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

特許第4708914号
(P4708914)

(45)発行日　平成23年6月22日(2011.6.22)　(24)登録日　平成23年3月25日(2011.3.25)

(51)Int.Cl.			F I		
H04L	9/08	(2006.01)	H04L	9/00	601B
H04L	9/14	(2006.01)	H04L	9/00	601E
			H04L	9/00	641

請求項の数 4　(全 21 頁)

(21)出願番号	特願2005-235212(P2005-235212)
(22)出願日	平成17年8月15日(2005.8.15)
(62)分割の表示	特願平8-39099の分割
原出願日	平成8年2月2日(1996.2.2)
(65)公開番号	特開2005-341625(P2005-341625A)
(43)公開日	平成17年12月8日(2005.12.8)
審査請求日	平成17年8月18日(2005.8.18)
審判番号	不服2009-20619(P2009-20619/J1)
審判請求日	平成21年10月26日(2009.10.26)

(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　100086841
弁理士　脇　篤夫
(74)代理人　100114122
弁理士　鈴木　伸夫
(72)発明者　江成　正彦
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

合議体
審判長　吉岡　浩
審判官　宮司　卓佳
審判官　石井　茂和

最終頁に続く

(54)【発明の名称】解読化方法

(57)【特許請求の範囲】

【請求項１】

暗号化されたデータと、該データの種類を示す情報と該データが暗号化されているか否かを示す情報とを含む送信データを受信し、前記暗号化されたデータを解読するようにした解読化方法であって、

メモリ手段に一つの前記データの種類を示す情報に対して割り当てられている、互いに異なる複数の鍵情報を書き込む書き込みステップと、

受信した前記送信データの中に含まれるデータの種類を示す情報とデータが暗号化されているか否かを示す情報とにより指示された鍵情報を、前記メモリ手段から読み出す読み出しステップと、

該読み出しステップで読み出された鍵情報に基づいて前記暗号化されたデータを解読する解読ステップとを含んでおり、

前記書き込みステップの実行期間と、前記読み出しステップの実行期間とが重なった場合であって、書き込みアドレスと読み出しアドレスとが一致している場合は、前記書き込みステップの実行を禁止するようにした

解読化方法。

【請求項２】

暗号化されたデータと、該データの種類を示す情報と該データが暗号化されているか否かを示す情報とを含む送信データを受信し、前記暗号化されたデータを解読するようにした解読化方法であって、

一つの前記データの種類を示す情報に対して割り当てられている、互いに異なる複数の鍵情報を記憶しているメモリ手段から、前記受信した送信データ中のデータの種類を示す情報とデータが暗号化されているか否かを示す情報による指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、

該読み出しステップで読み出された前記鍵情報に基づいて前記暗号化されたデータを解読する解読ステップとからなり、

前記読み出しステップでは、前記指示データにより前記データの種類を示す情報の鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報をサーチしており、前記指示データのデータの種類を示す情報に該当する前記鍵情報を読み出すための鍵アドレスが、前記鍵テーブルに２つ以上存在する場合は、最も小さい鍵アドレスを選択する

解読化方法。

【請求項３】

暗号化されたデータと、該データの種類を示す情報と該データが暗号化されているか否かを示す情報とを含む送信データを受信し、前記暗号化されたデータを解読するようにした解読化方法であって、

一つの前記データの種類を示す情報に対して割り当てられている、互いに異なる複数の鍵情報を記憶しているメモリ手段から、前記受信した送信データ中のデータの種類を示す情報とデータが暗号化されているか否かを示す情報による指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、

該読み出しステップで読み出された前記鍵情報に基づいて前記暗号化されたデータを解読する解読ステップとからなり、

前記読み出しステップでは、前記指示データにより前記データの種類を示す情報の鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、前記指示データのデータの種類を示す情報により参照される鍵アドレスが前記データの種類を示す情報の鍵テーブルに存在していない場合は、鍵アドレスのサーチを行わないようにした

解読化方法。

【請求項４】

暗号化されたデータと、該データの種類を示す情報と該データが暗号化されているか否かを示す情報とを含む送信データを受信し、前記暗号化されたデータを解読するようにした解読化方法であって、

初期設定を行う初期ステップと、

メモリ手段に一つの前記データの種類を示す情報に対して割り当てられている、互いに異なる複数の鍵情報を書き込む書き込みステップと、

前記メモリ手段から、前記受信した送信データ中のデータの種類を示す情報とデータが暗号化されているか否かを示す情報による指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、

該読み出しステップで読み出された前記鍵情報に基づいて前記暗号化されたデータを解読する解読ステップとからなり、

前記読み出しステップでは、前記指示データによりデータの種類を示す情報の鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、初期化期間内においては、サーチした結果、該当する鍵アドレスがないように、前記初期ステップにおいて初期処理される

解読化方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、暗号化されることによりスクランブルされた送信データを受けて、解読するようにした解読化方法に関する。

【背景技術】

【０００２】
通信における情報を秘匿するために、送信情報を暗号化し、暗号化された送信情報を受信して解読することにより、元の情報を得るようにした暗号化・解読化方式が従来から知られている。このような暗号化・解読化方式としては、米国における標準方式であるＤＥＳ（Data Encryption Standard）等の暗号アルゴリズムが知られている。
【０００３】
ところで、暗号アルゴリズムには多種・多様なものがあり、より安全性・高速性に優れた方式が開発されている。この一例として、米国特許第４，９８２，４２９号明細書、米国特許第５，１０３，４７９号明細書、および特開平１－２７６１８９号公報等に記載されている暗号方式（MULTI2方式）が知られている。
また、国際標準化機構（ＩＳＯ）においてもＩＳＯ９９７９／０００９として登録された暗号化方式や、ＩＳＯ／ＩＥＣ１０１１６として登録された暗号化利用モードがある。
【０００４】
上記MULTI2方式の暗号化方式においては、入力データサイズが６４ビット、出力データサイズが６４ビットとされており、２５６ビットサイズのシステム鍵と６４ビットのデータ鍵から、暗号化を行うために必要な２５６ビットサイズのワーク鍵が生成されている。さらに、暗号化段数は正の整数段とされている。
このMULTI2方式における暗号化アルゴリズムの概略構成を図１７に示す。MULTI2方式は、図１７に示すように６４ビットのデータ鍵Ｋｓに２５６ビットのシステム鍵Ｊを用いて暗号アルゴリズムの演算を施すことにより２５６ビットのワーク鍵Ｋｗを生成する。この暗号アルゴリズムの演算は暗号アルゴリズム実行手段Ｃにより実行される。生成されたワーク鍵Ｋｗは、暗号アルゴリズム実行手段Ｆに供給されて入力された６４ビットの平文が暗号化される。なお、暗号アルゴリズム実行手段Ｃと暗号アルゴリズム実行手段Ｆとで実行される暗号アルゴリズムは、同一の暗号アルゴリズムである。
【０００５】
このような暗号化がMULTI2方式の基本的な暗号化アルゴリズムであるが、これでは予め文字、あるいは単語が出現する頻度の分布を統計処理しておき、入手した暗号化文の文字列パターンの頻度分布とのマッチングを取ることにより、平文が推定されてしまうおそれがある。
そこで、暗号化された６４ビットの暗号ブロックと、次に入力される６４ビットの入力データとの排他的論理和を演算して暗号文を作成する手法がある。この手法を行って暗号化するモードをＣＢＣ（Cipher Block Chaining)モードとよんでいる。前記した暗号アルゴリズム実行手段Ｆにおいては、このようなＣＢＣモードの暗号アルゴリズムが実行されている。
【０００６】
また、例えばパケット通信のように通信を行うデータの単位が予め決められている通信方式があるが、６４ビットを１ブロックとするようなブロック暗号化方式では、１ブロックのビット数で割り切れないデータ単位が入力された場合に、データが余ってしまうようになる。そこで、その端数処理をＯＦＢ（Output Feedback ）モードで処理するようにしている。
このＯＦＢモードでは、データの端数部分が暗号アルゴリズム実行手段Ｇに供給され、乱数を使用して暗号化される。この乱数は、ワーク鍵Ｋｗを用いて暗号アルゴリズム実行手段Ｇにより生成されている。これにより、６４ビットを１ブロックとする暗号文を得ることができるようになる。なお、ＣＢＣモードおよびＯＦＢモードは暗号化利用モードと呼ばれる。
【０００７】
また、MULTI2方式における解読化アルゴリズムの概略構成を図１８に示す。図１８に示すように、６４ビットのデータ鍵Ｋｓに２５６ビットのシステム鍵Ｊを用いて暗号アルゴリズムの演算を施すことにより２５６ビットのワーク鍵Ｋｗを生成する。この暗号アルゴリズムの演算は暗号アルゴリズム実行手段ｃにより実行される。生成されたワーク鍵Ｋｗ

は、解読アルゴリズム実行手段ｆに供給されて入力された６４ビットの暗号文が解読化される。

なお、ＯＦＢモードで暗号化されている暗号文は、暗号アルゴリズム実行手段ｇに供給され、ワーク鍵Ｋｗを用いて暗号アルゴリズム実行手段ｇにより生成した乱数を使用することにより解読化される。これにより、１ブロック６４ビットの暗号文を解読化して６４ビットの平文を得ることができる。また、ＣＢＣモードとされている場合は、解読アルゴリズム実行手段ｆがＣＢＣモードの解読アルゴリズムを実行するようにされる。

【０００８】

ここで、暗号化利用モードの説明を図１９を参照しながら行うが、図１９（ａ）にＣＢＣモードの暗号化・解読化の概略構成を示し、図１９（ｂ）にＯＦＢモードの暗号化・解読化の概略構成を示している。

