第 25 卷第 6 期　　四川理工学院学报(自然科学版)　　Vol. 25　No. 6
2012 年 12 月　　Journal of Sichuan University of Science & Engineering(Natural Science Edition)　　Dec. 2012
文章编号:1673-1549(2012)06-0049-05　　DOI:10.3969/j.issn.1673-1549.2012.06.012

一种改进的伪随机数生成算法及随机性分析

刘　倩,范安东

(成都理工大学管理科学学院,成都　610059)

摘　要: 通过对信息安全系统中常用的伪随机数生成算法及其随机性进行分析,比较了不同的伪随机数生成算法实现的优缺点。在此基础上提出了一种将 Visual C++ 中伪随机数生成机制与线性反馈移位寄存器结合起来产生随机数的改进方案,通过数值模拟和对比分析表明,改进方案具有较好的计算性能和随机性。

关键词: 伪随机数;随机性;线性反馈移位寄存器;随机性检验

中图分类号: TP309.7　　**文献标志码:** A

引　言

随机数在密码算法、安全协议、数字水印、密码芯片等信息安全应用中[1],具有重要的作用。对于伪随机序列的理论与应用研究,一直是一个热门课题[2]。国外相关领域的科学家提出了多种伪随机数生成方法:包括线性同余法,非线性同余法,Fibonacci,移位寄存器序列发生器[3],进位加—借位减发生器法[4],复合素数发生器[5],基于混沌映射产生随机数的方法[6]和组合发生器[7]。在国内,我国的学术界也对伪随机数生成算法进行了较为广泛深入的研究,开展了量子随机数发生器的研究工作[8],研究了一种新方法-超素数法[9],提出了一种利用无限数无限不循环特性来产生随机数的模型[10]和具有本原多项式的 90/150 加性细胞自动机的随机数发生器模型[11]。伪随机序列在伪随机测距、导航、遥控遥测、扩频通信、多址通信、分离多径、数据加乱、信号同步、误码测试、线性系统测量、天线方向图测量、各种噪声源,均有一定的应用[12]。事实上,随机数不是一个孤立的问题,它与数学、通信、计算机等领域都有密切关系,具有很强的理论背景和应用背景,因此基于伪随机数的生成算法和相关应用具有理论上和应用上的双重价值,具有重要的研究意义。本文分析了典型的伪随机数生成算法的随机性,并比较不同的伪随机数生成算法实现的优缺点,提出了一种改进的线性反馈移位寄存器,通过编程模拟实现了该伪随机数生成算法,并使用 NIST 的随机性检测工具对改进的线性反馈移位寄存器产生的随机数进行了随机性测试,该算法通过了其中的六项测试,表明它具有了良好的安全性,可以应用于信息安全领域。

1　典型的伪随机数生成算法

1.1　Visual C++ 中伪随机数生成机制

Visual C++ 6.0 中通过使用函数 srand 和 rand 可以得到 0 ~ RAND_MAX(0x7fff) 之间的数。

如果伪随机数序列的初始种子相同,那么计算出来的伪随机数序列也是完全相同的。想要解决这个问题,就需要在每次产生随机数序列前,使用 time 函数值作为初始种子数,通过调用函数 srand((unsigned) time(NULL)),首先指定产生不同的种子,其次,由于两次调用 rand 函数的时间是不同的,这样计算出来的随机数序列就会不相同了,从而能够保证随机性。但是,rand 函数产生的随机数数量是有限的,最大为 32767。

收稿日期: 2012-10-26
基金项目: 四川省应用基础计划项目(2012JY0033)
作者简介: 刘　倩(1988-),女,陕西西安人,硕士生,主要从事计算数论与快速密码算法方面的研究,(E-mail) lqmova@msn.cn

1.2 平方取中法

平方取中法是由冯·诺伊曼在 1946 年提出的。它的实现算法是:首先任取一个非 0 的 $2s$ 位数,用它中间的 s 位数作为生成的随机数的第一个元素,然后用该数做平方运算,得到一个新的 $2s$ 位数,再取中间 s 位作为伪随机数的第二个元素,依次进行迭代。

