文章编号:1005－0523(2007)05－0148－03

t 半序在序列密码设计中的应用

左黎明

(华东交通大学 基础科学学院,江西 南昌 330013)

摘要:讨论了半序在序列密码设计中的应用问题,给出了生成伪随机序列的详细算法和 matlab 实现.

关 键 词:半序;序列密码;伪随机位生成器

中图分类号:O177 TP309 **文献标识码**:A

1 序列密码与伪随机数发生器

序列密码又称流密码.加密过程为首先将特定的种子密钥(对话密钥)注入到序列密码发生器,然后用序列密码发生器产生的伪随机比特序列与明文比特流进行逐位异或得到密文.使用序列密码加密时不受明文长度限制,只要产生的序列密码的周期足够大,可以加密任何长度的文件,非常灵活.与分组密码加密方法相比较,序列密码加密速度最快.因此,特别适合于大数据量、实时性要求较高的加密场合.

在密码系统中,其安全性在很大程度上依赖于是否能够随机地生成一些不可预测的数,这些不可预测的数包括 DES 加密算法中的密钥、RSA 加密算法中的素数 P、Q 以及 DSA 中的私钥等等.在这些情况下,要求产生的数必须足够大,并且是随机的,也就是说,产生任何特定的随机数的概率都必须足够小,以防止对手利用规律破译密码.

随机位生成器(random bit generator)是用来输出一系列独立随机的二进制位串.在实际应用中,有许多物理因素要考虑.在理想情况下,密码算法和协议中保密数需要通过随机位生成器来产生.但在有些情况下,要想保证安全地存取和传输一个非常大的随机位序列是不现实的,因此,往往用伪随机位生成器来代替随机位生成器.

伪随机位生成器(Pseudorandom bit generator,PRBG)采用确定性的算法,即输入长度 k 为的二进制随机序列,输出长度为 $l(l\gg k)$的二进制随机序列.PRBG 的输入称为种子数,输出称为伪随机序列.伪随机生成器其实并不全是随机的,它可能的输出只占其中的一小部分.它是把一个位数比较少的随机序列通过一定的方法进行扩展变为一个位数较长的随机序列,这样对方就不能有效地区别伪随机输出序列和真正的长度为 l 的随机序列.Ad－hoc 方法就是用来产生这样的伪随机序列.

传统我们构造序列密码所需要的伪随机序列都是基于数论和布尔代数的,常用的有 Brent 序列,钟控序列以及一些相关免疫函数.本文是文献[1]研究的继续,采用类似混沌系统迭代二进制编码的方法,另辟蹊径,利用半序空间的非线性迭代来构造伪随机位生成器,产生二进制伪随机序列.

2 基于半序关系的密码系统

在文献[1]中,我们引入了一种新的半序"$\leqslant_t$",我们称为 t 半序.在实 Hilbert 空间 H 中,取初始的某个 $x_0\in H$ 作为种子,选取一个非线性算子 A 作为迭代算子,建立一个迭代系统:$x_{n+1}=Ax_n=0,1,2,\cdots$ (1)

半序密码系统的思想是:将某个 $t\in H$ 和某个种子 $x_0\in H$ 作为密钥,利用(1)得到一个序列,若 $x_n\leqslant_t x_{n+1}$,则记录编码为"0";若 x_n 与 x_{n+1}不可比较,

收稿日期:2007－07－25

基金项目:江西省教育厅项目(赣教技字[2006]123 号)

作者简介:左黎明,男,1981 生,江西鹰潭人,助教,软件工程师,理学硕士,主要研究方向:信息安全,非线性系统分析.

跳过不记录;若 $x_n \geqslant_t x_{n+1}$,则记录编码为"1".通过这种方法,我们只要进行足够多次的迭代,我们就可以得到足够长的二进制序列.

