(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2007-73012
(P2007-73012A)
(43)公開日　平成19年3月22日(2007.3.22)

(51)Int.Cl.			F I			テーマコード（参考）
G06F	7/58	(2006.01)	G06F	7/58	B	5J104
G09C	1/00	(2006.01)	G09C	1/00	650B	

審査請求 未請求　請求項の数 6　ＯＬ　（全 9 頁）

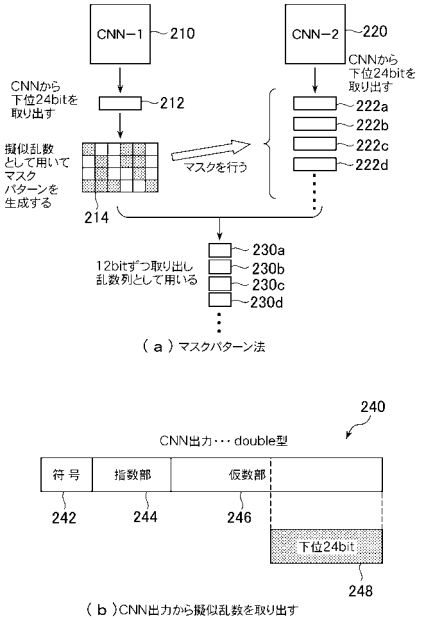
(21)出願番号　特願2005-262581 (P2005-262581)
(22)出願日　平成17年9月9日 (2005.9.9)

(71)出願人　504165591
国立大学法人岩手大学
岩手県盛岡市上田三丁目１８番８号
(74)代理人　100105371
弁理士　加古　進
(72)発明者　吉田　等明
岩手県盛岡市上田三丁目１８番８号　国立大学法人岩手大学内
(72)発明者　中西　貴裕
岩手県盛岡市上田三丁目１８番８号　国立大学法人岩手大学内

Ｆターム(参考)　5J104 FA01

(54)【発明の名称】乱数生成システム

(57)【要約】
【課題】擬似乱数発生手段を用いて暗号学的に安全な乱数を生成する乱数生成システムの提供。
【解決手段】本発明の手法は、ある擬似乱数発生手段（ＣＮＮ－２　２２０）から生成した擬似乱数（２２２ａ，２２２ｂ，２２２ｃ，２２２ｄ，…）を、異なった系の擬似乱数発生手段（ＣＮＮ－１　２１０）から生成した擬似乱数２１２を基に作り出したマスクパターン２１４によりマスクして、その結果（２３０ａ，２３０ｂ，２３０ｃ，２３０ｄ，…）を乱数列として出力する。これらの擬似乱数発生手段には、例えば、カオス・ニューラルネットワーク（ＣＮＮ）を用いることができる。出力された乱数列を従来のストリーム暗号方式等による暗号化システムに用いることにより、より堅牢性の高い暗号化を行なうことができる。
【選択図】図２



【特許請求の範囲】
【請求項１】
乱数を生成する乱数生成システムであって、
　擬似乱数を出力する２個の擬似乱数発生手段と、
　一方の前記擬似乱数発生手段から出力した擬似乱数を用いてマスクパターンを生成するマスクパターン生成手段と、
　他方の前記擬似乱数発生手段から出力した擬似乱数に前記マスクパターンをマスクした結果を出力する乱数出力手段と
　を備えることを特徴とする乱数生成システム。
【請求項２】
請求項１に記載の乱数生成システムにおいて、
　前記マスクパターンは、０と１とが同数であること
　を特徴とする乱数生成システム。
【請求項３】
請求項１または２に記載の乱数生成システムにおいて、
　前記擬似乱数発生手段はカオス・ニューラルネットワークであり、前記擬似乱数は前記カオス・ニューラルネットワークからの出力の下位ビットであること
　を特徴とする乱数生成システム。
【請求項４】
請求項１～３のいずれかに記載の乱数生成システムと、
　入力した平文と前記乱数出力手段からの乱数とを演算して暗号化文を出力する暗号化手段と
　を備えることを特徴とする暗号化システム。
【請求項５】
請求項１～３のいずれかに記載の乱数生成システムの機能をコンピュータ・システムに構築させるプログラム。
【請求項６】
請求項４に記載の暗号化システムの機能をコンピュータ・システムに構築させるプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
　本発明は、暗号学的に安全な擬似乱数を発生する技術に関するものである。
【背景技術】
【０００２】
　デジタルデータの暗号化および復号を行なうシステムにおいては、従来、ストリーム暗号方式という手法が広く用いられている。
　ストリーム暗号方式とは、平文を１ビット、または数ビットごとに暗号化・復号を行なう方法であり、暗号化鍵から鍵系列とよばれる擬似乱数を生成し、その擬似乱数により平文を逐次暗号化する暗号方式である。擬似乱数は、ソフトウェアでも比較的高速に生成することができるため、ストリーム暗号方式による暗号化及び復号システムは、ソフトウェアで実現する場合でも、高速な処理を行なうことができる。この方式では、擬似乱数の生成方法が非常に重要であり、暗号強度に大きく影響する。
【０００３】
　発明者らは、ホップフィールドネットワークや誤差逆伝搬学習則等で用いられている通常の人工ニューロンからなり、カオス出力が得られるニューラルネットワークを発明し、この独自のニューラルネットワークのことをカオス・ニューラルネットワーク（以下ＣＮＮ）と呼んでいる（例えば非特許文献１，特許文献１，特許文献２）。そしてＣＮＮを用いることにより、従来技術よりも乱数性に優れた擬似乱数を生成できることを明らかにしている（特許文献２）。

