

【公報種別】特許法第 17 条の 2 の規定による補正の掲載

【部門区分】第 6 部門第 3 区分

【発行日】平成 16 年 9 月 30 日 (2004.9.30)

【公開番号】特開 2000-182012 (P2000-182012A)

【公開日】平成 12 年 6 月 30 日 (2000.6.30)

【出願番号】特願 平 10-354156

【国際特許分類第 7 版】

G 0 6 K 19/073

G 0 9 C 1/00

H 0 4 L 9/10

【F I】

G 0 6 K 19/00 P

G 0 9 C 1/00 6 6 0 A

H 0 4 L 9/00 6 2 1 Z

【手続補正書】

【提出日】平成 15 年 9 月 12 日 (2003.9.12)

【手続補正 1】

【補正対象書類名】明細書

【補正対象項目名】発明の名称

【補正方法】変更

【補正の内容】

【発明の名称】情報処理装置、耐タンパ処理装置

【手続補正 2】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

データを演算する演算装置と、

データおよびプログラムを記憶する記憶装置と、

上記演算装置が上記記憶装置にアクセスするためのデータバスとを有し、

上記演算装置が、

攪乱用データを生成する攪乱用データ生成処理と、

入力データを上記攪乱用データを用いて暗号化した変形データを出力するデータ変形処理と、

上記変形データに所定の処理を行い、処理結果である処理済み変形データを出力する変形データ処理と、

上記攪乱用データに上記所定の処理を行い、処理結果である処理済み攪乱用データを出力する攪乱用データ処理と、

上記処理済み攪乱用データを用いて、上記処理済み変形データを復号化した処理済みデータを出力するデータ逆変形処理と、

を行うことを特徴とする情報処理装置。

【請求項 2】

データを演算する演算装置と、

データおよびプログラムを記憶する記憶装置と、

上記演算装置が上記記憶装置にアクセスするためのデータバスとを有し、

上記演算装置が、

攪乱用データを生成する攪乱用データ生成処理と、
上記攪乱用データを用いて、入力データを変形データに暗号化するデータ変形処理と、
上記変形データに所定の処理を行い、処理結果である処理済み変形データを出力する変形データ処理と、
上記攪乱用データに上記所定の処理を行い、処理結果である処理済み攪乱用データを上記記憶装置に記憶する処理済み攪乱用データ記録処理と、
上記記憶装置より読み出した上記処理済み攪乱用データを用いて、上記処理済み変形データを処理済みデータに復号化をデータ逆変形処理と、
を行うことを特徴とする情報処理装置。

【請求項 3】

請求項 1、2 のいずれかに記載の情報処理装置において、
上記攪乱用データ生成処理において、上記攪乱用データとして乱数を使用することを特徴とする情報処理装置。

【請求項 4】

請求項 1、2 のいずれかに記載の情報処理装置において、
上記データ変形処理は、入力データと攪乱用データとの排他的論理和演算の結果を上記変形データとして出力する処理であり、
上記データ逆変形処理は、上記処理済み変形データと上記処理済み攪乱用データとの排他的論理和演算の結果を上記処理済みデータとして出力する処理であることを特徴とする情報処理装置。

【請求項 5】

請求項 1、2 にいずれかに記載の情報処理装置において、
上記データ変形処理は、上記入力データと攪乱用データとの加減算の結果を上記変形データとして出力する処理であり、
上記データ逆変形処理は、上記処理済み変形データと上記処理済み攪乱用データとの加減算を上記処理済みデータとして出力する処理であることを特徴とする情報処理装置。

【請求項 6】

請求項 1、2 にいずれかの記載の情報処理装置において、
上記データ変形処理は、上記入力データと攪乱用データとの乗除演算の結果を上記変形データとして出力する処理であり、
上記データ逆変形処理は、上記処理済み変形データと上記処理済み攪乱用データとの乗除演算の結果を上記処理済みデータとして出力する処理であることを特徴とする情報処理装置。

【請求項 7】

請求項 1、2 にいずれかの記載の情報処理装置において、
上記データ変形処理は、上記入力データと攪乱用データとの排他的論理和演算の結果を上記変形データとして出力する処理であり、
上記所定の処理は、転置処理もしくは換字処理であり、
上記データ逆変形処理は、上記処理済み変形データと上記処理済み攪乱用データとの排他的論理和演算の結果を上記処理済みデータとして出力する処理であることを特徴とする情報処理装置。

