(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4559679号
(P4559679)
(45)発行日　平成22年10月13日(2010.10.13)　(24)登録日　平成22年7月30日(2010.7.30)

(51)Int.Cl.　F I
G09C 1/00 (2006.01)　G09C 1/00 650A

請求項の数 8 (全 17 頁)

(21)出願番号	特願2001-504202 (P2001-504202)
(86)(22)出願日	平成12年6月9日 (2000.6.9)
(65)公表番号	特表2003-526118 (P2003-526118A)
(43)公表日	平成15年9月2日 (2003.9.2)
(86)国際出願番号	PCT/US2000/016035
(87)国際公開番号	WO2000/078118
(87)国際公開日	平成12年12月28日 (2000.12.28)
審査請求日	平成19年5月8日 (2007.5.8)
(31)優先権主張番号	09/329,139
(32)優先日	平成11年6月9日 (1999.6.9)
(33)優先権主張国	米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　ラマラサナム　ベンカテサン
アメリカ合衆国　９８０５２　ワシントン州　レドモンド　ノースイースト　２２　コート　１７２０８

最終頁に続く

(54)【発明の名称】基本的なレジスタ演算を用いた暗号プリミティブのインプリメント

(57)【特許請求の範囲】

【請求項１】

入力のディジタル平文または暗号文の、ブロック各々を、出力のディジタル暗号文または平文の、ブロック各々に、暗号化または復号するための、機器で用いる暗号方法であって、

バイトスワップ、ワードスワップ、反転演算、加算、およびｍｏｄ（２^{n}）乗算の演算（ｎは事前定義済みの整数）の、事前定義済みシーケンスを、プリミティブＦ（ｘ）として含む所定のプロシージャを通して、前記ディジタル平文または暗号文ブロックを、各々前記出力ディジタル暗号文または平文ブロックに変換するステップを備え、前記演算は集合的に前記プリミティブを実施するがｍｏｄ（Ｍ）の値を計算することはなく、Ｍは大きな素数であり、前記プリミティブは短縮された処理時間を呈し、以下の式、

$x^{S} \leftarrow wordswap(x)$

$y \leftarrow Ax + Bx^{S} \bmod (2^{n})$

$y^{S} \leftarrow wordswap(y)$

$z \leftarrow Cy^{S} + yD \bmod (2^{n})$

$\theta \leftarrow z + y^{S}E \bmod (2^{n})$

に従って、前記プリミティブＦ（ｘ）を実施し、

係数Ａ、Ｂ、Ｃ、ＤおよびＥは、それぞれ２^{n}以下のランダムな奇数の整数であり、

θは出力ストリングである

ことを特徴とする暗号方法。

【請求項２】
平文と暗号文の前記ディジタルブロックは、両方ともｎビット長であ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】
Ｍは$2^{31}-1$以上の素数であることを特徴とする請求項１に記載の暗号方法。
【請求項４】
請求項１に記載の各ステップを実行するためのコンピュータ実行可能命令を記録したことを特徴とするコンピュータ読み取り可能な記録媒体。
【請求項５】
入力のディジタル平文または暗号文の、ブロック各々を、出力のディジタル暗号文または平文の、ブロック各々に、暗号化または復号するための、機器で用いる暗号方法であって、
バイトスワップ、ワードスワップ、反転演算、加算、およびｍｏｄ（2^n）乗算の演算（ｎは事前定義済みの整数）の、事前定義済みシーケンスを、プリミティブＦ（ｘ）として含む所定のプロシージャを通して、前記ディジタル平文または暗号文ブロックを、各々前記出力ディジタル暗号文または平文ブロックに変換するステップを備え、前記演算は集合的に前記プリミティブを実施するがｍｏｄ（Ｍ）の値を計算することはなく、Ｍは大きな素数であり、前記プリミティブは短縮された処理時間を呈し、以下の式、
$y \leftarrow Ax \bmod (2^n)$
$y^S \leftarrow wordswap(y)$
$z \leftarrow By^S \bmod (2^n)$
$z^S \leftarrow wordswap(z)$
$v \leftarrow Cz^S \bmod (2^n)$
$v^S \leftarrow wordswap(v)$
$w \leftarrow Dv^S \bmod (2^n)$
$w^S \leftarrow wordswap(w)$
$t \leftarrow Ew^S \bmod (2^n)$
$\theta \leftarrow t + Ly^S \bmod (2^n)$
に従って、前記プリミティブＦ（ｘ）を実施し、
係数Ａ、Ｂ、Ｃ、ＤおよびＥは、それぞれ2^n以下のランダムな奇数の整数であり、
Ｌは2^n以下のランダムな整数であり、
θは出力ストリングである
ことを特徴とする暗号方法。
【請求項６】
Ｍは$2^{31}-1$以上の素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７】
平文と暗号文の前記ディジタルブロックは、両方ともｎビット長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】
請求項５に記載の各ステップを実行するためのコンピュータ実行可能命令を記録したことを特徴とするコンピュータ読み取り可能な記録媒体。
【発明の詳細な説明】
【０００１】
（開示の背景）
（１．発明の分野）
本発明は、例えばチェックサムを計算するためにプリミティブ（primitive）をインプリメントする（implement；実施する）技術に関する。有利なことに、この技術は比較的単純であり、かなり基本的なレジスタ演算を用いる。したがって、例えばメッセージ認証コード（ＭＡＣ、ｍｅｓｓａｇｅ　ａｕｔｈｅｎｔｉｃａｔｉｏｎ　ｃｏｄｅ）を計算するため、またはストリーム暗号を実施するための処理時間が、従来必要とされていた処理時

間に打ち勝って大きく節約される。

【０００２】

（２．従来技術の説明）

現在使用されている多くの様々な暗号技術は、今日、モジュラー計算法を含めた数学関数を採用しており、モジュラー算法は通常、比較的大きな素数（Ｍ）、例えば$2^{32}-1$やそれ以上の素数などについて、数の剰余を計算するものである。このような関数の一例、ｆ（ｘ）は、2^{n}を超えるガロア体（ＧＦ）においてｆ（ｘ）＝ａｘ＋ｂｍｏｄ（Ｍ）の形となり、この場合、ｎ＝２ｍ＋１であり、ｎおよびｍは、体Ｚ（ｍｏｄＭ）における事前定義済みの整数である。関数自体は技術間で大きく異なるが、これらは共通して、何らかの形の、かつ通常は著しく反復的なｍｏｄ（Ｍ）演算の計算が必要である。

【０００３】

このようなモジュラー演算は、メッセージ中の平文の各々かつ全てのブロックを暗号化して対応する暗号文ブロックを生み出し、暗号文ブロックを復号して関連する平文ブロックを回復するのに用いられるだけでなく、メッセージ認証コード（ＭＡＣ）やストリーム暗号など、この技術の中間部分を計算する際にも用いられる。

