PLISKA
STUDIA MATHEMATICA BULGARICA
ПЛИСКА
БЪЛГАРСКИ МАТЕМАТИЧЕСКИ СТУДИИ

For further information on
Pliska Studia Mathematica Bulgarica
visit the website of the journal http://www.math.bas.bg/~pliska/
or contact: Editorial Office
Pliska Studia Mathematica Bulgarica
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Telephone: (+359-2)9792818, FAX:(+359-2)971-36-49
e-mail: pliska@math.bas.bg

Pliska Stud. Math. Bulgar. **16** (2004), 13-22

PLISKA
STUDIA MATHEMATICA
BULGARICA

PROBABILISTIC MODELS IN CRYPTOGRAPHY, CODING THEORY AND TESTS FOR PRNG

Verica Bakeva

This paper is a review of some applications of probabilistic models in cryptography, coding theory and tests for pseudo-random number generators (PRNG). Using quasigroup transformations, we design streams ciphers and error-correcting codes with suitable properties. Some tests for pseudo-random number generators are designed, too. They are based on random walk on discrete coordinate plane.

1. Introduction

Many processes in nature, technics, communications, transport and many other areas include the randomness in themselves, i.e. they are stochastic processes. Mathematical models which describe them are probability models. In this paper a review of some applications of probability models in cryptography and coding theory is given. Main parts of these results are published in paper [1], [2] and [3]. In Section 2, using the quasigroup transformation, we define a stream cypher which is resistant on brute force and statistical kind of attacks. In Section 3, we use quasigroups to define a suitable channel code of stream nature. This code corrects errors in the channel with high probability. In Section 4, using random walks on discrete plane, we design several tests for pseudo-random number generators.

2000 *Mathematics Subject Classification:* 94A29, 94B70

Key words: quasigroups, stream cypher, tests for pseudo-random number generators, error-correcting codes

2. Encryption and decryption functions

Cryptography is the study of mathematical techniques related to aspects of information security when messages are transmitted through an insecure channel. Information security means confidentiality (secrecy), data integrity, entity authentication and data origin authentication, etc. To make this possible it is necessary to transform a message in such way to hide its substance. This process is called encryption and an encrypted message is ciphertext. The process of turning ciphertext into origin message is decryption. Now, a cryptographic algorithm is a mathematical function used for encryption and decryption. There are two important classes of encryption algorithm: block ciphers and stream ciphers. Here, a stream cipher is proposed.

Let A be an alphabet ($|A| \geq 2$) and denote by $A^+ = \{x_1 \dots x_k | \quad x_i \in A, \quad k \geq 1\}$ the set of all finite strings over A. Assuming that $(A, *)$ is a given quasigroup, for a fixed letter $a \in A$ we define transformations $E = E_a^{(1)} : A^+ \to A^+$ and $D = D_a^{(1)} : A^+ \to A^+$ by

$$(1) \quad E(x_1 \dots x_k) = y_1 \dots y_k \Leftrightarrow \begin{cases} y_1 &= a * x_1, \\ y_{i+1} &= y_i * x_{i+1}, \quad (i = 1, \dots, k-1) \end{cases}$$

$$(2) \quad D(y_1 \dots y_k) = x_1 \dots x_k \Leftrightarrow \begin{cases} x_1 &= a \setminus y_1, \\ x_{i+1} &= y_i \setminus y_{i+1}, \quad (i = 1, \dots, k-1) \end{cases}$$

where $x_i, y_i \in A$, $k \geq 1$. Then, for given quasigroup operations $*_1, *_2, \dots, *_n$ on the set A, we can define mappings $E_1, E_2, \dots, E_n$, $D_1, D_2, \dots, D_n$ in the same manner as previous by choosing fixed elements $a_1, a_2, \dots, a_n \in A$ (such that E_i, D_i are corresponding to $*_i$ and a_i). Let

$$E = E_{a_n,\dots,a_1}^{(n)} = E_n \circ E_{n-1} \circ \dots \circ E_1, \quad D = D_{a_1,\dots,a_n}^{(n)} = D_1 \circ D_2 \circ \dots \circ D_n$$

where $\circ$ is the usual composition of mappings. It is easy to check that the mappings E and D are bijections and $D = E^{-1}$ is the inverse bijection of E. So, these functions can be used for encryption and decryption purposes. If the function E is used for encryption, we have proved that our encryption algorithm is resistant of brute-force attacks. It follows from the next theorem.

