DICHOTOMIC SEQUENCES AND THEIR PROPERTIES

Igor A. Kulakov

Random Art Labs Limited

chief@random-art.com

The article lays the theoretical foundations for the formation of stochastic systems. The fundamental properties inherent in the binary representation of natural sequence are considered. The generalization of properties of one natural sequence onto another, i.e., the so-called dichotomic sequence, is presented. Dichotomic sequences and their properties are analyzed. The examples of algorithms and their generation are presented. Multidimensional, parametric and essentially pronounced nonlinear character of the generation types of these sequences is noted as well as the possibility of obtaining the functional insolubility in essence and the highest efficiency of the implementation schemes.

This paper is based on the report presented at the 3rd Central European Conference on Cryptology in Bratislava, TATRACRYPT 2003, 26-28 June, 2003, Slovenia. The present report should be considered as a beginning of the review and discussion of the results obtained in the course of many years' research and experimental developments in the field of construction of stochastic systems. Using the theory of stochastic systems as the base, the complex of tools and technologies, combined under the general name of *randomization systems*, was developed; these systems being used to create stochastic cryptographic devices and systems of new generation, with consideration of the prospects for their development and perfection of cryptographic attacks [1]. The complex of tools covers in part and in whole all established sections of modern symmetric cryptography.

The results obtained touch upon some parts of the theory of number and of algebraic systems, concepts of functional and statistical analysis, system analysis and the dynamic system theory.

The implementation schemes of the cryptographic algorithms, the devices and the systems are universal and simple in execution. They permit a parallel one- and multi-digit treatment including a processorless hardware treatment on any platforms of computing devices. These schemes require a small storage capacity and are intended for using keys of variable length. The implementation schemes are transparent for analysis, algebraically complicated and closed; they are of a highly dynamic and parametric, hidden from the environment, simple or multiplex, multidimensional, essentially pronounced, constrained and nonlinear character. Consequently, they can possess sufficient for practical applications statistical and functional reliability. In software realizations, they have no match among the most perfect samples. As for the hardware design, they are distinguished by high efficiency and low prime cost.

Our data are based on the full-scale modeling, the statistical test package DIEHARD (G.Marsaglia, 1997, geo@stat.fsu.edu) and the criterions FIPS PUB 140-1,2 (NIST, 2001, http://cs-www,ncsl.nist.gov/rng/rng2.html). We have performed the verification as well as the preliminary statistical and functional analysis of basic and special cryptographic algorithms.

Taking into account a big amount of the accumulated theoretical and practical material, we assume to go from simple to complicated and as far as possible to support the presented results with examples and proofs.

As the first step, using the existing technologies as the base, we developed the so-called dichotomic generators. By their statistical properties, functional complexity and efficiency they far more exceed the most commonly used linear recurrence generators, which operate using the linear feedback shift registers (LFSRs). For example, for a hardware realization, the operation rate of such generators, as well as of the simplest ternary LFSRs, can be compared with the performance rate of one *XOR* (modulo 2-addition) operation independently of the length of the generation platform.

Dichotomic generators are used to form sequences of a special type which possess a set of properties inherent in the binary representation of natural sequence, i.e., the so-called dichotomic sequences. Dichotomic sequences and their properties are considered below.

Dichotomic generators can be parametric, multidimensional and multiplex. They can be easily adapted to any platforms of computing devices; they can have any repetition period that is not smaller than the one given in advance. These generators can form structural compositions of any complexity; they can be functionally insoluble in essence and at the same time to preserve the unrepeated properties of their elements.

Let us introduce a number of conceptions and terms, which are necessary for further presentation of the material.

Let us refer to a sequence $A = \{a_j\}$ defined on the set Π_A of all possible values of its elements $a_j \in \Pi_A$ as a *monocyclic* or *MP-sequence* if there exists such a minimum number $k = T$, where T is the maximum possible *repetition period*, for which the conditions $a_i = a_{i+k}$ are fulfilled for all $-\infty < i < \infty$. An *MP*-sequence can be subdivided into unrepeated and equirepeated sequences. An *MP*-sequence, which doesn't contain equal elements within the repetition period is called *M-sequence*. Further we'll consider the unrepeated sequences. Other types of sequences are stipulated separately.

Accordingly we will speak that the object (model, system, operator, etc) possesses the unrepeated properties, if the properties of unrepeated sequences are inherent in some realizations of values.

Generators of pseudorandom codes and numbers are the essential elements of any protection system. The main class of *pseudorandom* sequences, hereafter, random sequences, being formed on the basis of the above mentioned generators, is unrepeated sequences of binary numbers with a maximum repetition period. They are called maximum period sequences or *M-sequences*. *M*-sequences and the corresponding generators are the core of the symmetric cryptography.

