(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5772934号
(P5772934)
(45)発行日　平成27年9月2日(2015.9.2)　(24)登録日　平成27年7月10日(2015.7.10)

(51)Int.Cl.			F I		
H04L	9/06	(2006.01)	H04L	9/00	611Z
G09C	1/00	(2006.01)	G09C	1/00	610A

請求項の数 8　(全 32 頁)

(21)出願番号　特願2013-249046(P2013-249046)
(22)出願日　平成25年12月2日(2013.12.2)
(62)分割の表示　特願2012-102624(P2012-102624)の分割
原出願日　平成18年9月1日(2006.9.1)
(65)公開番号　特開2014-41389(P2014-41389A)
(43)公開日　平成26年3月6日(2014.3.6)
審査請求日　平成25年12月2日(2013.12.2)

(73)特許権者　000002185
ソニー株式会社
東京都港区港南１丁目７番１号
(74)代理人　100093241
弁理士　宮田　正昭
(74)代理人　100101801
弁理士　山田　英治
(74)代理人　100086531
弁理士　澤田　俊夫
(74)代理人　100095496
弁理士　佐々木　榮二
(74)代理人　110000763
特許業務法人大同特許事務所

最終頁に続く

(54)【発明の名称】データ変換装置、およびデータ変換方法、並びにコンピュータ・プログラム

(57)【特許請求の範囲】
【請求項１】
データ変換装置であり、
前記データ変換装置は、Ｆ関数を複数ラウンド繰り返し実行する構成を有し、
前記Ｆ関数の各ラウンドのラウンド鍵のビット数と同じビット数を有するように変換対象データを分割する第１データ分割部と、
前記第１データ分割部で分割された分割データに前記ラウンド鍵の鍵加算を実行する鍵加算部と、
前記鍵加算部で鍵加算された前記分割データを入力して非線形変換処理を実行するＦ関数内非線形変換部と、
前記Ｆ関数内非線形変換部の出力に対する線形変換処理を実行するＦ関数内線形変換部を有し、
前記Ｆ関数内非線形変換部は、
前記鍵加算部で鍵加算された前記分割データをさらに分割する第２データ分割部と、
前記第２データ分割部で分割された分割データ各々の非線形処理を実行する複数の小型非線形変換部からなる第１段サブ非線形変換部と、
前記第１段サブ非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行するサブ線形変換部と、
前記サブ線形変換部の出力データを分割する第３データ分割部と、
前記第３データ分割部で分割した分割データ各々の非線形変換処理を前記サブ線形変換

部による線形変換の直後に実行する複数の小型非線形変換部からなる第２段サブ非線形変換部とを有し、
前記サブ線形変換部は、
前記第１段サブ非線形変換部の出力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行する構成であることを特徴とするデータ変換装置。
【請求項２】
前記第１段サブ非線形変換部に対する入力データがｎビットデータである場合、
前記第１段サブ非線形変換部は、入力データであるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部から構成され、
前記サブ線形変換部は、前記ｋ個の小型非線形変換部の出力する総計ｎビットのデータを入力して、高分岐数の行列を適用したデータ変換処理によりｎビットの出力を生成する構成であり、
前記第２段サブ非線形変換部は、前記サブ線形変換部から出力されるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部を有する構成であることを特徴とする請求項１に記載のデータ変換装置。
【請求項３】
前記段サブ線形変換部は、最適拡散変換（ＯＤＭ（Ｏｐｔｉｍａｌ　Ｄｉｆｆｕｓｉｏｎ　Ｍａｐｐｉｎｇｓ））処理を実行するＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用したデータ変換処理を実行する構成であることを特徴とする請求項１に記載のデータ変換装置。
【請求項４】
前記サブ線形変換部は、
前記入力データがｎビットデータである場合、
ＧＦ（２）上で定義されるｎ次の既約多項式ｐ（ｘ）によって定義される拡大体ＧＦ（2^n）上の行列を適用したデータ変換処理を実行する構成であることを特徴とする請求項１に記載のデータ変換装置。
【請求項５】
前記データ変換装置は、
非線形変換処理を伴う暗号処理を実行する構成であることを特徴とする請求項１～４いずれかに記載のデータ変換装置。
【請求項６】
前記暗号処理は、共通鍵ブロック暗号処理であることを特徴とする請求項５に記載のデータ変換装置。
【請求項７】
データ変換装置において実行するデータ変換方法であり、
前記データ変換装置は、ラウンド関数を複数ラウンド繰り返し実行する構成を有し、
第１データ分割部において、前記ラウンド関数の各ラウンドのラウンド鍵のビット数と同じビット数を有するように変換対象データを分割する第１データ分割ステップと、
鍵加算部において、前記第１データ分割部で分割されたデータに前記ラウンド鍵の鍵加算を実行する鍵加算ステップと、
Ｆ関数内非線形変換部において、前記鍵加算部で鍵加算された前記分割データを入力して非線形変換処理を実行するＦ関数内非線形変換ステップと、
Ｆ関数内線形変換部において、前記Ｆ関数内非線形変換部の出力に対する線形変換処理を実行するＦ関数内線形変換ステップを実行し、
前記Ｆ関数内非線形変換ステップは、
第２データ分割部において、前記鍵加算部で鍵加算された前記分割データをさらに分割

する第２データ分割ステップと、
第１段サブ非線形変換部において、前記第２データ分割部で分割された分割データ各々の非線形変換処理を実行する第１段サブ非線形変換ステップと、
サブ線形変換部において、前記第１段サブ非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行するサブ線形変換ステップと、
第３データ分割部において、前記サブ線形変換部の出力データを分割する第３データ分割ステップと、
第２段サブ非線形変換部において、複数の小型非線形変換部を適用して、前記第３データ分割部で分割した分割データ各々の非線形変換処理を前記サブ線形変換部の線形変換直後に実行する第２段サブ非線形変換ステップとを有し、
前記サブ線形変換ステップは、
前記第１段サブ非線形変換部の出力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行するステップであることを特徴とするデータ変換方法。
【請求項８】
データ変換装置においてデータ変換処理を実行させるコンピュータ・プログラムであり、
前記データ変換装置は、ラウンド関数を複数ラウンド繰り返し実行する構成を有し、
第１データ分割部において、前記ラウンド関数の各ラウンドのラウンド鍵のビット数と同じビット数を有するように変換対象データを分割させる第１データ分割ステップと、
鍵加算部において、前記第１データ分割部で分割されたデータに前記ラウンド鍵の鍵加算を実行させる鍵加算ステップと、
Ｆ関数内非線形変換部において、前記鍵加算部で鍵加算された前記分割データを入力して非線形変換処理を実行させるＦ関数内非線形変換ステップと、
Ｆ関数内線形変換部において、前記Ｆ関数内非線形変換部の出力に対する線形変換処理を実行させるＦ関数内線形変換ステップを実行し、
前記Ｆ関数内非線形変換ステップにおいては、
第２データ分割部において、前記鍵加算部で鍵加算された前記分割データをさらに分割させる第２データ分割ステップと、
第１段サブ非線形変換部において、前記第２データ分割部で分割された分割データ各々の非線形変換処理を実行させる第１段サブ非線形変換ステップと、
サブ線形変換部において、前記第１段サブ非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行させるサブ線形変換ステップと、
第３データ分割部において、前記サブ線形変換部の出力データを分割させる第３データ分割ステップと、
第２段サブ非線形変換部において、複数の小型非線形変換部を適用して、前記第３データ分割部で分割した分割データ各々の非線形変換処理を前記サブ線形変換部の線形変換直後に実行させる第２段サブ非線形変換ステップとを有し、
前記サブ線形変換ステップは、
前記第１段サブ非線形変換部の出力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行させるステップであることを特徴とするコンピュータ・プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、データ変換装置、およびデータ変換方法、並びにコンピュータ・プログラムに関する。さらに詳細には、例えば、共通鍵ブロック暗号処理に適用可能な非線形変換処理を行なう拡張Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号を適用したデータ変換装置、および

データ変換方法、並びにコンピュータ・プログラムに関する。

【背景技術】

【０００２】

昨今、ネットワーク通信、電子商取引の発展に伴い、通信におけるセキュリティ確保が重要な問題となっている。セキュリティ確保の１つの方法が暗号技術であり、現在、様々な暗号化手法を用いた通信が実際に行なわれている。

【０００３】

例えばＩＣカード等の小型の装置中に暗号処理モジュールを埋め込み、ＩＣカードと、データ読み取り書き込み装置としてのリーダライタとの間でデータ送受信を行ない、認証処理、あるいは送受信データの暗号化、復号を行なうシステムが実用化されている。

【０００４】

暗号処理アルゴリズムには様々なものがあるが、大きく分類すると、暗号化鍵と復号鍵を異なる鍵、例えば公開鍵と秘密鍵として設定する公開鍵暗号方式と、暗号化鍵と復号鍵を共通の鍵として設定する共通鍵暗号方式とに分類される。

【０００５】

共通鍵暗号方式にも様々なアルゴリズムがあるが、その１つに共通鍵をベースとして複数の鍵を生成して、生成した複数の鍵を用いてブロック単位（６４ビット，１２８ビットなど）のデータ変換処理を繰り返し実行する方式がある。このような鍵生成方式とデータ変換処理を適用したアルゴリズムの代表的なものが共通鍵ブロック暗号方式である。

【０００６】

代表的な共通鍵ブロック暗号のアルゴリズムとしては、例えば過去に米国標準暗号であったＤＥＳ（Data Encryption Standard）アルゴリズム、現在の米国標準暗号であるＡＥＳ（Advanced Encryption Standard）アルゴリズムなどが知られている。

【０００７】

このような、共通鍵ブロック暗号のアルゴリズムは、主として、入力データの変換を繰り返し実行するラウンド関数実行部を有する暗号処理部と、ラウンド関数部の各ラウンドで適用するラウンド鍵を生成する鍵スケジューリング部とによって構成される。鍵スケジューリング部は、秘密鍵であるマスター鍵（主鍵）に基づいて、まずビット数を増加させた拡大鍵を生成し、生成した拡大鍵に基づいて、暗号処理部の各ラウンド関数部で適用するラウンド鍵（副鍵）を生成する。

【０００８】

このようなアルゴリズムを実行する具体的な構造として、線形変換部および非線形変換部を有するラウンド関数を繰り返し実行する構造が知られている。例えば代表的な構造にＦｅｉｓｔｅｌ構造がある。Ｆｅｉｓｔｅｌ構造は、データ変換関数としてのラウンド関数（Ｆ関数）の単純な繰り返しにより、平文を暗号文に変換する構造を持つ。ラウンド関数（Ｆ関数）においては、線形変換処理および非線形変換処理が実行される。なお、Ｆｅｉｓｔｅｌ構造を適用した暗号処理について記載した文献としては、例えば非特許文献１、非特許文献２がある。

【０００９】

この共通鍵ブロック暗号処理や、例えばハッシュ関数などでは非線形変換処理によるデータ変換が実行される。非線形変換にはＳボックス（Ｓ－ｂｏｘ）と呼ばれる非線形変換関数が適用可能である。Ｓボックスは、ブロック暗号やハッシュ関数の構成要素であり、その安全性や実装性能を決定付ける非常に重要な関数である。Ｓボックスは一般的にはｎ－ｂｉｔ入力、ｍ－ｂｉｔ出力の非線形変換関数である。入出力ビット数が同数でかつ、入出力が１対１に対応しているＳボックスは、全単射型Ｓボックスと呼ばれる。

【００１０】

暗号処理の非線形変換にＳボックスを適用する場合、適用するＳボックスの性質は、暗号の安全性に大きな影響を与える。すなわち、例えば差分解析、線形解析など、様々な暗号攻撃が知られているが、このような暗号攻撃によって鍵やアルゴリズムが解析される困難性が高いほど安全性が高い。ブロック暗号やハッシュ関数に利用されるＳボックスの性

質によってこの安全性が大きく異なってくる。
【００１１】
例えば、暗号アルゴリズム全体や暗号処理に適用されるラウンド関数自体の厳密な安全性評価は、その入出力サイズが大きい（例えば６４ビット、１２８ビットなど）ことから一般的に困難である。しかし、Ｓボックスの入出力サイズは一般的には小さい。例えば、８－ｂｉｔ入出力程度であり、厳密な安全性評価が可能となる。暗号アルゴリズム全体の安全性を高くするために、Ｓボックスには、少なくとも以下のような特性が求められることが知られている。
（１）最大差分確率が十分小さい
（２）最大線形確率が十分小さい
（３）ブール多項式表現を行った際のブール代数次数が十分大きい
（４）入出力を多項式表現した際の項数が十分多い
【００１２】
主に、（１）は差分攻撃に対する耐性、（２）は線形攻撃に対する耐性、（３）は高階差分攻撃に対する耐性、（４）は補間攻撃に対する耐性をそれぞれ決定付ける。さらには、Ｓボックスの入出力のビット相関が低いことや入力を１ビット変化させたときの出力の変化が１／２程度になることなどが安全性を高めるために重要である。
【００１３】
また、Ｓボックスは安全性の要求とは別に高い実装性能も要求される。例えば、ソフトウェアによる暗号処理やハッシュ関数を実行させる実装構成においては通常、入力に対する出力を示すテーブルをメモリに保持して、表引き（テーブル実装）と呼ばれる手法により、非線形変換処理を実行する構成とした実装がなされるため、その実装性能はＳボックスの内部構造にはあまり依存しない。しかしながら、ハードウェア実装においては、例えば入力値に基づいて、特定の出力を算出するための回路が構成される。この回路構成は、適用するＳボックスにより、構成が大きく変化することになり、回路規模に影響を与えることになる。
【先行技術文献】
【非特許文献】
【００１４】
【非特許文献１】K. Nyberg, "Generalized Feistel networks", ASIACRYPT'96, SpringerVerlag, 1996, pp.91--104.
【非特許文献２】Yuliang Zheng, Tsutomu Matsumoto, Hideki Imai: On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses. CRYPTO 1989: 461-480
【発明の開示】
【発明が解決しようとする課題】
【００１５】
本発明は、上記問題点に鑑みてなされたものであり、暗号処理などにおいて利用される非線形変換処理を実行するＳボックスの構成を改良し、各種の攻撃に対する耐性を高めた構成、すなわち安全性の高めた拡張Ｆｅｉｓｔｅｌ型共通鍵ブロック暗号を適用した暗号処理等を実行するデータ変換装置、およびデータ変換方法、並びにコンピュータ・プログラムを提供することを目的とする。
【課題を解決するための手段】
【００１６】
本発明の第１の側面は、
データ変換装置であり、
入力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部からなる第１段非線形変換部と、
前記第１段非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行する線形変換部と、

