(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2014 年 6 月 26 日(26.06.2014)

WIPO | PCT

(10) 国際公開番号
WO 2014/097414 A1

(51) 国際特許分類:
G09C 1/00 (2006.01)

(21) 国際出願番号: PCT/JP2012/082861

(22) 国際出願日: 2012 年 12 月 19 日(19.12.2012)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人: 株式会社日立製作所 **(HITACHI, LTD.)** [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目６番６号 Tokyo (JP).

(72) 発明者: 渡邉　大 **(WATANABE, Dai)**; 〒2440817 神奈川県横浜市戸塚区吉田町２９２番地　株式会社日立製作所　横浜研究所内 Kanagawa (JP). 井手口　恒太 **(IDEGUCHI, Kota)**; 〒1008220 東京都千代田区丸の内一丁目６番１号　株式会社日立製作所　金融システム営業統括本部内 Tokyo (JP).

(74) 代理人: 井上　学, 外 **(INOUE, Manabu** et al.**)**; 〒1008220 東京都千代田区丸の内一丁目６番１号　株式会社日立製作所内 Tokyo (JP).

(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

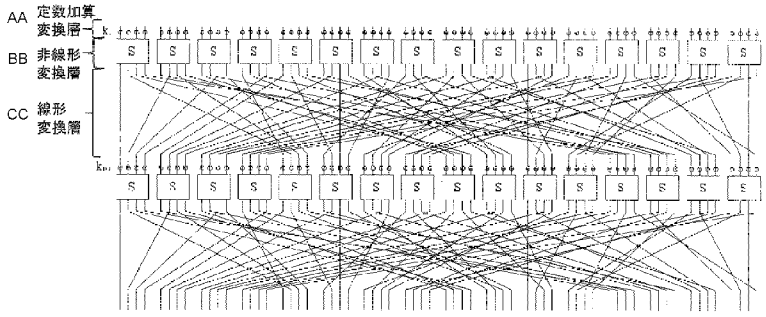
— 国際調査報告（条約第 21 条(3)）

WO 2014/097414 A1

(54) **Title**: SUBSTITUTION CONVERSION DEVICE AND SUBSTITUTION CONVERSION METHOD

(54) 発明の名称 : 置換変換装置及び置換変換方法

図2



AA Constant adding conversion layer
BB Non-linear conversion layer
CC Linear conversion layer

(57) **Abstract**: Provided is a high-security substitution conversion device that is compact when installed in hardware and is capable of allowing high-speed computation by software. In the substitution conversion device, input bit string data are converted by an initial conversion section (123), then, the converted data are converted repeatedly by a round conversion section (124), and the data finally converted by a final conversion section are output as a substitution value. The bit string data are represented with use of a plurality of indices. The round conversion section includes a non-linear conversion part and a linear conversion part. The non-linear conversion part performs conversion with respect to each data unit obtained by dividing the data by particular indices, and the linear conversion part performs conversion with respect to each data unit obtained by dividing the data by indices different from those of the non-linear conversion part.

(57) 要約:

[続葉有]

ハードウェアでの小規模実装とソフトウェアでの高速計算が可能で、かつ、安全性の高い置換変換装置を提供すること。　入力されたビット列のデータを初期変換部１２３で変換し、この変換されたデータをラウンド変換部１２４で繰り返し変換し、最後に最終変換部で変換したデータを、置換値として出力する。ビット列のデータは複数のインデックスを用いて表される。ラウンド変換部は、非線形変換と線形変換を含み、非線形変換は特定のインデックスにより分割されたデータ毎に変換を行い、線形変換は非線形変換とは異なるインデックスにより分割されたデータ毎に変換を行う。

明　細　書

発明の名称 ： 置換変換装置及び置換変換方法

技術分野

[0001] 本発明は、ビット列を置換変換する装置及び方法の技術に関する。

背景技術

[0002] 置換変換は、ある固定長のビット列を入力として、入力と同じ長さのビット列を生成する変換である。

[0003] 一般的に、ブロック暗号の鍵を固定した場合、その関数は置換変換となる。多くの場合、このようなブロック暗号は、入力された平文を繰り返しラウンド変換で変換することで、出力を計算する。ブロック暗号のうち、ハードウェア実装での必要なゲート数が少なくなるように設計された置換変換が存在する。特に、非特許文献１には、ビット列を入力にとり、ラウンド鍵のビット列への排他的論理和と、ビット列を４ビット毎に非線形変換を行う非線形層と、ビット列をビット置換により拡散する線形層とからなり、これらを繰り返すことで、置換変換を行い、ビット置換が結線だけで処理できるため、ハードウェアで小規模に実装できるブロック暗号ＰＲＥＳＥＮＴが開示されている。また、このブロック暗号は、差分攻撃に対して、安全性を有するものである。

先行技術文献

非特許文献

[0004] 非特許文献1：A. Bogdanov , L. R. Knudsen , G. Le , C. Paar , A. Poschmann , M. J. B. Robshaw , Y. Seurin , C. Vikkelsoe, ”PRESENT: An ultra-lightweight block cipher,” CHES 2007, LNCS4727, pp.450-466, Springer 2007.

発明の概要

発明が解決しようとする課題

[0005] しかし、非特許文献１に記載の技術ではブロック暗号ＰＲＥＳＥＮＴやそ

の置換変換は、ソフトウェアで実装した場合、計算速度が遅いという課題がある。

課題を解決するための手段

[0006] 以上の課題を解決するため、本発明は、複数のインデックスを用いて表されたビット列のデータに対して、特定のインデックスによりデータを分割し、分割されたデータ毎に非線形変換を行い、さらに、他のインデックスによりデータを分割し、分割されたデータ毎に線形変換を行い、これらを複数回繰り返すことで、置換値を生成する。

[0007] 例えば、2つの自然数mとdに対してm^d個のビットからなり、$n=m^{d-1}\times i_0+m^{d-2}\times i_1+...+m^0\times i_{d-1}$番目のビットがd個のインデックス$i_0, i_1, ..., i_{d-1}$を用いて$X_{i_0 i_1 ... i_{d-1}}$と表される入力データXに対して、$m^{d-1}$個のmビットのビット列$X_{i_0 i_1 ... i_{k-1} 0 i_{k+1} ... i_{d-1}} X_{i_0 i_1 ... i_{k-1} 1 i_{k+1} ... i_{d-1}} ... X_{i_0 i_1 ... i_{k-1} (m-1) i_{k+1} ... i_{d-1}}$ $(i_0, i_1, ..., i_{k-1}, i_{k+1}, ..., i_{d-1}=0,1,...,m-1)$のそれぞれを非線形置換Sボックスで非線形変換により変換し、次にインデックスi_kが同じ値を持つビットの間でビットの入れ替えを行う線形変換により変換し、非線形変換での変換と線形変換での変換をあらかじめ決められた回数繰り返すことで、前記入力データXの置換値を生成する。

発明の効果

[0008] 以上のように、本発明によれば、ハードウェアでの実装規模が小さく、差分攻撃に対する安全性を有し、かつ、ソフトウェア実装での計算高速性をもつ置換変換を提供することができる。

図面の簡単な説明

[0009] [図1]置換変換装置を示す概略図。

[図2]2ラウンド分のラウンド変換部を示す概略図。

[図3]一般的な計算機を示す概略図。

[図4]置換変換装置の処理を示すフローチャート図。

[図5]従来技術と本発明の比較図。

発明を実施するための形態

[0010] まず、本実施形態の説明に用いる記法について説明する。

本実施形態においては、置換変換は６４ビットの入力値を６４ビットの出力値に変換するようにしている。

[0011] 64ビット変数Wの各ビットは、左のビットから順に0ビット目、1ビット目と数えられ、一番右のビットが63ビット目と呼ばれ、各ビットは順にW{0}, W{1}, … W{63}と記載される。さらに、別の表記としてWのnビット目のビットW{n}はn=16×i+4×j+kなる3つの0以上3以下の整数i,j,kを用いて、W_{ijk} (=W{n})と記載される。また、W{i}からW{j}までの連続するj-i+1ビットをW{i-j}と記述する。また、kを固定しiとjを0から3まで走らせた16ビットの列をW_kと記載する。つまり、

$W_k = W_{00k}W_{01k}W_{02k}W_{03k}W_{10k}W_{11k}W_{12k}W_{13k}W_{20k}W_{21k}W_{22k}W_{23k}W_{30k}W_{31k}W_{32k}W_{33k}$

