(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5511803号
(P5511803)
(45)発行日　平成26年6月4日(2014.6.4)　(24)登録日　平成26年4月4日(2014.4.4)

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

請求項の数 13　(全 20 頁)

(21)出願番号　特願2011-513515(P2011-513515)
(86)(22)出願日　平成21年4月21日(2009.4.21)
(65)公表番号　特表2011-523103(P2011-523103A)
(43)公表日　平成23年8月4日(2011.8.4)
(86)国際出願番号　PCT/US2009/041225
(87)国際公開番号　WO2009/151793
(87)国際公開日　平成21年12月17日(2009.12.17)
審査請求日　平成24年4月23日(2012.4.23)
(31)優先権主張番号　12/136,772
(32)優先日　平成20年6月11日(2008.6.11)
(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　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行
(74)代理人　100153028
弁理士　上田　忠

最終頁に続く

(54)【発明の名称】対称暗号を実行するための技法

(57)【特許請求の範囲】
【請求項１】
鍵特定解読エンジンを発生する方法であって、
対称暗号技法の解読動作を実行するように構成された解読エンジンのリファレンス実装を送信側のコンピューターに提供するステップであって、前記解読エンジンのリファレンス実装は対称鍵を第１入力として暗号化されたメッセージを第２入力として受け取るように構成されており、前記暗号化されたメッセージは前記対称鍵を用いて先に暗号化された元のメッセージの暗号化されたバージョンである、ステップと、
前記リファレンス実装を用いて、前記第２入力に対して指定された異なる値に対応する期待出力を前記送信側コンピューターによって決定するステップであって、前記期待出力は前記暗号化されたメッセージに対応する前記元のメッセージを前記異なる値の中の対応する１つの値として含む、ステップと、
前記第１入力としての前記対称鍵と前記第２の入力として前記暗号化されたメッセージを含む前記異なる値とに基づいて、前記元のメッセージを含む前記期待出力を発生するように構成された１つ又は複数のブール関数を前記送信側コンピューターによって決定するステップと、
前記決定するステップに基づいて、前記１つ又は複数のブール関数を含む前記鍵特定解読エンジンを前記送信側コンピューターによって発生するステップと、
前記発生された鍵特定解読エンジンを前記送信側コンピューターによって受信側コンピューターに提供するステップであって、前記提供された鍵特定解読エンジンを含む前記受

信側コンピューターは、解読のために前記対称鍵を前記提供された鍵特定解読エンジンへの入力として要求することなく、前記暗号化されたメッセージを解読するように構成されている、ステップと、
を含む、方法。
【請求項２】
請求項１記載の方法において、前記１つ又は複数のブール関数はそれぞれが入力の１つ又は複数のビットに基づいて１つのビットを出力として発生し、前記出力は前記ブール関数の中の別のブール関数への入力か又は前記鍵特定解読エンジンの出力かのいずれかであり、前記入力の１つ又は複数のビットはそれぞれが暗号化されたメッセージの１ビットか又は前記ブール関数の中の別のブール関数によって発生された出力かのいずれかである、方法。
【請求項３】
請求項１記載の方法において、前記１つ又は複数のブール関数は、前記期待出力と前記第２入力に対して指定された異なる値とに応じて作成された真理値表を用いて決定される、方法。
【請求項４】
請求項１記載の方法において、１つ又は複数のブール関数を決定する前記ステップは複数の区画を決定するステップを含み、それぞれの区画は前記ブール関数の一部に対応する、方法。
【請求項５】
請求項４記載の方法において、前記複数の区画の中の第１区画の出力は前記複数の区画の中の第２区画への入力である、方法。
【請求項６】
請求項５記載の方法において、前記複数の区画はそれぞれが暗号ラウンドに対応する、方法。
【請求項７】
請求項５記載の方法において、前記複数の区画はそれぞれが前記対称暗号技法のために解読動作を実行することに関連して実行される処理の繰り返しに対応する、方法。
【請求項８】
請求項６記載の方法において、それぞれの暗号ラウンドは複数の段階を含む、方法。
【請求項９】
請求項３記載の方法であって、
前記１つ又は複数のブール関数に対応する積の和を前記真理値表に従って決定するステップを更に含む、方法。
【請求項１０】
請求項３記載の方法であって、
前記１つ又は複数のブール関数に対応する和の積を前記真理値表に従って決定するステップを更に含む、方法。
【請求項１１】
コンピューター実行可能な命令が記憶されたコンピューター可読記憶媒体であって、前記コンピューター実行可能な命令は実行されるとコンピューターに鍵特定解読エンジンを発生する方法を実行させる、コンピューター可読記憶媒体であって、前記方法は、
対称暗号技法の解読動作を実行するように構成された解読エンジンのリファレンス実装を送信側のコンピューターに提供するステップであって、前記解読エンジンのリファレンス実装は対称鍵を第１入力として暗号化されたメッセージを第２入力として受け取るように構成されており、前記暗号化されたメッセージは前記対称鍵を用いて先に暗号化された元のメッセージの暗号化されたバージョンである、ステップと、
前記リファレンス実装を用いて、前記第２入力に対して指定された異なる値に対応する期待出力を前記送信側コンピューターによって決定するステップであって、前記期待出力は前記暗号化されたメッセージに対応する前記元のメッセージを前記異なる値の中の対応

