マルチメディア通信と分散処理　85－21
（１９９７．１１．６）

神経細胞暗号方式

財団法人　鉄道総合技術研究所
〒１８５　国分寺市光町２－８－３８
長田　弘康

暗号技術に、数学と対応するものだけあるとすれば、代数的暗号、幾何的暗号、解析的暗号があると思われる。

代数的暗号は、ＲＳＡ社を代表として多数考案されている。

幾何的暗号は、最近、フラクタル暗号が考案された。

これに対して、丸め誤差を問題点として、今まで解析的暗号は考案されていない、このため、本文では解析的暗号の可能性を神経細胞に対する理論をもとに考案する。

Neuron Cryptography

Hiroyasu Osada
Railway Technical Research Institute
2-8-38 Hikari-cho, Kokubunji-shi, Tokyo, 185

If supposing that there is only one which corresponds to the mathematics in the cipher technique, there seems to be a cipher having to do with algebra, a cipher like several, an analyzing cipher. As for the cipher like several of a lot of ciphers having to do with algebra make RSA company a representative and are contrived., recently, a fractal cipher was contrived. Contrived therefore, on the other hand, the analyzing cipher contrives the possibility of the analyzing cipher

１．はじめに

暗号技術は基本的に整数に基く代数的なものである（文献１参照）。

また、最近、画像に対する暗号化としてフラクタル暗号が考案された（文献２、３、４参照）。

過去において、解析的暗号が発達しなかった理由は、浮動小数点形式が計算機毎に違うのが問題となり、丸め誤差が基本的問題となった点である。しかし、固定小数点数では問題がない。

解析的暗号とは、たとえば、３体問題が解析的には予測できないものを利用した暗号技術である。つまり、３つの玉（Ａ，Ｂ，Ｃ）を考え、まず、玉Ａは固定されているものとする。元信号により、玉の一つ（Ｂ）を震わせて（ｘ、ｙ、ｚ３方向の加速）、玉の他の

もの（C）の位置、速度で暗号とするものである。暗号のキーとなるのは、初期条件（Bのｘ、ｙ、ｚ座標と速度、Cのｘ、ｙ、ｚ座標と速度）、それぞれの玉の質量（A、B，Cの重さ）である。

初期条件、玉の重さを知り、数値解法を同一にしておけば、初期条件および玉の質量を知ることにより、玉Cの位置と速度から、玉Bの加速度（ｘ、ｙ、ｚ方向）が予想できる。これが基本的なアイデアである。

ただし、これでは、プログラムを組むのも大変なので簡単な解析的暗号技術として、神経細胞の方程式を利用したのが提案である。

２．神経細胞のモデル

神経細胞（ニューロン）モデルには沢山の種別があるが、ここでは最も単純なものを利用しよう（文献6､7参照）。数値としては３２ビットの固定小数点数を用い、ｂは(0<b<1)の値であり、0 <= a[n] <= 1である。

y[n+1] = b y[n] + a[n] -1 (if y[n] >= 0)

= b y[n] + a[n] (if y[n] < 0)

キーとしては、y[0],b を受け手、送り手で共にしているものとする。

３．エンコード

パラメータｂおよび初期値y[0]、が与えられており、前節の式を用いて、a[n]からy[n+1]へエンコードする。

手順：

（１）入力文字列を頭から３０ビットずつ読み出し、固定小数点数とする。。

（２）上記浮動小数点数を、a[0],a[1],a[2],...a[n]とする。

（３）キーとして、y[0]とｂを与える。ただし、(0<b<1)

（４）差分方程式

y[n+1] = b*y[n]-a[n] (y[n] < 0 の時)

b*y[n]-1-a[n] (y[n] >= 0 の時)

