DOI:10.13413/j.cnki.jdxblxb.2011.05.001

基于鼠标移动轨迹的真随机数产生方法

胡 亮,裴 莹,初剑峰,袁 巍,王文博,樊 丽,刘建男
(吉林大学 计算机科学与技术学院,长春 130012)

摘要: 给出一种运用计算机中鼠标移动轨迹这一随机事件产生真随机数的算法,与传统基于鼠标的随机数生成算法相比,在采样原始数据过程中,该方法得到的数据随机性更高,与其他用物理过程产生真随机数的算法相比,不用接额外的电路和设备,成本较低,从而解决了产生真随机数开销过大的问题. 通过对产生的随机数进行均匀性和独立性检验,结果表明该方法所产生的随机数具有较好的统计性质;同时测试该方法的程序执行时间结果表明,该方法时间开销较小.

关键词: 随机数;伪随机数;鼠标移动轨迹
中图分类号: TP391 **文献标志码:** A **文章编号:** 1671-5489(2011)05-0890-05

True Random Number Generator Based on Mouse Trajectory

HU Liang, PEI Ying, CHU Jian-feng, YUAN Wei, WANG Wen-bo, FAN Li, LIU Jian-nan
(*College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*)

Abstract: This article implements the algorithm of using a random event-the mouse movement tracks to produce real random numbers. Compared with the usual random-number-producing algorithm based on mouse, this one enjoys a higher randomicity in the process of sampling and getting original data. In addition, it does not need the extra circuit or facilities that other physical processes, the algorithms producing real random number. As a result, its costs will be reduced and the problem of a high expanse to produce real random numbers will be solved. The result of testing the uniformity and independence of the random numbers generated shows that the numbers produced by this algorithm have a fine statistical property. Moreover, the result of testing its program execution time proves that its time expending is very short.

Key words: random number; pseudo-random number; mouse moving path

随机数在网络环境中应用广泛,如服务器之间或服务器与路由器在建立TCP连接时生成的TCP序列号、对称加密算法中会话双方会话密钥的产生[1]以及RSA公钥加密算法中密钥的产生等. 此外,在一些安全协议和数字签名算法中也用到了随机数. 这些应用对随机数的产生提出了3个不同的要求[2]:1) 序列是随机的,没有规律可循. 随机序列应具有如下统计特性:① 分布一致性:序列中的随机数分布是一致的,即出现频率大约相等;② 独立性:序列中任何数不能由其他数推导出;2) 序列是不可预测的. 3) 要重复产生该序列是不可能的.

满足1),2)要求的随机数称为伪随机数,而真随机数需要满足1)~3). 例如投掷硬币,每次投掷都可能产生两种结果:正面向上或向下,它们的概率均为1/2. 投掷结果没有规律可循,不能准确预测下一次投掷结果,两次投掷产生完全相同随机序列的可能性也极小. 如果把正面向上的结果赋值为1,

收稿日期:2010-11-22.

作者简介:胡 亮(1968—),男,汉族,博士,教授,博士生导师,从事网络与信息安全的研究,E-mail:hul@jlu.edu.cn.
通讯作者:初剑峰(1978—),男,汉族,博士,讲师,从事网络与信息安全的研究,E-mail:chujf@jlu.edu.cn.

基金项目:国家自然科学基金(批准号:60873235;60473099).

正面向下的结果赋值为 0，产生的 0，1 序列即为真随机序列. 显然，用投掷硬币方法产生的随机数是绝对随机的，而在计算机上几乎无法实现. 通常在计算机上使用的随机数产生方法(e. g. rand( ))，所产生的序列都是伪随机的. 如令函数 rand( ) 的种子固定不变，则其每次产生的序列都相同.

一般真随机数发生器(TRNG) 都以一种称为熵源的不确定性源为基础，如放射性衰变、电子设备的热噪音、宇宙射线的出发时间等[3]，而伪随机数发生器(PRNG) 则利用种子，采用某种固定的算法产生类似于随机数序列的方法[4]. 显然，真随机数发生器的安全性比伪随机数发生器的安全性高很多，当安全级别较高时一般都采用 TRNG. 但要产生真随机数一般需要在计算机上连接额外辅助的电路和仪器[5-6]. 操作繁琐且成本也较高. 而伪随机数的产生相对成本较低，但安全性也低，只有通过统计随机性检验，才可以应用到网络安全体系中.