する１つの値として含む、ステップと、
前記第１入力としての前記対称鍵と前記第２の入力として前記暗号化されたメッセージを含む前記異なる値とに基づいて、前記元のメッセージを含む前記期待出力を発生するように構成された１つ又は複数のブール関数を前記送信側コンピューターによって決定するステップと、
前記決定するステップに基づいて、前記１つ又は複数のブール関数を含む前記鍵特定解読エンジンを前記送信側コンピューターによって発生するステップと、
前記発生された鍵特定解読エンジンを前記送信側コンピューターによって受信側コンピューターに提供するステップであって、前記提供された鍵特定解読エンジンを含む前記受信側コンピューターは、解読のために前記対称鍵を前記提供された鍵特定解読エンジンへの入力として要求することなく、前記暗号化されたメッセージを解読するように構成されている、ステップと、
を含む、コンピューター可読記憶媒体。
【請求項１２】
請求項１１記載のコンピューター可読記憶媒体において、前記鍵特定解読エンジンは、前記対称暗号技法のための解読動作を選択された対称鍵を用いて実行した場合の期待出力に従って、和、積又は和の積に対応する複数のブール関数を含む、コンピューター可読記憶媒体。
【請求項１３】
請求項１２記載のコンピューター可読記憶媒体において、前記鍵特定解読エンジンは、Ｎビットの暗号化されたメッセージを解読し、Ｎビットの解読されたメッセージを出力として生成し、前記鍵特定解読エンジンは複数のブール関数を含み、前記複数のブール関数はそれぞれが前記暗号化されたメッセージの前記Ｎビットの中の複数個を用いて、前記解読されたメッセージの１ビットを決定する、コンピューター可読記憶媒体。
【発明の詳細な説明】
【背景技術】
【０００１】
[0001] 送信側および受信側間でメッセージを伝達することができる。送信側および受信側は、例えば、２台のコンピューター・システム、同じコンピューター・システム上にある２つのモジュール等であってもよい。送信側および受信側の双方は、交換するメッセージの安全性および認証に関して関わりがある場合がある。受信されたメッセージが、元のメッセージを変更したバージョンでないことを保証するため、そして送信側のアイデンティティを検証するために、異なる技法を用いることができる。１つの手法は、メッセージを送るときにディジタル署名を用いることである。ディジタル署名のための既存の技法は、非対称暗号を用いる。この場合、ＰＫＩ（公開鍵インフラストラクチャ）を用いる公開／個人鍵対の使用というように、異なる鍵が暗号化および解読に用いられる。メッセージを送るとき、暗号ハッシュ・アルゴリズムを用いるというようにして、メッセージ・データまたはコンテンツのハッシュ値を発生することができる。送信側の個人鍵を用いてハッシュ値を暗号化し、メッセージのディジタル署名を生成することができる。メッセージ・データおよびそのディジタル署名は、受信側に送信される。次いで、受信側は送信側の公開鍵を用いてディジタル署名を解読し、送信側から送られたハッシュ値を明らかにする。次に、受信側は、送信されたメッセージ・データを用い送信側と同じハッシュ・アルゴリズムを適用して期待ハッシュ値を計算することによって、検証処理を実行することができる。次いで、受信側は、期待ハッシュ値を、解読によって生成したハッシュ値と比較することができる。両ハッシュ値が一致した場合、受信側は、メッセージ・データが変化していないこと、および送信側がこの受信したメッセージを発信したことを結論付けることができる。非対称暗号技法によるディジタル署名を用いることの欠点の１つは、ディジタル署名が、メッセージ・データと比較すると、比較的大きい場合があることである。つまり、ディジタル署名は、例えば、送信されるメッセージがサイズの制約を受ける用途における場合にあり得るように、送信されるメッセージのサイズを、容認できない量だけ増大