ＣＢＣモードでは、図１９（ａ）に示すようにｉ番目の平文ブロックM(i) は、排他的論理和回路１０１に入力され、レジスタ（ＲＥＧ）１０３により遅延された１ブロック前の暗号文ブロックC(i-1) との排他的論理和が演算される。演算されたデータは暗号アルゴリズム実行手段１０２において、データ鍵Ｋｓに基づいて生成されたワーク鍵により暗号化される。この暗号化されたｉ番目の暗号文ブロックC(i) は、

C(i) ＝ＥＫｓ（M(i) ．ＥＯＲ．C(i-1) ）

と表せる。ただし、ＥＫｓ（ｍ）はｍをＫｓで暗号化することを意味しており、ＥＯＲは排他的論理和の演算を行うことを示している。

【０００９】

そして、この暗号文ブロックC(i) は送信され、受信側において受信されることになる。受信された暗号文ブロックC(i) は、解読アルゴリズム実行手段１１１においてデータ鍵Ｋｓに基づいて生成されたワーク鍵を用いて解読され、排他的論和回路１１３に供給される。この排他的論理和回路１１３にはレジスタ（ＲＥＧ）１１２において遅延された、１ブロック前の暗号文ブロックC(i-1) が入力されて、両者の排他的論和が演算される。この時、送信側と受信側のデータ鍵Ｋｓは等しく、これにより、排他的論理和回路１１３からｉ番目の平文ブロックM(i) が解読される。ｉ番目の平文ブロックM(i) は次のように表せる。

M(i) ＝ＤＫｓ（C(i) ．ＥＯＲ．C(i-1) ）

ただし、ＤＫｓ（ｃ）はＫｓでｃを解読化することを示している。

【００１０】

また、ＯＦＢモード時では、ｉ番目の平文ブロックM(i) は排他的論理和回路１０５に供給される。この排他的論理和回路１０５には、データ鍵Ｋｓに基づいて生成されたワーク鍵により乱数化された暗号アルゴリズム実行手段１０４の出力が供給されている。なお、暗号アルゴリズム実行手段１０４の出力は、レジスタ１０３により１ブロック遅延されて暗号アルゴリズム実行手段１０４に戻されている。これにより、排他的論理和回路１０５からは乱数により暗号化された暗号文ブロックC(i) が出力される。

【００１１】

そして、この暗号文ブロックC(i) は送信され、受信側において受信されることになる。受信された暗号文ブロックC(i) は、排他的論和回路１１４に供給される。この排他的論和回路１１４には、暗号アルゴリズム実行手段１１５においてデータ鍵Ｋｓに基づいて生成されたワーク鍵を用いて乱数化された出力が供給されている。この暗号アルゴリズム実行手段１１５の出力は、レジスタ（ＲＥＧ）１１２において１ブロック遅延されて暗号アルゴリズム実行手段１１５に戻されている。この場合、排他的論理和回路１１４に供給される乱数は、排他的論理和回路１０５に供給される乱数と等しく、これにより、排他的論理和回路１１４からｉ番目の平文ブロックM(i) が得られる。

【００１２】

以上説明した暗号化利用モードを有する暗号化・解読化方式の概略構成を図２０に示す。

この図において、送信側にはスクランブラ１００が備えられており、スクランブラ１０

０により入力データがスクランブルされて送信されている。このスクランブルされた送信データは、空間等の伝送路を伝播されて受信側で受信される。受信側には、デスクランブラ１１０が備えられており、このデスクランブラ１１０によりスクランブルされた送信データがデスクランブルされて、元のデータに戻され出力されるようになる。

【００１３】

スクランブラ１００は、入力された入力データ（平文）を暗号化する暗号アルゴリズム実行手段であるEncryptor １０２と、レジスタ１０３と、排他的論理和回路（ＥＸ－ＯＲ）１０１からなるＣＢＣモード暗号化部と、暗号アルゴリズム実行手段であるEncryptor １０４と、排他的論理和回路（ＥＸ－ＯＲ）１０５からなるＯＦＢモード暗号化部から構成されている。なお、データ鍵とシステム鍵からワーク鍵を生成するEncryptor １０６もスクランブラ１００内に備えられている。生成されたワーク鍵はEncryptor １０２，１０４に供給される。

ところで、Encryptor １０２、Encryptor １０４、Encryptor １０６は同一構成とされているので、１つのEncryptor により３つのEncryptor を兼用することができる。ＣＢＣモード暗号化部およびＯＦＢモード暗号化部の動作は前述したとおりであるので、ここでは省略する。

【００１４】

また、デスクランブラ１１０は、入力された受信データ（暗号文）を解読化する解読アルゴリズム実行手段であるDecryptor １１１と、レジスタ１１２と、排他的論理和回路（ＥＸ－ＯＲ）１１３からなるＣＢＣモード解読化部と、暗号アルゴリズム実行手段であるEncryptor １１５と、排他的論理和回路（ＥＸ－ＯＲ）１１４からなるＯＦＢモード解読化部から構成されている。なお、データ鍵とシステム鍵からワーク鍵を生成するEncryptor １１６もデスクランブラ１１０内に備えられている。生成されたワーク鍵はDecryptor １１１と、Encryptor １１５に供給される。

なお、Encryptor １１５、Encryptor １１６は同一構成とされているので、１つのEncryptor により２つのEncryptor を兼用することができる。また、ＣＢＣモード解読化部およびＯＦＢモード解読化部の動作は前述したとおりであるので、ここでは省略する。

【発明の開示】

【発明が解決しようとする課題】

【００１５】

ところで、上述したMULTI2方式のような高度な暗号化・解読化方式を、ソフトウェアで実現して使用する場合には、現在の演算手段の演算速度ではリアルタイム処理を行うことができない。すなわち、例えば衛星ディジタルテレビジョン放送等に適用した場合は、画像や音声を途切らせないでリアルタイム再生する必要があることから、受信側における解読化処理は高速な処理の可能なハードウェアで行わなければならないことになる。

しかしながら、受信側においてリアルタイムで全ての解読処理を完了することができるハードウェアは、複雑な処理を行うことから大型なものになるという問題点があった。

【００１６】

そこで、本発明は高度な暗号化方式でスクランブルされたデータを、受信側においてリアルタイムで全ての解読処理を小型かつ安価なハードウェアを用いて完了することができる解読化方法を提供することを目的としている。

【課題を解決するための手段】

【００１７】

上記目的を達成するために、本発明の解読化方法は、暗号化されたデータと、該データの種類を示す情報と該データが暗号化されているか否かを示す情報とを含む送信データを受信し、前記暗号化されたデータを解読するようにした解読化方法であって、メモリ手段に一つの前記データの種類を示す情報に対して割り当てられている、互いに異なる複数の鍵情報を書き込む書き込みステップと、受信した前記送信データの中に含まれるデータの種類を示す情報とデータが暗号化されているか否かを示す情報とにより指示された鍵情報を、前記メモリ手段から読み出す読み出しステップと、該読み出しステップで読み出され

た鍵情報に基づいて前記暗号化されたデータを解読する解読ステップとを含んでおり、前記書き込みステップの実行期間と、前記読み出しステップの実行期間とが重なった場合であって、書き込みアドレスと読み出しアドレスとが一致している場合は、前記書き込みステップの実行を禁止するようにしている。

【００１８】

また、上記目的を達成するために、本発明の他の解読化方法は、前記複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された前記鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報をサーチしており、前記指示データに該当する前記鍵情報を読み出すための鍵アドレスが、前記鍵テーブルに２つ以上存在する場合は、最も少ない鍵アドレスを選択するようにしている。

【００１９】

さらにまた、上記目的を達成するために、本発明のさらに他の解読化方法は、前記複数の鍵情報を記憶しているメモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された前記鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データによりテーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、前記指示データにより参照される鍵アドレスが前記テーブルに存在していない場合は、鍵アドレスのサーチを行わないようにしている。

【００２０】

さらにまた、上記目的を達成するために、本発明のさらに他の解読化方法は、初期設定を行う初期ステップと、メモリ手段に複数の鍵情報を書き込む書き込みステップと、前記メモリ手段から、入力データ中の指示データに基づいていずれかの鍵情報を読み出す読み出しステップと、該読み出しステップで読み出された前記鍵情報に基づいて前記入力データを解読する解読ステップとからなり、前記読み出しステップでは、前記指示データにより鍵テーブルを参照することにより、前記メモリ手段から読み出す鍵情報の鍵アドレスをサーチしており、初期化期間内においては、サーチした結果、該当する鍵アドレスがないように、前記初期ステップにおいて初期処理されるようにしている。

【００２１】

このような本発明の解読化方法によれば、解読処理を行うために必要な鍵情報が格納されているメモリ手段の読み出しおよび書き込み制御を適切に行うことができるため、メモリ制御を容易に行うことのできる解読化方法とすることができる。したがって、受信側においてリアルタイムで全ての解読処理を行うことのできる解読化方法とすることができる。

また、このような解読化方法を実行する解読手段を備える電子機器においては、メモリ制御手段の構成を簡単化することができるので、電子機器を小型かつ安価に提供することができるようになる。

【発明の効果】

【００２２】

本発明は以上のように構成されているので、解読処理を行うための鍵情報が格納されているメモリ手段の読み出しおよび書き込み制御を適切に行うことができ、メモリ制御を容易に行うことのできる解読化方法とすることができる。したがって、受信側においてリアルタイムで全ての解読処理を実行することのできる解読化方法とすることができる。

また、このような解読化方法を実行する解読手段を備える電子機器におけるメモリ制御手段の構成を小さくすることができるので、電子機器を小型かつ安価に提供することができるようになる。

