



# (12) 发明专利申请

(10) 申请公布号 CN 115085904 A

(43) 申请公布日 2022. 09. 20

(21) 申请号 202210772306.3

(22) 申请日 2022.07.02

(71) 申请人 道和邦(广州)电子信息科技有限公司

地址 510440 广东省广州市白云区嘉禾街  
鹤龙二路96号粤旺大厦C栋305室

(72) 发明人 陈书增

(51) Int. Cl.

H04L 9/06 (2006.01)

H04L 9/32 (2006.01)

H04L 9/40 (2022.01)

权利要求书6页 说明书9页 附图3页

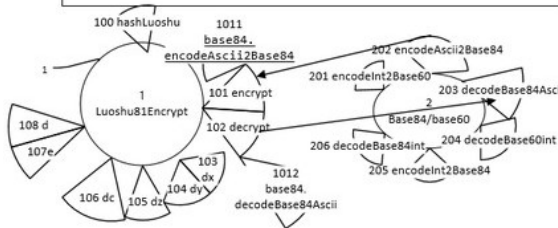
## (54) 发明名称

SMG-wscomm-Luoshu81Encrypt基于洛数81九宫方阵推衍的高安全高效高压压缩数据的超算加密技术

## (57) 摘要

洛数81九宫方阵,可实现对任意自然数通过与洛数发生作用产生81进制洛数,并通过变换洛数九宫方阵排列来实现对信息有效加密;信息元加密采用散列密钥元进行映射加密,每个信息元对应的元密钥不同,而密钥序列不可逆且难以攻破,这是本发明信息的安全基础;本发明较传统AES加密效率提升120倍;本发明为颠覆性技术超越传统计算理论,采用九宫方阵移形换位代替传统计算;本发明主要用于动态令牌加密编码传输;以及航天航空领域的大量观测大数进行压缩,可实现高安全性,高可靠性信息保密压缩传输;高倍空间压缩编码,高效率加密解密技术为超算力输出,超分布式存储输出必放一异彩。

SMG-wscomm-Luoshu81Encrypt基于洛数81九宫方阵推衍的高安全高效高压压缩数据的超算加密技术  
#author:walksing  
#date:20220630



1.一种基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密技术,其特征如下:

定义:洛数,也称洛书数,由洛书推理按九宫格LST排列方阵组合的1-81的自然数.

### 1.1 洛书81九宫方阵

```
publicstaticfinalint[] LST={
    //x+11                                //\|/y+10
    //x-> 1, 2, 3, 4, 5, 6, 7, 8, 9, //\|/
    47,58,69,80, 1,12,23,34,45, // 1
    57,68,79, 9,11,22,33,44,46, // 2
    67,78, 8,10,21,32,43,54,56, // 3
    77, 7,18,20,31,42,53,55,66, // 4
    6,17,19,30,41,52,63,65,76, // 5
    16,27,29,40,51,62,64,75, 5, // 6
    26,28,39,50,61,72,74, 4,15, // 7
    36,38,49,60,71,73, 3,14,25, // 8
    37,48,59,70,81, 2,13,24,35 // 9
    //z-> 47->35 z+21
    //v-> 37->45 v+1
};
```

1.2 其九宫方阵映射ascii字符编码表LSMT81如下:

LSTMapTable 1->81

```
publicstaticfinalchar[] LSMT81 = {
    '\0', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
    'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T',
    'U', 'V', 'W', 'X', 'Y', 'Z',
    'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
    'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't',
    'u', 'v', 'w', 'x', 'y', 'z',
    '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
    '@', '_', '!', '(', '$', ', , .', '/', ':', ')',
    '<', '~', '>', '[', ']', '^', '{', '|', '}', '`' //83
};
```

因为九宫方阵从1开始没有0,0其实是81;故为统一格式LSMT81表用'\0'作为第一位占位符,不代表任何数据,自1-81依次用LSMT81[1]-LSMT[81]标识编码表,多余的位占位符.

经过周密计算经过如上表格组合排列可推导如下:

### 1.3 洛数81九宫排列特征:

洛数81九宫排列特征如下:

1.3.1 数字始终遵循自西南向东北递增原则,遇到黑洞临界反弹下移一格,洛数加一,然后继续原方向前行递增;

### 1.3.2 遇边界自动向边界反向跨移9格:

如:遇上边界自动下移9格,遇下边界上移9格,遇右边界左右9格,遇左边界右移9格,遇到东南边界,西北移动移动9格,遇到东北边界东南移动九宫依此类推,可把九宫理解为观测宇宙显示仪的窗口;

1.3.3 遇到黑洞边界阻力(已被填充,可理解为黑洞临界,可理解慧星穿过太阳被太阳吸引力短暂拉回,然后继续被更大的引力推动前行),下移一格数字加1;如果遇到此规则与1.3.2规则相冲突,此规则优先;

### 1.3.4 洛数九宫方阵绘制过程

以x轴为横坐标,y轴为纵坐标西北为原点,自北向南画九行,自西向东画九列为九宫,依次填入下列数字:○内数字为坐标,未标记坐标的自动按1.3.1规律前行

1居中(5,1)→2(6,9),3,4,5→6(1,5),7,8,9→10(4,3),11,12→13(7,9),14,15→16(1,6),17,18→19(3,6),20,21,22,23→24(8,9),25→26(1,7),27→28(2,7)→29,30,31,32,33,34→35(9,9)→36(1,8)→37(1,9),38,39,40,41,42,43,44,45→46(9,2)→47(1,1)→48(2,9),49,50,51,52,53,54→55(8,4),56→57(1,2),58→59(3,9),60,61,62,63→64(7,6),65,66→67(1,3),68,69→70(4,9),71,72→73(6,8),74,75,76→77(1,4),78,79,80→81(5,9)

仔细研究81个数字刚好象征易经81卦象,易经爻辞里经常含九二,六三就是数字在九宫排列的坐标,遇到临界翻转,如46(9,2)→47(1,1)→48(2,9)坐标刚好翻转;标识事务的对立面;

1.3.5 九宫图三角形连线部分为z轴偏转路线,刚好遇到九格归位,九九归一即来源于此,这是本发明的重要基础用来进行偏移转变洛数数字加密基础;

1.3.6 俯视九宫方阵,将所有黑洞临界坐标的四个洛书圈为圆,即知一5,81洛数连线为中轴线,各有4个圆(黑洞),中间是个大圆(宇宙系轴心圆盘),而这些球体是通过推理计算洛数差计算得出来,并遵循相对论,当飞船靠近黑洞临界,速度时空变得弯曲(STEP位移变短),而其穿过黑洞临界恢复原长。

## 2.洛数81九宫方阵,其特征性质如下:

2.1 LST洛数81九宫方阵 横坐标x轴,纵坐标y轴,自西北到东南为z轴,其特征性质如下:

洛数定义:按LST九宫方阵排列的1—81自然数.

本发明重点研究洛数方阵之间数差存在特征规律,用于洛数之间的计算,利用这种天然数据组合的规律性用于数据加密,实现数据的高效率,安全性的偏移计算,可理解为超算计算,所谓超算借助于数字排列定理规律走入捷径,其算法比常规计算要快120倍;

黑洞临界:对任意洛数n,向x,y,z轴任一方向移动x0,y0,z0位后,遇到步长step不同于常规步长(步长变短),则标记为临界依据依次穿越的对应的偏移坐标标记h0,h1,h2;x轴最多2个,y轴只有一个h0=10-2x;z轴最多3个,用c标识其穿越临界的个数.如果把临界的x,y,z轴方向的四个数用圆圈起来,即可理解为一黑洞星球;如果把洛数移动看作飞船在星际穿越,则有当飞船靠近黑洞边界时,速度会变慢;如果速度不变,则时间变慢;时间变慢的原因是因为黑洞空间曲面产生弯曲(根据爱因斯坦相对论)而这一规则在洛数亦有相同规则,而其穿越后又恢复速度直到遇到下一个临界.