である。さらに、iとjを固定し、kを0から3まで走らせた4ビットの列をW_{ij}と記載する。

[0012] つまり、

$W_{ij}=W_{ij0}W_{ij1}W_{ij2}W_{ij3}$, である。

[0013] 図１は、本発明の第一の実施形態である置換変換装置１００の概略図である。

[0014] 図示するように、置換変換装置１００は、記憶部１１０と、制御部１２０と、入力部１３０と、出力部１４０と、を備える。

記憶部１１０は、初期値記憶領域１１１と、定数ラウンド値記憶領域１１２と、平文ラウンド値記憶領域１１３と、置換値記憶領域１１４と、を備える。

[0015] 初期値記憶領域１１１には、置換を計算する際の初期値を特定する情報が格納される。

[0016] 本実施形態においては、置換を計算する際の初期値として、定数ラウンド値の初期値が記憶される。

[0017] ここで、定数ラウンド値の初期値は、例えば、$c^{(0)}$＝0xffffffffffffffffと

いった64ビットの定数が記憶される。

[0018] なお、定数ラウンド値の初期値に使用される定数はこれらに限定されるわけではなく、例えば、疑似乱数生成器等で生成された乱数を使用することも可能である。

[0019] 定数ラウンド値記憶領域１１２には、各々のメッセージブロックにおけるラウンド毎の定数ラウンド値を特定する情報が格納される。

[0020] なお、本実施形態においては、定数ラウンド値は、後述するラウンド定数変換部１２２で生成される。

[0021] 平文ラウンド値記憶領域１１３には、ラウンド毎の平文ラウンド値を特定する情報が格納される。

[0022] なお、本実施形態においては、平文ラウンド値は、後述する全体制御部１２１と、初期変換部１２２と、ラウンド変換部１２３と、で生成される。

[0023] 置換値記憶領域１１４には、平文から計算される置換値が格納される。

[0024] 制御部１２０は、全体制御部１２１と、ラウンド定数変換部１２２と、初期変換部１２３と、ラウンド変換部１２４と、最終変換部１２５と、を備える。

[0025] 全体制御部１２１は、置換変換装置１００における処理の全体を制御する。

[0026] 特に、本実施形態においては、入力された平分を管理する処理や、ラウンド定数生成部１２２、初期変換部１２３、ラウンド変換部１２４、最終変換部１２５を管理する処理を行う。

[0027] ここで、本実施形態では、置換変換の入出力は６４ビットとしているがこのような態様に限定されるわけではない。

[0028] なお、本実施形態においては、平文をＸとし、ラウンド定数初期値を$C^{(0)}$とする。

[0029] ラウンド定数変換部１２２は、各ラウンドにおける定数ラウンド値及びラウンド定数を算出する。

[0030] 本実施形態においては、ラウンド定数変換部１２２は、最初のラウンドの

場合には初期値記憶領域１１１に記憶されている定数ラウンド値の初期値$C^{(0)}$から、または、最初のラウンドではない場合には定数ラウンド値記憶領域１１２に記憶されているラウンドにおける定数ラウンド値から、変換関数f_cを用いて、各ラウンドにおける定数ラウンド値を算出する。

[0031] 例えば、ラウンド定数変換部１２２は、前ラウンドｒの定数ラウンド値$c^{(r)}$（ｒ＝０の場合には定数ラウンド値の初期値$C^{(0)}$）を関数f_cに入力して算出された値を、今ラウンドｒ＋１の定数ラウンド値$c^{(r+1)}$とする。

[0032] 即ち、下記の（１）式に示されるようにして、前ラウンドの定数ラウンド値$c^{(r)}$（ｒ＝０の場合には定数ラウンド値の初期値$C^{(0)}$）から今ラウンドの定数ラウンド値$c^{(r+1)}$を算出する。

[0033] [数1]

【数1】

$$C^{(r)}=\pi^{r}(f_C(\pi^{r-1}(C^{(r-1)}))),$$
$$\pi(X)_{ijk}=X_{j(-i)k}$$

[0034] ここで、π^{r}はπをｒ回合成した関数である。π^{4}は恒等写像なることに注意。また、関数f_cは、線形フィードバックシフトレジスタ（ＬＦＳＲ）を用いる。

[0035] 一般に、ＬＦＳＲは多項式で決定されるが、ここではＬＦＳＲを決定する多項式ｇ（ｘ）を、下記の（３）式のように定義する。

[0036] [数2]

【数2】

$$g(x) = x^{64} + x^{61} + x^{58} + x^{55} + x^{47} + x^{46} + x^{42} + x^{41} + x^{39} + x^{38} + x^{37} + x^{35} + x^{34} + x^{33} + x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} + x^{25} + x^{24} + x^{20} + x^{19} + x^{18} + x^{16} + x^{14} + x^{12} + x^{8} + x^{7} + x^{2} + x^{1} + 1.$$

[0037] 但し、ｇ（ｘ）は有限体ＧＦ（２）において定義される多項式である。

[0038] なお、関数$f_c(x)$の擬似コードは、下記の（６）のように示される。

[0039] [数3]

【数3】

```
tmp=(x>>63)&0x1;
x=(x<<1);
if(tmp==1) {x^=0x2480c6eeff1d5187};
```

[0040] ここで、「＜＜Ｘ」は、Ｘビットの左シフトを示し、「＞＞Ｙ」は、Ｙビットの右シフトを示し、「＾」は、ビット毎の排他的論理和を示し、「＆」は、ビット毎の論理積を示す。

[0041] 初期変換部１２３は、始めのラウンド変換を行う前に一度だけ平文を変換する処理である。

[0042] 例えば、初期変換部１２３による変換は、６４ビットのビット列Ｘを入力としてとり、６４ビットのビット列Ｙを出力する線形変換$f_{initial}$である。線形変換$f_{initial}$は次の式（２）で記述される。

$Y = f_{initial}(X)$、

$Y_{ijk} = X_{kij}$ （ｉ，ｊ，ｋ＝０，１，２，３）

図１に戻り、ラウンド変換部１２４は、各ラウンドにおける平文ラウンド値を算出する処理を行う。

[0043] 例えば、ラウンド変換部１２４による平文ラウンド値の算出は、図２に示すラウンド関数f_{round}を用いて行われる。図２には2ラウンド分の処理が描かれている。

ラウンド関数f_{round}は、第ｒラウンドの平文ラウンド値$X^{(r)}$を第ｒ＋１ラウンドの平文ラウンド値$X^{(r+1)}$に変換する関数である。

[0044] 具体的にはラウンド関数f_{round}は定数加算変換層ＡＣと、非線形変換層ＳＬと、線形変換ＬＬ層と、からなる。まず、ラウンド変換部１２４は、平文ラウンド値記憶領域１１４に記憶されている第ｒラウンドの平文ラウンド値$X^{(r)}$とラウンド定数記憶領域に記憶されている（ｒ）ラウンドのラウンド定数$C^{(r)}$を定数加算変換層に入力し、値$Y^{(r)}$に変換する。定数加算変換層は、次の（４）式で表される。

[0045] [数4]

【数4】

$$Y^{(r)} = AC(X^{(r)}, C^{(r)})$$
$$Y^{(r)}_{ijk} = X^{(r)}_{ijk} \ XOR \ C^{(r)}_{kij}$$

[0046] 次に、ラウンド変換部１２４は、値$Y^{(r)}$を非線形変換層ＳＬに入力し値$Z^{(r)}$に変換する。非線形変換層ＳＬは、１６個の４ビットＳボックスを並列に並

べた変換であり、次の式５で表される。

[0047] [数5]

【数5】
$Z^{(r)}=SL(Y^{(r)})$、
$Z^{(r)}_{ij}=S(Y^{(r)}_{ij})$ (i, j=0, 1, 2, 3)

[0048] 非線形変換層で用いる４ビット入出力の置換テーブルＳボックスＳは、以下の（６）式と表により定義される。

[0049] [数6]

【数6】
y=S(x)、
S[16] ={8, 6, 7, 9, 3, 12, 10, 15, 13, 1, 14, 4, 0, 11, 5, 2};

[0050] ここで、S[16]はＳ（０）からＳ（１５）までの値を順に並べた配列である。

次に、ラウンド変換部１２４は、値$Z^{(r)}$を線形変換層ＬＬに入力しｒ＋１ラウンドの平文ラウンド値$X^{(r+1)}$を計算する。線形変換層ＬＬは、次の（７）式で表される。

[0051] [数7]

【数7】
$X^{(r+1)}=LL(Z^{(r)})$、
$X^{(r+1)}_{ij}k=Z^{(r)}_{j(k-i)k}$ (i, j, k=0, 1, 2, 3)

