初等数论及其应用

第五章 原根及其应用

5.5 伪随机数

背景介绍

- 在前面章节，已经多次涉及随机数. 事实上，随机数广泛应用于计算机模拟、计算机算法性能测试、随机算法、概率素性测试、密钥生成及密码协议等方面.

- 所谓随机数，其实是一个特殊的数列，该数列中的每项以同等的概率选取，这种选取不依赖于数列中的其他项. 因此，说一个具体的数(如50)是随机的是没有意义的，尽管它可以是某个随机数序列中的某一项.

- 自然界中的很多现象，例如抛掷硬币、抛掷骰子、转轮、洗牌等都展示出随机性，我们可以利用这些现象产生一些短周期的随机数.

背景介绍

- 利用计算机中的某些事件,如定时中断或时钟等也可以产生随机数,但这种机械方法由于计算机硬件故障经常会使随机数偏斜,并且这些数没法重复产生.

- 1946年,冯.诺依曼首次给出了使用计算机程序产生随机数的方法,但事实证明这种方法产生的数也并非是随机的.一个普遍的观点是,绝对随机的随机数只是一种理想的随机数,计算机不会产生绝对随机的随机数,它只能生成相对随机的随机数,即**伪随机数**.因此,伪随机数并不是假随机数,这里的“伪”是有规律的意思,就是产生的伪随机数既是随机的又是有规律的.

伪随机数生成方法

- 本节介绍使用初等数论知识的两种伪随机数生成方法.

- 第一种方法是莱梅于1949年发明的**线性同余方法**.

  设整数n, a, c, x_0满足下列条件:

  $2 \le a < n,\ 0 \le c < n,\ 0 \le x_0 \le n,$

  那么伪随机数序列由下面的公式给出:

  $x_{i+1} = (ax_i + c)\ mod\ n.$

  这里n, a, c, x_0分别被称为**模数**, **倍数**, **增量**和**种子**.

  上面的公式也称作**线性同余生成器**,

  它给出0到$n-1$的整数序列$x_0, x_1, \cdots, x_i, \cdots$.

  如果我们想生成$[0,1]$内的伪随机数, 那么只需用n去除序列$x_0, x_1, \cdots, x_i, \cdots$中的每项即可.

  如果把这个序列中的每项都模2, 则得到一个0, 1伪随机数序列.

伪随机数生成方法

- 容易从$x_{i+1} = (ax_i + c)\ mod\ n$中得到$\{x_i\}$的通项:

$$x_i = \left(a^i x_0 + \frac{a^i - 1}{a - 1} \cdot c\right)\ mod\ n$$

作为一个练习, 同学们可使用数学归纳法证明上式的确是$\{x_i\}$的通项.

- **例5.5.1** 设$n = 9,\ a = 7, c = 4, x_0 = 3,$
- 则由线性同余生成器可产生序列$3, 7, 8, 6, 1, 2, 0, 4, 5, 3, \cdots$.

显然, 这个序列在重复出现$3, 7, 8, 6, 1, 2, 0, 4, 5$之前有$9(n = 9)$个不同的数.

伪随机数生成方法

- **例5.5.2** 设$n=12, a=3, c=4, x_0=5$, 则由线性同余生成器计算有

$$x_1 = (3\times 5+4)\ mod\ 12 = 7,$$
$$x_2 = (3\times 7+4)\ mod\ 12 = 1,$$
$$x_3 = (3\times 1+4)\ mod\ 12 = 7,$$
$$\vdots$$

因此该线性同余生成器产生序列$5, 7, 1, 7, 1, 7, 1, \cdots$.

值得注意的是, 这个序列在重复出现$7, 1$之前仅有$3(<n)$个不同的数.

伪随机数生成方法

- 在上面两个例子中可以发现，线性同余生成器产生的伪随机数具有周期性.

