RANDOMIZATION GENERATORS

Igor A. Kulakov

Random Art Labs Limited
chief@random-art.com

Abstract. New methods for formation of sequences of (pseudo) random numbers with any repetition period given in advance are considered. The algorithms for realization of high-speed, single- and multi-digit unrepeated and equirepeated random-number generators, which allow efficient parallel processing, are presented. The conditions for obtaining their high cryptographic resistance are given. The analysis of their functional and statistic reliability is also presented.

In the paper new results obtained in the field of cryptography and development of stochastic systems [1] are presented. We focus on practical realization of high-speed random-number generators, which are simple at software and hardware realizations as well as statistically reliable and cryptographically resistant. They are considered as one of the most important elements of symmetric cryptography and stochastic systems.

Dichotomic generators *(Dh-generators)* are the basis of the random-number generators presented here [3]. Dichotomic generators are intended to form specific sequences, i.e., the so-called *dichotomic sequences* [2].

A binary sequence $D = \{d_i: i = \overline{1, T_n}\}$ of $T_n = 2^n$ nonnegative integers d_i, composed of n significant bits $b_{ij} \in d_i$ $(j = \overline{1, n})$, is called a *dichotomic* or *D-sequence* if the *perfect dichotomic order* is set up in it.

The perfect dichotomic order can be visually demonstrated using the binary representation of the first natural numbers:

00	0000	04	0100	08	1000	12	1100
01	0001	05	0101	09	1001	13	1101
02	0010	06	0110	10	1010	14	1110
03	0011	07	0111	11	1011	15	1111

The binary sequence formed this way shows that the repetition period T_k of every next k-th bit is equal to $T_k = 2^k$ $(k = \overline{1, n})$, for any n. Besides, for every k-th bit, i and $(i+T_k/2)$ the elements belonging to adjacent half-periods of this sequence are complementary

$$b_{ki} = \bar{b}_{k(i+Tk/2)},$$

i.e., associated with each other by the complementation operation $\bar{}$ *NOT*.

The mentioned properties are called a *dichotomic order* and a *dichotomic complement* of elements of a given sequence respectively.

Dichotomic generators within their repetition period $T_n = 2^n$, are unrepeated. Otherwise they can have a short transient interval (attractor), after which they phenomenally proceed to the stable state and then everywhere within the period T_n behave as unrepeated generators. *Dh*-generators can be multidimensional and parametric. They can form structural compositions of any complexity and can have an arbitrary large repetition period. The analysis shows that by using such generators, it is possible to reach the functional insolubility in essence from the direction of the high-order bits of the pseudorandom sequences being formed on their basis [3].

Dichotomic sequences being formed with the help of the above generators can be truncated beginning from the significant bits of the lowest order. Because of the truncation, these sequences lose the unrepeated properties.

In order to estimate the total repetition number, consider a binary sequence $B = \{b_k: k = \overline{1, L}\}$ of a length L, composed of elements $b_k \in \{0,1\}$. Let us divide this sequence into equal blocks s_i of a length n. Let the length be $L = n \cdot T_n$, where $T_n = 2^n$. Then, for every block $s_i = \{b_j: j = (i-1) \cdot n + k, i = \overline{1, T_n}, k = \overline{1, n}\}$

of the *n-digit binary sequence* $S_n = \{s_i: i = \overline{1, T_n}\}$, formed this way, an *n*-digit binary number $s_i \in [0, T_n - 1]$ is put in correspondence.

Assume that this sequence is one of the realizations of a random quantity ξ composed of n significant bits. Let us also distinguish *unrepeated* and *equirepeated* random quantities.

Random quantities ξ are called *unrepeated quantities* if all their realizations S_ξ do not contain equal elements within its period T_n, or if every their realizations starting with an element with the number $i_\xi \le h$ for some threshold value h, acquire unrepeated properties. These realizations are called the *unrepeated sequences*.

Random quantities which are not *unrepeated quantities* and for which the repetition numbers of any numbers $s_i \in [0, T_n - 1]$ of this sequence are equiprobable for their realization, are called *equirepeated quantities*. The realizations formed on their basis are called *equirepeated sequences*.

In order to estimate the total number of repetitions let us calculate the index

$$E_n = \left(\frac{T_n}{T_n - 1} \right)^{T_n}.$$

It is known that $\lim\limits_{T_n \to \infty} E_n = \mathrm{e}$, where $e = 2{,}718\ldots$ is the Hermite number.

The investigations of ρ-digit *D*-sequences S_ρ truncated by ε of their low-order significant bits, i.e., *truncated n*-digit dichotomic sequences $S_n \subset S_\rho$ $(n = \rho - \varepsilon)$, showed that the total average repetition number r_n of binary numbers in a given sequence S_n, when the sample length is 2^n, is equal to

$$r_n = \frac{2^n}{C_\varepsilon}, \qquad C_\varepsilon = \left(\frac{T_\varepsilon}{T_\varepsilon - 1} \right)^{T_\varepsilon}, \quad \text{where} \quad T_\varepsilon = 2^\varepsilon.$$

Expressing r_n through the probability of the total repetition number P_n at equiprobable arrangement T_n of different *n*-digit binary numbers among all T_n elements of the sequence S_n:

$$r_n = 2^n P_n,$$

we have:

$$P_n = \frac{1}{C_\varepsilon}.$$

Based on the experimental results and their approximation it is established that the analytic expression of the *empiric law* for distribution of the total repetition number $r_{(n-k)}$ has the form

$$r_{(n-k)} = 2^{n-k} P_n \prod_{i=0}^{k} 2^{-\frac{Q_i}{Q_i + Q_n}} \qquad (k = \overline{0, n-1}), \tag{1}$$

where the statistic sample $S_{(n-k)} \subseteq S_n$, is composed of 2^{n-k} *n*-digit binary numbers $s \in S_{(n-k)}$, the argument Q is given by the series $\{Q_i: i = \overline{0, k}\} = (0, 1, 2, 4, 6, \ldots p_k - 1\}$, for $p_o = 1$, resulting from the set of prime numbers p and the distribution constant Q_n, equal to

$$Q_n = \frac{1 - \frac{2}{T_\varepsilon}}{1 + E_n}.$$

For $r_{(n-k)} \le 1$, one can calculate the probability of repetition of the numbers $P_{(n-k)}$ in the statistic sample $S_{(n-k)}$ having the length 2^{n-k}

$$P_{(n-k)} = r_{(n-k)} / 2^{n-k}.$$

For large $k \ge 8$, one can estimate the probability of repetition of the numbers $P_{(n-k)}$: by the formula

$$P_{(n-k)} \approx 1/2^{k+1}.$$

In particular, from this relation it follows that the probability that two neighbor numbers of the sequence S_n coincide, i.e., $k = n - 1$, is equal to $1/2^n$.

Estimation of deviation values of the total repetition number, which are obtained using the empiric law $r_{(n-k)}$ and experimental data $r^*_{(n-k)}$ at the different lengths 2^{n-k} $(k = \overline{0, n-1})$, is performed on the

basis of mean ratio errors: $\Delta_{(n-k)} = \dfrac{r_{(n-k)} - R^*_{(n-k)}}{r_{(n-k)}} \cdot 100\%$.

Here, $R^*_{(n-k)}$ are calculated by using experimental data obtained with the help of averaging by m independent realizations $S_{i,(n-k)}$ $(i = \overline{1,m})$:

$$R^*_{(n-k)} = \frac{1}{m} \cdot \sum_{i=1}^{m} r^*_{i,(n-k)},$$

where $r^*_{i,(n-k)}$ are the results obtained in the corresponding realizations.

