DETECTION OF WEAKNESSES OF PSEUDORANDOM SEQUENCES WITH THE AID OF STATISTICAL TESTS*

ANDRZEJ PASZKIEWICZ[1], KAROL GÓRSKI[1], ZBIGNIEW KOTULSKI[2], JANUSZ SZCZEPAŃSKI[2], ANNA ZUGAJ[1]

[1]Institute of Telecommunications, Warsaw University of Technology, ul Nowowiejska 15/19, 00-665 Warsaw, Poland, email: anpa@tele.pw.edu.pl

[2]Institute of Fundamental Technological Research, Polish Academy of Sciences, ul. Świętokrzyska 21, 00-049 Warsaw, Poland, email: zkotulsk@ippt.gov.pl

Abstract: This work contains a summary of long term research into the statistical properties of sequences generated by least primitive roots of prime numbers. Extensive statistical tests were applied to sequences produced by linear congruential generators modulo a prime number with the generating element equal to the least primitive root of the prime number. The statistical investigations were based on Maurer's test and the group of tests contained in FIPS 140-1.

Keywords: pseudorandom sequences, statistical tests, cryptography

1. INTRODUCTION

An important class of pseudorandom sequence generators is the class of linear congruential generators described by the formula:

$$X_k = g \cdot X_{k-1} (\mathrm{mod}\, p) \qquad k = 1,2,..., \quad X_0 \neq 0 \tag{1}$$

where p is a prime number. As table 1 shows, the smallest numbers g with the property that the sequence $(X_k)_{k=0,1,...}$ includes all numbers from the set $\{1,2,...,p\text{-}1\}$ are small compared to p. The smallest natural numbers g which generate modulo p the multiplicative group $GF^*(p)$ are called the least primitive roots modulo p. Under certain hypotheses one can prove that for every natural number g which is not a power of another natural number, there exists an infinite number of primes p, for which g is the least primitive root. Numerical investigations show that prime numbers with a given least primitive root g form a fixed fraction of all primes (i.e. they have a fixed density) [6,7,8]. Below we present partial details in table 1 and figures 2, 3 and 4. For example, if we take $g = 2$, then 0.3739558136... of all prime numbers have 2 as their least primitive root. This fraction is equal to the so called Artin's constant.

As expected, the statistical properties of sequences produced by linear congruential generators (1) when g is a small natural number make them unsuitable for cryptographic purposes. Fig. 1 shows a diagram of the output of the generator $X_n = 3 \cdot X_{n-1} (\mathrm{mod} 17)$. From the diagram one can draw the conclusion that successive values of X_n are situated on segments of parallel lines. This is a fixed characteristic of these generators, which eliminates them from applications other than simple computer simulations.

Congruential generators have undergone many statistical tests, among them the tests contained in FIPS 140-1 [1], Maurer's test [5] and Feldman's test [2]. The FIPS tests belong to a group of 'weak' tests, which qualify some sequences (a little over 50% of all sequences) as statistically good. Stronger tests such as Maurer's and Feldman's eliminate these sequences as statistically non-random.

* This work was partly supported by grant no. 8 T11 D 011 12 from the Polish Scientific Research Committee.

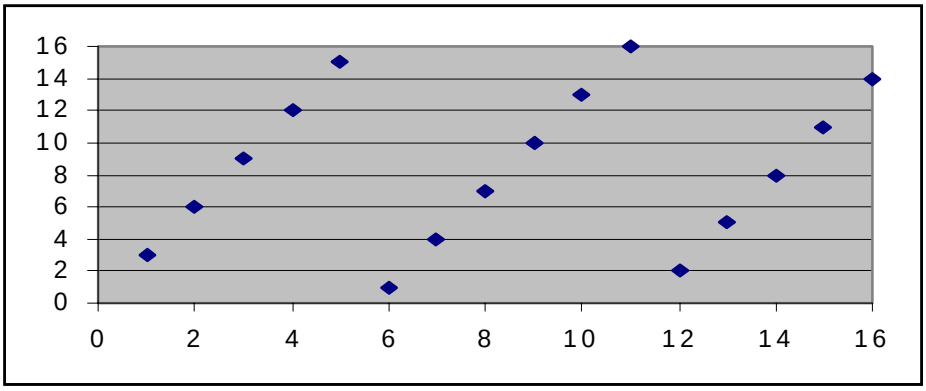

Fig. 1: Diagram of the output of the generator $X_n = 3 \cdot X_{n-1} (\mathrm{mod} 17)$