前記線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部からなる第２段非線形変換部とを有し、
前記線形変換部は、
前記入力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行する構成であることを特徴とするデータ変換装置にある。

【００１７】

さらに、本発明のデータ変換装置の一実施態様において、前記入力データがｎビットデータである場合、前記第１非線形変換部は、入力データであるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部から構成され、前記線形変換部は、前記ｋ個の小型非線形変換部の出力する総計ｎビットのデータを入力して、高分岐数の行列を適用したデータ変換処理によりｎビットの出力を生成する構成であり、前記第２非線形変換部は、前記線形変換部から出力されるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部を有する構成であることを特徴とする。

【００１８】

さらに、本発明のデータ変換装置の一実施態様において、前記線形変換部は、最適拡散変換（ＯＤＭ（Ｏｐｔｉｍａｌ　Ｄｉｆｆｕｓｉｏｎ　Ｍａｐｐｉｎｇｓ））処理を実行するＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用したデータ変換処理を実行する構成であることを特徴とする。

【００１９】

さらに、本発明のデータ変換装置の一実施態様において、前記線形変換部は、前記入力データがｎビットデータである場合、ＧＦ（２）上で定義されるｎ次の既約多項式ｐ（ｘ）によって定義される拡大体ＧＦ（2^n）上の行列を適用したデータ変換処理を実行する構成であることを特徴とする。

【００２０】

さらに、本発明のデータ変換装置の一実施態様において、前記データ変換装置は、ｎビット入出力の非線形変換処理を実行するＳボックスであり、前記第１段非線形変換部と、前記第２段非線形変換部に含まれる複数の小型非線形変換部は、ｎビットより少ないビット数の非線形変換処理を実行する小型のＳボックスによって構成されていることを特徴とする。

【００２１】

さらに、本発明のデータ変換装置の一実施態様において、前記データ変換装置は、非線形変換処理を伴う暗号処理を実行する構成であることを特徴とする。

【００２２】

さらに、本発明のデータ変換装置の一実施態様において、前記暗号処理は、共通鍵ブロック暗号処理であることを特徴とする。

【００２３】

さらに、本発明の第２の側面は、
データ変換装置において実行するデータ変換方法であり、
第１段非線形変換部において、複数の小型非線形変換部を適用して、入力データを分割した分割データ各々の非線形変換処理を実行する第１段非線形変換ステップと、
線形変換部において、前記第１段非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行する線形変換ステップと、
第２段非線形変換部において、複数の小型非線形変換部を適用して、前記線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行する第２段非線形変換ステップとを有し、
前記線形変換ステップは、

前記入力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行するステップであることを特徴とするデータ変換方法にある。

【００２４】

さらに、本発明の第３の側面は、

データ変換装置においてデータ変換処理を実行させるコンピュータ・プログラムであり、

第１段非線形変換部において、複数の小型非線形変換部を適用して、入力データを分割した分割データ各々の非線形変換処理を実行させる第１段非線形変換ステップと、

線形変換部において、前記第１段非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行させる線形変換ステップと、

第２段非線形変換部において、複数の小型非線形変換部を適用して、前記線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行させる第２段非線形変換ステップとを有し、

前記線形変換ステップは、

前記入力データのビットサイズと同サイズの要素を持つ行列演算によるデータ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行させるステップであることを特徴とするコンピュータ・プログラムにある。

【００２５】

なお、本発明のコンピュータ・プログラムは、例えば、様々なプログラム・コードを実行可能なコンピュータ・システムに対して、コンピュータ可読な形式で提供する記憶媒体、通信媒体、例えば、ＣＤやＦＤ、ＭＯなどの記録媒体、あるいは、ネットワークなどの通信媒体によって提供可能なコンピュータ・プログラムである。このようなプログラムをコンピュータ可読な形式で提供することにより、コンピュータ・システム上でプログラムに応じた処理が実現される。

【００２６】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【発明の効果】

【００２７】

本発明の一実施例の構成によれば、例えば、共通鍵ブロック暗号を適用した暗号処理などにおいて実行される非線形変換処理を、複数の小さなＳボックス（Ｓ－ｂｏｘ）による非線形変換を実行する第１段の非線形変換部と、第１段非線形変換部からの出力をすべて入力して線形変換を実行する線形変換部と、線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部からなる第２段非線形変換部とを適用したデータ変換を行う構成とし、線形変換部では、最適拡散変換を行う行列を適用したデータ変換を実行する構成としたので、データの入力から出力に至るクリティカルパスを過大にすることなく、適切なデータ拡散を実現して、各種の暗号攻撃に対する耐性の強い安全性の高いデータ変換を実現することができる。

【図面の簡単な説明】

【００２８】

【図１】共通鍵ブロック暗号アルゴリズムの基本構成を示す図である。

【図２】図１に示す共通鍵ブロック暗号処理部Ｅ１０の内部構成について説明する図である。

【図３】図２に示す暗号処理部１２の詳細構成について説明する図である。

【図４】ラウンド関数実行部の一構成例としてのＳＰＮ構造ラウンド関数について説明す

る図である。
【図５】ラウンド関数実行部の一構成例としてのＦｅｉｓｔｅｌ（フェイステル）構造について説明する図である。
【図６】ラウンド関数実行部の一構成例としての拡張Ｆｅｉｓｔｅｌ構造について説明する図である。
【図７】非線形変換処理部の具体例について説明する図である。
【図８】線形変換処理部の具体例について説明する図である。
【図９】ＣＲＹＰＴＯＮ　ｖｅｒ．０．５のＳ－ｂｏｘの構成について説明する図である。
【図１０】ＣＲＹＰＴＯＮ　ｖｅｒ．１．０のＳ－ｂｏｘの構成について説明する図である。
【図１１】ＷｈｉｒｌｐｏｏｌのＳ－ｂｏｘの構成について説明する図である。
【図１２】ＦＯＸのＳ－ｂｏｘの構成について説明する図である。
【図１３】本発明の一実施例に係る非線形変換処理構成としてのｎビット入出力Ｓ－ｂｏｘの構成例を示す図である。
【図１４】本発明の一実施例に係る具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの構成例について説明する図である。
【図１５】本発明の一実施例に係る具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの構成例について説明する図である。
【図１６】本発明の一実施例に係る具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの構成例について説明する図である。
【図１７】本発明の一実施例に係る具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの構成例について説明する図である。
【図１８】本発明の一実施例に係る非線形変換を実行するＳ－ｂｏｘの構成例について説明する図である。
【図１９】本発明に係る装置例としての暗号処理装置構成例を示す図である。
【発明を実施するための形態】
【００２９】
以下、本発明の詳細について説明する。説明は、以下の項目に従って行なう。
１．共通鍵ブロック暗号の概要
２．暗号攻撃の概要
３．複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成
（３－１）複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成の有効性について
（３－２）従来型の複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成と問題点
（３－３）本発明に係る複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成
４．暗号処理装置の構成例
【００３０】
［１．共通鍵ブロック暗号の概要］
まず、本発明の適用可能な通鍵ブロック暗号の概要について説明する。本明細書において、共通鍵ブロック暗号（以下ではブロック暗号）は、以下に定義するものを指すものとする。
【００３１】
ブロック暗号は平文Ｐと鍵Ｋを入力し、暗号文Ｃを出力する。平文と暗号文のビット長をブロックサイズと呼びここではｎで示す。ｎは任意の整数値を取りうるが、通常、ブロック暗号アルゴリズムごとに、予め１つに決められている値である。ブロック長がｎのブロック暗号のことをｎビットブロック暗号と呼ぶこともある。
【００３２】

鍵のビット長は、ｋで表す。鍵は任意の整数値を取りうる。共通鍵ブロック暗号アルゴリズムは１つまたは複数の鍵サイズに対応することになる。例えば、あるブロック暗号アルゴリズムＡはブロックサイズｎ＝１２８であり、鍵のビット長ｋ＝１２８、またはｋ＝１９２またはｋ＝２５６の各種の鍵サイズに対応するという構成もありうる。

平文［Ｐ］、暗号文［Ｃ］、鍵［Ｋ］の各ビットサイズは、以下のように示される。

平文Ｐ：ｎビット

暗号文Ｃ：ｎビット

鍵Ｋ：ｋビット

【００３３】

図１にｋビットの鍵長に対応したｎビット共通鍵ブロック暗号アルゴリズムＥの図を示す。図１に示すように、共通鍵ブロック暗号処理部Ｅ１０は、ｎビットの平文Ｐと、ｋビットの鍵Ｋを入力して、予め定められた暗号アルゴリズムを実行して、ｎビットの暗号文Ｃを出力する。なお、図１には平文から暗号文を生成する暗号化処理のみを示している。暗号文から平文を生成する復号処理は、一般的には暗号処理部Ｅ１０の逆関数が用いられる。ただし、暗号処理部Ｅ１０の構造によっては、復号処理においても、同様の共通鍵ブロック暗号処理部Ｅ１０が適用でき、鍵の入力順などのシーケンスの変更によって復号処理が可能となる。

【００３４】

図１に示す共通鍵ブロック暗号処理部Ｅ１０の内部構成について、図２を参照して説明する。ブロック暗号は２つの部分に分けて考えることができる。ひとつは鍵Ｋを入力とし、ある定められたステップにより入力鍵Ｋのビット長を拡大して拡大鍵Ｋ'（ビット長ｋ'）を出力する鍵スケジューリング部１１と、平文Ｐと鍵スケジューリング部１１から入力する拡大鍵Ｋ'を受け取り、平文Ｐを入力して、拡大鍵Ｋ'を適用した暗号処理を実行して、暗号文Ｃを生成するためのデータの変換を実行する暗号処理部１２である。なお、先に説明したように、暗号処理部１２の構造によっては、暗号文を平文に戻すデータ復号処理にも暗号処理部１２が適用可能な場合もある。

【００３５】

次に、図２に示す暗号処理部１２の詳細構成について図３を参照して説明する。暗号処理部１２は、図３に示すように、ラウンド関数実行部２０を適用したデータ変換を繰り返し実行する構成を持つ。すなわち、暗号処理部１２は、ラウンド関数実行部２０という処理単位に分割できる。ラウンド関数実行部２０は入力として、前段のラウンド関数実行部の出力X_iと、拡大鍵に基づいて生成されるラウンド鍵RK_iの２つのデータを受け取り、内部でデータ変換処理を実行して出力データX_{i+1}を次のラウンド関数実行部に出力する。なお、第１ラウンドでは、入力は、平文または平文に対する初期化処理データである。また最終ラウンドの出力は暗号文となる。

【００３６】

図３に示す例では、暗号処理部１２は、ｒ個のラウンド関数実行部２０を有し、ｒ回のラウンド関数実行部におけるデータ変換を繰り返して暗号文を生成する構成となっている。ラウンド関数の繰り返し回数をラウンド数と呼ぶ。図に示す例では、ラウンド数はｒとなる。

【００３７】

各ラウンド関数実行部の入力データX_iは暗号化途中のｎビットデータであり、あるラウンドにおけるラウンド関数の出力X_{i+1}が次のラウンドの入力として供給される。各ラウンド関数実行部のもう一つの入力データは鍵スケジュールから出力された拡大鍵のＫ'に基づくデータが用いられる。この各ラウンド関数実行部に入力され、ラウンド関数の実行に適用される鍵をラウンド鍵と呼ぶ。図で、ｉラウンドに適用するラウンド鍵をRK_iとして示している。拡大鍵Ｋ'は、例えば、ｒラウンド分のラウンド鍵RK_1～RK_rの連結データとして構成される。

【００３８】

図３に示す構成は、暗号処理部１２の入力側から見て１ラウンド目の入力データをX_0

とし、ｉ番目のラウンド関数から出力されるデータをX_i、ラウンド鍵をRK_iとして示した暗号処理部１２の構成である。なお、この暗号処理部１２の構造によっては、例えば、適用するラウンド鍵の適用シーケンスを、暗号化処理と逆に設定し、暗号文を暗号処理部１２に入力することで、復号文を出力する構成にできる。

【００３９】

図３に示す暗号処理部１２のラウンド関数実行部２０は、さまざまな形態をとりうる。ラウンド関数はその暗号アルゴリズムが採用する構造（ｓｔｒｕｃｔｕｒｅ）によって分類できる。代表的な構造として、

（ア）ＳＰＮ（Substitution Permutation Network）構造、

（イ）Ｆｅｉｓｔｅｌ構造、

（ウ）拡張Ｆｅｉｓｔｅｌ構造、

がある。以下、これらの具体的構成について、図４～図６を参照して説明する。

【００４０】

（ア）ＳＰＮ構造ラウンド関数

まず、図４を参照して、ラウンド関数実行部２０の一構成例としてのＳＰＮ構造ラウンド関数について説明する。ＳＰＮ構造ラウンド関数実行部２０ａは、非線形変換層（Ｓ層）と線形変換層（Ｐ層）を接続したいわゆるＳＰ型の構成を有する。図４に示すように、ｎビットの入力データすべてに対して、ラウンド鍵との排他的論理和（ＸＯＲ）演算を実行する排他的論理和演算部２１、排他的論理和演算部２１の演算結果を入力し、入力データの非線形変換を実行する非線形変換処理部２２、非線形変換処理部２２における非線形変換処理結果を入力し、入力データに対する線形変換処理を実行する線形変換処理部２３などによって構成される。線形変換処理部２３の線形変換処理結果が、次のラウンドに出力される。最終ラウンドでは暗号文となる。なお、図４に示す例では、排他的論理和演算部２１、非線形変換処理部２２、線形変換処理部２３の処理順を示しているが、これらの処理部の順番は、限定されるものではなく、他のシーケンスで処理を行なう構成としてもよい。