Theorem 1. *For finding all pairs of n-tuples $(*_1, \dots, *_n)$, $n \geq 2$, of quasigroup operations on A such that the equality*

$$E_{a_n,\dots,a_1}^{(n)}(b_1 b_2 \dots b_k) = c_1 c_2 \dots c_k$$

holds for given $a_1, \dots, a_n \in A$, one needs to make at least as many trials as there are $(n-1)$-tuples of quasigroup operations on A.

Now, the number of quasigroup operation on a set A is a huge one (for example there are more than 10^{58000} quasigroup operations when $|A| = 256$) and thus the brute force attack is not reasonable by Theorem 1.

In the next part we prove that the statistical kind of attacks are not promising, too.

2.1. The uniformity obtained by quasigroups

Let now take that the alphabet A be $\{\mathbf{0}, \ldots, \mathbf{s-1}\}$ where $0, 1, \ldots, s-1$ $(s > 1)$ are integers, and we define a sequence od random variables $\{Y_n |\ n \geq 1\}$ as follows. Let us have a probability distribution $(q_0, q_1, \ldots, q_{s-1})$ of the letters $\mathbf{0}, \mathbf{1}, \ldots, \mathbf{s-1}$, such that $q_i > 0$ for each $i = 0, 1, \ldots, s-1$ and $\sum_{i=0}^{s-1} q_i = 1$. Consider a transformation $E = E^{(1)}$ obtained by a quasigroup operation $*$ on A, and let $\gamma = E(\beta)$ where $\beta = b_1 \ldots b_k$, $\gamma = c_1 \ldots c_k \in A^+$ $(b_i, c_i \in A)$. We assume that the string β is arbitrary chosen. Then by $\{Y_m = i\}$ we denote the random event that the m-th letter in the string γ is exactly $\mathbf{i}$. The construction of the mapping E given by (1) implies

$$P(Y_m = j |\ Y_{m-1} = j_{m-1}, \ldots, Y_1 = j_1) = P(Y_m = j |\ Y_{m-1} = j_{m-1})$$

since the appearance of the m-th member in γ depends only of the $(m-1)$-th member in γ, and not of the $(m-2)$-th,..., 1-st ones. So, the sequence $\{Y_m |\ m \geq 1\}$ is a Markov chain, and we refer to it as a quasigroup Markov chain (qMc).

Let p_{ij} denote the transition probability that in the string γ the letter $\mathbf{j}$ appears immediately after the given letter $\mathbf{i}$, i.e.

$$p_{ij} = P(Y_m = j |\ Y_{m-1} = i), \qquad i, j = 0, 1, \ldots, s-1.$$

The definition of qMc implies that p_{ij} does not depend of m, so we have that qMc is a homogeneous Markov chain. The probabilities p_{ij} can be determined as follows. Let $\mathbf{i}, \mathbf{j}, \mathbf{t} \in A$ and let $\mathbf{i} * \mathbf{t} = \mathbf{j}$ be a true equality in the quasigroup $(A, *)$. Then

$$P(Y_m = j |\ Y_{m-1} = i) = q_t,$$

since the quasigroup equation $\mathbf{i} * x = \mathbf{j}$ has a unique solution for the unknown x. So, $p_{ij} > 0$ for each $i, j = 0, \ldots, s-1$, i.e. the transition matrix $\Pi = (p_{ij})$ of qMc is regular. Clearly, as in any Markov chain, $\sum_{j=0}^{s-1} p_{ij} = 1$. But for the qMc we also

have

$$\sum_{i=0}^{s-1} p_{ij} = \sum_{\mathbf{t}\in A} q_t = 1$$

i.e. the transition matrix Π of a qMc is doubly stochastic.