1　预备知识

密码学意义上安全的随机数比一般的随机数要求更高，一般要求通过统计随机性测验，通常运用均匀性检验和独立性检验进行随机统计性测验[5].

1.1　均匀性检验

均匀性检验[7] 用于检验由某个发生器产生的随机序列是否均匀地分布在 [0,1] 区间上，即检验经验频率与理论频率的差异是否显著. 常用的方法有 χ^2 检验、K-S 检验和序列检验，本文采用 χ^2 检验，过程如下:

1) 用随机数发生器产生随机序列$\{r_n\}$;

2) 将 [0,1] 区间分成 10 等份，统计$\{r_n\}$落在第 i 个小区间的个数 $n_i(i=1,2,\cdots,10)$;

3) 计算统计量 $V=\dfrac{10}{n}\sum_{i=1}^{10}\left(n_i-\dfrac{n}{10}\right)^2$ 的值，V 渐近服从$\chi^2(9)$ 分布;

4) 给定显著水平 α，查χ^2 表得到临界值 λ. 如果 $V\leqslant\lambda$，则检验通过，即认为产生的随机数均匀分布在 [0,1] 区间内. 否则，检验不通过.

1.2　独立性检验

独立性检验[7] 主要检验随机序列 $r_1,r_2,\cdots,r_n$ 间的统计相关性是否显著. 试验中选取相关系数检验Ⅱ的方法. 过程如下:

1) 用随机数发生器产生随机序列$\{r_n\}$;

2) 计算 $C_j=\dfrac{1}{n}\sum_{i=1}^{n}r_ir_k(i,j=1,2,\cdots,n)$，其中 $k=(i+j)\bmod n$;

3) 计算 $T_j=\dfrac{C_j-1/4}{\sqrt{13/144n}}(j=1,2,\cdots,n)$，$T_j$ 渐近服从标准正态分布;

4) 给定显著水平 α，查标准正态分布表得 λ，如果 $|T_j|\leqslant\lambda(j=1,2,\cdots,n)$，则检验通过，即认为产生的随机数之间统计相关性显著. 否则，检验不通过.

2　满足正态分布的伪随机数产生算法

高斯分布也称为正态分布或常态分布. 对于随机变量 X，其高斯分布记为 $N(\mu,\sigma^2)$，其中 μ 和 σ^2 为分布参数，分别为高斯分布的期望和方差[8]. 特别地，当 $\mu=0$，$\sigma^2=1$ 时，X 的分布为标准正态分布. 高斯分布概率密度函数为: $f(x)=(1/\sqrt{2\pi\sigma})\,e^{-(x-\mu)^2/(2\sigma^2)}(\sigma>0)$.

本文用二次产生法产生近似正态分布的随机数[9]，过程如下:

1) 用线性同余法产生随机数: $u(i)=b\cdot u(i-1)(\bmod M1)$，$k(i)=4\cdot u(i)+5$;

2) 以 $k(i)$ 做乘子，用混合同余法产生一个随机数: $v_i(j)=k(i)\cdot v_i(j-1)+c(\bmod M2)$;

3) 用近似抽样法变换成随机数: $x(i)=\sqrt{12/l}\cdot\left[\sum_{j=1}^{l}v_i(j)/M2-l/2\right]$.

$x(i)$ 即为第 i 次产生的随机数. 选定初值 $u(0)$，b，$M1$，$v_l(0)$，c 和 $M2$，且当 $i>1$ 时，取

$v_i(0)=v_{i-1}(l)$，利用上述方法可递推产生 $x(i)$，易证 $x(i)$ 在理论上近似服从标准正态分布. 因此，该方法产生的随机数只能称为伪随机的. 若将这种方法产生的伪随机数应用到网络安全中，必须通过统计随机性检验.

3 基于鼠标运行轨迹的真随机数产生算法模型

3.1 算法原理

由于鼠标运行轨迹具有随机性和不确定性的特点，即同一个人两次对鼠标的操作轨迹完全相同的可能性很小. 但同一个人两次对鼠标的操作具有相似性，为了避免这种相似性，必须对初始鼠标轨迹做预处理，基于鼠标移动轨迹的性质和预处理过程，本文把鼠标运行轨迹作为获得随机数的熵源[10].