- 这是因为$x_i \in \mathbb{Z}_n$仅有n个可能的取值，于是必然存在整数$j > i$，使得$x_j = x_i$，从而由线性同余生成器$[\boldsymbol{x_{i+1} = (ax_i + c) \bmod n}]$计算知，对任意正整数$k$都有$x_{j+k} = x_{i+k}$，所以序列在至多$n$个项后必定重复.

伪随机数生成方法

- **定义5.5.1** 设$\{x_i\}$是一个整数序列, 如果T是满足下列条件的最小正整数: 存在整数i_0使得所有$i \geq i_0$, 都有$x_{i+T} = x_i$, 则称T为$\{x_i\}$的**周期**.

- 例如, 例5.5.1中序列的周期是9, 而例5.5.2中序列的周期是2. 从应用的角度, 希望线性同余生成器产生的伪随机数序列的周期T尽可能大, 但由前面的分析, 显然$T \leq n$. 那么, 什么条件可以保证$T = n$呢?

- 下面的定理回答了这个问题.

伪随机数生成方法

- **定理5.5.1** 设线性同余生成器$x_{i+1} = (ax_i + c)\ mod\ n$产生的伪随机数序列的周期为$T$, 则$T = n$当且仅当下述条件成立:

  (1)$(c, n) = 1$.

  (2) $a \equiv 1(mod\ p)$, **这里p跑遍n的所有素因数.**

  (3) 当$4|n$时, $a \equiv 1\ (mod\ 4)$.

- 上述定理的证明相当复杂,有兴趣的同学可参见下面的参考文献.

  [D. Knuth(高德纳). The Art of Computer Programming, vol. 2: Seminumerical Algorithms (3rd Edition). Reading, MA: Addison-Wesley. 1997]

伪随机数生成方法

- 尽管线性同余生成器被广泛用于产生伪随机数，但它也存在一些不足. 例如，当已知该伪随机数序列的一些项后，我们有可能计算出$x_{i+1} = (ax_i + c) \, mod \, n$中的一些参数$a, c$和$n$，于是可以得到该序列的所有项. 如果这样的伪随机数应用于密码协议中，必然影响密码体制的安全.

- 为了增强伪随机数的安全性，1986年L. Blum等人发明了**平方伪随机数生成器**，该生成器产生的序列$x_0, x_1 \ldots, x_i, \ldots$具有这样的性质：从任意项$x_i$在合理的时间内无法计算出$x_{i-1}$. 这里，从$x_i$计算$x_{i-1}$的困难性是基于大整数分解的困难性假设的.

伪随机数生成方法

- **平方伪随机数生成器**

  设n是正整数, $x_0 \in \mathbb{Z}$, 那么**平方伪随机数生成器**由下面的公式给出: $x_{i+1} = {x_i}^2 \ mod \ n,$

  也即: $x_i = {x_0}^{2^i} \ mod \ n,$

  这里n和x_0分别是**模数**和**种子**.

- **例5.5.3** 设$n = 209, x_0 = 6$, 那么平方伪随机数生成器产生下面的序列:

  $6, 36, 42, 92, 104, 157, 196, 169, 137, 168, 9, 81, 82, 36, 42, \ldots,$

  它的周期为$T = 12.$

伪随机数生成方法

- **定理 5.5.2** 设$ord_n x_0 = 2^t s$, 其中s是奇数, t是非负整数, 则平方伪随机数生成器$x_{i+1} = x_i{}^2 \ mod \ n$生成的伪随机数序列的周期是$ord_s 2$.

- **证明:** 设平方伪随机数生成器生成的伪随机数序列的周期为l.

  欲证$l = ord_s 2$, **只需证明$ord_s 2 | l$和$l | ord_s 2$**即可.

  我们先证明$ord_s 2 | l$, 假设$x_{i+l} = x_i$,

  则由$x_i = x_0^{2^i} \ mod \ n$得$x_0^{2^{i+l}} \equiv x_0^{2^i} \ (mod \ n)$.

