bad
High	Excellent	Bad
High	Good	Bad
High	Bad	Bad

Settings of the fuzzy groups for every MF sufficiently affect the output of the pseudorandom generator. In [2] we studied their influence on the security of the proposed generator FRNG against correlation attacks [4], we concluded that the output of FRNG should be balanced to have a secure generator.

Balancing the output series could be achieved via tuning the MFs of the fuzzy linguistic variables f_0 and f_{12} for each LFSR separately, and here lies the main idea of this paper.

2. Tuning the MFs of the fuzzy linguistic variables f_0, f_{12}

Tuning the MFs of the linguistic variables f_0, f_{12} will directly affect the probability of appearing the bits of the related LFSR in the generated pseudorandom sequence. So any change in the configurations of the MFs will increase or decrease the percent of appearing its bits in the output of the FRNG. These changes shouldn't disturb the balance of the FRNG. The statistical property of the output of LFSR is mainly depend on its characteristic polynomial [5]. In [2] we decided to use a special type of primitive polynomials to construct our generator, and we found that this type of polynomials sufficiently increases the efficiency of FRNG and has many practical advantages. The following formula briefly describes this type of polynomials:

$$f(x) = (1+x^{b_1})(1+x^{b_2})...(1+x^{b_m}) + x^n \quad (1)$$

In addition to the primitivity tests, the parameters $(b_1,b_2,...,b_m,n)$ of selected polynomial should satisfy the following conditions:

$$b_1 \geq 1, b_1 < b_2, \left(b_1 + b_2\right) < b_3, \cdots, \left(b_1 + b_2 + ... + b_{m-1}\right) < b_m, \left(b_1 + b_2 + ... + b_m\right) < n$$

For simplicity we will use a simple version of FRNG that contains only two LFSRs, and we will investigate the relationship between the degrees of primitive characteristic polynomials and the associated MFs of the fuzzy linguistic variables f_0 and f_{12} for each of them, regarding the balance of output series of FRNG. So at the beginning we will briefly explain how to calculate the probability of appearing one of LFSR's bits in the output of the FRNG. Then we will discuss the process of tuning the MFs regarding the obtained value of P. Then Through number of examples we will study how these configurations related to the degrees of used characteristic primitive polynomials of LFSRs.

2.1. Calculating the probability value P

For simplicity only two LFSRs used in constructing the FRNG so we will calculate one probability value P(out_{sys}=out_{LFSR2}) which denotes the probability that the output of generator is equal to the output of the LFSR2 whose degree is bigger. So the second value P(out_{sys}=out_{LFSR1}) will be equal to (1-P(out_{sys}=out_{LFSR2})) so calculating one value is enough to conclude the other and estimate the balance of the FRNG. We should repeatedly calculate this value at every modifications in the MFs settings of the linguistic variables. To calculate the value of probability P(out_{sys}=out_{LFSR2}) after tuning we should count how many bits of the output series selected by the fuzzy non-linear function from the output of LFSR2, then divide the resulting value on the total number of generated bits (the length of the series in our case 1024000).

Finally, it's very important to mention that the tuning process should be done also when setting a new key values to FRNG (when changing the seeds of used LFSRs). So the resulting configurations of the membership functions of linguistic variables (f_0, f_{12}) of every LFSR will definitely depend on the initiate state of used LFSRs (the seeds).

2.2. Tuning the MFs regarding the calculated value of probability P

Firstly we initiate the membership functions of linguistic variables (f_0, f_{12}) of every LFSR then generate a 1024000 bits then calculate the first value of probability P, and here starts tuning process. Calculated value of P is often far from 0.5, so the constructed FRNG in this case is not balanced, and here lies the necessity of tuning process to make the generator secure against the correlation attacks. ‘Figure 2’ illustrates the process of tuning the MFs of the linguistic variables (f_0, f_{12}).

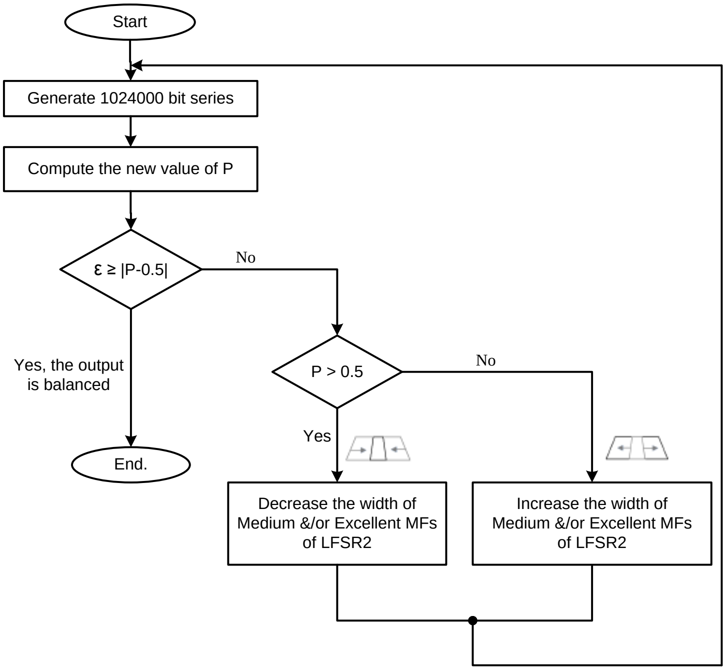
Start
Generate 1024000 bit series
Compute the new value of P
ε ≥ |P-0.5|
No
Yes, the output is balanced
End.
P > 0.5
No
Yes
Decrease the width of Medium &/or Excellent MFs of LFSR2
Increase the width of Medium &/or Excellent MFs of LFSR2

Figure 2. The process of tuning the MFs of linguistic variables of LFSR2.