The *linear recurrence* and the *linear congruent* methods are the known methods of formation of maximal period sequences [3, 4].

As is known, the basis of the linear recurrence method is the following monocyclic operator Γ_R with constant coefficients $a_j \in \{0,1\}$

$$\Gamma_R\, u_i \equiv \sum_{j=1}^{n} \oplus (a_j \wedge u_{j+i}),$$

given by the linear difference equation $\Gamma_R\, u_i = 0$ of the order n in the Galois field $GF(2)$ having the field elements $u_i \in \{0,1\}$ defined for all $-\infty < i < \infty$. The solution of this equation is given by the so-called linear recurrence *M*-sequences of n-bit binary numbers u with the period $T_n^R = 2^n - 1$ and the values of elements satisfying the relation

$$u_{n+i+1} = \sum_{j=1}^{n} \oplus (a_j \wedge u_{j+i}) \qquad (i = 0, 1, 2, \ldots).$$

In order to obtain these *M*-sequences one uses n-digit *shift registers* with the set $J = \{j\colon a_j \neq 0, j = \overline{1, n}\}$ of *XOR* elements realizing one-digit modulo-2 addition operations $\oplus$ in the feedback circuit. The operators of that type, LFSRs, or what is the same Γ_R-*generators*, have the following property. The integers u, which are formed on the basis of the mentioned register and composed of n bits $u_j \in u$, belong to the interval $u \in [1, T_n^R]$, and their sequence is unrepeated in character and doesn't contain zero elements.

The linear recurrence method is the easiest one to execute. This property is valid if the feedback circuit realization is fixed and structurally optimal, as well as if concurrently one can neglect the trivial solution $u \equiv 0$, which leads to generator singularity. The first of the mentioned restrictions is explained by the complexity of calculations of the coefficients a_j involved in the Γ_R-generator equation when a number m_Γ of half-adders in the feedback circuit is minimal, i.e. $m_\Gamma = min \sum_{j=1}^{n} aj - 1$. At best this number is unity but it can be very large for some n. The second of the mentioned restrictions is connected with an incomplete variation interval of random numbers. It appears if at every i-th iteration step of the Γ_R-generator a current random number u is taken equal to the shift register content. Thus, the restriction is caused by the absence of zero elements in the sequence being formed this way.

At the same time, the given method is not statistically and functionally reliable. The statistical unreliability is connected with a relatively slow change in the register state due to the one position shift as well as with the extremely small (limited by one bit) speed of influence expansion of bits at every iteration step of the Γ_R-generator. These reasons lead to a strongly pronounced correlation between elements of the sequence being formed by the generator. In order to obtain a statistically reliable sequence one usually limits himself to one bit or a part of register bits. However, this results in an abrupt slowdown of the generator efficiency due to the growth of the elementary operation number per one bit of the generated sequence and loss of its unrepeated properties. The functional unreliability of the method is caused by existence of sufficiently simple methods allowing to reconstruct any preceding values and to predict all consequent values of the sequence elements being formed by the Γ_R-generator. This allows relatively easy to reveal the generator parameters regardless of the number of register bits used to form the elements of this sequence [2, 5].

In contrast to the recurrence method, the linear congruent one is presented by the equation [4, 5] of the form:

$$x_i = a \cdot x_{i-1} + b \ mod \ g,$$

where x_o, a, b, g are the initial value, the multiplier, the increment, and the module, $\{x, a, b\}$ being non-negative integers and the module g is a positive number so that $g > x_o$, $g > a$, $g > b$.

The period length $T^C = g$ of a linear congruent sequence $\{x_i\}$ is maximal and equals g if and only if

1) b and g are coprime numbers;
2) $c = a - 1$ is divisible by p for any prime p that is the divisor of g;
3) c is divisible by 4 if g is divisible by 4.

At the corresponding choice of values of the coefficient and the module in the above equation, reliable statistical congruent M-sequences can be generated based on the linear congruent method. However, the analysis shows [2] that such sequences as well as the sequences considered above do not have the sufficient functional reliability even at a considerable truncation of a part of their significant bits.

Besides, in comparison with the recurrence method, a specialized arithmetic device or a processor is required for the realization of the congruent method. This circumstance strongly influences the final cost of the generators built using the method. Moreover the efficiency of the corresponding devices is strongly affected by multiplication operations and calculation of residues of division involved in the method. The method is good for an efficient realization of arithmetic operations and under the condition that the register digit capacity of the used computing devices is enough to perform the corresponding machine instructions. In the opposite case, it is necessary to use the arithmetic of high numbers that is inefficient and complicated for realization. These peculiarities strongly influence the efficiency and prime cost of the generators based on the congruent method.