[0052] 次に、ラウンド変換部１２４は、計算したｒ＋１ラウンドの平文ラウンド値$X^{(r+1)}$をｒラウンドの平文ラウンド値に入れ替えて平文ラウンド値記憶領域１１３に記憶する。

また、ラウンド変換部を繰り返す回数である全ラウンド数はＲで表す。

[0053] 本実施形態においては、Ｒ＝３１とする。

図１に戻り、最終変換部１２５は、最終ラウンドで出力された平文ラウンド値$X^{(R)}$を入力としてとり、線形変換で変換し、置換値Ｙに変換する。

[0054] 例えば、最終変換部１２５は、線形変換として、f_{final}を用いる。

具体的には、f_{final}は次の式で表される。

[0055]

[数8]

【数8】
$Y=f_{final}(X^{(R)})$、
$Y_{ijk}=X^{(R)}{}_{k(-j)i}$、

[0056] 入力部１３０は、情報の入力を受け付ける。

出力部１４０は、情報を出力する。

[0057] 以上に記載した置換変換装置１００は、例えば、図３（計算機２００の概略図）に示すような、ＣＰＵ（Central Processing Unit）２０１と、メモリ２０２と、ＨＤＤ（Hard Disk Drive）等の外部記憶装置２０４、ＣＤ－ＲＯＭ（Compact Disk Read Only Memory）やＤＶＤ－ＲＯＭ（Digital Versatile Disk Read Only Memory）等の可搬性を有する記憶媒体２０５に対して情報を読み書きする読書装置２０６と、キーボードやマウスなどの入力装置２０７と、ディスプレイなどの出力装置２０８と、通信ネットワークに接続するためのＮＩＣ（Network Interface Card）等の通信装置２０９と、を備えた一般的な計算機２００で実現できる。

[0058] 例えば、記憶部１１０は、ＣＰＵ２０１がメモリ２０２又は外部記憶装置２０４を利用することにより実現可能であり、制御部１２０は、外部記憶装置２０４に記憶されている所定のプログラムをメモリ２０２にロードしてＣＰＵ２０１で実行することで実現可能であり、入力部１３０は、ＣＰＵ２０１が入力装置２０７を利用することで実現可能であり、出力部１４０は、ＣＰＵ２０１が出力装置２０８を利用することで実現可能である。

[0059] この所定のプログラムは、読書装置２０６を介して記憶媒体２０５から、あるいは、通信装置２０９を介してネットワークから、外部記憶装置２０４にダウンロードされ、それから、メモリ２０２上にロードされてＣＰＵ２０１により実行されるようにしてもよい。また、読書装置２０６を介して記憶媒体２０５から、あるいは、通信装置２０９を介してネットワークから、メモリ２０２上に直接ロードされ、ＣＰＵ２０１により実行されるようにしてもよい。

[0060] ＣＰＵ２０１は、３２ビットＣＰＵであり、複数のレジスタ２０３を備え

る。

[0061] ＣＰＵ２０１が実行可能な命令セットには、排他的論理和命令ＸＯＲ　Ａ，Ｂと、論理積命令ＡＮＤ　Ａ，Ｂと、論理和命令ＯＲ　Ａ，Ｂと、コピー命令ＣＯＰＹ　Ａ，Ｂと、が含まれる。

排他的論理和命令ＸＯＲ　Ａ，Ｂは、２つのレジスタＡ、Ｂに格納された値に対して、ビット毎の排他的論理和を行い、その結果をＡに上書きする命令である。

[0062] 論理積命令ＡＮＤ　Ａ，Ｂは、２つのレジスタＡ、Ｂに格納された値に対して、ビット毎の論理積を行い、その結果をＡに上書きする命令である。

[0063] 論理和命令ＯＲ　Ａ，Ｂは、２つのレジスタＡ、Ｂに格納された値に対して、ビット毎の論理和を行い、その結果をＡに上書きする命令である。

[0064] コピー命令ＣＯＰＹ　Ａ，Ｂは、Ｂの値をＡにコピーする命令である。

[0065] さらに、ＣＰＵ２０１が実行可能な命令セットには、ビット反転命令ＮＯＴ　Ａ、が含まれる。

[0066] ビット反転命令ＮＯＴ　Ａは、レジスタＡに格納された値に対して、各ビットのビット反転を行い、その結果をＡに上書きする命令である。

[0067] さらに、ＣＰＵ２０１が実行可能な命令セットには、即値との論理積命令ＡＮＤｉ　Ａ，ＩＭＭ、が含まれる。

[0068] 即値との論理積命令ＡＮＤｉ　Ａ，ＩＭＭは、１つのレジスタＡと１つの３２ビットの即値ＩＭＭに対して、ＡとＩＭＭのビット毎の論理積を行い、その結果をＡに上書きする命令である。

[0069] さらに、ＣＰＵ２０１が実行可能な命令セットには、右シフト命令ＳＨＲ　Ａ，ｉｍｍと、左シフト命令ＳＨＬ　Ａ，ｉｍｍと、左１６ビット巡回シフト命令ＲＯＬ　Ａ，ｉｍｍと、が含まれる。

[0070] 右シフト命令ＳＨＲ　Ａ，ｉｍｍは、１つのレジスタＡと１つの即値ｉｍｍに対して、Ａをｉｍｍビットだけ右にシフトする命令である。

[0071] 左シフト命令ＳＨＬ　Ａ，ｉｍｍは、１つのレジスタＡと１つの即値ｉｍｍに対して、Ａをｉｍｍだけ左にシフトする命令である。

[0072] 左１６ビット巡回シフト命令ＲＯＬ　Ａ，ｉｍｍは、１つのレジスタＡと１つの即値ｉｍｍに対して、Ａをｉｍｍだけ左に１６ビット単位で巡回シフトする命令である。

[0073] さらに、ＣＰＵ２０１が実行可能な命令セットには、メモリロード命令ＬＯＡＤ　Ａ，Ｍが含まれる。

[0074] メモリロード命令ＬＯＡＤ　Ａ，Ｍは、メモリ番地Ｍに格納されている３２ビットデータをレジスタＡにコピーする命令である。

[0075] 上述した命令のうち、ＸＯＲ，ＡＮＤ，ＯＲ，ＣＯＰＹ，ＮＯＴ，ＡＮＤｉ，ＸＯＲｉ，ＲＯＬ，ＲＯＬＣ，ＳＨＲ，ＳＨＬ，ＲＯＬ，ＬＯＡＤ命令はＣＰＵ２０１を用いて１サイクルで実行できるものとする。

図４は、置換変換装置１００での処理の流れを表すフローチャートである。

まず、置換変換装置１００では、入力部１３０を介して、平文Xの入力を受け付け、置換変換装置１００は全体制御部１２１が、初期変換部１２３に平文Xを入力する（ステップＳ１）。

[0076] そして、初期変換部１２３では、平文Xを変換して０ラウンド目の平文ラウンド値$X^{(0)}$を計算し、平文ラウンド記憶領域１１４に記憶する（ステップＳ２）。

[0077] 次に全体制御部１２１は、初期値記憶領域１１１に記憶されているラウンド定数初期値$C^{(0)}$をラウンド定数記憶領域１１３にコピーし、ラウンド数のカウンタであるラウンドカウンタｒの値に「０」を代入する（ステップＳ３）。

[0078] 次に全体制御部１２１は、平文ラウンド記憶領域１１３に記憶されている平文ラウンド値$X^{(r)}$とラウンド定数記憶領域１１２に記憶されている$C^{(r)}$とを、ラウンド変換部１２４に入力する（ステップＳ４）。

[0079] 次に、ラウンド変換部１２４では、平文ラウンド値$X^{(r)}$とラウンド定数値$C^{(r)}$から平文ラウンド値$X^{(r+1)}$を計算し、$X^{(r+1)}$を平文ラウンド記憶領域１１３に記憶する（ステップＳ５）。

[0080] 次に、全体制御部１２１は、ラウンドカウンタｒに１を加算してから（ス

テップＳ６）、ｒ＝Ｒとなっているか否か判断する（ステップＳ７）。

[0081] ステップＳ７でｒ＝Ｒとなっていない場合には（ステップＳ７でＮｏ）、ステップＳ８に進み、ｒ＝Ｒとなっている場合には（ステップＳ７でＹｅｓ）、ステップＳ１０に進む。