【０００４】

単一のｍｏｄ（Ｍ）演算を行うには、これより多くはならないとしても１０～１５サイクルもの処理サイクルが必要となる可能性がある（法の値Ｍに基づく）。暗号技術にはこのような演算が多数必要なので、この技術を採用することに伴い、単にｍｏｄ（Ｍ）演算を計算することだけにかなりの量の処理時間が消費される可能性がある。

【０００５】

暗号技術は、幅広く拡大しつつある非常に多岐にわたる用途において、ならびに、例えばパーソナルコンピュータやワークステーションなどの高度に複雑な汎用目的の機器から、例えば「スマートカード」、リモートコントロール、電子器具などの比較的単純な専用機器までの、拡大しつつある多くの機器において、情報を保護するためにますます利用されている。

【０００６】

例えば、電子メールによる通信の容易さおよび低コストから、インターネットは（ネットワークモダリティの中でもとりわけ）、好適な通信媒体として爆発的かつ急激な成長を遂げている。しかし、インターネットは公衆のアクセスが可能なネットワークなのでセキュア（secure）ではなく、事実、盗聴、傍受、および／あるいはさもなければインターネットメッセージトラフィックに障害を生じさせるかさらには崩壊させること、または不正にインターネットサイトを通り抜けることを意図した、様々な個人および組織からの多様な攻撃の標的になっており、ますますそうなり続けている。好適な通信媒体としてのインターネット使用にますます信頼が置かれるようになっていることに鑑みて、こうしたセキュリティの脅威は、第三者による盗聴、傍受、および可能性ある改ざんから、メールメッセージやデータおよびコンピュータファイルなどの電子通信を守るセキュリティのレベルの改善をもたらす、ますます強力な暗号技術を開発するための当技術分野における努力を激化させる。したがって、セキュアなインターネットの接続性を提供するために、ますます多くのパーソナルコンピュータソフトウェア、特にウェブブラウザおよびその他のオペレーティングシステムコンポーネントと、電子メールプログラムおよびその他のアプリケーションプログラムに、暗号処理が組み込まれている。

【０００７】

これとは全く異なる暗号適用例は、いわゆる「スマートカード」に関係するものである。この場合、いくぶん複雑でなく安価なマイクロプロセッサを採用したクレジットカードサイズの専用デバイス、すなわち「スマートカード」が、対応する個人についての銀行および／または他の金銭残高を記憶する。マイクロプロセッサは、カード内部に記憶されたプログラムを使用し、取引の妥当性を検査して、そのような各残高を取引に基づいて適切に変更することができる。具体的には、その個人は、カードに記憶された残高の全部または一部を借方記入（debit）および／または貸方記入（credit）するために、単にカードを

適切なデータ端末に挿入して端末に結合されたキーボードに取引データを入力するだけで、ベンダや銀行などの別の当事者との電子取引を実施することができる。この方式の取引によれば、即座に金銭振替ができ、紙幣を、または小切手などの紙ベースの通貨手段を、処理する必要およびそれに関連するコストはどれも不要になる。記憶されたプログラムは、非常に強力な暗号技術を利用して、カードに記憶された情報、特に残高を第三者による不正なアクセスおよび改ざんから守る。

【０００８】

しかし前述のように、暗号は処理オーバーヘッドを招く。ＰＣやワークステーションなど、大きな処理能力を有する複雑な機器では、このオーバーヘッドによってシステムスループット全体が低下し、スマートカード、リモートコントロール、およびその他の「ローエンド」デバイスなど、やや限られた処理能力を有する他のデバイスでは、オーバーヘッドに耐えられず、十分に強力な暗号技術をそのようなデバイス中で使用することが阻まれる場合がある。

【０００９】

したがって、多様なデバイス、特に限られた処理力を有するデバイスに、暗号技術を組み込むことへの当技術分野における要望は急速であり、かつ絶えず増え続けるように見受けられるので、現在、暗号技術を実施するのに必要な処理時間を短縮する必要性が、当技術分野に存在する。

【００１０】

具体的には、等価だが処理集中性のより低い演算で、ｍｏｄ（Ｍ）演算を置き換えられれば、いくつかの暗号技術に関連する処理オーバーヘッドは、特にチェックサムの計算において、急激に減少する可能性がある。この結果が達成できれば、様々な暗号技術を採用したパーソナルコンピュータやワークステーションなど、複雑度の高い機器のスループット全体を有利に向上させることができる。さらに、このようなオーバーヘッドを削減することができれば、従来このような技術をうまくサポートする十分な処理力を有しなかった多数のコンピュータ関連デバイスにも、強力な暗号技術を組み込むことができる。

【００１１】

（発明の概要）

有利なことに我々の本発明は、チェックサムを計算するためのプリミティブを実施することによってこの必要性を満足するが、有利にもｍｏｄ（Ｍ）演算は必要ない。

【００１２】

我々の広範な発明の教示によれば、このプリミティブは、ｍｏｄ（Ｍ）演算を単純な一連の基本的レジスタ演算で置き換える。これらの演算は、ｍｏｄ2^n乗算、順序操作（例えばバイトスワップやワードスワップ）、および加算を含み、これらはすべて実施が非常に単純であり、実行するのにごくわずかな処理サイクルしか必要としない。我々の発明の教示を用いれば、様々な暗号パラメータ、例えばメッセージ認証コード（ＭＡＣ）を計算するため、またはストリーム暗号を実装するために必要な処理時間を、従来必要とされていた処理時間に打ち勝って大きく短縮することができる。