【００４１】

（イ）Ｆｅｉｓｔｅｌ構造

次に、図５を参照してラウンド関数実行部２０の一構成例としてのＦｅｉｓｔｅｌ（フェイステル）構造について説明する。Ｆｅｉｓｔｅｌ構造は、図５に示すように、前ラウンドからの入力（第１ラウンドでは入力文）であるｎビットの入力データをｎ／２ビットの２つのデータに分割して、各ラウンドにおいて入れ替えながら処理を実行する。

【００４２】

Ｆｅｉｓｔｅｌ構造を持つラウンド関数実行部２０ｂを適用した処理においては、図に示すように、一方のｎ／２ビットデータとラウンド鍵とがＦ関数部３０に入力される。Ｆ関数部３０は、上述したＳＰＮ構造と同様、非線形変換層（Ｓ層）と線形変換層（Ｐ層）を接続したいわゆるＳＰ型の構成を有する。

【００４３】

前ラウンドからのｎ／２ビットデータとラウンド鍵とがＦ関数部３０の排他的論理和演算部３１に入力され排他的論理和（ＸＯＲ）処理がなされる。さらに、この結果データを非線形変換処理部３２に入力して非線形変換を実行し、さらに、この非線形変換結果が線形変換処理部３３に入力され線形変換が実行される。この線形変換結果が、Ｆ関数処理結果データとして出力する。

【００４４】

さらに、このＦ関数出力と、前ラウンドから入力するもう１つのｎ／２ビット入力とを、排他的論理和演算部３４に入力し、排他的論理和演算（ＸＯＲ）を実行して、実行結果を、次のラウンドにおけるＦ関数の入力として設定される。なお、図に示す第ｉラウンドにおけるＦ関数入力に設定されたｎ／２ビットは次のラウンドのＦ関数出力との排他的論理和演算に適用される。このように、Ｆｅｉｓｔｅｌ構造は、各ラウンドにおいて入力を交互に入れ替えながらＦ関数を適用したデータ変換処理を実行する。

【００４５】
（ウ）拡張Ｆｅｉｓｔｅｌ構造
次に、図６を参照してラウンド関数実行部２０の一構成例としての拡張Ｆｅｉｓｔｅｌ構造について説明する。先に、図５を参照して説明したＦｅｉｓｔｅｌ構造は、ｎビットの平文を２つに分割してｎ／２ビットずつに区分して処理を実行していた。すなわち、分割数：ｄ＝２とした処理である。なお、この分割数は、データ系列数とも呼ばれる。
【００４６】
拡張Ｆｅｉｓｔｅｌ構造では、このデータ系列数（分割数）ｄを３以上の任意の整数とした設定である。データ系列数（分割数）ｄの値に応じたさまざまな拡張Ｆｅｉｓｔｅｌ構造を定義することができる。図６に示す例では、データ系列数（分割数）ｄ＝４であり、各系列に対してｎ／４ビットのデータが入力される。各ラウンドでは、１つ以上のラウンド関数としてのＦ関数が実行される。図に示す例は、１ラウンドにおいて２つのＦ関数部によるラウンド演算を行なう構成例である。
【００４７】
Ｆ関数部４１，４２の構成は、先に図５を参照して説明したＦ関数部３０の構成と同様であり、ラウンド鍵と入力値との排他的論理和演算と、非線形変換処理、線形変換処理を実行する構成となっている。なお、各Ｆ関数部に入力するラウンド鍵は、入力ビットとビット数が一致するように調整される。図に示す例では、各Ｆ関数部４１，４２に入力するラウンド鍵は、ｎ／４ビットとなる。これらの鍵は、拡大鍵を構成するラウンド鍵をさらにビット分割して生成する。なお、データ系列数（分割数）をｄとしたとき、各系列に入力されるデータはｎ／ｄビットであり、各Ｆ関数に入力する鍵のビット数もｎ／ｄビットに調整される。
【００４８】
なお、図６に示す拡張Ｆｅｉｓｔｅｌ構造では、データ系列数（分割数）をｄとして、各ラウンドにおいてｄ／２のＦ関数を並列に実行する構成例であるが、拡張Ｆｅｉｓｔｅｌ構造は、各ラウンドにおいて、１つ以上ｄ／２以下のＦ関数を実行する構成が可能である。
【００４９】
図４～図６を参照して説明したように、共通鍵ブロック暗号における暗号処理部１２のラウンド関数実行部２０は、
（ア）ＳＰＮ（Substitution Permutation Network）構造、
（イ）Ｆｅｉｓｔｅｌ構造、
（ウ）拡張Ｆｅｉｓｔｅｌ構造、
これらの構造をとり得る。これらのラウンド関数実行部は、いずれも非線形変換層（Ｓ層）と線形変換層（Ｐ層）を接続したいわゆるＳＰ型の構成を有する。すなわち、非線形変換処理を実行する非線形変換処理部と、線形変換処理を実行する線形変換処理部とを有する。以下、これらの変換処理構成について説明する。
【００５０】
（非線形変換処理部）
非線形変換処理部の具体例について、図７を参照して説明する。図７に示すように、非線形変換処理部５０は、具体的には、Ｓボックス（Ｓ－ｂｏｘ）５１と呼ばれるｓビット入力ｓビット出力の非線形変換テーブルがｍ個並んだものであり、ｍｓビットの入力データがｓビットずつ分割されてそれぞれ対応するＳボックス（Ｓ－ｂｏｘ）５１に入力されデータが変換される。各Ｓボックス５１では、例えば変換テーブルを適用した非線形変換処理が実行される。
【００５１】
入力されるデータのサイズが大きくなると実装上のコストが高くなる傾向がある。それを回避するために、図７に示すように、処理対象データＸを複数の単位に分割し、それぞれに対して、非線形変換を施す構成がとられることが多い。例えば入力サイズをｍｓビットとしたとき、ｓビットずつのｍ個のデータに分割して、ｍ個のＳボックス（Ｓ－ｂｏｘ

）５１それぞれに対してｓビットを入力し、例えば変換テーブルを適用した非線形変換処理を実行して、これらの各Ｓビット出力ｍ個を合成してｍｓビットの非線形変換結果を得る。

【００５２】

（線形変換処理部）

線形変換処理部の具体例について、図８を参照して説明する。線形変換処理部は、入力値、例えば、Ｓボックスからの出力データであるｍｓビットの出力値を入力値Ｘとして入力し、この入力に対して線形変換を施しｍｓビットの結果を出力する。線形変換処理は、例えば、入力ビット位置の入れ替え処理などの線形変換処理を実行して、ｍｓビットの出力値Ｙを出力する。線形変換処理は、例えば、入力に対して、線形変換行列を適用して入力ビット位置の入れ替え処理を行なう。この行列の一例が図８に示す線形変換行列である。

【００５３】

線形変換処理部において適用する線形変換行列の要素は拡大体：ＧＦ（2^8）の体の要素やＧＦ（２）の要素など、一般的にはさまざまな表現を適用した行列として構成できる。図８は、ｍｓビット入出力をもち、ＧＦ（2^s）の上で定義されるｍ×ｍの行列により定義される線形変換処理部の１つの構成例を示すものである。

【００５４】

［２．暗号攻撃の概要］

上述した共通鍵ブロック暗号に対する暗号攻撃、すなわち、適用している鍵の解析やアルゴリズムの解析手法として様々な手法が知られている。以下、この暗号攻撃の概要について説明する。以下の各攻撃について、順次説明する。

（２－１）差分攻撃

（２－２）線形攻撃

（２－３）高階差分攻撃

（２－４）補間攻撃

【００５５】

（２－１）差分攻撃

共通鍵暗号の攻撃法の一つとして差分解読法（Ｄｉｆｆｅｒｅｎｔｉａｌ Ｃｒｙｐｔａｎａｌｙｓｉｓ）がある。差分解読法については、例えば、文献「Ｅ．Ｂｉｈａｍ，Ａ．Ｓｈａｍｉｒ，"Ｄｉｆｆｅｒｅｎｔｉａｌ Ｃｒｙｐｔａｎａｌｙｓｉｓ ｏｆ ＤＥＳ－ｌｉｋｅ Ｃｒｙｐｔｏｓｙｓｔｅｍｓ，"Ｊｏｕｒｎａｌ ｏｆ Ｃｒｙｐｔｏｌｏｇｙ，Ｖｏｌ．４，Ｎｏ．１，ｐｐ．３－７２，１９９１．」に記述されている。

【００５６】

この攻撃法はある暗号に対し差分値と呼ばれるデータの伝播を観測し、高確率でその差分値の伝播が起こってしまった場合に攻撃、つまり鍵の推定ができるという攻撃法である。この差分値と呼ばれるデータの伝播の確率のことを差分確率と呼ぶ。

【００５７】

ｎビット入出力の関数ｆに対して、入力をｘ（ｎ－ｂｉｔ）、入力差分値をΔｘ（ｎ－ｂｉｔ）、出力差分値をΔｙ（ｎ－ｂｉｔ）と定義したとき、関数ｆに対する入力差分値Δｘ、出力差分値Δｙの差分確率DP_f（Δｘ、Δｙ）は以下のように定義される。

【数１】

$$DP_f(\Delta x,\Delta y)=\frac{\#\{x\in\{0,1\}^n \mid f(x)\oplus f(x\oplus\Delta x)=\Delta y\}}{2^n}$$

【００５８】

また、関数ｆに対する最大差分確率MDP_fは以下のように定義される。

【数２】

$$MDP_f = \max_{\Delta x \neq 0, \Delta y} DP_f(\Delta x, \Delta y)$$

【００５９】

これらの値は関数ｆによって一意に求まる値であり、最大差分確率ＭＤＰ$_f$が大きい関数ｆは差分攻撃に弱い関数ということになる。よって、暗号を設計する際はできる限りＭＤＰ$_f$が小さくなるよう関数ｆを設計することが望ましい。

【００６０】

（２－２）線形攻撃

さらに、共通鍵暗号の攻撃法の一つとして線形解読法（Ｌｉｎｅａｒ　Ｃｒｙｐｔａｎａｌｙｓｉｓ）がある。線形解読法については、例えば、文献「Ｍ．Ｍａｔｓｕｉ，"Ｌｉｎｅａｒ　Ｃｒｙｐｔａｎａｌｙｓｉｓ　Ｍｅｔｈｏｄ　ｆｏｒ　ＤＥＳ　Ｃｉｐｈｅｒ，"ＥＵＲＯＣＲＹＰＴ'９３，ＬＮＣＳ　７６５，ｐｐ．３８６－３９７，１９９４．」に記述されている。

【００６１】

この攻撃法はある暗号に対し、入力の特定ビットの排他的論理和と出力の特定ビットの排他的論理和の間の相関を観測し、強い相関関係が見つかってしまった場合に攻撃、つまり鍵の推定ができるという攻撃法である。この入出力の特定ビットの相関係数のことを線形確率と呼ぶ。

【００６２】

ｎビット入出力の関数ｆに対して、入力をｘ（ｎ－ｂｉｔ）、入力マスク値をΓｘ（ｎ－ｂｉｔ）、出力マスク値をΓｙ（ｎ－ｂｉｔ）と定義したとき、関数ｆに対する入力マスク値Γｘ、出力マスク値Γｙの線形確率ＬＰ$_f$（Γｘ、Γｙ）は以下のように定義される。

【数３】

$$LP_f(\Gamma x, \Gamma y) = \left(2 \cdot \frac{\#\{x \in \{0,1\}^n \mid x \bullet \Gamma x = f(x) \bullet \Gamma y\}}{2^n} - 1 \right)^2$$

ただし、上記式において［・］はｎビットベクトル同士の内積演算を表す。

【数４】

$$x \bullet y = \oplus_{i=1}^{n} (x_i \cdot y_i)$$

【００６３】

また、関数ｆに対する最大線形確率ＭＬＰ$_f$は以下のように定義される。

【数５】

$$MLP_f = \max_{\Gamma x,\Gamma y \neq 0} LP_f(\Gamma x,\Gamma y)$$

【００６４】

これらの値は関数ｆによって一意に求まる値であり、最大線形確率ＭＬＰ$_f$が大きい関数ｆは線形攻撃に弱い関数ということになる。よって、暗号を設計する際はできる限りＭＬＰ$_f$が小さくなるよう関数ｆを設計することが望ましい。

【００６５】

（２－３）高階差分攻撃

さらに、共通鍵暗号の攻撃法の一つとして高階差分攻撃法（Ｈｉｇｈｅｒ　Ｏｒｄｅｒ　Ｄｉｆｆｅｒｅｎｔｉａｌ　Ａｔｔａｃｋ）がある。高階差分攻撃法については、例えば、文献「Ｌ．Ｒ．Ｋｎｕｄｓｅｎ、"Ｔｒｕｎｃａｔｅｄ　ａｎｄ　Ｈｉｇｈｅｒ　Ｏｒｄｅｒ　Ｄｉｆｆｅｒｅｎｔｉａｌｓ，"ＦＳＥ　'９４、　ＬＮＣＳ　１００８，ｐｐ．１９６－２１１」に記述されている。

【００６６】

この攻撃法は、暗号アルゴリズムの持つ代数的性質を利用した攻撃法であり、暗号文を平文の全ビットでブール多項式表現した際に、そのブール代数次数が平文の特定数ビットに対し、ある次数より小さくなっているような場合に攻撃、つまり鍵の推定ができるという攻撃法である。