Tab. 1: Frequency of occurrence of initial natural numbers as least primitive roots modulo a prime number for primes < $36*10^9$

n	$\sum_{\substack{p<36\cdot10^9 \\ g(p)=n}} 1$	$\frac{1}{\pi(36\cdot10^9)} \sum_{\substack{p<36\cdot10^9 \\ g(p)=n}} 1$
2	578791091	0.373954801240
3	350730947	0.226605978588
5	215232325	0.139060815841
6	86496302	0.055884943506
7	106340251	0.068706046180
10	35721051	0.023079240048
11	57634400	0.037237374472
12	5049384	0.003262388484
13	35962076	0.023234965417
14	12799563	0.008269750713
15	6500887	0.004200199249
17	17903829	0.011567598257
18	625994	0.000404452428
19	11742098	0.007586526455
20	261044	0.000168659571
21	2474592	0.001598824816
22	3806069	0.002459087223
23	5993204	0.003872187125
24	35231	0.000022762620
26	2006882	0.001296639100
28	232955	0.000150511371
29	3412184	0.002204599569
30	162350	0.000104893740
31	2288902	0.001478851188

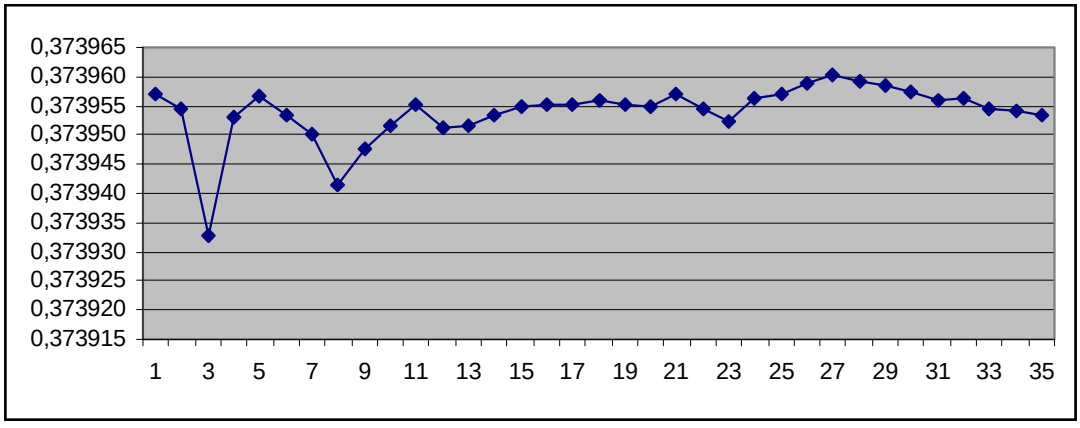

Fig. 2: Frequency of prime numbers with the least primitive root 2 tabulated with step 10^9

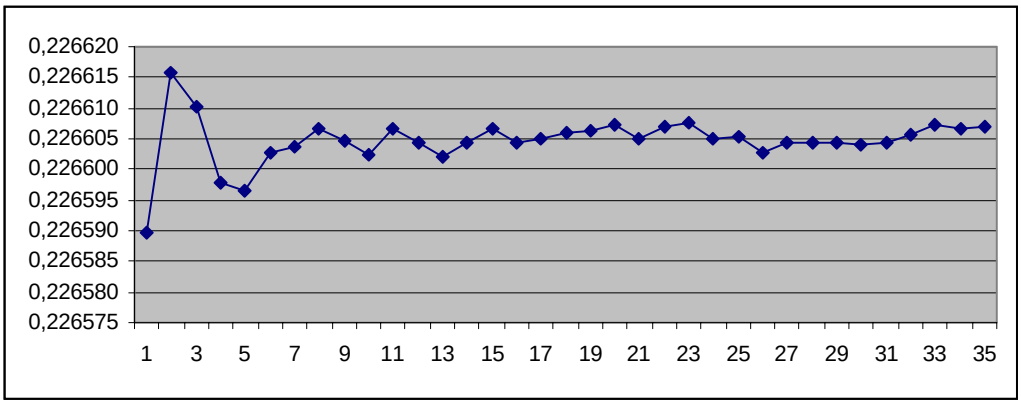

Fig. 3: Frequency of prime numbers with the least primitive root 3 tabulated with step 10^9