【００１３】

具体的には、我々の技術の基本的、例示的、かつ非可逆的なバージョンは、以下の一連の式を通してプリミティブを計算することを基にする。

$x^S \leftarrow wordswap(x)$

$y \leftarrow Ax + Bx^S \bmod (2^n)$

$y^S \leftarrow wordswap(y)$

$z \leftarrow Cy^S + yD \bmod (2^n)$

$\theta \leftarrow z + y^S E \bmod (2^n)$

上式で、係数Ａ、Ｂ、Ｃ、Ｄ、Ｅは、それぞれ2^n以下のランダムな奇数の整数であり、θはｎビットストリングである。

【００１４】

ＭＡＣまたは他の暗号パラメータを生成する際に使用する場合は、これらの係数は「秘密

」である。しかしチェックサムの生成に使用するときは、これらの係数は公開される。
【００１５】
有利にも我々の発明的技術は、その特徴として、可逆と非可逆の両方の変形を有する。
【００１６】
（詳細な説明）
以下の説明を考察した後には、チェックサムの計算を含む幅広い暗号技術のいずれにおいても、我々の本発明の教示が利用できることを、当業者ならはっきりと理解するであろう。このような技術とは、例えばメッセージ認証コード（ＭＡＣ）を計算する技術やストリーム暗号を実装する技術である。
【００１７】
読者が容易に理解できるように、我々はインターネットなどの、セキュアでない通信ネットワークを介して、トランザクションメッセージが通信されるクライアントサーバトランザクション処理環境で、採用できるような技術（かなり一般化されるが）における使用の面において我々の発明を論じ、特に、その技術において採用されるＭＡＣを計算する面において論じる。
【００１８】
Ａ．概観
図１は、我々の本発明を用いることによって、ＭＡＣを生成する端末相互間の暗号プロセス５全体のブロック図を示す。
【００１９】
図示されるように、到来した平文情報はいわゆる「メッセージ」に構成されている。Ｐとして示す、このような各メッセージ７をＮ個のブロック（P_1，P_2，．．．，P_N）として構成するが、各ブロックはｎビット幅であり、ここでは例示的にｎは３２ビットである。このような各平文ブロックを、線１０で表すように暗号化プロセス２０に適用する。このプロセスは例示的に、メッセージ暗号化プロセス２３および発明のＭＡＣ生成プロセス４００を含む。プロセス４００（図４および５に関連して後で詳細に述べる）は、平文メッセージＰまたは適したその暗号操作が入力として与えられれば、我々の発明により、このメッセージに固有かつ通常６４ビット長であるＭＡＣを生成する。メッセージ暗号化プロセス２３は、平文メッセージを暗号文に暗号化し、２つの最上位ブロック（C_{N-1}，C_N）として例示するように、６４ビットＭＡＣをそのメッセージに適切に挿入して、暗号文メッセージＣを生み出す（以下、括弧内の連続した値を分離するカンマを、これらの値の連鎖を示す演算子として用いる）。この２つの最上位ブロックは、集合的にＭＡＣ４２を形成する。プロセス２３内で採用される具体的な暗号化プロセスに応じて、ＭＡＣ４２は、周知のＤＥＳ（データ暗号化規格）暗号化や別の従来の擬似ランダム置換などを通して、それ自体で暗号化してもよく、そうしなくてもよい。暗号文メッセージは、連続するＮ個のｎビット暗号文ブロックで形成される。
【００２０】
得られた暗号文メッセージＣは、次いで記憶されるか、所与のモダリティ、例えば破線４５で表されインターネット接続に代表されるセキュアでない通信チャネルを介して、受信側の場所に転送される。ここで、受信済みバージョンの暗号文メッセージを
【００２１】
【数１】
$\tilde{C}$
【００２２】
として示すが（メッセージ４０'としても符号付けする）、これを復号プロセス５０によって復号して、回復された平文メッセージ７０を生み出す。これは平文メッセージ
【００２３】
【数２】

$\hat{P}$

【００２４】
としても示すが、有効でありしたがってダウンストリーム使用に適するものであるためには、すべての面において元の平文メッセージＰと同一でなければならない。復号プロセス５０は、メッセージ復号プロセス６０、ＭＡＣ生成プロセス４００、および識別コンパレータ９０を含む。
【００２５】
回復された平文メッセージが有効であるかどうか、例えば改変されていないかどうかを判定するために、メッセージ復号プロセス６０は、回復された平文を作成するだけでなく、暗号文メッセージ
【００２６】
【数３】

$\tilde{C}$

【００２７】
からＭＡＣを抽出する（かつ必要なら復号する）。得られる回復されたＭＡＣを、線６７で表すように、コンパレータ９０の入力の一方に加える。回復された平文もまた、線７７で表すように、ＭＡＣ生成プロセス４００に加える。プロセス４００は、回復された平文メッセージ
【００２８】
【数４】

$\hat{P}$

【００２９】
からＭＡＣを再計算し、線８０で表すように、得られた再計算済みＭＡＣをコンパレータ９０の入力の他方に加える。コンパレータ９０の対応する入力にこのとき加えたこれらのＭＡＣが両方とも等しく一致する場合、コンパレータ９０は、出力リード７３上にこの時現れる、回復された平文メッセージ
【００３０】
【数５】

$\hat{P}$

【００３１】
が、後続の使用に有効であることを示す適切なＩＤ（identification）を、出力９３上に生成する。そうではなく、回復したＭＡＣと再計算したＭＡＣが一致しない場合は、コンパレータ９０は、出力７３上にこのとき現れる、回復された平文メッセージ
【００３２】
【数６】

$\hat{P}$

【００３３】
が、無効であり無視すべきであることを示す適切なＩＤを、出力９７上に生成する。ＭＡＣの生成は別として、暗号化プロセス２３およびメッセージ復号プロセス６０で、それぞれ用いる暗号化技術および復号技術の具体的な性質は、本発明に無関係であり、このような様々な技術のいずれでもうまく用いることができるので、これらの態様についてはこれ以上詳細に論じない。しかし我々の、１９９８年４月２０日出願の「Ｃｒｙｐｔｏｇｒａｐｈｉｃ Ｔｅｃｈｎｉｑｕｅ Ｔｈａｔ Ｐｒｏｖｉｄｅｓ Ｆａｓｔ Ｅｎｃｒｙｐｔｉｏｎ ａｎｄ Ｄｅｃｒｙｐｔｉｏｎ ａｎｄ Ａｓｓｕｒｅｓ Ｉｎｔｅｇｒｉｔｙ ｏｆ ａ Ｃｉｐｈｅｒｔｅｘｔ Ｍｅｓｓａｇｅ」という名称の係属中の米国特許

出願、出願番号０９／０６２、８３６と、１９９８年４月２０日出願の「Ｍｅｔｈｏｄ ａｎｄ Ａｐｐａｒａｔｕｓ ｆｏｒ Ｐｒｏｄｕｃｉｎｇ Ａ Ｍｅｓｓａｇｅ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｃｏｄｅ」という名称の同時係属中の米国特許出願、出願整理番号０９／０６２、８３７（読者は参照されたい）の中で、このような暗号技術の一例を述べ、特許請求している。これらを両方とも参照により本明細書に組み込み、またこれらは本明細書と同一の譲受人に譲渡されている。

