(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第4317593号
(P4317593)

(45)発行日　平成21年8月19日(2009.8.19)　(24)登録日　平成21年5月29日(2009.5.29)

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

請求項の数 11　(全 7 頁)

(21)出願番号　特願平10-521101
(86)(22)出願日　平成9年11月4日(1997.11.4)
(65)公表番号　特表2001-503534(P2001-503534A)
(43)公表日　平成13年3月13日(2001.3.13)
(86)国際出願番号　PCT/FR1997/001975
(87)国際公開番号　WO1998/020643
(87)国際公開日　平成10年5月14日(1998.5.14)
審査請求日　平成16年10月13日(2004.10.13)
(31)優先権主張番号　96/13411
(32)優先日　平成8年11月4日(1996.11.4)
(33)優先権主張国　フランス(FR)

(73)特許権者　595040744
サントル・ナショナル・ドゥ・ラ・ルシェルシュ・シャンティフィク
CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
フランス国、７５７９４　パリ・セデックス　１６、リュ・ミシェル・アンジュ　３
(74)代理人　100092277
弁理士　越場　隆
(72)発明者　ヴォードネー，セルジュ
フランス国　９１２２０　ブレティニーシュ　ロルジュ　リュ　デ　ザカシア　１１

最終頁に続く

(54)【発明の名称】データの非相関化方法

(57)【特許請求の範囲】

【請求項１】

計算機によって使用可能な記録媒体に格納されたデータの暗号方法であって、計算機が入力情報ｘを関数Ｆでコード化した情報Ｆ（ｘ）を出力するキーを用いて処理する方法において、関数Ｆは少なくとも２に等しいランクの非相関モジュールM_Kを使用し、Ｆ（ｘ）＝［Ｆ’（M_K）］（ｘ）である（ここで、Ｋはランダムキーであり、Ｆ’はコード化関数である）ことを特徴とするデータの暗号方法。