【００６７】

（２－４）補間攻撃

さらに、共通鍵暗号の攻撃法の一つとして補間攻撃法（Ｉｎｔｅｒｐｏｌａｔｉｏｎ　Ａｔｔａｃｋ）がある。補間攻撃については、例えば、文献「Ｔ．Ｊａｋｏｂｓｅｎ　ａｎｄ　Ｌ．Ｒ．Ｋｎｕｄｓｅｎ，"Ｔｈｅ　Ｉｎｔｅｒｐｏｌａｔｉｏｎ　Ａｔｔａｃｋ　ｏｎ　Ｂｌｏｃｋ　Ｃｉｐｈｅｒ，"ＦＳＥ'９７，ＬＮＣＳ　１２６７，ｐｐ．２８－４０，１９９７．」に記述されている。

【００６８】

この攻撃法は、暗号アルゴリズムの持つ代数的性質を利用した攻撃法であり、暗号化関数を多項式関数として表現した際のその項数が少ない場合に、鍵値を含んだ暗号化関数を復元することで攻撃を行う攻撃法である。

【００６９】

［３．複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成］

（３－１）複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成の有効性について

上述したように、共通鍵ブロック暗号は、ラウンド関数の繰り返しによる暗号処理を行なう構成である。この共通鍵ブロック暗号処理は、ラウンド関数を繰り返し実行する構成であり、ラウンド関数では、線形変換処理、および非線形変換処理が実行される。非線形変換処理においては、例えば、図７を参照して説明したように、Ｓ－ｂｏｘ（Ｓボックス）を適用した非線形変換処理が実行される。上述した共通鍵ブロック暗号処理の他にも、例えばハッシュ関数などのデータ変換においても非線形変換処理にＳボックスが適用可能である。

【００７０】

先に、説明したように、このＳボックスの入出力サイズは一般的には小さく、厳密な安全性評価が可能であり、暗号やハッシュアルゴリズム全体の安全性を高くするためにＳボックスに求められる特性として以下の特性が知られている。

（１）最大差分確率が十分小さい

（２）最大線形確率が十分小さい
（３）ブール多項式表現を行った際のブール代数次数が十分大きい
（４）入出力を多項式表現した際の項数が十分多い

【００７１】
主に、（１）は差分攻撃に対する耐性、（２）は線形攻撃に対する耐性、（３）は高階差分攻撃に対する耐性、（４）は補間攻撃に対する耐性をそれぞれ決定付ける。さらには、Ｓボックスの入出力のビット相関が低いことや入力を１ビット変化させたときの出力の変化が１／２程度になることなどが安全性を高めるために重要である。

【００７２】
また、Ｓ－ｂｏｘは安全性の要求とは別に高い実装性能も要求される。例えば、ソフトウェアによる暗号処理やハッシュ関数を実行させる実装構成においては通常、入力に対する出力を示すテーブルをメモリに保持して、表引き（テーブル実装）と呼ばれる手法により、非線形変換処理を実行する構成とした実装がなされるため、その実装性能はＳ－ｂｏｘの内部構造にはあまり依存しない。しかしながら、ハードウェア実装においては、例えば入力値に基づいて、特定の出力を算出するための回路が構成される。この回路構成は、適用するＳ－ｂｏｘにより、構成が大きく変化することになり、回路規模に影響を与えることになる。

【００７３】
暗号アルゴリズムやハッシュアルゴリズムの安全性維持のためにＳ－ｂｏｘに求められる特性として、前述したように、
（１）最大差分確率が十分小さい
（２）最大線形確率が十分小さい
（３）ブール多項式表現を行った際のブール代数次数が十分大きい
（４）入出力を多項式表現した際の項数が十分多い
これらの特性があるが、このような要求を満たすＳ－ｂｏｘを効率よく生成する手法として、拡大体上のべき乗関数を用いる方法がよく知られている。この手法で拡大体の次数とべき乗の乗数を適切に選べば、非常に特性の良いＳ－ｂｏｘが生成できる。

【００７４】
実際にｎ－ｂｉｔ入出力Ｓ－ｂｏｘの入力をｘ、出力をｙとして、それぞれが拡大体ＧＦ（２^{n}）の元だとした場合、
ｙ＝ｆ（ｘ）が、
【数６】

$$y = x^{2^t+1}$$

（ｔは任意の整数）

$$y = x^{-1}$$

【００７５】
で与えられる場合、最大差分確率、最大線形確率の点で最適なＳ－ｂｏｘが構成できることがよく知られている。

【００７６】
このような手法で構成されたＳ－ｂｏｘの例としては、ＡＥＳ、Ｃａｍｅｌｌｉａ、ＭＩＳＴＹにおいて適用されているＳ－ｂｏｘなどが挙げられる。
ＡＥＳやＣａｍｅｌｌｉａのＳ－ｂｏｘはＧＦ（（２^{4}）^{2}）上の逆元関数としてＳ－ｂｏｘを構成することもできるため、安全性の観点からも特性が非常に良く、さらに非常に高いハードウェア実装性能も持つと言える。

【００７７】
しかしながら、近年このようなＳ－ｂｏｘに関しても、その特徴的な代数的性質を利用した攻撃法や、そのあまりに均一的な拡散性に関する問題点などが指摘されるようになった。
【００７８】
このような点から、特徴的な代数的性質を持たないＳ－ｂｏｘの構成法が考えられ、強い代数的構造を持たないように、ランダム、もしくはそれに準ずるような手法で要素を選択する手法によりＳ－ｂｏｘを生成する方法が考えられた。このような手法でＳ－ｂｏｘを生成すると、多くの場合、特徴的な代数的構造は持たず、さらに拡大体上のべき乗関数のような均一的な拡散はしないため、上記のような問題に関しては対策がされていると言うことができる。
【００７９】
しかしながら、ｎ－ｂｉｔ入出力の全単射Ｓ－ｂｏｘの数は、2^nの階乗個存在するため、実際にｎ－ｂｉｔ入出力のＳ－ｂｏｘをランダムに生成し、その特性をひとつひとつチェックするような手法ではｎがある程度大きくなると効率的に特性の良いＳ－ｂｏｘを生成することは難しくなる。
【００８０】
また、Ｓ－ｂｏｘの要素を完全にランダムに選択した場合、ハードウェアで実装する際も主にテーブル実装と呼ばれる手法しか用いることができなくなってしまうため、実装効率が大きく低下する。
【００８１】
このような問題から、小さなサイズの入出力を持つＳ－ｂｏｘをランダムに生成して、それらを複数個組み合わせて、より大きなＳ－ｂｏｘを生成するという試みがなされている。例えば、
ＣＲＹＰＴＯＮ　ｖｅｒ．０．５、
ＣＲＹＰＴＯＮ　ｖｅｒ．１．０、
Ｗｈｉｒｌｐｏｏｌ、
ＦＯＸ
などのＳ－ｂｏｘなどがこのような手法に基づいて生成されたＳ－ｂｏｘだと考えることができる。以下、これらの具体的な構成例およびその問題点について説明する。
【００８２】
（３－２）従来型の複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成と問題点
以下、従来型の複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成と問題点について、以下の各構成について説明する。
（ａ）ＣＲＹＰＴＯＮ　ｖｅｒ．０．５、
（ｂ）ＣＲＹＰＴＯＮ　ｖｅｒ．１．０、
（ｃ）Ｗｈｉｒｌｐｏｏｌ、
（ｄ）ＦＯＸ
【００８３】
（ａ）ＣＲＹＰＴＯＮ　ｖｅｒ．０．５、
例えば、ＣＲＹＰＴＯＮ　ｖｅｒ．０．５のＳ－ｂｏｘは、図９に示す構造を持つ。Ｓ－ｂｏｘ［$S4_0$］１０１、Ｓ－ｂｏｘ［$S4_1$］１０２、Ｓ－ｂｏｘ［$S4_2$］１０３は、４ビット入出力のＳ－ｂｏｘであり、これらの３つのＳ－ｂｏｘと、排他的論理和演算部１１１～１１３が組み合わされて、入力８ビットを非線形変換して、出力８ビットを得る構成を持つ。なお、図に示すＳ－ｂｏｘの表記［$S4_n$］において、４は４ビット入出力のＳボックスであることを示し、ｎは、Ｓ－ｂｏｘの識別子である。Ｓ－ｂｏｘ［$S4_0$］１０１、Ｓ－ｂｏｘ［$S4_1$］１０２、Ｓ－ｂｏｘ［$S4_2$］１０３は、すべて４ビット入出力のＳ－ｂｏｘであり、それぞれ異なる非線形変換処理を実行する異なるＳ－ｂｏｘである。

【００８４】
この構造は、いわゆる３段Ｆｅｉｓｔｅｌ構造を採用しているためブール代数次数が低くなってしまい、さらにクリティカルパスが長い。すなわち、入力から最終出力を得るまでには３つの小さなＳ－ｂｏｘを通過しなければいけない構成となっている。従って、実装性能があまり高くないという特徴を持つ。
【００８５】
（ｂ）ＣＲＹＰＴＯＮ　ｖｅｒ．１．０、
また、ＣＲＹＰＴＯＮ　ｖｅｒ．１．０のＳ－ｂｏｘは図１０に示す構造を持つ。Ｓ－ｂｏｘ［$S4_0$］１２１、Ｓ－ｂｏｘ［$S4_1$］１２２と、Ｓ－ｂｏｘ［$S4_1$］１２２の逆変換を実行するＳ－ｂｏｘ［$S4_1^{-1}$］１２３、Ｓ－ｂｏｘ［$S4_0$］１２１の逆変換を実行するＳ－ｂｏｘ［$S4_0^{-1}$］１２４と、ビット演算による線形変換処理を実行するビット演算部１３１，１３２から構成される。
【００８６】
入力８ビットの上位４ビットをＳ－ｂｏｘ［$S4_0$］１２１において、非線形変換を実行した後、ビット演算部１３１に入力した線形変換を行い、その結果をＳ－ｂｏｘ［$S4_1^{-1}$］１２３に入力して非線形変換結果を出力する。下位４ビットについては、Ｓ－ｂｏｘ［$S4_1$］１２２において、非線形変換を実行した後、ビット演算部１３１に入力した線形変換を行い、その結果をＳ－ｂｏｘ［$S4_0^{-1}$］１２３に入力して非線形変換結果を出力する。
【００８７】
この構造は、中間で用いているビット演算部１３１におけるビット演算を１－ｂｉｔ要素８ｘ８の行列、つまり６４ビットの空間から選択しなければいけないため、その選択が難しくなってしまう。すなわち、非線形変換では、基本的に非線形変換対象の入力８ビットから、各入力８ビットに対応する非線形変換結果としての出力８ビットを偏りのない出力が得られる構成とすることが要求され、中段のビット演算部１３１，１３２において適用する行列を、この要求を満足する行列とするための選択が困難になるという問題がある。
【００８８】
（ｃ）Ｗｈｉｒｌｐｏｏｌ、
また、ＷｈｉｒｌｐｏｏｌのＳ－ｂｏｘは図１１のような構造をしている。Ｓ－ｂｏｘ［$S4_0$］１４１、１４４，Ｓ－ｂｏｘ［$S4_0$］の逆変換を実行するＳ－ｂｏｘ［$S4_0^{-1}$］１４２，１４５と、Ｓ－ｂｏｘ［$S4_1$］１４３を用い、排他論理和演算部１５１～１５３によって、これらのＳ－ｂｏｘを結合した構成を持つ。
【００８９】
この構造では、必要な小さなＳ－ｂｏｘの数が５個と多く、さらにクリティカルパスが長い。すなわち、入力から最終出力に至るまでには３つの小さなＳ－ｂｏｘを通過しなければいけない構成であり、実装性能があまり高くないという問題がある。
【００９０】
（ｄ）ＦＯＸ
さらに、ＦＯＸのＳ－ｂｏｘは図１２に示す構成をしている。３種類の４ビット入出力Ｓ－ｂｏｘ１６１～１６３と、４ビット入出力ＯＲ回路１７１，１７２を用い、これらを排他的論理和演算部によって接続した構成を持つ。
【００９１】
この構造では、必要な小さなＳ－ｂｏｘの数は３個と少ないが、全体的に排他的論理和演算部（ＸＯＲ）の数が多く、クリティカルパスも長い。入力から最終出力を得るためには３つの小さなＳ－ｂｏｘを通過しなければいけない。このため、実装性能があまり高くないという問題がある。
【００９２】
（３－３）本発明に係る複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成

次に、本発明に係る複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成について説明する。上述したように、複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成については、すでに複数のタイプが提案されているが、それぞれクリティカルパスが長いという問題点や、線形変換行列の設定の困難性等の問題点を有している。

【００９３】

まず、以下の本発明に従った非線形変換処理構成の理解に必要となる以下の用語について説明する。

（ａ）分岐数および最適拡散層

（ｂ）拡大体上の行列演算

【００９４】

（ａ）分岐数および最適拡散層

まず、分岐数、および最適拡散変換（Ｏｐｔｉｍａｌ　Ｄｉｆｆｕｓｉｏｎ　Ｍａｐｐｉｎｇｓ）について説明する。

【００９５】

ｎ×ａビットデータからｎ×ｂビットデータへの写像

θ：｛０，１｝ｎａ→｛０，１｝ｎｂ

に対して分岐数Ｂｎ（θ）を次のように定義する。

【数７】

$$B_n(\theta) = \min_{\alpha \neq 0} \{ hw_n(\alpha) + hw_n(\theta(\alpha)) \}$$

【００９６】

ただし、$\mathrm{min}_{\alpha \neq 0}\{X_\alpha\}$はα≠０を満たすすべての$X_\alpha$のうちの最小値を表すものとし、$hw_n$（Ｙ）はビット列Ｙをｎビットごとに区切って表したときにｎビットのデータすべてが０ではない（非ゼロ）要素の数を返す関数とする。このとき、分岐数Ｂ（θ）がｂ＋１であるような写像θを最適拡散変換（Ｏｐｔｉｍａｌ　Ｄｉｆｆｕｓｉｏｎ　Ｍａｐｐｉｎｇｓ）と定義する。また、以降ではＯＤＭを満たす行列をＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列と定義する。