在 R^n 中定义标准的内积,$\{e_i\}_{i=1}^n$ 为其单位向量,设 $x=\sum_{i=1}^n x_i e_i (x_i \geqslant 0)$,取 $t=\sum_{i=1}^n t_i e_i, (t_i \geqslant 1)$,显然 $<x,t> \geqslant 0$,因为 $t_i \geqslant 1$,所以:

$$<x,t>^2=(\sum_{i=1}^n x_i t_i)^2 \geqslant (\sum_{i=1}^n x_i)^2 \geqslant \sum_{i=1}^n x_i^2, <x,x> \leqslant <x,t>^2, \| x \| \leqslant <x,t>$$

故当取 $t=\sum_{i=1}^n t_i e_i, (t_i \geqslant 1)$,锥 $P=\{x \mid x=\sum_{i=1}^n x_i e_i, \geqslant 0\}$ 是一个 R_n 中的 P_t 锥.这样,我们只要限制算子 A 在锥 P_t 中进行迭代就行了,方便算法的建立与实现.

为了使得迭代具有足够高的混乱度,我们可以在建立非线性算子 A 中引入混沌系统,下面我们给出一个具体的例子.我们选取的空间为 R^8,设:

$x(n)=(x_{n,1}, x_{n,2}, x_{n,3}, x_{n,4}, x_{n,5}, x_{n,6}, x_{n,7}, x_{n,8}), x_{ni} \in R^+$.

建立以下非线性算子迭代:$x(n+1)=A(x(n))$,其中:

$$x_{n+1,1}=|p \cdot \sin^2(qx_{n,8}x_{n,1}-r)| \quad (2)$$

$$x_{n+1,2}=|p \cdot \sin^2(qx_{n,1}x_{n,2}-r)| \quad (3)$$

$$x_{n+1,3}=|p \cdot \sin^2(qx_{n,2}x_{n,3}-r)| \quad (4)$$

$$x_{n+1,4}=|p \cdot \sin^2(qx_{n,3}x_{n,4}-r)| \quad (5)$$

$$x_{n+1,5}=|p \cdot \sin^2(qx_{n,4}x_{n,5}-r)| \quad (6)$$

$$x_{n+1,6}=|p \cdot \sin^2(qx_{n,5}x_{n,6}-r)| \quad (7)$$

$$x_{n+1,7}=|p \cdot \sin^2(qx_{n,6}x_{n,7}-r)| \quad (8)$$

$$x_{n+1,8}=|p \cdot \sin^2(qx_{n,7}x_{n,8}-r)| \quad (9)$$

(其中 $p,q,r>0$,其中 q 一般取较大的数)

取这种形式的算子主要是考虑每个算子分量都采用混合光学双稳模型[2].选择好参数后在若干次迭代后将陷入混沌状态.半序控制密钥为:

$$t=(t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8), \text{其中 } t_i \geqslant 1, i=1,2,\cdots,8 \quad (10)$$

初始种子密钥为:

$$x(n)=(x_{0,1}, x_{0,2}, x_{0,3}, x_{0,4}, x_{0,5}, x_{0,6}, x_{0,7}, x_{0,8}) \in R^+ \quad (11)$$

初始迭代次数为 m(m 为整数).

3　算法的 MATLAB 实现

使用半序序列密码算法很简便,只要先将明文通过预处理加密过程得到明文二进制流,在与半序序列密码二进制流进行异或运算即可(如图 1 所示).解密过程只要与半序序列密码二进制流再进行异或运算.算法的关键是得到半序序列密码二进制流.