【発明を実施するための最良の形態】

【００２３】
本発明の解読化方法の実施の形態である解読装置の構成例を示すブロック図を図１に示す。
この図において、解読装置ＡはＣＢＣモードの解読化部Ｂと、ＯＦＢモードの解読化部Ｃと、パケットＩＤ（ＰＩＤ）テーブルを備えるデータ鍵間接検索器（ＩＤＴ）１６と、入力データの暗号化に使用されたデータ鍵を解読化部Ｂ，Ｃに渡すデュアルポートメモリ（ＤＰＭＥＭ）１７と、ＤＰＭＥＭ１７の書き込みアドレスと読み出しアドレスとを比較する比較器（ＣＯＭＰ）１１から構成されている。
【００２４】
解読装置Ａの端子１には受信データである暗号文データが入力され、この暗号文データは切換手段３に入力される。この時、ＣＢＣモードとされた場合は、切換手段３、および切換手段４が共に端子ａ側に切り換えられ、入力された暗号文データはＣＢＣモード解読化部Ｂに供給されて解読化されて出力される。また、ＯＦＢモードとされた時は、切換手段３、および切換手段４は共に端子ｂ側に切り換えられて、暗号文データはＯＦＢモード解読化部Ｃ側に入力される。そして、入力された暗号文データは、ＯＦＢモード解読化部Ｃにおいて解読化されて平文データが出力される。
ＣＢＣモード解読化部Ｂ、およびＯＦＢモード解読化部Ｃにおいて、入力された暗号文データは解読処理されて平文データとされるが、その解読化アルゴリズムは、前記図１８に示す解読化アルゴリズムと同様である。なお、この場合解読処理に必要なデータ鍵はＤＰＭＥＭ１７から供給される。
【００２５】
また、端子１に入力される受信データのフォーマットは、例えばＩＳＯ／ＩＥＣ１３８１８として規定されているトランスポートストリーム（以下、ＴＳと記す。）とされる。ＴＳは１８８バイトのパケット構造とされており、通常４バイトのヘッダのあとに１８４バイトのペイロードが続くようにされている。さらに、伝送エラーに対するエラー訂正のためにパリティーを付加することから、１６バイトのパリティ用のダミー期間が付加されて、これらの繰返しのストリームとされる。
【００２６】
このＴＳのヘッダにはパケットが映像データで構成されているのか、音声データで構成されているのか、あるいは他のデータで構成されているのかを示すＰＩＤ情報や、暗号化されているＴＳなのか否かを示すＴＳＣ（Transport Scrambling Control）フラグ等が含まれており、図示しない解析部においてヘッダを解析することにより、これらのパケットの属性が解釈されている。
この時、ＴＳＣにより暗号化されていないパケットと解釈された場合は、解読化処理を施すことなくＴＳを遅延部を通してそのまま出力するようにする。この遅延部の遅延時間は、解読装置Ａが解読処理に要する時間と等しくされる。
【００２７】
また、１つのＰＩＤに対しては、データ鍵Ｋｓｅ（Ks_even ）とデータ鍵Ｋｓｏ（Ks_odd）との２つが割り当てられている。これは、データ鍵が数秒ないし数十秒毎に更新されるため、更新に間に合うようにデータ鍵の書き替えを行う必要があること、および、解読処理時にはデータ鍵Ｋｓｅとデータ鍵Ｋｓｏはその一方しか使用されず、使用されていないデータ鍵を更新可能として、そのデータ鍵を更新することができるからである。すなわち、データ鍵を更新する更新制御を容易に行うためである。
なお、前記解析部は図示しないが、例えばＣＢＣモード解読化部Ｂに備えられており、ＣＢＣモード解読化部Ｂから１５ビットのデータサイズのＰＩＤ／ＴＳＣ情報がＩＤＴ１６に送られている。
【００２８】
ＩＤＴ１６においては、受けた１３ビットサイズのＰＩＤ情報を用いてＰＩＤテーブルを参照し、ＰＩＤから読出されたアドレス情報と、２ビットサイズのＴＳＣ情報とを組み合わせてＤＰＭＥＭ１７の読出アドレスＲＡ（データサイズは９ビット）を生成している

。生成された読出アドレスＲＡはＡＤ端子からＤＰＭＥＭ１７のＲＡ端子に供給され、この時にリードイネーブル（ＲＥ）信号がアクティブ状態になるよう制御して、読出アドレスＲＡに該当するデータ鍵をＤＰＭＥＭ１７から読み出している。読み出されたデータ鍵は、入力されたＴＳの暗号化時に使用されたデータ鍵と等しく、端子ＲＤから出力されてDecryptor ５に供給される。

Decryptor ５は、供給されたデータ鍵と、システム毎に異なるシステム鍵からワーク鍵を生成して、生成されたワーク鍵に基づいて入力された暗号文データの解読処理を行う。

なお、システム鍵は予めＣＰＵ１０から解読装置Ａに渡されている。

【００２９】

また、ＣＰＵ１０は解読化部Ｂ，Ｃで必要とされる更新されるデータ鍵を、解読処理において必要になる前にＤＰＭＥＭ１７に書き込むようにされており、書き込む場合には、ライトイネーブル（ＷＥ）をアクティブ状態とすると共に、９ビットのデータサイズの書込アドレス（ＷＡ）と、８ビットのデータサイズのデータ鍵の書込データ（ＷＤ）をＤＰＭＥＭ１７に供給している。この場合、ＣＰＵ１０にはＲＯＭやＲＡＭが備えられており、これらのメモリに解読装置Ａに与えるデータが書き込まれている。

なお、ＤＰＭＥＭ１７にデータ鍵情報を書き込む時に、書込アドレスと同一の読出アドレスでＩＤＴ１６がデータ鍵を読み出すようになった場合は、読み出されたデータが不定となることから、この時はＤＰＭＥＭ１７への書き込みを禁止している。

【００３０】

このために、比較器１１とアンドゲート１２、オアゲート１３，１４、インバータ１５が設けられている。この動作を説明すると、ＩＤＴ１６から出力される読出アドレスＲＡと、ＣＰＵ１０から出力される書込アドレスＷＡとが比較器１１のＰ端子およびＱ端子にそれぞれ入力される。そして、Ｐ端子の入力データとＱ端子の入力データとが一致（Ｐ＝Ｑ）した時に、比較器１１からハイレベル信号が出力される。この時、反転ＲＥがローレベルでアクティブ状態になっていると、インバータ１５からハイレベルが出力されるので、結局のところアンドゲート１２からハイレベルが出力されることになる。このアンドゲート１２の出力はオアゲート１３に入力されて、オアゲート１３の出力がハイレベルとされるので、反転ＷＥは非アクティブ状態となり、ＤＰＭＥＭ１７への書き込みが禁止されることになる。

【００３１】

次に、ＣＰＵ１０から見た解読装置Ａのレジスタのメモリ空間を図３に示すが、９ビットのデータサイズのアドレス（ＨＡＤＤ）の上位６ビットが”０００１００”とされた６４bit ×１のエリアが、ＣＢＣモードの初期値テーブル（CBC Initial value table ）に割り当てられている。この初期値テーブルに記憶される初期値は、電源投入時等に実行される初期処理時に、ＣＢＣモード解読化部Ｂに供給されてレジスタ６にセットされる。

また、アドレス（ＨＡＤＤ）の上位４ビットが”００１０”とされた２５６bit ×１のエリアが、システム鍵テーブル（SYSTEM_Key table）に割り当てられている。このシステム鍵はシステム毎に異なるものとされるが、１つのシステムでは固定された鍵である。

【００３２】

さらに、アドレス（ＨＡＤＤ）の上位４ビットが”０１００”とされた１３bit ×１２のエリアが、ＰＩＤテーブル（PID value table ）に割り当てられている。このＰＩＤテーブルはパケット化されたデータ種類を示す情報であり、１チャンネルでは最大１２種類とされデータ種類毎に異なるＰＩＤとされる。

なお、システム鍵テーブルは所定のレジスタに、ＰＩＤテーブルの情報はＩＤＴ１６に、初期処理時にＣＰＵ１０から書き込まれる。

【００３３】

さらにまた、アドレス（ＨＡＤＤ）の上位２ビットが”１０”とされた６４bit ×１２のエリアが、データ鍵Ｋｓｅテーブル（Ks_even value table ）に割り当てられ、アドレス（ＨＡＤＤ）の上位２ビットが”１１”とされた６４bit ×１２のエリアが、データ鍵Ｋｓｏテーブル（Ks_odd value table）に割り当てられている。このデータ鍵Ｋｓｅテー

ブル、およびデータ鍵Ｋｓｏテーブルの情報はＤＰＭＥＭ１７に書き込まれるが、数秒ないし数１０秒の所定タイミング毎に、ＣＰＵ１０により更新されている。

【００３４】

次に、ＩＤＴ１６の詳細構成の一例を示すブロック図を図２に示す。この図において、４１～５２はそれぞれ１３ビット幅を有する１２個のフリップフロップ（ＤＦ０～ＤＦ１１）であり、６１～７２はそれぞれ１３ビット幅を有する１２個の比較器（ＣＰ０～ＣＰ１１）である。また、４０はアドレスデコーダ（ＡＤＥＣ）であり、９ビットのデータサイズのアドレス情報が入力されると、１２本の出力のうちの１本だけに、ＤＦ０～ＤＦ１１のいずれかをアクティブにする出力が出される。さらに、３９はプライオリティ・エンコーダ（ＰＥ）であり、ＴＳＣ情報と組み合わされるアドレス情報ＴＭＰ０～ＴＭＰ３が出力される。