Taking further into account the generality of the binary representation of data as well as effectiveness and simplicity of the binary command realization it is advisable to restrict oneself to the so-called linear binary congruent method given by the equation

$$x_i = a \cdot x_{i-1} + b \ \ mod\ T_n^C .$$

At $a \equiv 1(\ mod\ 4\ )$ and an odd b this equation allows to obtain the *M*-sequences of binary numbers $x \in [0, T_n^C - 1]$ composed of n significant bits with the period $T_n^C = 2^n$.

Let us consider in more detail these binary congruent sequences. Let us begin with the natural sequence given by the equation

$$x_i = x_{i-1} + 1\ mod\ 2^n,$$

where the initial condition is $x_0 = 0$ and the bits of the variable x are numbered from right to left starting from one. The following property of the given binary sequence attracts attention (see column 2 of Table 1 in the Appendix). The repetition period T_l of the first bit equals 2^1, for the second bit 2^2, for the third 2^3, for the k-th $T_k = 2^k$, and for the n-th 2^n, respectively.

Let us now consider a binary congruent *M*-sequence, presented, for example, by the equation

$$x_i = 133 \cdot x_{i-1} + 1\ \ mod\ 2^{16}.$$

The behavior law of the bits of this sequence (see column 3 of Table 1 in the Appendix) is similar to the behavior law of the bits of the natural sequence. In this case, for every k-th bit, i and $(i+T_k/2)$ elements belonging to the adjacent half-cycles of this sequence are connected by the complementary relation or the *NO* operation

$$x_{ki} = \overline{x}_{k(i+Tk/2)}.$$

The mentioned properties are inherent not only in the examples given above but in all other congruent *M*-sequences denoted further by the symbol *C*. Taking into account all the aforesaid, one can generalize the given results in the following natural way.

A binary sequence $D = \{d_i\!: i = \overline{1, T_n}\}$ of non-negative integers d_i, composed of n significant bits $b_{ij} \in d_i$ $(j = \overline{1, n})$, is called a *dichotomic* or a *D-sequence* if any of its subsequences $D_k \in D$ composed of $T_k = 2^{\,k}$ elements of the original sequence D_k by truncation of its $n - k \in [0, n-1]$ high-order $k < j \leq n$ significant bits b_{ij} is unrepeated within the period T_k, i.e. for any pair $d_{ki} \in D_k$ and $d_{k(i+Tk)} \in D_k$ of its elements, the following condition is fulfilled:

$$d_{ki} = D_{ki}\ mod\ T_k = d_{k(i+Tk)} = D_{k(i+Tk)}\ mod\ T_k \quad (1 \leq k \leq n).$$

The mentioned order will be referred to as the *dichotomic order*. The dichotomic order has the hierarchical structure of the binary tree type. It consists of levels $k \in [1,n]$ and $e_k = T_n / T_k$, the so-called *dichotomic classes* $D_{kj} \subseteq D\ mod\ T_k$ $(j = \overline{1, e_k})$ which don't intersect each other on these k levels, are independent of each other, and are equal to $D_{k1} = D_{k2} = ... = D_{kj}$ in the values and the number T_k of the elements forming the part of the classes. The subclasses $D^{\circ}_{kj} = \{D_{kji}\!: i = \overline{1, T_k/2}\}$ and $D^{\bullet}_{kj} = \{D_{kjl}\!: l = \overline{1 + T_k/2, T_k}\}$ composed of the elements of the class D_{kj}, separated by the half-cycle $T_k/2$ will be called the *self-conjugate* dichotomic classes.

A dichotomic order and the *D*-sequence corresponding to it are called *homogeneous* or *perfect* if all pairs $\{b_{ki} \in B^{\circ}_k, b_{k(i+Tk/2)} \in B^{\bullet}_k\}$ of bits i and $i + T_k/2$ $(i = \overline{1, T_k/2})$ and generated by every k-th bit of the conjugate classes $\{B^{\circ}_k \subseteq D^{\circ}_k, B^{\bullet}_k \subseteq D^{\bullet}_k\} \in D_k$ of the class D_k are complementary connected with each other by the relation:

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

The mentioned complementary correspondence between elements of conjugate classes D_k will be called *dichotomic complement*.

A dichotomic order and a sequence will be called *inhomogeneous,* if the self-conjugate classes are not complementary, i.e. among all the bit pairs $(b_{ki}, b_{k(i+Tk/2)})$ from the self-conjugate classes $\{D^{\circ}_k, D^{\bullet}_k\}$ there is at least one pair that is presented by the relation

$$b_{ki} = b_{k(i+Tk/2)}.$$

Binary quantities, realizations of which have the mentioned properties, are called *dichotomic.* Further we consider perfect dichotomic sequences and quantities and this is not specially stipulated.