[0082] ステップＳ８では、全体制御部１２１は、ラウンド定数記憶領域１１２に記憶されているラウンド定数$C^{(r-1)}$を、ラウンド定数変換部１２２に入力する（ステップＳ８）。

[0083] 次に、ラウンド定数変換部１２２は、ｒ－１ラウンド目のラウンド定数$C^{(r-1)}$からｒラウンド目のラウンド定数$C^{(r)}$を計算し、$C^{(r)}$を定数ラウンド記憶領域に記憶し、全体制御部１２１は、ステップＳ４に戻る（ステップＳ９）。

[0084] ステップＳ１０では、全体制御部１２１は、平文ラウンド記憶領域１１３に記憶されている平文ラウンド値$X^{(R)}$を最終変換部１２５に入力する（ステップＳ１０）。

[0085] 次に、最終変換部１２５は、平文ラウンド値$X^{(R)}$から、置換値Ｙを計算し、置換値記憶領域１１４に記憶する（ステップＳ１１）。

[0086] そして、全体制御部１２１は、置換値記憶領域１１４に記憶されている置換値Ｙを、出力部１４０を介して出力する（ステップＳ１２）。

[0087] 本実施形態においては、図のステップＳ１からＳ１２は図３に示される一般的な計算機で実現できる。

[0088] まず、ステップＳ１においては、６４ビットの平文ＸはＣＰＵの４つのレジスタＲＥＧ０，ＲＥＧ１，ＲＥＧ２，ＲＥＧ３に以下のように保持される。

ＲＥＧ０＝Ｘ{０－１５}、

ＲＥＧ１＝Ｘ{１６－３１}、

ＲＥＧ２＝Ｘ{３２－４７}、

ＲＥＧ３＝Ｘ{４８－６３}.

次に、ステップＳ２の初期変換部１２３の処理は、計算機上での処理としては、一切行われない。初期変換部１２３は、ラウンド変換部１２４のＳボ

ックスのビットスライス実装を行うために行われるビット置換である。

[0089] 次に、ステップＳ３においては、メモリのラウンド定数を格納する領域にラウンド定数の初期値$C^{(0)}$をロードし、メモリのラウンドカウンタｒを保持する領域に０をロードする。

[0090] 次に、ステップＳ４では、ラウンド定数$C^{(r)}$をＲＥＧ４からＲＥＧ７のレジスタに次のように保持する。

ＲＥＧ４＝$C^{(r)}${０－１５}、

ＲＥＧ５＝$C^{(r)}${１６－３１}、

ＲＥＧ６＝$C^{(r)}${３２－４７}、

ＲＥＧ７＝$C^{(r)}${４８－６３}.

$X^{(r)}$に関しては、ステップ４に入ったときにすでにＲＥＧ０，ＲＥＧ１，ＲＥＧ２，ＲＥＧ３に格納されており、それをそのまま利用する。

[0091] ステップＳ５では、定数加算層変換、Ｓボックス層変換、線形層変換が以下のように順に行われる。

[0092] まず、定数加算層変換では、$X^{(r)}$と$C^{(r)}$を保持したレジスタに対して以下が行われる。

ＸＯＲ　ＲＥＧ０、ＲＥＧ４

ＸＯＲ　ＲＥＧ１、ＲＥＧ５

ＸＯＲ　ＲＥＧ２、ＲＥＧ６

ＸＯＲ　ＲＥＧ３、ＲＥＧ７

これにより、ＲＥＧ０からＲＥＧ３には式の$Y^{(r)}$がＲＥＧ０＝$Y^{(r)}_0$、ＲＥＧ１＝$Y^{(r)}1$、ＲＥＧ２＝$Y^{(r)}_2$、ＲＥＧ３＝$Y^{(r)}_3$と格納される。命令数は4命令である。

[0093] 次に、Ｓボックス層変換では、$Y^{(r)}$を格納するＲＥＧ０からＲＥＧ３に対して以下の命令が行われれ、$Z^{(r)}$が計算される。

ＭＯＶ ＲＥＧ４，　ＲＥＧ０

ＡＮＤ　ＲＥＧ０，ＲＥＧ２

ＸＯＲ　ＲＥＧ０，ＲＥＧ３

ＸＯＲ　ＲＥＧ２，ＲＥＧ１

ＸＯＲ　ＲＥＧ２，ＲＥＧ０

ＯＲ　ＲＥＧ３，ＲＥＧ４

ＸＯＲ　ＲＥＧ３，ＲＥＧ１

ＸＯＲ　ＲＥＧ４，ＲＥＧ２

ＭＯＶ　ＲＥＧ１，ＲＥＧ３

ＯＲ　ＲＥＧ３，ＲＥＧ４

ＸＯＲ　ＲＥＧ３，ＲＥＧ０

ＡＮＤ　ＲＥＧ０，ＲＥＧ１

ＸＯＲ　ＲＥＧ４，ＲＥＧ０

ＸＯＲ　ＲＥＧ１，ＲＥＧ３

ＸＯＲ　ＲＥＧ１，ＲＥＧ４

ＮＯＴ　ＲＥＧ４

ＭＯＶ　ＲＥＧ０、ＲＥＧ２

ＭＯＶ　ＲＥＧ２，ＲＥＧ１

ＭＯＶ　ＲＥＧ１、ＲＥＧ３

ＭＯＶ　ＲＥＧ３、ＲＥＧ４

これにより、ＲＥＧ０からＲＥＧ３には式の$Z^{(r)}$がＲＥＧ０＝$Z^{(r)}_0$、ＲＥＧ１＝$Z^{(r)}_1$、ＲＥＧ２＝$Z^{(r)}_2$、ＲＥＧ３＝$Z^{(r)}_3$と格納される。命令数は、16命令である。

[0094] そして、線形層変換ではｒの値によって、ＲＥＧ１からＲＥＧ３に対して以下の命令が行われ、$Z^{(r)}$から$X^{(r+1)}$が計算される。

[0095] ｒを４で割った余りが０の場合は、

ＲＯＴＬ　ＲＥＧ１、４

ＲＯＴＬ　ＲＥＧ２、８

ＲＯＴＬ　ＲＥＧ３、１２

とする。

[0096] ｒを４で割った余りが１の場合は、

```
MOV　REG4、REG1
SHL　REG1、1
SHR　REG4、3
ANDi　REG1、0xEEEE
ANDi　REG4、0x1111
OR　REG1、REG4
MOV　REG5、REG2
SHL　REG2、2
SHR　REG5、2
ANDi　REG2、0xCCCC
ANDi　REG5、0x3333
OR　REG2、REG5
MOV　REG6、REG3
SHL　REG3、3
SHR　REG6、1
ANDi　REG3、0x8888
ANDi　REG6、0x7777
OR　REG3、REG6
```

とする。

[0097] rを4で割った余りが2の場合は、

```
ROTL　REG1、12
ROTL　REG2、8
ROTL　REG3、4
```

とする。

[0098] rを4で割った余りが3の場合は、

```
MOV　REG4、REG1
SHL　REG1、3
SHR　REG4、1
```

ＡＮＤｉ　ＲＥＧ１、０ｘ８８８８

ＡＮＤｉ　ＲＥＧ４、０ｘ７７７７

ＯＲ　ＲＥＧ１、ＲＥＧ４

ＭＯＶ　ＲＥＧ５、ＲＥＧ２

ＳＨＬ　ＲＥＧ２、２

ＳＨＲ　ＲＥＧ５、２

ＡＮＤｉ　ＲＥＧ２、０ｘＣＣＣＣ

ＡＮＤｉ　ＲＥＧ５、０ｘ３３３３

ＯＲ　ＲＥＧ２、ＲＥＧ５

ＭＯＶ　ＲＥＧ６、ＲＥＧ３

ＳＨＬ　ＲＥＧ３、１

ＳＨＲ　ＲＥＧ６、３

ＡＮＤｉ　ＲＥＧ３、０ｘＥＥＥＥ

ＡＮＤｉ　ＲＥＧ６、０ｘ１１１１

ＯＲ　ＲＥＧ３、ＲＥＧ６

とする。これから分かるとおり、ｒを４で割ったあまりが０か２の場合は、３命令かかっており、１か３の場合は、１８命令かかっている。従って、平均すると１ラウンドあたり、(3+18+3+18)/4=10.5命令かかる。

[0099] 次に、ステップＳ６においては、ラウンドカウンタｒに１を加算し、ラウンドカウンタｒを保持する領域に保存し、ｒ＝Ｒか否かを判断する。

[0100] ステップＳ７では、定数ラウンド格納領域に格納されたラウンド定数$C^{(r)}$をレジスタＲＥＧ４からＲＥＧ７に以下のように格納し、ラウンド定数変換部１２５を呼び出す。

ＲＥＧ４＝$C^{(r)}${０－１５}、

ＲＥＧ５＝$C^{(r)}${１６－３１}、

ＲＥＧ５＝$C^{(r)}${３２－４７}、

ＲＥＧ６＝$C^{(r)}${４８－６４}.