発明者らのＣＮＮのニューロンモデルを図１（ａ）および式（１），式（２）に示す。

【０００４】

【数１】

$$x_j(t) = f_s(u_j) = \frac{1}{1+\exp(-u_j(t))} \qquad \cdots (1)$$

【０００５】

【数２】

$$u_j(t) = \sum_{i=1}^{n} w_{ij} x_i(t-1) - \theta_j + I_j \qquad \cdots (2)$$

【０００６】

ここで、式（１）で表される非線形出力関数ｆ$_s$は特異な関数ではなく、一般的に用いられているｓｉｇｍｏｉｄ関数である。図1（ａ）および式（１），式（２）におけるその他の各係数は、以下のように定義する。

ｘ$_j$（ｔ）：時刻ｔにおけるニューロンｊの出力（ｔ＝０，１，２，…）

ｕ$_j$：ニューロンｊの内部状態

ｘ$_i$（ｔ－1）：時刻ｔ－1におけるニューロンｉからの入力

ｗ$_{ij}$：ニューロンｉからニューロンｊへの結合荷重

θ$_j$：ニューロンｊの閾値

Ｉ$_j$：ニューロンｊへの外部入力値

【０００７】

自然界では単一で振動する出力を発生するニューロンもあるが、ＣＮＮに用いた上述のニューロンの場合には、１個のニューロンのみでは振動しない。ニューロン・ネットワークの例として、ニューロン４個から構成されるカオス・ニューラルネットワーク（ＣＮＮ）の構成を図１（ｂ）に示す。図１（ｂ）は、ニューロン１（Ｎ$_1$）１２１，ニューロン２（Ｎ$_2$）１２２，ニューロン３（Ｎ$_3$）１２３，ニューロン４（Ｎ$_4$）１２４の４個のニューロンから構成されている。構造の特徴としてニューロンのサイクルが１つあるので、ｃｙｃｌｉｃ－４ｎｎ（以下Ｃ－４ｎｎ）と呼んでいる。Ｃ－４ｎｎのパラメータ（結合荷重𝕎、閾値θおよび外部入力Ｉ$_j$）を表１に示す。

【０００８】

【表１】

W_{12}	-40
W_{13}	20
W_{24}	16.8
W_{31}	-20
W_{43}	-8.6
θ_1	0
θ_2	0
θ_3	0
θ_4	0
I_1	-0.497
I_3	-0.497

【０００９】

さらに、発明者らは、ＣＮＮより生成された出力の下位ｂｉｔがよい乱数性を持っていることを明らかにし、従来の乱数発生方法よりも一様性に優れ、容易に真性乱数と区別できない一様な乱数を生成することができる乱数生成システムを発明している（特許文献２）。