A cardinal number *card* D_Z of the set of all D_Z dichotomic sequences in a modulo-2^n residue field $\mathbf{Z}/2^n$ is given by the relation

$$card\, D_Z \approx 2^{n(n+3)/2\,-\,1}$$

and it is defined with the accuracy up to isomorphism by the recurrence expression

$$d = \omega_0 + \sum_{k=1}^{n} \omega_k\, b_k\, 2^{k-1} \bmod 2^n,$$

with respect to an n-digit binary variable d and the bits $b_k \in d$ composing it, for any odd values of the coefficients $\omega_0 < 2^n$ and $\omega_k < 2^{(n-k)+1}$ and condition $\omega_1 \equiv 1 (mod\ 4)$, which further is called the *dichotomic equation in the field* $\mathbf{Z}/2^n$ *or the linear dichotomic equation*.

The congruent sequences C formed on the basis of the linear binary congruent method are a particular case $C \subseteq D_Z$ of the dichotomic sequences D_Z. The cardinal number *card* C of these binary sequences, compared to the number *card* D_Z, at high n is negligibly small and is about $2^{3(n-1)}$.

Let us consider some other, most characteristic properties of dichotomic sequences and quantities. Thus, the low-order bits of dichotomic quantities have strongly pronounced deterministic and functional irregular properties. At the same time the behavior of other bits essentially depends on values of the coefficients of the above dichotomic equation. Depending on the coefficients, the high-order bits of dichotomic quantities and the sequences being formed on their basis can have more or less pronounced irregular, non-deterministic and chaotic properties.

By virtue of the essentially pronounced irregular behavior of the low-order bits of dichotomic quantities and the complementation of the dichotomic classes generated by them, the bits composing these quantities also have a sufficiently strong correlation that leads to the functional dependence between elements of realizations of these quantities. Meanwhile, in contrast to other methods this dependence rapidly decreases due to an exponential increase in the period $T_k = 2^k$ $(k \le n)$ from bit to bit, and due to linked with it essentially pronounced, complexly predicted, non-deterministic behavior of the high-order bits; the behavior being more pronounced than the digit capacity n of elements of the sequence being generated.

In other words one can say that dichotomic sequences and quantities have dual, deterministic and non-deterministic, regular and irregular complementary functional and stochastic properties.

In spite of all the merits, the formation of dichotomic sequences based on the above mentioned dichotomic equation in the field $\mathbf{Z}/2^n$ requires to use considerable computing resources. Even 64-digit supercomputers are not sufficient for the efficient realization of practically important generators of these sequences. Therefore this equation is of greater importance for theory but not for practice.

Meanwhile, the cardinal number *cardD* of the set of all the possible states of linear dichotomic sequences D

$$card\, D = 2^{2^n - 1}$$

is incommensurably large and considerably exceeds the number of all the admissible linear recurrence, congruent and other similar dichotomic sequences. A cardinal number points out that there are other algebraic structures, which can generate *D*-sequences. Then the following question is obvious. Are there any algebraic structures, more common than the residue field $\mathbf{Z}/2^n$, which can be used to form dichotomic sequences with the required statistical properties and a high functional complexity as well as to organize efficient sequential and parallel processing? The theoretic investigations, particularly in the field of dynamic systems, and the full-scale modeling have justified the existence of such algebraic structures. By virtue of them, in whole or in part, linear properties that are inherent in the field $\mathbf{Z}/2^n$, on the one hand, and discontinuous properties, that are inherent in the field $\mathbf{Z}/2$, on the other hand, these algebraic structures were called *lined structures*. The so-called *randomization method* has been developed based on these structures. It is intended for the construction of ordinary, one- and multi-parametric, simple and multiplex multidimensional dichotomic generators having any given repetition period which is not smaller than the one given in advance; the generators have relatively good statistical properties and a high functional complexity. They permit high-performance sequential and multi-channel parallel, one and multi-digit (including processorless) treatment on any platforms of computing devices.