为求得n向任意方向移动k位后所得的洛数，

通项计算公式：

$d1=(n+step*(k-1)-9*c)\%81$ ；且 $d1\leq 0$  时 $d1+81:d1$ ； l代表： $n(x,y,z)$  的坐标位移，d1为洛数n在九宫方阵l方向偏移k位的洛数；

2.2  $dx(n,x0)$  表示洛数n向x轴移动x0位后的洛数x,当x0逢9归位即 $dx(n,9)=dx(n,-9)=n$ ,或当 $x0\%9==0$ ,则有 $dx(n,0)=n$ ;x0偏移一步的步长,记 $step=11$ ,当x0穿越黑洞临界时, $step-9=2$ ,当其穿过,临界时step步长恢复11; 基于2.1通项公式: $dx=(n+step*(x0-x)-9*c)\%81$ ; 2.3  $dy(n,y0)$  表示洛数n向y轴移动y0位后的洛数x,当y0逢9归位即 $dy(n,9)=dy(n,-9)=n$ ,或当 $y0\%9==0$ ,则有 $dy(n,0)=n$ ;y0偏移一步的步长,记 $step=10$ ,当y0穿越黑洞临界时, $step-9=1$ ,当其穿过,临界时step步长恢复10; 基于2.1通项公式: $dy=(n+step*(y0-y)-9*c)\%81$ ; 2.4  $dz(n,y0)$  表示洛数n向y轴移动y0位后的洛数x,当y0逢9归位即 $dz(n,9)=dz(n,-9)=n$ ,或当 $y0\%9==0$ ,则有 $dz(n,0)=n$ ;y0偏移一步的步长,记 $step=21$ ,当y0穿越黑洞临界时, $step-9=12$ ,当其穿过,临界时step步长恢复21;

基于2.1通项公式: $dz=(n+step*(y0-y)-9*c)\%81$ ;

2.5绕41轴按时针顺时或逆势坐标对称的两个数字和总为82,

$dc(n)=82-n\Rightarrow d(82-n)=82-(82-n)=n\Rightarrow dc(dc(n))=n$ ;

2.6 基于1,2.1 对任意洛数 $dx(n,9)=n$ ,或 $dx(n,m)=n$  ,当且仅当 $m=n\%9,m==0$ 时,存在;

2.7 中心椭圆可视为于宇宙星系中心

20,31,42,

30,41,52,

40,51,62,

几个数字用椭圆圈起来,41为心,十位数

左	中	右
2	3	4
3	4	5
4	5	6

自x轴,y轴方向其数字排列始终:234,345,456

个位数依次

左	中	右
0	1	2
0	1	2
0	1	2

2.8 穿越黑洞临界点，

LST洛数81九宫方阵对任何洛数 $n(x,y)$  ,其黑洞临界点依次标记 $h0,h1,h2$

则有特征如下，

2.8.1 x轴临界: $c\leq 2$

```
switch (y) { //black hole criticality 临界
  case 1: {h0=4;h1=9;break;}
  case 2: {h0=4;h1=8;break;}
```

```

    case 3: {h0=3;h1=8;break;}
    case 4: {h0=3;h1=7;break;}
    case 5: {h0=2;h1=7;break;}
    case 6: {h0=2;h1=6;break;}
    case 7: {h0=1;h1=6;break;}
    case 8: {h0=1;h1=5;break;}
    case 9: {h0=5;h1=9;break;}
}

```

2.8.2 y轴临界: $c=1, h_0=(10-2*x)$ ; if  $(h_0 \leq 0) h_0 += 9$ ;

2.8.3 z轴临界: $c \leq 3$ ; z轴临界  $h_0, h_1, h_2$  依次由低位到高位延伸, 为方便计算当临界小于前一临界数坐标时, 统一+9, 确保始终 $h_0 < h_1 < h_2$ 的状态.

```

switch (y) { //black hole criticality 临界
    case 1: {h0=3;h1=6;h2=9;break;}
    case 2: {h0=4;h1=7;h2=10;break;}
    case 3: {h0=4;h1=7;h2=10;break;}
    case 4: {h0=5;h1=8;h2=11;break;}
    case 5: {h0=6;h1=9;h2=12;break;}
    case 6: {h0=6;h1=9;h2=12;break;}
    case 7: {h0=7;h1=10;h2=13;break;}
    case 8: {h0=8;h1=11;h2=14;break;}
    case 9: {h0=11;h1=14;h2=17;break;}
};

```

2.9 俯视九宫方阵, 自西北向东南依次连接洛数,  $x, y$  九宫边界点形成回路三角形;

如  $\{37, 58, 25\}, \{36, 69, 15\}, \{26, 80, 5\}, \{16, 1, 76\}, \{6, 12, 66\}, \{77, 23, 56\}, \{67, 34, 46\},$   
 $\{57, 45, 24\} \dots$

任意三角形洛书连接点之洛数存在如下规律:

洛数三角定理: 任意洛数三角连接之洛数,  $A, B, C$  构成洛数三角, 若  $A < B < C$ ; 令  $a = B - A, b = C - B, c = C - A$ ; 则必有  $c = a + b$ , (若洛数为负数可通过  $+81$  转为洛数);

证明: 设推理为真,  $c = a + b \Rightarrow C - A = B - A + C - B \Rightarrow C - A = -A + C \Rightarrow$  满足交换律定理 ;

注: 洛书三角定理不仅仅是回路三角形, 在非回路三角形同样存在, 即九宫方阵任意三个洛数连线构建三角形或直线 (180度) 均满足洛数三角定理;

2.10 基于2.9 俯视九宫方阵观测这些连线洛数三角发现有规律的进行旋转切分类似微积分绕着轴心圆旋转; 而这些三角形洛数存在回路, 即自z轴按照dz(x) 运算算到第九次回到原点;

2.11 基于2.9, 2.8, 2.10 俯视观测从九宫方阵可观测到旋转的运动的时空星系, 从1维点与点, 到二维度连线, 圆; 到三维黑洞(星球), 到四维旋转的微分洛数三角, 推测到一个多维时空抽象图。

3. 基于2.5, 幻数  $f(x)$  (参幻数定理) 性质, 可实现对任意字符串, 可实现一种高效率,

高安全性的不可逆单向信息加密摘要算法,其特征如下:

3.1 对应任意字符串s可求其ascii码数字之和标记为long ks=hashSum(s);在此运算过程存在私钥keyBase 非公开私钥加盐;不可预测性;

3.2 依据3 f(x) 对 3.1 ks 进行幻数运算;即 long ks=MagicNumberEncrypt.f(hashSum(s));

3.3 基于3.2 得到ks 包含所有ascii串基础,再以此基础数对key串进行以,余长: mlen=32-String.valueOf(ks).length 为基数,对key进行分组g,每组取1位 dx= (int) s.charAt(g\*i);

然后将ascii码转为洛数,并进行dc运算,dx=dc(dx);再用ks对其求幻数;即dx=MagicNumberEncrypt.f(ks+dx);然后截取倒数第2位(寻找最能代表其信息的一位,测试观测法);直到收集到32位,信息头用6位ks代替;返回buf即32位散列数字;