【００１０】
【非特許文献１】S.Kawamura, H.Yoshida, M.Miura, and M.Abe: Implementation of Uniform Pseudo Random Number Generator and Application to Stream Cipher based on Chaos Neural Network, Proceedings of Papers, the International Conference on Fundamentals of Electronics, Communications and Computer Sciences, R-18, 2002.
【特許文献１】特開２００１－１４４７４６号公報
【特許文献２】特開２００３－７６２７２号公報
【発明の開示】
【発明が解決しようとする課題】
【００１１】
ストリーム暗号方式への攻撃を防御する観点から、より堅牢性の高い暗号化を実現できる擬似乱数の生成方法が望まれている。本発明の課題は、擬似乱数の生成において、ストリーム暗号方式などに利用可能な暗号系の暗号化鍵空間を広げる方法（以降、マスクパターン法と呼ぶ）を提案し、さらに堅牢性の高い暗号化を可能にすることである。
【課題を解決するための手段】
【００１２】
上記の課題を解決するために、本発明は、乱数を生成する乱数生成システムであって、擬似乱数を出力する２個の擬似乱数発生手段と、一方の前記擬似乱数発生手段から出力した擬似乱数を用いてマスクパターンを生成するマスクパターン生成手段と、他方の前記擬似乱数発生手段から出力した擬似乱数に前記マスクパターンをマスクした結果を出力する乱数出力手段とを備えることを特徴とする乱数生成システムである。
この乱数生成システムにおいて、前記マスクパターンは、０と１とが同数であることを特徴としてもよい。
また、前記擬似乱数発生手段はカオス・ニューラルネットワークであり、前記擬似乱数は前記カオス・ニューラルネットワークからの出力の下位ビットであることを特徴としてもよい。
さらに、上記に記載の乱数生成システムと、入力した平文と前記乱数出力手段からの乱数とを演算して暗号化文を出力する暗号化手段とを備えることを特徴とする暗号化システムも、本発明である。
また、上記の乱数生成システムの機能をコンピュータ・システムに構築させるプログラム、および上記の暗号化システムの機能をコンピュータ・システムに構築させるプログラムも、本発明である。
【発明の効果】
【００１３】
本発明で提案するマスクパターン法を用いた乱数生成システムでは、ある擬似乱数発生手段から生成された擬似乱数を、異なった系の擬似乱数発生手段からの擬似乱数を基に作り出したマスクパターンでマスクして、その結果の乱数列を暗号化の擬似乱数として用いる。そのため、発生可能な暗号化鍵の数を必要に応じて増やすことが可能となり、暗号化鍵空間を広げることができた。これにより、従来よりも暗号学的に安全な擬似乱数を生成することができ、暗号化システムを堅牢化できる。
特に、発明者らの技術であるＣＮＮより生成された出力の下位ｂｉｔを擬似乱数として本実施形態のマスクパターン法を用いることにより、ＣＮＮの出力の下位ｂｉｔの優れた乱数性を落とすことなく暗号化鍵空間を広げることができる。具体的には、発明者らの既存技術（非特許文献１，特許文献１，特許文献２を参照）に対して、０と１とが同数のマスクパターンを用いて上述のマスクパターン法を適用すると、暗号鍵空間を標準で10^{48}倍に改良することができた。これにより解読に要する時間を10^{29}年以上に延ばすことができる。もともと、発明者らの既存技術であるＣＮＮによる暗号化方法は、米国の次世代標準暗号であるＲｉｊｎｄａｅｌより高速かつ堅牢であるが、本発明の手法によれば、これをさらに堅牢化できる。
【発明を実施するための最良の形態】