  于是由定理5.1.2知 $2^{i+l} \equiv 2^i \ (mod \ ord_n x_0)$,

  即$2^{i+l} \equiv 2^i (mod \ 2^t s)$.

伪随机数生成方法

- **证明(续):**因为 $2^t | 2^{i+l} - 2^i = 2^i(2^l - 1)$, 所以$i \geq t$,

  于是由$2^{i+l} \equiv 2^i (mod\ 2^t s)$得$2^{i+l-t} \equiv 2^{i-t} (mod\ s)$.

  再次使用定理5.1.2, 有$i + l - t \equiv i - t (mod\ ord_s 2)$,

  所以$l \equiv 0 (mod\ ord_s 2)$, 即$ord_s 2 | l$.

  下证$l | ord_s 2$,

  也即证明:若整数$j \geq i \geq t$满足, $j \equiv i\ (mod\ ord_s 2)$, 则$x_i = x_j$.

  因为$j \equiv i\ (mod\ ord_s 2)$, 所以由定理5.1.2知$2^j \equiv 2^i (mod\ s)$.

  又因为$j \geq i \geq t$, 所以$2^j \equiv 2^i (mod\ 2^t)$.

  因为$(s, 2^t) = 1$, 所以$2^j \equiv 2^i (mod\ 2^t s)$, 即$2^j \equiv 2^i (mod\ ord_n x_0)$.

  再次使用定理5.1.2得$x_0^{2^j} \equiv x_0^{2^i} (mod\ n)$.

  即$x_j \equiv x_i (mod\ n)$, 因此$x_j = x_i$. 这就完成了定理的证明.

伪随机数生成方法

- **定理 5.5.2** 设$ord_n x_0 = 2^t s$, 其中s是奇数, t是非负整数, 则平方伪随机数生成器$x_{i+1} = x_i{}^2 \ mod \ n$生成的伪随机数序列的周期是$ord_s 2$.

- 例如, 在例5.5.3中$n = 209, x_0 = 6$, 计算知$ord_n x_0 = ord_{209} 6 = 90 = 2 \times 45$, 所以平方伪随机数生成器用模数209和种子6生成的伪随机数序列的周期为$T = ord_s 2 = ord_{45} 2 = 12$, 这与我们在例5.5.3中观察到的一致.

伪随机数生成方法

- 为了满足密码学安全性要求，在利用平方伪随机数生成器生成伪随机数时，通常假设模数$n = pq$，其中p, q是两个满足$p \equiv q \equiv 3 (mod\ 4)$的大素数.这样，如果假设模平方根问题是难解的(与Shamir的零知识证明协议同样的假设)，那么在不知道p, q的情况下，由$x_{i+1} = {x_i}^2\ mod\ n$将很难从$x_{i+1}$得到$x_i$.

- 由于$x_{i+1} = {x_i}^2\ mod\ n$，因此可能存在不同$x_i$的对应于同一个$x_{i+1}$，但下面的命题表明，**如果要求$x_i$是模$n$的二次剩余，那么$x_{i+1}$可以确定惟一的$x_i$.**

伪随机数生成方法

- **命题5.1.1** 设a是模$n = pq$的二次剩余, 其中p, q是满足$p \equiv q \equiv 3(mod\ 4)$的素数, 则存在惟一的$x$, 使得

  (1) $x^2 \equiv a(mod\ n)$

  (2) x是模n的二次剩余.