In order to perform a comparative analysis of experimental and empiric results, we used the dichotomic generator [3] truncated by ε low-order significant bits; the generator also has a 16-digit output $r_i = (B_{i-1} \oplus D_{i-1})/2^{\varepsilon} \bmod 2^{16}$, which is given by the self-synchronizing parametric $\{P,D,Q\}$ randomization operator $R(P_{i-1},D_{i-1},Q_{i-1})$: $B_{i-1} \to \{B_i,P_i,D_i,Q_i\}$, composed of the operators:

$$B_i = B_{i-1} \oplus P_{i-1}, \quad P_i = (4 \cdot B_{i-1}) \oplus \bar{D}_{i-1} \bmod 2^{\rho}, \quad Q_i = 2 \cdot (B_{i-1} \wedge P_{i-1}) \bmod 2^{\rho}, \qquad (2)$$

$$D_i = Q_{i-1} \oplus (4 \cdot D_{i-1}) \bmod 2^{\rho} \qquad (\rho = 16 + \varepsilon).$$

Besides, we performed a statistic analysis of the repetition-number distribution for sequences formed on the basis of the RC4 pseudorandom number generator and the GOST 28147-89 block cipher, which operates in the output feedback mode. The randomization Gamma generator presented below was also analyzed. For the analysis of such generators, assume that $C_{\varepsilon} = E_n$.

The calculations over 500 statistic samples, for a different number of truncated bits and different sizes of the statistic samples are presented in Table 1.

Table 1. Testing of repetition number

Statistical sample		65536	32768	16384	8192	4096	2048	1024	512	256	128
Truncation	**Estimates**										
1 bit	Empiric	**16384**	4096	1024	256	64	16	4	1	0.25	0.06
	Mean	16350	4043	1011	250	62	16	4	1	0.21	
Ratio Error	Δ (%)	0.209	1.288	1.313	2.190	3.069	-0.16	-5.65	-21.8	17.6	
2	Empiric	**20736**	5628	1470	376	95	24	6	2	0.38	0.10
	Mean	20708	5632	1469	375	95	24	6	2	0.35	0.03
	Δ (%)	0.136	-0.073	0.067	0.139	0.302	-1.27	-3.96	-7.69	9.17	
4	Empiric	**23336**	6657	1790	465	119	30	8	2	0.49	0.12
	Mean	23303	6657	1788	463	118	31	8	2	0.40	0.02
	Δ (%)	0.139	0.012	0.115	0.359	1.516	-0.97	-5.49	-2.58	17.4	
8	Empiric	**24062**	6961	1888	493	127	32	8	2	0.52	0.13
	Mean	24049	6954	1877	489	124	32	8	2	0.45	0.03
	Δ (%)	0.053	0.109	0.574	0.836	1.926	2.30	3.15	0.07	13.7	
16 bits	Empiric	**24109**	6981	1895	495	127	32	8	2	0.53	0.13
	Mean	24110	6984	1888	493	126	32	8	2	0.47	0.04
	Δ (%)	-0.003	-0.036	0.355	0.400	1.435	2.69	4.94	6.94	9.88	
Equirepeated Empiric Distribution Law		**24109**	**6981**	**1895**	**495**	**127**	**32**	**8**	**2**	**0.53**	**0.13**
RC4	Mean	24108	6981	1886	490	126	31	8	2	0.42	0.05
	Δ (%)	0.006	-0.001	0.465	0.901	1.282	4.15	5.65	3.86	20.5	
GOST	Mean	24113	6983	1885	492	125	31	8	2	0.42	0.04
	Δ (%)	-0.017	-0.026	0.522	0.563	1.526	3.15	5.18	2.81	19.8	
Randomization Generator	Mean	24106	6981	1885	490	124	32	8	2	0.46	0.06
	Δ (%)	0.013	-0.006	0.494	1.016	0.962	1.97	0.30	2.71	13.3	

From the presented table one can see that with increasing the number of truncated bits of the dichotomic sequence formed on the basis of equation (2) the repetition number distribution rapidly converges to the distribution inherent in the RC4 random number generator and the GOST block cipher. The small deviations show sufficiently high agreement of the experimental data and the empirical results. Taking

into account the high statistic characteristics of the sequences formed by such generators as well as the surjective character of the transformations used by them, note that these sequences are of a "perfect" equirepeated character. Below, the sequences having the repetition number law close to ideal empirical formula (1), will be called *perfect equirepeated sequences*.

It is possible to construct cryptographically reliable random-number generators and other stochastic devices and systems with given statistic properties using dichotomic generators that possess the functional insolubility in essence.

The solution of these problems is the subject of the section of stochastic cryptography, which includes the problems for creating statistically and functionally reliable generators of random numbers, one-way functions, block ciphers, and other stochastic operators. Further we consider methods for formation of unrepeated and equirepeated generators of random numbers, i.e., *Running Key generators* and *Gamma generators*. Both of them are called *randomization generators* [1]. The block diagram of an *N*-channel *m*-digit randomization generator, for which a generation platform length is $n = m \cdot N$ bits, is presented in FIG.1. The generator initialization is performed via setting its initial state in accordance with the key. Further, the initial state of the randomization generator, which realizes formation of an unrepeated keystream or equirepeated gamma, i.e., a *randomization sequence*, as well as the initial state of the dichotomic generator, which is used to generate keys, can vary depending on values of the used modifiers.

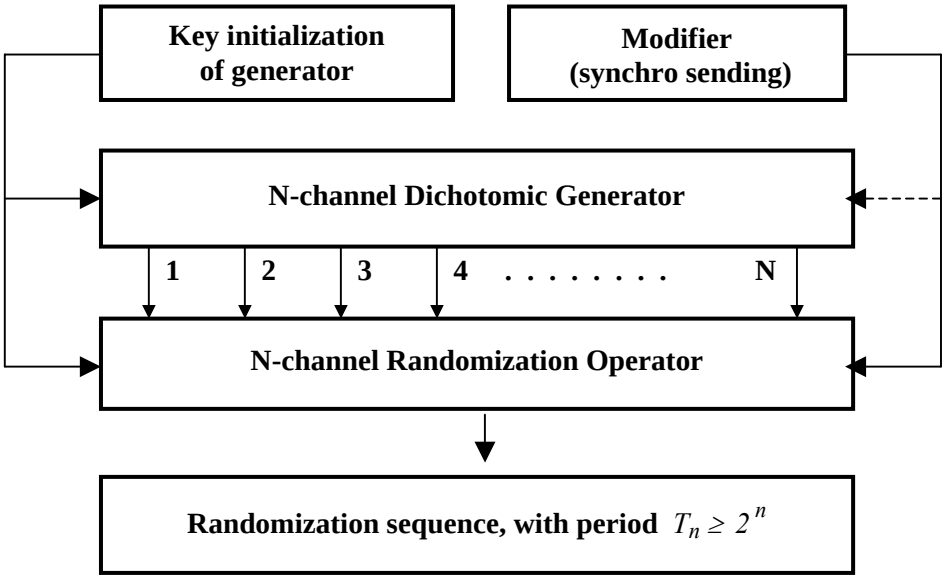

FIG.1

When analyzing randomization generators it is required to follow the general requirements imposed for (pseudo) random-number generators. Namely, these are the requirements to ensure the statistic and functional reliability for the extremely possible period of random sequences formed on their basis [4].

The first requirement is caused by the necessity to check the correspondence of statistic properties of these sequences to the "ideal" uniform distribution law. From the constructive point of view, note the *statistic reliability* of randomization generators, if the sequences formed by them satisfy the known statistic criteria inherent in the above-mentioned distribution law [5] and are defined on the whole set of permissible initial states of the generator.