【００１４】
以降、本発明の乱数生成システムの実施形態の一例を詳細に説明する。まず、ストリーム暗号方式の概要と考えられる主な攻撃方法について説明し、次に、これらの攻撃に対処するための方法として、本実施形態の手法であるマスクパターン法を提案する。
＜１．暗号化＞
本実施形態では、ＣＮＮより生成した擬似乱数から、本実施形態で提案する新しい手法であるマスクパターン法により暗号化に用いる乱数を生成し、上述のストリーム暗号方式による暗号化を行うシステムを実装する場合を例として説明するが、他の擬似乱数発生手段を用いてもよい。
以降、この暗号化システムの実装に向け、ストリーム暗号方式の概要と、考えられる主な攻撃方法について述べる。
【００１５】
（１－１．ストリーム暗号方式）
用いる暗号として、秘密鍵暗号方式（共通鍵暗号方式）の一種である、上述のストリーム暗号方式を使用する。その特徴を以下に示す。
・１ｂｉｔ毎に暗号化および復号を行う。
・一般的に堅牢性よりも速度を重視する。
【００１６】
（１－２．暗号化の概要）
本実施形態で用いるストリーム暗号方式の暗号化と復号について概要を示す。
（１）暗号化
平文　ＸＯＲ　乱数列＝暗文　…（３．１）
（２）復号
暗文　ＸＯＲ　乱数列＝平文　…（３．２）
ここで、平文とは暗号化されていないデータであり、暗文は、暗号化され、そのままでは読めない状態になっているデータである。
【００１７】
（１－３．主な攻撃方法）
一方、ストリーム暗号を攻撃する主な方法として、（１）既知平文攻撃、（２）選択平文攻撃、（３）差分攻撃、等の方法が挙げられる。
（１）既知平文攻撃
攻撃者が、平文と暗文の両方を手に入れることができる立場にある場合の攻撃方法。
平文　ＸＯＲ　暗文＝乱数列　…（３．３）
式（３．３）により、乱数列を取得されてしまう。これを繰り返し行うことによって、次の乱数列を予測される恐れがある。
【００１８】
（２）選択平文攻撃
攻撃者が平文の内容を自由に選択できる立場にある場合の攻撃方法。式（３．１）において、平文の内容を全て０にすることにより、
０　ＸＯＲ　乱数列＝暗文　…（３．４）
乱数列＝暗文　…（３．５）
となり、乱数列の内容が暗文にそのまま表れ、乱数列を取得されてしまう。
【００１９】
（３）差分攻撃
平文と暗文のセットを複数持つことができる立場にある場合の攻撃方法。
平文１　ＸＯＲ　乱数列１＝暗文１　…（３．６）
平文２　ＸＯＲ　乱数列２＝暗文２　…（３．７）
（平文１　ＸＯＲ　平文２）ＸＯＲ（乱数列１　ＸＯＲ　乱数列２）＝（暗文１　ＸＯＲ　暗文２）　…（３・８）
このとき、乱数列１と乱数列２が等しくなる可能性があると、

乱数列１　ＸＯＲ　乱数列２＝０　…（３．９）
となり、式（３．８）と式（３．９）から、
平文１　ＸＯＲ　平文２＝暗文１　ＸＯＲ　暗文２　…（３．１０）
となる。式（３．１０）より、同じ乱数列を使用すると、乱数列の内容に関わらず、解除されてしまう。

【００２０】

＜４．マスクパターン法＞

（４－１．目的）

上述したように、カオス出力の下位ｂｉｔは乱数性に優れている（特許文献２参照）が、本実施形態では、暗号化システムにおいて上述の攻撃への対処をより強化するために、以降説明する「マスクパターン法」を提案する。なお、本実施形態では、擬似乱数を発生する擬似乱数発生手段として、上述のＣＮＮを用いる場合を例として説明するが、他の擬似乱数発生手段を用いてもよい。

【００２１】

（４－２．概要）

本実施形態のマスクパターン法とは、ある擬似乱数発生手段（ここではＣＮＮを例として説明する）からの擬似乱数を、異なった系の擬似乱数発生手段（ここではＣＮＮを例として説明する）からの擬似乱数を基に作り出したマスクパターンでマスクすることにより、暗号学的に安全な擬似乱数を生成する方法である。本実施形態のマスクパターン法の処理の流れを図２（ａ）に示す。

本実施形態の乱数生成システムにおいて、乱数生成に用いるための擬似乱数の生成は、擬似乱数発生手段により行なう。なお、上述したように、本実施形態では擬似乱数発生手段の一例として発明者らのカオス・ニューラルネットワーク（ＣＮＮ）を用いる例で説明するが、他の擬似乱数発生手段を用いてもよい。マスクパターンの生成は、マスクパターン生成手段により行なう。擬似乱数発生手段からの擬似乱数をマスクパターンでマスクして暗号化に用いる乱数を生成・出力する処理は、乱数出力手段により行なう。

以降、図２（ａ）および図２（ｂ）を参照しながら、本実施形態で提案するマスクパターン法について詳しく説明する。

【００２２】

（４－３．擬似乱数の生成）

図２（ａ）に示すように、マスクパターン法は、あるＣＮＮ（ＣＮＮ－２）２２０から生成された擬似乱数（２２２ａ，２２２ｂ，２２２ｃ，２２２ｄ，…）を、異なった系のＣＮＮ（ＣＮＮ－１　２１０）から生成された擬似乱数２１２を基に作り出したマスクパターン２１４によりマスクして、より堅牢性を高める方法である。

