A SIMPLE METHOD TO GENERATE PSEUDORANDOM SEQUENCES

Ciprian Răcuciu[1], Nicolae Jula[1], Ştefan Dochițoiu[2]
[1]Academia Tehnică Militară, Bucureşti, România
[2]Universitatea *Hyperion*, Bucharest, Romania
ciprian.racuciu@gmail.com, nicolae.jula@gmail.com, cdochitoiu@rdslink.ro

Abstract. *This paper aims generation pseudorandom sequences using binary representation of rational numbers. Cryptography is one of the areas using such sequences.*

Keywords: *pseudorandom sequences, period length, generator of a finite cyclic group, randomness tests.*

I. Introduction

Although the construction process presented in this paper for pseudorandom sequences is valid for all integer $b \geq 2$ as a base, we set $b = 2$. This case is about exclusively used in cryptography, on the one hand because it provides the bit-level control . On the other hand in this case the encryption and decryption operations are perfectly symetric and require the minim operations number.

Many cryptographic systems use such sequences and different systems require appropriate qualities. [5].

EXAMPLE: ***stream* cypher**

Given a random bit sequence $C = (c_1, c_2, ..., c_n)$, the message, converted into a number, binary represented, $M = (m_1, m_2, ..., m_k)$, $k < n$, *is encrypted.* That is, it's changed into the encrypted mesage:

$$E = (e_1, e_2, ..., e_k); e_i = m_i + c_i$$

where the adding is made modulo 2. The encrypted message ***E*** is transmited. The receiver can recover the message *M*, by *decrypting* operation which is the same as te encripting:

$$D = (d_1, d_2, ..., d_k); d_i = e_i + c_i = (m_i + c_i) + c_i = m_i$$

The bit sequence $C = (c_1, c_2, ..., c_n)$ must be known by the sender and the receiver too has to know it. On the other hand, for system safety is necessary that this sequence, the cipher key, not to be known or deductible by any intruder, an entity that may intercepts the message. The sequence randomness can eliminate any of its regularly that would allow the intruder to infer it.

There is a wide range of methods for obtaining random string ***C***, some of them use more or less the string ***M***, the message itself

This paper considers the string ***C*** obtained independently of message ***M***. The method is highly efficent and offer a wide key space.

I I. The length of the binary digit sequence period of a rational number

Let ***a*** be a rational number 0 < ***a*** <*1.* The following recurrence relations get the binary digits sequence $d_1\, d_2\, ...\, d_n$ of the number ***a*** :

$$a_0 = a; \forall i > 1\ d_i = \begin{cases} 1\ if\ 2a_{i-1} \geq 1 \\ 0\ if\ 2a_{i-1} < 1 \end{cases}; a_i = 2a_{i-1} - d_i$$

In the binary representation of the subunit number a, namely: $a = 0.\, d_1\, d_2...$ we can make distinction between the unperiodical part having h digits and the periodical part having k digits:

$$a = 0.d_1 d_2 ... d_h (d_{h+1} d_{h+2} ... d_{h+k})$$

If the length k of the period is large enough we can say that the string of binary digits obtained by the previous recurrence relations is a pseudorandom one.

Let u and v the binary represented integers: $u = d_1\, d_2\, ...\, d_h$; $v = d_{h+1} d_{h+2} ... d_{h+k}$. The number a can be expressed in terms of u and v as follows:

$$a = \frac{u}{2^h} + \frac{v}{2^h} \cdot \left(\frac{1}{2^k} + \frac{1}{2^{2k}} + \frac{1}{2^{3k}} + ... \right) = \frac{u}{2^h} + \frac{v}{2^h} \cdot \frac{1}{2^k} \cdot \frac{2^k}{2^k - 1} = \frac{u}{2^h} + \frac{v}{2^h} \cdot \frac{1}{2^h - 1} \qquad (*)$$

The folowing theorem evaluates the length k of the period.

THEOREM

Let m and n be relative prime natural numbers, $0 < m < n$, the last an odd one. Using the above notations,
if $a = m/n$ then:

$$2^k = 1 (\mathrm{mod}\, n)$$

PROOF

Using (*) we obtain:

$$\frac{m}{n} = a = \frac{u}{2^h} + \frac{v}{2^h} \cdot \frac{1}{2^k - 1} \Rightarrow m \cdot 2^h \left(2^k - 1\right) = n \cdot \left[u \cdot \left(2^k - 1\right) + v\right]$$

Taking account that m and n are relative prime numbers and n is an odd one, the last number, n, divides $2^k - 1$, that is, $2^k = 1(\mathrm{mod}\, n)$. Q.E.D.

We are interested to find a numerous set of values of n for which the length k of the period has the largest ratio k/n.

