コンピュータセキュリティ　14－2
（2001．7．25）

多重アファイン鍵システムと擬似乱数の周期

鈴木秀一

東京電機大学

〒270-1382 千葉県印西市武西学園台 2-1200
ssuzuki@chiba.dendai.ac.jp

概 要

多重アファイン鍵システムによって擬似乱数を生成した場合，生成された擬似乱数から鍵を特定できないなどの暗号学的な安全性がある．しかしその擬似乱数の周期を数学的に証明できないという問題点も指摘できる．拙論では，周期を求められる既知の擬似乱数生成法と多重アファイン鍵暗号の一形式である「暗号の塔」のアイディアと組み合わせることにより，擬似乱数の周期を明示できる擬似乱数の生成法提案する．

キーワード
多重アファイン鍵システム，加法的乱数生成，擬似乱数，暗号の塔

Multi affine key cipher and the period of pseudo random number

Shuichi Suzuki

Tokyo Denki University

2-1200 Muzai-Gakuendai, Inzai-City,Chiba 270-1382, Japan

Key Words
multi affine key system, additive generator of pseudo random number, pseudo random number, tower of cipher.

1　加法的擬似乱数生成と暗号学的安全性

LFSR などによる擬似乱数生成法は非常に高速であるが，少数の擬似乱数のデータから連立一次方程式を解くことにより秘密鍵に対応する特性多項式の係数を確定できるので，暗号学的に安全でない．このため，加法的な擬似乱数生成法が直接ストリーム暗号などに使用されることはない．しかし，既約多項式に対応してその擬似乱数の周期が比較的容易に導かれるなど優れた性質も持っている．

[2, 4] には，次のような擬似乱数生成法が示されている．

- x_j 32 ビット整数

- $x_0,\ x_1,\ \cdots,\ x_{99}$ を適当に定める.
- $x_{100+j} = x_{55+j} - x_j \bmod 2^{32}$
- この $\{x_j\}$ の周期は $2^{31}(2^{100}-1)$ である.

このような擬似乱数生成法を暗号に活用したい. このために多重アファイン鍵システムを活用できることを以下に示す.

2 多重アファイン鍵システムと暗号の塔

[5, 6] では暗号学的に安全な擬似乱数生成法としての多重アファイン鍵システムと様々なバージョンの多重アファイン鍵暗号と「暗号の塔」を提案した.

多重アファイン鍵システムを $\mathcal{K} = \{\{K_i\}, v, w(i,j)\}$ $(K_i = \{a,b,c,n\})$ とする.

まず、ここで扱う整数は全て 32 ビット符号無し整数である。整数 $x,\ a,\ b$ に対して、アファイン演算 $ax+b$ は特に断らない限り、

$$((ax+b)\ \texttt{shr 16})\ \texttt{xor}\ (ax+b)$$

を表すものとする。この演算は **stable operation** と言うことがある。stable operation はアファイン演算の上位 16 ビットを下位 16 ビットに排他的論理和したものである。これは mod 2^{32} 演算で、下位ビットから少しずつ 0 が蓄積して行く傾向が認められるので、これを良質な上位のデータで拡散する意味がある。 mod 2^{32} 演算とは、実際には mod 演算をまったく使用しないことを意味する。

アファイン鍵 $K = \{a,\ b,\ c,\ n\}$ は 4 個の整数からなる構造体である。アファイン鍵 K の整数 x への作用は

$$K(x) = ax + b$$

で定義する。ここでの演算はもちろん stable operation である。c はこのアファイン鍵が何回使用されたかをカウントするカウンタである。n はこのアファイン鍵を使用できる回数の上限である。これをアファイン鍵の**寿命**という。通常は $n = 3$ に設定する。部分的に大きな寿命を設定すると暗号は高速化されるが、安全性が低下する傾向が出てくる。

多重アファイン鍵システム $\mathcal{K} = \{K_0,\ K_1, \cdots, K_{M-1},\ v,\ w(i,j)\}$ は、複数のアファイン鍵 $K_0,\ K_1, \cdots,\ K_{M-1}$ とその鍵の間の相互作用を定義するプロシージャ $w(i,j)$ の組である。$w(i,j)$ は i 番目の鍵を j 番目の鍵で書き換えることを意味する。v は鍵回転パラメータといわれる整数である。**鍵回転モデル**といわれる暗号化アルゴリズムを以下に示す。$w(i,j)$ は例えば次のようなものを使用する。ここでは鍵の個数は 32 個としている。

「暗号の塔」[5] は, この多重アファイン鍵システムを複数の階層として積み上げ, アファイン鍵の書き換え $w(i,j)$ を因果関係のない上位の層で生成される擬似乱数で書き換えるものである.

```
procedure w(i,j:integer):integer;
begin
   K[i].a:=(K[i].a*K[j].a+K[j].b) or 1;
   j:=(j+1) and 31;
   K[i].b:=(K[i].b*K[j].a+K[j].b) or 2;
end;
```

このようなプロシージャを用いて以下のアルゴリズムで多重アファイン鍵暗号を定義する。

1. i=0, k=1, v=0
2. x0=K[i](x0) , K[i].c=K[i].c+1
3. c[k]=m[k] xor (x0 and 65535) (encription)
4. j=(x0 shr 16) and 31
5. if K[i].c $\geq$ K[i].n then
   v=(v+1) and 31, j=(j+v) and 31, w(i,j) , K[i].c=0
6. i=j, k=k+1 goto 2.

このモデルは非常に高速であり、弱鍵も現在まで一例も見出されていない。弱鍵は多数のアファイン鍵の一部のみを集中的に使用する状況で発生する。このような状況になると鍵回転パラメータ v は多数のアファイン鍵を強制的に全てを使用するように作用する。

これらの暗号は以下のような単純な根拠で暗号学的に安全である.