これらの擬似乱数の生成は、擬似乱数発生手段により行なう。ここでは、擬似乱数発生手段として発明者らのＣＮＮを用いた例で説明するが、他の擬似乱数発生手段を用いてもよい。

ＣＮＮから擬似乱数を出力する例を図２（ｂ）に示す。図２（ｂ）に示すように、例えばｄｏｕｂｌｅ型変数を格納できるレジスタ２４０を用意し、図１（ｂ）に示すようなＣＮＮを構成するニューロンから、時系列で発生する出力を取り出して、レジスタ２４０に格納する。次に、レジスタ２４０から、仮数部の下位のｘビットを取り出す。取り出したビット列が乱数となっている。この乱数は、ＣＮＮを構成しているニューロンであれば、どのニューロン（図１（ｂ）ではN_1　１２１～N_4　１２４）から取り出してもよい。また、数表現がＩＥＥＥ７５４形式以外の場合でも、浮動小数点または固定小数点表現を用いている場合は、本手法を適応できる。（詳しくは特許文献２を参照）。なお、本実施形態では、レジスタ２４０から仮数部の下位２４ｂｉｔを取り出すものとしているが、これは、下位２４ビットのランダム性がよいからである。

【００２３】

（４－４．マスクパターンの生成）

次に、本実施形態で用いるマスクパターン２１４の生成方法を、一例を挙げて説明する

。

マスクパターン２１４は、本実施形態の乱数生成システムのマスクパターン生成手段により生成する。なお、ここで説明するマスクパターン生成方法は一例であり、他の方法により生成してもよい。

本実施形態で用いるマスクパターン２１４は２４ｂｉｔの長さとする。ここでは、例として、０と１が同数（０が１２ｂｉｔ、１が１２ｂｉｔ）でマスクパターンを構成する場合を示す。本実施形態のマスクパターン法では、０と１とが必ずしも同数でなくてもよいが、０と１とが同数の場合にマスクパターンの組合せの数が最大となり、堅牢性も最大になるからである。このマスクパターンを８個生成し、１９２ｂｉｔを１セットとして用いる。

この１セットのマスクパターンでＣＮＮ－２　２２０からの擬似乱数（２２２ａ，２２２ｂ，２２２ｃ，２２２ｄ，…）のマスクを行なった結果の乱数列（２３０ａ，２３０ｂ，２３０ｃ，２３０ｄ，…）を、暗号化の際の乱数列として用いる。

ここで、２４ｂｉｔのマスクパターンの可能な組み合わせの数は、${}_{24}C_{12}$　10^{6}となる。１セットのマスクパターン（１９２ｂｉｔ）を考えると、可能なマスクパターンの数は${}_{106}P_{8}$　10^{48}となる。

【００２４】

上述したように、本実施形態では、０と１の個数が同数になるように２４ｂｉｔのマスクパターンを作り、それを８個生成して１セット（１９２ｂｉｔ）のマスクパターンとする。ここで、０と１の個数を同数にするため、例えば、２４ｂｉｔ内で順にペアとなるビットを決め、片方に０、もう片方に１を入れていく。以降、マスクパターン生成の例として、ＣＮＮ－１　２１０から生成した擬似乱数２１２を用いて、ペアとなるビットの決定、およびペアとなったビットのどちらが０でどちらが１になるかを決定する方法を説明する。なお、マスクパターンの生成方法はこれに限られず、他の方法を用いてもよい。また、上述したように０と１とが同数でなくてもよい。

【００２５】

（１）ペアとなるビットの決定

本実施形態の例では、ＣＮＮ－１　２１０から２４ｂｉｔの擬似乱数２１２を生成し、この擬似乱数の値を用いて、ペアを順次１組ずつ決めていく。２４ｂｉｔのマスクパターンを生成するには、１２組のペアを決定することになる。

２４ｂｉｔのマスクパターン生成において、ＣＮＮ－１　２１０からの擬似乱数x_n（２４ｂｉｔの場合ｎ＝１，２，…，１２）を用いて、ｎ組目のペアの相手となるビットの番号y_n（２４ｂｉｔの場合ｎ＝１，２，…，１２）を決定する処理は、次の式（４）で表される。

【００２６】

【数３】

$$y_n = \left[x_n \div \frac{2^{24}}{23\text{-}2(n\text{-}1)} \right] + 1 \qquad \cdots(4)$$