平方取中法的递推公式为:

$$\begin{cases} X_{n+1} = (X_n^2/10^s)\ (\mathrm{mod}10^{2s}) \\ R_n = X_n/10^{2s} \end{cases},(n = 1,2,\cdots) \quad (1)$$

产生的伪随机数序列 $\{R_n\}$。

1.3 ANSI X9.17 伪随机数生成算法

算法描述:

输入:DT_i 表示 64 位的当前时间,V_i 表示初始种子,K_1 和 K_2 是 3DES[13] 算法的两个 64 位的密钥。

$S1$: $T_i = 3\mathrm{DES}(DT_i, K_1K_2)$;

$S2$: $R_i = 3\mathrm{DES}(T_i \oplus V_i, K_1K_2)$;

$S3$: $V_{i+1} = 3\mathrm{DES}(T_i \oplus R_i, K_1K_2)$,$V_{i+1}$ 为新的种子,用来初始化下一个随机数;

输出:伪随机数序列为 $\{R_i\}$。

2 改进方案设计

通过对 Visual C++ 中伪随机数生成机制的分析,可以看到它产生的随机数数量是有限的,最大为 32767,所以一般不直接利用其产生随机数序列。而通过线性反馈移位寄存器产生随机数需要指定一个初始种子,进而通过反馈函数来产生随机数序列。我们可以将这两种算法结合起来,首先利用 rand 函数来产生初始种子,进而通过线性反馈移位寄存器来产生随机数序列。

2.1 线性反馈移位寄存器

如图 1 所示,a_i 表示二值(0,1)存储单元,a_i 的个数 n 表示反馈移位寄存器的级。某时刻,这些级的内容构成反馈移位寄存器的状态,用 $(a_1,a_2,\cdots,a_n)$ 表示[14]。若反馈函数 $f(a_1,a_2,\cdots,a_n) = k_na_1 \oplus k_{n-1}a_2 \oplus \cdots \oplus k_1a_n$,$0 \leqslant k_i \leqslant 1(i = 1,2,\cdots,n)$,称为 n 级线性反馈移位寄存器。

引入一个多项式:

$$g(x) = k_nx^n + k_{n-1}x^{n-1} + \cdots + k_1x + k_0 \quad (2)$$

其中,$k_n = k_0 = 1$,$g(x)$ 称为线性反馈移位寄存器的连接多项式。

在算法设计过程中,为了使所产生的随机数达到 1024 比特而不出现周期性重复,因此采用反馈函数为 $x^{10} + x^9 + x^6 + x + 1$,保证随机数周期为 $2^{10} - 1$。

令 $a_i(t)$ 表示 t 时刻第 i 级的内容,$a_i(t+1)$ 表示 $a_i(t)$ 的下一时刻内容,则有:

移位:

$$a_i(t+1) = a_{i+1}(t)\ (i = 1,2,\cdots,n-1) \quad (3)$$

反馈:

$$a_n(t+1) = k_na_1(t) \oplus k_{n-1}a_2(t) \oplus \cdots \oplus k_1a_n(t) \quad (4)$$

最终得到随机序列 $(a_1,a_2,\cdots,a_n,\cdots)$。

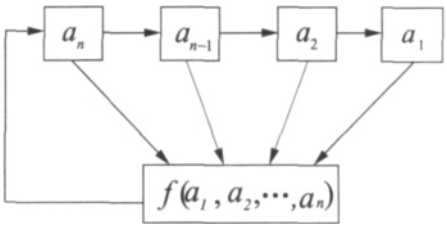

图 1 反馈移位寄存器

2.2 算法设计

算法描述:

S1:利用 srand(time(0)) 产生种子;

S2:利用 1 + rand() % 32767 产生下一个随机数 RandNumber;

S3:将 RandNumber 转化为二进制序列 BRandNumber 作为线性反馈移位寄存器的初始种子;

S4:输入反馈函数 $x^{10} + x^9 + x^6 + x + 1$;

```
S5: for(j=0; j<10; j++)
{
BRandNumber[j] = BRandNumber[j+1];
}
BRandNumber[j] = (BRandNumber[9] + BRandNumber[8] + BRandNumber[5] + BRandNumber[0]) % 2;
```