させる虞れがある。別の技法には、対称暗号化を用いるものがあり、この場合、送信側が行う暗号化、および受信側が行う解読に同じ鍵が用いられる。これに伴う１つの欠点は、例えば、鍵を読み取り可能な形態で、および／またはアクセス可能な場所に格納することもできるので、鍵が他人に容易に入手される虞れがあることであり、これによって、メッセージの改竄を検出し、特定の送信側から送られたメッセージの認証を検証する等の受信側の能力に悪影響を及ぼす可能性がある。

【発明の概要】

【０００２】

[0002] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を判断する際に補助として用いられることを意図するのでもない。

【０００３】

[0003] 対称鍵毎に発生する鍵特定解読エンジンを用いて、解読を行うための技法について記載する。暗号化データ部分を含むメッセージを受信する。暗号化データ部分は、対称鍵を用いて対称暗号化動作を実行することによって形成される。この暗号化データ部分を、鍵特定解読エンジンを用いて解読する。鍵特定解読エンジンは、対称鍵を入力として用いない。鍵特定解読エンジンは、対称鍵について決定されたブール関数を用いて実装することができる。

【図面の簡単な説明】

【０００４】

[0004] 本発明の特徴および利点は、添付図面と合わせて行われる以下の実施形態例の詳細な説明から、一層明らかとなろう。図面において、

【図１】図１は、本明細書において記載する技法に関連して一実施形態において利用することができる環境の一例である。

【図２】図２は、一実施形態に含まれ、本明細書における技法の実行に関連して用いることができるコンポーネントの一例である。

【図３】図３は、本明細書における技法の実行に関連して用いられる鍵特定解読エンジンの一実施形態を示す別の例である。

【図４】図４は、対称鍵を入力として用いる、選択された対称鍵技法に合わせた解読エンジンのリファレンス実装を示す例である。

【図５】図５は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図６】図６は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図７】図７は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図８】図８は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図９】図９は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図１０】図１０は、鍵特定解読エンジンのブール関数を決定する際に用いることができる技法を示す例である。

【図１１】図１１は、本明細書における技法を用いる一実施形態において行うことができる処理ステップのフローチャートである。

【図１１】図１１は、本明細書における技法を用いる一実施形態において行うことができる処理ステップのフローチャートである。

【図１２Ａ】図１２Ａは、本明細書における技法を用いる一実施形態において行うことができる処理ステップのフローチャートである。

【図１２Ｂ】図１２Ｂは、本明細書における技法を用いる一実施形態において行うことが

できる処理ステップのフローチャートである。
【図１３】図１３は、本明細書における技法を用いる一実施形態において行うことができる処理ステップのフローチャートである。
【発明を実施するための形態】
【０００５】
[0011] 図１を参照すると、本明細書において記載する技法を利用する実施形態を実現することができる、適したコンピューティング環境の一例が示されている。図１に示すコンピューティング環境は、適したコンピューティング環境の一例に過ぎず、本明細書において記載する技法の使用または機能の範囲に関して、何の限定を示唆することも意図していない。当業者には、本明細書において記載する技法が、他の汎用および特殊目的コンピューティング環境ならびに構成と共に用いるのにも適していると考えられることは認められよう。周知のコンピューティングシステム、環境、および／または構成の例には、限定ではないが、パーソナル・コンピューター、サーバー・コンピューター、ハンドヘルドまたはラップトップ・デバイス、マルチプロセッサー・システム、マイクロプロセッサー主体システム、プログラマブル消費者電子機器、ネットワークＰＣ、ミニコンピューター、メインフレーム・コンピューター、以上のシステムまたはデバイスの内いずれでも含む分散型コンピューティング環境等が含まれる。
【０００６】
[0012] 本明細書において明記する技法は、１つ又は複数のコンピューターまたはその他のデバイスによって実行するプログラム・モジュールのような、コンピューター実行可能命令という一般的なコンテキストで説明することができる。一般に、プログラム・モジュールは、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造等を含み、特定のタスクを実行するかまたは特定の抽象的データ・タイプを実現する。通例、プログラム・モジュールの機能は、種々の実施形態において説明するように、組み合わせることまたは分散させることもできる。
【０００７】
[0013] 図１に含まれるのは、コンピューター１２、ネットワーク１４、およびサーバー１６である。コンピューター１２は、標準的な市販のコンピューターまたは特殊目的コンピューターを含むことができ、これらは１つ又は複数のプログラム・モジュールを実行するために用いることができる。以下の節および図において更に詳しく説明するのは、本明細書において記載する技法を用いて受信したメッセージを解読することに関連して、コンピューター１２によって実行することができるプログラム・モジュールである。コンピューター１２は、ネットワーク型環境において動作し、サーバー１６、そして図１には示されていないその他のコンピューターのような、他のコンピューターと通信することができる。
【０００８】
[0014] 尚、前述の例ではコンピューター１２はネットワーク型環境において通信するように示されているが、コンピューター１２は、異なる通信媒体を利用して、他のコンポーネントと通信することもできることは、当業者には認められよう。例えば、コンピューター１２は、ネットワーク接続、および／または当技術分野では周知のその他のタイプのリンクを利用して、１つ又は複数のコンポーネントと通信することができる。その他のタイプのリンクには、限定ではないが、インターネット、イントラネット、あるいはその他のワイヤレスおよび／または有線接続（１つまたは複数）が含まれる。
【０００９】
[0015] 図１に示すように、コンピューター１２は、１つ又は複数の演算装置２０、メモリー２２、ストレージ３０、およびコンピューター１２のコンポーネントと図１には示されていないその他のコンポーネントとの間における通信をし易くするために用いられるシステム・バス３２を含むことができる。
【００１０】
[0016] コンピューター１２の構成およびタイプに応じて、メモリー２２は、揮発性（