【請求項 8】

請求項 7 に記載の情報処理装置において、
上記転置処理もしくは換字処理は、テーブルに従って行われることを特徴する情報処理装置。

【請求項 9】

データを演算する演算装置と、
データおよびプログラムを記憶する記憶装置と、
上記演算装置が上記記憶装置にアクセスするためのデータバスとを有し、
上記演算装置が、

入力データに剰余演算の法 N の任意の倍数を加算し、変形データとして出力するデータ変形処理と、

上記記変形データに上記剰余演算の加減算を行い、処理結果である処理済み変形データを出力する変形データ処理と、

を行うことを特徴とする情報処理装置。

【請求項 10】

データを演算する演算装置と、

データおよびプログラムを記憶する記憶装置と、

上記演算装置が上記記憶装置にアクセスするためのデータバスとを有し、

上記演算装置が、

入力データと剰余演算の法 N の任意の倍数の乗算結果と 1 との加算を行い、変形データとして出力するデータ変形処理と、

上記記変形データに上記剰余演算の乗算を行い、処理結果である処理済み変形データを出力する変形データ処理と、

を行うことを特徴とする情報処理装置。

【請求項 11】

データを演算する演算装置と、

データおよびプログラムを記憶する記憶装置と、

上記演算装置が上記記憶装置にアクセスするためのデータバスとを有し、

上記演算装置が、

ある数 X を生成する攪乱用データ生成処理と、

入力データに上記 X を所定回数乗算もしくは上記攪乱用データの倍数を乗算し、乗算結果を変形データとして出力するデータ変形処理と、

上記変形データに所定の処理を行い、処理結果である処理済み変形データを出力する変形データ処理と、

上記処理済み変形データに上記ある数 Y を上記所定回数乗算もしくは上記ある数 Y の倍数を乗算し、乗算結果を処理済データとして出力するデータ逆変形処理と、を行い

上記ある数 X とある数 Y は、 N をある整数として

$$1 = X \times Y \bmod N$$

を満たすことを特徴とする情報処理装置。

【請求項 12】

請求項 11 に記載の情報処理装置において、

上記攪乱用データと上記攪乱用データの逆数は、いずれか一方が 2 であり、他方が $(N + 1) / 2$ であることを特徴とする情報処理装置。

【手続補正 3】

【補正対象書類名】明細書

【補正対象項目名】0026

【補正方法】変更

【補正の内容】

【0026】

xor X1 R1 (式 4)

xor X2 R2 (式 5)

logica_shiftl R1 (式 6)

xor R1 R2 (式 7)

logica_shiftl X1 (式 8)

xor X1 X2 (式 9)

xor X2 R2 (式 10)

ここで、 $X1$ と $X2$ は勝手に選んだ乱数であり、攪乱用データである。式 4 と式 5 では、レジスタ $R1$ と $R2$ に乱数 $X1$ と $X2$ を排他的論理和を行い、攪乱用データを使い、本物のデータの変形を行う変形処理である。式 6 と式 7 は式 2 と式 3 と同じデータ処理である

が、変形処理を行ったため、R 1 と R 2 の値は、本物のデータの値とは異なったデータとなっている。式 8 と式 9 は、攪乱用データ自体をデータ処理を行っているものである。式 10 で、攪乱用データの処理結果と式 7 の処理結果を排他的論理和することにより、本物のデータにもどす逆変形処理である。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0043

【補正方法】変更

【補正の内容】

【0043】

$$\begin{aligned} \text{Output} &= g(\text{Xoutput}) = \text{OutputX} - \text{Xoutput} \\ &= \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3) - (\text{X}(1) + \text{X}(2) - \text{X}(3)) \\ &= \text{Input}(1) + \text{Input}(2) - \text{Input}(3) \end{aligned} \quad (\text{式 3 5})$$

正しい出力データを求めることができる。これは、加減算の演算では、ある値を加算して演算し、最期の結果に加算した値を減算すれば、正しい結果が求まることからである。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0044