S6:输出随机数序列。

3 算法的比较实现

3.1 平方取中法的实现

此算法计算起来非常简单,很容易利用计算机编程实现,但是具有较为明显的缺点:容易产生循环现象,很难保证有足够长的周期;当初值 X_1 取值不当时,容易出现退化现象(以后的随机数为同一个常数或者为零)。从图 2 可以看出当 $s = 2$,$X_1 = 6100$,$X_2 = 0.2100\ 00$,$X_3 = 0.4100\ 00$,$X_4 = 0.8100\ 00$,$X_5 = 0.6100\ 00$,$X_6 = 0.2100\ 00$,$X_7 = 0.4100\ 00$,$X_8 = 0.8100\ 00$,$X_9 = 0.6100\ 00$,从此之后开始出现循环,可见初值的选择对随机数的质量影响很大。

3.2 ANSI X9.17 伪随机数发生器的实现

算法实现的实例效果,如图 3 所示。

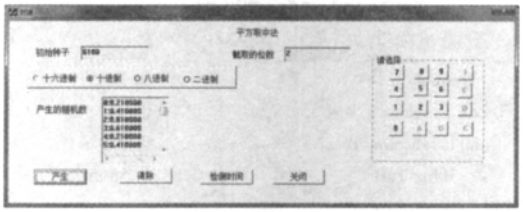

图 2　平方取中法的实现

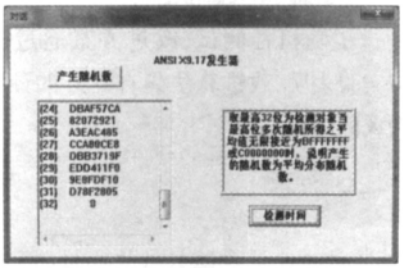

图 3　ANSI X9. 17 伪随机数发生器的实现

该算法的实现过程中采用了 112 比特长的密钥和 9 个 DES 加密,安全强度是增强了,但是相应的运行效率比较低下。

很明显,若攻击者不知道 K_1 和 K_2,对于已知输入,选择输入和重放输入来说,攻击者不知道发生器的输出。

若攻击者知道 K_1 和 K_2,可以得到两次连续的输出的 R_i 和 R_{i+1}。假设系统时间戳中有 20 比特对攻击者是未知的,攻击者可以对公式(5)和(6)进行攻击得到中间状态 V_{i+1}:

$$V_{i+1} = 3DES^{-1}(R_{i+1}, K_1K_2) \oplus T_{i+1} \quad (5)$$

$$V_{i+1} = 3DES(R_i \oplus T_i, K_1K_2) \quad (6)$$

由于攻击者知道 K_1 和 K_2,就可以对 T_i 和 T_{i+1} 进行 2^{20} 次穷举攻击,一共使用 2^{21} 次攻击,可以分别得到两个 V_{i+1} 的状态表,正确的中间状态必然在这两个状态表中。

假如攻击者知道 V_i,以及 R_{i+1} 的某个函数,则由 $R_i = 3\mathrm{DES}(T_i \oplus V_i, K_1K_2)$,通过假设的对 T_i 的 2^{20} 次穷举攻击,可以得到 R_i,再由 $V_{i+1} = 3DES(T_i \oplus R_i, K_1K_2)$,可以很容易得到 V_{i+1}。

从上面的分析可知,ANSI X9. 17 伪随机数生成算法的安全性主要取决于 DES 的密钥 K_1,K_2 和系统时间 DT_i 以及初始种子 V_0。

3.3　改进的线性反馈移位寄存器的实现

本文改进算法的实现情况,如图 4 所示。

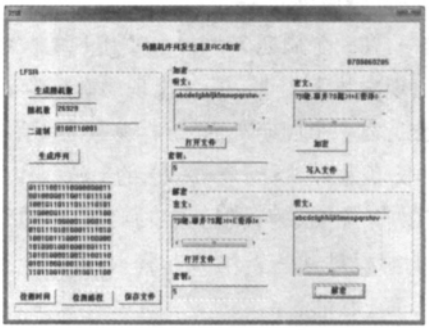

图 4　改进的线性反馈移位寄存器及 RC4 加密的实现

算法首先利用系统时间作为参数,通过 srand 函数产生初始种子,增加了初始种子的随机性,然后利用 rand 函数计算一个随机种子,再通过反馈和移位操作来产生周期为 1023 的随机数序列,周期性得到保证,并且运行速度快。