3.2 算法整体过程描述

假设需要的二元随机序列长度为 n.

1) 根据需要采样鼠标坐标数据. 在鼠标移动时采样鼠标坐标(x,y)，将一系列坐标值保存在二维数组 $R[N][2]$ 中，$R[i][0]$ 保存横坐标，$R[i][1]$ 保存纵坐标$(i=0,1,\cdots,N-1)$.

2) 对已经获得的坐标进行再处理，使其消除相似性. ① 用 rand() 函数产生 N 个随机数; ② 将产生的 N 个随机数，映射到 $[0,N-1]$ 区间内，存放在数组 $S[N]$ 中; ③ 去掉 $S[N]$ 中的重复数据，产生 N 个没有重复的并且在 $[0,N-1]$ 区间内的数据; ④ 将 $R[i][0]$ 按照 $S[N]$ 重新排序$(i=0,1,\cdots,N-1)$.

3) 对随机排好序的坐标值进行操作，从而获得一系列的随机数.

3.3 捕获原始数据

每当鼠标移动时就捕获鼠标的坐标值，直到捕获到的鼠标坐标数量达到要求为止.

3.4 对初始鼠标坐标值的再处理

为了消除两次鼠标运行轨迹的相似性，需要对初始的坐标值进行再处理. 本文主要是将得到的横坐标序列随机打乱顺序，即产生 N 个在 $[0,1]$ 区间内没有重复的随机整数序列，存放在数组 $S[N]$ 中，并将产生的初始横坐标序列按照数组 $S[N]$ 中的值改变顺序，即初始横坐标序列中第一个位置的数被第 $S[0]$ 位置的数代替，第二个位置的数被第 $S[1]$ 位置的数代替，以此类推.

3.4.1 产生 $[0,N-1]$ 区间内随机数序列的方法 产生随机数步骤如下:

1) 用当前系统的时间作种子，使用 rand() 函数产生任意 N 个随机数，存放在中间数组 $S[N]$ 中. 此时的数据在 0x7fff 范围内，还不能满足要求.

2) 将产生的 N 个随机数对应到 $[0,N-1]$ 区间内. 不用 rand()%N 直接产生 $[0,N-1]$ 区间内的数，假设有两个数 10 和 22，要产生 0~3 之间的数. $10\%3=22\%3=1$，破坏了数据的随机性. 本文采用的方法用 $s[i]=\left\lfloor\frac{s[i]}{\max(s[N])+1}\times N+0.5\right\rfloor(i=0,1,2,\cdots,N-1)$ 表示.

此外，如果经过四舍五入计算后 $S[i]=N$，则令 $S[i]=N-1$，从而获得了 N 个在 $[0,N-1]$ 区间内的数. 虽然 $[0,N-1]$ 区间内的数已经产生，但不能保证他们完全没有重复，需要去掉重复的数字，最后得到 $S[N]$.

3.4.2 对初始横坐标序列按 $S[N]$ 中的数据重新排序 本文按已经产生的 $S[N]$ 中的数据对存放在 $R[N][0]$ 中的初始横坐标序列进行重新排序，即 $R[0][0]$ 用 $R[S[0]][0]$ 代替，$R[1][0]$ 用 $R[S[1]][0]$ 代替，$\cdots$，$R[N-1][0]$ 用 $R[S[N-1]][0]$ 代替.

至此，已完成了对捕获坐标序列的再处理，下面将坐标中的二维数据转换为一维数据，以获得一系列的随机数.

3.5 将二维坐标值转换为一维数据

传统的方法只是简单的将横纵坐标相加，可能破坏数据的随机性，如 $a(3,5)$ 和 $b(5,3)$ 是两个完全不同的点，但横纵坐标相加后相等. 本文采用的方法如图 1 所示.