The second requirement is caused by the necessity to provide the cryptographic strength of randomization generators and is achieved due to analytic insolubility and statistic indeterminacy of the generation equations. The latter conditions determine the *functional reliability* of the generator, namely, the possibility of determination or prediction of all n bits r_{kj} $(k = \overline{1,n})$ or their part $n_r \leq n$ for all known preceding r_{i-1} and following r_{i+1} elements of the sequence with probabilities $p_{kj} \leq p_k$ not exceeding the

guaranteed values p_k on the basis of any known fragments $\{r_0, r_1,\ldots r_i\}$ of the sequence, which are formed by the generator.

The third requirement being concerned with the period is fulfilled automatically due to the properties of dichotomic sequences, which are used as initial ones to form random-number sequences with the given distribution law.

One can use bijective or surjective transformations of dichotomic sequences in order to obtain statistically and functionally reliable random sequences. The transformations provide the dissipation of the high-order bit influence on low-order bits, the dissipation of the low-order bit influence on high-order ones, catenations and truncation of bits. Such sequences can be also obtained by using efficient methods for confusion of the bits transformed this way. Based on these transformations, one can obtain not only high statistic characteristics of sequences being formed, but also ensure the functional insolubility in essence due to reliable hiding of the dichotomic properties inherent in *D*-sequences.

The practical result is optimal if the problems of formation and transformation of *D*-sequences are cryptographically strong, mutually conditional and superadditive, irreversible in essence, supplement each other nearly everywhere and harmoniously associate with each other.

Taking the above-mentioned into account, consider particular realization versions of randomization generators, which are also called ***RA-generators,*** namely unrepeated Running Key generators and equirepeated Gamma generators.

Depending on applications, dichotomic generators are usually used as Running Key generators, since dichotomic generators are cryptographically strong and functionally insoluble in essence from the direction of high-order significant bits are usually used [3]. For example, one can attribute to such generators, *N*-channel *m*-digit three-parametric $\{P,D,G\}$ dichotomic generators given by the randomization operator $R_\Delta(P_{i-1},D_{i-1},G_{i-1})$: $B_{i-1} \to \{B_i,P_i,D_i,G_i\}$ of the form:

$$B_{ki} = B_{k(i-1)} \oplus P_{k(i-1)}, \quad P_{ki} = (4\cdot B_{k(i-1)}) \oplus \bar{D}_{k(i-1)} \bmod 2^m \quad (k=\overline{1,N}), \tag{3}$$

$$D_{ki} = G \oplus G_{k(i-1)}, \quad G_{ki} = (2\cdot(B_{k(i-1)} \wedge P_{k(i-1)})) \vee Q \bmod 2^m,$$

$$G = \mathbf{rot}_L(B_{k(i-1)},S_m), \quad Q = Q_{k(i-1)}, \quad Q_{ki} = (B_{k(i-1)} \wedge P_{k(i-1)})/\, 2^{m-1},$$

with the direct or indirect output, $r_{ki} = B_{k\{i-1)}$ or $r_{ki} = B_{k\{i-1)} \oplus D_{k\{i-1)}$, respectively, for $G|_{k=1} = G_0$ and $Q|_{k=1} = G_0 \bmod 2$. Here, $\mathbf{rot}_L$ is the cycle left-shift operation $\mathbf{rot}_L(B_{k\{i-1)},S_m)$ of the binary *m*-bit variable B_k by S_m bits, where S_m is the nearest prime for $m/3$ and aliquant with m.

Due to the properties inherent in such generators, high-order bits of a dichotomic sequence can be used as private keys and low-order bits can be used as public keys or identifiers. When the number of the low-order significant bits being used is $k_S \geq \rfloor \log_2 S_K \lfloor + 1$, where S_K is the maximal possible number of keystream elements being used, then the set of public identifiers has an unrepeated and non-complementary character. In other cases, in order to produce statistically and functionally reliable keystream sequences it is necessary to perform frequency equalization and hiding of frequency properties of bits of the *D*-sequences. These actions are performed by using randomization operators for forming keystream (FIG.1), which depend on the so-called *H-modifiers*. The mentioned operators should be bijective or injective if it is necessary to inherit unrepeated properties of the initial sequence that is transformed by them. For this purpose, values of *H*-modifiers are usually fixed and do not change during the generator iteration.

In order to perform efficient, statistically and functionally reliable transformations of dichotomic sequences, simple single or complex *r*-fold modifications or, what is the same, single- or *r*-round randomization operators can be used [1].

As practice shows [4], if the task is to form cryptographically strong generators, then these operators should satisfy the following structural requirements:

1. A catenation order of generator channels (block elements), which is prescribed by the operator, should ensure even and the quickest equalization of frequencies of bit variations in dichotomic sequences

used.

2. A catenation algorithm should ensure efficient confusion of bits within elements of the output generator block and essentially pronounced nonlinear transformation of channel bits from round to round in accordance with the above-established order.

3. Displacement of elements of the generator output block between rounds should ensure avalanche (exponential) increase from round to round in the dissipation speed of influence of high-order significant bits of the dichotomic sequences on low-order ones.

Let us consider the most typical examples of structural realization of these operators.

At the beginning let us consider the main types of orders used for channel catenation of randomization generators.

From the formal point of view the platform of an *m*-digit *N*-channel generator with the length $n = m \cdot N$ can be presented by the binary vector quantity $X = \{x_k: k = \overline{1, N}\}$, composed of *N, m*-digit binary vector components x_k. Then, by definition, the catenation order of the generator channels can be expressed through the binary order given by pairs $(i = I_k, j = J_k)$, defined on partially-ordered *ordinal* sets *{I,J}*. Under the dichotomic conditions, among binary orders the *laminar (N,N–1)(N–1,N–2)… (2,1)*, *contrlaminar (1,2)(2, 3) … (N–1,N)*, *tornadic (N,2)(2,N–1)… (⌋(N+1)/2⌊,1)* and *rotary (N,1) (N–1,2)… (⌊(N+3)/2⌋,⌊N/2⌋)* catenation order of vector components are of the most interest. A binary order can be represented by an associative vector $A = \{a_k: k = \overline{1, N}\}$ with elements $a_k \in [1,N]$, which point out the numbers of the corresponding binary components $(a_1, a_2)(a_2, a_3)\ldots(a_{N-1}, a_N)$ of *X*.

On the basis of the rotary and tornadic order it is sufficiently simple to realize the dissipation speed of influence of high-order significant bits of dichotomic sequences on the low-order ones, with the speed exponentially increasing from round to round. In the first case before every next round $r \le k_r$, except the first one, it is necessary to perform the cyclic right shift by $2^{r-2} \bmod N$ of binary vector components of the generator block. In the second case the cyclic right shift by $2^{r-2} \bmod (N-1)$ of all components, except the first binary vector component of the generator block, is performed.

In different cases a catenation order can be refined if it is justified from the point of view of the functional reliability and hardware topology of a generator.

Finally, there are many versions and methods to construct the algorithms of bit catenation of generator blocks. The operators formed on their basis must ensure efficient confusion and nonlinear catenation of the generator channels, where the catenation increases from round to round, as well as single- and multi-digit binary parallel processing.