【請求項２】

前記入力情報ｘを所定の長さの要素x_0に分割することを特徴とする請求項１に記載の暗号方法。

【請求項３】

前記関数Ｆが、Ｆ（ｘ）＝Ｆ’（M_K（ｘ））の形式であることを特徴とする請求項１および２のいづれか１項に記載の暗号方法。

【請求項４】

前記コード化関数Ｆ’が、以下に示すよう２つの関数Ｆ”およびＧ”に分割されることを特徴とする請求項１および２のいづれか１項に記載の暗号方法：

Ｆ（ｘ）：Ｆ”（M_K（Ｇ”（ｘ））

【請求項５】

前記非相関モジュールM_Kが、逆変換可能であることを特徴とする請求項１から４のいづれか１項に記載の暗号方法。

【請求項６】
前記非相関モジュールが、M_K（ｘ）=ａｘ＋ｂ（ここで、K=（ａ，ｂ）、ａ≠０）であることを特徴とする請求項５に記載の暗号方法。
【請求項７】
前記非相関モジュールが、M_K（ｘ）=ａ/（ｘ＋ｂ）＋ｃ（ここで、K=（ａ，ｂ，ｃ）、ａ≠０）であることを特徴とする請求項５に記載の暗号方法。
【請求項８】
前記関数Ｆが、関数F_iをｎ回の反復処理するファイステル関数であることを特徴とする請求項５に記載の暗号方法。
【請求項９】
前記各反復において、F_i（ｘ）=F'_i（M_K（ｘ））であることを特徴とする請求項８に記載の暗号方法。
【請求項１０】
前記各反復において、F_i（ｘ）=F''_i（M_K（G''_i（ｘ））であることを特徴とする請求項８に記載の暗号方法。
【請求項１１】
M_K（ｘ）=$k_1+k_2x+k_3x^2+\cdots+k_tx^{t-1}$（ここで、K=（$k_1$，$k_2$，$k_3$，・・・，$k_t$））であることを特徴とする請求項１０に記載の暗号方法。
【発明の詳細な説明】
本発明は処理装置で利用可能な媒体に記録したデータを非相関化する方法に関する。
知られているデータの暗号化方法または暗号方法は多くある。これらはデータをコード化する役目をし、キーを所有する認定された受信者のみデータを読むことができる。暗号化の重要性は情報通信網と同時に高まっており、法律制定に伴いその利用が普及することは予想できる。
いくつかの暗号化方法により、無条件機密保護を提供することができるが、処理に手間がかかる技術手段を必要とするため、通信が遅くなったり、キーの交換管理の費用が高くなったりし、実際に使用できないものさえある。
例えば、暗号化されていないメッセージのフローを暗号化するために、バーナム暗号化方法(Vernam encryption method)では同じ長さのキーのフローを必要とし、送信者と受信者間の同期を達成するのが難しくなる。
無条件機密保護の条件は、1949年、シャノン(Shannon)によって形式化された。彼は情報理論を基に、無条件機密保護には、キーが、劣化することなく暗号化できるメッセージ全体の大きさと少なくとも同等でなくてはならないことを示した。
暗号化操作は、処理装置で利用可能な媒体に記録され、よく送信されるデータの保護を確実にするため実施する。一連の機密にすべきメッセージの暗号化には、少数のメッセージに対して独立にこれらの操作を実施しなければならない。
現在用いられる主な暗号化操作は米国政府が採用するデータ暗号化規格(DES:digital data encryption standard)である。この操作はいわゆるファイステル(Feistel)スキームに続く単純な関数の(16段の)反復に基づく。反復数が多いのは、暗号化されたメッセージ間の相関を弱めるためである。
DESについては多くの文献、特にダグラス スティンソン(Douglas STINSON)による「暗号化、理論と実践」(＂Encryption,Theory and Practice＂(International Thompson Publishing))という題名の出版物で説明されている。
暗号化の信頼性を改善し、徹底的な調査から保護するため、キーの長さを長くし、オーダー1の非相関の導入が提案されている。これは下記2つの文献の著者達が提案してきたことである:Advances in Cryptology-CRYPTO 96,16th Annual International Cryptology Conference,Santa Barbara,August 18-22 1996,Proceedings no.Conf.16,August 18th 1996,Koblitz N(ED),pages 252-267 KILIAN J.et al.、およびAdvances in Cryptology-ASIACRYPT,Fujiyoshida,November 11-14 1991,no.Conf.1,November 11th 1991,Hideki Imai;Rivest R L;Tsutomu Matsumoto,pages 210-224 by EVEM S.et al.