【補正方法】変更

【補正の内容】

【0044】

データ処理 f が乗除算であれば、変形操作 h に乗算、あるいは除算をえらび、逆変形操作に除算、あるいは乗算を選ぶことにより、変形処理と逆変形処理を実現することができる。この理由は、加減算と同じであり、乗除算の演算では、ある値を乗算（除算）して演算し、最期の結果に乗算（除算）した値を除算（乗算）すれば、正しい結果が求まることからである。

【手続補正 6】

【補正対象書類名】明細書

【補正対象項目名】0048

【補正方法】変更

【補正の内容】

【0048】

$$\text{OutputX} = \text{InputX}(1) + \text{InputX}(2) - \text{InputX}(3) \bmod N \quad (\text{式 3 8})$$

この式は、式 3 7 を使うことにより

$$\begin{aligned} \text{OutputX} &= (\text{Input}(1) + k(1)*N) + (\text{Input}(2) + k(2)*N) - (\text{Input}(3) + k(3)*N) \bmod N \\ &= (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) + (k(1)*N + k(2)*N - k(3)*N) \bmod N \end{aligned} \quad (\text{式 3 9})$$

と変形できる。そして、剰余演算の特徴である

$$0 = k * N \bmod N \quad (\text{式 4 0})$$

を使うことにより、式 3 9 の 2 番目の括弧内の値は 0 となり、式 3 9 は、

$$\text{OutputX} = (\text{Input}(1) + \text{Input}(2) - \text{Input}(3)) \bmod N \quad (\text{式 4 1})$$

となる。すなわち、変形した入力データの演算結果は、本来の演算結果と同じである。これは、剰余演算の特徴を使うことにより、攪乱用データのデータ処理と逆変形処理が不要である例である。これは、攪乱データのデータ処理の結果が

$$\text{Xoutput}(i) = k(1) * N + k(2) * N - k(3) * N \bmod N \quad (\text{式 4 2})$$

が、そもそも 0 になるため、攪乱用データのデータ処理と逆変形処理が不要であるためである。

【手続補正 7】

【補正対象書類名】明細書

【補正対象項目名】0 0 5 1

【補正方法】変更

【補正の内容】

【0 0 5 1】

$$\text{InputX}(i) = \text{Input}(i) * (k(i) * N + 1) \quad (\text{式 4 4})$$

そして、変形した入力データを使い、乗算剰余演算を行う。

【手続補正 8】

【補正対象書類名】明細書

【補正対象項目名】0 0 5 2

【補正方法】変更

【補正の内容】

【0 0 5 2】

$$\text{OutputX} = \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) \bmod N \quad (\text{式 4 5})$$

この式は、式 4 4 を使うことにより

$$\begin{aligned} \text{OutputX} &= (\text{Input}(1) * (k(1)*N + 1)) * (\text{Input}(2) * (k(2)*N + 1)) * (\text{Input}(3) \bmod N) \\ &= (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * \\ &\quad ((k(1)*N + 1) * (k(2)*N + 1) * (k(3)*N + 1)) \bmod N \quad (\text{式 4 6}) \end{aligned}$$

と変形できる。そして、剰余演算の特徴である

$$0 = k * N \bmod N \quad (\text{式 4 7})$$

を使うことにより、式 4 6 は、

$$\begin{aligned} \text{OutputX} &= (\text{Input}(1) * \text{Input}(2) * \text{Input}(3)) * (1 * 1 * 1) \bmod N \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N \quad (\text{式 4 8}) \end{aligned}$$

となる。すなわち、変形した入力データの演算結果は、本物の演算結果と同じである。これも、剰余演算の特徴を使うことにより攪乱データのデータ処理と逆変形処理が不要である例である。

【手続補正 9】

【補正対象書類名】明細書

【補正対象項目名】0 0 5 8

【補正方法】変更

【補正の内容】

【0 0 5 8】

$$\begin{aligned} \text{Output} &= \text{OutputX} * Y * Y * Y \bmod N \\ &= \text{InputX}(1) * \text{InputX}(2) * \text{InputX}(3) * Y * Y * Y \bmod N \\ &= \text{Input}(1) * X * \text{Input}(2) * X * \text{Input}(3) * X * Y * Y * Y \bmod N \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * X * X * Y * Y * Y \bmod N \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) * X * Y * X * Y * X * Y \bmod N \\ &= \text{Input}(1) * \text{Input}(2) * \text{Input}(3) \bmod N \quad (\text{式 5 3}) \end{aligned}$$

