バーナム暗号に基づく改良暗号法

大前　義次*・高橋　貞良*・大瀧　勝久**

Improved cryptograph based on Vernam cipher

Yoshitsugu OHMAE*, Sadayoshi TAKAHASHI*
and Katsuhisa OHTAKI**

Abstract

In this paper, traditional Vernam cipher is investigated and the improved cryptograph having a procedure-open-type algorithm with a new mixing function in the key generation part of the Vernam cipher is proposed. The main improvement point is to generate a non-periodic long random sequence and to attain the high security, by using the AMIDA lottery structure for the part of generation on the key stream of the Vernam cipher.

In this method, the entrance of the AMIDA structure and also the random seed of the random sequence generator situated at the exit of the AMIDA structure, are changed after each one block encipher (decipher).

Key words: Vernam cipher, Improved cryptograph, Stream cipher, AMIDA structure

あ ら ま し

本研究では，古くから使われているバーナム暗号について検討し，バーナム暗号の鍵生成部に新たな攪拌機能を付けることにより改善した新しい逐次暗号方式として改良暗号法を提案する。

主な改良点は，バーナム暗号の鍵ストリーム生成部にあみだ構造を導入し，1ブロック暗号化(複号化)ごとに，あみだ構造の入口を変え，またあみだ構造出口にある乱数生成器のシードを変えることによって，非周期的な長い乱数列を生成させ，安全性を高めるものである。

キーワード：バーナム暗号，改良暗号法，逐次暗号法，あみだ構造

1993年7月10日受理

* 神奈川工科大学情報工学科，厚木市
Department of Information and Computer Sciences, Kanagawa Institute of Technology, 1030 Shimo-ogino, Atsugi, 243-02 Japan

** ㈱西武百貨店，東京都
Seibu department store Co. Ltd., Tokyo

1. ま え が き

今日情報セキュリティが重要な問題となっている。情報セキュリティを保護するもっとも有効な手段の一つとして暗号がある。バーナム暗号はアルゴリズムを秘匿して使用するアルゴリズム秘匿型の逐次暗号方式である。これに対して，DES暗号やFEAL暗号は，アルゴリズムを公開して鍵を秘密にするアルゴリズム公開型のブロック暗号方式である。本研究では，古くから使われているバーナム暗号について検討し，バーナム暗号の鍵生成部に新たな攪拌機能を付けることによりこれを改善し，アルゴリズムを公開して使用するように変更した逐次暗号方式(以下改良暗号法という)を提案する。

主な改良点は，バーナム暗号の鍵ストリーム生成部にあみだ構造を導入し，1ブロック暗号化(復号化)ごとに，あみだ構造の入口を変え，またあみだ構造出口にある乱数生成器のシードを変えることによって，非周期的な長い乱数列を生成させ，安全性を高めるものである。

2. 逐次暗号における擬似乱数生成法

逐次暗号は平文 M を連続した文字ないしビット $m_1, m_2, \ldots,$ に分割して，m_i の暗号化には鍵ストリーム $K=k_1, k_2, \ldots,$ の第 i 番目の要素 k_i を適用する。すなわち暗号化は

$$E_K(M)=E_{k1}(m_1)E_{k2}(m_2)\ldots$$

逐次暗号に使う鍵ストリームの中に，一定数の文字ごとに繰り返しがあれば周期暗号であり，鍵に繰り返しがなければ非周期暗号である。

バーナム暗号は逐次暗号の一種であり，鍵を使い捨てにする非周期暗号である。

バーナム暗号は鍵ストリームの各ビットが，0.1等頻度でかつ独立に生成される非周期的なランダム系列で平文と等しいビット長である限り，暗号文から平文を知ることは原理的に不可能であり，その意味で完全な安全性を備えているといえる。しかしながら，超機密データを扱う場合を除き，実際問題として，平文の量と同じ量の鍵ストリームを別に受信先に送ることは非現実的である。そこで見かけ上ランダムなビット列を生成する多くの方法が考え出されている[1,2]。

逐次暗号のうち同期逐次暗号は，鍵ストリームを平文ストリームと独立に生成する。したがって，伝送中に暗号文の1ビットが他のビットに変化しても同期はずれないし，エラーの波及はないが暗号文の1ビットが消滅したら，そのビット以降のすべての暗号文に対し同期がずれ，誤りが伝搬することになるので，送信側と受信側で同期をとり直す必要がある。