As an example of realization of such a generator the dichotomic sequence is represented in the Appendix (see Table 1, column 4). It is obtained using one of the hypothetic versions of the one-parametric dichotomic 16-digit generator calculated for 2-digit registers and oriented on sequential program realization. Below the original text of the algorithm is represented in the language C,

```
for( s = 3, i = 0; i < 8; i++ )            /* a realization of the sequential dichotomic generator */
{
    r = dBlock[i];                          /*    the dichotomic generator output              */
    g = r ^ ( ( s > r ) ? 1: 2 );  dBlock[i] = ( r  –  s ) & 3;  s ^= Pg[i];
    Pg[i] = g;                              /*   a change in an internal state of the generator    */
}
```

for zero initial conditions of dichotomic sequence generation, which are given by the vectors dBlock and Pg except for the first element of the parameter Pg[0], that is taken equal to 1.

The presented algorithm can be characterized by the compound algebraic structure in the residue field. It differs from well-known methods by the way of result criterion assignment and parametric representation of generator functioning equations.

Among the peculiarities of the behavior of dichotomic parametric generators it is necessary to note the presence of a small transient nonstationary interval τ that is inherent in a wide class of dynamic systems. The interval being passed, the mentioned generators proceed to a stable state and then within the period 2^n they behave as unrepeated generators of pseudorandom numbers. For example, the represented above parametric generator has such an interval with the length 4 (see column 4 of the Table 1 in Appendix). Depending on the way of a generator realization and number of its parameters p (in the best cases it will be 2-3) a transient interval length can obtain the value $p{\cdot}n$. This drawback is not a restriction for the practical employment of the generators represented here, since it can be either easily eliminated or under conditions of maximal used length of statistical sampling L_S in bits be accepted equal to $p \log_2 L_S$.

Transient intervals indicate that there is an attracting set, the so-called *attractor*. It characterizes the set of possible states or states of stable equilibrium of the generators, at which these generators have non-degenerate and unrepeated character. Due to certain perturbations appearing during the transition from one dichotomic sequence to another through the boundaries of an interval with the length 2^n, the mentioned generators can possess the monocyclic properties with the period $2^{\lambda \cdot n}$. Here λ is the monocyclic index depending on initial conditions and a generator structure, and does not exceed $p+1$. Further, we will distinguish, due to the presence of transient processes and attractors, the *monocyclic* $(\lambda > 1)$ and *quasi-periodic* $(\tau > 0)$ dichotomic sequences and generators from the above-mentioned *unrepeated* $(\tau = 0)$ generators and sequences.

Dichotomic generators are able to form the structural compositions of any complexity. Formation of functionally complicated dichotomic sequences and structural compositions can be realized on the basis of dichotomic generator complexation. In Table 2 of the Appendix we present the results of complexation (column 3) of the represented above linear congruent (column 1) and parametric dichotomic (column 2) 26-digit generators assigned by dichotomic r and x variables respectively on the basis of the lined function

$$z = r + 2 \cdot (h \oplus x) \bmod 2^{26},$$

where $\{h,z\}$ are the binary 26-digit vectors, h is the constant (modifier) and z is the resultant dichotomic quantity.

Dichotomic generators allow a multi-channel parallel processing. In the Appendix (Table 3, column 3) we present the dichotomic sequence obtained on the basis of one of the variants of the realizations of the one-parametric 13-channel dichotomic 26-digit generator designed to operate with 2-digit registers. The original text of the algorithm is represented in the language C,

```
for( s = 3, i = 0; i < 13; i++ )          /* a realization of the multi-channel dichotomic generator */
{
        r = dBlock[i];
        rBlock[i] = ((s^r) + Hg[i] & 3;         /*   the dichotomic generator output                */
        g = r ^ ( ( s > r ) ? 1: 2 );  dBlock[i] = ( r  –  s ) & 3;  s = Pg[i];
        Pg[i] = g;                              /*   a change in an internal state of the generator   */
}
```

for zero vector modification Hg and under zero initial conditions of dichotomic sequence generation, which are given by the vectors dBlock and Pg except for the first element of the parameter Pg[0], that is taken equal to 1. The length of the nonstationary interval of the generator is equal to 10. In column 4 of Table 3 of the Appendix the sequence being formed on the basis of the binary vector variable Pg is presented. This sequence describes the internal state of the generator.

The above mentioned properties of dichotomic sequences allow, using simple and clear for analysis ways, to obtain statistically and functionally reliable for practical applications sequences of uniformly distributed numbers. In this case, by the statistical reliability of the formed sequences in wide sense is meant an equal probability and independence by Shannon of their elements, and in restricted sense is meant insensitivity to the universally recognized system of statistical tests at the all-possible initial conditions of their generation. On the other hand, by the functional reliability of the formed sequence in wide sense is meant the search for the intensity needed for the reconstruction of the preceding values of its elements and prediction of the consequent values of its elements. In restricted sense we mean insensitivity to restoration of the dichotomic properties of its original *D*-sequence at any available amount of sampling of its elements that is restricted by technical possibilities.