At the beginning we start from the following initial configurations for both LFSRs:

- The first fuzzy linguistic variable f_0 has three MFs they are fuzzyficated as following: when f_0 belongs to {0,...,8} is mapped to {Low}, when f_0 belongs to {9,...,24} is mapped to {Medium}, and when f_0 belongs to {25,...,32} is mapped to {High}.
- The second fuzzy linguistic variable f_{12} has also three MFs they are fuzzyficated as following: when f_{12} belongs to {0,1,2} is mapped to {Excellent}, when f_{12} belongs to {3,...,6} is mapped to {Good}, and when f_{12} belongs to {7,...,10} is mapped to {Bad}.

Taking ‘table 1’ into account we see that the most important MF of the first linguistic variable f_0 is "Medium" and when its width increases (increasing number of elements in it's fuzzy group) the probability of appearing of the bits of related LFSR will sufficiently increase and vice versa. Also the same thing with the second linguistic variables f_{12} but with the first MF "Excellent" which is the most important one for the second variable. It's worth mentioning that the other MFs of both linguistic variables affect the calculated value of probability but their effect is smaller and tuning them is very useful when resulting value of probability P is not too far from 0.5. So if we want to increase the probability of appearing the output bits of the LFSR1 in the output we should increase the width of "Medium" MF of f_0 or increase the width of "Excellent" MF of f_{12} that are associated with LFSR1 or increase them both, or we have another option in tuning them by decreasing these parameters for LFSR2 to make it's probability of appearing less. Then we should evaluate the balance of the generated sequence by computing the new value of P, then according to this value the process of tuning will continue repeatedly until reaching $P=P(out_{sys}=out_{LFSR2})\cong 0.5$ within the acceptance value of difference $\varepsilon \geq |P-0.5|$.

Finally, it's very important to mention that the tuning process should be done also when setting a new key values to FRNG (when changing the seeds of used LFSRs). So the resulting configurations of the membership functions of linguistic variables (f_0, f_{12}) of every LFSR will definitely depend on the initiate state of used LFSRs (the seeds).

3. Studying the relationship between the configurations of linguistic variables and the degrees of used primitive polynomials of LFSRs

It's worth mention that all used primitive characteristic polynomials were accurately selected to have the defined type by equation (1), that makes them fit all the requirements of high performance and randomness. The selected polynomials has excellent statistical properties and they successfully passed the primitivity tests[6]. So here we will pay more attention on their degrees and how they related to the associated linguistic variables (f_0, f_{12}). To study the relationship between linguistic variables and the degrees of used primitive characteristic polynomials of LFSRs, we have made a number of numerical experiments using MATLAB environment (version 7.14.0.739 (R2012a)), in each of them we used different primitive polynomials with different degrees then we tuned the MFs as previously described to reach the balanced version of FRNG, then we recorded the obtained results in ‘table 2’.

It's clear from the resulting table that there is a strong relation between the degrees of used polynomials and the configurations of membership functions (setting of the fuzzy set of every MF of the linguistic variables f_0, f_{12}). As seen there is an inversely proportional relationship between the degree of the polynomial and the width of Medium and Excellent membership functions of the linguistic variables f_0, f_{12} respectively. Using a polynomial with a big degree will lead us to make the width of Medium and/or Excellent membership functions of the associated linguistic variables f_0, f_{12} more narrow relatively depending on the resulting value of the calculated probability P. Also from the practical experiments especially the last one (number 5 in the table 2) we can conclude that this relation mainly depends on the difference between the degrees of used polynomials; so using a polynomials with a big differ in degrees will lead to a big differ in configurations of membership functions of the linguistic variables. So the width of Medium and Excellent membership functions associated with the polynomial that has bigger degree will be narrower and vice versa for the other polynomial.

Finally, we can say that the new version of FRNG became more secure against algebraic attacks by virtue of using such kind of polynomials that have high diffusion capacity and high linear complexity, in addition to getting benefits of using the fuzzy logic techniques to build a non-linear function to combine two LFSRs that makes the proposed generator more secure and has high immunity against correlation attacks.

Table 2. The experimental results of tuning process.