同期逐次暗号としてこれまでに線形フィードバックシフトレジスタ方式，出力フィードバック方式，カウンタ方法などが考えられている[1]。

逐次暗号では，鍵生成の第1段階として初期シード（I_0）を入力して，長い擬似乱数列に変換して，それを使い捨て乱数として使用する。このやり方のうち線形フィードバックシフトレジスタ方式では，シフトレジスタを用い，フィードバックループによって，I_0 を長い擬似乱数列に変換してそれを使い捨て乱数に近似しようとするものである。しかし生成される各段階のレジスタの状態が線形性をもつため安全性が必ずしも高くない。これを改善するため非線形フィードバック方式が考え出された。この方式は，シフトレジスタの出力を非線形ブロック暗号化アルゴリズムの入力にフィードバックさせ，その出力の一部を鍵ストリームとするものである。

カウンタ方式は非線形ブロック暗号化アルゴリズムを用いるが，その出力結果を繰り返し循環させることをしないで乱数鍵生成器のカウンタを I_0+i-1（$i=1, 2, \ldots$）として，i 番目の鍵 K_i を生成する。

カウンタ方式はこのように以前の内部状態に依存しない方式であり，フィードバック方式より処理は効率的である。

同期式逐次暗号では，平文の内容が同一でもブロックが異なれば，暗号化鍵が異なるので，暗号解読を防御できる。また，暗号文中に外部から削除や挿入などの手を加えた場合は同期がとれなくなるという欠点がある。しかし，これは考えようによっては，この種の妨害を防ぐことができ利点ともいえる[1]。

以上のようにこれまで研究され発表されている逐次暗号方式に対して，本論文で提案する改良暗号法はバーナム暗号の鍵ストリーム生成部に，あみだ構造を導入し1ブロック暗号化（復号化）ごとにあみだ構造の入口を変える，またあみだ構造の出口の，乱数生成器のシードを変えるという方法によって非周期的な長い乱数列を効率的に生成させるものである。同種の研究としては，出力結果の循環をさせないで，乱数鍵生成器のカウンタを進めて鍵を生成するカウンタ方式に類似している方式がある。

3. バーナム暗号のアルゴリズム

バーナム暗号は，2進符号で表された入力データに2進乱数の鍵を排他的論理和で合成し，送信する。暗号化の式は次の通りである。

$$C_i=P_i+K_i \qquad (3.1)$$

ここで C_i は暗号文，P_i は平文，K_i は鍵である。受信側でも同じ鍵を用意し，その乱数系列と受信データとの排他的理論和によって，暗号を平文に翻訳することができる。復号化は次の通りである。