3.4 基于3.3 可实现对任意key进行散列运算且不可逆推;并保持信息的摘要,具有如下特征:

3.4.1 高安全性基于私钥基加盐加密功能,不可预测性;私钥非公开或可定期变化性;

3.4.2 算法高效率,经测试比传统md5散列算法快10倍,本算法用到洛数偏转对称性,以及幻数俩大高效率超算算法;幻数类似于珠算,将传统乘法转为俩个大数移位加法或10进位乘法;效率大大提升;

3.4.3 基于3.1,3.3 保障了信息的分布性,代表全局信息;

3.4.4 基于3.2 ,3.3 幻数运算保障了散列数据的放大性;因为幻术运算形式类似抛物线走势;任何微细的数据改变经过幻数超算,使得摘要信息放大性,且均衡性;

3.4.5 基于3.4.1,3.4.2 ,3.4.3,3.4.4 可直接将hashLuoshu 算法对key密钥进行散列运算以其结果作为流密钥进行洛数密钥基础进行加密。

4.基于2.4,2.5 ,3.4.5实现的一种高效率,高压缩比,高安全性超算数据加密技术,其特征如下:加密过程,先对字符串进行base84编码,再对key进行散列加密,以散列结果为key进行遍历轮转取散列key元组,对元字符加密,每个元字符,偏移量不同,最终加密后的数据直接按LSMT81字符表输出;

4.1 高数据压缩率,基于base84(参前专利申请) encodeAscii2Base84 可实现对数字数据实现高达接近50%的压缩率,且高效率;对ascii混合字符本发明可接近数据压缩率接近110%,幻数压缩率151%,aes+base64算法接近135%;洛数是接近1:1数字偏转除少量从ascii码转洛数需要损失一部分编码空间这个比例接近20%;

4.2 高效率,利用了洛数在九宫方阵排列的神奇轮转特征,使得洛数可通过移位变换数字,且这种变换不会轻易被识别;利用洛数轮转特性很容易设计机械运算,故很多古代机关陷阱为何都是基于洛数设计,经过实测:本发明洛数加密较幻数快2倍;较传统AES,ECB,Base64合并(因为AES是基于byte加密的不适合传输必须结合base64才能传输)算法快120倍,数据压缩率,纯数字接近50%,ascii混合110%,幻数是150%;

4.3 高安全性,单纯的1次元字符数据容易破解,比如轮转九次即可破解;但是信息元字符只是组成信息的基本单元;整段信息是靠信息的合理排序进行表达;而合理排序是基于3.4.5 hashLuoshu在用户密钥基础上散列的32位流密钥组合;用户密钥不可预测,在散列过程中存在基keybase私钥更是不可预测,这种算法保障了流密钥的安全;由此密钥组产

生的加密信息亦更安全。

## SMG-wscomm-Luoshu81Encrypt基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密技术

### 技术领域

[0001] 本发明为一种新型颠覆性超算技术,通过推理洛数81九宫方阵图,得到一些基本的性质,通过研究其数据空间排列数差,找到数差运算规律,并用于数据加密,信息安全领域应用;本发明是经过反复推导洛数81个洛数,计算发现洛数定理,并加以推衍应用于计算机领域信息安全领域高效率高安全性的超算信息加密信息通信服务,该发明为基础api应用服务,可应用在所有领域,无论是信息安全,还是数据压缩,快算,超算等基础应用,极大的加快信息的安全共享服务,与基础通信保密服务,经过测试比传统AES+Base64 快尽120倍,比最新公布的幻数超算MagicNumberEncrypt 快2倍,数据压缩率纯数字接近50%,ascii码混合接近110%,无论速度压缩率都高于传统。

[0002] 本项发明,为SMG-VME可迭代分布式操作系统,SMG-VME-AFS可迭代分布式存储系统延伸的价值,iSBS/mbs可迭代超级商城动态令牌技术的应用基础,主要应用于信息的高倍编码,高效率的加密基础超算应用。

[0003] 本项发明,为SMG-VME系列下软件工程实现的社会使命,其目标为解决超算,超存储输出,以及信息的安全传输,安全获取。

[0004] 本项发明依赖于前专利:幻数MagicNumberEncrypt,Base84进制高效率编码技术专利审查:

截至2022-06-21网络搜索暂无同案例

### 背景技术

[0005] 本发明主要应用在工业互联网,万物智能互联,分布式计算,万物互联数据信息的加密交换,信息的高速运算。本发明最初用于iSBS/mbs可迭代超级商城用户动态令牌ECToken的加密领域,ECToken动态令牌技术(详见前专利),是基于先进的MagicNumberEncrypt幻数Cookie加密压缩技术将用户的基础信息库压缩推入令牌,如同用户微数据库每次请求携带此令牌,服务端系统进行解密并动态刷新流密钥到客户端完成安全通信,可实现基于http非安全环境构建安全无状态私有协议代替传统的http状态协议,并支持高并发超级量用户访问服务端无内存压力,该协议要求加密解密速度足够快,且密文尽可能被压缩不能超过4K,最初用幻数加密方案;幻数美中不足压缩率151%稍高,普通用户明文数据1422字节,幻数加密压缩后密文:约2151字节,数据压缩率(ASCII码)151%,采用本发明数据加密后降为1566字节,数据压缩率(ASCII码)110%;为提高压缩率偶然发现并研究了洛数81九宫方阵,经过探索研究终于成功研发了本发明Luoshu81Encrypt 算法;成功将幻数算法改为本算法压缩率从151%到了接近110%;对纯数字可接近50%;调用效率方面较幻数提高了2倍,因为输出加密密文不再进行base84编码而该用自身的洛数81编码;只对原始明文含中文进行Base84.encodeAscii2base84;本发明较传统AES,Base64快120倍,而后者数据压缩率135%多,且不支持中文编码,同时本算法采用自研发的新型散列运算hashLuoshu 算法;较传统md5散列算法快10倍,且采用私钥加密更安全。



## 发明内容

## 附图说明

[0006] 图1是Luoshu81Encrypt 洛数加密类的逻辑函数图;

1Luoshu81Encrypt加密

100 hashLuoshu 基于幻数洛数hash散列摘要算法

101 encrypt 加密

102 decrypt 解密

1011 84进制ascii压缩编码

1012 84进制ascii解码

103 dx 求洛数n 在x轴偏移的洛数

104 dy 求洛数n 在y轴偏移的洛数105 dz 求洛数n 在z轴偏移的洛数

106 dc 求洛数n 在绕41轴心的对称洛数

107 e 经过105 dz,106 dc运算后的洛数

108 d 经过106 dc后,经105dz的逆运算

图2是洛数九宫图(九宫方阵);

图3是LSMT81 洛书方阵对应ascii表;

图4是hashLuoshu洛书hash散列信息摘要加密算法;

标号说明

100 shLuoshu

1000:hashSum 求输入信息的ascii和中参入keybase基私钥(加盐)

1001 :MagicNumberEncrypt.f(x) 对自然数x求幻数运算(放大自然数)

1002 :dc(x) 求洛数x 在绕41轴心的对称洛数

4.1 1000 hashSum 对任意字符串进行hash散列运算返回总hash值,在运算前对字符串s进行加盐keybase(非公开)处理,返回long ks=MagicNumberEncrypt.f(hashSum(s));

buf(ks)