ステップＳ８では、ラウンド定数変換部１２５の式に従い、ＲＥＧ４から

ＲＥＧ７に格納されたＣ$^{(r)}$に対して、（３）式の擬似コードに対応した命令が行われる。

[0101] ステップＳ９では、最終変換部１２４が呼び出される。ただし、ステップＳ９に入った時点ですでに、Ｘ$^{(R)}$はＲＥＧ０からＲＥＧ３に格納されており、それをそのまま用いるため、Ｘ$^{(R)}$を明示的に格納しなおす必要はない。

[0102] ステップＳ１０の最終変換部では、コンピュータによる処理を行う必要はない。これは、最終変換部の式により、ＲＥＧ０＝Ｙ{０－１５}、ＲＥＧ１＝Ｙ{１６－３１}、ＲＥＧ２＝Ｙ{３２－４７}、ＲＥＧ３＝Ｙ{４８－６３}、が成り立つためである。

ステップＳ１１では、レジスタＲ０からＲ３に格納されている置換出力値Ｙを、出力部を介して出力する。

[0103] 非特許文献１に記載の従来技術と比較するため、本発明による置換変換１００の命令数を見積もる。比較においては、置換変換の核となる部分の比較とするため、本発明の定数変換部１２２と従来技術のラウンド鍵更新部は除いて、命令数を比較する。

[0104] 本発明においては、前記のとおり、ラウンド定数変換部１２２を除いた1ラウンドあたりの命令数は、4+20+10.5=34.5命令である。全31ラウンドの場合、1062命令かかる。

[0105] 文献１に記載の従来技術では、1ラウンド分の鍵加算は、32ビットレジスタ2つに平文ラウンド値を格納する実装をする場合は、2回の排他的論理和、つまり2命令で計算できる。1ラウンド分の非線形変換層は、16個の4ビット値xi（i=0,…,15)に対して、$y_i=S(x_i)$をメモリ参照により行う。一つのiに対して、x_iのレジスタへの移動とx_iのオフセットによるSbox表のメモリ参照の2命令が必要となるため、1ラウンド分の非線形変換層は2×16=32回のＲＩＳＣ命令がかかると見積もれる。また、従来技術の線形変換層は、Yijk=Xjkiと記述されるが、これを1ビットずつの移動で計算する場合、1ビットあたり、ビットの読み出しで1回のAND命令、ビットの移動で1回のシフト命令、移動結果同士の結合で1回のOR命令がかかる。つまり、1ラウンド分の線形変換層は、3×64

=192命令かかると見積もることができる。従って、ラウンド鍵更新部を除いた1ラウンドあたりの計算量は、2+32+192=226命令と見積もられる。全31ラウンドの場合、7006命令かかる。

[0106] 従って、上記見積もりによると、本発明では文献１に記載の従来技術に比べて7006/1062=約6.6倍の高速化を達成できる。以上をまとめると図５の通りである。

[0107] 図５は、従来技術と本発明の命令数の比較図である。

図５に記載の通り、非線形変換層を、３２命令から２０命令に減らせたのは、テーブル参照ではなく、レジスタ値同士の論理演算を組み合わせて非線形変換層を実現することによる。線形変換層を、１９２命令から１０．５命令に減らせたのが、レジスタの入れ替えとレジスタ値の順かシフトでこの処理を実現することによる。上記2点を同時に実現可能となるように、線形変換のビット位置入れ替え方を定めたことにより演算量を削減できた。

[0108] また、本実施形態の置換変換１００は差分攻撃に対して文献１記載の従来技術と同等の安全性を有する。つまり、文献１において従来技術の置換変換の差分特性確率は25ラウンドで$1/2^{100}$以下であることが示されているが、本実施例の置換変換１００の差分特性確率も25ラウンドで$1/2^{100}$以下であることが示せる。本実施形態のラウンド変換部１２４は、定数加算層と非線形変換層、線形変換層からなるが、差分特性確率の上界を証明する際には非線形変換層と線形変換層に着目する。また、4ビットＳボックスに関しても、差分特性確率の上界の証明に関わる性質は、Ｓボックスの差分確率は1/4以下であることと、1ビット差分から1ビット差分への伝播は存在しないこと、の2つである。本実施形態の4ビットＳボックスは、この２つの性質を満たしている。また、線形変換層は、（７）式で表されているが、これは次のように3つの変換の合成としても記述できる。

[0109]

[数9]

【数9】
$X=LL(Z)=\sigma(LL_0(\rho(Z)))$、
$Z'=\rho(Z)$、
$Z'_{ijk}=Z_{ij(k+j)}$、
$Z''=LL_0(Z')$、
$Z''_{ijk}=Z'_{jki}$、
$Z'''=\sigma(Z'')$、
$Z'''_{ijk}=Z''_{ij(k-i)}$.

[0110] ここで、LL_0は文献１の線形変換層と同一のものである。また、ρとσは一番右のインデックスをずらす変換である。つまり、σはＳボックスへのビットの入力順、ρはＳボックスからのビットの出力順をずらすものである。従って、文献1の置換変換と本実施例の置換変換１００とでは、Ｓボックス入出力でのビット順を考えなければ、結線構造は同じものとなる。このことから、差分特性確率の上界の証明は文献1と同等のものが本実施例の置換変換１００にも適用できることになる。つまり、本実施例の置換変換１００の25ラウンドの差分特性確率は$1/2^{100}$で上から抑えられる。従って、この意味において差分攻撃に関して安全性を有するといえる。

[0111] 本実施形態では、線形変換層は（７）式で定義したが、これに限られるものではない。例えば、（７）式の代わりに、次のようにすることもできる。

[0112] [数10]

【数10】
$X^{(r+1)}=LL(Z^{(r)})$,
$X^{(r+1)}_{ijk}=Z^{(r)}_{f(i,j,k),g(i,j,k),k}$,

[0113] ここで、f(i,j,k)，g(i,j,k)は値域が{0,1,2,3}の関数である。ただし、kを固定し、i,jそれぞれを{0,1,2,3}の全ての値を走らせた時、(f(i,j,k),g(i,j,k))は、(a,b) a=0,1,2,3, b=0,1,2,3 の16通りの各値を1回ずつとるものとする。

[0114] また、本実施形態では、非線形変換層でのＳボックスを（６）式としたが、これ以外のＳボックスを用いることもできる。

[0115] 本実施形態では、64ビット入出力の置換変数を構成したが、本発明は64ビットに限定されるものではない。本発明によれば、例えば、任意の自然数mと

dに対して、m^dビット入出力の置換変換を構成できる。ここで、mは用いるSボックスの入出力ビット長、dはインデックスの個数である。m^dビットの平文Wのnビット目をW{n}とする。一番左のビットは0ビット目と数え、一番右のビットはm^{d-1}ビット目と数える。また、$n=i_0\times m^{d-1}+i_1\times m^{d-2}+\cdots+i_{d-1}\times m^0$を満たす、d個の0以上m-1以下の整数を用いて、$W\{n\}=W_\{i_0 i_1\cdots i_{d-1}\}$とも表す。ここでは、二段のインデックスとなるため、表記上の問題から1段目のインデックスはW_{インデックス}で表されている。この場合も、置換変換は前記実施形態と同様に、初期変換、Ｒ回のラウンド変換、最終変換をこの順番で計算するもので構成される。ただし、ここでの各変換は、例えば以下のようなものがある。以下ではm^dビットの平文をX、m^dビットの置換値をYとする。

[0116] 初期変換$f_{initial2}$は、

[0117] [数11]

【数11】

$$X^{(0)}=f_{initial2}(X),$$
$$X^{(0)}_\{i_0\ldots i_{d-1}\}=X_\{i_{d-1}i_0\ldots i_{d-2}\},$$

[0118] で表される。

ラウンド変換f_{round2}は、定数加算変換層AC_2と非線形変換層SL_2、線形変換層LL_2の合成である。

[0119] f_{round2}は、ｒラウンド目のラウンド定数をC(r)とすると、

[0120] [数12]