【００３４】

Ｂ．例示的な処理環境

以上のことを念頭に置いて、図２を考察されたい。図２には、本発明を利用するクライアントサーバ処理環境２００の高レベルのブロック図が示してある。

【００３５】

図示の通り、この環境は、サーバ２１０を実装するコンピュータ２０５を含み、サーバ２１０は例示的にウェブサーバとする。リモートに位置するいくつかの個別クライアントコンピュータを、それぞれ例示的にパーソナルコンピュータ（ＰＣ）とし、このようなクライアントのうちの１つだけ、すなわちクライアントコンピュータ１００だけを具体的に示すが、これは、チャネル１４０や１６０などの適切な通信チャネルを使用して、セキュアでない通信ネットワークを介してコンピュータ２０５に接続されており、通信ネットワークはここでは例示的にインターネット１５０として示す。クライアントコンピュータ１００に配置され、サーバから情報を得たいと思っているユーザ（具体的には図示せず）は、クライアントコンピュータ１００にある対応するクライアントプログラム１３０を呼び出すことができる。クライアントプログラムは、クライアントコンピュータ１００内に集合的にあってクライアントコンピュータ１００によって実行される、いくつかのアプリケーションプログラム１２０のうちの１つを形成する。クライアントプログラムは、アプリケーションプログラム内にあるものとして具体的に示してあるが、クライアントプログラムはまた、ウェブブラウザやオペレーティングシステム（Ｏ／Ｓ）、例えば図３に示すＯ／Ｓ３３７などのコンポーネントとして実装してもよい。図２に示すサーバ２１０は、例えば、コマースサーバ、バンキングサーバ、電子メールサーバまたはファイルサーバを含めた様々なアプリケーション機能のいずれを実装していてもよい。電子商取引に関しては、ユーザは、クライアントコンピュータ１００およびサーバ２１０を介して、商業取引を行いたいと思う場合があり、これは、金融機関におけるユーザの口座番号や、受取人に資金を振り替えるための支払い指示などの情報を、サーバに提供すること（線１１０で表す）、またはユーザの利用可能な口座残高またはクレジット残高などの情報をサーバから得ること（線１３５で表す）を含み、いずれの場合でもそれはそのユーザの秘密である。あるいは、サーバ２１０は、リポジトリに記憶された様々なファイルへのアクセス権をユーザに与えるファイルサーバとすることもでき、ユーザは任意のファイルをダウンロードすることができる。このようなファイルをダウンロードすると、そこでローカル利用するためにそれをクライアントコンピュータ１００内にあるメモリ３３０（図３参照）内に記憶することができる。しかしこのようなファイルは、その所有者がユーザアクセスを制御したいと思うプロプラエタリ情報および／または機密の情報を含む場合がある。例えば、このようなファイルは、所与のプログラムに対する更新の自己インストール実行可能ファイルとすることができ、このプログラムに対して、その所有者例えばソフトウェア製造者は、不正な公衆アクセスを防止すること、すなわち、それに対して適切な支払いを送金していない個人によってその更新が使用されるのを防止することを望む。図２に示すように、サーバ２１０自体がまた、ユーザから発せられた（かつネットワーク（ここではインターネット）１５０を介してサーバに送信された）機密のまたはプロプラエタリ情報を、後続の処理のためにダウンストリーム機器（具体的には図示せず）に提供する（線２１５で表す）か、あるいは最終的に、ネットワークを介して、ユーザに送信するために、機密のまたはプロプラエタリ情報をダウンストリーム機器から受け取る（線２１８で表す）場合もある。

【００３６】

例示的にインターネットとするネットワーク１５０は、第三者によって障害が引き起こされやすい。これに関して、従来通り暗号化された後でネットワークを介して搬送されているメッセージであって、例えばクライアントコンピュータ１００に位置するユーザに関係する進行中の金融取引のために、例えばクライアントコンピュータ１００から発せられたメッセージを、第三者が傍受する可能性もある。第三者は、利用可能な処理能力または時間からみて、メッセージを暗号化するのに使用された従来型の暗号を破って、送信メッセージに内在する平文を回復するのに、十分な資源を有しない場合もあるが、それでもこの当事者は、その暗号文メッセージに関する十分な知識、具体的にはその構造上の構成と、そのメッセージをユーザの損害になるように首尾よく変更するのに必要な手段とを有する可能性がある。これに関して、この第三者は、１つまたは複数の事前定義済み暗号文ブロックで対応する元の暗号文ブロックを置き換え、次いで、得られた修正済み暗号文メッセージを、コンピュータ２０５に搬送されてそこで処理されるようにネットワーク上に返送することにより、暗号文メッセージを不正に改ざんする可能性がある。

【００３７】

クライアントコンピュータ１００とコンピュータ２０５との間で、ネットワーク１５０上を通過する機密のまたはプロプラエタリな性質の情報を、第三者アクセスから保護するために、クライアントプログラム１３０とサーバ２１０は両方とも、それぞれ、暗号化プロセス２０および復号プロセス５０を組み込むことによって、暗号通信を利用する。したがって、ネットワークで搬送されることになっておりクライアントプログラム１３０とサーバ２１０のいずれかのネットワークアプリケーションピア（peer）によって生成されたメッセージは、それぞれその中で暗号化プロセス２０によって暗号化されて、ＭＡＣが埋め込まれた対応する暗号文メッセージが生み出され、これらの暗号文メッセージは、順に、それぞれネットワーク１５０を介して、他方のネットワークアプリケーションピアに送信される。同様に、各ピアによってネットワークから受け取られた暗号文メッセージは、各ピアの中で復号プロセス５０によって復号されて、適切な回復済み平文メッセージおよびその妥当性に関してＩＤが生み出される。暗号化プロシージャ２０と復号プロシージャ５０とは、相互の逆のプロシージャである。

【００３８】

Ｃ．クライアントコンピュータ１００

図３に、クライアントコンピュータ（ＰＣ）１００のブロック図を示す。

【００３９】

図示のように、クライアントコンピュータ１００は、入力インタフェース（Ｉ／Ｆ）３２０、プロセッサ３４０、通信インタフェース３５０、メモリ３３０、および出力インタフェース３６０を備え、これらはすべて従来通りバス３７０によって相互接続される。メモリ３３０は一般に、データおよび命令を一時的に記憶するためのランダムアクセスメモリ（ＲＡＭ）３３２と、ユーザコマンドにより、フロッピー（登録商標）ディスケットで情報を交換するためのディスケットドライブ３３４と、通常は磁気性質であるハードディスクによって実装される不揮発性大容量記憶装置３３５とを例示的に含めた、異なるモダリティを含む。大容量記憶装置３３５はまた、適した光学記憶媒体から情報を読み取る（かつそれに情報を書き込む）ための、ＣＤ－ＲＯＭまたは他の光学媒体リーダ（具体的には図示せず）（またはライタ）も含むことができる。大容量記憶装置は、オペレーティングシステム（Ｏ／Ｓ）３３７およびアプリケーションプログラム１２０を記憶する。後者は例示的に、我々の発明的技術を組み込んだクライアントプログラム１３０（図２参照）を含む。図３に示すＯ／Ｓ３３７は、ＷＩＮＤＯＷＳ（登録商標）　ＮＴオペレーティングシステム（「ＷＩＮＤＯＷＳ（登録商標）　ＮＴ」はワシントン州Ｒｅｄｍｏｎｄにある Ｍｉｃｒｏｓｏｆｔ　Ｃｏｒｐｏｒａｔｉｏｎの登録商標）など、従来のオペレーティングシステムのいずれによっても実装することができる。このことから、Ｏ／Ｓ３３７のコンポーネントはすべて関係がないので、そのいずれについても論じない。アプリケーションプログラム１２０のうちの１つであるクライアントプログラムがＯ／Ｓの制御下で実行される、と言えば十分であろう。