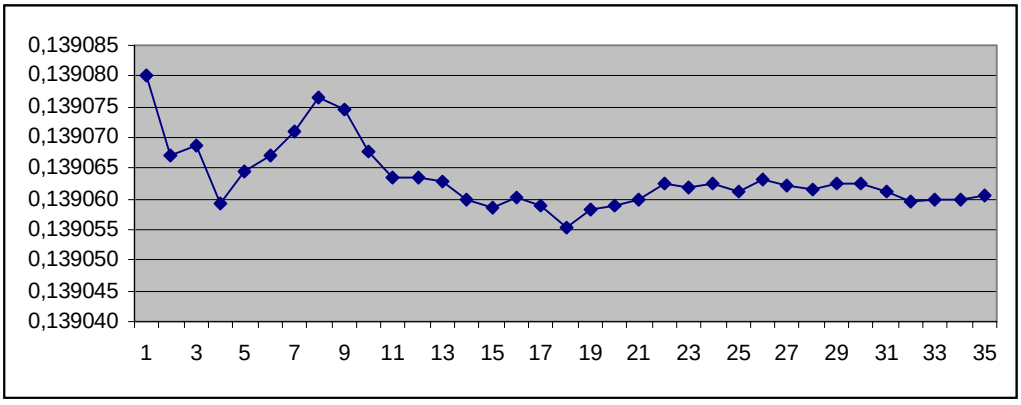

Fig. 4: Frequency of prime numbers with the least primitive root 5 tabulated with step 10^9

In view of the above, sequences generated by the least primitive roots of prime numbers may be called predictable. To avoid this weakness the sequences obtained from linear generators may be subjected to various transformations. Usually these transformations also improve the statistical properties of the sequences.

2. TRANSFORMATIONS OF SEQUENCES PRODUCED BY LINEAR CONGRUENTIAL GENERATORS

The simplest transformation (other than an affine transformation) of a sequence produced by a linear congruential generator is a quadratic transformation. We will not consider the affine transformation because it does not change the character of the generated sequence. Worth mentioning is the fact that the quadratic transformation (quadratic congruential generator), despite a significant improvement of the statistical properties, is relatively easy to predict [4]. The quadratic generator is defined with a recursive formula:

$$X_{n+1} = aX_n^2 + bX_n + c \pmod{p} \qquad (2)$$

One can prove that for an appropriate choice of parameters a, b, c [4] the sequence obtained from a quadratic generator has period p regardless of the choice of the initial value (seed). However, no transformation in which the next element of the sequence is described by a low degree polynomial with the argument set to the last element does not ensure sufficient unpredictability.

A large class of transformations which may be used for the construction of pseudorandom sequences with high unpredictability is the class of so called permutation polynomials. One can prove that every bijective mapping $f: \mathrm{GF}(p) \to \mathrm{GF}(p)$ may be expressed as a polynomial f of degree not higher than p-2 over GF(p). In particular, when Y_n is a permutation of the set $\{1,2,...,p\text{-}1\}$ for n=1,2,...,p-1 then a polynomial f of degree not higher than p-2 may be found which generates this permutation. This means that:

$$Y_n = f(X_n) \pmod{p} \qquad (3)$$

for n=1,2,...,p-1.
Deciding whether a given polynomial with coefficients from a finite field is a permutation polynomial is algebraically and computationally hard,

although in specific cases the answer may be immediate.

Example 1.
a) Every linear transformation over F_q is a permutation polynomial
b) The monomial x^n is a permutation polynomial over F_q if $(n,q\text{-}1) = 1$

Example 2.
If F_q is a finite field of characteristic p then

$$f(x) = \sum_{i=1}^{n} a_i x^{p^i} \in F_q \tag{4}$$

is a permutation polynomial over F_q if and only if $f(x)$ has root 0 in F_q.
Another example of permutation polynomials are so called Dickson polynomials [4].