しかし、この方法では近年開発された線形および微分暗号解読技術で可能になった攻撃から保護するのに十分ではない。
本発明の目的は、最適な機密性を提供し、大規模でない計算ソースのみを必要とする比較的単純な関数を用いて実施できるデータ暗号法を提供することにある。
この目的のため、本発明は計算機によって使用可能な媒体に格納されたデータの暗号方法に関し、この方法では、計算機が入力情報xを関数Fでコード化した情報F(x)を出力するキーを用いて処理する。
本発明において、関数Fは少なくとも2に等しいランクの非相関モジュールM_Kを使用し、F(x)=[F'(M_K)](x)である(ここで、Kはランダムキーであり、F'は暗号関数である)。
一般に、非相関モジュールは、キーを含む関数M_Kでメッセージxを変換する役目を行い、ランダムな変数のキーを有する任意のt個の異なるメッセージから得た分散$M_K(x_1),...,M_K(x_t)$が均一またはほぼ均一な分散を有するようにする。
こうした非相関モジュールは、入力情報xに対して所定の長さのデータx_0を出力する情報分割装置(information dividing device)の後、データ暗号化装置内で使用できる。
本発明は、関数Fでコード化したメッセージ$c_1,...,c_t$のt個のブロックがその関数についてのいかなる統計的情報も与えないように実施できる。
それぞれが個々の利点を有する種々の実施例において、本発明は任意の技術的に実行可能な組み合わせによる下記特性を有する:
－入力情報xは所定の長さの要素x_0に分割され、
－関数Fは、F(x)=F'(M_K(x))の形式であり、
－コード化関数F'は2つの関数F"およびG"に分割され、
F(x)=F"(M_K(G"(x)))、
－非相関モジュールM_Kは逆変換可能であり、
－非相関モジュールは、$M_K(x)=ax+b$であり(ここで、K=(a,b)、$a\neq0$)、
－非相関モジュールは、$M_K(x)=a/(x+b)+c$であり(ここで、K=(a,b,c)、$a\neq0$)、
－関数Fは、関数F_iをn回の反復それぞれに適用するファイステル関数であり、
－非相関モジュールM_Kは逆変換可能であり、
－各反復で、$F_i(x)=F'_i(M_K(x))$であり、
－各反復で、$F_i(x)=F"_i(M_K(G"_i(x)))$であり、
－$M_K(x)=k_1+k_2x+k_3x^2+...+k_tx^{t-1}$
(ここで、$K=k_1,k_2,k_3,...,k_t$)である。
以下、本発明を図および実施例を用いてより詳細に説明する。
図1は、8段の反復のファイステルスキームに適用した本発明を示す。
記録データを暗号化する本発明の方法は、F(x)=[F'(M_K)](x)(ここで、Kはランダムなキー、F'はコード化関数である)である非相関モジュールM_Kを用いた秘密キーを実施するのが有利である。
コード化関数F'は2つの関数F"およびG"に分割され、F(x)=F"(M_K(G"(x)))であるのが有利である。
このような非相関モジュールは、逆変換可能な関数Fおよび任意の関数Fに使用できる。
関数Fが逆変換可能である時、暗号方法は暗号化方法である。つまり、キーの保有者が入力情報を再生できる。関数Fが逆変換できない場合、暗号方法でデータを認証できる。
パラメータ値t=2で、関数M_Kは:
$M_K(x)=ax+b$
であるのが有利である(ここで、K=(a,b)、$a\neq0$、記号+はメッセージスペースの翻訳を表す)。
非相関はオーダー2で完全になる。逆変換操作は:
$(M_K)^{-1}(y)=a^{-1}y-a^{-1}b$
パラメータ値t=3の場合の他の実施例で、関数M_Kは:
$M_K(x)=a/(x+b)+c$
であるのが有利である。(ここで、K=(a,b,c)、$a\neq0$。この演算では、1/0=0とする。)逆変

換操作は下記のとおり:
$(M_K)^{-1}(y)=a/(y-c)-b$
非相関モジュールM_Kの使用は逆変換不可能な関数の場合でも同様に有利である。
加算および乗算を定義するメッセージのような代数構造を用いる。例えば、有限な集合における演算または素数の切り捨てモジュロ演算を用いる。
全てのパラメータ値tに、$M_K(x)=k_1+k_2x+k_3x^2+...+k_tx^{t-1}$ここで、$K=(k_1,k_2,k_3,...,k_t)$の形式の非相関関数を提案できる。
ファイステル1暗号化のスキームを図1に示す。
64ビットの平文xのブロックに対して、$x=L_0R_0$を設定する(ここで、L_0は数列xの初めの32ビットを、R_0は残りの32ビットを有する)。
等しい関数fの4段の反復をxに適用する。$1 \leqq i \leqq 4$で、L_iR_iを下記の法則に従って計算する:
$L_i=R_{i-1}$
$R_i=L_{i-1}+f(R_{i-1}+K_i)$
ここで、＋記号は2本の数列のビットごとのEOR(排他的OR)を表す。K_1,K_2,K_3,K_4はKから計算された32ビットの数列である。
この結果は(L_4,R_4)となる。これは、例えば下記のように、非相関モジュールを適用する式L_4R_4に組み込まれる:
$K_5K_6 \times L_4R_4+K_7K_8$
(K_5,K_6,K_7,K_8は各々32ビットの数列である)。
この結果は、ファイステルスキームによる2番目の関数に対する入力$L'_0R'_0$の役目をし、$L'_4R'_4=F(x)$という結果を導く前記のものに類似する。
K'_1,K'_2,K'_3,K'_4もまた32ビットの数列である。
ここでキーは$K_1,K_2,K_3,K_4,K'_1,K'_2,K'_3,K'_4,K_5,K_6,K_7,K_8$である。
一般に、関数F”およびG”は任意の暗号化関数でよい。
ここで、2つの実施例を詳細に説明する:
これらの好ましい実施例の1番目では、8段の反復のファイステルスキームを用いる。G”は4つの関数f_1,f_2,f_3,f_4の連続処理、F”は4つの関数f_5,f_6,f_7,f_8の連続処理である(関数f_iは関数fおよびランダムキーKから定義する)。
関数fは下記の方法で定義する:
もしxが32ビットワードであれば、まず下記φ(x)を定義する:
$\phi(x)=x+256.S(x \bmod 256) \bmod 2^{32}$
ここで、Sは例えば付録1の表に表した関数で、uは横軸に、vは縦軸に表し、各々が16進数である。(u,v)=xを座標(u,v)に示す値を有する値S(x)に対応させる。
f(x)を下記式で定義する:
$f(x)=\phi(R^{11}_L(\phi(x))+r \bmod 2^{32})$
(ここで、R^{11}_Lは、11ビットの左への巡回置換であり、rは定数で、例えば下記のようにsで定義される:
r=b7s15162および