【００９７】

（ｂ）拡大体上の行列演算

基礎体ＧＦ（２）上で定義されるｎ次の既約多項式ｐ（ｘ）によって作られる拡大体ＧＦ（2^n）上で行われる行列演算を拡大体上の行列演算と呼ぶ。

例えば、

【数８】

$$\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$$

上記の行列演算を考える。

上記の行列演算において、出力y_0，y_1はそれぞれ、

$y_0 = a_{0,0} x_0 + a_{0,1} x_1$

$y_1 = a_{1,0} x_0 + a_{1,1} x_1$

と表現される。

【００９８】
このとき$a_{0,0}$，x_0などといった変数を全て、拡大体ＧＦ（2^n）上の元だと考え、演算を全て拡大体ＧＦ（2^n）上で行う。このような方法で行列演算を行う手法を拡大体上の行列演算と呼ぶ。
【００９９】
本発明に係る非線形変換処理構成は、上述した従来の複数の小型Ｓボックスの組み合わせからなる非線形変換処理構成における問題点、すなわち、クリティカルパスが長いという問題点や、線形変換行列の設定の困難性等の問題点を解決した非線形変換処理構成を実現する。
【０１００】
本発明の構成では、拡大体ＧＦ（2^n）上で簡単な表現を行うことが難しく、かつ高い実装性能（特にハードウェア）を持つｎ－ｂｉｔ入出力の全単射Ｓ－ｂｏｘを効率よく生成するために、小さなＳ－ｂｏｘの組み合わせ位置に分岐数の高い小さなＳ－ｂｏｘの入出力サイズと同サイズの要素を持つ行列演算を組み合わせて、大きなＳ－ｂｏｘとしての非線形変換処理構成を実現するものである。具体的に分岐数の高い行列演算というのは，ｍ×ｍ行列に対して、ｍ以上の分岐数を持つ行列を表す。
【０１０１】
分岐数の高い行列として、例えば、前述した最適拡散変換を実行するＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を選択する。ＭＤＳ行列は、行列を構成する任意の小行列が正則行列となる行列である。なお、正則行列は、逆行列を持つ行列であり、行列をＡとし、逆行列をA^{-1}とすると、
$AA^{-1}=A^{-1}A=E$、
ただしＥは単位行列、
上記式が成立する逆行列A^{-1}を持つ行列Ａが正則行列である。
【０１０２】
本発明の一実施例に係る非線形変換処理構成としてのｎビット入出力Ｓ－ｂｏｘの構成例を図１３に示す。図１３に示す構成は、４つのｎ／２ビット入出力の全単射Ｓ－ｂｏｘ２０１～２０４と、拡大体ＧＦ（$2^{n/2}$）において定義される２×２ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列からなる線形変換処理部２１１から構成されるｎビット入出力の非線形変換処理を実行する構成である。
【０１０３】
入力ｎビットは、上位ｎ／２ビット、下位ｎ／２ビットずつ、それぞれｎ／２ビット入出力の全単射Ｓ－ｂｏｘ［S_0］２０１，Ｓ－ｂｏｘ［S_1］２０２に入力され、それぞれのＳ－ｂｏｘにおいて非線形変換処理がなされた後、拡大体ＧＦ（$2^{n/2}$）において定義される２×２ＭＤＳ行列からなる線形変換処理部２１１に入力されＭＤＳ行列に基づく線形変換処理がなされた後、線形変換結果のｎビットの上位ｎ／２ビット、下位ｎ／２ビットずつ、それぞれｎ／２ビット入出力の全単射Ｓ－ｂｏｘ［S_2］２０３，Ｓ－ｂｏｘ［S_3］２０４に入力され、それぞれのＳ－ｂｏｘにおいて非線形変換処理がなされて出力結果としてのｎビットを得る構成となっている。
【０１０４】
図１３に示す線形変換処理部２１１において適用されるＭＤＳ行列のように分岐数の高い拡散変換行列を利用すると、前段の小さなＳ－ｂｏｘ［S_0］２０１，Ｓ－ｂｏｘ［S_1］２０２の出力が後段の小さなＳ－ｂｏｘ［S_2］２０３，Ｓ－ｂｏｘ［S_3］２０４の入力にバランスよく配置されることが保証されるため、全体として特性の良い大きなＳ－ｂｏｘが生成できる。また、線形変換処理部２１１において適用される拡散変換行列に相当するＭＤＳ行列の要素のサイズは小さなＳ－ｂｏｘの入出力サイズ（ｎ／２）によって一意に決まるため、大きなＳ－ｂｏｘの入出力サイズ（ｎ）が大きくなったとしても選択できる行列の空間は限られるため、容易にその行列の分岐数をチェックすることができる。
【０１０５】

図１３に示す構成における線形変換処理部２１１において適用される拡散変換行列としてのＭＤＳ行列の要素のサイズはｎ／２ビットであり、２×２の行列であるため、要素の数は４つとなり、全行列を調べたとしてもその数は、$2^{n/2\times4}$種程度であるため、容易に全行列を調べることが可能である。ただし、行列演算はＧＦ（２）上で定義されるｎ次の既約多項式ｐ（ｘ）によって作られる拡大体ＧＦ（2^{n}）上の行列演算とする。

【０１０６】

この手法でＳ－ｂｏｘを生成すると、簡易な代数表現が可能なのは、図１３に示す構成中の入出力サイズｎ／２ビットの小さなＳ－ｂｏｘ２０１～２０４に限られると考えられるため、拡大体ＧＦ（2^{n}）上での要素はランダムのように見える。よって、ＧＦ（2^{n}）上で単純な代数的表現を行うことは難しいと考えることができる。また、拡大体上でのべき乗演算のように均一的な拡散はしないことが期待される。

【０１０７】

さらに、行列を適切に選択すれば、簡易な演算と短いパスで構成することが可能なため、ｎ－ｂｉｔでランダムに生成されたＳ－ｂｏｘや、従来の小さなＳ－ｂｏｘを組み合わせて構成したＳ－ｂｏｘよりもより実装効率は高いと言える。

【０１０８】

図１３に示す構成を適用した具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの構成例について、図１４～図１７を参照して説明する。

【０１０９】

図１４（ａ）、図１５（ａ）に示す構成は同一であり、図１３に示す構成を適用した具体的な８ビット入出力の非線形変換を実行するＳ－ｂｏｘの一構成例としての８ビットＳ－ｂｏｘ３００の構成を示す図である。図１４（ｂ）は、図１４（ａ）に示す８ビットＳ－ｂｏｘ３００の構成要素である４つの４ビット入出力Ｓ－ｂｏｘ３０１～３０４の変換テーブルを示している。このテーブルで用いている数値は１６進数である。入力：ｘ＝０～Ｆ（１６進数）に対する各Ｓ－ｂｏｘ３０１～３０４の出力値を示すテーブルである。また、図１５（ｃ）には、図１５（ａ）（＝図１４（ａ））に示す線形変換処理部３１１として分岐数の高い拡散変換、具体的には最適拡散変換［ＯＤＭ］であるＭＤＳ行列（ｍａｔｒｉｘ）３１１を使用した具体例を示している。

【０１１０】

図１４（ａ）に示す８ビット入出力の非線形変換を実行するＳ－ｂｏｘの処理について説明する。まず、８ビットＳ－ｂｏｘ３００の入力であるｘ（８－ｂｉｔ）は１／２、つまり４－ｂｉｔごとにｘ［０］，ｘ［１］に分割される。次にｘ［０］は４－ｂｉｔ入出力のＳ－ｂｏｘ［$S4_0$］３０１へ入力され、ｘ［１］は４－ｂｉｔ入出力のＳ－ｂｏｘ［$S4_1$］３０２へそれぞれ入力され、各Ｓ－ｂｏｘ３０１，３０２における非線形変換結果としての出力ｗ［０］，ｗ［１］を得る。Ｓ－ｂｏｘ［$S4_0$］３０１、Ｓ－ｂｏｘ［$S4_1$］３０２は、ランダムに選択した４ビットＳ－ｂｏｘであり、非線形変換は、図１４（ｂ）のテーブルに従った変換処理として実行される。

【０１１１】

なお、Ｓ－ｂｏｘ［$S4_0$］３０１、Ｓ－ｂｏｘ［$S4_1$］３０２は、最大差分確率が十分低いことや最大線形確率が十分低いことなどＳ－ｂｏｘが満たすべきあらゆる性質を満たすようなＳ－ｂｏｘである。また、これらのＳ－ｂｏｘは４－ｂｉｔ入出力なので、このような性質を満たすＳ－ｂｏｘを探索することが現実的に可能である。

【０１１２】

Ｓ－ｂｏｘ［$S4_0$］３０１、Ｓ－ｂｏｘ［$S4_1$］３０２の出力ｗ［０］，ｗ［１］は、最適拡散変換［ＯＤＭ］であるＭＤＳ行列（ｍａｔｒｉｘ）による線形変換を実行する線形変換処理部３１１に入力され、ＭＤＳ行列（ｍａｔｒｉｘ）による線形変換結果として、ｚ［０］，ｚ［１］を出力する。線形変換処理部３１１では、ＧＦ（２）上で定義される４次の既約多項式：ｐ（ｘ）＝x^4+x+1によって作られる拡大体ＧＦ（2^4）上の行列演算による線形変換が実行され、ｚ［０］，ｚ［１］を出力する。線形変換処理部３１１としてのＭＤＳ行列（ｍａｔｒｉｘ）は、ＧＦ（２）上の４次の既約多項式で定

義される拡大体ＧＦ（２^{4}）上の元を要素として持つ２×２の４要素からなる行列に基づく線形変換処理を実行する。

【０１１３】

行列演算の具体的処理を図１５（ｃ）に示す。図１５（ｃ）に示す２×２のＭＤＳ行列を適用した場合、入力ｗ［０］，ｗ［１］に基づいて、出力ｚ［０］，ｚ［１］は、以下のように算出される。

ｚ［０］＝ｗ［０］（ＸＯＲ）ｗ［１］×２

ｚ［１］＝ｗ［０］×２（ＸＯＲ）ｗ［１］

【０１１４】

最適拡散変換［ＯＤＭ］であるＭＤＳ行列（ｍａｔｒｉｘ）による線形変換結果として得られた８ビットデータ中の上位４ビットｚ［０］は４－ｂｉｔ入出力のＳ－ｂｏｘ［Ｓ４$_{2}$］３０３へ、下位４ビットｚ［１］は４－ｂｉｔ入出力のＳ－ｂｏｘ［Ｓ４$_{3}$］３０４へそれぞれ入力され、これらの４ビットＳ－ｂｏｘにおいて非線形変換が実行され、出力ｙ［０］，ｙ［１］を得る。Ｓ－ｂｏｘ［Ｓ４$_{2}$］３０３、Ｓ－ｂｏｘ［Ｓ４$_{3}$］３０４は、Ｓ－ｂｏｘ［Ｓ４$_{0}$］３０１、Ｓ－ｂｏｘ［Ｓ４$_{1}$］３０２と同様，それぞれランダムに選択した特性のよい４ビットＳ－ｂｏｘであり、図１４（ｂ）に示すテーブルに従ったデータ変換を実行する。最終的に、Ｓ－ｂｏｘ［Ｓ４$_{2}$］３０３、Ｓ－ｂｏｘ［Ｓ４$_{3}$］３０４の出力する４ビットデータｙ［０］とｙ［１］を結合した８ビットデータを、８ビット入出のＳ－ｂｏｘ３００の最終的な出力ｙ（８－ｂｉｔ）とする。

【０１１５】

図１６（ａ）に示す構成は、図１４（ａ）に示す構成と同様、図１３に示す構成を適用した具体的な８ビット入出力の非線形変換を実行する８ビットＳ－ｂｏｘ３２０の一構成例であり、図１６（ｂ）は、図１６（ａ）に示す８ビットＳ－ｂｏｘ３２０の構成要素である４つの４ビット入出力Ｓ－ｂｏｘ３２１～３２４の変換テーブルを示している。このテーブルで用いている数値は１６進数である。入力：ｘ＝０～Ｆ（１６進数）に対する各Ｓ－ｂｏｘ３２１～３２４の出力値を示すテーブルである。また、図１７（ｃ）には、図１７（ａ）（＝図１６（ａ））に示す線形変換処理部３３１として最適拡散変換［ＯＤＭ］であるＭＤＳ行列（ｍａｔｒｉｘ）３３１を用いた具体例を示している。

【０１１６】

図１６、図１７に示す構成例は、

４ビット入出力Ｓ－ｂｏｘ３２１～３２４が、図１６（ｂ）に示すテーブルに従った非線形変換を実行すること、

線形変換処理部３３１が、図１７（ｃ）に従った変換処理を行なうこと、

これらの処理が異なっている。

【０１１７】

線形変換処理部３３１では、ＧＦ（２）上で定義される４次の既約多項式：ｐ（ｘ）＝ｘ^{4}＋ｘ＋１によって作られる拡大体ＧＦ（２^{4}）上の行列演算による線形変換が実行され、ｚ［０］，ｚ［１］を出力する。線形変換処理部３１１としてのＭＤＳ行列（ｍａｔｒｉｘ）は、４次の既約多項式で定義される２の拡大体ＧＦ（２^{4}）上の元を要素として持つ２×２の４要素からなる行列に基づく線形変換処理を実行する。

【０１１８】

行列演算の具体的処理を図１７（ｃ）に示す。図１７（ｃ）に示す２×２のＭＤＳ行列を適用した場合、入力ｗ［０］，ｗ［１］に基づいて、出力ｚ［０］，ｚ［１］は、以下のように算出される。