本文以 $a(x_1,y_1)$，$b(x_2,y_2)$，$c(x_3,y_3)$ 这 3 个点为例. 每两个点之间都有一个夹角. 设 a 和 b 之间的夹角为 θ_1，b 和 c 之间的夹角为 θ_2，a 和 c 之间的夹角为 θ_3. 只要计算出 3 个夹角的弧度值即可得到

3 个数值，从而就把二维数据转换成一维数据，避免了简单的坐标相加可能发生的问题. 假设有 N 个点 1,2,…,N，只需要计算 1,2 之间的夹角，2,3 之间的夹角，…，N,1 之间的夹角即可. 因此，当 $x_i \neq x_{i+1}$ 时，

$$\theta_i = \arctan \frac{|y_{i+1} - y_i|}{|x_{i+1} - x_i|} \quad (i = 0,1,2,\cdots,N-2),$$

$$\theta_{N-1} = \arctan \frac{|y_{N-1} - y_1|}{|x_{N-1} - x_1|};$$

当 $x_i = x_{i+1}$ 时，$\theta_i = \pi/2$. 从而得到了一系列的浮点随机数.

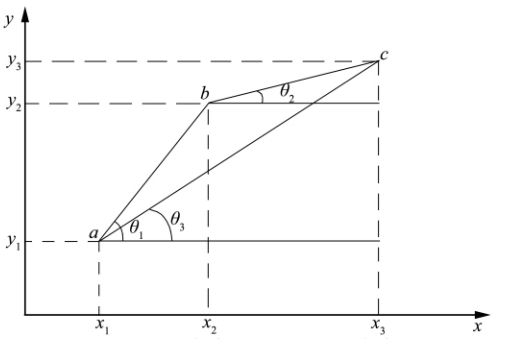

图 1　二维数据转化为一维数据

Fig. 1　Schematic diagram of two-dimensional data into one-dimensional data

4　实验数据的随机性分析

均匀性和独立性分析数据列于表 1. 由表 1 可见，用本文方法产生的数据满足均匀性和独立性分布，因此，具有科学性.

表 1　均匀性和独立性分析

Table 1　Analysis of uniformity and independence

实验数据	均匀性检测		独立性检测		
	V	临界值 $\lambda(\alpha=0.05)$	C_j	T_j	临界值 $\lambda(\alpha=0.05)$
0.540 420	3	16.919	0.339 645	0.334 290	0.519 9
0.620 250			0.313 614	0.446 846	
0.010 128			0.304 542	0.311 816	
0.285 348			0.292 125	0.126 999	
0.076 772			0.283 700	0.501 596	
0.141 870			0.261 050	0.164 472	
0.278 300			0.248 541	−0.021 713	
0.539 293			0.225 778	0.360 522	
0.211 093			0.207 797	−0.128 150	
0.321 751			0.208 497	−0.117 741	
0.558 599			0.208 950	−0.210 988	
0.643 501			0.208 497	0.217 741	
0.785 398			0.207 797	−0.428 150	
0.678 300			0.225 778	−0.360 523	
0.876 058			0.248 541	−0.021 713 8	
0.927 295			0.261 050	0.164 472	
0.896 055			0.283 700	0.501 595	
0.900 753			0.292 125	0.126 999	
0.463 648			0.304 542	0.311 816	
0.482 244			0.313 614	0.446 846	

下面比较基于鼠标轨迹方法和二次产生法的执行效率. 本文采用在 C 语言中嵌入汇编的方法，即通过获取 edx,eax 寄存器的值精确获得所需的 CPU 周期数，结果列于表 2. 其中运行程序的计算机内存为 1.0 G，CPU 频率为 2.2 GHz. 由表 2 可见，在产生 5,10,100 个原始数据时，基于鼠标轨迹的方法所需的 CPU 周期数是 10 亿数量级的，而二次产生法在产生 5 个原始数据和 10 个原始数据时是 10 万数量级的，并且随着原始数量的增加，基于鼠标轨迹的方法 CPU 周期数增加幅度不大，而二次产生法的增加幅度较大. 当差距最大时，前者只比后者高出 10^4 个数量级，在程序执行时间上，两者差别较小.

两种方法(即代表真随机数产生方法和伪随机数产生方法)的比较列于表 3.