Exp.N.	*Primitive Polynomials used in constructing the FRNG*	*Balanced configurations of MFs*		*resulting P*
		f₀ → f_0	f_{12}	
1	$P_1 = (1+x)(1+x^2)(1+x^4)(1+x^9)(1+x^{41}) + x^{67}$	Low : $\{0,..,12\}$ Medium : $\{13,..,19\}$ High : $\{20,..,32\}$	Excellent : $\{0,1\}$ Good : $\{2,..,6\}$ Bad : $\{7,..,10\}$	0.5017

	$P_2 = (1+x^3)(1+x^4)(1+x^9)(1+x^{17})(1+x^{37}) + x^{71}$	Low:{0,..,14}
		Medium:{15,..,17}
		High:{18,..,32}
			Excellent : $\{0,1,2\}$
			Good : $\{3,..,6\}$
			Bad : $\{7,...,10\}$	

2	$P_1 = (1+x^2)(1+x^5)(1+x^8)(1+x^{19})(1+x^{41}) + x^{79}$	Low : $\{0,..,7\}$
		Medium : $\{8,..,22\}$
		High : $\{23,..,32\}$
			Excellent:{0,..,2}
			Good:{3,4}
			Bad:{5,..,10}
				0.5004

	$P_2 = (1+x)(1+x^5)(1+x^{10})(1+x^{17})(1+x^{39}) + x^{89}$	Low={0,..,13}
		Medium:{14,..,17}
		High:{18,..,32}
			Excellent:{0,..,3}
			Good:{4,..,7}
			Bad:{8,..,10}	

3	$P_1 = (1+x)(1+x^5)(1+x^{10})(1+x^{17})(1+x^{39}) + x^{89}$	Low : $\{0,..,9\}$
		Medium : $\{10,..,20\}$
		High : $\{21,..,32\}$
			Excellent:{0,..,4}
			Good:{5,6,7}
			Bad:{8,..,10}
				0.5017

	$P_2 = (1+x)(1+x^4)(1+x^7)(1+x^{20})(1+x^{53}) + x^{97}$	Low:{0,..,11}
		Medium:{12,..,18}
		High:{19,..,32}
			Excellent:{0,1}
			Good:{2,3}
			Bad:{4,..,10}	

4	$P_1 = (1+x)(1+x^4)(1+x^7)(1+x^{20})(1+x^{53}) + x^{97}$	Low:{0,..,8}
		Medium:{9,..,23}
		High:{24,..,32}
			Excellent:{0,..,3}
			Good:{4,5}
			Bad:{6,..,10}
				0.4980

	$P_2 = (1+x)(1+x^2)(1+x^9)(1+x^{19})(1+x^{70}) + x^{103}$	Low:{0,..,12}
		Medium:{13,..,19}
		High:{20,..,32}
			Excellent:{0}
			Good:{2,..,7}
			Bad:{8,..,10}	

5	$P_1 = (1+x^2)(1+x^3)(1+x^{15})(1+x^{39}) + x^{61}$	Low:{0,..,8}
		Medium:{9,..,20}
		High:{21,..,32}
			Excellent:{0,1,2}
			Good:{3,4,5}
			Bad:{6,..,10}
				0.4962

	$P_2 = (1+x)(1+x^3)(1+x^5)(1+x^{10})(1+x^{27})(1+x^{69}) + x^{149}$	Low:{0,..,12}
		Medium:{13,..,18}
		High:{19,..,32}
			Excellent:{0,1}
			Good:{2,3}
			Bad:{4,..,10}	

4. Conclusion

According to the obtained results of this study we can conclude that there is a very strong and mutual relationship between the configurations of the MFs that are related to the fuzzy linguistic variables and the degree of the characteristic primitive polynomial of the associated LFSR. And applying the tuning process is very important and it sufficiently increases the security of the proposed pseudo-random number generator based on fuzzy logic against the correlation attacks.

As a result the generated sequences by the balanced version of FRNG will have low correlation magnitude, high linear complexity, less power consumption, and they will have very good statistical properties that make the resulting FRNG fits the requirements of most telecommunication applications such as cryptography, authentication, etc.

In the future, we will try to use greater number of LFSRs in constructing the generator (for example 4 or 8) in order to increase the period of the suggested FRNG to get a pseudorandom number sequences that are very close to true random.

5. References

[1] Anikin, I.V. Fuzzy stream cipher system / I.V. Anikin, K. Alnajjar // Proc. Int. Siberian Conf. on Control and Communications. – Omsk, 2015. – P. 64-68.

[2] Anikin, I.V. Pseudo-random number generator based on fuzzy logic / I.V. Anikin, K. Alnajjar // Proc. Int. Siberian Conf. on Control and Communications Moscow. – 2016. – P. 68-72.

[3] Zadeh, L.A. Fuzzy Sets / L.A. Zadeh // Information and control. – 1965. – Vol. 8(3). – P. 338-353.

[4] Siegenthaler, T. Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications / T. Siegenthaler // IEEE Transactions on Information Theory. – 1984. – Vol. 30(5). – P. 776-780.

[5] Klapper, A. Algebraic feedback shift registers / A. Klapper, J. Xu // Theoretical Computer Science. – Vol. 226(1999). – P. 61-93.

[6] Menezes, A.J. Handbook of Applied Cryptography / A.J. Menezes, P.C. van Oorschot // Vanstone – Boca Raton: CRC Press, 1997.