ｚ［０］＝ｗ［０］（ＸＯＲ）ｗ［１］

ｚ［１］＝ｗ［０］（ＸＯＲ）ｗ［１］×２

【０１１９】

図１４～図１７に示す例は、８ビット入出力方の非線形変換処理を実行するＳボックスであり、いずれも線形変換処理部３１１，３３１において適用されるＭＤＳ行列の要素のサイズはｎ／２ビットであり、２×２の行列であるため、要素の数は４つとなり、全行列

を調べたとしてもその数は、$2^{n/2\times4}$種程度であるため、容易に全行列を調べることが可能であり、前述したように、この手法でＳ－ｂｏｘを生成すると、簡易な代数表現が可能なのは小さなＳ－ｂｏｘに限られると考えられるため、拡大体ＧＦ（2^n）上での要素はランダムのように見え、ＧＦ（2^n）上で単純な代数的表現を行うことは難しいと考えることができる。また、拡大体上でのべき乗演算のように均一的な拡散はしない。また、入力から出力を得るためのパスには小さなＳ－ｂｏｘが２つあるのみであり、行列を適切に選択すれば、簡易な演算と短いパスで構成することが可能なため、ｎ－ｂｉｔでランダムに生成されたＳ－ｂｏｘや、従来の小さなＳ－ｂｏｘを組み合わせて構成したＳ－ｂｏｘよりもより実装効率は高くなる。

【０１２０】

図１３～図１７に示した構成例では、入力ｎビットを２分割して上位ｎ／２ビット、下位ｎ／２ビットずつを、それぞれｎ／２ビット入出力のＳ－ｂｏｘに入力する構成であり、また、線形変換処理部からの出力ｎビットの上位ｎ／２ビット、下位ｎ／２ビットずつ、それぞれｎ／２ビット入出力のＳ－ｂｏｘ［S_2］に入力して非線形変換処理結果を得る構成であるが、入力ビットを分割して入力する小さなＳ－ｂｏｘの数は、２に限らず、この他の数でもよい。図１８に、入力ｎビットを４分割してｎ／４ビットずつの非線形変換処理を行なう小さなｎ／４ビット入出力Ｓ－ｂｏｘを利用したｎビットＳ－ｂｏｘ４００の構成例を示す。

【０１２１】

図１８に示すｎビットＳ－ｂｏｘ４００の構成は、８つのｎ／４ビット入出力の全単射Ｓ－ｂｏｘ４０１～４０８と、分岐数の高い拡散変換として最適拡散変換［ＯＤＭ］である拡大体ＧＦ（$2^{n/4}$）において定義される４×４ＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列からなる線形変換処理部４２１から構成されるｎビット入出力の非線形変換処理を実行する構成である。

【０１２２】

入力ｎビットは、ｎ／４ビットずつ分割されて、それぞれｎ／４ビット入出力の全単射Ｓ－ｂｏｘ［S_0］４０１，Ｓ－ｂｏｘ［S_1］４０２，Ｓ－ｂｏｘ［S_2］４０３，Ｓ－ｂｏｘ［S_3］４０４に入力され、それぞれのＳ－ｂｏｘにおいて非線形変換処理がなされた後、拡大体ＧＦ（$2^{n/4}$）において定義される４×４ＭＤＳ行列からなる線形変換処理部４２１に入力されＭＤＳ行列に基づく線形変換処理がなされた後、線形変換結果のｎビットのｎ／４ビットずつが、それぞれｎ／４ビット入出力のＳ－ｂｏｘ［S_0］４０５，Ｓ－ｂｏｘ［S_1］４０６，Ｓ－ｂｏｘ［S_2］４０７，Ｓ－ｂｏｘ［S_3］４０８に入力され、それぞれのＳ－ｂｏｘにおいて非線形変換処理がなされて出力結果としてのｎビットを得る構成となっている。

【０１２３】

図１８に示す線形変換処理部４２１において適用される拡散変換も最適拡散変換［ＯＤＭ］であるＭＤＳ行列であり、前段の小さな４つのＳ－ｂｏｘの出力が後段の小さなＳ－ｂｏｘの入力にバランスよく配置されることが保証されるため、全体として特性の良い大きなＳ－ｂｏｘが生成できる。

【０１２４】

上述した本発明に従った非線形変換処理を実行するＳボックスによって構成されるデータ変換装置の構成について要約する。本発明のデータ変換装置は、以下のような構成を有することが特徴である。すなわち、

入力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部（小さなＳボックス）からなる第１段非線形変換部と、

第１段非線形変換部を構成する複数の小型非線形変換部からの出力をすべて入力して線形変換を実行する線形変換部と、

線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部（小さなＳボックス）からなる第２段非線形変換部とを有し、

線形変換部は、入力データのビットサイズと同サイズの要素を持つ行列演算によるデー

タ変換を実行する構成であり、行列がｍ×ｍ行列である場合、少なくともｍ以上の分岐数を有する高分岐数の行列を適用したデータ変換処理を実行する構成である。

【０１２５】

具体的には、入力データがｎビットデータである場合、第１非線形変換部は、入力データであるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部から構成され、線形変換部は、ｋ個の小型非線形変換部の出力する総計ｎビットのデータを入力して、高分岐数の行列を適用したデータ変換処理によりｎビットの出力を生成する構成であり、第２非線形変換部は、線形変換部から出力されるｎビットデータの分割データであるｎ／ｋビットを各々入力して非線形変換処理結果としてのｎ／ｋビットを出力するｋ個の小型非線形変換部を有する構成である。

【０１２６】

ここで、線形変換部は、分岐数がｍ以上である拡散変換、例えば最適拡散変換ＯＤＭ（Ｏｐｔｉｍａｌ　Ｄｉｆｆｕｓｉｏｎ　Ｍａｐｐｉｎｇｓ）を実行するＭＤＳ（Ｍａｘｉｍｕｍ　Ｄｉｓｔａｎｃｅ　Ｓｅｐａｒａｂｌｅ）行列を適用したデータ変換処理を実行する構成であり、さらに、具体的には、入力データがｎビットデータである場合、ＧＦ（２）上で定義されるｎ次の既約多項式ｐ（ｘ）によって定義される拡大体ＧＦ（2^n）上の行列を適用したデータ変換処理を実行する構成である。

【０１２７】

なお、本発明に係るデータ変換装置は、先に説明したような共通鍵ブロック暗号処理を実行する装置として具現化可能であり、例えば、

（ア）ＳＰＮ（Substitution Permutation Network）構造、

（イ）Ｆｅｉｓｔｅｌ構造、

（ウ）拡張Ｆｅｉｓｔｅｌ構造、

これらの構造を適用した暗号処理を行なう装置における非線形変換部に適用可能である。なお、本発明の構成は、暗号処理以外にもハッシュ関数など、非線形変換を行う演算装置において適用可能である。

【０１２８】

［４．暗号処理装置の構成例］

最後に、上述した実施例に従ったデータ変換処理を実行する装置例として暗号処理装置としてのＩＣモジュール７００の構成例を図１９に示す。上述の処理は、例えばＰＣ、ＩＣカード、リーダライタ、その他、様々な情報処理装置において実行可能であり、図１９に示すＩＣモジュール７００は、これら様々な機器に構成することが可能である。

【０１２９】

図１９に示すＣＰＵ(Central processing Unit)７０１は、暗号処理の開始や、終了、データの送受信の制御、各構成部間のデータ転送制御、その他の各種プログラムを実行するプロセッサである。メモリ７０２は、ＣＰＵ７０１が実行するプログラム、あるいは演算パラメータなどの固定データを格納するＲＯＭ（Read-Only-Memory）、ＣＰＵ７０１の処理において実行されるプログラム、およびプログラム処理において適宜変化するパラメータの格納エリア、ワーク領域として使用されるＲＡＭ（Random Access Memory）等からなる。また、メモリ７０２は暗号処理に必要な鍵データや、暗号処理において適用する変換テーブル（置換表）や変換行列に適用するデータ等の格納領域として使用可能である。なおデータ格納領域は、耐タンパ構造を持つメモリとして構成されることが好ましい。

【０１３０】

暗号処理部７０３は、例えば上述した各種の暗号処理構成、すなわち、

（ア）ＳＰＮ（Substitution Permutation Network）構造、

（イ）Ｆｅｉｓｔｅｌ構造、

（ウ）拡張Ｆｅｉｓｔｅｌ構造、

これらの各構成のいずれかの構造を適用した共通鍵ブロック暗号処理アルゴリズムに従った暗号処理、復号処理、さらに、ハッシュ関数を適用した演算処理などを実行する。

【０１３１】
また、暗号処理部７０３は、上述した実施例に対応した非線形変換処理を実行する構成、すなわち、ｎビット入出力Ｓ－ｂｏｘの構成として、ｎより小さな入出力ビットの非線形変換を実行する小さなＳ－ｂｏｘによる非線形変換を実行し、その結果を、ＭＤＳ行列などの分岐数の高い拡散変換行列に入力して行列を適用した変換を実行し、その変換結果をさらに、ｎより小さな入出力ビットの非線形変換を実行する小さなＳ－ｂｏｘを適用して非線形変換を実行させてｎビットの非線形変換結果を得る非線形変換処理を実行する構成を持つ。
【０１３２】
なお、ここでは、暗号処理手段を個別モジュールとした例を示したが、このような独立した暗号処理モジュールを設けず、例えば暗号処理プログラムをＲＯＭに格納し、ＣＰＵ７０１がＲＯＭ格納プログラムを読み出して実行するように構成してもよい。
【０１３３】
乱数発生器７０４は、暗号処理に必要となる鍵の生成などにおいて必要となる乱数の発生処理を実行する。
【０１３４】
送受信部７０５は、外部とのデータ通信を実行するデータ通信処理部であり、例えばリーダライタ等、ＩＣモジュールとのデータ通信を実行し、ＩＣモジュール内で生成した暗号文の出力、あるいは外部のリーダライタ等の機器からのデータ入力などを実行する。
【０１３５】
このＩＣモジュール７００は、上述した実施例に従った非線形変換処理部としてのＳ－ｂｏｘを適用した処理、すなわち、図１３～図１８を参照して説明した構成を持つＳ－ｂｏｘを適用した非線形変換処理を実行する構成を持つ。
【０１３６】
以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、特許請求の範囲の欄を参酌すべきである。
【０１３７】
なお、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。
【０１３８】
例えば、プログラムは記録媒体としてのハードディスクやＲＯＭ（Read Only Memory）に予め記録しておくことができる。あるいは、プログラムはフレキシブルディスク、ＣＤ－ＲＯＭ(Compact Disc Read Only Memory)，ＭＯ(Magneto optical)ディスク，ＤＶＤ(Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体は、いわゆるパッケージソフトウエアとして提供することができる。
【０１３９】
なお、プログラムは、上述したようなリムーバブル記録媒体からコンピュータにインストールする他、ダウンロードサイトから、コンピュータに無線転送したり、ＬＡＮ(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。
【０１４０】

なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

【産業上の利用可能性】

【０１４１】

上述したように、本発明の一実施例の構成によれば、例えば、共通鍵ブロック暗号を適用した暗号処理などにおいて実行される非線形変換処理を、複数の小さなＳボックス（Ｓ－ｂｏｘ）による非線形変換を実行する第１段の非線形変換部と、第１段非線形変換部からの出力をすべて入力して線形変換を実行する線形変換部と、線形変換部の出力データを分割した分割データ各々の非線形変換処理を実行する複数の小型非線形変換部からなる第２段非線形変換部とを適用したデータ変換を行う構成とし、線形変換部では、最適拡散変換を行う行列を適用したデータ変換を実行する構成としたので、データの入力から出力に至るクリティカルパスを過大にすることなく、適切なデータ拡散を実現して、各種の暗号攻撃に対する耐性の強い安全性の高いデータ変換を実現することができる。

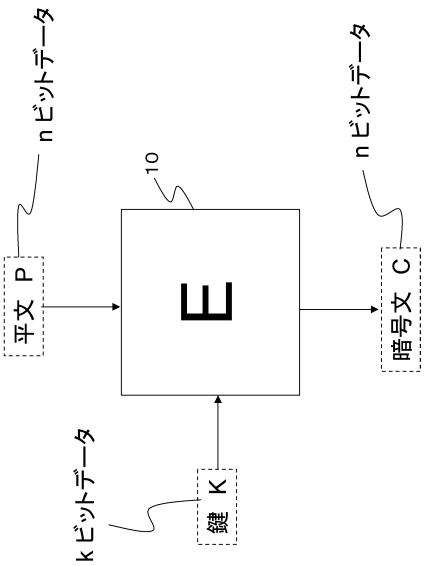
【符号の説明】

【０１４２】

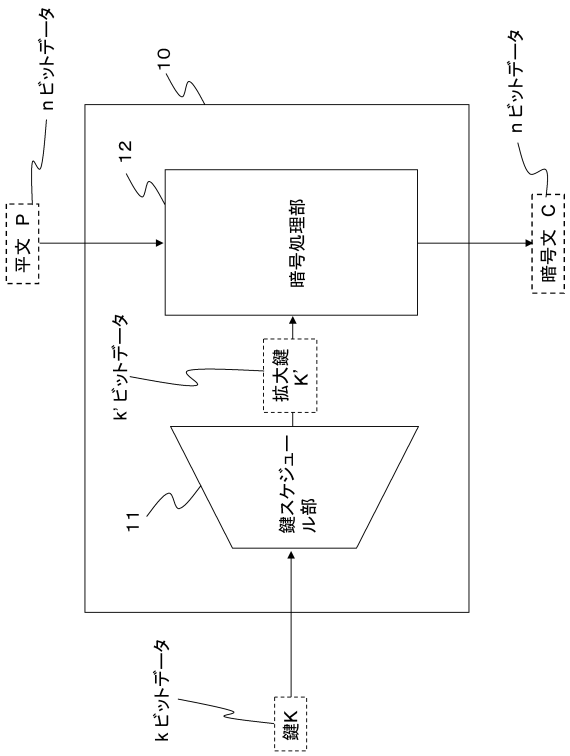
１０　共通鍵ブロック暗号処理部Ｅ
１１　鍵スケジューリング部
１２　暗号処理部
２０　ラウンド関数実行部
２１　排他的論理和演算部
２２　非線形変換処理部
２３　線形変換処理部
３０　Ｆ関数部
３１　排他的論理和演算部
３２　非線形変換処理部
３３　線形変換処理部
３４　排他的論理和演算部
４１，４２　Ｆ関数部
５０　非線形変換処理部
５１　Ｓボックス
１０１～１０３　Ｓボックス（Ｓ－ｂｏｘ）
１１１～１１３　排他的論理和演算部
１２１～１２４　Ｓボックス（Ｓ－ｂｏｘ）
１３１　ビット演算部
１４１～１４５　Ｓボックス（Ｓ－ｂｏｘ）
１５１～１５３　排他的論理和演算部
１６１～１６３　Ｓボックス（Ｓ－ｂｏｘ）
１７１，１７２　ＯＲ回路
２００　Ｓボックス（Ｓ－ｂｏｘ）
２０１～２０４　Ｓボックス（Ｓ－ｂｏｘ）
２１１　線形変換処理部
２００　Ｓボックス（Ｓ－ｂｏｘ）
３０１～３０４　Ｓボックス（Ｓ－ｂｏｘ）
３１１　線形変換処理部
３２０　Ｓボックス（Ｓ－ｂｏｘ）
３２１～３２４　Ｓボックス（Ｓ－ｂｏｘ）
３３１　線形変換処理部
７００　ＩＣモジュール