4　算法性能分析

4.1　计算性能分析

本次实验的实验环境: 处理器为 32 位,主频为 2. 27 GHz; 工具为 Microsoft Visual C + + 6. 0 的编译器。通过对三种伪随机数生成算法产生 1024 比特随机数所用的时间进行多次测试,得到的平均值如表 1 所示。可以看出,产生相同比特的伪随机数,改进的线性反馈移位寄存器和平方取中法的效率更高,ANSI X9. 17 伪随机数生成算法效率相对较低,而平方取中法产生的随机数周期较短,因此优先考虑挑选改进的线性反馈移位寄存器来产生伪随机数。

表 1　各种伪随机数生成算法的性能分析

算法名称	算法执行效率
平方取中法	68. 27bit/ms
ANSI X9. 17 随机数发生器	21. 79bit/ms
改进的线性反馈移位寄存器	68. 27bit/ms

4.2　随机性分析

在实际的应用过程中,理想的随机数序列应该无限长,即具有无限长的周期。但对于利用计算机的伪随机数发生器而言,由于计算机自身的有限状态空间,使它所产生的随机数序列必然会出现周期性重复。因此,我们希望伪随机数发生器所产生的随机数序列的周期应该越长越好。

线性反馈移位寄存器(LFSR)的输出序列的性质完全是由反馈函数决定的,也即完全由连接多项式来决定。如果能判断其连接多项式为本原多项式[15],就可以得到较长周期的随机序列。一个 n 级 LFSR 最多有 2^n 个内部状态,其周期最大为 $2^n - 1$,当输出序列的周期能达到 $2^n - 1$ 时,则称这样的序列为 m - 序列。因此分析 LFSR 的随机性就是分析 m - 序列的随机性。

4.2.1　m - 序列的随机性

为了分析 m - 序列的随机性,Golomb 提出了序列的

伪随机性的三条假设[16]:

假设一:在一个周期 T 内,0 和 1 出现的次数接近相等,即 0 出现次数为 $2^{n-1}-1$,1 出现的次数为 2^{n-1}。

假设二:在一个周期 T 内,游程总数为 2^{n-1},当 $n>2$ 时,其中长度为 i $(1\leqslant i\leqslant n-2)$ 的 1 游程和 0 游程的数目各有 2^{n-i-2} 个,长为 i 的游程占游程总数的 2^{-i}。长度为 $n-1$ 的游程只有一个,为 0 游程;长度为 n 的游程也只有一个,为 1 游程。

假设三:自相关函数 R(t) 为一常数。

在文献[16]中,这三条假设均得到了成立的证明。

通过对 n 级 LFSR 的随机性进行分析,可以看到:m-序列满足 Golomb 三条随机性假设,具有良好的随机特性,是一个预先可以确定,并且可以重复实现的确定性序列,在现实的密码体系结构中都是以 m-序列的伪随机性为基础进行设计和实现的,但是在应用 n 级 LFSR 来产生伪随机序列时,为了获得最大的周期 2^n-1,需要寻找 n 阶的 GF(2) 上的本原多项式。随着级数的增大,产生 n 阶本原多项式就会越来越困难。通常情况下,确定一个 n 阶的多项式是本原多项式需要进行 2^n-1 次因式分解。另外,一个 n 阶本原多项式的项数越多,算法就越复杂,就越安全,实现起来就会越复杂。

4.2.2 随机性检测

根据 NIST 800-22[17] 提出的随机性测试方法,对改进的线性反馈移位寄存器产生的随机数序列进行了比较全面的随机性测试。在随机序列通过某项检验后会产生一个 $P-value$ 值,当 $P-value$ 值大于显著性水平 0.01 时,则认为通过了该项检验;否则未通过。

随机性测试结果,如表 2 所示。从表 2 可以看出,改进的线性反馈移位寄存器产生的随机数序列通过了频率测试(Frequency Test)、块内频率测试(Block Frequency Test)、累积和测试(Cumulative Sum Test)、游程测试(Runs Test)、二进制矩阵测试(Rank Test)和离散傅里叶变换测试(Dft Test)。因此,本文研究的改进的线性反馈移位寄存器产生的随机数序列已经有了足够的随机性。