【００３５】

このＩＤＴ１６において、端子３０にはＣＰＵ１０から出力された８ビットのデータサイズのＰＩＤ情報が、１３ビットのデータサイズに展開されて入力される。また、端子３２にはＣＰＵ１０から出力された９ビットのデータサイズのアドレス情報が入力され、ＡＤＥＣ４０においてデコードされる。このデコード出力はストローブ信号としてＤＦ０～ＤＦ１１に入力され、ＤＦ０～ＤＦ１１のうちのいずれか１つのみがラッチ可能とされる。

この時、入力された１３ビットのデータサイズのＰＩＤ情報は、１２個のＤＦ０～ＤＦ１１に共通に入力されており、ＡＤＥＣ４０のデコード出力によりＤＦ０～ＤＦ１１が順次１つづつ選択されて、供給されているＰＩＤ情報がラッチされる。このようにして、ＤＦ０～ＤＦ１１に最大１２種類のＰＩＤ情報をラッチすることができ、ＤＦ０～ＤＦ１１によりＰＩＤテーブルが構成されるようになる。

【００３６】

また、端子３３には１３ビットのデータサイズのＰＩＤ情報が入力され、比較器ＣＰ０～ＣＰ１１のＢ入力端子に共通に入力される。このＰＩＤ情報は、ＣＢＣモード解読化部Ｂにおいて、受信データであるパケットのヘッダから抽出されてＩＤＴ１６に供給された１５ビットのデータサイズのＰＩＤ／ＴＳＣ情報のうちの１３ビットのデータサイズのＰＩＤ情報である。

この場合、比較器ＣＰ０～ＣＰ１１にはＤＦ０～ＤＦ１１よりのＰＩＤ情報がそれぞれＡ端子に入力されており、端子３３より入力されたＰＩＤ情報と一致するＰＩＤ情報がＡ端子に入力されている比較器から一致信号が出力される。従って、ＰＩＤテーブルにあるＰＩＤ情報が端子３３から入力されると、ＣＰ０～ＣＰ１１のいずれか１つから一致信号が出力される。

【００３７】

このＣＰ０～ＣＰ１１の論理表を図４に示すが、ＣＰ０～ＣＰ１１のＡ端子およびＢ端子にオール”１”が入力されていない時は、一致した時（Ａ＝Ｂ）に”１”信号が出力される。また、一致してない時は”０”が出力される。

さらに、ＰＥ３９の論理表を図５に示す。このＰＥ３９の論理表において、例えば、ＣＰ２から”１”信号が出力された時は、ＰＥ３９からは”００１０”の４ビット（ＴＭＰ０～ＴＭＰ３）のデータが出力される。このＰＥ３９では図５に示すように、Ｄ０が最も優先された入力とされ、Ｄ１１が最下位の優先度の入力とされる。

ところで、ＤＦ０～ＤＦ１１に設定されたＰＩＤテーブルのＰＩＤ情報と一致しないＰＩＤ情報が入力されることがあるが、この場合にはＰＥにオール”０”が入力されることになる。この時、ＰＥのＮＫ出力が”１”となる。なお、ＮＫ出力が”１”の時には、後述するデータ鍵の鍵テーブルを読みに行くことが禁止される。

【００３８】

また、電源投入時等の初期処理時には端子３１のレベルが”０”となり、ＤＦ０～ＤＦ１１が全てプリセットされることから、ＤＦ０～ＤＦ１１から”１”が出力される。従って、この場合はＣＰ０～ＣＰ１１からオール”０”が出力されることになり、上述のよう

にデータ鍵の鍵テーブルを読みに行くことが禁止される。
なお、鍵テーブルを読みに行くことが禁止されている時は、鍵テーブルが使用されないので、この鍵テーブルにＣＰＵ１０からデータ鍵を書き込むことができる。すなわち、データ鍵の初期設定をすることができる。また、端子３１はパワーオンリセット端子であり、所定時間後にそのレベルは”１”に復帰する。

【００３９】
次に、受信データ中のＰＩＤ情報とＴＳＣ情報により、データ鍵の鍵テーブルをサーチする動作を図６を参照しながら説明する。
図６に示すステップＳ２０にて、受信データのヘッダから１３ビットのデータサイズのＰＩＤ情報として”ＰＩＤ－Ｆ”が、２ビットのデータサイズのＴＳＣ情報として”１０”が入力されたとする。この”１０”は、データ鍵Ｋｓｅでスクランブルされていることを示しているものとする。
次いで、ステップＳ２１にてこのＰＩＤ情報はＣＰ０～ＣＰ１１のＡ端子に入力されて、ＤＦ０～ＤＦ１１に格納されているＰＩＤ情報と比較される。この結果、ＣＰ５から”１”信号が出力され、ＰＥ３９のＴＥＭＰ３～ＴＥＭＰ０の出力ＴＭＰ［３．．０］が、”０１０１”となる。すなわち、”ＰＩＤ－Ｆ”がサーチされる。この、ＰＥ３９の出力である”０１０１”は、アドレスＨＡＤＤの第４ビットから第１ビットとされる。

【００４０】
次いで、ステップＳ２２にてＴＳＣ情報ＴＳＣ［１．．０］の２ビット（”１０”）がＨＡＤＤの第５ビット，第６ビットとされて、６ビットサイズのアドレスＨＡＤＤ［８．．３］が生成される。したがって、アドレスＨＡＤＤ［８．．３］は”１００１０１”となる。
そして、ステップＳ２３にて、生成された”１００１０１”のアドレスＨＡＤＤ［８．．３］により鍵テーブルを参照すると、Ks_even Table から６４ビット幅のデータ鍵Ｋｓｅ－Ｆが得られるようになる。そして、得られたデータ鍵Ｋｓｅ－Ｆに基づいて入力された暗号文の解読が解読装置Ａで実行される。
このように、本発明の解読化方法では間接的にデータ鍵を検索する間接検索方法としているので、ＰＩＤ情報に対応する１３ビット幅のテーブルを用意することなく、６ビット幅の鍵テーブルを用意すればよいのでメモリの容量を削減して小型化することができる。

【００４１】
なお、このように解読中では鍵テーブルのKs_even Table , Ks_odd Table　の一方しか使用されないため、使用していない鍵テーブルのデータ鍵を更新することができる。これは、前述したようにＣＰＵ１０が実行するが、ＣＰＵ１０のタイミングと解読装置Ａのタイミングとは非同期で動作するので、ＣＰＵ１０は非同期で更新するデータ鍵を書き込みに行く。
このため、鍵テーブルの格納されるデュアルポートメモリ１７においては、前述したように書き込みと読み出しが同時に同一アドレスで行われる場合が生じるのである。この時のタイミング例を図９に示すが、図示するタイミングで８クロック幅の反転ＲＥが発生した場合には、この８クロック期間においては、前述したようにＣＰＵ１０からの同一アドレスの書き込みが禁止される。

【００４２】
ところで、何らかの原因によりＰＩＤテーブルに同一のＰＩＤが格納されている場合が生じる。例えば、１チャンネル当り最大１２種類のＰＩＤ情報とされるが、１２種類のＰＩＤを必要としない場合はＰＩＤテーブルには１２種類のＰＩＤを書き込む必要はなく、必要する種類のＰＩＤだけを書き込むことになる。すると、書き込まれていないＰＩＤの欄のデータが偶然ＰＩＤと同じデータになることがある。
このような場合には、誤ったデータ鍵を読み出して解読できなくなってしまう恐れがあるので、本発明においては次のようにしてこれを防止している。

【００４３】
図７に示すようにＰＩＤテーブルに複数の”ＰＩＤ－Ｆ”がある場合は、その内のアド

レスＨＡＤＤが小さい方を優先するようにしている。これは、アドレスＨＡＤＤの小さい方からＰＩＤテーブルに書き込みに行くので、アドレスＨＡＤＤの大きい方が誤っている確率が高いからである。

これにより、ＰＩＤ情報として”ＰＩＤ－Ｆ”が入力されると、アドレスＨＡＤＤ［８．．１］として”０１０００１０１”が得られるようになり、前述した処理と同様の処理が行われ、Ks_even Table から６４ビット幅のデータ鍵Ｋｓｅ－Ｆが得られるようになる。

【００４４】

また、入力データのヘッダからのＰＩＤ情報により、ＰＩＤテーブルを参照しても該当するＰＩＤがない場合がある。例えば、図８に示すようにＰＩＤ情報として”ＰＩＤ－Ｆ”が入力されても、ＰＩＤテーブル中には”ＰＩＤ－Ｆ”に該当するＰＩＤがない。

このような場合には、鍵テーブルのKs_even Table , Ks_odd Table のいずれも読みに行くことなく、データ鍵を読み出さない。この場合は、入力データであるＴＳはそのまま出力することになる。

【００４５】

次に、図１に示す解読装置Ａの解読フローを図１０に示すが、トランスポートストリーム（ＴＳ）が入力されると、ステップＳ１０にてスクランブルされているか否かが判定される。この判定はスクランブルされたことを示すヘッダ中のフラグが立っているか否かを検出することにより判定される。この場合、フラグが立っている場合はスクランブルｏｎと判定されて、ステップＳ１１に進み、ここで所望のフラグ等の書き換えが行われる。次いで、ステップＳ１２にてヘッダからＰＩＤ情報が抽出されて、鍵テーブルが参照される。この場合の鍵テーブルは、鍵テーブル処理のステップＳ１６にてホストインターフェース処理が行われて、ステップＳ１７にて書き込まれた鍵テーブルが参照される。

なお、ステップＳ１２の処理は、前記図６に示す処理である。

【００４６】

以上のステップＳ１０ないしステップＳ１２の処理がヘッダコントロール処理である。なお、ステップＳ１０にてスクランブルｏｆｆと判定された場合には、ＴＳはそのまま出力される。