Use of permutation polynomials in conjunction with linear generators is advantageous only when the permutation polynomials are of low degree or are sparse (only a small number of coefficients are non-zero). Only then can the transformation be effectively implemented. When we have (hardware) implementations of DES, IDEA or other classical block ciphers available, their superposition with the linear congruential generator gives good results. In particular linear congruential generators (1) with at least 64 bit moduluses and least primitive root 2 were used for testing the quality of DES and IDEA ciphers. Excellent statistical properties can also be obtained through the use of the knapsack transformation [9] or a linear change of base. The idea is to associate with each number 2^i, $0 \le i \le n\text{-}1$, where n is the number of bits in the binary representation of p-1, a certain binary vector from an n-dimensional space. The set of all n vectors associated with the numbers 2^i, $0 \le i \le n\text{-}1$ is a set of linearly independent vectors. Given the k-th element of the sequence produced by a linear congruential generator we construct a vector which is the sum (exclusive or) of all vectors associated with the non-zero bits of the binary representation of X_k. From this vector we can calculate the number Y_k using the reverse procedure.

3. STATISTICAL TESTS

Three types of tests were used for testing the sequences generated by linear congruential generators: combinatorial tests from FIPS 140-1 (their description may be found in [1]), Feldman's spectral test [2] and Maurer's entropy test [5]. We shall devote some attention to the last of these as it is relatively less well known.

Maurer's test is described by three parameters L, Q and K. The binary sequence is divided into successive L-bit long blocks. The total length of the sample is $N = (Q+K)L$ bits. An initialisation procedure consists of Q steps and the test consists of K steps.

Let $b_n(s^N) = [u_{Ln}, u_{Ln+1}, \ldots, u_{Ln+L-1}]$ denote the n-th block for $0 \le n \le Q+K-1$, where s^N denotes the complete sequence under test. The test statistic is defined as follows:

$$f_T(s^N) = \frac{1}{K} \sum_{n=Q}^{Q+K-1} \log_2 A_n(s^N) \tag{5}$$

The function $A_n(s^N)$ is the distance of the n-th block from its last occurrence in the whole sequence, i.e. :

$$A_n(s^N) = \begin{cases} n, \text{ if there is no } i \ge 1 \text{ such that } b_n(s^N) = b_{n-i}(s^N) \\ \min\{i : i \ge 1, b_n(s^N) = b_{n-i}(s^N)\} \text{ otherwise} \end{cases} \tag{6}$$

It is recommended to choose L from 8 to 16 bits and $Q \ge 5 \cdot 2^L$.

The test is relatively easy to implement with the aid of a table (denoted below as *Tab*) of size 2^L elements. The main part of the program for calculating the value of the function f_T may be constructed in the following way:

```
FOR i := 0 TO 2^L-1 DO Tab[i] := 0;
FOR n := 0 TO Q-1 DO Tab[b_n(s^N)] := n;
sum := 0.0;
FOR n := Q TO Q + K - 1 DO BEGIN
  sum := sum + log2(n - Tab[b_n(s^N)]);
  Tab[b_n(s^N)] := n;
END;
f_T(s^N) := sum/K;
```

For an ideal sequence the blocks b_n are independent and identically distributed. For a given level of significance α the critical values of the test equal:

$$\begin{aligned} t_1 &= E(f_T) - y\sigma \\ t_2 &= E(f_T) + y\sigma \end{aligned} \tag{7}$$

where: $E[f_T]$- expected value of the test statistic, $\sigma = \sqrt{\frac{Var[\log_2 A_n(s^N)]}{K}} \approx \sqrt{Var[f_T]}$ - standard deviation, $N(-y) = \frac{\alpha}{2}$, where N is the cumulative distribution function of the standardised normal distribution.

For the critical values t_1, t_2 the acceptance interval of the test statistic f_T is the interval $[t_1, t_2]$. The verification of the hypothesis of randomness consists of calculating the value of the statistic and checking whether it lies in the acceptance interval. If not then the generator is rejected at the given level of significance α.