REMARKS

A. Let U_n be the multiplicative group of the ring Z/nZ. Its elements are the classes modulo n represented by the numbers which are prime to n. Their number, the order of the group U_n, is the Euler phi- function $j(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_r}\right)$ [2] where $p_1, p_2, ..., p_r$ are the prime divisors of the number n. If n is an odd number the class represented by the number 2 belongs to U_n. Let $\omega(n)$ be the order of this class in the group U_n, i.e., $\omega(n)$ is the least power of 2 which is congruent to 1 modulo n.

The above Theorem claims that **the length k of the period of the rational number m/n is divisible by $\omega(n)$, therefore we have $k \geq \omega(n)$.**

B. According to Lagrange Theorem[2], the order $\omega(n)$ is a divisor of $\varphi(n)$. In order to maximize the ratio k/n we have to find the numbers n for which $\omega(n) = \varphi(n)$. That happens only when the group U_n is a cyclic one [3].

It is known that the group U_n is cyclic if and only if the number n satisfies one of the folowing three situations[4]:

1). $n = p^r$ where p is an odd prime number and r is a natural one, $r \geq 1$.
2). $n = 2p^r$ with p and r as above.
3). $n = 4$ and $n = 2$.

Because n must be an odd number we have to consider only $n = p^r$. It is important to consider separately the case $r = 1$.

C₁. The case $n = p$ = a great prime number.

In this case $\varphi(n) = \varphi(p) = p - 1$is the order of the cyclic group U_p. The order $\omega(p)$ may be equal to the order of U_p if and only if the class represented by the number 2 is a generator. It is known that this happens if and only if $2^{\frac{p-1}{q_i}} \neq 1(\bmod p)$ for each q_i prime divisors of $p - 1$.

In order to verify the last condition one need to know the prime divisors of the number $p - 1$. It is not known a polynomial algorithm for integer factorisation so that for large numbers p finding the prime divisors of $p - 1$ can't be made. On the other hand, if these divisors are known, the condition $2^{\frac{p-1}{q_i}} \neq 1(\bmod p)$ can be tested using modular exponential algorithm which is polynomial. In this case the ratio k/n is almost equal to 1.

C₂. The case $n = p^r$.

In this case $\varphi(n) = n(1 - 1/p)$ and the difference $n - \varphi(n) = n/p$ is considerable. But from computational complexity standpoint may be considered as having the same size order.

On the other hand it is easy to find the prime factors of $\varphi(n) = p^{r-1}(p - 1)$: one of them is p and the others are the prime factors of $p - 1$. The last factors can be easy obtained if p is not very large. (The number $n = p^r$ can be large enough using the exponent r).

As a result this case represent the best source for the values of the numbers n having the period length k of the rational number $a = m/n$ as large as we want.

III. The cryptographic safety.

This section analizes the cryptographic safety conditions mentioned in [5] for the the binary representation sequences of some rational numbers $a = m/n$; $n = p^r$.

A. Imput versus output sequence length.

Every pseudorandom bit generator uses an input bit sequence and outputs another, the first having i and the other o bits. The input sequence must be much shorter then the second: $i << o$.

In the above section the inputs are the great numbers n such that the obtained bit sequence (the period) length is almost n (they have the same size order from computer complexity standpoint).

As a result, $i = \log_2 n$ and $o = n$, and $\log_2 n << n$.

B. The key-space size.

For cryptographic saphety the key-space must large enough to preclude intruders the exhaustive search of the key.

This condition is satisfied, there are a lot of key: almost all number $n = p^r$ with p a prim number and r a natural one.

C. Pseudorandomniss.

This condition means that the generated n sequences can't be recognized in the entire set of all n-sequences. There are a lot of statistic test which brings out some regularities making the sequence to be not random. All these tests are only necessary but not sufficient conditions for the sequence to be random.

IV. Conclusions

1) This bit sequence generator satisfy the base conditions in order to be a pseudorandom one, so it can be important for cryptographic uses, especially the stream cypher discussed in [1].
2) The generation algorithm is simple enough to be considered as highly efficient, which offer a remarcable flexibility.
3) The cryptographic safety can be considerably improved by often changing the key. This can be do using performant key managemement, taking account that the sequence is easy to obtain.

V. References

1. Ciprian RĂCUCIU, Cristina DĂSCĂLESCU, Marinela NIDELEA Stephen DOCHIȚOIU *Building highly primitive polynomials with coefficients as classes modulo 2*
2. S. LANG, *Algebra,* Addison-Wesley, 1964
3. Neal KOBLITZ, *A Course in Number Theory and Cryptography,* Springer Verlag, 1988
4. Neal KOBLITZ, *Algebraic Aspects of Cryptography,* Springer Verlag, 1999
5. A. Menezes, P. van OORSHOT, S. Vanstone, *Handbook of Applied Cryptography,* CRC Press, 1996