A&QT-R 2004 (THETA 14)
2004 IEEE-TTTC - International Conference on Automation, Quality and Testing, Robotics
May 13 – 15, 2004, Cluj-Napoca, Romania

ORTHOGONAL ARRAYS BASED PSEUDORANDOM NUMBERS GENERATOR

T.A. Mazurova, A.G. Chefranov

Taganrog State University of Radio-Engineering, Taganrog, Russia, mta777@rambler.ru
Eastern Mediterranean University, Gazimagusa, North Cyprus,
Alexander.chefranov@emu.edu.tr

Abstract. We propose a pseudo random numbers generator based on the usage of orthogonal arrays. It provides long period sequences (greater than 10^{100}). Experiments showed quality of the proposed algorithm similar to of known ones. It may be used as stream cipher with number of possible keys estimated from below as 10^{500} which provides high level of security. Mixing functions used in it (as part of a key) can vary which makes full algorithm to be not known to opponent.

Key words: orthogonal array, pseudo random generator

1. Introduction

Orthogonal array OA(L, k) of level L and strength k provides in L^k rows of L-columned array with elements from SL={0,..,L-1} all possible combinations of these numbers in any subset of k columns[1]. If we take any k columns and run through all the rows, the sequence of encountered numbers will look random. Sequence of numbers generated in such a way from SL will be without repetitions since each possible combination of numbers is guaranteed to appear just once.

Previously, we have suggested approaches to PRNG (pseudo random numbers generators) built on the base of OA. In one [2, 3] of these algorithms (A-1) it was assumed that L and k are small (L=7, k=4, for example). In such a case, before generating, we created OA(L,k) and then its elements were used multiple times. To get sequences with large periods we used enumeration of all possible combinations of k columns out of L-1 (their number is equal to (L-1)!/k! =6*5=30 for the case of L=7, k=4, not considering the 1st column). Then all possible permutations of such an enumeration were performed (their number is 30!, approximately 10^{30} combinations). Each of such enumerations outputs L^k*k=10000 numbers for the given enumeration of rows, but if we consider all L^k! permutations of the rows also, we get approximately 10^{7500} such combinations without repetition for mentioned above values of L and k. Such long period sequences may be viewed as infinite, but due to the small number of raw data which is to be mixed again and again, consideration of just sequence of OA elements is not sufficient from the point of view of statistical quality, and it should be transformed by additional operations.

Other previously considered [3] approach (algorithm A-2) doesn't assume the preliminary calculation of OA's elements, but these elements are obtained on-line according to formula proposed in [4], which is a concretization for prime L of Bush's construction considered in [1], where L is a prime power. Such an approach allows us to work with large numbers of L and k (for example, L=257, k=43 as in [3]) because we don't need to simultaneously store all L^k*L elements. The idea in PRNG here is to use a sequence of rows elements, starting from some arbitrary row. Such sequence will be not repeatable (it is shown that OA rows are pair wise different) and rather long (257^{43}*257, approximately 10^{100}). To get more security strength, each row before output was mixed with some vector obtained by the other PRNG. These approaches were tested and compared with RC4 [5, p. 194-195] and standard Delphi (SD) generator in [3]. They showed similar characteristics, A-1 having less performance, because relative cost of control is significant (we are to define next combination of columns or next permutation of rows after 2500 iterations, and we need in extra efforts to mix outputs for getting good randomness).

Here we present OA-based PRNG A-3 which may be viewed as combination of previously considered A-1 and A-2. It does not use a preliminary built OA as A-2, but scans combinations of columns as A-1, which is guessed to improve statistical quality in comparison with A-1, having computational complexity as A-2. Due to the large values of L and k, A-3 runs just through two columns combinations, yet providing a long period sequence.

The remainder of the paper is organized as follows. Section 2 introduces new algorithm A-3, section 3 presents experimental results of testing A-3 in comparison with A-2, RC4 and SD. Section 4 contains the conclusion.