$$P_i=C_i+K_i \qquad (3.2)$$

バーナム暗号の原理を図3-1に示す。

バーナム暗号を完全暗号系にするには使用ずみの乱数系列を再使用できないようにするという厳しい条件があるため，コンピュータ暗号として完全系を実現するのは難しい。

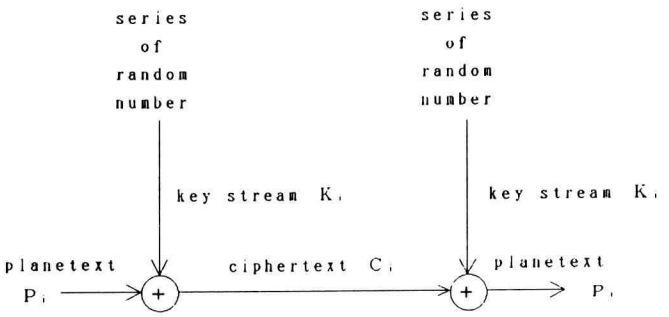


図 3-1.　バーナム暗号の原理

4.　改良暗号法とそのアルゴリズム

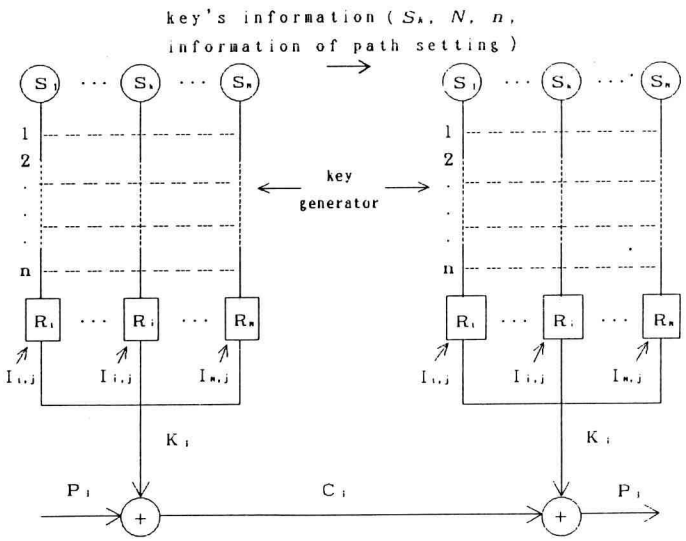
4.1　改良暗号法

バーナム暗号は平文と同量かそれ以上の鍵を必要とし，その鍵が確実に秘匿され，かつ1回限りの使用でなければならないという制約がある。このように実用性や経済性を無視したときのみ完全暗号系を構成する。このような欠点を改善するために，本研究ではあみだ構造を取り入れた方式を提案する。

この方式は，鍵として，次の6種類の要素を用いる。

(1)　あみだの本数，N

(2)　段数，n



S_k : start point ($1 \leq k \leq N$)

R_i : random number generator ($1 \leq i \leq N$)

$I_{i,j}$: random seed ($1 \leq i \leq N,\ 0 \leq j \leq M$)

K_i : key stream ($1 \leq i \leq N$)

N : number of files

n : number of stairs

M : maxmum value of seeds

図 4-1.　改良暗号法の原理

(3)　パス生成のための乱数シード
(4)　あみだ構造への入り口，
　　S_k ($1 \leqq k \leqq N$, N：最大あみだ本数)
(5)　あみだ構造の出口の乱数生成器，
　　R_k ($1 \leqq k \leqq N$)，乱数列生成シード
　　$I_{i,j}$ ($1 \leqq i \leqq N$, $0 \leqq j \leqq M$, M：最大シード)
(6)　1ブロック暗号化（復号化）文字数

これら6種類の要素を設定し，送信側，受信側で管理する必要がある。このことはバーナム暗号の一連続の長い鍵の管理に比べて決して難しいとはいえない。

4.2　改良暗号法のアルゴリズム

鍵生成部にあみだ構造を使用し，暗号化はバーナム暗号の逐次暗号法を踏襲する。改良暗号法の原理を図4-1に，改良暗号法のアルゴリズムを図4-2に示す。本暗号法ではブロック長は任意の長さに設定できるが，図4-2は64ビット単位にブロック化した場合を示す。

まず最初に，あみだ構造の縦の本数 N，横の段数 n，およびあみだ構造のパスを生成するための乱数生成器の乱数シードを設定する。この設定値にもとづいてあみだを生成する。ここで生成したあみだ構造は，一連