Under the condition of the functional insolubility in essence beginning from the significant high-order bits of *D*-sequences used as initial ones for forming random-number sequences, which are sufficiently strong for cryptographic applications, the following three groups of operators allow solving the aforesaid problems simply and efficiently:

$$x_j = (z_j \oplus H_{rk}) \oplus ((g_j \lessdot\gtrdot 1) \vee p_i) \bmod 2^m, \quad g_j = z_j \wedge x_i, \quad p_i = g_i \lessdot\gtrdot (m-1) \bmod 2^m, \qquad (4.\text{I})$$

$$x_j = (z_j \oplus x_i) \oplus ((g_j \lessdot\gtrdot 1) \vee p_i) \oplus H_{rk} \bmod 2^m, \quad g_j = z_j \wedge x_i, \quad p_i = g_i \lessdot\gtrdot (m-1) \bmod 2^m, \qquad (4.\text{II})$$

$$x_j = (z_j \oplus x_i) \oplus ((g_j \lessdot\gtrdot 1) \vee p_i) \oplus \{H^{\bullet}_{rk}: h_i = 1, H^{\circ}_{rk}: h_i = 0\} \bmod 2^m, \qquad (4.\text{III})$$

$$h_i = f_h(x_i) \in \{0,1\}, \quad g_j = z_j \wedge x_i, \quad p_i = (g_i \oplus q_i) \lessdot\gtrdot (m-1) \bmod 2^m, \quad q_i = f_q(x_i) \quad (k = \overline{2, N}).$$

Here, $(i = a_{k-1}, j = a_k)$ is the ordered pair of the catenated components $z_j = \mathbf{rot}_L(x_j, S_m)$ at the left shift by the mentioned bit prime number S_m; $\lessdot$ and $\gtrdot$ is the left and right shift operation depending on the direction of bit catenation within the components; $\{h_i, q_i\}$ are the predicative variables of influence dissipation and concatenation of the components; $\{H_{rk}, H^{\bullet}_{rk}, H^{\circ}_{rk}\}$ are the binary *m*-digit modifiers of transformation of the components of the generator output block in rounds $r = \overline{1, k_r}$. The cases, in which the component $j = a_1$ called the top being the first in the associative series *A* is processed, are stipulated especially.

Assuming the above-presented operator group (4.I-III) as a basis and taking into account the

requirements of bijectivity of the transformations, which are necessary to keep the unrepeated properties inherent in *D*-sequences, the randomization keystream operators having the top of the associative list being in the channel *N* can have the following form:

$$x_N = (x_N \oplus H_{r1}) \oplus (g_N \succ 1), \quad g_N = x_N \wedge (x_N \succ 1), \tag{5.I}$$

$$x_j = (z_j \oplus H_{rk}) \oplus ((g_j \succ 1) \vee p_i), \quad g_j = z_j \wedge x_i, \quad p_i = g_i \prec (m-1) \bmod 2^m,$$

$$x_N = (z_N \oplus x_N) \oplus (g_N \succ 1) \oplus H_{r1}, \quad z_N = x_N \succ 1, \quad g_N = z_N \wedge x_N, \tag{5.II}$$

$$x_j = (z_j \oplus x_i) \oplus ((g_j \succ 1) \vee p_i) \oplus H_{rk}, \quad g_j = z_j \wedge x_i, \quad p_i = g_i \prec (m-1) \bmod 2^m,$$

$$x_N = (z_N \oplus x_N) \oplus (g_N \succ 1) \oplus \{H^{\bullet}_{r1}: h_N \neq 0, H^{o}_{r1}: h_N = 0\}, \tag{5.III}$$

$$z_N = x_N \succ 1, \quad g_N = z_N \wedge x_N, \quad h_N = x_N \wedge SIGN \quad (SIGN = 2^m - 1),$$

$$x_j = (z_j \oplus x_i) \oplus ((g_j \succ 1) \vee p_i) \oplus \{H^{\bullet}_{rk}: h_i \neq 0, H^{o}_{rk}: h_i = 0\}, \quad g_j = z_j \wedge x_i,$$

$$p_i = (g_i \oplus q_i) \prec (m-1) \bmod 2^m, \quad q_i = x_i \succ S_m, \quad h_i = x_i \wedge SIGN,$$

with the synchronization conditions $H^{\bullet}_{r1} \wedge SIGN = H^{o}_{r1} \wedge SIGN$ being fulfilled for all round modifiers $\{H^{\bullet}_{r1}, H^{o}_{r1}\}$ connected with the channel *N*. In order to attain optimal conditions for influence distribution of bits of the generator block it is necessary to assume $x_N = \mathbf{rot}_L(x_N, S_m)$ before every next round r, except the first one. Let us consider other typical peculiarities of the presented algorithms.

The algorithms are nonlinear due to the conjunctive component $g_j = z_j \wedge x_i$ being in their part. The last two have the essentially pronounced avalanche properties due to the entry x_i being part of the additive component $(z_j \oplus x_i)$ of equations (5.II) and (5.III), which is responsible for bit influence distribution between the channels. For the hardware realization, due to the binary algebraic structure the operational speed of such operators per one round can be comparable with the speed of execution of three operations *XOR* independently of the platform length of generators constructed on their basis. For the randomization operators of type (I) this speed is measured by two such operations.

As the analysis shows, the presented algorithms have different polynomial complexity $O(L^{\alpha})$ of the possibility to predict frequency properties of low-order significant bits of *D*-sequences, which are used as initial ones in order to obtain keystream sequences. Here, $L = n^{kr}$, and the index $\alpha = \{1,2,1+N\}$ for the corresponding algorithms (5.I-III). It is obvious that under the condition $L^{\alpha} > 2 \cdot S_K$, where S_K is the maximal possible number of used elements of a keystream, one can speak about the functional insolubility of equations (5).

The analysis will not be complete without estimation of statistic properties of the above generators. One can speak about the statistic reliability of the generators if sequences *Z* with a prescribed number *V* of bits formed on their basis and composed of (V/n) binary *n*-digit numbers z_s $(s = \overline{1,n})$, as well as sequences or *s*-slices $Z_s = \{z_{si}: i = \overline{1,V}\}$, composed of all $z_{si} \in \{0,1\}$ bits of their initial binary sequence *Z* with the number $(V \cdot n)$ of bits, all together and each taken separately satisfy the criteria of statistic tests [5]. Below, generators with the rotary and tornadic orders of channel catenation, which are functionally and statistically strong, are considered. The laminar order is not considered due to its drawbacks.

The test results show that generators of type I are not statistically reliable without a special previous round that provides distribution of high-order bit influence on low-order ones. In the last case for a generator of type I and in all other cases for generators of types II and III the statistic reliability is attained just after the first round, beginning with the 64-digit platform.

Two-round 4-channel 8-digit generators of types II and III with the 32-bit generation platform length are sufficiently reliable from the statistic point of view, due to the avalanche effects inherent in them. Table 2 in the Appendix presents a pseudo C realization version of a generator of type III. The detailed analysis is given for the functional dependence of keystream bits on bits of its initial *D*-sequence after termination of the first round and at the beginning of channel catenation during the second round of the transformations for the rotary and tornadic orders of channel catenation, respectively. In the sense of the functional and statistic reliability of a sequence formed, the latter is more preferable and that is indirectly supported by the results of

the statistic tests. The estimates of ω characterizing the minimal period 2^{ω} for slices of the resultant sequence are also presented.

For all above-considered types of keystream generators, no weak keys resulting in appreciable degeneration of sequences formed on their basis are revealed.

Consider now the next class of random-number generators known as Gamma generators. Unlike unrepeated Running Key generators, sequences of random numbers formed by using Gamma generators have the aforementioned equirepeated character (1) inherent in the sequences formed by RC4 generators and block ciphers operating in the output feedback mode. As shown above, such sequences can be obtained on the basis of truncated *Dh*-generators. Besides, equirepeated sequences can be obtained by means of diffusion and confusion of bits of the *D*-sequences, which are original for them. These operations can be realized by using the randomization operators for gamma formation (FIG.1), i.e., the randomizers, depending on the set of *P-modifiers*. In order to impart the necessary equirepeated properties to their original *D*-sequences, the output of the mentioned operators should be surjective. Values of the *P*-modifiers can be fixed or varied during the generator iteration by a well-defined or non-determined law.

By the construction ideology and structure randomizers are similar to *Dh*-generators and the randomizers can have a parametric and multidimensional character and can be counted on parallel single- and multi-digit processorless processing [3] as well as *Dh*-generators. Due to the surjective, parametric and multidimensional character of such operators they can be functionally insoluble in essence independently of functional and statistic properties of *D*-sequences being initial for them.