表 2 NIST 800-22 测试随机序列的情况

测试项目	$P-value$
Frequency Test	0.498579
Block Frequency Test	0.882242
Cumulative Sum Test	0.718241
Runs Test	0.585504
Rank Test	0.950389
Dft Test	0.450552

5 结束语

本文通过对典型伪随机数生成算法进行分析,将 Visual C++中伪随机数生成机制与线性反馈移位寄存器结合起来,提出了一种改进的随机数生成算法。利用 VC++编程模拟实现了平方取中法、ANSI X9.17 伪随机数生产算法和改进的线性反馈移位寄存器三种算法,对比分析了算法的计算性能和随机性,对改进算法产生的随机数进行了随机性测试,改进方案通过了六项测试。分析与实验表明,改进算法具有良好的计算效率和满足安全需求的随机特性。

参考文献:

[1] 张咏. 随机数发生器和随机数性能检测研究[D]. 成都:电子科技大学,2006.

[2] 白恩健. 伪随机序列构造及其随机性分析研究[D]. 西安:西安电子科技大学,2004.

[3] Tausworthe R C. Random numbers generated by linear recurrence modulo two[J]. Mathematics of Computation, 1965,19(90):201-209.

[4] Marsaglia G, Zaman A. A new class of random numbers generators[J]. Ann, Appl. Prob., 1991,1(3):462-480.

[5] Hass A. The multiple prime random number generator[J]. ACM Transactions on Mathematical Software, 1987, 13(4):268-381.

[6] An H Z. A note on chaotic maps and time series[C]// Robinson P M, Rosenblatt M. Athens Conference on Applied Probability and Time Series: Time Series Analysis in Memory of E. J. Hannan. Berlin: Springer-Verlag, 1996: 15-26.

[7] Deng L Y, George E O. Generation of uniform variate from several nearly uniformly distributed variables[J]. Communications in Statistics Simulation and Computation, 1990, 19(1):145-154.

[8] 冯凯锋,吕述望,刘振华. 量子密钥分发系统和量子随机数发生器[D]. 北京:中国科学院研究生院,2002.

[9] 李世刚,刘辉,陈标华. 素数的一个特殊性质及其用于伪随机数生成的方法[J]. 北京化工大学学报:自然科学版,2003,30(3):1-4.

[10] 冯艳. 一种产生随机数新方法的研究与实现[D]. 北京:北京工业大学,2004.

[11] 张传武. 细胞自动机在密码学中的应用研究[D]. 成都:电子科技大学,2003.

[12] 肖国镇,梁传甲,王育民. 伪随机序列及其应用[M]. 北京: 国防工业出版社,1985.
[13] 张焕国,王张宜. 密码学导引[M]. 武汉: 武汉大学出版社,2010.
[14] 胡向东,魏琴芳. 应用密码学[M]. 北京: 电子工业出版社,2011.
[15] 刘晓阳. 伪随机序列中本原多项式的研究[D]. 辽宁: 辽宁科技大学,2008.
[16] 覃中平,张焕国. 信息安全数学基础[M]. 北京: 清华大学出版社,2006.
[17] NIST. FIPS PUBS 800-22,Statistical test suite for random and pseudorandom number generators for cryptographic applications [S]. [s.l.]: NIST,2000.

Improved Pseudo-random Number Generation Algorithm and Randomness Analysis

LIU Qian, *FAN An-dong*
(College of Management Sciences, Chengdu University of Technology, Chengdu 610059, China)

Abstract: Aiming at studying pseudo-random number generation algorithm, which is commonly used in information security system, the advantages and disadvantages of different pseudo-random number generation algorithms are compared and the typical pseudo-randomness of random number generation algorithm is analyzed. Combined the visual C++ in a pseudo-random number generator mechanism with a linear feedback shift register, an improved algorithm is proposed to generate random numbers. The numerical simulation and comparative analysis results show that the improvement algorithm has a good computing performance and randomness.

Key words: pseudo-random number; randomness; linear feedback shift register; random testing