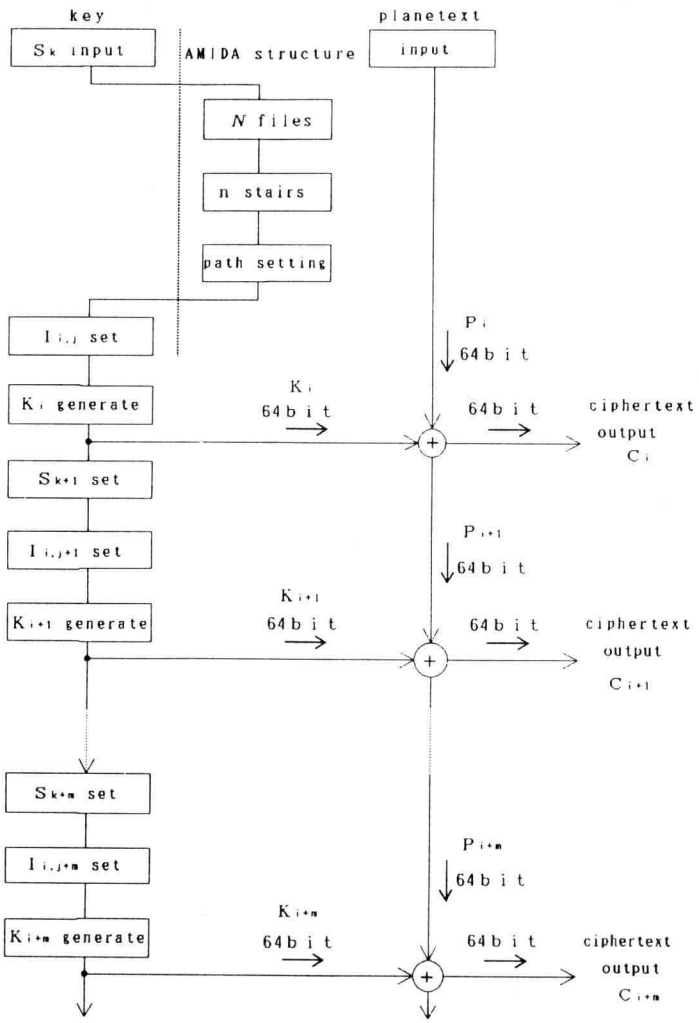


図4-2.　改良暗号法のアルゴリズム

続の暗号化（または復号化）の最初から最後まで使用する。あみだ構造の設定が終了した後，入口 $S_k(1 \leqq k \leqq N)$ にもとづいてあみだ構造の出口の乱数生成器 R_k が選ばれる。これにより1ブロック目の暗号化(復号化）のための擬似乱数列 $K_i(1 \leqq i \leqq N)$ を発生させる。

1ブロック暗号化（復号化）実施ごとに次を実行する。

(1) あみだ構造の入口を S_k から S_{k+1} にかえる。
(2) あみだ構造の出口の乱数生成器のシードを $I_{i,j}$ から $I_{i,j+1}$ に変える。

これによって，非周期的な長い乱数列を生成させるものである。

暗号化に当たって，平文を文字ブロック(8ビット×文字数）単位に暗号化する。本方式は，暗号化と復号化は反転式であり，復号化のときも，暗号化と同じ手順で行われる。

5. 安全性評価のための基準

暗号用乱数列の安全性評価の基準としては次のものが考えられる[3]。

(1) 発生シンボル 0, 1 の等頻度性，無相関性
(2) 乱数列発生の非線形性
(3) 乱数列としての長周期性
(4) 原理的あるいは計算量的な予測不能性

(1)に関して本論文では，ランダム性評価を理想二項分布近似度によって行っている。

(2)に関して本研究では，鍵生成過程で次の二つの方法をとることによって対応している。すなわち1ブロック暗号化（復号化）ごとに，

① あみだ構造の入口を変える。
② あみだ構造の出口の乱数生成器の乱数シードを変える。

(3)に関しては乱数シードの上限（M）によって周期が決まる。

(4)に関しては最も強力な評価基準であるがこれをこの場合具体的に示すことは困難である。

6. 各暗号方式の検討結果

以下の検討ではバーナム暗号，DES暗号，FEAL-8暗号，および改良暗号法のいずれに対しても，入力データとしてはASCIIコードをもちいる。

DES暗号およびFEAL-8暗号の入力データの単位が64ビット（半角文字で8文字）であるので，各暗号方式の比較検討のための入力データ単位を64ビットとする。改良暗号法は，基本的には1ブロック当り任意ビットの暗号化を行うことができるがここでは64ビットずつまとめ比較する。入力データ単位が64ビット(8文字)に足りないときは，不足文字分を空白文字(ASCIIコード20H)でうめる。ここで用いた入力サンプルデータは33524(ブロック)×64ビットである。

暗号化の所用時間，理想二項分布近似値 θ_v，理想二項分布近似率の加重平均 $\bar{\theta}$ を求める。ここで暗号強度の比較に θ_v と $\bar{\theta}$ を用いているが，この理想二項分布近似率については文献5）による。

表6-1. バーナム暗号の8ビット層別化の場合

	暗文の層別化									X	·Y	θ
	0	1	2	3	4	5	6	7	8			
0	341	0	0	0	0	0	0	0	0	0.0	0.0	0.0
原 1	0	424	0	0	0	0	0	0	0	1.0	0.0	0.0
文 2	0	0	1094	0	0	0	0	0	0	2.0	0.0	0.0
の 3	0	0	0	1424	0	0	0	0	0	3.0	0.0	0.0
層 4	0	0	0	0	782	0	0	0	0	4.0	0.0	0.0
別 5	0	0	0	0	0	239	0	0	0	5.0	0.0	0.0
化 6	0	0	0	0	0	0	18	0	0	6.0	0.0	0.0
7	0	0	0	0	0	0	0	0	0	0.0	0.0	0.0
8	0	0	0	0	0	0	0	0	0	0.0	0.0	0.0