Randomizers, as well as the above-considered operators for formation of a keystream, belong to the class of one-way operators. Description of the general principles for construction of those operators is a special topic, which will be analyzed in further publications. Here, not going into details of the realization, let us briefly consider as an example, *n*-digit four-parametric $\{P,D,Q,Z\}$ randomizer $R(P_{i-1},D_{i-1},Q_{i-1},Z_{i-1},G_{i-1})$: $X_i \to \{G_i,P_i,D_i,Q_i\}$ with an input $x \in \{X_i\}$ and an direct or indirect output, $g = G_{i-1}$ or $g = G_{i-1} \oplus Q_{i-1}$ respectively, where $g \in \{G_i\}$ and the operators composing the randomizer have the form:

$$G_i = Z_{i-1} \oplus Q_{i-1}, \quad Q_i = D_{i-1} \oplus P_{i-1}, \quad D_i = G_{i-1} \oplus (G_{i-1} \lessdot d) \bmod 2^n, \qquad (6)$$
$$Z_i = \mathbf{rot}_L(G_{i-1} \oplus X_i, S_n), \quad P_i = ((Z_{i-1} \wedge D_{i-1}) \lessdot 1) \vee H_D \bmod 2^n,$$

and S_n is a constant equal to the nearest prime number for $n/3$ and aliquant with n, as well as d is a constant equal to the nearest integer for $\log_2(n/3)$ and $H_D = \{1: d = 1, 0: d > 1\}$. Equations (6) are analytically soluble with respect to their arguments, if randomizers with a direct output are used and if the used input sequence $\{X_i\}$ is known or easily predicted with respect to the bits forming it. If the indirect output is used, then the randomization variables $\{G,P,D,Q,Z\}$ characterizing the internal state of the randomizer can be expressed through its known output state, which is determined by the randomization variable *g*, in the following form:

$$G_i = \mathbf{g_i} \oplus G_{i-1} \oplus \mathbf{rot}_L(G_{i-2} \oplus \mathbf{x_{i-1}}, S_n), \quad Q_i = \mathbf{g_{i+1}} \oplus G_i, \quad D_i = G_{i-1} \oplus (2^d \cdot G_{i-1}) \bmod 2^n,$$
$$Z_i = \mathbf{rot}_L(G_{i-1} \oplus \mathbf{x_i}, S_n), \quad P_i = (2 \cdot (\mathbf{rot}_L(G_{i-2} \oplus \mathbf{x_{i-1}}, S_n) \wedge (G_{i-2} \oplus (2^d \cdot G_{i-2})))) \vee H_D \bmod 2^n.$$

From the presented expressions it follows that equations (6) are analytically insoluble, since for any iteration *i*, the law of variation of variables characterizing the internal state of the randomizer cannot be unambiguously expressed even through all its known external states. In other words, if the arguments of equations (6) are unknown and hidden for the external admission, then by all their known external states it is impossible to calculate analytically one of the following values and each of previous ones of their output variables. Taking into account the avalanche properties and essentially pronounced nonlinear character of the presented randomizers, we can assume that the resulted effect of using analytically insoluble *D*-sequences as original ones will be superadditive in character. In these cases, dichotomic or *Dh-counters* [3] can be used, which are given by a two-parametric $\{P,D\}$ randomization operator $R_C(H,P_{i-1},D_{i-1})$: $X_{i-1} \to \{X_i,P_i,D_i\}$ that is the most simple in realization and for the odd fixed value of the modifier *H* and the direct output *X* with the operator having the form:

$$X_i = X_{i-1} \oplus P_{i-1}, \quad P_i = H \oplus D_{i-1}, \quad D_i = (X_{i-1} \wedge P_{i-1}) \lessdot 1 \bmod 2^n. \qquad (7)$$

The statistic analysis of the randomization gamma generators [5] constructed by using operator compositions (7) and (6) shows a sufficiently high statistic reliability of sequences formed on their basis. Thus, for all n, beginning from 12, any noticeable bit correlations were not revealed both in majority and in part of slices under any initial conditions. Note that the period increases by more than 2 times, the sequences being equirepeated. That is confirmed by the results of the statistic tests (see Table 1).

Thus, taking into account analytic insolubility of the equations describing the operation of the represented randomizers and performed statistic analysis, in fact one can speak about a possibility of reaching the functional insolubility of all significant bits of random-number sequences formed on their basis. Thus, $\rfloor \log_2(n) \lfloor$ idle iterations of the generator are enough to obtain the stable statistics under any initial conditions of the generation.

The considered Gamma generators are strictly aimed at using single-digit computing devices or devices having a register digit capacity not smaller than the generation platform length, which is not enough for the overwhelming majority of applications. The use of multi-channel randomizers can eliminate the mentioned restrictions.

Multi-channel Gamma generators are constructed by using single-channel generators, with the help of channel coupling by the criterion for result formed at the channel interface and distribution of influence of the low-channel bits on the high-channel bits. As an example, consider an N-channel m-digit 5-parametric $\{W,P,D,Q,Z\}$ randomizer $R_G(W_{i-1},P_{i-1},D_{i-1},Q_{i-1},Z_{i-1},G_{i-1})\colon X_i \to \{G_i,W_i,P_i,D_i,Q_i\}$, with the input $x \in \{X_{ik}\}$ and the direct or indirect outputs, $g = G_{i-1}$ or $g = W_{i-1} \oplus Q_{i-1}$ respectively, where $g \in \{G_{ik}\}$. The randomizer is composed of the operators

$$G_{ki} = Z_{k(i-1)} \oplus Q_{k(i-1)}, \quad Q_{ki} = D_{k(i-1)} \oplus P_{k(i-1)}, \quad D_{ki} = W_{k(i-1)} \oplus (W_{k(i-1)} \lessdot d) \bmod 2^m, \qquad (8)$$
$$W_{ki} = G_{k(i-1)} \oplus w_k, \quad w_{k+1} = G_{k(i-1)} \oplus X_i, \quad Z_{ki} = \mathbf{rot}_L(W_{k(i-1)} \oplus X_i, S_m),$$
$$p = Z_{k(i-1)} \wedge D_{\overline{k(i-1)}}, \quad P_{ki} = (p \lessdot 1) \vee q_k \bmod 2^m, \quad q_{k+1} = p \gtrdot (m-1),$$

with respect to every channel $k = \overline{1, N}$, with the above-mentioned constants S_m and d as well as the criterion for result q_k, where $q_1 = \{1\colon d = 1,\ 0\colon d > 1\}$ before every i-th iteration of the generator. The randomization variable w, which is responsible for distribution of influence of the low-channel bits on the high-channel bits, is initially taken equal to $w_1 = W_0$, and after termination of every next iteration it is taken equal to: $w_1 = \mathbf{rot}_L(w_{N+1},S_m)$. Then, the analytic insolubility of equations (8) immediately follows from the analytic insolubility of equations (6).

One can use N-channel m-digit Dh-counters as reference generators in order to form equirepeated sequences. The counters are given by two-parametric $\{P,D\}$ randomization operator $R_C(H,P_{i-1},D_{i-1})\colon X_{i-1} \to \{X_i,P_i,D_i\}$, of the form:

$$X_{ki} = X_{k(i-1)} \oplus P_{k(i-1)}, \quad P_{ki} = H_k \oplus D_{k(i-1)}, \quad p = X_{k(i-1)} \wedge P_{k(i-1)}, \qquad (9)$$
$$D_{ki} = (p \lessdot 1) \vee q_k \bmod 2^n, \quad q_{k+1} = p \gtrdot (m-1),$$

where $H = \{H_k\}$ are arbitrary constants for every channel $k = \overline{1, N}$ and q_k is the result criterion, and $q_1 \equiv H_1 \oplus 1 \ (\bmod 2)$ before next i-th iteration of the generator.