式 5 3 では、式 4 9 の性質を使った。この例では、攪乱用データのデータ処理は不要であるが、攪乱用データを掛けた回数だけ、法 N でその逆数を逆変形処理で掛ければ、正しい結果を得ることができる。

【手続補正 1 0】

【補正対象書類名】明細書

【補正対象項目名】0 0 6 5

【補正方法】変更

【補正の内容】

【0 0 6 5】

$\text{DataX1} = \text{TableX1X2X3}(\text{GyouX2}, \text{RetsuX3})$ (式 6 0)

図 3 5 のテーブルはすでに攪乱用データ X1 で変形されているため、これ以降は、攪乱用データを X1 として処理を行う。式 5 6 から式 6 0 までの処理で真のデータを使用していない。

【手続補正 1 1】

【補正対象書類名】明細書

【補正対象項目名】0 0 6 8

【補正方法】変更

【補正の内容】

【0 0 6 8】

図 4 は、基本的な手順である。攪乱用データ生成手段で攪乱用データ X_i を生成する (4 0 1)。この一般的な方法としては、乱数発生器や擬似乱数を使って必要な長さの乱数を生成する方法がある。次に、データ変形処理手段 (4 0 6) で、入力データ D1 を攪乱用データ X_i (4 0 5) で変形し、変形データ H1 (4 0 7) を生成する。変形方法は、先に述べたように、排他的論理和や、加減算、乗除算などがある。そして変形データ H1 を使ってデータ処理を変形データ処理手段 (4 0 8) で行い、処理済み変形データ H2 を生成する。一方、攪乱用データ X_i は入力データと同じデータ処理を攪乱用データ処理手段 (4 0 3) で行い、処理済みの攪乱用データ X_0 (4 0 4) を作成する。そして、攪乱用データ X_0 と処理済み変形データ H2 を使い、データ逆変形処理手段 (4 1 0) で真の処理済みデータ D2 (4 1 1) を求める。データ変形処理手段 (4 0 6) やデータ逆変形処理手段 (4 1 0) の方法には、すでに説明した、排他的論理和や加減算、乗除算、剰余演算などがある。

【手続補正 1 2】

【補正対象書類名】明細書

【補正対象項目名】0 0 7 0

【補正方法】変更

【補正の内容】

【0 0 7 0】

$H1 = D1 \text{ xor } X1_i$

$H2 = f1(H1)$

$X1_0 = f1(X1_i)$

$H31 = H2 \text{ xor } X2_i$

$H32 = D2 \text{ xor } X2_i$ (式 6 1)

$H4 = f2(H31, H32)$

$X2_0 = f2(X2_i, X2_i)$

$H5 = H4 \text{ xor } X2_0$

$D2 = H5 \text{ xor } X1_0$

ここで $f1$ と $f2$ はデータ処理操作である。この例のように、第 2 のデータ処理 $f2$ で、別のデータ D2 を使う場合で、そのデータ D2 を第 2 の攪乱用データで変形して使用する場合に、本実施例の処理手順を使うと有効である。

【手続補正 1 3】

【補正対象書類名】明細書

【補正対象項目名】0 0 7 2

【補正方法】変更

【補正の内容】

【 0 0 7 2 】

H1 = D1 xor X10

H2 = f1(H1)

X10 = f1(X1i)

H3 = H2 xor X2i (式 6 2)

H4 = H3 xor X10

H5 = f2(H4)

X20 = f2(X2i)

D2 = H5 xor X2i

これは、処理操作が複数あり、攪乱用データを複数使う必要な場合に有効である。__

【 手続補正 1 4 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 0 0 7 3

【 補正方法 】 変更

【 補正の内容 】

【 0 0 7 3 】

図 7 の実施例は、攪乱用データの面からは、データ処理をあらかじめ計算しておき、処理を効率化するものである。あらかじめ、攪乱用データ処理手段で処理済み攪乱用データ X0 を生成し (7 0 3)、処理済み攪乱用データ記録手段 (7 0 6) で、記録しておく。処理の中では、データ逆変形処理手段 (7 1 3) では、記録しておいた処理済み攪乱用データ (7 1 4) を読みだし、使用する。これは