【００４０】
有利なことに、我々のこの発明の技術は、暗号法による暗号化モジュールおよび復号モジュール内で用いられるように組み込まれたとき、有利に処理時間を節約し、それによりクライアントコンピュータ１００とサーバ２１０（図２参照）の両方のスループットを向上させる。
【００４１】
図３に示すように、例示的な２つの外部ソースからの到来情報が発生する可能性がある。すなわち、例えばネットワーク接続１４０を介してインターネットおよび／または他のネットワーク化された機能から、通信インタフェース３５０に供給されるか、あるいはパス３１０を介して、専用入力ソースから入力インタフェース３２０に供給される、ネットワーク供給情報である。専用入力は幅広い、様々なソース、例えば外部データベースから発することができる。さらに、入力情報は、その情報を含むディスケットをディスケットドライブ３３４に挿入することによって、ファイルまたはその中の具体的な内容の形で、提供することもでき、コンピュータ１００は、ユーザの命令下で、ディスケットからその情報にアクセスして読み取る。入力インタフェース３２０は、入力情報に対する異なる専用ソースそれぞれを、コンピュータシステム１００に物理的に接続およびインタフェースするのに必要とされる、必要かつ対応した電気的接続を提供するのに、適した回路を備える。アプリケーションプログラム１２０は、オペレーティングシステムの制御下で、ネットワーク接続１４０またはパス３１０を介して、外部ソースとコマンドおよびデータを交換して、プログラム実行中に通常はユーザから要求される情報を送受信する。
【００４２】
入力インタフェース３２０はまた、キーボードやマウスなどのユーザ入力装置３９５を、コンピュータシステム１００に電気的に接続およびインタフェースする。従来のカラーモニタなどのディスプレイ３８０、および従来のレーザプリンタなどのプリンタ３８５は、それぞれリード３６３および３６７を介して、出力インタフェース３６０に接続される。出力インタフェースは、ディスプレイおよびプリンタをコンピュータシステムに電気的に接続およびインタフェースするのに、必要な回路を備える。理解できるように、我々のこの発明の暗号技術は、クライアントコンピュータ１００が情報を入手し、記憶し、および／または通信するモダリティに関係なく、どんなタイプのディジタル情報でも動作することができる。
【００４３】
さらに、コンピュータシステム１００の具体的なハードウェアコンポーネントならびにメモリ３３５内に記憶されるソフトウェアのすべての態様は、本発明を実装するモジュールは別として、従来型かつ周知であるので、これ以上詳細には論じない。概して言えば、コンピュータ２０５は、クライアントコンピュータ１００と非常に類似するアーキテクチャを有する。
【００４４】
Ｄ．従来の暗号技術におけるモジュロ算法によって生じる制限
従来の暗号技術は、ｍｏｄ（Ｍ）の計算を必要とするチェックサムを、プリミティブとして頻繁に採用し、Ｍは例えば$2^{31}-1$やそれ以上などの、大きな素数である。
【００４５】
残念なことに、ｍｏｄ（Ｍ）演算は、これより多くはならないとしても、少なくとも約１０～１５マシンサイクルが計算に必要である（法の値Ｍに基づく）。この関数は、従来の暗号化演算中と復号演算中の両方で繰り返し計算される。したがって、ＰＣやワークステーションなど、大きな処理能力を有する機器上にこのような技術を実装した場合、ｍｏｄ（Ｍ）計算は、スループット全体をおそらく顕著に低下させるであろう。しかし、やや限られた処理能力しか有さない機器内では、この計算オーバーヘッドは耐えられない場合があり、したがって、この暗号技術の使用が非常に有益となる可能性のあるこれらの機器内で、この暗号技術を使用することが阻まれる。
【００４６】

Ｅ．我々の発明的技術およびその実施
当技術分野におけるこの欠陥を認識し、有利にもｍｏｄ（Ｍ）演算を必要としないチェックサム実施技術を開発した。
【００４７】
我々の技術は、チェックサムを比較的単純な一連の基本的レジスタ演算として実施する。これらの演算は、ｍｏｄ２^{n}乗算、順序操作（ブロック中のビット順序付けを変更する演算、例えばバイトスワップやワードスワップなど）、および加算を含み、これらはすべて実施が非常に単純であり、実行するのにごくわずかな処理サイクルしか必要としない。プリミティブ中で用いる演算はまた、いくぶん効果的にパイプライン化することもできる。したがって、我々の発明に基づくプリミティブを使用すると、特にパイプライン化されている場合には、様々な暗号パラメータ、例えばメッセージ認証コード（ＭＡＣ）を計算するため、およびストリーム暗号を実装するために必要な処理時間を、従来必要とされていた処理時間に打ち勝って、大きく短縮することができる。我々の発明的技術はまた、有利にも、ある暗号に組み込んで、何らかの平文－暗号文の攻撃に対して、それらの暗号のセキュリティを高めることもできる、と考えられる。
【００４８】
以下の数学定義Ｆ（ｘ）＝θから始めるが、添字「Ｓ」、すなわちｘ^{S}におけるＳは、適切なバイトスワップ演算またはワードスワップ演算のいずれかを示す。
【００４９】
やや本題から外れるが、図６Ａおよび６Ｂに、ワードスワップ演算およびバイトスワップ演算をそれぞれ示す。２つの１６ビットワード（例えばワード６１３および６１７。それぞれ「左」および「右」にあたるＬおよびＲの符号も付けてある）を有するｎビットのブロック６１０（例示的に３２ビット長）を仮定した場合、線６２０で表すワードスワップ演算により、場所が入れ替わったこれらのワード（すなわちワード６１７および６１３とそれぞれ同一のワード６３３および６３７）を有するｎビットのブロック６３０が生み出される。このような演算は、矢印６２５で示すように単に個々のワードを交換するだけで、１つの処理サイクルで実施することができる。個別の８ビットバイト６５２、６５４、６５６、６５８（やはりそれぞれバイトＡ、Ｂ、Ｃ、Ｄとして符号をつける）を有するｎビットのブロック６５０（やはり例示的に３２ビット長）を仮定した場合、線６６０で表すバイトスワップ演算により、順番に反転されたこの４つのバイト（すなわちバイト６５８、６５６、６５４、６５２とそれぞれ同一のバイト６７２、６７４、６７６、６７８）を有するｎビットのブロック６７０が生み出される。このバイトスワップ演算は、矢印６６５で示すように個々のバイトを並列に交換することにより、１つの処理サイクルで実施することができる。
【００５０】
これらの定義を念頭に置いた場合、非可逆バージョンのプリミティブＦ（ｘ）は、我々の発明的な教示により、以下の式（１）～（５）を順番に計算することによってチェックサム、具体的にはｆ（ｘ）＝ａｘ＋ｂｍｏｄ（Ｍ）を実施する。

$x^{S} \leftarrow wordswap(x)$ （１）
$y \leftarrow Ax + Bx^{S} \bmod (2^{n})$ （２）
$y^{S} \leftarrow wordswap(y)$ （３）
$z \leftarrow Cy^{S} + yD \bmod (2^{n})$ （４）
$\theta \leftarrow z + y^{S}E \bmod (2^{n})$ （５）

上式で、係数Ａ、Ｂ、Ｃ、Ｄ、Ｅは、それぞれ２^{n}以下のランダムな奇数の整数であり、θはｎビットのストリングである。
【００５１】
見れば分かるように、これらの式は、基本的なレジスタ演算すなわち順序操作（例えばワードスワップおよびバイトスワップ、加算、ｍｏｄ（２^{n}）乗算）を用いて実施される。したがって、これらの演算は比較的わずかな処理サイクルを用いて実施することができ、これはｍｏｄ（Ｍ）演算を行うのに必要な１０～１５サイクルよりも確実にかなり少ない