次いで、ステップＳ１３にてＣＢＣモードの解読処理が行われ、ステップＳ１４にて解読処理される暗号文が６４ビットの整数倍か否かが判定される。暗号文に端数がありＮｏと判定された場合は、ステップＳ１５にて端数部分についてＯＦＢモードの解読処理が行われ、解読処理された平文が出力される。また、ステップＳ１４にて６４ビットの整数倍と判定された場合は、解読処理された６４ビットの平文が出力されるようになる。

【００４７】

ところで、図１０に示すような解読フローの実行は、前記図１８に示す解読化アルゴリズムにより実行されている。図１８に示す解読化アルゴリズムは前述したとおりであるのでここでは省略するが、解読化アルゴリズム中の解読アルゴリズムおよび暗号アルゴリズムを実行する構成の詳細を図１１ないし図１５を参照して説明する。

図１１は暗号アルゴリズムを実行する暗号処理の構成を示す。図１１において、６４ビット幅の入力データは、上位３２ビットのデータと下位の３２ビットのデータに分割されて最初の暗号８段に入力される。この暗号８段は、関数の演算を行う４段の演算段が２回繰り返された構成とされる。そして、入力された上位３２ビットのデータと下位の３２ビットのデータに、演算段２０の初段において関数π１の演算が施される。ついで、第２段において初段の出力に関数π２の演算が施される。この場合、第２段には３２ビット幅のワーク鍵Ｋ１が入力され、このワーク鍵Ｋ１を用いて第２段の演算が行われている。

【００４８】

さらに、第３段において第２段の出力に関数π３の演算が施される。この場合、第３段には３２ビット幅のワーク鍵Ｋ２，Ｋ３が入力され、このワーク鍵Ｋ２，Ｋ３を用いて演算が行われている。続いて、第４段において第３段の出力に関数π４の演算が施される。この場合、第４段には３２ビット幅のワーク鍵Ｋ４が入力され、このワーク鍵Ｋ４を用い

て演算が行われている。
さらに残る４段の演算を行う演算段２１において、演算段２０からの出力に初段において関数π１の演算が施される。ついで、第２段において初段の出力に関数π２の演算が施される。この場合、第２段には３２ビット幅のワーク鍵Ｋ５が入力され、このワーク鍵Ｋ５を用いて演算が行われている。
【００４９】
さらに、第３段において第２段の出力に関数π３の演算が施される。この場合、第３段には３２ビット幅のワーク鍵Ｋ６，Ｋ７が入力され、このワーク鍵Ｋ６，Ｋ７を用いて演算が行われている。続いて、第４段において第３段の出力に関数π４の演算が施される。この場合、第４段には３２ビット幅のワーク鍵Ｋ８が入力され、このワーク鍵Ｋ８を用いて演算が行われている。
このようにして暗号処理の行われた上位３２ビット、下位３２ビットの合計６４ビット幅のデータは、さらに暗号８段２２に入力される。この暗号８段２２において、上述した暗号８段の演算と同様の演算が施されて、上位３２ビット、下位３２ビットの合計６４ビット幅のランダム化された出力データが得られる。
【００５０】
なお図示しているように、暗号８段の繰返し数は２回に限らず、所望の回数繰り返すことができる。この回数を多く繰り返すほど、出力データは高度にランダム化されて、暗号強度を強いものとすることができる。
なお、演算段の格段で行われている関数の演算は、一定の規則に従ってある文字を他の文字に置き換える換字と、文字の順序を入れ替える転置を行う演算とされている。
【００５１】
次に、解読アルゴリズムを実行する解読処理の構成を図１２に示すが、前述した暗号処理と異なる点は、暗号処理の構成の出力側から逆に演算を行うようにしている点である。すなわち、暗号８段のうちの最初の４段の演算段２３においては、上位３２ビットと下位３２ビットに分割された６４ビット幅の暗号化されている入力データに、３２ビット幅のワーク鍵Ｋ８を用いて関数π４の演算を施している。次いで、第２段目において、初段の出力データにワーク鍵Ｋ７を用いて関数π３の演算を施している。さらに、第３段目において、第２段の出力データにワーク鍵Ｋ６，Ｋ７を用いて関数π２の演算を施している。さらにまた第３段目において、第２段の出力データにワーク鍵Ｋ５を用いて関数π２の演算を施し、第４段目において、第３段の出力データに関数π１の演算を施している。
【００５２】
このような４段の演算が、演算２４においてワーク鍵Ｋ４～Ｋ１を用いて同様に行われる。
さらに、上記した暗号８段の演算が縦続されている暗号８段２５においても実行されて、解読化された上位３２ビット、下位３２ビットの計６４ビット幅の出力データが得られるようになる。なお、暗号８段の繰り返し回数は、暗号処理において実行された暗号８段の繰返し回数と同じ回数とされる。
【００５３】
次に、演算段で行われている演算の詳細を暗号処理の演算段２０を例に上げて図１３を参照しながらに詳細に説明する。
初段の関数π１の演算では、入力された３２ビットに分割された上位ビットは、演算されることなくそのまま出力され、上位ビットと下位ビットの排他的論理和が演算されて下位ビットとして出力される。
続く、第２段の関数π２の演算では、下位３２ビットのデータｘにワーク鍵Ｋ１が加算されて、ｘ＋Ｋ１がまず演算される。次いで、ｘ＋Ｋ１をｙとした時に、ｙを１ビット左巡回シフトし、その値にｙ－１を加算してｚを得る。次に、ｚを４ビット左巡回シフトし、その値とｚとの排他的論理和を得る。この演算結果と、上位３２ビットの排他的論理和が演算されて、演算された上位３２ビットのデータが出力される。この場合、下位３２ビットは入力されたデータが、演算されることなくそのまま出力される。

【００５４】
また、第３段の関数π３の演算では、上位３２ビットのデータｘにワーク鍵Ｋ２が加算されて、ｘ＋Ｋ２がまず演算される。次いで、ｘ＋Ｋ２をｙとした時に、ｙを２ビット左巡回シフトし、その値にｙ＋１を加算してｚを得る。次に、ｚを８ビット左巡回シフトし、その値とｚとの排他的論理和ａを得る。
さらに、ａにワーク鍵Ｋ３が加算されて、ａ＋Ｋ３が演算される。次いで、ａ＋Ｋ３をｂとした時に、ｂを１ビット左巡回シフトし、その値に－ｂを加算してｃを得る。次に、ａとｘのビット毎の論理和と、ｃを１６ビット左巡回シフトした値との排他的論理和を演算する。この演算結果と、下位３２ビットのデータとの排他的論理和を演算して、演算された下位３２ビットのデータを出力する。なお、上位３２ビットのデータは、演算されることなくそのまま上位３２ビットの出力データとなる。
【００５５】
さらにまた、第４段の関数π４の演算では、下位３２ビットのデータｘにワーク鍵Ｋ４が加算されて、ｘ＋Ｋ４がまず演算される。次いで、ｘ＋Ｋ４をｙとした時に、ｙを２ビット左巡回シフトし、その値にｙ＋１を加算する。この演算結果と、上位３２ビットの排他的論理和が演算されて、演算された上位３２ビットのデータが出力される。この場合、下位３２ビットのデータは演算されることなく、そのまま下位３２ビットのデータとして出力される。
【００５６】
上記演算において、ワーク鍵Ｋ１～Ｋ４をデータに加算することにより、文字を他の文字で置き換える換字処理が行われ、データを巡回シフトさせることにより文字の位置を入れ替える転置が行われる。このように、換字と転置のアルゴリズムを行うことにより平文が暗号文に暗号化される。
また、解読化する場合には、暗号化と逆の換字と転置のアルゴリズムを行うことにより元の平文に解読することができる。
【００５７】
次に、上述した関数の演算を行う構成をさらに詳細に説明するが、関数π２の例を図１４に上げて説明するものとする。
図１４において、第１加算器Ａｄｄ８０において、下位３２ビットの入力データｘと３２ビットのワーク鍵Ｋ１とが加算され、加算データｙが出力される。この加算データｙは第１左巡回シフター８１において１ビット左巡回シフトされると共に、第２加算器８２において、第１左巡回シフター８１の出力と加算される。この加算結果に第３加算器８４において－１が加算されて、データｚが演算される。このデータｚは第２左巡回シフター８５において４ビット左巡回シフトされると共に、排他的論理和回路８６に供給される。この排他的論理和回路８６には第２左巡回シフター８５の出力データ、データｚ、上位３２ビット入力データが入力され、３つのデータの排他的論理和が演算される。
この演算結果は、次段に入力される上位３２ビット入力データとなる。また、下位３２ビット入力データは、演算されることなく次段に入力される下位３２ビット入力データとなる。
【００５８】
次に、６４ビット幅のデータ鍵と２５６ビット幅のシステム鍵から２５６ビット幅のワーク鍵を生成する鍵スケジュール処理の構成を図１５に示す。
鍵スケジュール処理は図１５に示すように４段の演算段２６，２７が２段と、１段の演算段２８が１段縦続接続された構成とされている。また、４段の演算段２６，２７においては、初段において関数π１の演算が行われ、２段目において関数π２の演算が行われ、３段目において関数π３の演算が行われ、４段目において関数π４の演算が行われている。
【００５９】
このような演算アルゴリズムは、前述した暗号処理のアルゴリズムと同じであるのでその説明は省略するが、鍵スケジュール処理においては、入力データが６４ビットのデータ

鍵とされ、それぞれ３２ビットのシステム鍵Ｊ１～Ｊ８を用いて関数π１ないし関数π４の演算が行われて、それぞれ３２ビットの８つのワーク鍵Ｋ１～Ｋ８が生成されている。ただし、全体で９段の演算を行っており、最終段において関数π１の演算を行う点で、前述した暗号処理のアルゴリズムと相違している。