6.1　バーナム暗号

6.1.1　平文の8ビット層別化の場合

m ビットの平文 P における変化の集合を δP で表し，成分0の零ブロック ϕ_m とのハミング距離 $\nu=H(\delta P, \phi_m)$ ごとに，次の通り層別化する。

$$\delta P=\bigcup_{\nu}\delta P_\nu \tag{6.1}$$

ただし

$$\delta P_\nu=\{x \mid x\in\delta P,\ H(x, \phi_m)=\nu\} \tag{6.2}$$

平文が δP_ν だけ変化したときの n ビットの暗号の変化を δCP_ν とし，成分0の零ブロック ϕ_n とのハミング距離の変化値について次を定義する。

$$\Omega_{P_\nu}=\{r \mid r=H(\delta C_{P_\nu}, \phi_n)\} \tag{6.3}$$

Ω_{P_ν} から層別サンプルを取り出して平均 X，分散 Y を求め，これを用いて理想二項分布近似度を求める。

サンプルデータとして，平文に英文を用い，平文と暗文のそれぞれのハミング距離による8ビット層別化を行ったときの結果を表6-1に示す。この場合鍵は，8ビットに固定してある。

表6-1より明らかなように，同じ鍵を繰り返して使用すると，平文のハミング距離による層別化の結果がそのまま暗号文にも出てきてしまう。これは，バーナム暗号では，単に平文と鍵（乱数列）の排他的論理和しか行わないからである。

ここで平文の層別化が6ビットまでしかないのは，データとして8ビットの英文字（ASCIIコード）をいれたためである。通常の英文では，これ以上の層別化は無理であると考えられる。

6.1.2　平文の64ビットブロック化の場合

8ビット層別化の場合と同様に平文の層別化がそのまま出てくる。このため Y および θ は0となる。理想二項分布近似率の加重平均 $\bar{\theta}$ は，θ がすべて0なので $\bar{\theta}=0$ となる。

この場合のパソコンでの処理時間は，4秒かかる。バーナム暗号はただ排他的論理和をとるだけなので，33,524（ブロック）×64ビットものデータであってもこのような時間で暗号化できる。

6.2　改良暗号法

改良暗号法では，まずあみだ構造が串刺し型の場合と，短刺し型の場合の2種類についてそれぞれ，10段，20段，30段を比較した。串刺し型と短刺し型の定義並びに特性については付録に示す。

条件として，すべてについて

(1)　あみだの本数 N を10本

(2)　あみだの入口 S_k を左から2本目

(3)　あみだのパス生成のためのランダムシードを0

(4)　1ブロックあたりの文字数を8文字とする。

1ブロックあたりの平文の入力文字数を，8文字(64ビット）ごと，または改行文字まで読み込み8文字に足らない不足文字数分を空白文字（ASCIIコード20H）で埋める。

表6-2.　改良暗号法串刺し型10段の検討結果

原文層別化	ブロック数	X	Y	θ
1	14	32.4286	12.1020	0.6034
2	7	34.2857	23.3469	0.2048
3	12	30.7500	2.5208	0.0858
4	12	31.2500	18.8542	0.6101
5	15	32.6667	32.2222	0.0000
6	15	32.0667	14.7289	0.8513
7	21	31.5238	19.5828	0.5301
8	26	31.4615	14.7101	0.8634
9	39	31.8718	18.0605	0.7374
10	72	31.9028	16.4211	0.9441
11	106	32.0472	16.2525	0.9700
12	185	32.1784	15.7357	0.9623
13	307	31.5570	14.9699	0.8900
14	534	31.7285	15.7296	0.9748
15	785	31.7962	17.5406	0.7997
16	1184	32.0583	15.4839	0.9359
17	1655	31.8773	17.1263	0.8548
18	2075	31.7436	16.3256	0.9508
19	2605	32.0856	16.1059	0.9895
20	3000	32.0103	15.8889	0.9859
21	3142	31.9764	16.0255	0.9961
22	3232	31.8769	16.3122	0.9569
23	2958	32.0358	16.0346	0.9968
24	2565	31.9517	16.8109	0.8970
25	2070	32.1174	16.0553	0.9963
26	1519	31.9974	16.0737	0.9907
27	1078	31.9462	16.0435	0.9929
28	673	32.0921	15.3882	0.9239
29	436	31.9908	15.8256	0.9787
30	215	31.8605	15.0782	0.8945
31	126	32.8333	13.3294	0.7012
32	61	31.3279	11.8597	0.5964
33	33	31.3333	15.4949	0.9579
34	23	31.7391	8.5406	0.3652
35	6	31.0000	5.6667	0.2167
36	3	25.3333	6.2222	0.2578
			$\overline{\theta}$	0.9546