【数12】

$$X^{(r+1)}=f_{round2}(X^{(r)},C^{(r)})=LL_2(SL_2(AC_2(X^{(r)},C^{(r)})))$$

[0121] と定義される。

[0122] 定数加算層AC_2は、

[0123] [数13]

【数13】

$$Y^{(r)}=AC_2(X^{(r)},C^{(r)}),$$
$$Y^{(r)}_\{i_0 i_1\ldots i_{d-1}\}=X^{(r)}_\{i_0 i_1\ldots i_{d-1}\}\ \mathrm{XOR}\ C^{(r)}_\{i_{d-1}i_0\ldots id\text{-}2\},$$

[0124] と定義される。

[0125] 非線形変換層SL_2は、

[0126] [数14]

【数14】
$Z^{(r)}=SL_2(Y^{(r)})$,
$Z^{(r)}_\{i_0i_1...i_{d-2}\}=S(Y^{(r)}_\{i_0i_1...i_{d-2}\})$,

[0127] と定義される。ただし、$Y^{(r)}_\{i_0i_1...i_{d-2}\}$は、

[0128] [数15]

【数15】
$Y^{(r)}_\{i_0i_1...i_{d-2}\}=Y^{(r)}_\{i_0i_1...i_{d-2}0\}Y^{(r)}_\{i_0i_1...i_{d-2}1\}...Y^{(r)}_\{i_0i_1...i_{d-2}(m-1)\}$,

[0129] で定義されるmビット列である。$Z^{(r)}_\{i_0i_1...i_{d-2}\}$も同様に定義される。

[0130] 線形変換層LL_2は、

[0131] [数16]

【数16】
$X^{(r+1)}=LL_2(Z^{(r)})$,
$X^{(r+1)}_\{i_0i_1...i_{d-1}\}=Z^{(r)}_\{i_1i_2...i_{d-2}(i_{d-1}-i_0)i_{d-1}\}$,

[0132] と定義される。

[0133] このように定義されたラウンド変換部は、前記64ビット入出力の置換変換と類似の方法によりソフトウェアで実装でき、ソフトウェア実装での計算高速性を有する。

また、LL_2はより一般的に次の式とすることもできる。

[0134] $F_0(i_0,i_1,...,i_{d-1})$, $f_1(i_0,i_1,...,i_{d-1})$, $f_{d-2}(i_0,i_1,...,i_{d-1})$は、$i_{d-1}$を固定して$i_0,i_1,...,i_{d-2}$をそれぞれ0からm-1まで走らせたとき、$(f_0(i_0,...,i_{d-1}),...,f_{d-2}(i_0,...,i_{d-1}))$は、$(a_0,...,a_{d-2})$ $(a_0,...,a_{d-2}=0,1,...,m-1)$の計$m^{d-1}$通りの組み合わせを各1回ずつとるものとする。

[0135] 本実施形態では、置換変換を構成しているが、前記ラウンド定数の初期値の代わりに鍵を与えることで、ブロック暗号を構成することもできる。この際、前記ラウンド定数変更部は（１）式に限られるものではなく、鍵のビット長は置換変換のビット長と同一のものに限られるものではない。例えば、文献１の従来技術のように、64ビットの置換変換に対して、80ビットの鍵を用いるブロック暗号とすることもできる。この場合、例えば前記ラウンド定数変換部の代わりに、文献１に記載の鍵更新部を用いることもできる。

符号の説明

[0136] １００　置換変換装置

１１０　記憶部

１１１　初期値記憶領域

１１２　ラウンド定数記憶領域

１１３　平文ラウンド値記憶領域

１１４　置換値記憶領域

１２０　制御部

１２１　全体制御部

１２２　ラウンド定数変換部

１２３　初期変換部

１２４　ラウンド変換部

１２５　最終変換部

１３０　入力部

１４０　出力部

２００　計算機

２０１　ＣＰＵ

２０２　メモリ

２０３　レジスタ

２０４　外部記憶装置

２０５　記憶媒体

２０６　読書装置

２０７　入力装置

２０８　出力装置

２０９　通信装置

請求の範囲

[請求項1] 入力データXを受信する入力部と、

複数のビット列Xのそれぞれを非線形置換Sボックスを記憶する記憶部と、

前記入力データXを前記非線形置換Sボックスで非線形変換により変換し、インデックスi_kが同じ値を持つビットの間でビットの入れ替えを行う線形変換により変換し、

前記非線形変換での変換と前記線形変換での変換をあらかじめ決められた回数繰り返すことで、前記入力データXの置換値を生成するラウンド変換部とを備える

ことを特徴とする置換変換装置。

[請求項2] 請求項１に記載の置換変換装置であって、

前記入力データＸとは2つの自然数mとdに対してm^d個のビットからなり、$n=m^{d-1}\times i_0+m^{d-2}\times i_1+...+m^0\times i_{d-1}$番目のビットがd個のインデックス$i_0,i_1,...,i_{d-1}$を用いて$X_{i_0i_1...i_{d-1}}$と表され、

前記ビット列とは、m^{d-1}個のmビットのビット列$X_{i_0i_1...i_{k-1}0i_{k+1}...i_{d-1}}X_{i_0i_1...i_{k-1}1i_{k+1}...i_{d-1}}...X_{i_0i_1...i_{k-1}(m-1)i_{k+1}...i_{d-1}}$ $(i_0,i_1,...,i_{k-1},i_{k+1},...,i_{d-1}=0,1,...,m-1)$であることを特徴とする置換変換装置。

[請求項3] 請求項２に記載の置換変換装置であって、

前記ラウンド変換部は、前記非線形変換での変換の前に、前記入力データXをあらかじめ決められた定数回、排他的論理和することで変換し、

定数の排他的論理和による変換と、前記線形変換による変換と、前記線形変換による変換と、を繰り返しの単位とし、

排他的論理和される定数は繰り返し毎に異なるあらかじめ決められた定数である、

ことを特徴とする置換変換装置。

[請求項4] 請求項３に記載の置換変換装置であって、

前記ラウンド変換部は、前記線形変換は前記入力データXを$Y_{i_0 i_1 \ldots i_{d-2} i_{d-1}}=X_{i_1 i_2 \ldots (i_{d-1}-i_0) i_{d-1}}$に従い変換し、出力を出力データYとするものであること、

を特徴とする置換変換装置。

[請求項5] 請求項３に記載の置換変換装置であって、

前記ラウンド変換部は、前記自然数mは4、dは3であって、前記線形変換は前記入力データXを$Y_{i_0 i_1 i_2}=X_{i_1 (i_2-i_0) i_2}$に従い変換し、出力を出力データYとするものであること、

を特徴とする置換変換装置。

[請求項6] 請求項５に記載の置換変換装置であって、

前記非線形変換Ｓボックスは、置換表Sbox[16]={ 8, 6, 7, 9, 3, 12, 10, 15, 13, 1, 14, 4, 0, 11, 5, 2}であること、

を特徴とする置換変換装置。

[請求項7] 請求項６に記載の置換変換装置であって、

処理を始める前に前記データXをYに$Y_{i_0 i_1 i_2}=X_{i_2 i_0 i_1}$と変換してから、Yに対して前記定数加算変換と、前記非線形変換と、前記線形変換と、を繰り返し行い中間値Zを計算し、中間値Zを$W_{i_0 i_1 i_2}=Z_{i_2 (-i_1) i_0}$に従い変換し、値Wを置換値として出力すること、

を特徴とする置換変換装置。

[請求項8] 請求項７に記載の置換変換装置であって、

前記値Xを、$REG_i=X_{00i}X_{01i}X_{02i}X_{03i}X_{10i}X_{11i}X_{12i}X_{13i}X_{20i}X_{21i}X_{22i}X_{23i}X_{30i}X_{31i}X_{32i}X_{33i}$ i=0,1,2,3と4つに分割しておき、前記非線形変換をREG_i間のビット毎論理演算で計算し、前記線形変換をREG_i毎のビットシフトとビット巡回シフトとビット毎論理演算で計算すること、

を特徴とする置換変換装置。

[請求項9] 置換変換装置が、

2つの自然数mとdに対してm^d個のビットからなり、$n=m^{d-1}\times i_0+m^{d-2}\times i_1+.$

..+m^0×i_{d-1}番目のビットがd個のインデックス$i_0, i_1, ..., i_{d-1}$を用いてX_{$i_0 i_1 ... i_{d-1}$}と表される入力データXに対して、m^{d-1}個のmビットのビット列X_{$i_0 i_1 ... i_{k-1} 0 i_{k+1} ... i_{d-1}$}X_{$i_0 i_1 ... i_{k-1} 1 i_{k+1} ... i_{d-1}$}...X_{$i_0 i_1 ... i_{k-1} (m-1) i_{k+1} ... i_{d-1}$} ($i_0, i_1, ..., i_{k-1}, i_{k+1}, ..., i_{d-1}$=0,1,...,m-1)のそれぞれを非線形置換Sボックスで非線形変換により変換し、次にインデックスi_kが同じ値を持つビットの間でビットの入れ替えを行う線形変換により変換し、