【００６０】

なお、演算段２６の関数π２演算後の上位３２ビット出力データがワーク鍵Ｋ１として出力され、関数π３演算後の下位３２ビット出力データがワーク鍵Ｋ２として出力され、関数π４演算後の上位３２ビット出力データがワーク鍵Ｋ３として出力されている。

さらに、演算段２７の関数π１演算後の下位３２ビット出力データがワーク鍵Ｋ４として出力され、関数π２演算後の上位３２ビット出力データがワーク鍵Ｋ５として出力され、関数π３演算後の上位３２ビット出力データがワーク鍵Ｋ６として出力され、関数π４演算後の上位３２ビット出力データがワーク鍵Ｋ７として出力され、最終段２８の関数π１演算後の下位３２ビット出力データがワーク鍵Ｋ８として出力されている。

【００６１】

上述した、図１１に示す暗号処理と図１３に示す鍵スケジュール処理を参照すると、４段の演算段の構成、すなわち演算アルゴリズムは等しくされており、この４段の演算段の演算を繰り返し行うことにより、暗号処理あるいは鍵スケジュール処理を実行することができる。このことから、演算コアを図１６（ａ）に示すように、関数π１の演算段、関数π２の演算段、関数π３の演算段、関数π４の演算段を縦続接続した構成とすれば、演算コアを繰返し実行することで、暗号処理あるいは鍵スケジュール処理を実行することができる。

なお、この演算コアは、図２０に示すデータ鍵とシステム鍵からワーク鍵を生成すると共に、ＣＢＣモードおよびＯＦＢモードで暗号処理を行っているEncryptor に相当し、Encryptor コアとされる。この場合、Encryptor コアにはデータ鍵Ｋｓ１～Ｋｓ４とデータ鍵Ｋｓ５～Ｋｓ８とが時分割で供給される。

【００６２】

また、図１２を参照すると、４段の演算段の構成、すなわち演算アルゴリズムは等しくされており、この４段の演算段の演算を繰り返し行うことにより、解読処理を実行することができる。このことから、解読演算コアを図１６（ｂ）に示すように、関数π４の演算段、関数π３の演算段、関数π２の演算段、関数π１の演算段を縦続接続した構成とすれば、解読演算コアを繰返し実行することで、解読処理を実行することができる。

なお、この解読演算コアは、図２０に示すＣＢＣモードおよびＯＦＢモードの解読処理を行っているDecryptor に相当し、Decryptor コアとされる。この場合、Decryptor コアにはデータ鍵Ｋｓ８～Ｋｓ５とデータ鍵Ｋｓ４～Ｋｓ１とが時分割で供給される。

このように、Encryptor コアを繰返し実行することにより暗号処理および鍵スケジュールのアルゴリズムを実行することができ、Decryptor コアを繰返し実行することにより解読アルゴリズムを実行することができる。

【００６３】

以上、本発明の解読化方法を実行する解読装置の説明をしたが、本発明の電子機器は、このような解読装置を少なくとも備えているチューナーやテレビジョン装置等である。

また、以上の説明では６４ビットブロックの暗号文を６４ビットのデータ鍵、および２５６ビットのシステム鍵を用いて６４ビットブロックの平文を生成するものとして説明したが、本発明がこれらの数値に限定されるものではなく、任意の数値とすることができる。

さらに、本発明は上述した転置および換字を繰り返すような暗号化・解読化方式に限定されるものではなく、他の暗号化・解読化方式にも適用することができる。

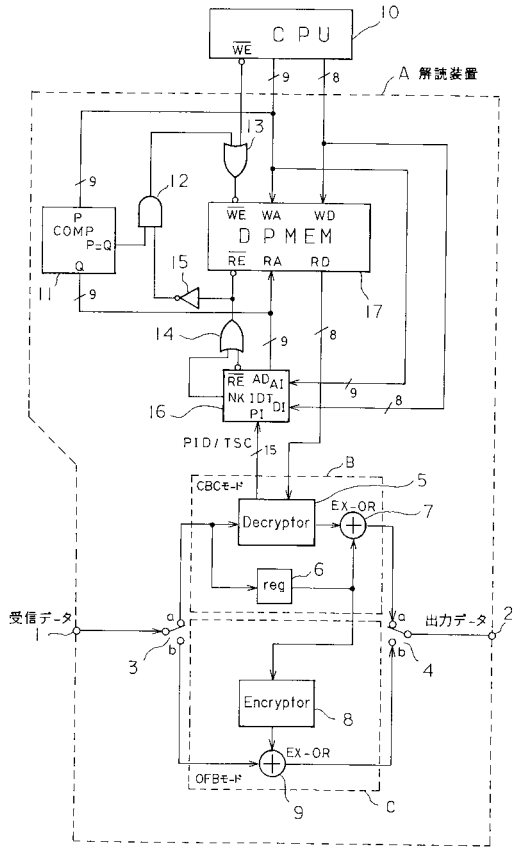
【図面の簡単な説明】

【００６４】

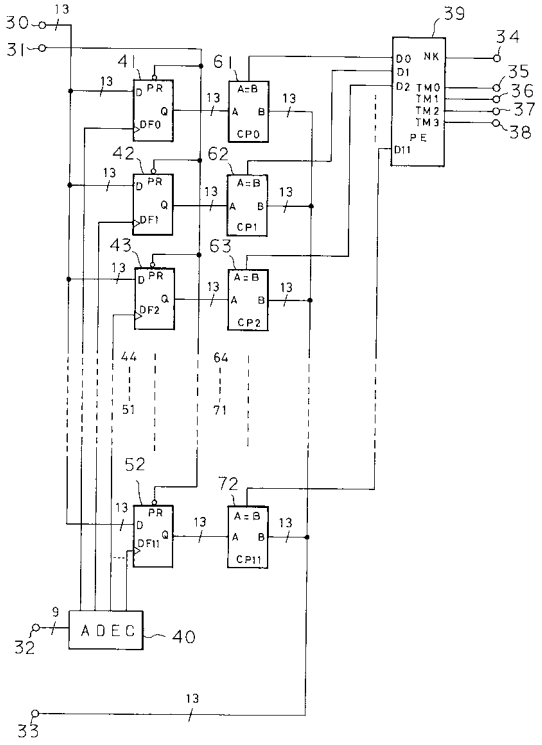
【図１】本発明の解読化方法の実施の形態である解読装置の構成例を示すブロック図である。

【図２】図１に示す解読装置におけるＩＤＴの構成を示すブロック図である。
【図３】図１に示すＣＰＵから見た解読装置におけるレジスタのメモリ空間を示す図表である。
【図４】図２に示すＩＤＴの比較器ＣＰ０～ＣＰ１１の論理表を示す図表である。
【図５】図２に示すＩＤＴのＰＥの論理表を示す図表である。
【図６】図１に示す解読装置において、入力データのヘッダ中の情報から間接検索方法によりデータ鍵をサーチする方法を説明するための図である。
【図７】ＰＩＤテーブル中にＰＩＤが重複している場合の動作を説明するための図である。
【図８】ＰＩＤテーブル中に該当するＰＩＤが存在しない場合の動作を説明するための図である。
【図９】図１に示す解読装置におけるＤＰＭＥＭのリードタイミングの例を示す図である。
【図１０】図１に示す解読装置の解読フローを示すフローチャートである。
【図１１】暗号処理の構成を示す図である。
【図１２】解読処理の構成を示す図である。
【図１３】暗号処理における基本関数の詳細を示す図である。
【図１４】基本関数中の関数π２を演算するための詳細な構成を示す図である。
【図１５】鍵スケジュール処理の構成を示す図である。
【図１６】Encryptor コアとDecryptor コアの構成を示す図である。
【図１７】従来の暗号化のアルゴリズムを示す図である。
【図１８】従来の解読化のアルゴリズムを示す図である。
【図１９】ＣＢＣモードとＯＦＢモードの暗号化利用モードの構成を示す図である。
【図２０】従来の暗号化・解読化方式の構成を示す図である。
【符号の説明】
【００６５】
１　受信データ、２　出力データ、３，４　切り換え手段、５　Decryptor 、６　レジスタ、７，９，８６　排他的論理和回路、８　Encryptor 、１０　ＣＰＵ、１１　比較器、１２　アンドゲート、１３，１４　オアゲート、１５　インバータ、１６　ＩＤＴ、１７　ＤＰＭＥＭ、２０～２８　演算段、３９　ＰＥ、４０　ＡＤＥＣ、４１～５２　フリップフロップ、６１～７２　比較器、８０，８２，８４　加算器、８１，８５　左巡回シフター

【図１】



【図２】



IDTブロック図

【図３】

HADD[8..0]	内容	ビット数
0 0010 0×××	CBC Initial value table	64bit× 1
0 010× ××××	SYSTEM_Key table	256bit× 1
0 100× ××××	PID value table	13bit×12
1 0××× ××××	Ks_even value table	64bit×12
1 1××× ××××	Ks_odd value table	64bit×12