4. A DISCUSSION OF THE RESULTS

Statistical testing undertaken during this work was directed at sequences produced by linear congruential generators based on least primitive roots of prime numbers. The work presented here is a continuation of our earlier publication [3]. There we used the three types of tests mentioned: FIPS 140-1 tests, Feldman's spectral test and Maurer's test. Quantitative results of these tests are given in that publication.

In this paper we consider only Maurer's test and its capability to detect simple deviations from randomness in binary sequences. As the source of binary sequences we used Lehmer's congruential generator with a 64 bit prime modulus possessing a least primitive root of 2, 3, 5, 6 or 7. The sequence produced by this generator was subjected to a non-linear mixing transformation (of the type described above). After the application of this transformation the sequence was statistically indistinguishable from a random sequence. The FIPS 140-1 tests, Feldman's test and Maurer's test all accepted these sequences as random. Next, we divided the sequences into 64 bit blocks and within each block forced some bits to depend functionally on the others, obtaining once more statistically defective sequences. Then we investigated the sensitivity of Maurer's test to various types of defects. A full discussion of the results is not possible within the limits of this paper. We will present some interesting partial results.

A sample of 16 Mbytes was used. Within each 64 bit block we made one bit depend linearly on all the remaining bits. Then, for various values (from 8 to 16 bits) of the test window used in Maurer's test we checked how quickly the test will reject the sequence as non-random. By 'how quickly' we mean the first multiple of the test window after which the test statistic will assume a value unacceptable for a random sequence. Below we show representative results for the case when bit number 36 was linearly dependent on the remaining bits in each 64 bit block. The symbol f_T refers to the value of the test statistic while L refers to the length in bits of the test window. The diagrams below illustrate some typical results.

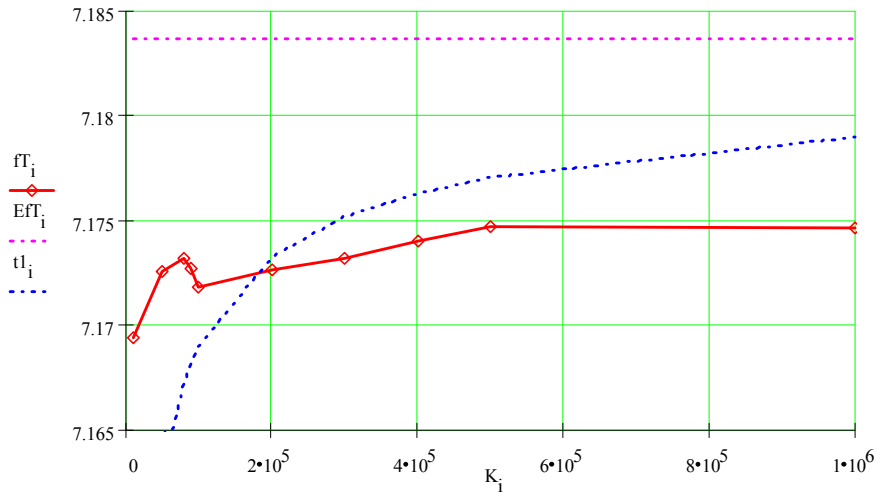

Fig. 5: Diagram of the dependence of Maurer's test statistic on the number of test steps K for block length L=8.

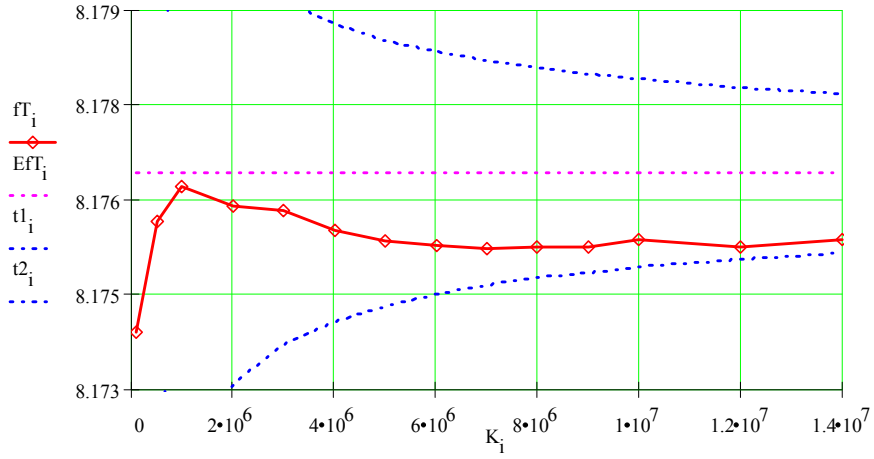

Fig. 6: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=9.