上記の式により、y[n+1]を求める。

（５）固定小数点列、y[1],y[2],....y[n+1]を文字列とみなす（４バイト毎）。

４．デコード

パラメータ y[0], b、が与えられており逆演算することにより、y[n]からa[n]へとデコードする。

手順：

（１）受信文字列を頭から４バイト毎に固定小数点化する。

（2）これを、y[1],y[2],.....,y[n+1]とする。

（3）キーとしてy[0]とbを与える。

（4）差分方程式

$$a[n] = b*y[n]-y[n+1] \quad (y[n] < 0 \text{ の時})$$

$$b*y[n]-1-y[n+1] \quad (y[n] >= 0 \text{ の時})$$

上記式により a[0],a[1],...,a[n]を求める。

（5）浮動小数点数 a[0],a[1],...,a[n]から３０ビットづつを取り出し、詰め合わせて文字列としてみなす。

これにより復号化できた。プログラムを付録に示す、このプログラムでは文献５を参照した。ただし、簡単にするため、平文は１６ビットであらわした。

５．バリエーション

他の神経モデル（たとえば、文献8,9）を用いるともっと複雑なものが可能である。つまり、

$$y[n+1] = C\,y[n] + \lambda[n] \quad (\text{if } y[n] < 0)$$

$$= C\,D\,y[n] - E(1-\lambda[n]) \quad (\text{if } y[n] >= 0)$$

ただし、ここで、0<C<1, 0＜CD＜1、0＜E、0＜＝λ[n]＜＝1とする。キーとなるのは、C,D,E,y[0]であり、入力はλ[n]で、出力はy[n+1]である。

この式の逆変換は、以下の通りである。

$$\lambda[n] = y[n+1] - C\,y[n] \quad (\text{if } y[n] < 0)$$

$$=(y[n+1] - CD\,y[n])/E + 1 \quad (\text{if } y[n] >= 0)$$

等を用いる。このとき、入力はy[n+1]であり、出力はλ[n]である。キーは前の式と同じである。

６．その他

前提条件として、送り手、受け手とも暗号キーy[0]と b を知っていることがある。これらのキーは６バイトの浮動小数点数である。従って、２の６１乗程度の組み合わせがある。これに対してＲＳＡ社等の代数方式だと、これ以下になるのは常識的に分かる。

７．結論

この暗号技術では、浮動小数点演算をそろえておけば、どのようなマイコンでも可能であり、インプリメントも容易である。この暗号技術が代数的暗号技術が素数に基づくものなので、その数がキーの長さに対して n/ log(n-B)の大きさでしか増えないために、いずれ解読されてしまう問題があるのに対して、文献６、７、８、９で見るように、入力が固定してさえいてもカオスになっているために予測ができない点である。フラクタル暗号は有望であるが、基本的に画面に特化している問題がある。代数暗号はスクランブルをかける

ためにＤＥＳ(Data Encryption Standard)の技術を使用しているが、神経細胞暗号では、スクランブル技術は不要である。この技術は、通常の暗号ばかりでなく、クライアント・サーバーにおけるキャッシュの暗号化にも適用できる。（文献10参照）

参考文献：

(1)Dorohy Elizabeth Robling Denning : Crypograpy and Data Security, Addison-Weley,1982

(2)Davern,P & Scott,M:"Fractal Based Image Steganography",V.1174,pp279-294,Lect Notes Comput Sci(1996).

(3)Jan J-K, Tseng Y-M:"On the security of image encryption method",Inf Process Lett, V.60,N.5,261-265(1996)

(4)Alexopoulos C, Ionanou N:"Image encryption method using a class of fractals", J Electron Imaging,V.4,N.3,pp251-259(1995)

(5)JohnHauser:"SoftFloat"http://HTTP.CS.Berkeley.EDU/~jhauser/arithmetic/softfloat.html,1997年８月

(6)南雲仁一、佐藤俊輔：“神経モデルの応答特性について”、電子通信学会論文誌、Ｖｏ１５５－Ｄ，Ｎｏ．４、ＰＰ２６９—２７６

(7)J.Nagumo and S.Sato:"Respones Characteristic of Mathematical Neuron Model",Kybenetick,10,155-164(1972)