In the general plan, the formation of statistically reliable sequences can be implemented by using the influence distribution of the high-order bits on the low-order ones and vice versa, as well as by confusion of the propagated bits in order to impart to the high-order bits the irregular properties and to the low-order bits the strongly expressed non-deterministic ones including destruction of the complementary properties inherent in dichotomic sequences.

The attainment of the statistical reliability indicates a high level of dissipation and confusion of bits of a generated sequence. In its turn the functional reliability of a generated sequence is ensured by parameterization and multidimensionality of the used transformations. Besides, it is ensured by hiding the dichotomic properties of its original *D*-sequence owing to the nonlinear and non-deterministic bit catenation in its elements that keeps the required statistical properties.

As a rule, a practical result is optimal if the problems of statistical and functional reliability assurance are mutually specified and super additive, nearly everywhere they supplement each other and harmonize with each other nearly everywhere.

To summarize the aforesaid, note the following properties inherent in dichotomic sequences and generators:

1. Frequency properties of bits expressed in an exponential growth of the repetition period from bit to bit.
2. Complementary properties relating the half-cycles by *NO* negation.
3. Unrepeated properties, conditioned by the order of lower bit catenaries with the higher ones.
4. Avalanche properties that conditioned the influence distribution of the high-order bits on the low-order ones by means of the *XOR* operation.
5. Nonlinear properties conditioned by *AND* conjunction of lower bits with the higher ones.
6. Ability of dichotomic sequences and generators to form structural compositions that possess dichotomic properties.
7. Functional indeterminacy conditioned by multiplex, parametric and multidimensional nature of generation equations.

Preliminary analyses show that parameterization and multidimensionality allow to achieve functional insolvability of generation equations in essence, on the one hand. On the other hand, the cryptographic resistance of dichotomic generators becomes apparent in avalanche and, in case of need, exponential rate of the influence distribution and essentially nonlinear catenation of low-order bits with the high-order ones. At the same time, this resistance being not detriment to the efficiency, can be appreciably increased and adjusted for expense of using of more strong algebraic structures and composing them primitives.

In conclusion it is necessary to note the conceptual connection of dichotomic sequences, processes and quantities with dynamic systems and real natural processes. This connection is confirmed by the duality of properties and by the lined character of dichotomic sequences, as well as by transient processes and attractors, which are inherent in multiplex dichotomic systems and generators, their phenomenal ability to overcome nonstationary intervals without assistance and to attain unrepeated properties.

We think that the present material will be interesting not only for cryptographic applications, but also for different natural scientific disciplines.

«DICHOTOMIC SEQUENCES AND THEIR PROPERTIES»,

Article based on the materials of the report presented at the 3rd Central-European Conference in Bratislava, TATRACRYPT 2003, 26-28 June, 2003, Slovakia

Keywords. Dynamic system. Attractor. Chaos. Stochastic system. Symmetric cryptography. Network cryptography. Random-number generator. Dichotomic order. Dichotomic sequence. Dichotomic generator. Randomization method. Randomization system.

REFERENCES

1. I. A. Kulakov The Method of the Randomization Properties Imparting to a Real Object and a Randomization System, An application for the International Patent.
2. E. F.Brickell, A. M. Odlyzko "Cryptanalysis: A Survey of Recent Results," Proc. of the IEEE, Vol.76, no.5, May 1988.
3. R. Lidl, H. Niederreiter "Finite Fields," Encyclopedia of Mathematics and its Applications, v.20, Addison-Wesley, 1983..
4. D. Knuth The Art of Computer Programming, Volume.2, Seminumerical Algorithms, 2nd edition Addison-Wesley, 1981.
5. B. Schneier APPLIED CRYPTOGRAPHY. Protocols, Algorithms, and Source Code in C, John Wiley & Sons, Inc, 1996.