前記非線形変換での変換と前記線形変換での変換をあらかじめ決められた回数繰り返すことで、前記入力データXの置換値を生成する置換生成方法。

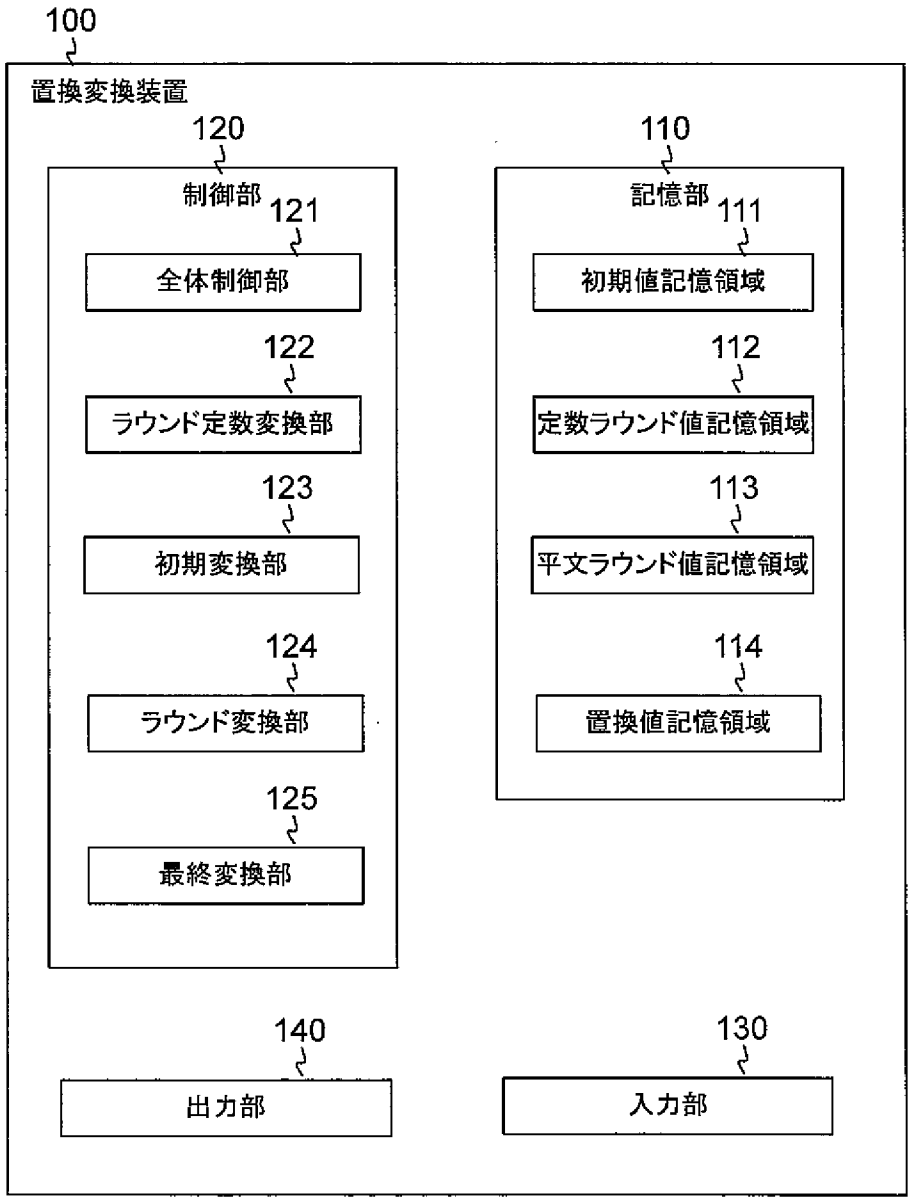
［請求項10］ 請求項２に記載の置換変換装置であって、

排他的論理和される前記定数の代わりに鍵データを排他的論理輪する、

ことを特徴とする置換変換装置。

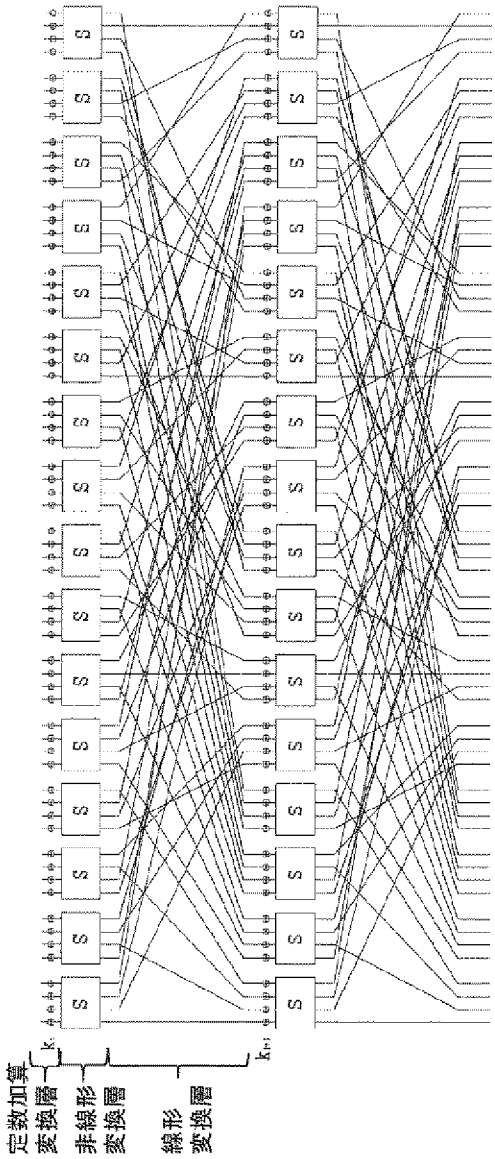
[図1]

図1


100
置換変換装置
120
制御部
121
全体制御部
122
ラウンド定数変換部
123
初期変換部
124
ラウンド変換部
125
最終変換部
110
記憶部
111
初期値記憶領域
112
定数ラウンド値記憶領域
113
平文ラウンド値記憶領域
114
置換値記憶領域
140
出力部
130
入力部

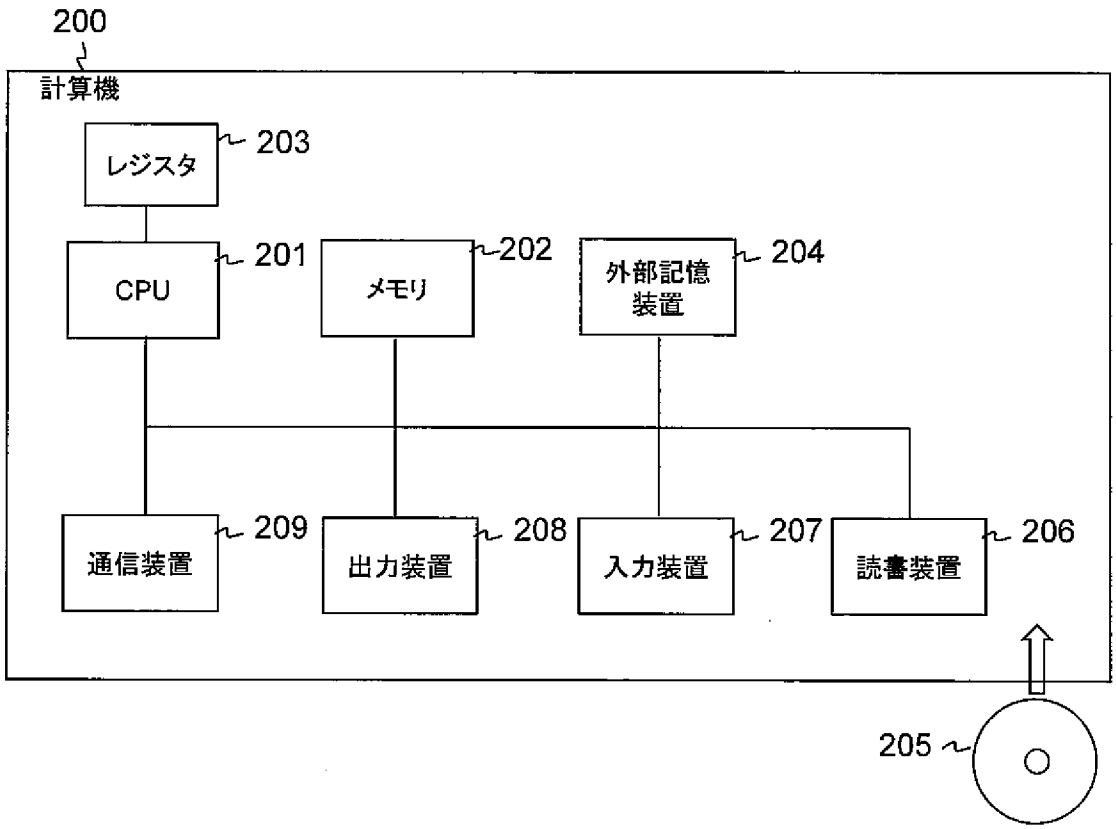
[図2]