、同じようなデータ処理を何回も実行する場合には効率の面で有効である。しかし、攪乱用データが何回も使われるため、情報の隠蔽性にとって、図 4 の実施例のように攪乱用データを毎回変更する方が、有効である。これは、処理速度と情報の隠蔽性のトレードオフから決まる。__

【 手続補正 1 5 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 0 0 7 4

【 補正方法 】 変更

【 補正の内容 】

【 0 0 7 4 】

図 8 の実施例は、第 1 と第 2 の攪乱用データの逆変形を統合して行い、その後その結果を使って、データの逆変形を行うものである。第 1 と第 2 の攪乱用データは、それぞれ、第 1 攪乱用データ処理手段 (8 0 3) と第 2 攪乱用データ処理手段 (8 0 7) で処理済み攪乱用データ X10 と X20 が作られる。それらのデータをデータ逆変形統合手段で統合し、統合処理済み攪乱用データ X0 を生成する。そのデータを使い、第 1 変形データ処理 (8 1 4) 及び第 2 変形データ処理 (8 1 8) を処理した結果である処理済み変形データ H 4 (8 1 9) の逆変形処理手段 (8 2 0) を行い、真の処理済みデータ D 2 を生成する。これは、個別に逆変形する場合よりも、処理済み攪乱用データを統合し、そのあと、まとめて変形処理を行う方法である。逆変形処理に処理時間がかかる場合に有効である。__

【 手続補正 1 6 】

【 補正対象書類名 】 明細書

【 補正対象項目名 】 0 0 8 1

【 補正方法 】 変更

【 補正の内容 】

【 0 0 8 1 】

変形処理 a (9 0 1) と変形処理 b (9 0 2) は基本的に同じ処理である。図 1 1 を用いて変形処理 a の平文の変形データの作成処理について説明する。攪乱用データ X 1 をランダムに生成する。これは、DES の暗号化 (あるいは復号化) 処理ごとに乱数発生器が擬似乱数を使って生成する (1 1 0 2)。毎回、異なった攪乱用データを使用する。次に攪乱

用データX 1と平文PをXOR(排他的論理和)を実行し、変形平文(変形した平文)PX1を生成する(1103)。DESの場合は、平文は64ビットであるが、生成する乱数は、64ビットでも8ビットでも構わない。しかし、64ビット以下ならば、拡張するようなことを行い、64ビットの攪乱用データX 1を生成する必要がある。生成した乱数が8ビットであれば、それを8回繰り返して64ビットの攪乱用データX 1を生成してもよい。ここで、排他的論理和(XOR)を使って変形したため、攪乱用データX 1と変形平文PX1をXORすると、平文Pが生成される。

【手続補正17】

【補正対象書類名】明細書

【補正対象項目名】0083

【補正方法】変更

【補正の内容】

【0083】

次に、IP処理(902)について説明する。IP処理は、図37に示すテーブルにしたがって平文64ビットの並びを入れ替えるものである。テーブルに従い、出力の第1ビットは入力第58ビット、出力の第2ビットは入力第50ビット、出力の第64ビットは入力第7ビットと入れ替える。本実施例でのIP処理を、図12を用いて説明する。まず、変形平文PX1をIP処理し、IP処理済み変形平文PX1IPを生成する(1202)。ビットの入れ替えは、図37のテーブルに従う。次に、攪乱用データX 1も同じようにIP処理し、IP処理済み攪乱用データX1IPを生成する(1203)。IP処理済み変形平文PX1IPとIP処理済み攪乱用データX1IPとは排他的論理和を取ると、平文をIP処理した結果を生成することができる。これは、IP処理がビットの移動であるため、攪乱用データも変形平文PX1と同じように移動したため、ビットごとに排他的論理和を実行すると真のデータが求まるという関係が維持されているためである。IP処理の下位32ビットが第1段のf関数(903)と第2段の排他的論理和に使われ、上位32ビットは、排他的論理和(909)の入力となる。