Theorem 2. *Let $\beta = b_1 b_2 \dots b_k \in A^+$ and $\gamma = E^{(1)}(\beta)$. Then the probability of the appearance of a letter $\mathbf{i}$ at the m-th place of the string $\gamma = c_1 \dots c_k$ is approximately $\frac{1}{s}$, for each $\mathbf{i} \in A$ and each $m = 1, 2, \dots, k$.*

P r o o f. As we have shown above, the transition matrix Π is regular and doubly stochastic. The regularity of Π implies that there is a unique fixed probability vector $p = (p_0, \dots, p_{s-1})$ such that $p\Pi = p$, and all components of p are positive. Also, since Π is a doubly stochastic matrix too, one can check that $\left(\frac{1}{s}, \frac{1}{s}, \dots, \frac{1}{s}\right)$ is a solution of $p\Pi = p$. So, $p_i = \frac{1}{s}$ $(i = 0, \dots, s-1)$. □

The theorem 2 tells us that the distribution of the letters in the string $\gamma = E(\beta)$ obtained from a sufficiently large string β is uniform.

Let consider the distributions of the substrings $c_{i+1} \dots c_{i+l}$ of the string $\gamma = E^{(n)}(\beta)$ $(\beta = b_1 b_2 \dots b_k \in A^+)$, where $l \geq 1$ is fixed and $i \in \{0, 1, \dots, k-l\}$. As usual, we say that $c_{i+1} \dots c_{i+l}$ is a substring of γ of length l.

Define a sequence $\{Z_m^{(n)} |\ m \geq 1\}$ of random variables by

$$Z_m^{(n)} = t \Longleftrightarrow \begin{cases} Y_m^{(n)} = i_m^{(n)},\ Y_{m+1}^{(n)} = i_{m+1}^{(n)},\ \dots, Y_{m+l-1}^{(n)} = i_{m+l-1}^{(n)}, \\ t = i_m^{(n)} s^{l-1} + i_{m+1}^{(n)} s^{l-2} + \dots + i_{m+l-2}^{(n)} s + i_{m+l-1}^{(n)} \end{cases}$$

where the superscripts (n) denote the fact that we are considering substrings of a string $\gamma = \mathbf{i}_1^{(n)} \mathbf{i}_2^{(n)} \dots \mathbf{i}_k^{(n)}$ obtained from a string β by transformations of kind $E^{(n)}$. Thus, $Y_m^{(n)}$ is just the random variable Y_m defined as before. We have proved that the sequence $\{Z_m^{(n)} |\ m \geq 1\}$ is also a Markov chain (n-qMc). By using the induction of the number n of quasigroup operations, we have also proved that its transition matrix is regular and doubly stochastic for each $1 \leq l \leq n$. On the same way as previous, we obtained that the next theorem holds.

Theorem 3. *Let $1 \leq l \leq n$, $\beta = b_1 b_2 \dots b_k \in A^+$ and $\gamma = E^{(n)}(\beta)$. Then the distribution of substrings of γ of length l is uniform.*

Remark. Generally, the distribution of the substrings of lengths l for $l > n$ in a string $\gamma = E^{(n)}(\beta)$ is not uniform, since the transition matrix must not be doubly stochastic.

According to Theorem 2 and Theorem 3, it is sufficient to choose enough large number of quasigroup operations, and the distributions of letters, pairs of letters, triplets, and so on, in the transformed text will be uniform and statistical kind of attacks are not reasonable. The detailes of the proof of the Theorem 3 are published in the paper [1].

3. Error-correcting code

Error-correcting and error-detecting codes take the most important place in coding theory. Error-correcting codes are widely used in applications such as returning pictures from deep space, design of registration numbers and so on. They are used to correct errors when messages are transmitted through a noisy communication channel. A channel may be a telephone line, a high frequency radio link, or a satellite communication link. A noise may be produced by human errors, lightnings, thermal noises, imperfections in equipment, etc., and may result in errors so that the data received is different from that sent. The object of an error-correcting code is to encode the data, by adding a certain amount of redundancy to the message, so that the original message can be recovered if (not too many) errors have been occurred.