2. OA-based proposed algorithm

PRNG algorithm A-3 is given in Fig. 1.

```
1. Take L, k sufficiently large, L being a prime, k<L (for example, L=257, k=140), two combinations C1[k], C2[k] of k column numbers, value of initial row R=R0 from {0,..,$L^k$-1}, NR – number of rounds for mixing, Fi(x,y), i=1,..,NR – set of functions for mixing integers x,y from SL, outputting also integer from SL.
2. Generate two sets S1[k], S2[k] of OA elements: S1[i]=OA[C1[i],R+L], S2[i]=OA[C2[i],R], i=1,..,k. These sets are elements of (R+L)-th and R-th OA's rows taken from positions pointed by C1 and C2 elements respectively.
3. Output $k^2$*NR values Fi(S1[l],S2[j]), l,j=1,..,k, i=1,..,NR.
4. Increment R: R=R+1 mod $L^k$
5. If (R<>R0) return to step 2.
6. Stop – we have scanned all rows of OA cyclically.
```

Fig. 1. Algorithm A-3

Algorithm A-3 provides a sequence of pseudorandom numbers without repetition with length= L^k*k^2*NR. In the case of L=257, k=140, NR=2 we shall have approximately length=10^{330}. Number of possible keys for sequence generation can be estimated as Keys=L^k*A(L,k)^2, where A(L,k)=L!/k! is a number of possible combinations of k numbers out of L and where we don't take into the account number of possible choices for mixing functions. For L=257, k=43, Keys >= 257^{43} *257! *257! /(43!*43!) >

$10^{100}*10^{200}*10^{200}>10^{500}$, which provides high cryptographic resistance of algorithm A-3. Values L, k, R together with NR and set of functions Fi(x,y), i=1,..,NR, form a seed for algorithm A-3, SEED=(L,k,R0,NR,Fi, i=1,..,NR). If we use A-3 as a stream cipher, the seed as a key need to be transferred between exchanging sides. Functions may be transferred as binaries and linked together with static part of the algorithm, so, generally speaking, such PRNG algorithm will not be known to an opponent.

Step 2 of A-3 requires calculation of elements of OA; it may be done by the following formula [4]:

$$i=\sum_{l=0}^{k-1} i_l\, L^l, i_l \in \{0,..,L-1\}, k \in \{1,..,L-1\},$$

$$OA[i,j]=(\sum_{l=0}^{k-1} i_l\, j^l) \bmod L, i=\overline{0,L^k-1}, j=0,L-1 \qquad (1)$$

where L is assumed to be prime, i is a row's number, j is a column's number, the number of columns is L, strength of the OA is represented by k. In the implementation of A-3 we have problem of termination: step 4 assumes taking modulo on L^k which can be a very large number for the mentioned values of L and k (257,140). To cope with this problem, row number i may be represented as a set I[k] of k values i_l from SL, treated as a k-digit number in the L-based system of numbers. The second issue is that powers of j in (1) may become very large also. To cope with this problem we may use the rule

(a*b) mod L = ((a mod L) *(b mod L)) mod L,

i.e. after finding next power we may find its modulo and use it in the further calculations.

3. Experimental algorithm evaluation and comparison

Algorithm A-3 was compared with A-2, RC4, SD. The current version of A-3 uses NR=2. Also, it uses SD to get an initial array IA of k bytes for mixing with A-3 outputs. The mixing operation for A-3 is just addition, but this result is further added with one of two possible elements of IA. It was observed that algorithm outputs twice greater zeroes than is necessary, so when zero is obtained, it is immediately replaced by some uniformly scattered non-zero value.