【手続補正18】

【補正対象書類名】明細書

【補正対象項目名】0092

【補正方法】変更

【補正の内容】

【0092】

$$Z = P \text{ xor } K \quad (\text{式65})$$

$$Z1 = PX1 \text{ xor } KX2$$

$$= (P \text{ xor } X1) \text{ xor } (K \text{ xor } X2)$$

$$= P \text{ xor } X1 \text{ xor } K \text{ xor } X2$$

$$= (P \text{ xor } K) \text{ xor } (X1 \text{ xor } X2)$$

$$= Z \text{ xor } (X1 \text{ xor } X2) \quad (\text{式66})$$

すなわち、Z 1を真のデータに戻すための攪乱用データとしてPとKの攪乱用データを排他的論理和を使えばよいことが分かる。暗号鍵と選択的転置の実行結果の排他的論理和の処理では、Sボックス入力データSinputXの攪乱用データとして、平文用の攪乱用データから生成したIP処理及びE転置処理済み攪乱用データX1PEと、暗号鍵用の攪乱用データから生成したPC-1及びLS、PC-2処理済み攪乱用データX2PC1LSPC2とをXORして生成したSボックス入力データ攪乱用データXSinputを使えばよいことが分かる。

【手続補正19】

【補正対象書類名】明細書

【補正対象項目名】0093

【補正方法】変更

【補正の内容】

【0093】

次に、Sボックスの処理について図18を用いて説明する。Sボックス入力データSinputXから6ビットずつ取り出し、8つのSボックスについて処理を行う。DESで使われている1番目のSボックスを、図25で示す。8つのSボックスは形式は同じであるが、それぞれフィールドのデータが異なる。それぞれのSボックスの処理では、まず、Sボックス入力データSinputの上位i番目の6ビットのサブデータSubSinputX(i)を取り出す(1805)。前もって作成しておいたSボックスを変形した変形Sボックステーブルのアドレス攪乱用データXsa(i)とSubSinputX(i)をXORし、SubSinputXXsa(i)を生成する(1806)。そのSubSinputXsa(i)をSボックス入力データ攪乱用データXSinputの上位i番目の6ビットとXORし、SubSinputXsa(i)を生成する(1807)。SubSinputXsa(i)は、i番目のSボックスを取り出すための本物のアドレスデータに対して、アドレス攪乱用データXsa(i)が排他的論理和された値である。SubSinputX(i)とXSinput(i)を排他的論理和(XOR)すると、それは、真のデータに戻るなのである。XSinput(i)と排他的論理和する前に、SubSinputX(i)とXsa(i)の排他的論理和を行い、その後で、XSinput(i)を排他的論理和(XOR)を実行する。こうすることにより、本物のデータの処理を行う必要がなくなり、その電流波形からデータを推定することが困難になっている。次に、SubSinputXsa(i)を使って、変形Sボックステーブルのアドレスを計算する(1808)。本来のSボックスのテーブルを参照するためのアドレスが変形されているため、テーブルをあらかじめ変形しておく必要がある。計算したアドレスを元に、変形SボックステーブルS(i)からSボックス出力データSoutputX3(i)を取り出す(1809)。そして、同時にSボックスの出力データSoutputX3(i)用の攪乱用データX3(i)を取り出す(1810)。8つのSボックスを処理し、SoutputX3(i)とX3(i)をiを1から8までのデータをそれぞれ連結することによって、SoutputX3とX3を生成する。これ以降は、処理データは、SoutputX3となり、攪乱用データはX3となる。

【手続補正20】

【補正対象書類名】明細書

【補正対象項目名】0095

【補正方法】変更

【補正の内容】

【0095】

次に、アドレスの攪乱を行う。まず、Xsa1をXsa(i)の上位1ビットと下位1ビットから作った2ビットとし、Xsa2をXsa(i)の上位2ビットから5ビットから作った4ビットのデータとする。これは、Sボックスのアドレスの計算方法に由来している。そして、図26の行番号と列番号をそれぞれkとlとすると、それぞれにXsa1とXsa2を排他的論理和を行う(2412)。新しくできた行番号と列番号をk2とl2とすると、i番目の変形SボックステーブルS(i)のk2行l2列にデータd2を格納する(2413)。この処理の例を図27に示す。図27は、図26の行と列に対して攪乱用データとして“2”と“9”を選んで作成したものである。わかりやすいように、行と列のデータの位置はそのままにして行と列の番号だけを変更している。図25で、3行1列のデータ12(2504)は、図27では、1行8列に移り、値が11に変形されている。この例では、データの攪乱用データは“7”であり、アドレス用の攪乱データは行が“2”であり、列が“9”である。このようにして8つのSボックスを変形する。本実施例では、この処理は、DESの最初に行っておく。変形したSボックステーブルは、DESの16段で使用する。