The statistic analysis of the 8-digit randomization Gamma generators, as well as the generators of the higher digit capacity, formed on the basis of the N-channel operators shows that by all the indices and under all initial conditions there is a sufficiently high statistic reliability of sequences formed by using them, beginning from the 16-bit minimal generation platform length and up to the nearest foreseeable 16384-bit. Thus, $N \cdot \log_2(m)$ idle iterations of the generator are enough to obtain the stable statistics under any initial conditions of the generation.

From all the above-stated material, one can assume that there is a possibility to reach the essential functional insolubility of all significant bits of the equirepeated random-number sequences, which are formed by using the multi-channel Gamma generators.

The represented Gamma generators perform highly efficient parallel and sequential processing. In the first case, the calculations of the critical path length show that the operation speed of the everywhere parallel Gamma generators can be comparable with the speed of performance of one *XOR* operation independently of the length of the generator platform. In the second case, due to sequential multi-beat processing with the beat number equal to the bit number in the platform, the number of elements required for realization of the main network of the generator is minimal and equal to 5-8. This circumstance allows to lower substantially the energy consumption and the production cost of similar implementation schemes.

Conclusions.

1. The implementation schemes of ***RA-***generators are of simple design, have high statistic reliability, permit parallel single- and multi-digit processing, including processorless one, on any platforms of computing devices as well as require a small memory.

2. The software versions of the algorithms of ***RA-***generator realizations exceed many times the most perfect specimens. For example, the productivity of Gamma generators is 4 times greater than the productivity of RC4 random-number generators. The productivity of single- and two-round generators of a 128-digit keystream is respectively 5-9 and 3-5 times greater than the productivity of RC6 block ciphers during realizations in DOS in the programming language C. The hardware versions of ***RA-***generators are distinguished by a small manufacturing cost and maximum high processing speed. So, for a hardware realization the Gamma generator speed is comparable with the speed of one *XOR* operation, and the Running Key generator speed is comparable with 1-3, sometimes up to 4-6 speed of those operations, independently of the platform length of the mentioned generators.

3. Implementation schemes are simple for analysis, are of a high dynamic parametric, hidden from environment, multidimensional, essentially pronounced nonlinear and non-determined character. They are also counted on using keys with a variable and very large length. As the analysis shows, such realizations are able to have the functional reliability and cryptographic resistance, which are sufficiently high and necessary for practical applications.

4. ***RA-***generators can have any repetition period that is not smaller than one given in advance. By using dichotomic generators and systems they can form structural compositions of any complexity, can be insoluble in essence and at the same time keep the stochastic properties inherent in their elements.

5. ***RA-***generators lay the foundation for development of stochastic and network symmetric cryptography, as well as for technologies for construction of distributed multidimensional, parametric, holistic, structurally complicated cryptographic devices and systems of the new generation, which are intended for network solution of any complexity, super-high-speed and highly reliable processing.

6. ***RA-***generators and technologies represented by them ensure high-level unification and standardization due to universal and homogeneous character of cryptographic units and components used.

7. The presented technologies keep succession and remain unchanged in time due to fundamental character of the theoretical basis and the conceptual connection of them with processes and phenomena inherent in real dynamic systems. Processorless highly productive parallel and highly profitable sequential methods of processing, which laid the basis of the technology, allow speaking about highly efficient and energy-consumption computations. These peculiarities are especially actual at the present stage in the field of development of nano-, holographic, quantum and radio-frequency cryptographic processing.

«RANDOMIZATION GENERATORS»,

Article based on the materials submitted at the Fast Software Encryption (FSE 2004),
February 5-7, 2004, New Delhi, India.

Keywords. Stochastic system. Symmetric cryptography. Random-number generator. Dichotomic generator. Randomization method. Randomization operator. Randomization generator. Key generator. Gamma generator. One-way function.

References

1. I. A. Kulakov "A Method of the Randomization Properties Imparting to a Real Object and a Randomization System" Application for the International Patent.
2. I. A. Kulakov Dichotomic sequences and their properties, Article based on the materials submitted at the 3rd Central-European Conference in Bratislava, TATRACRYPT 2003, June 26-28, 2003, Slovakia.
3. I. A. Kulakov Dichotomic generators and their properties, Article based on the materials submitted at the 6th International Conference of Information Security and Cryptology (ICISC 2003), November 27-28, Seoul, Korea.
4. B. Schneier APPLIED CRYPTOGRAPHY. Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc, 1996.
5. G. Marsaglia DIEHARD Statistical Test Package, 1997, geo@stat.fsu.edu