4.2 对字符串s进行以剩余32-ks.length 为基数进行分组,每组取倒数底二位,至倒数第一位;buf至总长为32位结束(截取表达信息单元最多的一位,与其它单元重复性最少的一位,亦可取最后一位)

4.3 循环体其算法如下: dx=(int) s.charAt(g\*i);g为组号i为buf hash结果字符序号总长32位数字

dx=dc((int) dx);

dx=MagicNumberEncrypt.f(ks+dx);

t=String.valueOf(dx);

t=t.substring(t.length()-2,t.length()-1);

o.append(t);至此,输出key=32位数字即为s的散列摘要,通过key=hashLuoshu(s);s的任意改动必将导致key输出很大改变本发明散列摘要为安全加密型散列摘要可作为密钥,签名使用,且不可逆算

4.4 本发明散列算法特征如下：

4.4.1 散列算法：目标是信息的摘要，微小信息的改变，其信息摘要会引起很大的改变，且通过信息摘要对原信息不可推测，散列单调性；

4.4.2本散列hash算法较md5快10倍，且输出32位数字；4.4.3本算法含私钥加盐密散列，并采用幻数，洛数超算功能速度快，安全；4.4.4本算法是解决洛书九宫方阵为9防暴力破解，因在加密过程中，任何密钥被散列为32位数字散列密钥的每一位数字为流密钥k参与元信息的加密，即每一位字符采用不同的流密钥进行加密，其加密规则依据密钥流信息的改变而改变，进而提高破解难度，其核心依据是暴力破解一位元很容易，但破解密钥流的序列很难，因为key不可预测，经过hash散列亦不可预测，且放大了这种不可预测性；

图5是encrypt洛书加密算法；

101:encrypt 加密

10100:hashLuoshu 本发明之一基于Luoshu散列

101001:以key为密钥散列的32位数字信息摘要

10101:encodeAscii2Base84 Base84 ascii混合编码压缩技术

101020:buf LSMT81[x] 编码输出依照LSMT81 ascii表进行映射输出

10102:x=e(n,k) 洛数单元加密对洛数n,按z轴偏移k得到洛数并取其轴心对称洛数x

101021:dz(x) 对洛数x 求z轴,偏移k 的洛数

101022: dx(x) 对洛数x 求 x轴,偏移k的洛数

101023: dy(x) 对洛数x 求 y轴,偏移k的洛数

101024: dc(x) 对洛数x 求绕轴心对称的洛数

注:10102 e(n,k) 是基于101021,101022,101023,101024 的综合调用；调用：

图5,101 encrypt->输入key,s ,key为可变长密钥字符串,s为任意字符串(可包含中文)->图5, 10100 key=hashLuoshu(key) 将key进行hasuLuoshu散列得到新key.此过程是为保障洛数算法安全防暴力破解；把明文s进行Base84压缩编码,如果含长数字会被压缩:即调图5,10101 s=Base84.encodeAscii2Base84(s)进行base84 encode编码->遍历s,取元  $x=s.charAt(i)$  ,x为ascii码转为洛数,并标记;进而对其调10102  $x=e(x,k) \Rightarrow$  101021  $dz(x,k) \rightarrow$  101022  $dx(x,k) \rightarrow$  101023  $dy(x,k) \rightarrow$  101024  $dc(x)$  完成元加密并调 101020 buf LSMT81[x] 缓存;最后输出缓存o 即得密文

图6是decrypt洛书解密算法；

102:decrypt

10201:hashLuoshu key 对key按hashLuoshu算法散列

102011:生成以key为密钥散列为32位信息摘要

102012:int81(x) 取基于LSMT81的索引号

102013: x=d(n,k) 解密为10102 e(n,k)的逆运算,k为102011 散列的流密钥元；每个加密、解密元不同

1020131:buf(char) x 得到ascii码

10204:Base84.decodeBase84Ascii 为10101的逆运算 Base84解压缩解码明文

解密调用逻辑:参图6

输入密钥key,密文S →10201 key=hashLuoshu(key) 对key进行hashLuoshu算法散列为32位流密钥→遍历S 取LSMT81洛数映射ASCII表调102012 x=int81(x) 取索引号,根据密文偏移求得k流密钥元,调102013 x=d(x,k) 解密→调1020131 buf(char) x - > 遍历结束,再调10204 Base84.decodeBase84Ascii码即明文。

[0007] 发明要义:

1.一种基于基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密技术,其特征如下:

定义:洛数,也称洛书数,由洛书推理按九宫格LST排列方阵组合的1-81的自然数.

1.1 洛书81九宫方阵

```
publicstaticfinalint[] LST={
    //x+11                                /\|/y+10
    //x-> 1, 2, 3, 4, 5, 6, 7, 8, 9, /\|/
    47,58,69,80, 1,12,23,34,45, // 1
    57,68,79, 9,11,22,33,44,46, // 2
    67,78, 8,10,21,32,43,54,56, // 3
    77, 7,18,20,31,42,53,55,66, // 4
    6,17,19,30,41,52,63,65,76, // 5
    16,27,29,40,51,62,64,75, 5, // 6
    26,28,39,50,61,72,74, 4,15, // 7
    36,38,49,60,71,73, 3,14,25, // 8
    37,48,59,70,81, 2,13,24,35 // 9
    //z-> 47->35 z+21
    //v-> 37->45 v+1
};
```

1.2 其九宫方阵映射ascii字符编码表LSMT81如下:

LSMapTable 1->81

```
publicstaticfinalchar[] LSMT81 = {
    '\0','A','B','C','D','E','F','G','H','I','J',
    'K','L','M','N','O','P','Q','R','S','T',
    'U','V','W','X','Y','Z',
    'a','b','c','d','e','f','g','h','i','j',
    'k','l','m','n','o','p','q','r','s','t',
    'u','v','w','x','y','z',
    '0','1','2','3','4','5','6','7','8','9',
    '@','_','!','(','$','.',',','/',':',')',
    '<','~','>','[',']','^','{','|','}','`'//83
};
```

因为九宫方阵从1开始没有0,0其实是81;故为统一格式LSMT81表用'\0'作为第一

位占位符,不代表任何数据,自1-81依次用LSMT81[1]-LSMT[81]标识编码表,多余的位占位符.

经过周密计算经过如上表格组合排列可推导如下:

### 1.3 洛数81九宫排列特征:

洛数81九宫排列特征如下:

1.3.1 数字始终遵循自西南向东北递增原则,遇到黑洞临界反弹下移一格,洛数加一,然后继续原方向前行递增;

#### 1.3.2 遇边界自动向边界反向跨移9格:

如:遇上边界自动下移9格,遇下边界上移9格,遇右边界左右9格,遇左边界右移9格,遇到东南边界,西北移动移动9格,遇到东北边界东南移动九宫依此类推,可把九宫理解为观测宇宙显示仪的窗口;

1.3.3 遇到黑洞边界阻力(已被填充,可理解为黑洞临界,可理解慧星穿过太阳被太阳吸引力短暂拉回,然后继续被更大的引力推动前行),下移一格数字加1;如果遇到此规则与1.3.2规则相冲突,此规则优先;