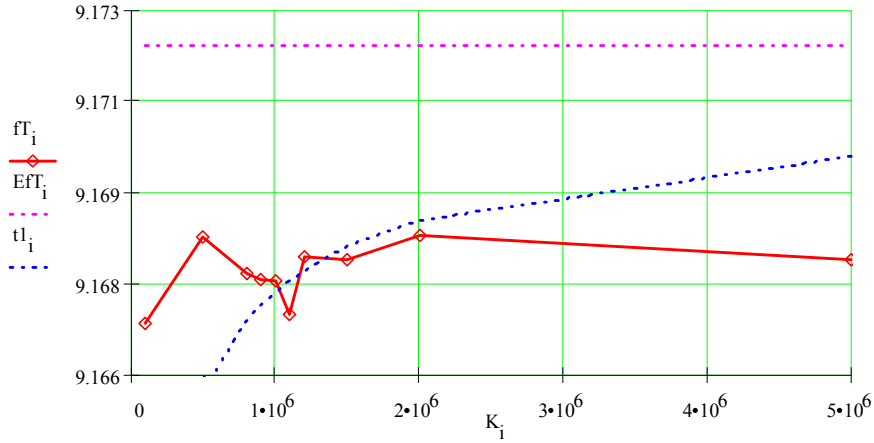

Fig. 7: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=10.

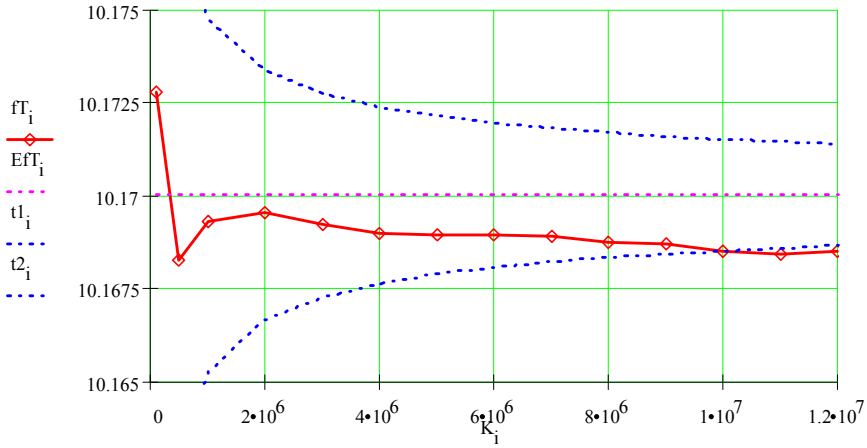

Fig. 8: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=11.

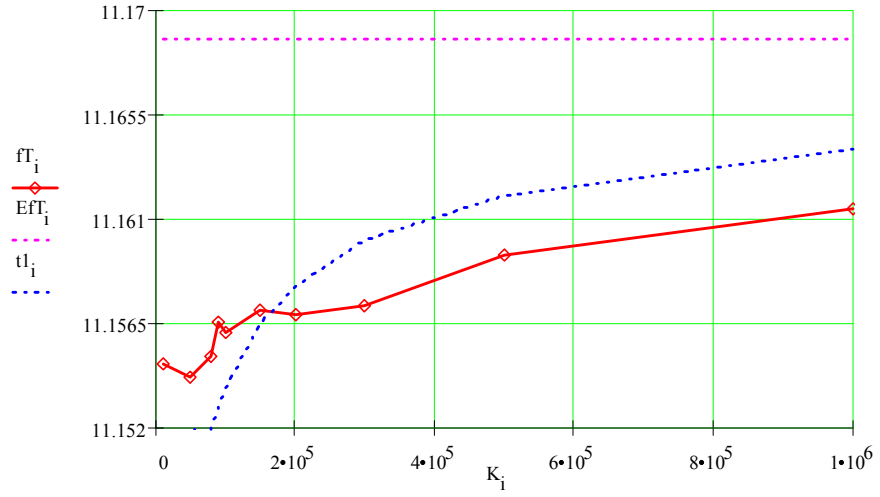

Fig. 9: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=12.