Here, we define an error-correcting code using quasigroups. In what follows we consider only the set $A = \{0, 1\}$, and $*$ will denote a quasigroup operation on the set A. There are only two quasigroup operations on the set A, and here we took $(A, *)$ to be defined by the table

(3)

$*$	0	1
0	1	0
1	0	1

Let $a_1a_2 \dots a_n \dots$ be a source message, where $a_i \in A = \{0, 1\}$, for each i and let $b_0 \in A$ be a given (known) binary letter. The sequence $b_1b_2 \dots b_n \dots$ is obtained from the sequence $a_1a_2 \dots a_n \dots$ such that $b_i = b_{i-1} * a_i$, for each $i = 1, 2, \dots$. Actually, $b_1b_2 \dots b_n = E(a_1a_2 \dots a_n)$, where the transformation E is defined as in (1). Now, we send the sequence $a_1b_1a_2b_2 \dots a_nb_n \dots$ through a noisy channel. Since there are noises in the channel, the sequence obtained in the exit of the channel can be different than the sent ones. We consider binary symmetrical channel, which means that 0 can be replaced by 1 (and opposite, 1 by 0) with probability p $(0 < p < 1/2)$.

The reason why the sequence $a_1b_1a_2b_2 \dots a_nb_n \dots$ is sent, is for checking the correctness of the transmission of the binary data $a_1a_2 \dots a_n \dots$. The checking of the correctness and correction of the incorrect transmitted data can start immediately after receiving of the first two letters of the message. Namely, when the letters a_1 and b_1 are received, it can be checked if $b_0 * a_1 = b_1$, after that if $b_1 * a_2 = b_2$, and so on. Since there are noises in the channel, some of the equations in the previous sequence should not be satisfied. Therefore, we propose the following algorithm for error correction (Table 1).

if $b_{i-1} * a_i \neq b_i$ and $b_i * a_{i+1} \neq b_{i+1}$	then $b_i \leftarrow 1 - b_i$
if $b_{i-1} * a_i \neq b_i$ and $b_i * a_{i+1} = b_{i+1}$	then $a_i \leftarrow 1 - a_i$

Table 1: Algorithm for error-correction

Theorem 4. *(see [3]) If the error distance is at least 3 then all of the errors will be corrected (i.e. the obtained message at the exit of the channel will be identical with the source one).*

In the worst case, a subsequence of a given message will be incorrectly received if the errors will happen on the distance less than 3. The probability of that event is

$$p^2 + p(1-p)p + p(1-p)^2p = p^2[3(1-p) + p^2],$$

and this probability is small for enough small values of p. On this way, we have constructed codes which correct incorrectly transmitted letters with high probability. The advantage of our code is in its stream nature. All codes which we know use blocks with fixed or variable length as codewords and decoding can start after receiving of one block. Here, the decoding can start immediately after receiving of the first two letters.

4. Tests for pseudo-random number generator

There are many situations in cryptography where it is important to be able to generate random numbers, bit-strings, etc. But, in practice, we cannot design a perfect random generator, since the way we are building the device is not a random one, which affects the uniformity of the produced sequences. That is why we use the word "pseudo" and we have to measure the randomness of the obtained sequences. There are a lot of tests for such measurements and all of them measure the difference between the obtained sequences by a PRNG and the

theoretically supposed ideal random sequence. We can classify PRNGs depending of the tests which they have passed. So, for obtaining a better classification we should have many different tests. Here, using the random walk (with fixed and random number of steps) on a discrete coordinate plane and three different ways of dividing the plane of regions, we propose 6 new tests for PRNG's.

Random walking with fixed number of steps is defined on the following way. Let k be a fixed positive integer. Let $\alpha = s_1 s_2 \dots s_d$ be a given sequence, where $s_i \in \{0, 1, 2, 3\}$. Beginning from the coordinate center $(0, 0)$ we make k steps (left, right, up, down) according to the values of the first k elements $s_1 s_2 \dots s_k$ and we add 1 to the weight of the coordinate (m, n) where the walk is stopped. After that, beginning again from $(0, 0)$, we continue to walk following the next k elements $s_{k+1} \dots s_{2k}$ and we increase the weight of the point where the walk stopped, and so on. This walk is called "chess-walk". For a given pseudo-random sequence, we can count the weights of the points of the plane. Note that the weight of a point is, in fact, the frequency of stops at that point. On the other hand, assuming that we have a perfectly uniform random sequence, we can count the weights as a product of the probability of the arrival at the point (m, n) and the number of trials, obtaining in such a way the theoretical frequency of arrivals. Since the walk is according to a random sequence, the points of stops can be described by a random vector (X, Y) and its probability distribution is determined by the following theorem.