【図４】

CP0～CP11の論理表

入力 A，B	出力 A＝B
A＝B（AもBもall '1'でないとき）	1
A＝B（AもBもall '1'のとき）	0
A＜B	0
A＞B	0

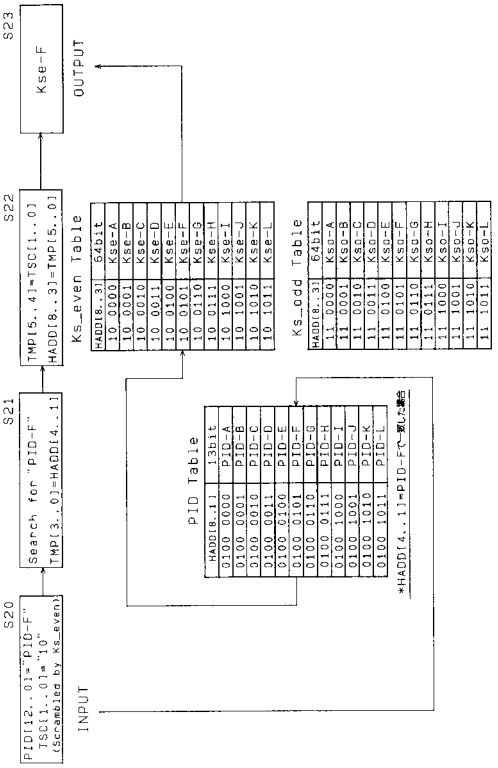
【図５】

PEの論理表

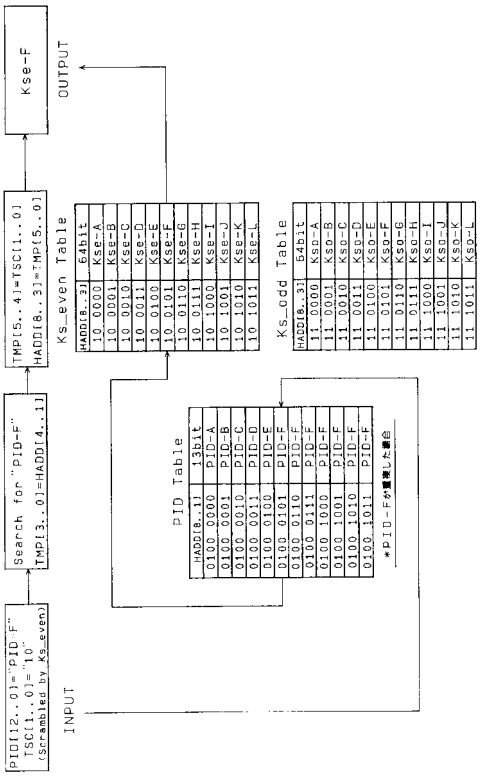
入力 D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	出力 TMP3	TMP2	TMP1	TMP0	NK
×	×	×	×	×	×	×	×	×	×	×	1	0	0	0	0	0
×	×	×	×	×	×	×	×	×	×	1	0	0	0	0	1	0
×	×	×	×	×	×	×	×	×	1	0	0	0	0	1	0	0
×	×	×	×	×	×	×	×	1	0	0	0	0	0	1	1	0
×	×	×	×	×	×	×	1	0	0	0	0	0	1	0	0	0
×	×	×	×	×	×	1	0	0	0	0	0	0	1	0	1	0
×	×	×	×	×	1	0	0	0	0	0	0	0	1	1	0	0
×	×	×	×	1	0	0	0	0	0	0	0	1	0	0	0	0
×	×	×	1	0	0	0	0	0	0	0	0	1	0	0	1	0
×	×	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0
×	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

×＝Do not care

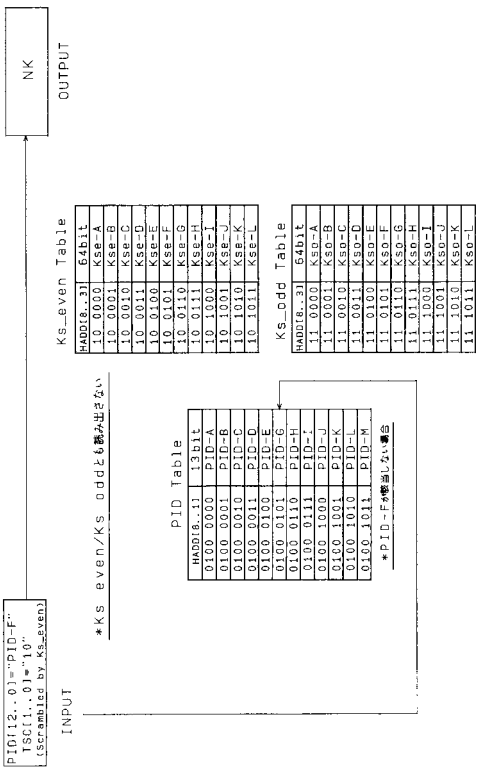
【図６】


S20
S21
S22
S23
INPUT
OUTPUT
PID[12..0]="PID-F"
TSC[1..0]="10"
(Scrambled by Ks_even)
Search for "PID-F"
TMP[3..0]=HADDI[4..1]
TMP[5..4]=TSC[1..0]
HADDI[8..3]=TMP[5..0]
Ks-F
Ks_even Table
Ks_odd Table
PID Table
*HADDI[4..1]=PID-Fへ一致した場合

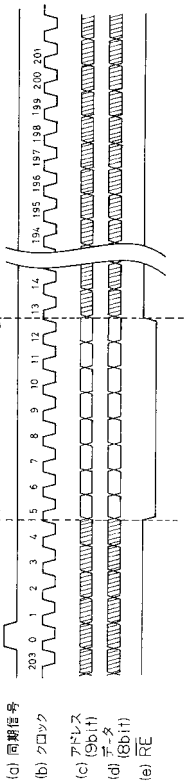
【図７】


S20
S21
S22
S23
INPUT
OUTPUT
PID[12..0]="PID-F"
TSC[1..0]="10"
(Scrambled by Ks_even)
Search for "PID-F"
TMP[3..0]=HADDI[4..1]
TMP[5..4]=TSC[1..0]
HADDI[8..3]=TMP[5..0]
Kse-F
Ks_even Table
Ks_odd Table
PID Table
*PID-Fが一致した場合

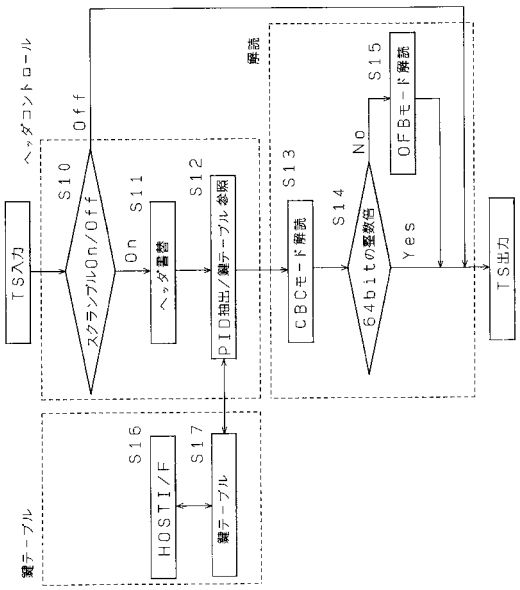
【図８】


INPUT
OUTPUT
PID[12..0]="PID-F"
TSC[1..0]="10"
(Scrambled by Ks_even)
NK
*Ks even/Ks oddとも読み出さない
Ks_even Table
Ks_odd Table
PID Table
*PID-Fが一致しない場合