Appendix, Table 2. FUNCTIONAL ANALYSIS OF BIT CATENATION OPERATORS IN THE GENERATOR BLOCK

byte	bit	Rotary Catenation Order of the generator channels: First round	2-th round before catenation		ω period $\geq 2^{\omega}$		Tornadic Catenation Order of the generator channels: First round	2-th round before catenation		ω period $\geq 2^{\omega}$
	32	$02 \oplus 29 \oplus (03 \wedge 30) \oplus 32^{*}_{29}$	0	IV	$\mathbf{29}\ldots32^{*}_{29}, 32^{\circ\circ}_{08}$		$10 \oplus 29 \oplus (11 \wedge 30) \oplus 32^{*}_{21}$	0	IV	$\mathbf{29}\ldots32^{*}_{29}, 32^{\circ\circ}_{08}$
	31	$01 \oplus 28 \oplus (02 \wedge 29) \oplus 32^{*}_{28}$	$02 \oplus 29 \oplus (03 \wedge 30) \oplus 32^{*}_{29}$		$\mathbf{28}\ldots32^{*}_{28,29}, 32^{**}_{07}$		$09 \oplus 28 \oplus (10 \wedge 29) \oplus 32^{*}_{20}$	$10 \oplus 29 \oplus (11 \wedge 30) \oplus 32^{*}_{21}$		$\mathbf{28}\ldots32^{*}_{28,29}, 32^{**}_{07}$
	30	$08 \oplus 27 \oplus (01 \wedge 28) \oplus 32^{*}_{27}$	$01 \oplus 28 \oplus (02 \wedge 29) \oplus 32^{*}_{28}$		$\mathbf{27}\ldots32^{*}_{27,28}, 32^{**}_{06}$		$16 \oplus 27 \oplus (09 \wedge 28) \oplus 32^{*}_{19}$	$09 \oplus 28 \oplus (10 \wedge 29) \oplus 32^{*}_{20}$		$\mathbf{27}\ldots32^{*}_{27,28}, 32^{**}_{06}$
	29	$07 \oplus 26 \oplus (08 \wedge 27) \oplus 32^{*}_{26}$	$08 \oplus 27 \oplus (01 \wedge 28) \oplus 32^{*}_{27}$		$\mathbf{26}\ldots32^{*}_{26,27}, 32^{**}_{05}$		$15 \oplus 26 \oplus (16 \wedge 27) \oplus 32^{*}_{18}$	$16 \oplus 27 \oplus (09 \wedge 28) \oplus 32^{*}_{19}$		$\mathbf{26}\ldots32^{*}_{26,27}, 32^{**}_{05}$
	28	$06 \oplus 25 \oplus (07 \wedge 26) \oplus 32^{*}_{25}$	$07 \oplus 26 \oplus (08 \wedge 27) \oplus 32^{*}_{26}$		$\mathbf{25}\ldots32^{*}_{25,26}, 32^{**}_{04}$		$14 \oplus 25 \oplus (15 \wedge 26) \oplus 32^{*}_{17}$	$15 \oplus 26 \oplus (16 \wedge 27) \oplus 32^{*}_{18}$		$\mathbf{25}\ldots32^{*}_{25,26}, 32^{**}_{04}$
	27	$05 \oplus 32 \oplus (25 \wedge 26) \oplus 32^{*}_{32} \oplus 28$	$06 \oplus 25 \oplus (07 \wedge 26) \oplus 32^{*}_{25}$		$\mathbf{28}\ldots32^{*}_{25,32}, 32^{**}_{03}$		$13 \oplus 32 \oplus (25 \wedge 26) \oplus 32^{*}_{24} \oplus 28$	$14 \oplus 25 \oplus (15 \wedge 26) \oplus 32^{*}_{17}$		$\mathbf{28}\ldots32^{*}_{25,32}, 32^{**}_{03}$
	26	$04 \oplus 31 \oplus (05 \wedge 32) \oplus 32^{*}_{31}$	$05 \oplus 32 \oplus (25 \wedge 26) \oplus 32^{*}_{32} \oplus 28$		$\mathbf{31}\ldots32^{*}_{31,32}, 32^{**}_{02}$		$12 \oplus 31 \oplus (13 \wedge 32) \oplus 32^{*}_{23}$	$13 \oplus 32 \oplus (25 \wedge 26) \oplus 32^{*}_{24} \oplus 28$		$\mathbf{31}\ldots32^{*}_{31,32}, 32^{**}_{02}$
	25	$03 \oplus 30 \oplus (04 \wedge 31) \oplus 32^{*}_{30}$	$04 \oplus 31 \oplus (05 \wedge 32) \oplus 32^{*}_{31}$		$\mathbf{30}\ldots32^{*}_{30,31}, 32^{**}_{01}$		$11 \oplus 30 \oplus (12 \wedge 31) \oplus 32^{*}_{32}$	$12 \oplus 31 \oplus (13 \wedge 32) \oplus 32^{*}_{23}$		$\mathbf{30}\ldots32^{*}_{30,31}, 32^{**}_{01}$
III	24	$32 \oplus 32^{\circ}_{08}$	$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$	I	$\mathbf{29}\ldots32^{*}_{05}, 8^{**}_{16}$		$32 \oplus 32^{\circ}_{08}$	$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$	II	$\mathbf{29}\ldots32^{*}_{05}, 16^{**}_{16}$
	23	$31 \oplus 32 \oplus 32^{*}_{07}$	$28 \oplus (31 \wedge \neg 30) \oplus 32^{*}_{04}$		$\mathbf{28}\ldots32^{*}_{04}, 8^{**}_{15}$		$31 \oplus 32 \oplus 32^{*}_{07}$	$28 \oplus (31 \wedge \neg 30) \oplus 32^{*}_{04}$		$\mathbf{28}\ldots32^{*}_{04}, 16^{**}_{15}$
	22	$30 \oplus (31 \wedge \neg 32) \oplus 32^{*}_{06}$	$27 \oplus (31 \wedge \neg 29) \oplus 32^{*}_{03}$		$\mathbf{27}\ldots32^{*}_{03}, 8^{**}_{14}$		$30 \oplus (31 \wedge \neg 32) \oplus 32^{*}_{06}$	$27 \oplus (31 \wedge \neg 29) \oplus 32^{*}_{03}$		$\mathbf{27}\ldots32^{*}_{03}, 16^{**}_{14}$
	21	$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$	$26 \oplus (31 \wedge \neg 28) \oplus 32^{*}_{02}$		$\mathbf{26}\ldots32^{*}_{02}, 8^{**}_{13}$		$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$	$26 \oplus (31 \wedge \neg 28) \oplus 32^{*}_{02}$		$\mathbf{26}\ldots32^{*}_{02}, 16^{**}_{13}$
	20	$28 \oplus (31 \wedge \neg 30) \oplus 32^{*}_{04}$	$25 \oplus (31 \wedge \neg 27) \oplus 32^{*}_{01}$		$\mathbf{25}\ldots32^{*}_{01}, 8^{**}_{12}$		$28 \oplus (31 \wedge \neg 30) \oplus 32^{*}_{04}$	$25 \oplus (31 \wedge \neg 27) \oplus 32^{*}_{01}$		$\mathbf{25}\ldots32^{*}_{01}, 16^{**}_{12}$
	19	$27 \oplus (31 \wedge \neg 29) \oplus 32^{*}_{03}$	$32 \oplus 32^{\circ}_{08}$		$\mathbf{32}\ldots32^{\circ}_{08}, 8^{**}_{11}$		$27 \oplus (31 \wedge \neg 29) \oplus 32^{*}_{03}$	$32 \oplus 32^{\circ}_{08}$		$\mathbf{32}\ldots32^{\circ}_{08}, 16^{**}_{11}$
	18	$26 \oplus (31 \wedge \neg 28) \oplus 32^{*}_{02}$	$31 \oplus 32 \oplus 32^{*}_{07}$		$\mathbf{31},32\ldots32^{*}_{07}, 8^{**}_{10}$		$26 \oplus (31 \wedge \neg 28) \oplus 32^{*}_{02}$	$31 \oplus 32 \oplus 32^{*}_{07}$		$\mathbf{31},32\ldots32^{*}_{07}, 16^{**}_{10}$
	17	$25 \oplus (31 \wedge \neg 27) \oplus 32^{*}_{01}$	$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$		$\mathbf{29}\ldots32^{*}_{05}, 8^{**}_{09}$		$25 \oplus (31 \wedge \neg 27) \oplus 32^{*}_{01}$	$29 \oplus (31 \wedge \neg 31) \oplus 32^{*}_{05}$		$\mathbf{29}\ldots32^{*}_{05}, 16^{**}_{09}$
II	16	$21 \oplus 08 \oplus (06 \wedge 25) \oplus 08^{*}_{16} \oplus 04$	$18 \oplus 05 \oplus (19 \wedge 06) \oplus 08^{*}_{13}$	III	$\mathbf{28}\ldots24^{**}_{24}$		$21 \oplus 16 \oplus (14 \wedge 25) \oplus 16^{*}_{16} \oplus 12$	$18 \oplus 13 \oplus (19 \wedge 14) \oplus 16^{*}_{13}$	IV	$\mathbf{25}\ldots32^{**}_{32}$
	15	$20 \oplus 07 \oplus (21 \wedge 08) \oplus 08^{*}_{15}$	$17 \oplus 04 \oplus (18 \wedge 05) \oplus 08^{*}_{12}$		$\mathbf{31}\ldots24^{**}_{23}$		$20 \oplus 15 \oplus (21 \wedge 16) \oplus 16^{*}_{15}$	$17 \oplus 12 \oplus (18 \wedge 13) \oplus 16^{*}_{12}$		$\mathbf{28}\ldots32^{**}_{31}$
	14	$19 \oplus 06 \oplus (20 \wedge 07) \oplus 08^{*}_{14}$	$24 \oplus 03 \oplus (17 \wedge 04) \oplus 08^{*}_{11}$		$\mathbf{30}\ldots24^{**}_{22}$		$19 \oplus 14 \oplus (20 \wedge 15) \oplus 16^{*}_{14}$	$24 \oplus 11 \oplus (17 \wedge 12) \oplus 16^{*}_{11}$		$\mathbf{27}\ldots32^{**}_{30}$
	13	$18 \oplus 05 \oplus (19 \wedge 06) \oplus 08^{*}_{13}$	$23 \oplus 02 \oplus (24 \wedge 03) \oplus 08^{*}_{10}$		$\mathbf{29}\ldots24^{**}_{21}$		$18 \oplus 13 \oplus (19 \wedge 14) \oplus 16^{*}_{13}$	$23 \oplus 10 \oplus (24 \wedge 11) \oplus 16^{*}_{10}$		$\mathbf{26}\ldots32^{**}_{29}$
	12	$17 \oplus 04 \oplus (18 \wedge 05) \oplus 08^{*}_{12}$	$22 \oplus 01 \oplus (23 \wedge 02) \oplus 08^{*}_{09}$		$\mathbf{28}\ldots24^{**}_{20}$		$17 \oplus 12 \oplus (18 \wedge 13) \oplus 16^{*}_{12}$	$22 \oplus 09 \oplus (23 \wedge 10) \oplus 16^{*}_{09}$		$\mathbf{25}\ldots32^{**}_{28}$
	11	$24 \oplus 03 \oplus (17 \wedge 04) \oplus 08^{*}_{11}$	$21 \oplus 08 \oplus (06 \wedge 25) \oplus 08^{*}_{16} \oplus 04$		$\mathbf{27}\ldots24^{**}_{19}$		$24 \oplus 11 \oplus (17 \wedge 12) \oplus 16^{*}_{11}$	$21 \oplus 16 \oplus (14 \wedge 25) \oplus 16^{*}_{16} \oplus 12$		$\mathbf{28}\ldots32^{**}_{27}$
	10	$23 \oplus 02 \oplus (24 \wedge 03) \oplus 08^{*}_{10}$	$20 \oplus 07 \oplus (21 \wedge 08) \oplus 08^{*}_{15}$		$\mathbf{26}\ldots24^{**}_{18}$		$23 \oplus 10 \oplus (24 \wedge 11) \oplus 16^{*}_{10}$	$20 \oplus 15 \oplus (21 \wedge 16) \oplus 16^{*}_{15}$		$\mathbf{31}\ldots32^{**}_{26}$
	9	$22 \oplus 01 \oplus (23 \wedge 02) \oplus 08^{*}_{09}$	$19 \oplus 06 \oplus (20 \wedge 07) \oplus 08^{*}_{14}$		$\mathbf{25}\ldots24^{**}_{17}$		$22 \oplus 09 \oplus (23 \wedge 10) \oplus 16^{*}_{09}$	$19 \oplus 14 \oplus (20 \wedge 15) \oplus 16^{*}_{14}$		$\mathbf{30}\ldots32^{**}_{25}$
I	8	$13 \oplus 24 \oplus (22 \wedge 01) \oplus 24^{*}_{24} \oplus 20$	$10 \oplus 21 \oplus (11 \wedge 22) \oplus 24^{*}_{21}$	IV	$\mathbf{25}\ldots32^{**}_{32}$		$05 \oplus 24 \oplus (17 \wedge 22) \oplus 24^{*}_{32} \oplus 20$	$02 \oplus 21 \oplus (03 \wedge 22) \oplus 24^{*}_{29}$	III	$\mathbf{28}\ldots24^{**}_{32}$
	7	$12 \oplus 23 \oplus (13 \wedge 24) \oplus 24^{*}_{23}$	$09 \oplus 20 \oplus (10 \wedge 21) \oplus 24^{*}_{20}$		$\mathbf{28}\ldots32^{**}_{31}$		$04 \oplus 23 \oplus (05 \wedge 24) \oplus 24^{*}_{31}$	$01 \oplus 20 \oplus (02 \wedge 21) \oplus 24^{*}_{28}$		$\mathbf{31}\ldots24^{**}_{31}$
	6	$11 \oplus 22 \oplus (12 \wedge 23) \oplus 24^{*}_{22}$	$16 \oplus 19 \oplus (09 \wedge 20) \oplus 24^{*}_{19}$		$\mathbf{27}\ldots32^{**}_{30}$		$03 \oplus 22 \oplus (04 \wedge 23) \oplus 24^{*}_{30}$	$08 \oplus 19 \oplus (01 \wedge 20) \oplus 24^{*}_{27}$		$\mathbf{30}\ldots24^{**}_{30}$
	5	$10 \oplus 21 \oplus (11 \wedge 22) \oplus 24^{*}_{21}$	$15 \oplus 18 \oplus (16 \wedge 19) \oplus 24^{*}_{18}$		$\mathbf{26}\ldots32^{**}_{29}$		$02 \oplus 21 \oplus (03 \wedge 22) \oplus 24^{*}_{29}$	$07 \oplus 18 \oplus (08 \wedge 19) \oplus 24^{*}_{26}$		$\mathbf{29}\ldots24^{**}_{29}$
	4	$09 \oplus 20 \oplus (10 \wedge 21) \oplus 24^{*}_{20}$	$14 \oplus 17 \oplus (15 \wedge 18) \oplus 24^{*}_{17}$		$\mathbf{25}\ldots32^{**}_{28}$		$01 \oplus 20 \oplus (02 \wedge 21) \oplus 24^{*}_{28}$	$06 \oplus 17 \oplus (07 \wedge 18) \oplus 24^{*}_{25}$		$\mathbf{28}\ldots24^{**}_{28}$
	3	$16 \oplus 19 \oplus (09 \wedge 20) \oplus 24^{*}_{19}$	$13 \oplus 24 \oplus (22 \wedge 01) \oplus 24^{*}_{24} \oplus 20$		$\mathbf{28}\ldots32^{**}_{27}$		$08 \oplus 19 \oplus (01 \wedge 20) \oplus 24^{*}_{27}$	$05 \oplus 24 \oplus (17 \wedge 22) \oplus 24^{*}_{32} \oplus 20$		$\mathbf{37}\ldots24^{**}_{27}$
	2	$15 \oplus 18 \oplus (16 \wedge 19) \oplus 24^{*}_{18}$	$12 \oplus 23 \oplus (13 \wedge 24) \oplus 24^{*}_{23}$		$\mathbf{31}\ldots32^{**}_{26}$		$07 \oplus 18 \oplus (08 \wedge 19) \oplus 24^{*}_{26}$	$04 \oplus 23 \oplus (05 \wedge 24) \oplus 24^{*}_{31}$		$\mathbf{26}\ldots24^{**}_{26}$
	1	$14 \oplus 17 \oplus (15 \wedge 18) \oplus 24^{*}_{17}$	$11 \oplus 22 \oplus (12 \wedge 23) \oplus 24^{*}_{22}$		$\mathbf{30}\ldots32^{**}_{25}$		$06 \oplus 17 \oplus (07 \wedge 18) \oplus 24^{*}_{25}$	$03 \oplus 22 \oplus (04 \wedge 23) \oplus 24^{*}_{30}$		$\mathbf{25}\ldots24^{**}_{25}$