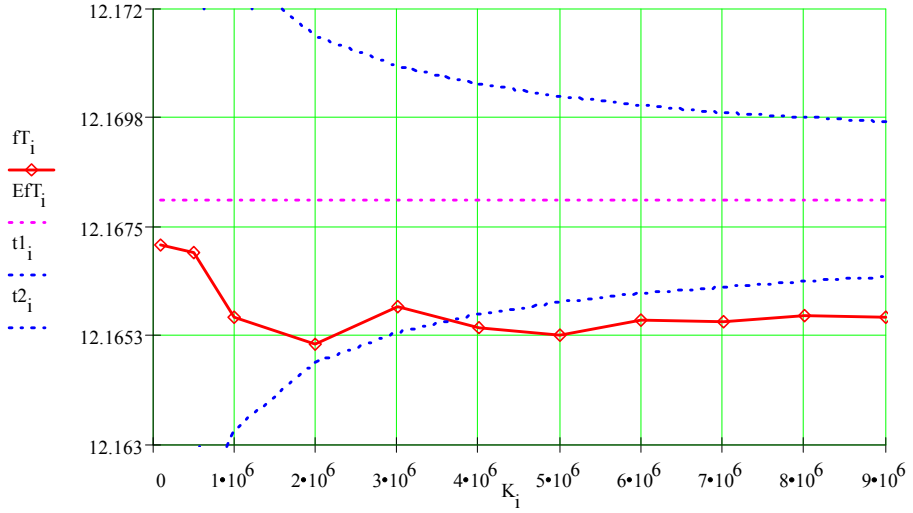

Fig. 10: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=13.

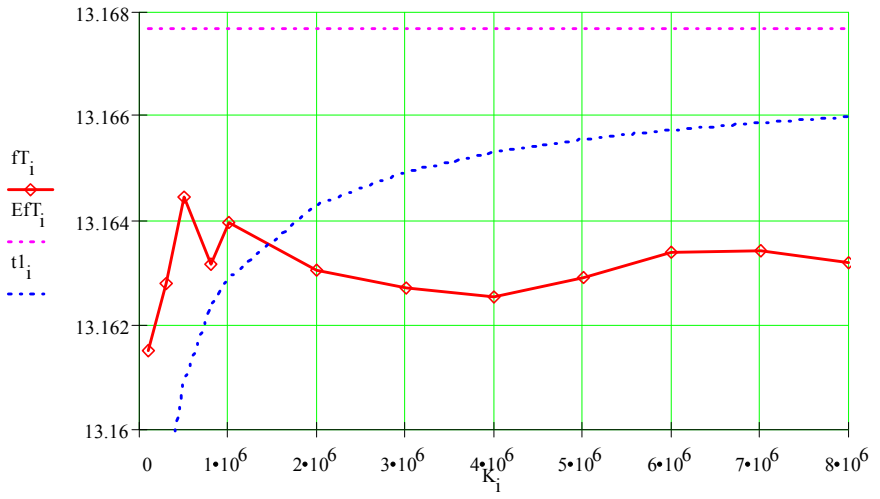

Fig. 11: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=14.