Theorem 5. *Let (m, n) be a point of the discrete plane and let k be a positive integer. Then the probability $P_k(m, n) = P\{X = m, Y = n\}$ that a walk beginning from the coordinate center $(0, 0)$ will stop at the point (m, n) after k steps is equal to 0 in the case when $|m| + |n| > k$ or the number $|m| + |n| + k$ is odd, and in the opposite case it is equal to*

$$P_k(m,n) = \frac{1}{4^k} \sum_{q=0}^{\frac{k-|m|-|n|}{2}} \binom{k}{|m|+q} \binom{k-|m|-q}{q} \binom{k-|m|-2q}{\frac{k-|m|-|n|-2q}{2}}.$$

The coordinate X (or Y) can be presented as a sum $X = \sum_{i=1}^{k} X_i$, where X_i is a r.v. denoting the walking in the i-th step and

$$X_i : \begin{pmatrix} -1 & 0 & 1 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}. \tag{4}$$

Namely, $X_i = -1$ if the step is to the left, $X_i = 1$ if the step is to the right and $X_i = 0$ if the step is up or down. The mean and the variance of X_i are $EX_i = 0$

and $DX_i = 1/2$ which imply that $EX = 0$ and $DX = k/2$. By the central limit theorem we have:

Corollary 1. *The distribution of the random variable X converge to the normal $N(0, k/2)$ distribution for enough large k.*

We define another kind of walk called a "sun-walk". The difference between the chess-walk and the sun-walk is only in choosing a different number of steps before stop. Namely, now at first we fix an integer $l > 1$. Then we read the numbers $s_1, s_2, \ldots, s_l$ of the sequence $\alpha = s_1 s_2 \ldots s_d$ and after that beginning from $(0, 0)$ we make k_1 steps following the sequence $s_{l+1} \ldots s_{l+k_1}$, where $k_1 = 4^{l-1} \cdot s_1 + 4^{l-2} \cdot s_2 + \ldots + 4 \cdot s_{l-1} + s_l$ (i.e. we consider that the sequence $(s_1 s_2 \ldots s_l)_4$ is the notation of k_1 in 4-base system). As previous, we increase the weight of the point where the walk stopped. After that we choose the next l members $s_{k_1+l+1}, \ldots, s_{k_1+2l}$ and beginning again from $(0, 0)$ we make $k_2 = (s_{k_1+l+1} \ldots s_{k_1+2l})_4$ steps following the sequence $s_{k_1+2l+1} \ldots s_{k_1+2l+k_2}$, and so on. Note that $0 \leq k_i \leq 4^l - 1$ for each $i = 1, 2, \ldots$ So, the number of steps k_i can be considered as a random variable K with set of values $\{0, 1, \ldots 4^l - 1\}$. Let consider the case of a perfectly uniform random sequence. Then using the total probability theorem, the probability $P(m, n) = P\{X = m, Y = n\}$ that a walk beginning from the coordinate center $(0, 0)$ will stop at the point (m, n) is given by $P(m, n) = \sum_{k=0}^{4^l-1} P_k(m, n) P\{K = k\}$, where $P_k(m, n)$ is defined as for chess-walk. Also, in this case, K has the uniform distribution on the set $\{0, 1, \ldots 4^l - 1\}$ and so $P\{K = k\} = \dfrac{1}{4^l}$. Thus, we have proved

Theorem 6.

$$P(m, n) = \frac{1}{4^l} \sum_{k=0}^{4^l-1} P_k(m, n).$$

Also, can be proved that

Corollary 2. *The distribution of the random variable X can be approximated by the normal $N\left(0, \dfrac{4^l - 1}{4}\right)$ distribution.*

For designing of tests we consider three ways of dividing the plane by using:

1) the coordinate axis - the plane is divided on four quadrants: $\{(x, y) | x \geq 0, y > 0\}$, $\{(x, y) | x < 0, y \geq 0\}$, $\{(x, y) | x \leq 0, y < 0\}$, $\{(x, y) | x > 0, y \leq 0\}$;

2) circles - the plane is divided on rings $\{(x,y) \mid (2i)^2 \leq x^2+y^2 < (2i+2)^2\}$ for $i = 0, 1, 2 \ldots$;

3) squares - the plane is divided on bands $\{(x,y) \mid 2i \leq |x|+|y| < 2i+2\}$ for $i = 0, 1, 2, \ldots$.