なお、式（４）において［ａ］はガウス記号であり、実数ａを超えない最大の整数であることを示している。

【００２７】

以降、式（４）を参照しながら、ペアとなるビットを決定する処理の一例を説明する。

まず、２４ｂｉｔのマスクパターンの１ビット目とペアになるビットを決定する（１組目のペアの決定）。ＣＮＮ－１　２１０からの２４ｂｉｔの擬似乱数x_1を用いて、１ビット目とペアになるビットを決める。

具体的には、２４ｂｉｔで表現できる数（すなわち、０から$2^{24}-1$までの2^{24}個の数）を、マスクパターン内のまだ決定していないビットの数（ここでは１ビット目を除いた数である２３）で割って、そのビット数（ここでは２３）のグループに分割し、数が小さい方から順にグループ＿１，グループ＿２，…，とする。一方、まだ決定していない

ビット（ここでは1ビット目を除いた２３ｂｉｔ）を順に、ビット＿１，ビット＿２，…，とする。

次に、２４ｂｉｔの擬似乱数x_1の値が含まれているグループ＿y_1を見つける。そして、y_1番目のビットであるビット＿y_1を、１ビット目とペアになるビットに決定する。

【００２８】

同様に、２組目以降も、マスクパターンからまだ決定していないビット（ｎ組目では２４－２（ｎ－１）ビット残っている）のうち最初のビットを選び、これとペアになるビットを決定する。ＣＮＮ－１　２１０から次の擬似乱数x_nを生成して、上述の１組目と同様に、２４ｂｉｔで表現できる数を、まだ決定していないビット数（ｎ組目では２３－２（ｎ－１））で割ってグループに分割し、擬似乱数x_nが何番目のグループに含まれるかによってペアとなるビットを決定する。

上記の処理を、全てのペアが決定するまで繰り返す。なお、最後の（１２組目の）ペアは残りの2ｂｉｔであるので、ペアの決定処理を行なう必要はなく、次に説明するビットの値の決定（0か1かの決定）のみ行なえばよい。

【００２９】

（２）ペアとなったビットの値の決定

ｎ組目（２４ｂｉｔの場合ｎ＝１，２，…，１２）のペアとなるビットが決定したら、擬似乱数x_nの最下位ビットにより、ｎ組目のビットのどちらが0でどちらが1になるかを決定する。本実施形態では、例えば、擬似乱数x_nの最下位ビットが0の場合には、先の（まだ決定していないビットのうち最初の）ビットを0とし、ペアに決定したビットを1とする。逆に擬似乱数x_nの最下位ビットが1の場合には、先のビットを1とし、ペアに決定したビットを0とする。のように決めておく。

以上の（１）（２）の処理により、１個のマスクパターンが生成される。

さらに、ここまでの処理を計8回繰り返し、1セット１９２ｂｉｔのマスクパターンとする。なお、マスクパターンのビット数は２４ｂｉｔ以外でもよく、1セットのマスクパターン数は8個以外でもよい。

【００３０】

（４－５．暗号化に用いる乱数列の生成）

次に、上述で生成したマスクパターンセット２１４を用いて、暗号化に用いる擬似乱数を生成する方法について説明する。この乱数は、本実施形態の乱数生成システムの乱数出力手段により生成し、出力する。

マスクパターン２１４生成時に使ったＣＮＮ－１　２１０とは違う系のＣＮＮ（ＣＮＮ－２　２２０）を使い、２４ｂｉｔの擬似乱数を8個（２２２ａ，２２２ｂ，２２２ｃ，２２２ｄ，…）を取り出す。これを上述で生成した1セット（8個）のマスクパターン２１４によりマスクする。本実施形態では、例えばマスクが0のビットは破棄し、マスクが1のビットだけを取り出すものとする。この結果、出力（２３０ａ，２３０ｂ，２３０ｃ，２３０ｄ，…）は1セット９６ｂｉｔ（＝１２ｂｉｔ×８）となる。これを暗号化の際の乱数列として用いる。

ここで、組み合わせるマスクパターンの数は，上述したように、必ずしも8個でなくとも良いので，２４ビットｉ個使ったとすれば、２４ｉビットのマスクパターンとなり，可能な組み合わせの数は，１０の６ｉ乗個になる。即ち、発生可能な乱数列の数は，１０の６ｉ乗個になる。