- 各アファイン鍵は 2WORD の係数 a, b (合計 4WORD) の未知数を持つ.
- 各アファイン鍵は 3WORD の擬似乱数を生成すると他の鍵で書き換えられる. 鍵を書き換える情報はこのシステムの外部には出力されない.
- a は奇数なので, 4WORD の未知数を持ち 3 個の連立方程式は実際に 1WORD の不定性を持つ.
- このことから多重アファイン鍵暗号は使用される鍵の順番を特定できなければ解読できない. このことは生成される擬似乱数の下位 1WORD と上位 1WORD の間に明白な相関が存在するときにのみ可能である. 通常, このような現象は観測できない. このような現象は弱鍵を使用した場合にのみ観測され得る.
- 実用的な多重アファイン鍵システムは 32 ビットのアファイン鍵を 32 個以上使用する. これに対して, 8 ビットのアファイン鍵を 8 個使用する非常に弱いミニチュア版の多重アファイン鍵システムを 2001 年 3 月から公開しているが, このミニチュア版にすら弱鍵は見出されていない.

多重アファイン鍵システムが生成する擬似乱数の不満な点は一般には, その周期が数学的に明示されないことである. しかし以下に示すように, その中のあるバージョンでは周期を明確に評価できることを示そう.

3 暗号の塔に加法的乱数生成法を組み込む

高さが 2 階の「暗号の塔」[5] を考える. これは擬似乱数を出力する多重アファイン鍵システム $\mathcal{K}_1$ の鍵の書き換えをまったく別の擬似乱数生成システム $\mathcal{K}_2$ で行うものである. この $\mathcal{K}_2$ を周期が確定できる擬似乱数生成法にすり換えればよい. $\mathcal{K}_1$ が暗号学的に安全なので $\mathcal{K}_2$ は高速な加法的擬似乱数生成法を用いることができる. 特にある種の加法的生成法は統計的にも優れた性質を持っているので効果的である.

拙論の最初に述べた [2, 4] の方法を用いてもよい. 単純に LFSR を使用することもできる. また数列の番号を高速に処理するには次の方法もある.

- x_j 32 ビット整数
- x_0, x_1, $\cdots$, x_{127} を適当に定める (少なくとも一つは奇数).
- $x_{128+j} = x_{65+j} - x_j \bmod 2^{32}$

この場合に対応する特性多項式は $x^{128}-x^{65}+1$ である．この方法で生成される擬似乱数の周期は，最下位ビットの周期によって容易に評価できる．特性多項式は mod 2 で次のように因数分解できる．

$$(1+x^2+x^6+x^8+x^{10}+x^{12}+x^{14}+x^{16}+x^{20}+x^{22}+x^{33}+x^{35}+x^{39}+x^{41}+x^{43}+x^{45}+x^{47}+x^{51}+x^{53})\times$$

$$(1+x^2+x^4+x^{10}+x^{14}+x^{16}+x^{22}+x^{28}+x^{30}+x^{33}+x^{34}+x^{35}+x^{37}+x^{40}+x^{42}+x^{43}+x^{44}+x^{47}+x^{51}+x^{53}+$$

$$x^{55}+x^{57}+x^{61}+x^{65}+x^{71}+x^{73}+x^{75})$$

このことからこの方法で生成される擬似乱数の周期は $(2^{53}-1)(2^{75}-1)$ より小さくないことが分かる．

特に，$\mathcal{K}_1$ のアファイン鍵の寿命を 1 に設定すると，4WORD の不定性に対して，2WORD の擬似乱数を暗号化に使用しても安全なので，2WORD 単位で暗号化することにすると，アルゴリズムも単純化され，さらに高速化される可能性がある．このアルゴリズムを以下に示す．

```
function randomK2:longword;
begin
    x[(128+i) and 127]:=x[(65+i) and 127]-x[i and 127];
    i:=(i+1) and 127;
    randomK2:=x[(128+i) and 127];
end;

        procedure w(i,j:integer);
        begin
           K[i].a:=randomK2 or 1;
           K[i].b:=randomK2;
        end;
```

ここで random(K2) は $\mathcal{K}_2$ が生成する擬似乱数である.。

1. i=0, k=1
2. x0=K[i](x0)
3. c[k]=m[k] xor x0 (encription with width 2WORD)
4. j=randomK2 and 31
5. w(i,j)
6. i=j, k=k+1 goto 2.

このアルゴリズムで生成される擬似乱数にビットごとの 01 頻度検定を実行してみた．検定は 10000 ビットごとの検定を 1000000 回実行した．他のアルゴリズムと比較しても真正乱数と区別できる事実は見出せなかった．

アルゴリズム	平均	分散
多重アファイン鍵+$x^{128}-x^{65}+1$	0.500007252	0.000024987
RC4	0.499990643	0.000024973
Mersenne Twister	0.500010625	0.000025030
8bit 8Keys multi affine	0.500083274	0.000025844
32bit multi affine	0.499995386	0.000025010

参考文献

[1] 盛合，宮野，下山:「多重アファイン鍵暗号について」，SCIS2000 技術報告集 (沖縄)，2000.

[2] Donald, E. Knuth: The Art of Commputer Progaram 3「準数値算法/乱数」, サイエンス社, 1993.

[3] 岡本龍明, 山本博資：「現代暗号」, 産業図書, 1997.

[4] 松本眞, 西村拓士：「擬似乱数の重み分布による検定」，RIMS, 2001.

[5] 鈴木秀一:「秘密鍵を特定できない暗号」, 電子情報通信学会技術報告集, ISEC2000-34,2000-7.

[6] 鈴木秀一:「多重アファイン鍵暗号とその周辺技術」, 東京電機大学工学部研究報告 48 号, 2000.