Now, using the combinations of the three ways of dividing the discrete plane on regions and the two types of walking, we design the following six tests.

– Chess-Quadrant Test (CQT)
– Sun-Quadrant Test (SQT)
– Chess-Circle Test (CCT)
– Sun-Circle Test (SCT)
– Chess-Square Test (CST)
– Sun-Square Test (SST)

In each of them, we compare the random sequences obtained by PRNGs with the supposed theoretical ones by using the Pearson χ^2-test.

Remark. We have divided the discrete plane on circles because of the normal distribution (Propositions 1 and 2). On the other side, the limitations $|m|+|n| \leq k$ in Theorem 5 suggested the division of the discrete plane by this kind of squares.

We made many experiments in order to check several PRNGs presented in [5] with our tests. We checked MWC (multiply-with-carry) generator, KISS generators, ULTRA which combines a Fibonacci generator, CG (congruential generator), RAN2 from Numerical Recipes [4] and MSRAN (system generator in Microsoft Fortran). The obtained results are given in Table 2 below. For each PRNG we have presented results of two experiments, and the bold numbers denote the cases when the PRNG did not pass the corresponding test. In our experiments we wanted to have about $r = 10^6$ stops, i.e. the weight of the plane to be about 10^6. We took $k = 256$ (and then d is about 256×10^6) when chess-walk was used, and $l = 4$ for the sun-walk (in which case the number of steps before stops is between 0 and 255, and the average value of d is about 130×10^6).

It can be seen from the Table 2 that we can classify different PRNGs. So, MWC and ULTRA passed the tests quite well, KISS passed the tests relatively well, while RAN2 and MSRAN did not pass the tests designed by sun-walk (and it seems that MSRAN is better then RAN2 according to these tests). Depending on the parameters of CG, we obtained quite different values of χ^2-statistics, i.e. we can conclude that CG is a kind of unstable PRNG. The obtained results are published in the paper [2].

	CQT	SQT	CCT	SCT	CST	SST
MWC	3.6507	1.1157	28.2029	31.0619	30.1743	**51.8553**
	1.9320	5.2171	29.1727	15.4776	28.3449	27.2860
KISS	7.6002	0.4745	**39.5095**	16.6549	**50.3437**	26.7905
	5.1371	4.4888	**41.9264**	17.6388	**64.0689**	23.4839
ULTRA	7.4117	2.9033	17.3133	20.9626	34.0260	25.2775
	1.8869	**10.2128**	24.4770	18.8330	34.1187	25.2625
CG	**42.0610**	**11.8853**	**596.0693**	**161.7642**	**536.1579**	**156.8406**
	646.3155	**389.4645**	26.2560	35.5271	41.2609	28.1710
RAN2	**16.5810**	**31.7990**	25.5687	**517.5578**	29.4951	**554.2418**
	11.3912	**13.3155**	27.8888	**551.4363**	28.9031	**606.0271**
MSRAN	5.0328	**9.9846**	19.7406	**444.5602**	31.1509	**508.2729**
	4.4327	**8.3007**	32.1389	**447.7816**	43.5622	**521.6509**

Table 2: The values of χ^2-test statistics

REFERENCES

[1] S. Markovski, D. Gligoroski, V. Bakeva. Quasigrouop string processing: Part 1, *Contributions, Section of Mathematucal and Technical Sciences, MANU,XX* **1-2** (1999), 13–28.

[2] S. Markovski, D. Gligoroski, V. Bakeva. Random walk tests for pseudo random number generators, *Mathematical Communications, Osijek*, **6**(2) (2001), 135–143.

[3] S. Markovski, V. Bakeva. Quasigroup Based Stream Error Correcting Codes, *Proceedings of the 2nd CIIT, Molika* (2001), 14–19.

[4] Press, Teukolsky, Portable Random Number Generators, *Computers in Physics,* **6**(2) (1992), 522–524.

[5] ftp://stat.fsu.edu/pub/diehard

Verica Bakeva
Faculty of the Natural Sciences and Mathematics,
The Institute of Informatics, P.O.Box 162
Skopje, Republic of Macedonia
e-mail: `verica@ii.edu.mk`