。式（１）および（３）はワードスワップ演算を用いて示したが、代わりにバイトスワップ演算（または場合によっては、ビット順序付けを変更する他の操作）を用いることもできる。係数の値Ａ、Ｂ、Ｃ、Ｄ、Ｅは、ＭＡＣまたは他の様々な暗号タームを生成する際に使用する場合は、「秘密の」値すなわち非公開の値である。

【００５２】

可逆バージョンのプリミティブＦ（ｘ）は、やはり我々の発明的教示により、以下の式（６）～（１５）を通してｆ（ｘ）を実施する。

$y \leftarrow Ax \bmod (2^n)$ （６）

$y^S \leftarrow wordswap(y)$ （７）

$z \leftarrow By^S \bmod (2^n)$ （８）

$z^S \leftarrow wordswap(z)$ （９）

$v \leftarrow Cz^S \bmod (2^n)$ （１０）

$v^S \leftarrow wordswap(v)$ （１１）

$w \leftarrow Dv^S \bmod (2^n)$ （１２）

$w^S \leftarrow wordswap(w)$ （１３）

$t \leftarrow Ew^S \bmod (2^n)$ （１４）

$\theta \leftarrow t + Ly^S \bmod (2^n)$ （１５）

上式で、係数Ａ、Ｂ、Ｃ、Ｄ、Ｅは、それぞれ2^n以下のランダムな奇数の整数であり、Ｌは2^n以下のランダムな整数である。

【００５３】

ここでもまた、ＭＡＣまたは他の様々な暗号タームを生成するときは、係数の値Ａ、Ｂ、Ｃ、Ｄ、Ｅ、Ｇはすべて「秘密の」値である。別法として、式（６）～（１２）を使用してＦ（ｘ）＝ｗでプリミティブを実施することもできる。さらに、別のタイプの順序操作である「反転」演算（ブロック中のすべてのビットが順番に完全に反転される）をバイトスワップまたはワードスワップの代わりに用いることもできる。例えば、我々の発明により、以下の式（１６）～（１９）を通して可逆形式のｆ（ｘ）に対する、プリミティブＦ（ｘ）を実施することもできる。

$y \leftarrow Hx \bmod (2^n)$ （１６）

$z \leftarrow reverse(y)$ （１７）

$s \leftarrow Jz \bmod (2^n)$ （１８）

$\theta \leftarrow s + K \bmod (2^n)$ （１９）

上式で、係数Ｈ、Ｊ、Ｋはそれぞれ2^n以下のランダムな整数である。

【００５４】

ＭＡＣまたは他の暗号タームを生成するのにこのプリミティブを使用する場合は、係数Ｈ、Ｊ、Ｋは「秘密の」値となる。反転演算はバイトスワップまたはワードスワップ演算と比較して相対的に遅いので、式（１６）～（１９）によって得られるプリミティブよりも、上記の式（６）～（１２）または（６）～（１５）によって得られるプリミティブを使用する方が好ましい。

【００５５】

以上の記述に基づけば、ｆ（ｘ）＝ａｘ＋ｂｍｏｄ（Ｍ）に対して等価な暗号特徴を備え、$mod\ 2^n$乗算、順序操作、および加算を本発明により利用するがｍｏｄ（Ｍ）演算は利用せず、したがって前述の具体的なプリミティブの代わりとすることのできる、他の様々なプリミティブＦ（ｘ）を、当業者なら容易に案出することができることは明らかである。

【００５６】

先に論じたように、我々の発明的技術に基づく一般化されたプリミティブを使用して、ＭＡＣを生成することができる。そうするためには、非可逆であり前述と同じ形（Ｆ（ｘ））の、関数ｆ（ｘ）に対する一連のプリミティブ$F_1(x)$，$F_2(x)$，．．．，$F_p(x)$が選択されるが、これらは対応する「秘密の」係数の値が異なる。すなわち、$F_1(x)$が「秘密の」係数Ａ、Ｂ、Ｃ、Ｄ、Ｅを有する場合は、$F_2(x)$は「秘密の」係数

ａ、ｂ、ｃ、ｄ、ｅを有する等である。その後、ｎビットのストリングの入力シーケンス$X = x_1, x_2, \ldots, x_N$が与えられた場合、これらのｐ個のプリミティブ（ただしｐ＜ｎ）のうちの連続したプリミティブを、対応する連続した入力値x_iに使用して、対応する出力値（中間結果）$Y = y_1, y_2, \ldots, y_N$が以下の式（２０）～（２５）に従って計算される。

$y_1 = F_1(x_1)$ （２０）
$y_2 = F_2(x_2 + y_1)$ （２１）
$y_3 = F_3(x_3 + y_2)$ （２２）
$y_p = F_p(x_p + y_{p-1})$ （２３）
.
.
.
$y_{p+1} = F_1(y_p + x_{p+1})$ （２４）
$y_{p+2} = F_2(y_{p+1} + x_{p+2})$ （２５）
.
.
.

【００５７】
この中間結果の関数として、以下の式（２６）に従って、ＭＡＣを形成することができる。

【００５８】
【数７】

$$MAC = (y_N, \sum_{i=1}^{N} y_i) \qquad (26)$$

【００５９】
追加のセキュリティのために、以下の式（２７）によって示すように、合計の中の各y_i項に対して秘密またはランダムの置換（γ_i）を導入することにより、式（２５）を修正することができる。

【００６０】
【数８】

$$MAC = (y_N, \sum_{i=1}^{N} \gamma_i y_i) \qquad (26)$$

【００６１】
上式で、γ_iは、ｋを含めた±ｋの範囲内すなわち$\gamma_i \in \{k, k-1, k-2, \ldots, 0, -1, -2, \ldots, -k\}$で、ランダムにまたは「秘密の」事前定義値として選択され、ｋは事前定義済みの整数である。簡単にするために、各γ_iは値＋１または－１に設定してもよく、そのようなすべてのγ_iのうちでランダムな、擬似ランダムな、または「秘密の」事前定義済みの変動を伴う。

【００６２】
式（２０）～（２５）は同じｐ個のプリミティブを繰り返した連続を利用しているが、代わりに、このような連続で異なるものを使用してもよい。関数の各連続は別々の出力ハッシュ値ｙを生成し、次いでこれを共に連結させてＭＡＣ形成することができ、あるいは式（２６）を用いて各プリミティブの個別の出力ｙを合計してＭＡＣ値を生み出すことができる。さらに、例えば式（２０）～（２３）によって示すように、ある連続を前向き連鎖で実行することもできる。例えば式（２４）および（２５）によって示すものなど、同じ連続の次の実行を、または異なる連続の次の実行を「逆方向」連鎖で実行することもできる。逆方向連鎖を使用する場合、関連する入力値は、前向き連鎖で使用した入力値に対して、逆の順番で、その連続の中にある個々のプリミティブに加えることができる。