$$s=\sum_{i=0}^{\infty}\frac{1}{i!}=\text{b7e15162 8aed2a 6abf71 58809c f4f3c7 62e716...}$$

キーKは、各々64ビットの数列K_iを8個連結してできた256ビットの数列である:$K=(K_1K_2K_3...K_8)$。
ファイステルスキームは、関数f_iを用いて実施する:
$f_i(x)=f(x▲+▼k_i)$
ここで、k_iを下記のように定義し:

i	1	2	3	4
k_i	K1	$K_1 \oplus K_3$	$K_1 \oplus K_3 \oplus K_4$	$K_1 \oplus K_4$

i	5	6	7	8
k_i	K2	$K_2 \oplus K_3$	$K_2 \oplus K_3 \oplus K_4$	$K_2 \oplus K_4$

非相関モジュールは下記式となる:

$M(uv)=(uv\blacktriangle+\blacktriangledown K_5K_6)\times K_7K_8$

2番目の好ましい実施例では、32段の反復のファイステルスキームを使用する:

最初の実施例に比べて、キーKは2048ビットの数列、rはその値を保持し、関数fはf'に置換される:

$f'(x)=R^{11}{}_L(x)+r \bmod 2^{32}$

関数f_iは関数f'_iに置換される:

$f'_i(x)=f'(x.K_{2i-1}+K_{2i} \bmod 2^{32}-5)$。

付録1

	.0	.1	.2	.3	.4	.5	.6	.7
0.	8aed2a	6abf71	58809c	f4f3c7	62e716	0f38b4	da56a7	84d904
1.	bb1185	eb4f7c	7b5757	f59584	90cfd4	7d7c19	bb4215	8d9554
2.	cfbfa1	c877c5	6284da	b79cd4	c2b329	3d20e9	e5eaf0	2ac60a
3.	78e537	d2b95b	b79d8d	caec64	2c1e9f	23b829	b5c278	0bf387
4.	bbca06	0f0ff8	ec6d31	beb5cc	eed7f2	f0bb08	801716	3bc60d
5.	94640d	6ef0d3	d37be6	7008e1	86d1bf	275b9b	241deb	64749a
6.	f10de5	13d3f5	114b8b	5d374d	93cb88	79c7d5	2ffd72	ba0aae
7.	571121	382af3	41afe9	4f77bc	f06c83	b8ff56	75f097	9074ad
8.	5a7db4	61dd8f	3c7554	0d0012	1fd56e	95f8c7	31e9c4	d7221b
9.	c6b400	e024a6	668ccf	2e2de8	6876e4	f5c500	00f0a9	3b3aa7
a.	d1060b	871a28	01f978	376408	2ff592	d9140d	b1e939	9df4b0
b.	c703f5	32ce3a	30cd31	c070eb	36b419	5ff33f	b1c66c	7d70f9
c.	6d8d03	62803b	c248d4	14478c	2afb07	ffe78e	89b9fe	ca7e30
d.	df2be6	4bbaab	008ca8	a06fda	ce9ce7	048984	5a082b	a36d61
e.	558aa1	194177	20b6e1	50ce2b	927d48	d7256e	445e33	3cb757
f.	6b6c79	a58a9a	549b50	c58706	90755c	35e4e3	6b5290	38ca73