The following tests from [6] were used for comparison:

1) number of 1-s in generated bit sequence (length>=20000 bit), statistics is $F_1=\frac{1}{n}(\sum_{s=1}^{K}\frac{Q_s^2}{p_s})-n$, where K=2, p_s is estimated probability of appearance of s-th value, Q_s is really counted number of appearances of s-th value in the sequence of n bits, $s=\overline{0,1}$; must have asymptotic distribution as c^2 with K-1 levels of freedom;

2) number of m-bit vectors, statistics is given by: $F_2=\frac{2^m}{K}(\sum_{i=0}^{2^m-1}N_i^2)-K$, where N_i is the frequency of appearance of i-th vector in their sequence of n vectors; must have asymptotic distribution as c^2 with 2^m-1 levels of freedom;

3) number of sequences of 1-s and 0-s, statistics is $F_3 = \sum_{i=1}^{K} \frac{(B_i - E_i)^2}{E_i} + \sum_{i=1}^{K} \frac{(G_i - E_i)^2}{E_i}$, where B_i is number of series of 0-s of length i, G_i is number of series of 1-s of length i, $E_i=(n-i+3)/(2^{i+2})$ is expected number of series, K is maximal integer i for which $E_i > 5$; must have asymptotic distribution as c^2 with 2K-2 levels of freedom;

4) coefficient of sequential correlation showing dependency of the next symbol on the previous one, calculated as $F_4 = \frac{n(\sum_{i=0}^{n-2} U_i U_{i+1} + U_{n-1}U_0) - (\sum_{i=0}^{n-1} U_i)^2}{n\sum_{i=0}^{n-1} U_i^2 - (\sum_{i=0}^{n-1} U_i)^2}$. The value of this statistics must be in the range $[m_n - 2s_n, m_n + 2s_n]$ in 95% of occasions, where $m_n = -\frac{1}{(n-1)}$, $s_n = \frac{1}{(n-1)}\sqrt{\frac{n(n-3)}{n+1}}$, n>2;

5) the size Sz after compression of the sequence by archive utilities WinZip and WinRar as a ratio of resulting size to the initial one, measured in percents. It shows level of predictability of the sequence. Good sequence must not be compressed by such utilities.

Results of experiments conducted on 1.7 GHz Pentium with 256 Mb RAM are given in Table 1. Cells corresponding to violations of respective requirements are shown by shadowing. OA-based algorithms were studied for L=257 and two values of k: 43, 140. Experiments were conducted for sequences of 2 500 and 250 000 bytes (1st column of Table 1). On all such sequences algorithms showed nearly the same execution times, but RC4 and SD are slightly faster, up to 2 times. The second column of Table 1 contains names of measured characteristics. Ranges required for good randomness are given for F1-F4; numbers of levels of freedom are shown for F2, F3. Each row-characteristic is split into two sub rows for OA-2 and OA-3: top sub row corresponds to k=43, and next – to k=140. Statistics F1-F3 are shown for all algorithms together with range where they fall in. Last row of Table 1 contains Sz, it is assumed that result of compression is good enough if Sz>=95%. No one algorithm violated requirements on the short sequence of 2500, but there are violations for the long sequence of 250 000 bytes. Algorithm RC4 violated one requirement (not good F1), SD and OA-2 violated on F1, F2, F3. Algorithm OA-3 had not violations in the case of k=43, but had 3 violations for k=140. Best Sz of 100 was for RC4 and SD. In the case of OA-2 in some cases there were no reduction, but in some cases there was a reduction up to 33%. Compression of the sequences produced by OA-3 was not more than 95%. Results of the comparison show that OA-3 has characteristics not worse than those of RC4, SD and OA-2, but it guarantees periods of sequences more than 10^{100} (known estimation for the period of RC4).

4. Conclusion

Algorithm A-3 gives sequences with significantly less period than A-1 but this sequence is expected to be more random, less predictable, and also it should be less predictable than in A-2, because of less dependencies between subparts of neighboring OA rows and because of steady mixture of two streams of data taken from different rows (distance between them is L or more; this ensures absence of sets comprised of all same numbers, for example, all nulls are not possible). The sequence output by A-3 for considered values of L=257 and k=140 has a period of 10^{330} which is greater than for RC4 (estimated as 10^{100} [5, p.194]). The speed of A-3 algorithm is comparable with that provided by RC4. Experiments showed nearly the same characteristics of algorithms under investigation. If we use A-3 as a stream cipher, each next random number may be obtained by mixing also with the previous encoded symbol. Such approach may significantly increase the period of the generator, because each number will be obtained by mix of two internal streams (taken from different columns combinations) and a third external stream – ciphered text, available both for sender and receiver.