表 6-3.　名暗号の比較

暗号形式		$\bar{\theta}$	処理時間　（秒）
バーナム		0.000	4
DES		0.945	5,121
FEAL－8		0.931	25
改良暗号法	串刺し型		
	10段	0.955	28
	20段	0.900	44
	30段	0.948	60
	短刺し型		
	10段	0.935	28
	20段	0.948	43
	30段	0.957	59

使用パソコン：PC－9801DA

また，理想二項分布近似率の加重平均 $\bar{\theta}$，および所用時間を表 6-2，表 6-3 に示す。

表 6-3 から，串刺し型のほうより短刺し型のほうが θ の値が比較的安定している。

あみだくじの場合串刺し型のほうが，どの始点から出発しても，早く各終点に到達する確率が一定になる。しかし，1 ブロックごとに鍵を生成するための乱数シードを変えているため，生成される乱数の質によって短刺し型のほうがよく見えることが起こりうる。

このデータでの処理時間は，従来のバーナム暗号の 7～15 倍程度である。

表 6-4 は，改良暗号法のブロック長を変えた場合の結果を示したものである。これからブロック化するビット数を少なくするほど暗号強度は増し，反面処理時間が増えることがわかる。

表 6-4.　改良暗号法のブロック長を変化させた場合

ブロック長（ビット数）		64	32	16	8
串刺し型（10段）	$\bar{\theta}$	0.955	0.968	0.981	0.984
	処理時間（秒）	28	49	88	175
短刺し型（10段）	$\bar{\theta}$	0.935	0.965	0.982	0.989
	処理時間（秒）	28	47	86	162

6.3　各暗号方式の評価

表 6-3 から暗号強度 θ は，DES 暗号，FEAL-8 暗号，改良暗号法で数値に若干の違いがみられる。これは試行によるばらつきと考えられる。しかしいずれも，だいたい近い値となっている。また表から改良暗号法の串刺し型と短刺し型の暗号化処理速度はほぼ同じといえる。DES 暗号は，ソフトウェア処理では暗号化処理に時間がかかり過ぎることを示している。改良暗号法は，ソフトウェア処理での処理速度上問題が少ないといえる。

これらの結果より，改良暗号法は，従来のバーナム暗号と比べて，実用性が高いと考えられる。

改良暗号法は，串刺し型，短刺し型ともに暗号の強度および処理速度においてそれほど変わらない。しかし実用性を考えると，段数が 10 段を越えると処理時間が普通のバーナム暗号の 7 倍を越えるので，10 段程度が良いと思われる。

特に，串刺し型は原理的に優れており，10 段でも実用上十分な強度に達しているので，短刺し型よりは，串刺し型を利用したほうがよいと考えられる。

7. むすび

本研究では，従来のバーナム暗号に，あみだ構造を取り入れた新しい方式を提案した。この構造の導入によって，本暗号法では，非周期的な長い乱数列が自動生成することにより実用性が高いものになっている。

処理時間は，従来のバーナム暗号より若干多くかかるが，コンピュータでのソフトウェア処理上問題のない処理速度と考える。

本改良暗号法は，暗号化のブロック長を任意のビット単位ごとに設定できる。このことから新しい適用が可能となり，情報セキュリティ対策上有効と考える。

文　　献

1) D.E.R. デニング著，上園忠弘，小嶋　格，奥島晶子訳："暗号とデータセキュリティ"，p. 146，培風館（1988）.

2) Michael Luby："Pseudo-random Generators from One-way Functions"，Advances in Cryptology-Crypto '91，Lecture Notes in Computer Science No. 576，p. 300 Springer-Verlag（1991）.