このように、例えば、ｉ＝１００なら１０の６００乗個，ｉ＝１０００なら１０の６０００乗個といった具合に、いくらでも増すことができる。

【００３１】

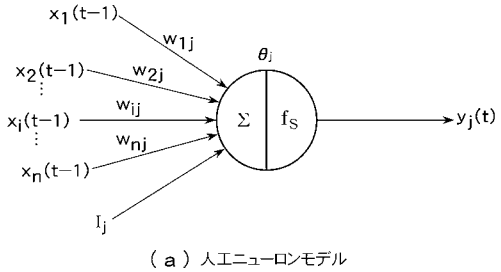
（４－６．マスクパターン法を用いた暗号化システム）

上述の（１－２．暗号化の概要）で説明したように、ストリーム暗号方式による暗号化および復号の手法は、次のとおりである。

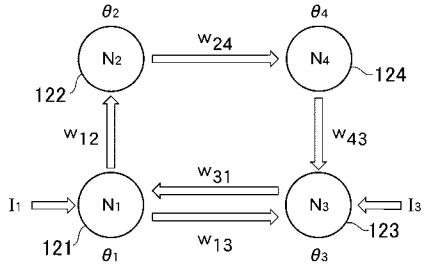
（１）暗号化

平文　ＸＯＲ　乱数列＝暗文　…（３．１）
（２）復号
暗文　ＸＯＲ　乱数列＝平文　…（３．２）
　ここで、本実施形態のマスクパターンを適用して得られた乱数列を、演算（３．１）および（３．２）の乱数列として用いることにより、入力した平文の暗号化および復号を行なうことができる。
【００３２】
（４－７．マスクパターン法の効果）
　本実施形態のマスクパターン法を用いることにより、発生可能な暗号化鍵の数を必要なだけ増やすことができ（暗号化鍵空間の拡張）、安全なストリーム暗号方式を実現することができる。例えば発明者らの既存技術であるＣＮＮ（非特許文献１，特許文献１，特許文献２を参照）に対して、上述のマスクパターン法を適用すると、暗号鍵空間を標準で10^{48}倍に改良することができた。これにより解読に要する時間を10^{29}年以上に延ばすことができる。もともと、発明者らの既存技術による暗号化方法は、米国の次世代標準暗号であるＲｉｊｎｄａｅｌより高速かつ堅牢であるが、本実施形態の手法によれば、これをさらに堅牢化できる。また、現在の米国標準であるＤＥＳなどの暗号系は、数十時間で解読可能な暗号であるのに対して、本技術を用いると理論的には解読できない暗号系を構成できる点が優位である。
【図面の簡単な説明】
【００３３】
【図１】（ａ）ＣＮＮのニューロンモデルである。（ｂ）ニューロン４個から構成されるカオス・ニューラルネットワーク（Ｃ－４ｎｎ）のモデル図である。
【図２】（ａ）本実施形態のマスクパターン法の処理の流れを示す図である。（ｂ）ＣＮＮ出力から擬似乱数を取り出す処理を示す図である。

【図１】

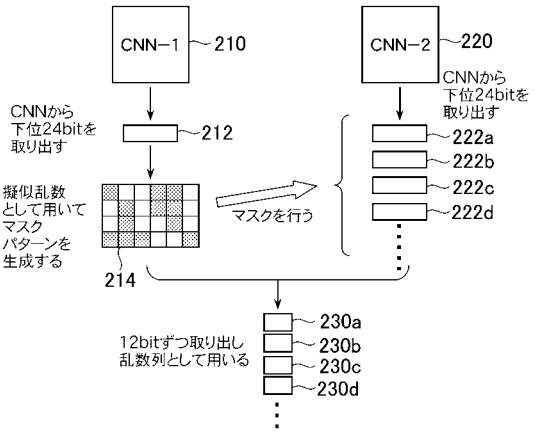


（ａ）人工ニューロンモデル

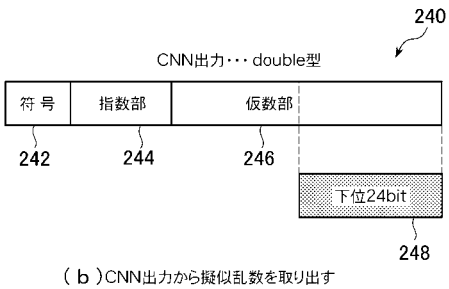


（ｂ）C−4nn カオス・ニューラルネット

【図２】



（ａ）マスクパターン法



（ｂ）CNN出力から擬似乱数を取り出す