The work is supported by grants of Russian Fund for Basic Research 03-07-90075-?, 01-07-90211-??

References

1. A.S.Hedayat, N.J.A. Sloane, J.Stufken. [1999] Orthogonal arrays: theory and applications. N.Y. Springer-Verlag New York, 405 p.

2. Chefranov A.G., Mazurova T.A., Babenko L.K. [2002] About Application of orthogonal Arrays for Generating of Pseudorandom Sequences. - *Proceedings of the 4th International Workshop on Computer Science and Information Technologies CSIT-2002*, Editor P. Groumpos, 8-20 September 2002, Patras, Greece.

3. Chefranov A.G., Mazurova T.A., Sidorov I.D., Letia T. [2003] Orthogonal arrays application to pseudorandom numbers generation and optimization problems. – *Proceedings CSCS14. 14 Int. Conf. on Control Systems and Computer Science, 2-5 July, 2003, Politechnica University of Bucharest*, v.1, Editors Dumitrache I, Buiu C, Editura Politechnica Press, Bucharest, p. 254-259.

4. Chefranov A.G., Mazurova T.A. [2001] Support facilities of optimization of technological and organizational processes. - *Proc. 3rd Int. Workshop on Computer Science and Information Technologies, Ufa, Yangantau, Russia, Sept. 21-26, 2001*, Ufa: USATU, v.2, p. 45-49.

5. Stallings W. [2003] Cryptography and Network Security. Principles and Practices. 3rd Edition, Pearson Education Int., Upper Saddle River, 681 p.

6. Varfolomeev A.A., Zukov A.E., Pudovkina M.A. [2000] Stream cryptosystems. The basic properties and methods of cryptanalysis resistance. PAIMS, Moscow (in Russian).

Table 1. Results of experiments for PRNG RC4, OA-2, OA-3, SD

Number of bytes		RC4	L=257		SD
			OA-2	A-3	
2500	duration (sec)	0	0	0	0
			0	0	
	F4 [-4.04e-2; 3.96e-2]	0.0026	0.0094	0.01	0.026
			0.0094	0.0002	
	F3 16 [0.01;0.99]	22.22 [0.8;0.9]	18.58 [0.2;0.8]	21.41 [0.8;0.9]	8.05 [0.05;0.1]
			18.58 [0.2;0.8]	12.1 [0.2;0.8]	
	F2 256 [0.01;0.99]	235.5 [0.05;0.25]	301.66 [0.95;0.99]	340.06 [0.99;1]	235.51 [0.05;0.25]
			301.66 [0.95;0.99]	220.05 [0.01;0.5]	
	F1 [0.01;0.99]	0.06 [0.2;0.25]	5.45 [0.98;0.99]	1.48 [0.75;0.8]	0.03 [0.1;0.2]
			5.45 [0.98;0.99]	0.10 [0.2;0.25]	
250000	duration (sec)	0.07	0.12	0.11	0.06
			0.12	0.1	
	F4 [-4e-3;4e-3]	9.60E-06	0.0013	0.0007	-0.002
			0.0013	-0.0089	
	F3 30 [0.01;0.99]	35.18 [0.75;0.8]	406.14 [0.999;1]	32.23 [0.5;0.75]	316.42 [0.999;1]
			406.14 [0.999;1]	68.16 [0.999;1]	
	F2 256 [0.01;0.99]	248.56 [0.25;0.5]	6620.53 [0.99;1]	212.89 [0.01;0.05]	1252.48 [0.99;1]
			6620.53 [0.99;1]	556.32 [0.99;1]	
	F1 [0.01;0.99]	231.17 [0.999;1]	231.17 [0.999;1]	0.99 [0.5;0.75]	9.48 [0.99;0.999]
			231.17 [0.999;1]	24.65 [0.999;1]	
	Sz >=95	100.00	[62.7;100]	95.00	100
			[62.7;100]	98.00	