3) 辻井重男，笠原正雄："暗号と情報セキュリティ"，p. 163，昭晃堂（1990）.

4) 池野信一，小山謙二："現代暗号理論"，p. 72，電子情報通信学会（1989）.

5) 宮口庄司，平野光徳："暗号"認証アルゴリズム強度指標"，電子情報通信学会誌A分冊，J-69-A，10，pp. 1252-1259（1986）.

6) 宝木和夫，中村　勤："暗号方式と応用"，情報処理学会誌，**32**，6，pp. 714-720（1991）.

7) 清水明宏，宮口庄司："高速データ暗号アルゴリズムFEAL"，電子情報通信学会論文誌D分冊，**J70-D**，7，pp. 1413-1423（1987）.

8) 森口繁一："あみだくじと酔歩の問題"，数学セミナーVol. 23，No. 9，pp. 16-21，日本評論社（1984）.

付録．あみだくじの原理[8)]

あみだくじの線をたどるときの規則は，

(1) 進行方向は下向きまたは水平

(2) 三叉路（T字路）では必ず曲がる

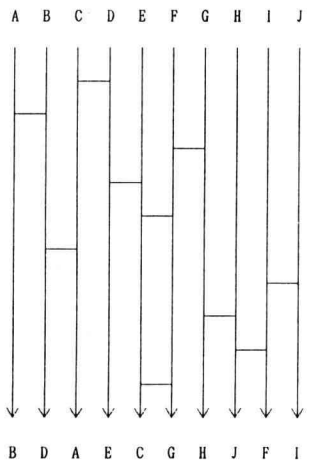
というものである。なお，十字路がどこにもないように横線を入れるのが一般の約束である。

あみだくじでは，上から下へたどっても下から上へたどっても結果は変わらない。逆にたどるときの規則

表　推移確率行列 P^n

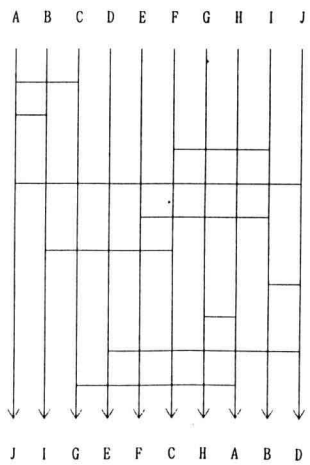
n	対角要素	非対角要素
1	0.8	0.0222
2	0.6444	0.0395
3	0.5235	0.0529
4	0.4294	0.0634
5	0.3562	0.0715
10	0.1729	0.0919
20	0.1059	0.0993
30	0.1005	0.0999
39	0.1000	0.1000

串刺し型　N=10 段の場合



（1）　短刺し型

(1) Short Edge Type of AMIDA Lottery



（2）　串刺し型

(2) Spitting Type of AMIDA Lottery

図

は，(1)を「進行方向は上向きまたは水平」という規則に変える。

いま，一つのモデルとして，10本の線の間に横線を入れることとし，その各段でどこに横線を入れるかは乱数シードにもとづいてランダムに決める。

このモデルについての理論計算を行うと，まず任意の1段で，たとえば線1から1段通過後，隣の線2に移る推移確率は，区間(1, 2)に横線のある確率1/9に等しく，そのほかの場合には通過後も線1の上にとどまっているはずで，その確率は8/9である。

また線2の上から1段通過後，隣の線1，または線3に移っている確率はそれぞれ1/9であり，線2の上にとどまる確率は7/9である。

以下同様にしてn段の確率行列P^nがもとまる。どの始点から出発しても，各終点に到達する確率が等しくするためには，この場合757段が必要となる。

このあみだくじをここでは短刺し型と呼ぶこととする。

次にこのあみだくじに，「十字路」を用いた次の規則を追加する。

(3)　十字路では直進する

このあみだくじを以下串刺し型と呼ぶこととする。上のようなモデルについての理論的計算は次のようになる。

任意の1段で，特定の互換が生じる確率は，

$1/{}_{10}C_2=1/45=0.02222$ であるから

同じ線上に留まる確率は0.8となる。

この場合のn段の推移確率行列P^nを前頁表に示す。各終点に到達する確率を等しくするためには，39段が必要となる。表から明らかなように，10段くらいでも，かなり収束値に近づいていることがわかる。