(8)長田弘康、吉沢修治、南雲仁一：“神経モデルのカントール関数型応答特性に関する理論的考察”、電子通信学会論文誌、Ｖｏｌ６４、ＮＯ．７、ＰＰ５６６—５７３

(9)S.Yoshizawa,H.Osada, and J.Nagumo: "Plus Sequences Generated by a Degenerate Analog Neuron Model", Biol. Cybern. 45,23-33(1982)

(10)長田弘康：クライアント・サーバーにおける検索キャッシュの提案、マルチメディア通信と分散処理ワークショップ、１９９６年１０月

参考：

プログラムとしては次のようになる。

```
/* Neuron Cryptography (c) Hiroyasu Osada RTRI  */


#include <stdio.h>
#define DEBUG 1
typedef unsigned int fix32;
typedef unsigned short int fix16;
fix32 fix32_one;
```

```
fix16 extract_fix16(fix32 a) { return (a >> 14) & 0xFFFF;}
fix32 fix32_mul(fix32 a, fix32 b)
{
    fix16 aHigh, aLow, bHigh, bLow;
    fix32 z0, zMiddleA, zMiddleB, z1;
    aLow = a;      aHigh = a>>16;
    bLow = b;      bHigh = b>>16;
    z1 = ( (fix32) aLow ) * bLow;
    zMiddleA = ( (fix32) aLow ) * bHigh;      zMiddleB = ( (fix32) aHigh ) * bLow;
    z0 = ( (fix32) aHigh ) * bHigh;      zMiddleA += zMiddleB;
    z0 += ( ( (fix32) ( zMiddleA < zMiddleB ) )<<16 ) + ( zMiddleA>>16 );
    zMiddleA <<= 16;      z1 += zMiddleA;
    z0 += ( z1 < zMiddleA );
    return z0;
}
void ency(fix32 b, fix32 init, fix16 *a, fix32 *y)
{
  int n;   fix32 aa;
  y[0] = init;
  for (n=0;n<31; n++) {
    aa = ((fix32)(a[n]) << 14 ) & 0x3FFF0000;
    if (y[n] & 0x80000000)          y[n+1] = fix32_mul(b , y[n]) - aa;
    else                            y[n+1] = fix32_mul(b,y[n]) - (fix32_one + aa);
#ifdef DEBUG
    printf("Ency a[%d] = %8x, y[%d] = %16x¥n", n, a[n],n+1,y[n+1]);
#endif
  }
}
void decy(fix32 b, fix32 init, fix32 *y, fix16 *a)
{
  int n;

  y[0] = init;
  for (n=0;n<31;n++) {
    if (y[n] & 0x80000000)          a[n]=extract_fix16(fix32_mul(b,y[n]) - y[n+1]) ;
```

```
    else                    a[n]=extract_fix16(fix32_mul(b,y[n]) - (fix32_one + y[n+1]));
#ifdef DEBUG
    printf("Decy a[%d] = %8x, y[%d] = %16x¥n", n, a[n],n+1,y[n+1]);
#endif
  }
}

void main(int argc, char *argv[])
{
  char text[80];
  char result[160];
  fix32 b, init;
  int size,i;

  if (argc != 3) { fprintf(stderr, "usage; nuf b init¥n"); exit(1);}
  sscanf(argv[1], "%d", &b);
  sscanf(argv[2], "%d", &init);
  if (b & 0xC0000000) {
    fprintf("b is too large¥n");
    exit(1);
  }
  gets(text);
  size = strlen(text);
  if (size > 80) {fprintf(stderr, "Too large text¥n"); exit(1);}
  fix32_one = 0x40000000;
  ency(b, init, (fix16 *)text, (fix32 *)result);
  printf("Cry Result = ");
  for(i=1;i<160;i++) printf("%2d,", result[i]);
  printf("¥n");
  decy(b, init, (fix32 *)result, (fix16 *)text);
  printf("Decy Result =");
  puts(text);
  printf("¥n");
}
```