図2



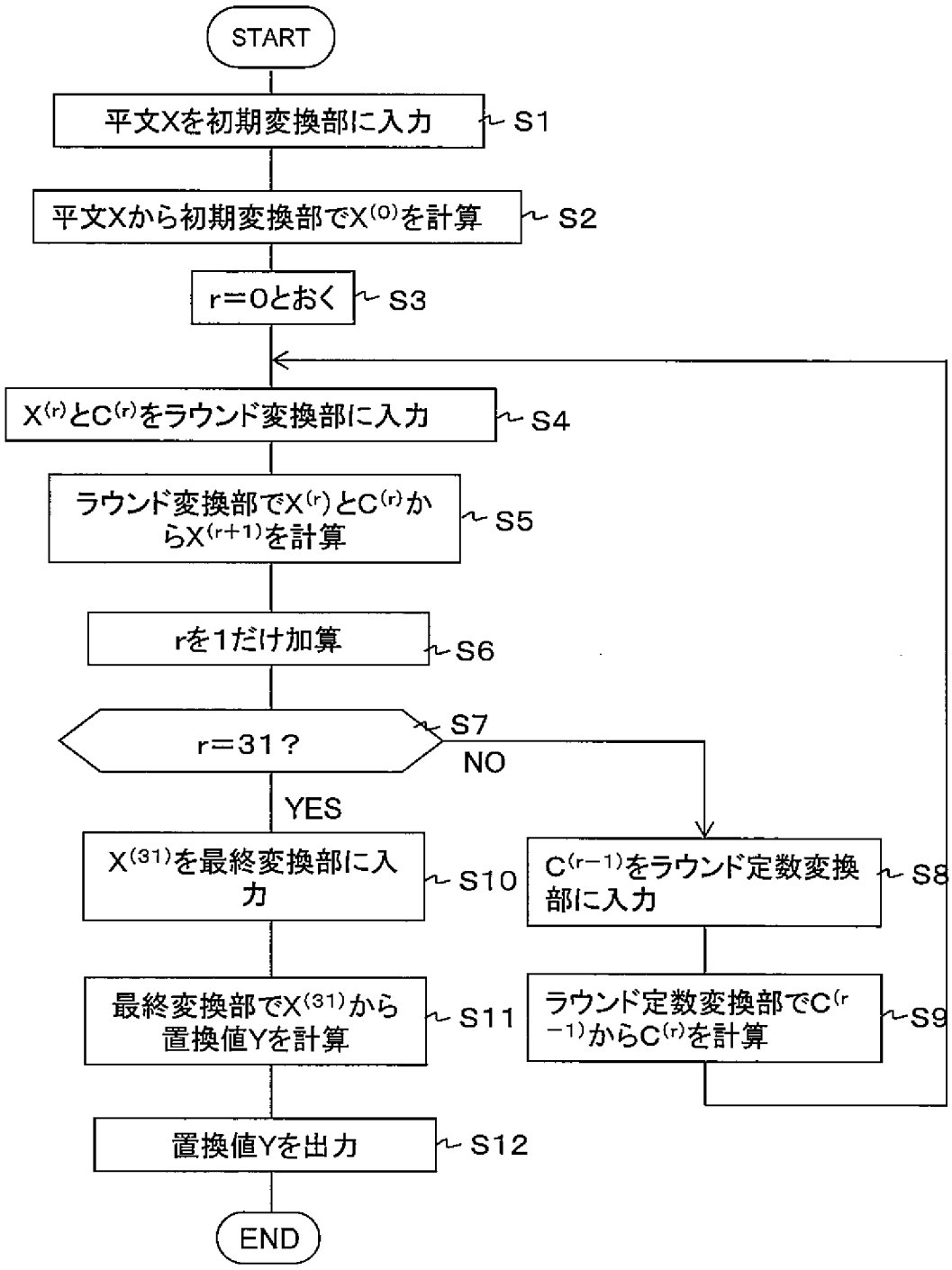
[図3]

図3


200
計算機
レジスタ
203
CPU
201
メモリ
202
外部記憶装置
204
通信装置
209
出力装置
208
入力装置
207
読書装置
206
205

[図4]

図4


START
平文Xを初期変換部に入力
S1
平文Xから初期変換部でX(0)を計算
S2
r=0とおく
S3
X(r)とC(r)をラウンド変換部に入力
S4
ラウンド変換部でX(r)とC(r)からX(r+1)を計算
S5
rを1だけ加算
S6
r=31?
S7
NO
YES
X(31)を最終変換部に入力
S10
C(r-1)をラウンド定数変換部に入力
S8
最終変換部でX(31)から置換値Yを計算
S11
ラウンド定数変換部でC(r-1)からC(r)を計算
S9
置換値Yを出力
S12
END

[図5]

図5

	従来技術	本発明	減った理由(本願の処理構成)
鍵加算	2	4	—
非線形変換層	32	20	テーブル参照ではなく、レジスタ値同士の論理演算を組み合わせて非線形変換層を実現することにより、演算量を削減した。
線形変換層	192	10.5	従来技術も本願もビット位置の入れ替えであるが、従来技術では、1ビットずつ取り出して新しい位置に設置している。本願では、レジスタの入れ替えとレジスタ値の順かシフトでこの処理を実現することで、演算量を削減した。
計(1ラウンド)	226	34.5	上記2点を同時に実現可能となるように、線形変換のビット位置入れ替え方を定めた。

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2012/082861

A. CLASSIFICATION OF SUBJECT MATTER
G09C1/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G09C1/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2013
Kokai Jitsuyo Shinan Koho	1971-2013	Toroku Jitsuyo Shinan Koho	1994-2013

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	Kenji OKUMA et al., "Block Ango PRESENT ni Taisuru Senkei Kaidoku", 2009 Nen Symposium on Cryptography and Information Security (SCIS2009), 20 January 2009 (20.01.2009), 1A2-3	1-3,9,10 4-8
Y A	W. Wu, et al., "LBlock: A Lightweight Block Cipher", [online], Cryptology ePrint Archive, 2011.06.29, Report 2011/345. [retrieved on 2013.01.16]. Retrieved from the Internet: <URL: http://eprint.iacr.org/2011/345.pdf>.	1-3,9,10 4-8

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 January, 2013 (16.01.13)	29 January, 2013 (29.01.13)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１２／０８２８６１

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00(2006.01)i

Ｂ．　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Int.Cl.　G09C1/00

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２０１３年
日本国実用新案登録公報	１９９６－２０１３年
日本国登録実用新案公報	１９９４－２０１３年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

Ｃ．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求項の番号
Y A	大熊建司, 他, "ブロック暗号 PRESENT に対する線形解読", 2009 年 暗号と情報セキュリティシンポジウム(SCIS2009), 2009.01.20, 1A2-3.	1-3,9,10 4-8
Y A	W. Wu, et al., "LBlock: A Lightweight Block Cipher", [online], Cryptology ePrint Archive, 2011.06.29, Report 2011/345. [retrieved on 2013.01.16]. Retrieved from the Internet: <URL: http://eprint.iacr.org/2011/345.pdf>.	1-3,9,10 4-8

☐　Ｃ欄の続きにも文献が列挙されている。

☐　パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
１６．０１．２０１３	２９．０１．２０１３

国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
東京都千代田区霞が関三丁目４番３号

特許庁審査官（権限のある職員）　５Ｓ　４５３８
金沢　史明
電話番号　０３－３５８１－１１０１　内線　３５４６