表1：v<8のときのS(u,v)

	.8	.9	.a	.b	.c	.d	.e	.f
0.	5190cf	ef324e	773892	6cfbe5	f4bf8d	8d8c31	d763da	06c80a
1.	f7b46b	ced55c	4d79fd	5f24d6	613c31	c3839a	2ddf8a	9a276b
2.	cc93ed	874422	a52ecb	238fee	e5ab6a	dd835f	d1a075	3d0a8f
3.	37df8b	b300d0	1334a0	d0bd86	45cbfa	73a616	0ffe39	3c48cb
4.	f45a0e	cb1bcd	289b06	cbbfea	21ad08	e1847f	3f7378	d56ced
5.	47dfdf	b96632	c3eb06	1b6472	bbf84c	26144e	49c2d0	4c324e
6.	7277da	7ba1b4	af1488	d8e836	af1486	5e6c37	ab6876	fe690b
7.	9a787b	c5b9bd	4b0c59	37d3ed	e4c3a7	939621	5edab1	f57d0b
8.	bed0c6	2bb5a8	7804b6	79a0ca	a41d80	2a4604	c311b7	1de3e5
9.	e6342b	302a0a	47373b	25f73e	3b26d5	69fe22	91ad36	d6a147
a.	e14ca8	e88ee9	110b2b	d4fa98	eed150	ca6dd8	932245	ef7592
b.	391810	7ce205	1fed33	f6d1de	9491c7	dea6a5	a442e1	54c8bb
c.	60c08f	0d61f8	e36801	df66d1	d8f939	2e52ca	ef0653	199479
d.	1e99f2	fbe724	246d18	b54e33	5cac0d	d1ab9d	fd7988	a4b0c4
e.	2b3bd0	0fb274	604318	9cac11	6cedc7	e771ae	0358ff	752a3a
f.	3fd1aa	a8dab4	0133d8	0320e0	790968	c76546	b993f6	c8ff3b

表 2：v≥8のときのS(u,v)

【図１】

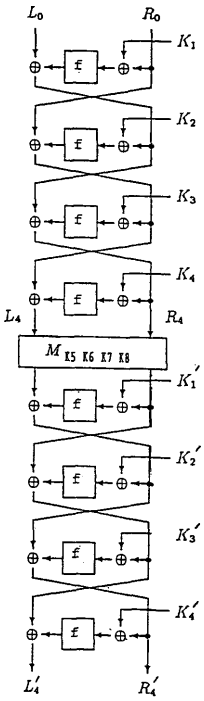

FIG. 1

フロントページの続き

審査官　青木　重徳

(56)参考文献　特開平０２－０２７３８９（ＪＰ，Ａ）
Serge VAUDENAY, “An Experiment on DES Statistical Cryptanalysis”, ECOLE NORMALE SUPERIEURE, １９９５年１１月, LIENS-95-29, ｐ．１－１０, ＵＲＬ, http://citeseer.ist.psu.edu/vaudenay95experiment.html
Serge VAUDENAY, “Provable security for block Ciphers by decorrelation”, ECOLE NORMALE SUPERIEURE, １９９８年　６月, LIENS-98-8, ｐ．１－２７, ＵＲＬ, http://citeseer.ist.psu.edu/8984.html
Serge Vaudenay, “Feistel Cipers with L2-Decorrelation”, Lecture Notes in Computer Science, １９９９年　５月２５日, Vol.1556, p.1-14, Selected Areas in Cryptography

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