Appendix

Table 1. **Congruential & Dichotomic Sequences**

1	2	3	4
001	00000001	0000000000000001	1010101010101001
002	00000010	0000000010000110	0011001100110010
003	00000011	0100010110011111	0000011100000111
004	00000100	0010101110011100	0110100100110100
005	00000101	1010100000001101	1111111100101101
006	00000110	0100111011000010	1001101001000110
007	00000111	1110101011001011	0010001111101011
008	00001000	1111101101111000	1000101010101000
009	00001001	1010010101011001	1101001100000001
010	00001010	1110011100111110	1100110110101010
011	00001011	0010001100110111	1010001111001111
012	00001100	0100101110010100	1001001100011100
013	00001101	0100001111100101	1001100010100101
014	00001110	0100010111111010	0111111011111110
015	00001111	0101101011100011	0010011111000011
016	00010000	0011011111110000	1011111011110000
017	00010001	0000111110110001	0010101010111001
018	00010010	0010011011110110	0100111100000010
019	00010011	0011110111001111	1110100110010111
020	00010100	0001110010001100	1101100010000100
021	00010101	1101010010111101	1000011000111101
022	00010110	1000011000110010	0110111101010110
023	00010111	1011011111111011	0101100110111011
024	00011000	1001010101101000	0101010110111000
025	00011001	1001111100001001	1001011011010001
026	00011010	1001111110101110	1010011000111010
027	00011011	1111010101100111	1010101110011111
028	00011100	0111111010000100	1010100000101100
029	00011101	1011101010010101	0000000101110101
030	00011110	1110111101101010	0000000001001110
031	00011111	0110001000010011	1000101111010011
032	00100000	1111001111100000	0000011111000000
033	00100001	1011001101100001	1101100100001001
034	00100010	0011000101100110	1100010111010010
035	00100011	1010100111111111	0000011001100111
036	00100100	0101000101111100	1101011000010100
037	00100110	0101010101101101	0000110010100110
039	00100111	0110000110100010	0111111011001011
040	00101000	1011100100101011	1100011111001000
041	00101001	0011001101011000	0101111011100001
042	00101010	1010110010111001	1110000000001010
043	00101011	1011110000011110	0100010110101111
044	00101100	1011101110010111	1001111000111100
045	00101101	0111010101110100	0101111000000101
046	00101110	0000010101000101	1100111010011110
047	00101111	1011110011011010	1010001100100011

Transient section

Table 2. **Complexation of Dichotomic Generators**

1	2	3
000000000000000000000001	101010101010101010101001	010111101101100100110100 01
000000000000000000010000110	110011001100110011001100 10	100000001111101110010110 00
000000000001000101100111 11	110000011100000111000001 11	011101011100111000000110 11
000010010000101011100111 00	010011011001101001001101 00	111100111101010110010001 10
101100101010101000000011 01	110010111111111111001011 01	100111010111101001010111 01
110100100101001110110000 10	100100011010011010010001 10	101000100011000000111101 00
010001010111101010110010 11	001001111100100011111010 11	011001101010010011101001 11
000110001011111011011110 00	001000100110001010101010 00	101001111011010111100000 10
110110110010100101010110 01	110110111111010011000000 01	110111100111000011100110 01
110111000111100111001111 10	100010000011001101101010 10	100101010001000010000000 00
100010110100100011001101 11	110010110010100011110011 11	011011011110100000000100 11
010111001101001011100101 00	111000001110010011000111 00	111011100111000000000111 10
001110011001000011111001 01	101000111110011000101001 01	011010101111010110100001 01
111010000101000101111110 10	111111111001111110111111 10	011001000010110000110111 00
101100100101011010111000 11	001110110100100111110000 11	000010111100110111101011 11
101001110000110111111100 00	001010100010111110111100 00	001001000110010100101110 10
110010100100000111101100 01	000101100100101010101110 01	001101101111010000000000 01
000101010100100110111101 10	101111001001001111000000 10	010110110101000110110010 00
000011110100111101110011 11	110000011111101001100101 11	011011001010101011010010 11
111101000100011100100011 00	010110101111011000100001 00	101101111001011011101101 10
111010001111010100101111 01	000101011110000110001111 01	011110111010011001100011 01
000001110110000110001100 10	100000001101101111010101 10	001110111100100101101001 00
110101011010110111111110 11	001110111101011001101110 11	010110110100101111100101 11
000000110110010101011010 00	010111010001010101101110 00	000011111111001101011001 0
110000011101001111000010 01	111011000010010110110100 01	000111111000011011000010 01
101001100010011111101011 10	111010100010100110001110 10	111000101000101011101100 00
010100101011110101011001 11	010011000010101011100111 11	010111010111111100100000 11
111111000101111110100001 00	001101100010101000001011 00	100000101011101111010011 10
000111011010111010100101 01	111100101000000001011101 01	100000011111010000011101 01
011010111011101111011010 10	000111011000000000010011 10	100000001101100101010011 00
111110001001100010000100 11	000011011010001011110100 11	010100101100001001110111 11
001001110011110011111000 00	101111101100000111110000 00	101110010001100101101010 10
011000101010110011011000 01	000001001011011001000010 01	011101100010110101111100 01
010000111100110001011001 10	001101000111000101110100 10	011010000010011110101110 00
001110010010101001111111 11	100001101100000110011001 11	010011010011110000000110 11
101100110001010001011111 00	111000101111010110000101 00	101101001111101110111001 10
000010011001010101011011 01	011100101011001100110011 01	000110011011011101101111 01
111110101001100001101000 10	111001110000001100101001 10	001010000010000110000101 00
001100010010111001001010 11	101000011001111110110010 11	011110000000100011101001 11
100011010000110011010110 00	101001111111000111110010 00	010101100010100100011000 10
010001111010101100101110 01	101000100101011110111000 01	110111011101000010001110 01
001110111110111100000111 10	101001001011100000000010 10	011010001010100010001000 00
001000110010111011100101 11	111111100001000101101011 11	111100001000010010101100 11
010001110101110101011101 00	110000001010011110001111 00	111110110011101110111111 10
000100111000000101010001 01	111100000101011110000001 01	100000111000001110101001 01
001000100010111100110110 10	000100101111001110100111 10	000000110110010010001111 00
110000101000011101010000 11	011100011010100011001000 11	101000101100100011110011 11
000100000010011001111010 000	011110111100111010110100 00	000100001000011000010110 10