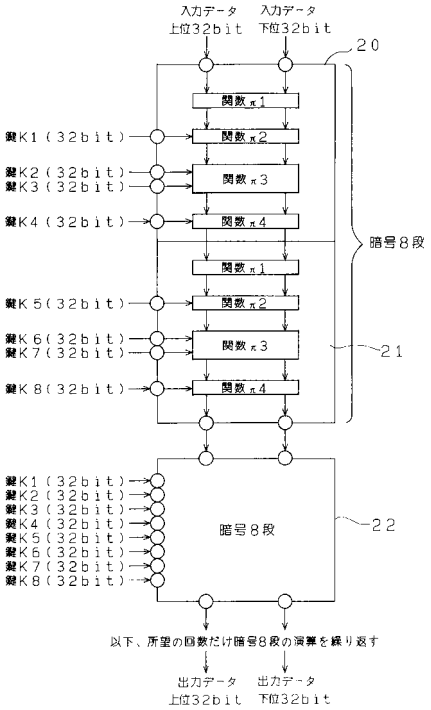
【図９】


8クロック
(a) 同期信号
(b) クロック
(c) アドレス (9bit)
(d) データ (8bit)
(e) RE

【図１０】

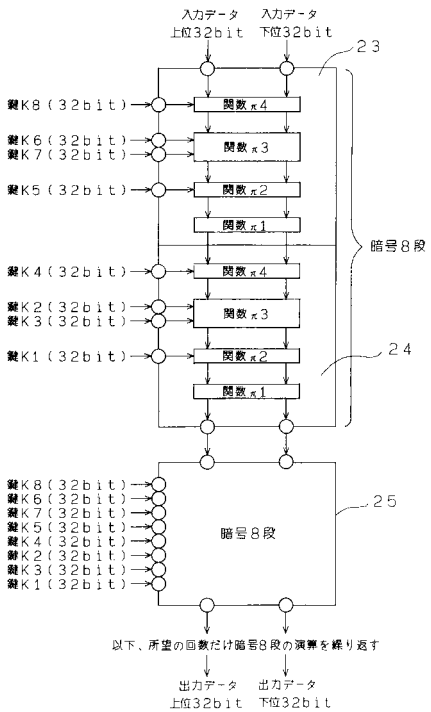


【図１１】



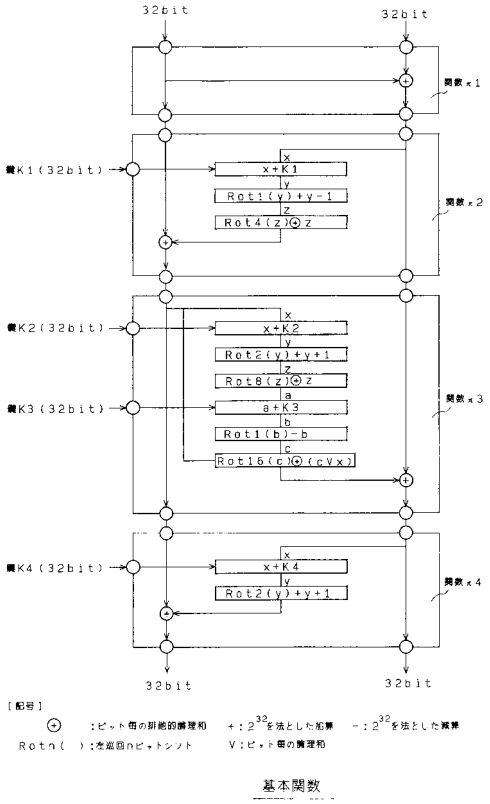
暗号処理

【図１２】



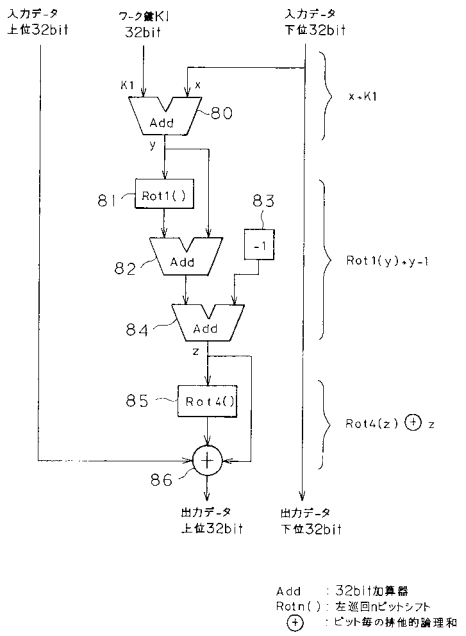
解読処理

【図１３】



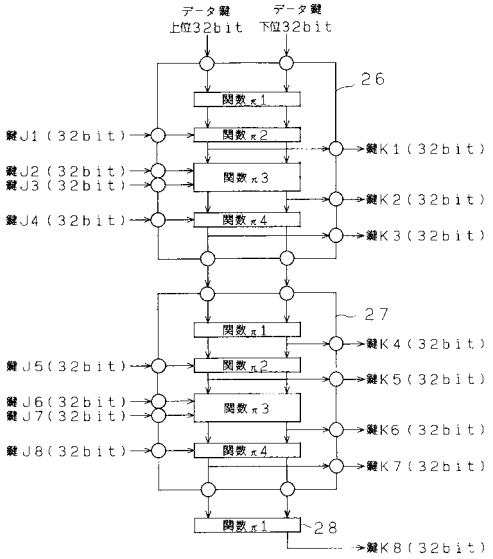
基本関数

【図１４】



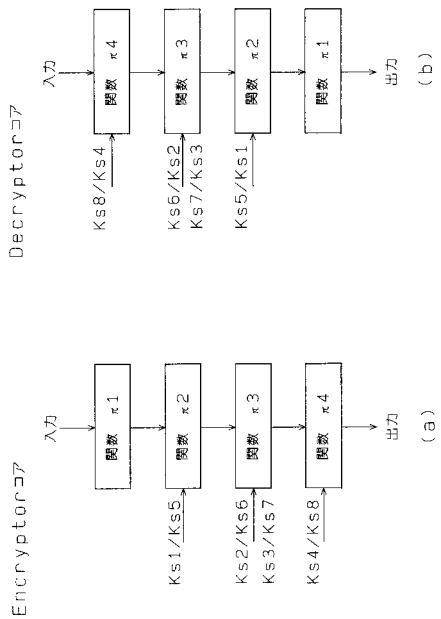
基本関数π2の回路構成例

【図１５】

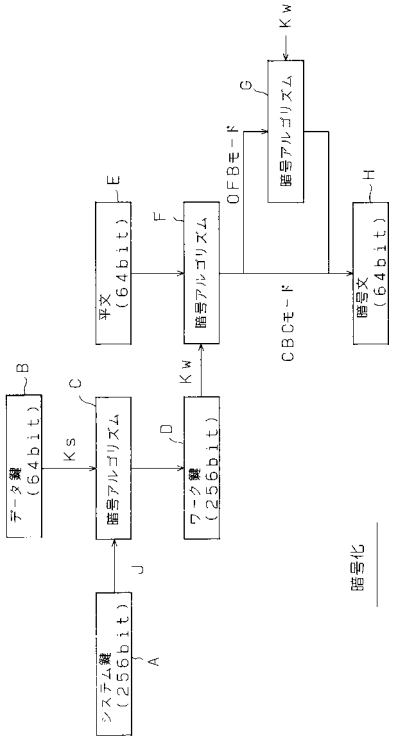


鍵スケジュール処理

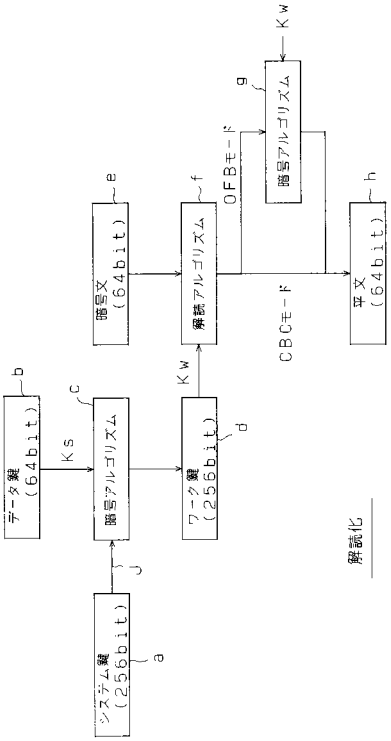
【図１６】



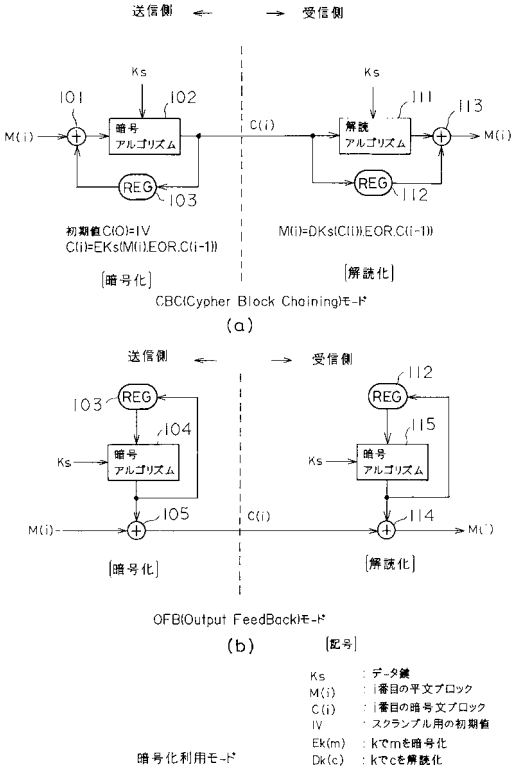
【図１７】



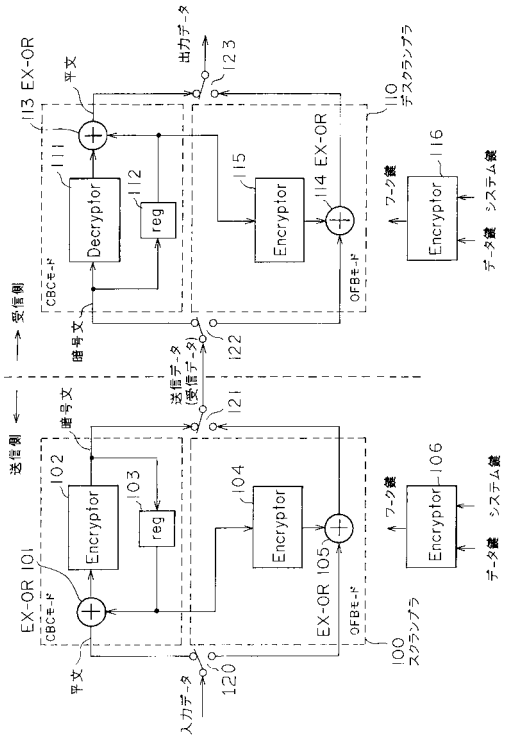
【図１８】


システム鍵 (256bit)
a
データ鍵 (64bit)
b
暗号アルゴリズム
c
Ks
ワーク鍵 (256bit)
d
暗号文 (64bit)
e
解読アルゴリズム
f
暗号アルゴリズム
g
Kw
OFBモード
CBCモード
平文 (64bit)
h
解読化

【図１９】


送信側
受信側
Ks
M(i)
101
暗号アルゴリズム
102
C(i)
REG
103
解読アルゴリズム
111
113
112
初期値C(0)=IV
C(i)=EKs(M(i).EOR.C(i-1))
M(i)=DKs(C(i)).EOR.C(i-1))
[暗号化]
[解読化]
CBC(Cypher Block Chaining)モード
(a)
104
105
114
115
OFB(Output FeedBack)モード
(b)
暗号化利用モード
[記号]
Ks : データ鍵
M(i) : i番目の平文ブロック
C(i) : i番目の暗号文ブロック
IV : スクランブル用の初期値
Ek(m) : kでmを暗号化
Dk(c) : kでcを解読化

【図２０】


送信側
受信側
入力データ
出力データ
送信データ
受信データ
平文
暗号文
EX-OR 101
EX-OR 113
EX-OR 105
EX-OR 114
Encryptor
Decryptor
reg
CBCモード
OFBモード
スクランブラ
ワーク鍵
システム鍵
データ鍵

フロントページの続き

(56)参考文献　特開昭６１－１７７０４６（ＪＰ，Ａ）
特開平６－２５９９５６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L9/00