７０１　ＣＰＵ(Central processing Unit)
７０２　メモリ
７０３　暗号処理部
７０４　乱数発生器
７０５　送受信部

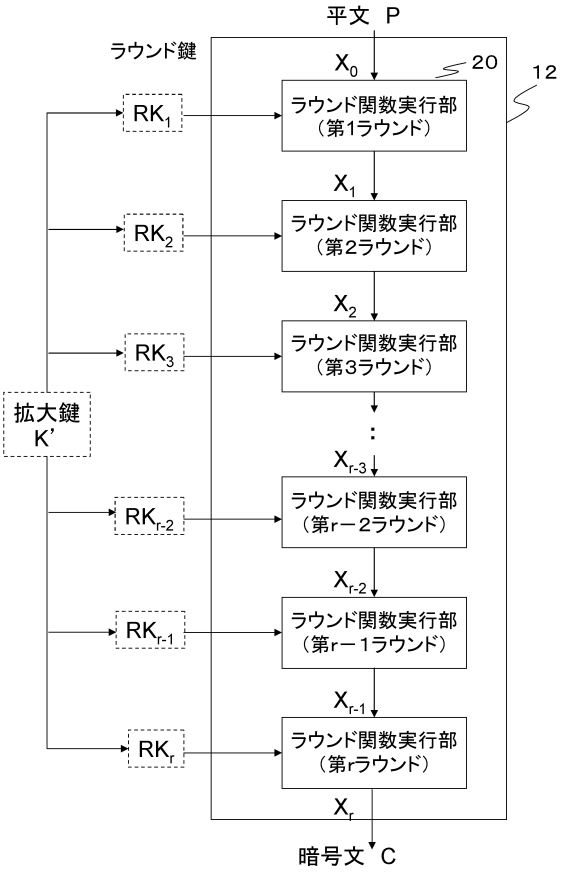
【図１】



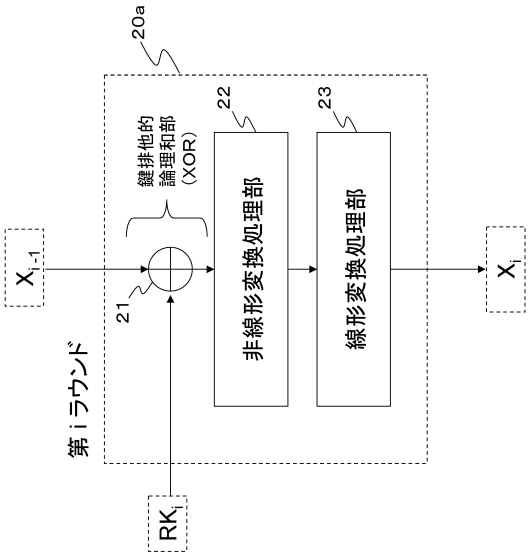
【図２】



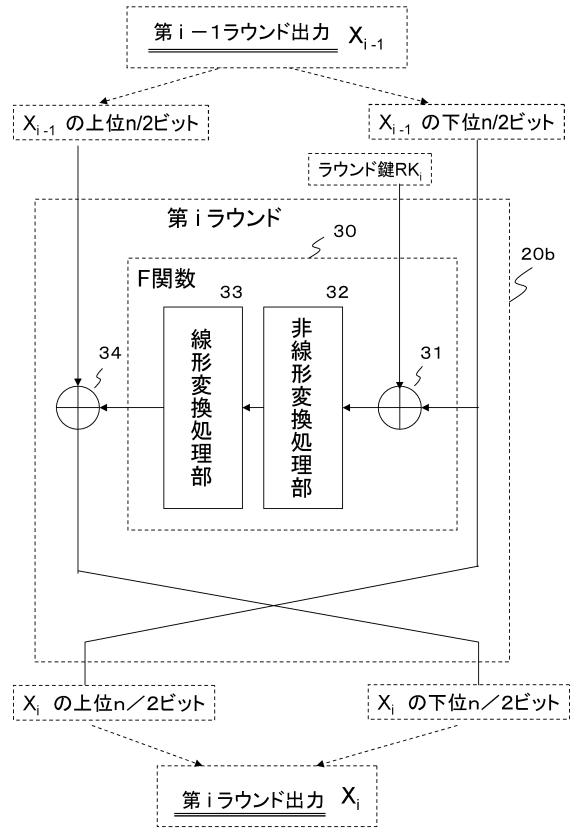
【図３】


平文 P
ラウンド鍵
X_0
20
12
RK_1
ラウンド関数実行部（第1ラウンド）
X_1
RK_2
ラウンド関数実行部（第2ラウンド）
X_2
RK_3
ラウンド関数実行部（第3ラウンド）
拡大鍵 K'
X_{r-3}
RK_{r-2}
ラウンド関数実行部（第r－2ラウンド）
X_{r-2}
RK_{r-1}
ラウンド関数実行部（第r－1ラウンド）
X_{r-1}
RK_r
ラウンド関数実行部（第rラウンド）
X_r
暗号文 C

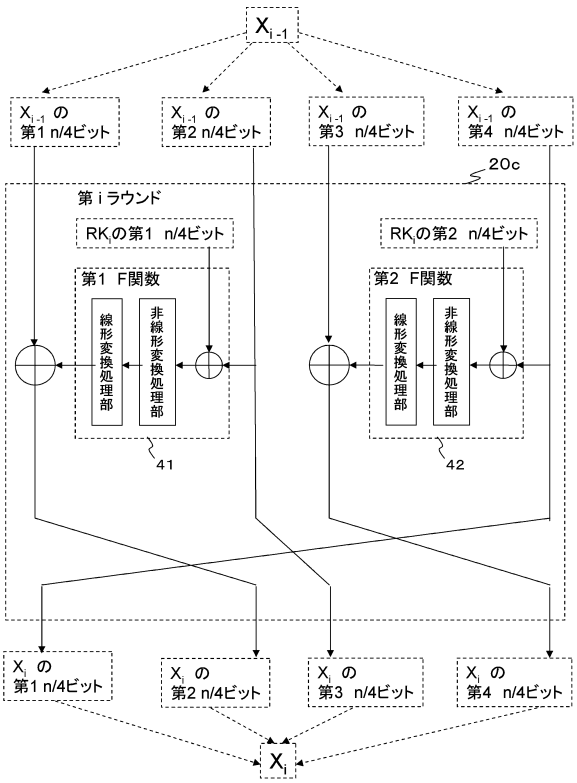
【図４】


20a
鍵排他的論理和部（XOR）
22
23
非線形変換処理部
線形変換処理部
X_{i-1}
21
X_i
第ｉラウンド
RK_i

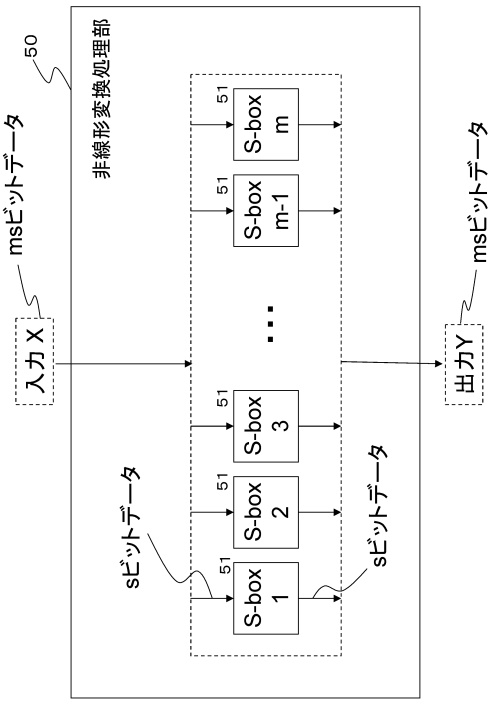
【図５】


第ｉ－1ラウンド出力 X_{i-1}
X_{i-1} の上位n/2ビット
X_{i-1} の下位n/2ビット
ラウンド鍵RK_i
第ｉラウンド
30
20b
F関数
33
32
線形変換処理部
非線形変換処理部
34
31
X_i の上位n／2ビット
X_i の下位n／2ビット
第ｉラウンド出力 X_i

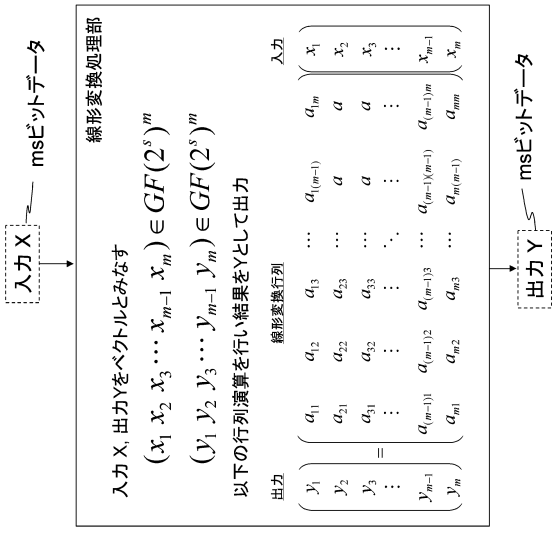
【図６】


X_{i-1}
X_{i-1} の第1 n/4ビット
X_{i-1} の第2 n/4ビット
X_{i-1} の第3 n/4ビット
X_{i-1} の第4 n/4ビット
20c
第ｉラウンド
RK_iの第1 n/4ビット
RK_iの第2 n/4ビット
第1 F関数
第2 F関数
線形変換処理部
非線形変換処理部
41
42
X_i の第1 n/4ビット
X_i の第2 n/4ビット
X_i の第3 n/4ビット
X_i の第4 n/4ビット
X_i

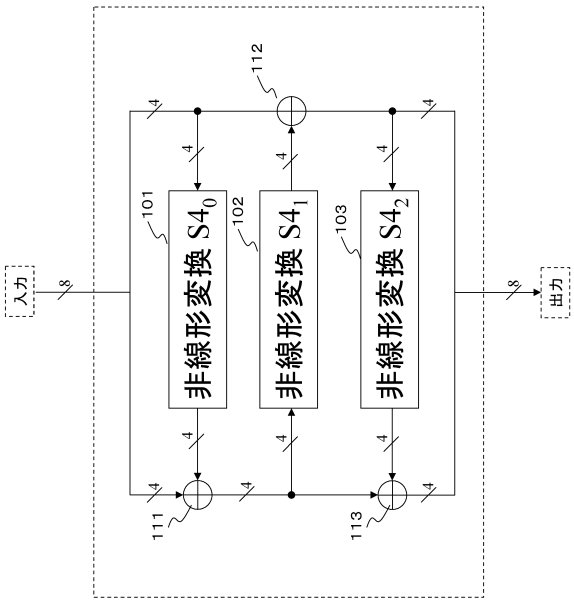
【図７】


50
非線形変換処理部
msビットデータ
入力X
出力Y
msビットデータ
sビットデータ
sビットデータ
S-box 1
S-box 2
S-box 3
S-box m-1
S-box m
51

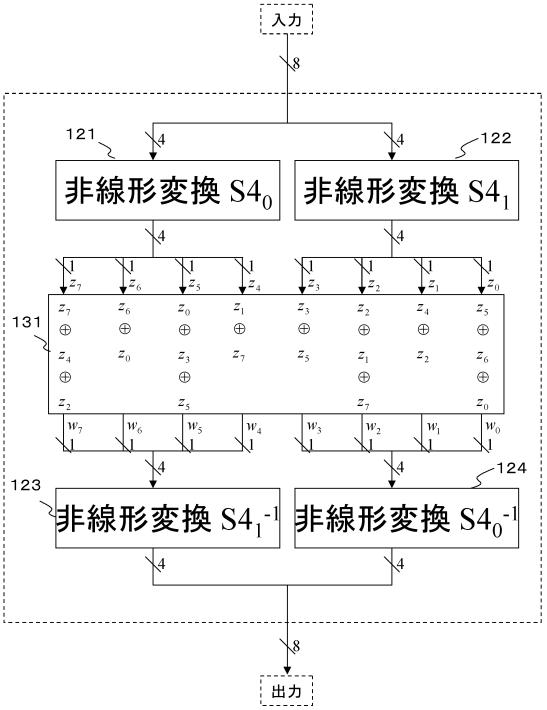
【図８】


msビットデータ
入力X
線形変換処理部
入力X, 出力Yをベクトルとみなす
$(x_1\ x_2\ x_3 \cdots x_{m-1}\ x_m) \in GF(2^s)^m$
$(y_1\ y_2\ y_3 \cdots y_{m-1}\ y_m) \in GF(2^s)^m$
以下の行列演算を行い結果をYとして出力
線形変換行列
出力
入力
出力Y
msビットデータ