【００６３】
我々の発明的技術を使用してチェックサムを計算する場合、この計算は、ＭＡＣの計算に用いられる計算と同一とは言わないまでも非常に類似するが、すべての係数の値、ならびに、使用するならば、すべてのγ_iの値は、公開される。
【００６４】
以上を念頭に置きながら、次に、暗号化プロセス２０によって使用される、かつ、我々の発明的技術を実装したプリミティブに従って、ＭＡＣを生成するのに必要なソフトウェアの説明に移る。
【００６５】
図４に、図１に示したプロセス５でＭＡＣの生成に用いるＭＡＣ生成プロセス４００の高レベルのフローチャートを示す。このルーチンは、プリミティブＦ（ｘ）およびＧ（ｘ）が完全に選択されていると仮定して、先に論じた式（２０）～（２５）を実施する。
【００６６】
具体的には、暗号化プロセス２０と復号プロセス５０のいずれかの実行中にルーチン４１０に入ると、図４に示すようにまずブロック４１０に進む。このブロックは、ポインタ（ｉ）を１に、かつ合計変数（y_s）を０に初期化する。その後、ブロック４２０、４３０、４４０、および４５０で形成されるループに入って、入力された入力平文ブロック（P_i）ごとに連続する出力値y_iを計算し、これらの出力値を合計変数y_sに累積する。
【００６７】
具体的には、このループに入るとまずブロック４２０に進んで、Ｆ（P_1）に等しい出力値y_1を計算する。これを行った後で、合計計算プロシージャ４３０に進み、そこでブロック４３５を介して単純に出力値y_iを合計変数y_sに加算する。これを行った後で、判断ブロック４４０に進んで、入力平文メッセージＰのＮ個のブロックをすべて処理したかどうか、すなわちポインタｉの現在の値がこのときＮに等しいかどうかを判定する。このようなブロックが残っている場合、すなわちｉの現在の値がＮ未満の場合は、判断ブロック４４０はＮＯパス４４３を介してブロック４５０に実行をルーティングする。この後者のブロックは、ポインタｉの値を１つインクリメントし、次いでフィードバックパス４５５を介してブロック４２０に実行が戻されて、次の連続する出力値が計算され、以下同様である。この時点では、ブロック４２０によって実施される計算は、ブロック４２０を介する所与の反復について、ｉの値がその場合に偶数か奇数かによって決まる。したがって、連続するｉに対してプリミティブＦ（ｘ）とＧ（ｘ）が交互になる。
【００６８】
すべての出力値を計算して合計した後で、判断ブロック４４０はＹＥＳパス４４７を介してブロック４６０に実行をルーティングする。この後者のブロックは、値y_Nを合計変数の現在の値と連結させて、得られたＭＡＣとしての６４ビット値を出力として供給することにより、単純にＭＡＣを形成する。これを行った後で、ルーチン４００を終了する。
【００６９】
図５に、ＭＡＣ生成プロセス４００の一部をなす合計計算プロシージャ４３０の代わりに用いることのできる代替の合計計算プロシージャ５００の、高レベルのフローチャートを示す。プロシージャ５００は、上記の式（２６）を実施する。
【００７０】
具体的には、プロシージャ５００に入るとまずブロック５１０に進み、ここでγ_iの値を適切に設定する。前述のようにこの値は、±ｋ（ただし通常は±１の値を使用する）の範囲内でランダム、擬似ランダム、または事前定義済みとすることができる。この値を設定した後で、ブロック５２０に進み、ここで現在の出力値y_iに、対応するγ_iの値を掛け、得られた値を合計変数y_sに加算する。これを行った後で、プロシージャ５００を終了する。
【００７１】
ＭＡＣ（またはチェックサム）を６４ビット長すなわち２つの３２ビットブロックとして述べたが、この代わりに、単一の３２ビットブロックや６４ビットよりも長いもの（ただ

し整数ブロックのサイズのもの）など、他のビット（およびブロック）サイズのＭＡＣ（またはチェックサム）を使用することもできることを、明らかに当業者なら理解するだろう。ＭＡＣを生成するためと、必要な場合に、それを暗号化および復号するために、増加した処理時間がかかることが見込まれるものの、ＭＡＣが大きいほど、保証される限度までの、より高いレベルのセキュリティがもたらされる。

【００７２】

本発明の教示を組み込んだ詳細な実施形態について、いくつかの変形と共に本明細書に詳細に示し、説明したが、これらの教示をやはり利用した本発明の他の多くの実施形態および適用例を、当業者なら容易に案出することができる。

【図面の簡単な説明】

本発明の教示は、添付の図面とともに詳細な説明を考察することによって容易に理解することができる。

【図１】　本発明の教示を利用して、例示的にメッセージ認証コード（ＭＡＣ）を生成する、端末相互間の暗号プロセス５全体のブロック図である。

【図２】　例示的に本発明を利用する、典型的なインターネットベースのクライアントサーバ処理環境の、高レベルのブロック図である。

【図３】　図２に示したクライアントコンピュータ１００のブロック図である。

【図４】　我々の本発明の教示によってＭＡＣを生成するために、図１に示したプロセス５で用いられる、ＭＡＣ生成プロセス４００の高レベルのフローチャートである。

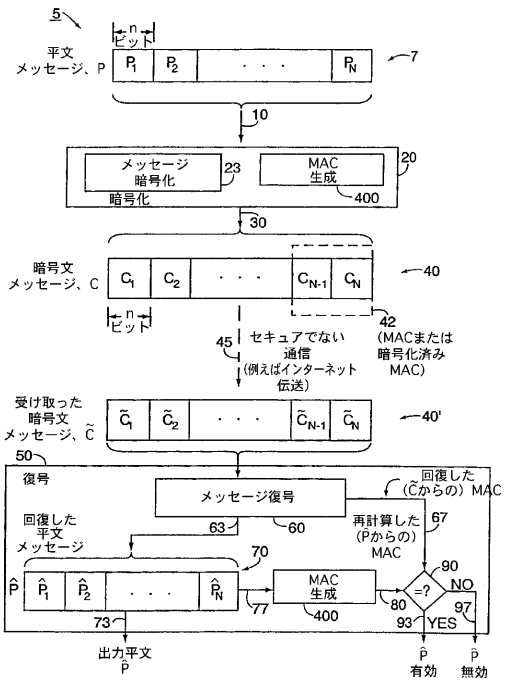
【図５】　図４に示したＭＡＣ生成プロセス４００の一部をなす、合計計算プロシージャ４３０の代わりに用いることのできる、代替の合計計算プロシージャ５００の、高レベルのフローチャートである。