#### 1.3.4 洛数九宫方阵绘制过程

以x轴为横坐标,y轴为纵坐标西北为原点,自北向南画九行,自西向东画九列为九宫,依次填入下列数字:○内数字为坐标,未标记坐标的自动按1.3.1规律前行

1居中1(5,1)→2(6,9),3,4,5→6(1,5),7,8,9→10(4,3),11,12→13(7,9),14,15→16(1,6),17,18→19(3,6),20,21,22,23→24(8,9),25→26(1,7),27→28(2,7)→29,30,31,32,33,34→35(9,9)→36(1,8)→37(1,9),38,39,40,41,42,43,44,45→46(9,2)→47(1,1)→48(2,9),49,50,51,52,53,54→55(8,4),56→57(1,2),58→59(3,9),60,61,62,63→64(7,6),65,66→67(1,3),68,69→70(4,9),71,72→73(6,8),74,75,76→77(1,4),78,79,80→81(5,9)

仔细研究81个数字刚好象征易经81卦象,易经爻辞里经常含九二,六三就是数字在九宫排列的坐标,遇到临界翻转,如46(9,2)→47(1,1)→48(2,9)坐标刚好翻转;标识事务的对立面;

1.3.5 九宫图三角形连线部分为z轴偏转路线,刚好遇到九格归位,九九归一即来源于此,这是本发明的重要基础用来进行偏移转变洛数数字加密基础;

1.3.6 俯视九宫方阵,将所有黑洞临界坐标的四个洛书圈为圆,即知一5,81洛数连线为中轴线,各有4个圆(黑洞),中间是个大圆(宇宙系轴心圆盘),而这些球体是通过推理计算洛数差计算得出来,并遵循相对论,当飞船靠近黑洞临界,速度时空变得弯曲(STEP位移变短),而其穿过黑洞临界恢复原长。

[0008] 洛数81九宫方阵,其特征性质如下:

2.1 LST洛数81九宫方阵横坐标x轴,纵坐标y轴,自西北到东南为z轴,其特征性质如下:

洛数定义:按LST九宫方阵排列的1-81自然数.

本发明重点研究洛数方阵之间数差存在特征规律,用于洛数之间的计算,利用这种天然数据组合的规律性用于数据加密,实现数据的高效率,安全性的偏移计算,可理解为超算计算,所谓超算借助于数字排列定理规律走入捷径,其算法比常规计算要快120倍;

黑洞临界:对任意洛数n,向x,y,z轴任一方向移动x0,y0,z0位后,遇到步长step不同于常规步长(步长变短),则标记为临界依据依次穿越的对应的偏移坐标标记h0,h1,h2;x轴最多2个,y轴只有一个h0=10-2x;z轴最多3个,用c标识其穿越临界的个数.如果把临界的x,y,z轴方向的四个数用圆圈起来,即可理解为一黑洞星球;如果把洛数移动看作飞船在星际穿越,则有当飞船靠近黑洞边界时,速度会变慢;如果速度不变,则时间变慢;时间变慢的原因是因为黑洞空间曲面产生弯曲(根据爱因斯坦相对论)而这一规则在洛数亦有相同规则,而其穿越后又恢复速度直到遇到下一个临界.

为求得n向任意方向移动k位后所得的洛数,

通项计算公式:

$d1 = (n + step * (k - 1) - 9 * c) \% 81$ ; 且  $d1 <= 0$  时  $d1 + 81 : d1$ ; 1代表:n(x,y,z)的坐标位移,d1为洛数n在九宫方阵1方向偏移k位的洛数;

2.2  $dx(n, x0)$  表示洛数n向x轴移动x0位后的洛数x,当x0逢9归位即  $dx(n, 9) = dx(n, -9) = n$ , 或当  $x0 \% 9 = 0$ , 则有  $dx(n, 0) = n$ ; x0偏移一步的步长,记  $step = 11$ , 当x0穿越黑洞临界时,  $step - 9 = 2$ , 当其穿过, 临界时step步长恢复11; 基于2.1通项公式:  $dx = (n + step * (x0 - x) - 9 * c) \% 81$ ; 2.3  $dy(n, y0)$  表示洛数n向y轴移动y0位后的洛数x,当y0逢9归位即  $dy(n, 9) = dy(n, -9) = n$ , 或当  $y0 \% 9 = 0$ , 则有  $dy(n, 0) = n$ ; y0偏移一步的步长,记  $step = 10$ , 当y0穿越黑洞临界时,  $step - 9 = 1$ , 当其穿过, 临界时step步长恢复10; 基于2.1通项公式:  $dy = (n + step * (y0 - y) - 9 * c) \% 81$ ; 2.4  $dz(n, y0)$  表示洛数n向y轴移动y0位后的洛数x,当y0逢9归位即  $dz(n, 9) = dz(n, -9) = n$ , 或当  $y0 \% 9 = 0$ , 则有  $dz(n, 0) = n$ ; y0偏移一步的步长,记  $step = 21$ , 当y0穿越黑洞临界时,  $step - 9 = 12$ , 当其穿过, 临界时step步长恢复21;

基于2.1通项公式:  $dz = (n + step * (y0 - y) - 9 * c) \% 81$ ;

2.5绕41轴按时针顺时或逆势坐标对称的两个数字和总为82,

$dc(n) = 82 - n \Rightarrow d(82 - n) = 82 - (82 - n) = n \Rightarrow dc(dc(n)) = n$ ;

2.6 基于1,2.1 对任意洛数  $dx(n, 9) = n$ , 或  $dx(n, m) = n$ , 当且仅当  $m = n \% 9, m = 0$  时, 存在;

2.7 中心椭圆可视为于宇宙星系中心

20,31,42,

30,41,52,

40,51,62,

几个数字用椭圆圈起来,41为心,十位数

左	中	右
2	3	4
3	4	5
4	5	6

自x轴,y轴方向其数字排列始终:234,345,456

个位数依次

左	中	右
0	1	2
0	1	2

0      1      2

2.8 穿越黑洞临界点,

LST洛数81九宫方阵对任何洛数 $n(x,y)$ ,其黑洞临界点依次标记 $h_0,h_1,h_2$ 则有特征如下,

2.8.1 x轴临界: $c \leq 2$

```
switch (y) { //black hole criticality 临界
  case 1: {h0=4;h1=9;break;}
  case 2: {h0=4;h1=8;break;}
  case 3: {h0=3;h1=8;break;}
  case 4: {h0=3;h1=7;break;}
  case 5: {h0=2;h1=7;break;}
  case 6: {h0=2;h1=6;break;}
  case 7: {h0=1;h1=6;break;}
  case 8: {h0=1;h1=5;break;}
  case 9: {h0=5;h1=9;break;}
}
```

2.8.2 y轴临界: $c=1, h_0=(10-2*x); \text{if}(h_0 \leq 0) h_0 += 9;$

2.8.3 z轴临界: $c \leq 3$ ; z轴临界  $h_0, h_1, h_2$  依次由低位到高位延伸,为方便计算当临界小于前一临界数坐标时,统一+9,确保始终 $h_0 < h_1 < h_2$ 的状态.

```
switch (y) { //black hole criticality 临界
  case 1: {h0=3;h1=6;h2=9;break;}
  case 2: {h0=4;h1=7;h2=10;break;}
  case 3: {h0=4;h1=7;h2=10;break;}
  case 4: {h0=5;h1=8;h2=11;break;}
  case 5: {h0=6;h1=9;h2=12;break;}
  case 6: {h0=6;h1=9;h2=12;break;}
  case 7: {h0=7;h1=10;h2=13;break;}
  case 8: {h0=8;h1=11;h2=14;break;}
  case 9: {h0=11;h1=14;h2=17;break;}
};
```

2.9 俯视九宫方阵,自西北向东南依次连接洛数 $x,y$ 九宫边界点形成回路三角形;