Table 3. **Dichotomic Sequence & Multichannel Generator**

1	2	3	4
001	00000001	0000000000000000000000111	1010101010101010101100101
002	00000010	1010101010101010101011010	0101010101010101010101010100
003	00000011	1111111111111111111101001	0000000000000000000010011
004	00000100	0101010101010101010100100100	1111111111111111111001101
005	00000101	1010101010101010100011011	0000000000000000000010110101
006	00000110	1010101010101010100011101110	0000000000000000000001010000
007	00000111	1010101010101010100100100101	0000000000000000000111001111
008	00001000	1010101010101011000001100 0	0000000000000000000110110001
009	00001001	1010101010101000101111111	0000000000000110101010001
010	00001010	1010101010100101110011001 0	0000000000001101001101110 0
011	00001011	1010101010011011100100000 1	0000000000110101000101101 1
012	00001100	1010101001101111111010110 0	0000000011010001000101010 1
013	00001101	1010100110100011001111001 1	0000001101001101110000110 1
014	00001110	1010011010010100000110011 0	0000110100110101101100010 0
015	00001111	1001101001101010110111110 1	0011010011010101011100011 1
016	00010000	0110100110111111100100000 0	1101001101000000001111010 1
017	00010001	1010011010010101011001011 1	0100110100111111110001010 1
018	00010010	1001101001101010111110101 0	0011010011000000010001010 0
019	00010011	0110100111101011110011100 1	1101001101000000011100001 1
020	00010100	1010011011101000000001010 0	0100110101000011011011110 1
021	00010101	1001101111100101101110101 1	0011010001001101010100010 1
022	00010110	0110100011011010110111111 0	1101001101110101001100000 0
023	00010111	1010010100101110100101010 1	0100110101010000001111111 1
024	00011000	1001101001100110011010100 0	0011010101001100110000000 1
025	00011001	0110111111010101111010111 1	1101000001111111010000000 1
026	00011010	1010010000001010111010001 0	0100111101010001010000110 0
027	00011011	1001100101001111111001000 1	0011000101100000001001110 11
028	00011100	0111111001000100110101110 0	1101000101101111011101010 1
029	00011101	1011001111111001001000001 1	0101110000010001010111110 1
030	00011110	1100000110011110010111011 0	0001101101110001011101010 0
031	00011111	0010110001010011001000110 1	1101011111111100010111011 1
032	00100000	0111101100100000001010100 00	0100000101001011011101010 1
033	00100001	1101110000000110100101001 11	0111001110000101010101010 1
034	00100010	0001101000010010011111101 0	0111000110011101010000010 0
035	00100011	1101110001000111110100100 1	0111000000101010001111001 1
036	00100100	1001010110011100001000010 0	0011110101010111010010110 1
037	00100101	0110001011100001000011101 1	1100110100010101100001010 1
038	00100110	1101011000110101000100111 0	0111010011000101100111000 0
039	00100111	0010110100001011010000010 1	0101011110110100010110111 1
040	00101000	0111000111011010101011100 0	0100010101010001010101000 1
041	00101001	1100010000111111111101111 1	0111011111010000000001100 01
042	00101010	0001001101110101011001001 0	0101010101011111110011110 0
043	00101011	0110010010001010010110000 1	0101011101010000111100101 1
044	00101100	1011000111001001101100110 0	0100010100100011000100010 1
045	00101101	1000011111000111111101001 1	0011010100101101000101110 1
046	00101110	0101101101111011001000011 0	1101010101010101110101010 0
047	00101111	1010111000001100011101110 1	0101000111110001010001011 1
048	00110000	1110000011100000111001000 00	0000111101101101011101010 1