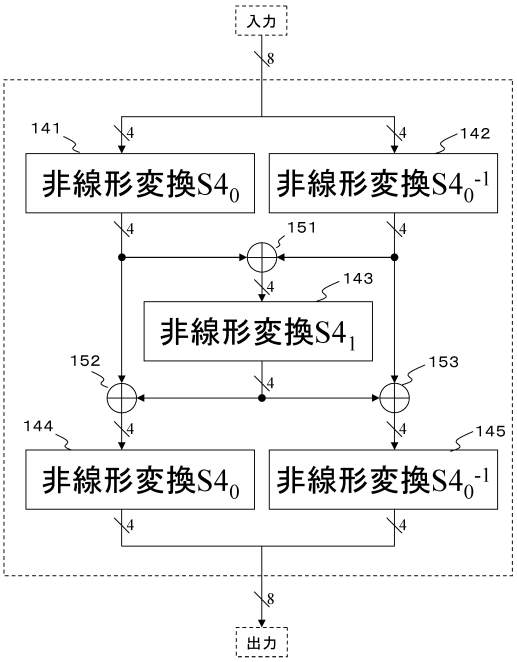
【図９】


入力
出力
非線形変換 $S4_0$
非線形変換 $S4_1$
非線形変換 $S4_2$
101
102
103
111
112
113
8
4

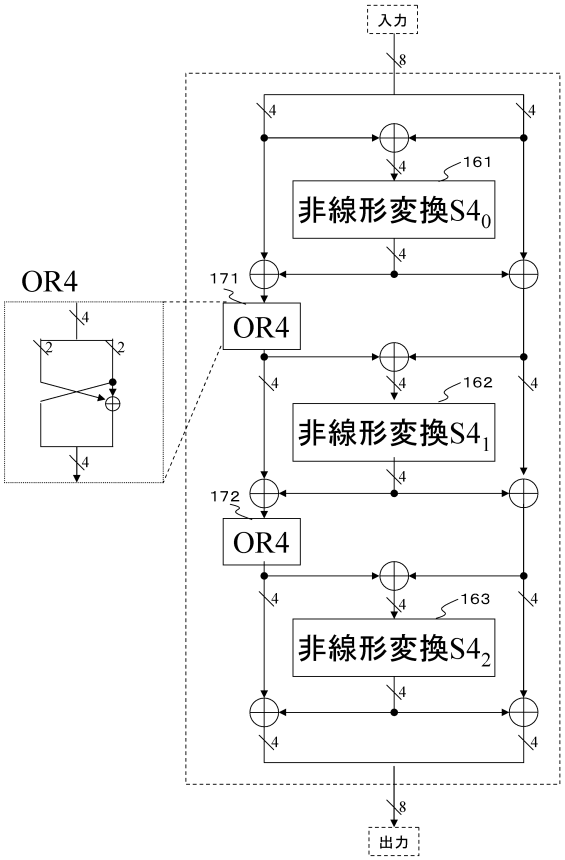
【図１０】


入力
出力
非線形変換 $S4_0$
非線形変換 $S4_1$
非線形変換 $S4_1^{-1}$
非線形変換 $S4_0^{-1}$
121
122
123
124
131
8
4
1

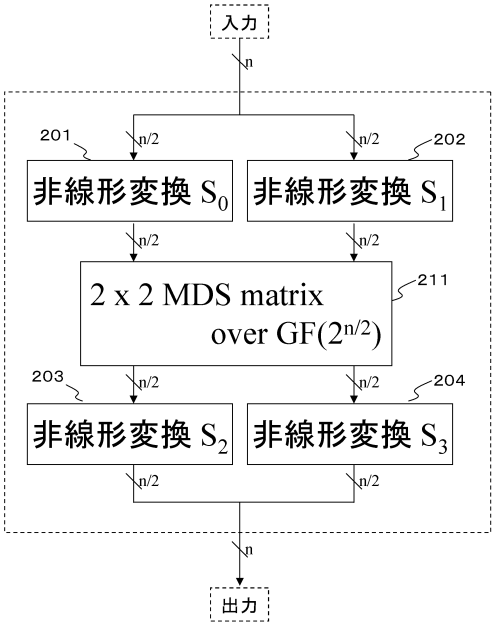
【図１１】



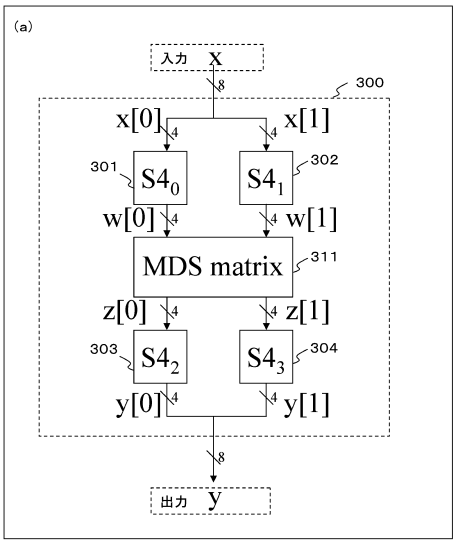
【図１２】



【図１３】



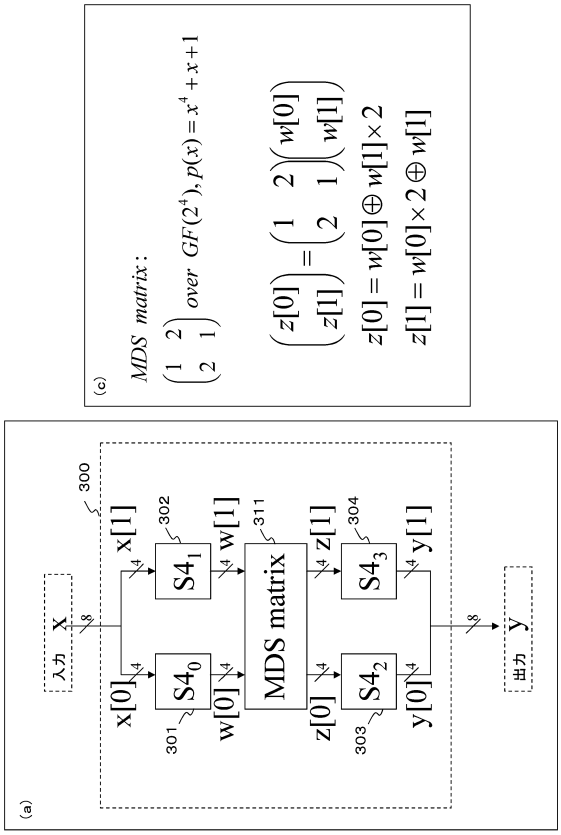
【図１４】



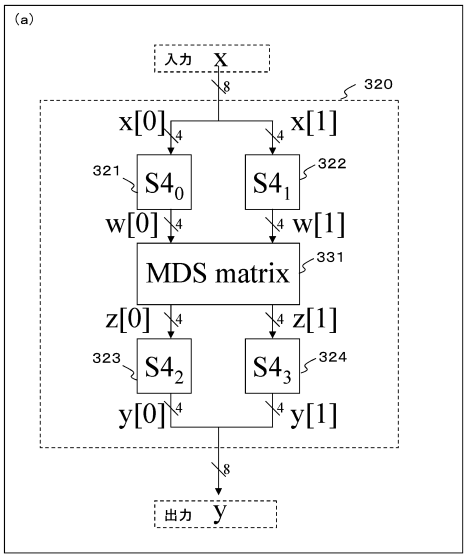
(b)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S4_0(x)$	E	6	C	A	8	7	2	F	B	1	4	0	5	9	D	3
$S4_1(x)$	6	4	0	D	2	B	A	3	9	C	E	F	8	7	5	1
$S4_2(x)$	B	8	5	E	A	6	4	C	F	7	2	3	1	0	D	9
$S4_3(x)$	A	2	6	D	3	4	5	E	0	7	8	9	B	F	C	1

【図１５】



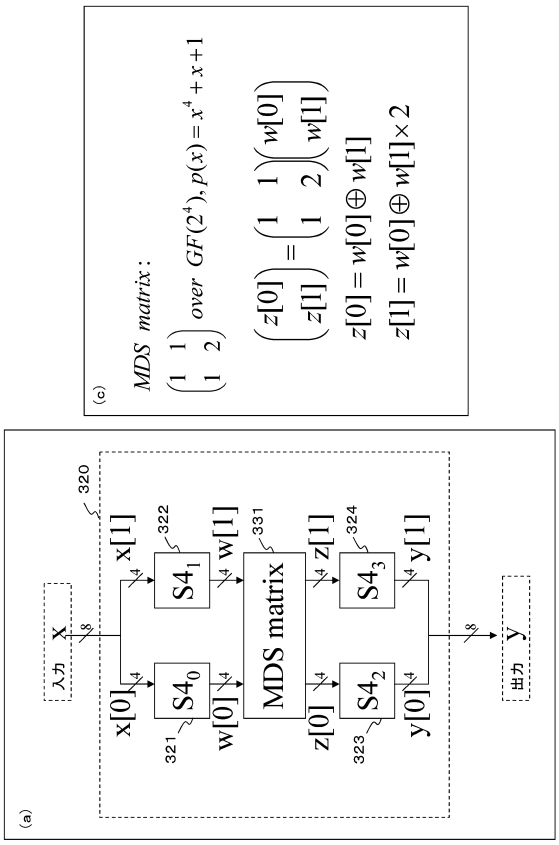
【図１６】



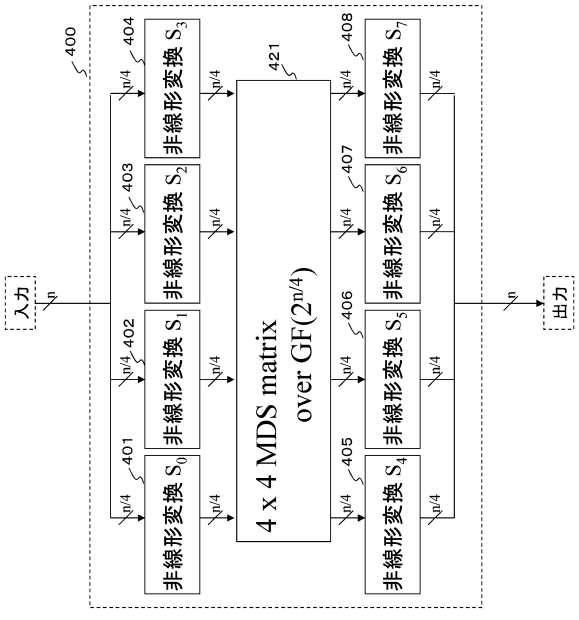
(b)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S4$_0$(x)	8	7	D	0	E	2	3	B	6	4	9	5	F	C	1	A
S4$_1$(x)	4	5	E	1	D	0	2	B	A	8	7	9	3	6	F	C
S4$_2$(x)	3	B	C	6	F	2	8	0	A	4	E	D	9	7	5	1
S4$_3$(x)	C	0	B	E	6	7	F	2	3	1	5	9	D	8	4	A

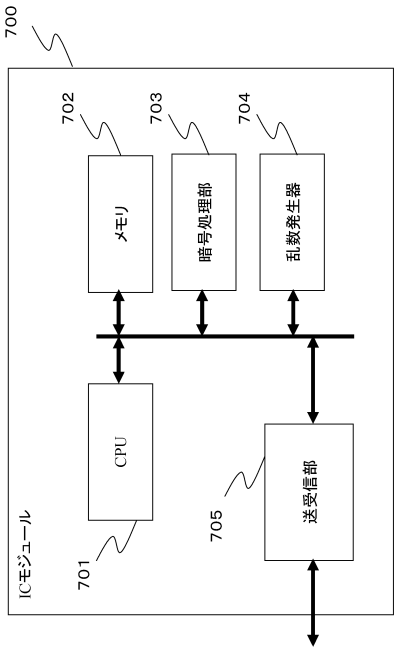
【図１７】



【図１８】



【図１９】



フロントページの続き

(72)発明者 渋谷 香士
東京都港区港南１丁目７番１号 ソニー株式会社内
(72)発明者 白井 太三
東京都港区港南１丁目７番１号 ソニー株式会社内
(72)発明者 秋下 徹
東京都港区港南１丁目７番１号 ソニー株式会社内
(72)発明者 盛合 志帆
東京都港区南青山二丁目６番２１号 株式会社ソニー・コンピュータエンタテインメント内

審査官 中里 裕正

(56)参考文献 特開２００６－７２０５４（ＪＰ，Ａ）
米国特許出願公開第２００４／０２４７１１７（ＵＳ，Ａ１）
特開２００１－３２４９２４（ＪＰ，Ａ）
国際公開第０１／０６７４２５（WO，Ａ１）
特開平５－９５３５０（ＪＰ，Ａ）
Zheng, Y., et al., On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses, Lecture Notes in Computer Science, １９９０年, Vol.435, p.461-480
暗号技術評価報告書 CRYPTREC Report 2000, 情報処理進行事業協会 セキュリティセンター, ２００１年 3月, p.15-17, ＵＲＬ, http://www.cryptrec.go.jp/report.html
Lim, C. H., A Revised Version of CRYPOTON, CRYPTON V1.0, Lecture Notes in Computer Science, １９９９年, Vol.1636, p.31-45
共通鍵ブロック暗号の選択／設計／評価に関するドキュメント, 通信・放送機構 情報セキュリティ技術に関する研究開, ２０００年 6月, p.103-105

(58)調査した分野(Int.Cl., ＤＢ名)
Ｈ０４Ｌ ９／０６
Ｇ０９Ｃ １／００
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）