如  $\{37, 58, 25\}, \{36, 69, 15\}, \{26, 80, 5\}, \{16, 1, 76\}, \{6, 12, 66\}, \{77, 23, 56\}, \{67, 34, 46\}, \{57, 45, 24\} \dots$

任意三角形洛书连接点之洛数存在如下规律:

洛数三角定理:任意洛数三角连接之洛数 $A, B, C$ 构成洛数三角,若 $A < B < C$ ;令 $a=B-A, b=C-B, c=C-A$ ;则必有 $c=a+b$ , (若洛数为负数可通过+81转为洛数);

证明:设推理为真, $c=a+b \Rightarrow C-A=B-A+C-B \Rightarrow C-A=-A+C \Rightarrow$ 满足交换律定理;

注:洛书三角定理不仅仅是回路三角形,在非回路三角形同样存在,即九宫方阵任意三个洛数连线构建三角形或直线(180度)均满足洛数三角定理;

2.10 基于2.9 俯视九宫方阵观测这些连线洛数三角发现有规律的进行旋转切分类似微积分绕着轴心圆旋转;而这些三角形洛数存在回路,即自z轴按照 $dz(x)$ 运算算到第九次回到原点;

2.11 基于2.9,2.8,2.10 俯视观测从九宫方阵可观测到旋转的运动的时空星系,从1维点与点,到二维度连线,圆;到三维黑洞(星球),到四维旋转的微分洛数三角,推测到一个多维时空抽象图。

[0009] 基于2.5 ,幻数 $f(x)$ (参幻数定理)性质,可实现对任意字符串,可实现一种高效率,高安全性的不可逆单向信息加密摘要算法,其特征如下:

3.1 对应任意字符串s可求其ascii码数字之和标记为 $long\ ks=hashSum(s)$ ;在此运算过程存在私钥keyBase 非公开私钥加盐;不可预测性;

3.2 依据3.1  $f(x)$  对 3.1 ks 进行幻数运算;即  $long\ ks=MagicNumberEncrypt.f(hashSum(s))$ ;

3.3 基于3.2 得到ks 包含所有ascii串基础,再以此基础数对key串进行以,余长: $mLen=32-String.valueOf(ks).length$  为基数,对key进行分组g,每组取1位  $dx=(int)s.charAt(g*i)$ ;

然后将ascii码转为洛数,并进行dc运算, $dx=dc(dx)$ ;再用ks对其求幻数;即 $dx=MagicNumberEncrypt.f(ks+dx)$ ;然后截取倒数第2位(寻找最能代表其信息的一位,测试观测法);直到收集到32位,信息头用6位ks代替;返回buf即32位散列数字;

3.4 基于3.3 可实现对任意key进行散列运算且不可逆推;并保持信息的摘要,具有如下特征:

3.4.1 高安全性基于私钥基加盐加密功能,不可预测性;私钥非公开或可定期变化性;

3.4.2 算法高效率,经测试比传统md5散列算法快10倍,本算法用到洛数偏转对称性,以及幻数俩大高效率超算算法;幻数类似于珠算,将传统乘法转为俩个大数移位加法或10进位乘法;效率大大提升;

3.4.3 基于3.1,3.3 保障了信息的分布性,代表全局信息;

3.4.4 基于3.2 ,3.3 幻数运算保障了散列数据的放大性;因为幻术运算形式类似抛物线走势;任何微细的数据改变经过幻数超算,使得摘要信息放大性,且均衡性;

3.4.5 基于3.4.1,3.4.2 ,3.4.3,3.4.4 可直接将hashLuoshu 算法对key密钥进行散列运算以其结果作为流密钥进行洛数密钥基础进行加密。

[0010] 基于2.4,2.5 ,3.4.5实现的一种高效率,高压缩比,高安全性超算数据加密技术,其特征如下:加密过程,先对字符串进行base84编码,再对key进行散列加密,以散列结果为key进行遍历轮转取散列key元组,对元字符加密,每个元字符,偏移量不同,最终加密后的数据直接按LSMT81字符表输出;

4.1 高数据压缩率,基于base84(参前专利申请)  $encodeAscii2Base84$  可实现对数字数据实现高达接近50%的压缩率,且高效率;对ascii混合字符本发明可接近数据压缩率接近110%,幻数压缩率151%,aes+base64算法接近135%;洛数是接近1:1数字偏转除少量

从ascii码转洛数需要损失一部分编码空间这个比例接近20%;

4.2 高效率,利用了洛数在九宫方阵排列的神奇轮转特征,使得洛数可通过移位变换数字,且这种变换不会轻易被识别;利用洛数轮转特性很容易设计机械运算,故很多古代机关陷阱为何都是基于洛数设计,经过实测:本发明洛数加密较幻数快2倍;较传统AES, ECB, Base64合并(因为AES是基于byte加密的不适合传输必须结合base64才能传输)算法快120倍,数据压缩率,纯数字接近50%,ascii混合110%,幻数是150%;

4.3 高安全性,单纯的1次元字符数据容易破解,比如轮转九次即可破解;但是信息元字符只是组成信息的基本单元;整段信息是靠信息的合理排序进行表达;而合理排序是基于3.4.5 hashLuoshu在用户密钥基础上散列的32位流密钥组合;用户密钥不可预测,在散列过程中存在基keybase私钥更是不可预测,这种算法保障了流密钥的安全;由此密钥组产生的加密信息亦更安全。

[0011] 测试数据如下:

线上用户基础数据

明文含中,英文,符号,数字混合,明文:1422

各算法对比

算法	密文长	耗时ns	压缩率	效率比(M:洛数)
洛数Luoshu81Encrypt	1566	6478443	110%	1:1
幻数MagicNumberEncrypt	2151	13270274	151%	1:2.04
AES-ECB-Base64	1920	781055596	135%	1:120

测试环境,win7,JDK1.78,eclipse 3.0,多次刷新取近似值。

[0012] 现实意义:

SMG-wscomm-Luoshu81Encrypt基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密技术,可实现对任意自然数通过与洛数发生作用产生81进制洛数,并通过变换洛数九宫方阵排列来实现对信息有效加密,为防止暴力破解,又设计出hashLuoshu散列函数,可实现对key为密钥,进行私钥加盐散列;提高安全性;信息元加密采用散列密钥元进行映射加密,每个信息元对应的元密钥不同,而密钥序列不可逆且难以攻破,这是本发明信息的安全基础;本发明较传统AES加密效率提升120倍,较幻数加密快2倍;本发明为颠覆性技术超越传统计算理论,采用九宫方阵移形换位代替传统计算;本发明主要用于动态令牌加密编码传输;以及航天航空领域的大量观测大数进行压缩,结合Base84算法(参同期专利)可实现高安全性,高可靠性信息保密压缩传输;高倍空间压缩编码,高效率加密解密技术为超算力输出,超分布式存储输出必放一异彩。