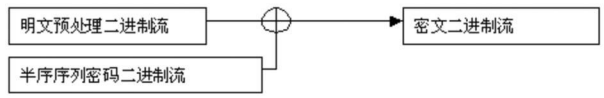

图 1　半序序列密码算法加密过程

根据这种思想,下面我们结合一个生成半序序列密码二进制流的 MATLAB 例子程序,来说明算法怎么建立和工作的:

```
% 程序 1:生成半序序列密码二进制流演示程序 1.0
t=[2.5,3.2,4,1.2,3,5,2.5,4];%半序控制密钥
q=2137;                          %参数 q
p=3.7;                           %参数 p
r=7;                             %参数 r
m=43;                            %初始迭代次数
MAXSIZE=512;                     %要产生的二进制序列长度
%定义初始种子密钥
x(1)=0.43;
x(2)=2.3;
x(3)=1.52;
x(4)=3.4;
x(5)=2.78;
x(6)=5.34;
x(7)=7.42;
x(8)=4.51;
y=[0,0,0,0,0,0,0,0];
z=[0,0,0,0,0,0,0,0];
k=1;
%进行初始迭代,进入混沌状态
for i=1:m
    y(1)=abs(p*sin(q*x(8)*x(1)-r)*sin(q*x(8)*x(1)-r));
    y(2)=abs(p*sin(q*x(1)*x(2)-r)*sin(q*x(1)*x(2)-r));
    y(3)=abs(p*sin(q*x(2)*x(3)-r)*sin(q*x(2)*x(3)-r));
    y(4)=abs(p*sin(q*x(3)*x(4)-r)*sin(q*x(3)*x(4)-r));
    y(5)=abs(p*sin(q*x(4)*x(5)-r)*sin(q*x(4)*x(5)-r));
    y(6)=abs(p*sin(q*x(5)*x(6)-r)*sin(q*x(5)*x(6)-r));
    y(7)=abs(p*sin(q*x(6)*x(7)-r)
```

```
*sin(q*x(6)*x(7)-r));
    y(8)=abs(p*sin(q*x(7)*x(8)-r)
*sin(q*x(7)*x(8)-r));
    for j=1:8
      x(j) = y(j);
    end
end
while k <=MAXSIZE
    y(1)=abs(p*sin(q*x(8)*x(1)-r)
*sin(q*x(8)*x(1)-r));
    y(2)=abs(p*sin(q*x(1)*x(2)-r)
*sin(q*x(1)*x(2)-r));
    y(3)=abs(p*sin(q*x(2)*x(3)-r)
*sin(q*x(2)*x(3)-r));
    y(4)=abs(p*sin(q*x(3)*x(4)-r)
*sin(q*x(3)*x(4)-r));
    y(5)=abs(p*sin(q*x(4)*x(5)-r)
*sin(q*x(4)*x(5)-r));
    y(6)=abs(p*sin(q*x(5)*x(6)-r)
*sin(q*x(5)*x(6)-r));
    y(7)=abs(p*sin(q*x(6)*x(7)-r)
*sin(q*x(6)*x(7)-r));
    y(8)=abs(p*sin(q*x(7)*x(8)-r)
*sin(q*x(7)*x(8)-r));
    s1=sqrt((y-x)*(y-x)');
    %计算 s1=||y-x||
    s2=(y-x)*t';
    %计算 s2=<y,t>-<x,t>
    s3=abs(s2);
      if s1<s3
    %是否可以比较? 可以进行序列取值
      if s2>=0                %如果 y>x
      wts(k)=1;               %序列取 1
      else
      wts(k)=0;               %序列取 0
      end
      k=k+1;
    end
    for j=1:8
      x(j)=y(j);
    end
end
wts                           %输出序列
```

4 总结与进一步的讨论

通过半序方法建立起来的序列密码系统由于不局限于整数域,可扩展到使用泛函空间中的元素.因此可以使得现有的密码分析学中的线性分析方法和基于代数和数论的分析方法彻底的失效,大大增强了密文的安全性,利用泛函方法建立密码系统具有良好的发展前景,目前研究尚处于起步阶段,还有许多理论工作要做,比如如何建立泛函框架下分组加密算法和公钥加密算法,如何使用函数取代整数的地位等等.还有我们必须保证算子 A 经过若干次迭代后不出现不动点和渐进不动点,因为出现不动点之后将导致常序列,失去了散列的伪随机性,鉴于同样的原因 A 不能是增算子和减算子.

参考文献:

[1] 左黎明,盛梅波,郑雄军,董祥南. 实 Hilbert 空间中锥的性质及其不动点定理[J]. 华东交通大学学报,2006,(2):156-159.

[2] Zhang H.,Dai J.,Wang P..Bifurcation and Chaos in a optically bitable Liquid-crystal device.Journal of Optical Soc Am B,1986,3(2).

[3] 黄润生. 混沌及其应用[M]. 武汉:武汉大学出版社,2000.

[4] 江成顺,刘霖雯,李晓聪,一类动力系统的不动点和分岔[J],郑州大学学报(理学版),2004,36(1):12~15.

A New Sequence Cryptography System Base on t-partial Order

ZUO Li-ming

(School of Natural Science, East China Jiaotong Uni., Nanchang 330013, China)

Abstract: The paper discusses how to apply -partial order to designing a new kind of sequence cryptography system.Finally, it gives the detailed algorithms of PRBG and MATLAB realization based on t- partial order.

Key words: partial order; sequence cryptography system; PRBG