- **证明:** 因为a是模$n = pq$的二次剩余, 所以同余方程$x^2 \equiv a(mod\ n)$有解.由4.5节的讨论知, $x^2 \equiv a(mod\ n)$有4个解, 并且, 若x_0是一个解, 则所有4个解可由下面的同余方程组给出:

$$\begin{cases} x \equiv x_0(mod\ p), \\ x \equiv x_0(mod\ q), \end{cases} \qquad \begin{cases} x \equiv x_0(mod\ p), \\ x \equiv -x_0(mod\ q), \end{cases}$$

$$\begin{cases} x \equiv -x_0(mod\ p), \\ x \equiv x_0(mod\ q), \end{cases} \qquad \begin{cases} x \equiv -x_0(mod\ p), \\ x \equiv x_0(mod\ q), \end{cases}$$

伪随机数生成方法

- **证明(续):** 这4个解产生勒让德符号,$\left(\frac{x}{p}\right)$和$\left(\frac{x}{q}\right)$的4种组合,即

  $\left\{\left(\frac{x_0}{p}\right),\left(\frac{x_0}{q}\right)\right\},\left\{\left(\frac{x_0}{p}\right),\left(\frac{-x_0}{q}\right)\right\},\left\{\left(\frac{-x_0}{p}\right),\left(\frac{x_0}{q}\right)\right\},\left\{\left(\frac{-x_0}{p}\right),\left(\frac{-x_0}{q}\right)\right\}.$

  因为$p \equiv q \equiv 3(mod\ 4)$,所以$\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1.$

  于是无论$\left(\frac{x_0}{p}\right)$和$\left(\frac{x_0}{q}\right)$的值是多少,4种可能的符号组合$\{1,1\},\{1,-1\},\{-1,1\},\{-1,-1\}$每个必须出现一次, **(Why?)**

  因此存在惟一的x,使得$\left(\frac{x}{p}\right) = 1$且$\left(\frac{x}{q}\right) = 1,$

  即$x^2 \equiv a(mod\ n)$的4个解中只有一个使得x是模n的二次剩余,

  故命题得证.

伪随机数生成方法

- 例如, 设$n = 142661 = 331 \times 431, a = 11535$, 这里331和431均是素数, 且$331 \equiv 431 \equiv 3(mod\ 4)$, 易验证11535 是模$n$的二次剩余. 使用4.5节的方法, 解$x^2 \equiv 11535(mod\ n)$可得$x = \pm 127060, \pm 58962(mod\ n)$, 容易验证这些解中仅127060是模$n$的二次剩余.

- 下面利用伪随机数和命题5.5.1构造公钥密码系统中的**一次一密(one-time pad)**加密方案. 一次一密是指在流密码中使用与消息长度等长的随机密钥, 密钥本身只使用一次.

随机密钥

- 设一次一密的明文, 密钥和密文都是0, 1数字串, 它们由下表给出:

明文	x_1	x_2	…	x_s
密钥	k_1	k_2	…	k_s
密文	y_1	y_2	…	y_s

- 其中$y_i = (x_i + k_i)\ mod\ 2, 1 \leq i \leq s.$

随机密钥

- 假设的Alice公钥$n = pq$, 其中p, q 是满足$p \equiv q \equiv 3(mod\ 4)$的大素数, Bob想发送信息$x_1\ x_2\ \ldots x_s$给Alice, 那么他们可以执行下面的协议:

  (1) Bob选取整数z_0, 用平方伪随机数生成器计算得到序列z_0, $z_1, \ldots, z_s, z_{s+1}$.

  (2) Bob进一步计算比特$k_i = z_i\ mod\ 2, 1 \le i \le s.$

  (3) Bob将随机比特流$k_1 k_2 \ldots k_s$作为密钥对明文$x_1\ x_2\ \ldots x_s$加密, 将密文$y_1\ y_2\ \ldots y_s$及z_{s+1}发送给Alice, 其中 $y_i = (x_i + k_i)\ mod\ 2, 1 \le i \le s.$

  (4) Alice使用z_{s+1}算出$z_s, z_{s-1}, \ldots, z_1$, 进而得到解密密钥$z_1$, $z_2, \ldots, z_s$和明文.

- 由命题5.5.1知, 上面第4步中可惟一确定$z_s, z_{s-1}, \ldots, z_1$. 该方案的安全性是**基于求解$x^2 \equiv a(mod\ n)$中的$x$等价于分解整数$n$**, 即**求模平方根与分解大整数是同样困难**的假设.