【図６Ａ】　我々の本発明によって採用できる典型的なワードスワップ演算を示す図である。

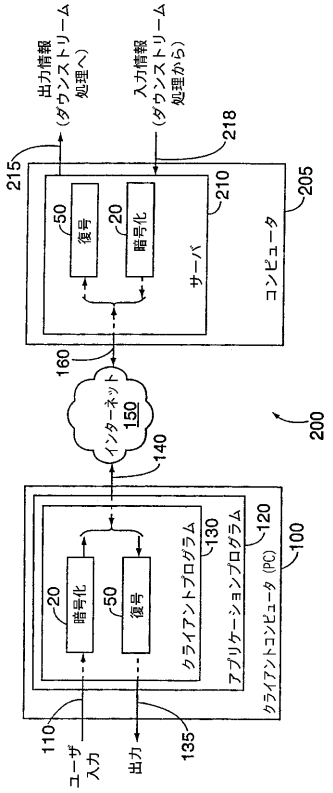
【図６Ｂ】　我々の本発明によって採用できる典型的なバイトスワップ演算を示す図である。

理解を容易にするために、各図に共通の同一要素を指すためには、可能な限り、同一の参照番号を使用している。

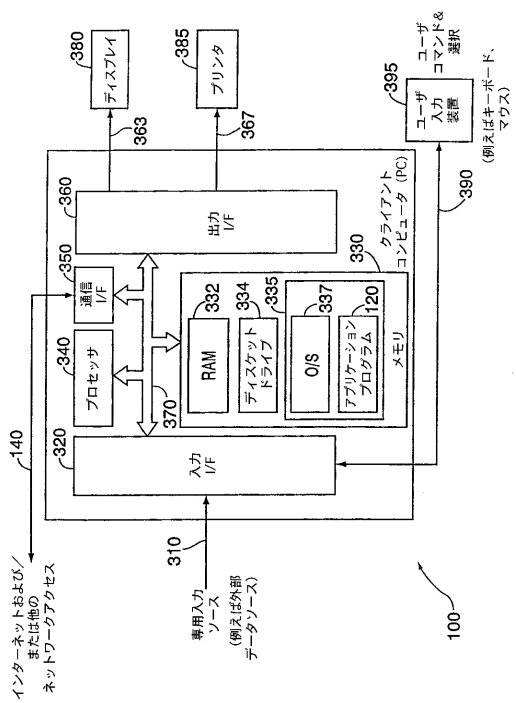
【図１】


平文
メッセージ、P
nビット
メッセージ
暗号化
MAC
生成
暗号化
暗号文
メッセージ、C
セキュアでない
通信
（例えばインターネット
伝送）
（MACまたは
暗号化済み
MAC）
受け取った
暗号文
メッセージ、C~
復号
メッセージ復号
回復した
（C~からの）MAC
回復した
平文
メッセージ
再計算した
（Pからの）
MAC
MAC
生成
=?
NO
YES
出力平文
P^
有効
無効

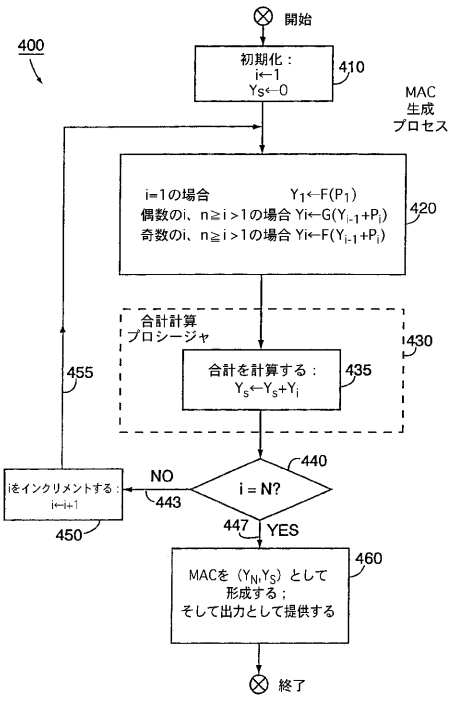
【図２】


出力情報
（ダウンストリーム
処理へ）
入力情報
（ダウンストリーム
処理から）
復号
暗号化
サーバ
コンピュータ
インターネット
クライアントプログラム
アプリケーションプログラム
クライアントコンピュータ（PC）
ユーザ
入力
出力

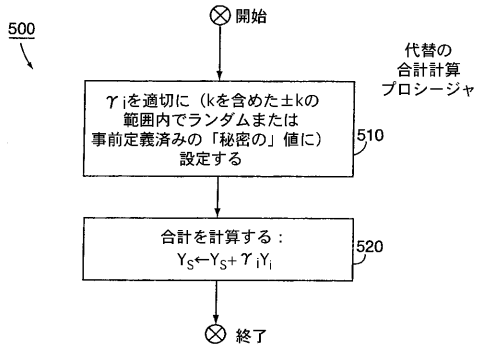
【図３】


ディスプレイ
プリンタ
ユーザ
コマンド&
選択
ユーザ
入力
装置
（例えばキーボード、
マウス）
出力
I/F
通信
I/F
プロセッサ
RAM
ディスケット
ドライブ
O/S
アプリケーション
プログラム
メモリ
クライアント
コンピュータ（PC）
入力
I/F
インターネットおよび/
または他の
ネットワークアクセス
専用入力
ソース
（例えば外部
データソース）

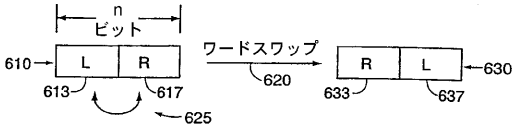
【図４】


開始
初期化：
i←1
Ys←0
MAC
生成
プロセス
i=1の場合 Y1←F(P1)
偶数のi、n≧i>1の場合 Yi←G(Yi-1+Pi)
奇数のi、n≧i>1の場合 Yi←F(Yi-1+Pi)
合計計算
プロシージャ
合計を計算する：
Ys←Ys+Yi
i=N?
NO
YES
iをインクリメントする：
i←i+1
MACを（YN,YS）として
形成する；
そして出力として提供する
終了

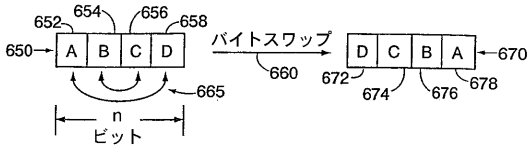
【図５】


500
開始
代替の
合計計算
プロシージャ
γiを適切に（kを含めた±kの
範囲内でランダムまたは
事前定義済みの「秘密の」値に）
設定する
510
合計を計算する：
YS←YS+γiYi
520
終了

【図６Ａ】


n
ビット
610
L
R
613
617
625
ワードスワップ
620
R
L
630
633
637

【図６Ｂ】


652
654
656
658
650
A
B
C
D
665
n
ビット
バイトスワップ
660
D
C
B
A
670
672
674
676
678

フロントページの続き

(72)発明者　マリウス　エイチ. ジャクボウスキー
アメリカ合衆国　９８００７　ワシントン州　ベルビュー　１５４　アベニュー　ノースイースト
１８４０　アパートメント　シー－２２２

審査官　石田　信行

(56)参考文献　特開２０００－１２２５３８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G09C　1/00