$\oplus$, $\wedge$, $\neg$ are the bitwise operations XOR, AND, NOT connecting the bits $k = \overline{1,n}$ of the platform of N-channel generator, the length $n = N \cdot m$. Here, m is the digit capacity of the generation platform registers. $\succ, \prec$ are the right and left bit shifts respectively. **SIGN** $= 2^{m-1}$
AssBlock is the associative vector, which established the order of the channel catenation.
KeyBlock is the block of the generator. $\mathbf{H}^{\bullet}$ and $\mathbf{H}^{o}$ are the vectors of block modification (***** in the first round, ****** in the second round). RotL is the cyclic left by **SHFL** bits

```
x = z = *AssBlock[1];  z ≻= 1;  g = z ∧ x;  // bit catenation round of the generator block
*AssBlock[1] = ((z ⊕ x) ⊕ ((x ∧ SIGN) ? H•[1] : Ho[1])) ⊕ (g ≻ 1);
for ( i = 2;  i ≤ N;  x = q,  i++) {        // bit catenation procedure of the generator block
   p = g ⊕ (x ≻ SHFL);  rb = AssBlock[i];  q = *rb;  z = RotL(q,SHFL);  g = z ∧ x;
  *rb = ((z ⊕ x) ⊕ ((x ∧ SIGN) ? H•[i] : Ho[i])) ⊕ ((g ≻ 1) ⊕ (p ≺ (m – 1));
} RotateBlock(KeyBlock,ShfR);      // displacement block elements interround
```