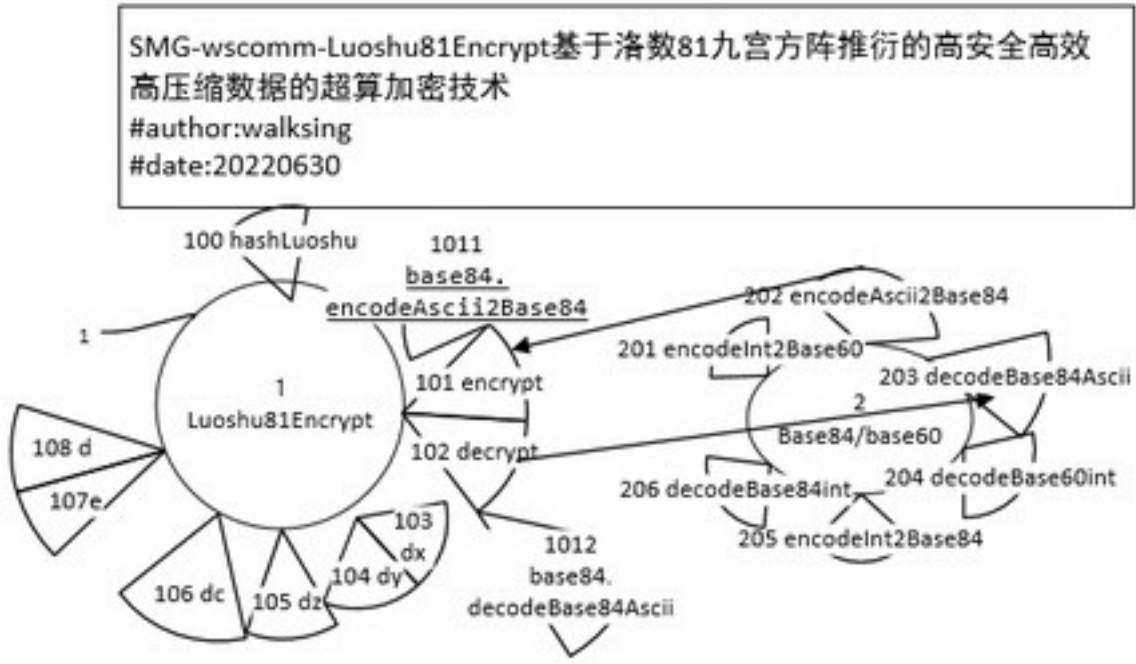


图1

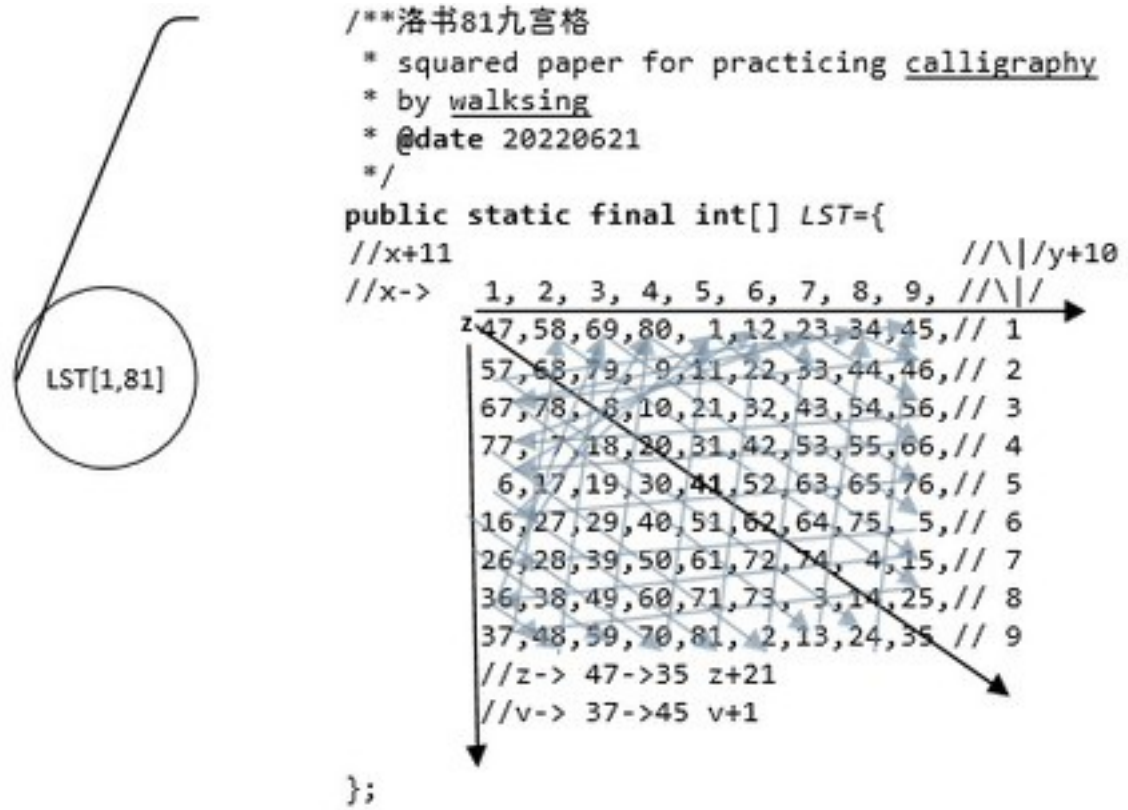


图2

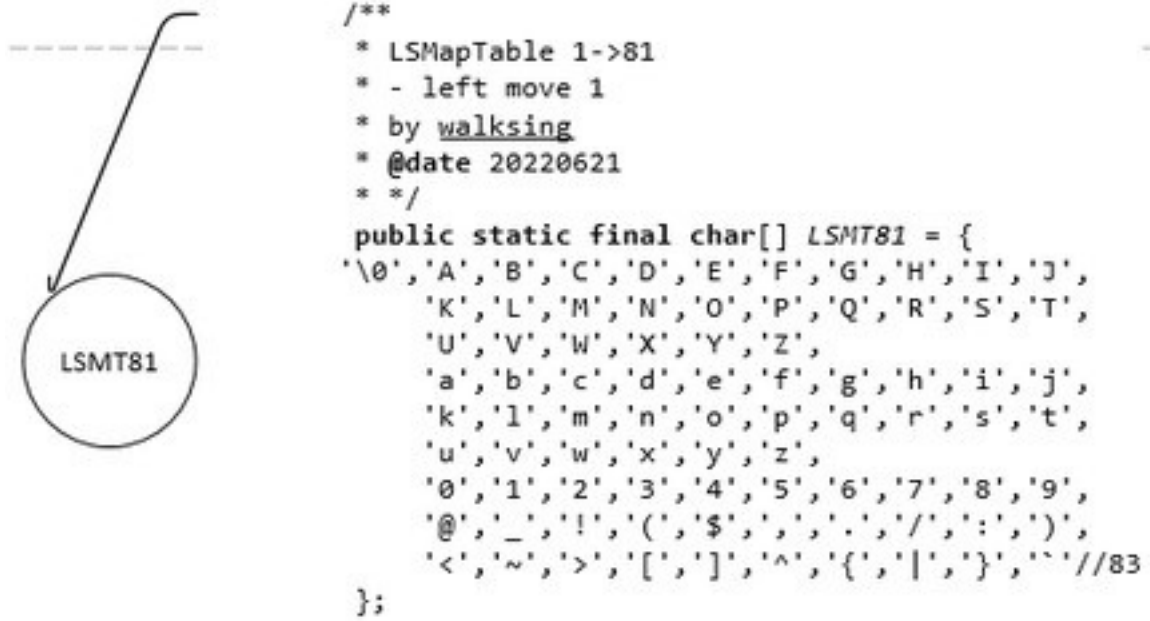


图3