表2 程序执行所需的CPU周期数

Table 2 CPU cycles to perform required procedures

运行参数	基于鼠标轨迹的方法	二次产生法
产生5个原始数据时程序运行所需的CPU周期数	1 448 872 171	535 768
产生10个原始数据时程序运行所需的CPU周期数	1 595 392 903	976 031
产生100个原始数据时程序运行所需的CPU周期数	1 981 220 031	9 484 380

表3 两种方法的比较

Table 3 Comparison of two methods

比较参数	基于鼠标轨迹的方法	二次产生法
优点	产生的数据具有真随机性，数据没有周期；方法简单，不需要介入额外的电路，开销小；程序执行速度快，接近伪随机数算法；随机数性质较好，安全性较高	方法简单，不受环境影响；程序执行效率高；生成的随机数性质较好，满足正态分布规律
应用场景	相互认证或会话密钥的生成应用中	只要算法通过统计随机性检测，即可应用到密码学中；否则，序列的安全性较低，不能应用于密码学

参 考 文 献

[1] YUAN Wei-zhong, XIE Jun-yuan, XIE Li, et al. Analysis and Application of the Techniques of Random Number in the Network Security [J]. Computer Engineering, 2001, 27(6): 116-118. (袁卫忠，谢俊元，谢立，等. 网络安全中随机数技术分析与应用 [J]. 计算机工程，2001，27(6)：116-118.)

[2] LIN Guo-shun, HUANG Ti-yun. Statistical Analysis for Algorithms of Simulation Random Numbers [J]. Mathematical Statistics and Management, 2000, 19(2): 30-34. (林国顺，黄梯云. 模拟随机数统计性质比较 [J]. 数理统计与管理，2000，19(2)：30-34.)

[3] WANG Lai, LIU Song-qiang. A True Random Alumber Generator by Sampling Thermal Noise Source [J]. Nuclear Electronics & Detection Technology, 1998, 18(6): 452-455. (王莱，刘松强. 真随机数发生器的设计和实现 [J]. 核电子学与探测技术，1998，18(6)：452-455.)

[4] LI You-ping. Capture of Random Number and Its Application Based on Web Techniques [J]. Microelectronics and Computer, 2005, 22(11): 108-110. (李幼平. 随机数的捕获及其基于Web的应用 [J]. 微电子学与计算机，2005，22(11)：108-110.)

[5] XIN Qian, ZENG Xiao-yang, ZHANG Guo-quan, et al. Design of Truly Random Number Generator Based on Thermal Noise of Resistor [J]. Microelectronics and Computer, 2004, 21(7): 143-146. (辛茜，曾晓洋，张国权，等. 基于电阻热噪声的真随机数发生器设计 [J]. 微电子学与计算机，2004，21(7)：143-146.)

[6] ZHANG Xiao-feng, BAI Guo-qiang, CHEN Hong-yi. True Random Number Generator for Network Security Co-Processor [J]. Computer Enginering, 2009, 35(10): 229-231. (张晓峰，白国强，陈弘毅. 应用于网络安全协处理器的真随机数产生器 [J]. 计算机工程，2009，35(10)：229-231.)

[7] LUAN Zhong-lan, LÜ Qiang. On Statical Properties of PRNG and Its Application [J]. Computer Applications and Software, 2010, 27(10): 168-170. (栾忠兰，吕强. 伪随机数发生器的统计性质检验及其应用 [J]. 计算机应用与软件，2010，27(10)：168-170.)

[8] WU Yu-xin, YU Song-yu. An Algorithm for Generalized Gaussian Random Variable Creation [J]. Journal of Data Acquisitien & Processing, 1999, 14(4): 443-446. (吴宇新，余松煜. 广义高斯分布随机变量生成算法 [J]. 数据采集与处理，1999，14(4)：443-446.)

[9] GUO Cheng-an. Generation and Statistical Test of Computer Simulated Normal Random Numbers [J]. Journal of Dalian University of Technology, 1990, 30(4): 473-477. (郭成安. 计算机模拟正态随机数的产生与统计检验 [J]. 大连理工大学学报，1990，30(4)：473-477.)

[10] ZHOU Qing, HU Yue, LIAO Xiao-feng. True Random Number Generators Based on Mouse Movement and Chaos System [J]. Journal of Physics, 2008, 57(9): 5413-5418. (周庆，胡月，廖晓峰. 基于鼠标轨迹和混沌系统的真随机数产生器研究 [J]. 物理学报，2008，57(9)：5413-5418.)