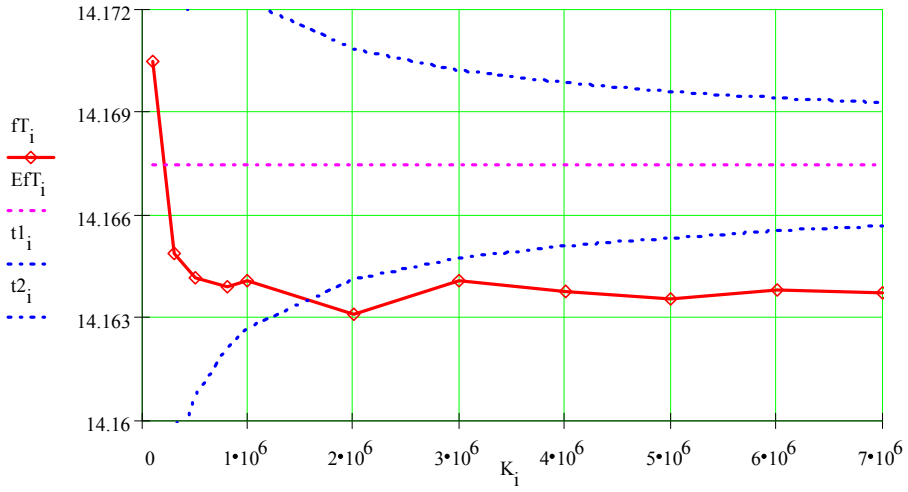

Fig. 12: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=15.

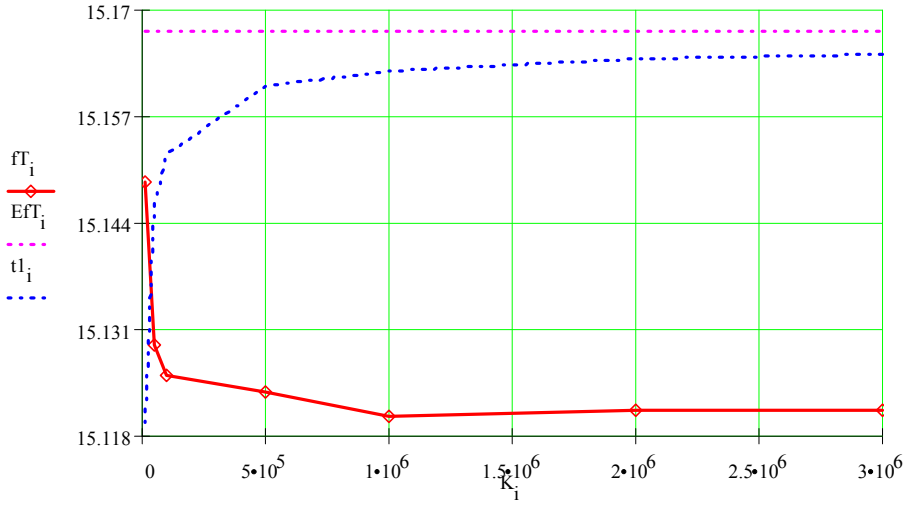

Fig. 13: Diagram of the dependence of Maurer's test statistic on the number of test steps *K* for block length *L*=16.

The diagrams lead to the conclusion that when the distance between two successive modified bits and the length of the test window have common factors the defect is detected earlier.

REFERENCES

1. FIPS PUB 140-1, *Federal Information Processing Standards Publication, Security Requirements for Cryptographic Modules.*
2. F.A. Feldman: *Fast Spectral Tests for Measuring Nonrandomness and the DES*. Crypto'87, pp.242-254.
3. K. Górski, A. Paszkiewicz, A. Zugaj, Z. Kotulski, J. Szczepański, *Properties of sequences generated by least primitive roots of prime numbers*, Proceedings of XIV National Telecommunications Symposium KST'98, t. B, pp. 143-151 (in Polish).
4. D.E. Knuth: *The Art of Computer Programming, v.2, Seminumerical Algorithms*, Addison-Wesley , 1998.
5. U.M. Maurer: *A Universal Statistical est for Random Bit Generators*, Eurocrypt 1996.
6. A. Paszkiewicz, *Experimental investigations of least primitive roots in finite fields GF(p) for primes p smaller than 500 million*, Procs. X National Telecommunications Symp. KST'94, t. B, pp. 72-81 (in Polish).
7. A. Paszkiewicz, *Least primitive roots in finite fields. A practical verification of some hypotheses,* Proceedings X National Telecommunications Symp. KST'95, t. B, pp. 255-263, (in Polish).
8. A. Paszkiewicz, *Computer aided investigations of least primitive roots in finite fields GF(p) for prime numbers smaller than 1000 million*. Inf. WIŁ 15/1995, WIŁ, (in Polish).
9. R. Rueppel, *Analysis and design of stream ciphers*, Springer-Verlag, Berlin, Heidelberg 1986.