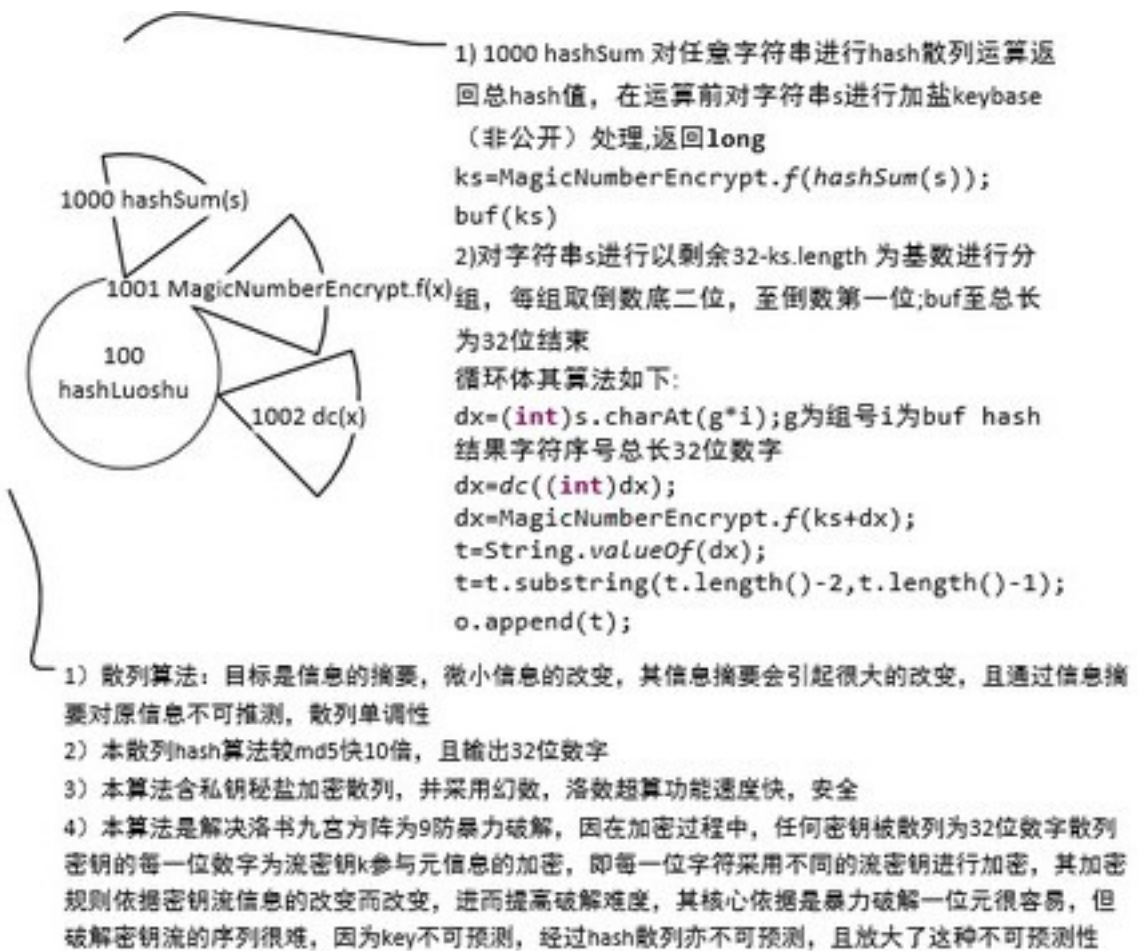


图4

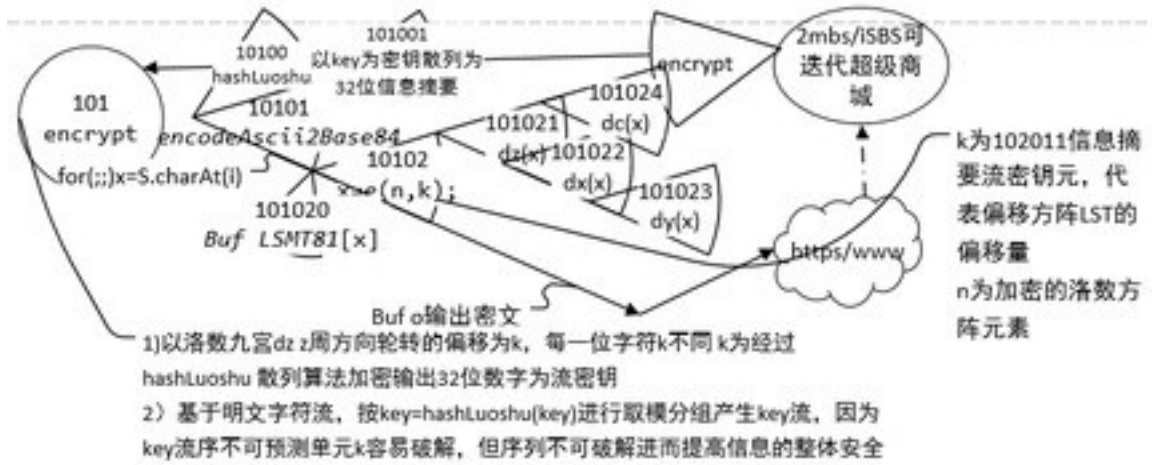


图5

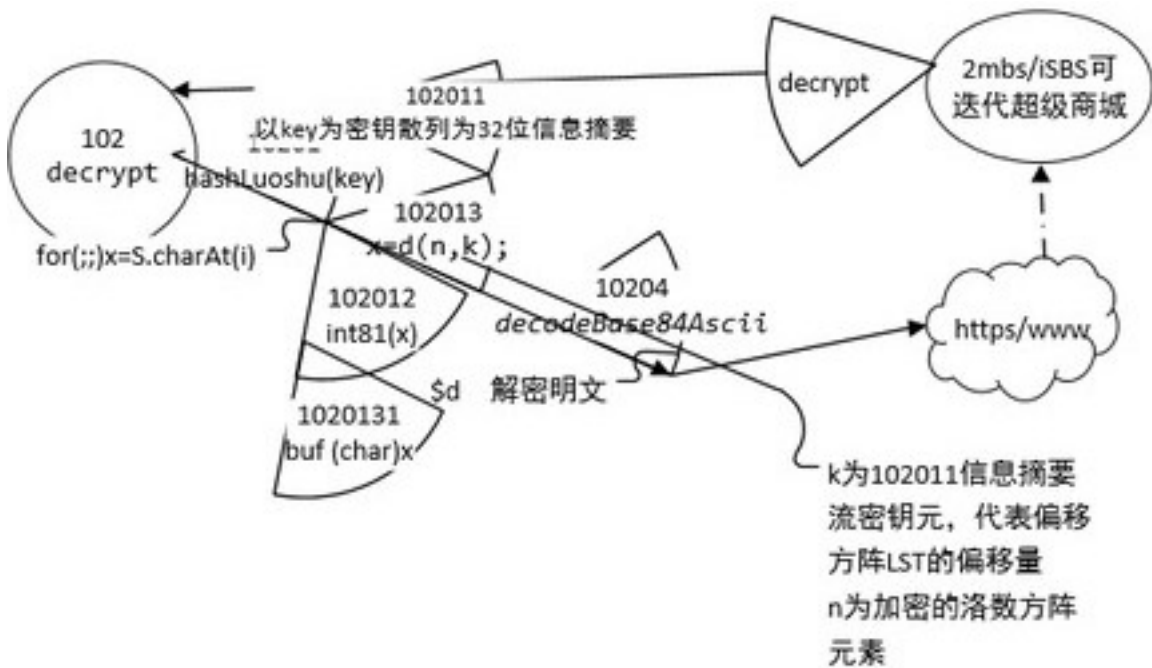


图6