ＲＡＭのような）、不揮発性（ＲＯＭ、フラッシュ・メモリー等のような）、またはこれら２つの何らかの組み合わせとすることができる。加えて、コンピューター１２は、追加のストレージ（リムーバブルおよび／または非リムーバブル）も有することができる。追加のストレージには、限定ではないが、ＵＳＢデバイス、磁気または光ディスク、あるいはテープが含まれる。このような追加のストレージは、図１ではストレージ３０によって例示されている。ストレージ３０は、１つ又は複数のリムーバブルおよび非リムーバブル記憶デバイスを含むことができ、コンピューター１２が利用することができるコンピューター読み取り可能媒体と関連付けられている。一実施形態では、ストレージ３０は、ハード・ディスクおよび／またはＣＤ－ＲＯＭドライブを含むことができる。一例として、そして限定ではなく、メモリー２２およびストレージ３０はコンピューター読み取り可能媒体の例である。コンピューター読み取り可能媒体は、揮発性および不揮発性、リムーバブルおよび非リムーバブル媒体を含み、コンピューター読み取り可能命令、データ構造、プログラム・モジュール、またはその他のデータのような情報の格納のために、いずれの方法または技術でも実現される。コンピューター読み取り可能媒体は、限定ではないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリーまたはその他のメモリー技術、ＣＤ－ＲＯＭ、（ＤＶＤ）またはその他の光ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージまたはその他の磁気記憶デバイス、あるいは所望の情報を格納するために用いることができ、そしてコンピューター１２によってアクセスすることができるのであればその他のいずれの媒体でも含まれる。以上の媒体は、通例、コンピューター読み取り可能命令、データ構造、プログラム・モジュール、またはその他のデータを具体化する。

【００１１】

[0017] 本明細書に記載する一実施形態では、コンピューター１２は、図１に示すようなネットワーク型環境において、ネットワークを通じて、サーバー１６のようなリモート・コンピューターへの論理接続を用いて動作することができる。１つ又は複数のソフトウェア・モジュールおよび／またはデータ・ファイルは、コンピューター１２のストレージ３０に含めることができる。コンピューター１２の動作中、ストレージ３０に含まれるこれらのモジュールの１つ又は複数は、例えば、コンピューター１２の動作を制御するためのＲＡＭのような、メモリー２２の一部に位置する場合もある。

【００１２】

[0018] サーバー１６は、ネットワーク１４に接続されているサーバー・コンピューター・システムを代表することができる。サーバー・コンピューター・システムは、要求に答えるためのソフトウェア・モジュールおよび１つ又は複数のプロセッサー、メモリー、ストレージ等を含むことができ、これらは、コンピューター１２に関して本明細書において説明したものと同様である。以下の節において更に詳しく説明するが、本明細書における技法は、コンピューター１２およびサーバー１６がネットワーク１４を通じて通信することができる一実施形態において用いることができる。コンピューター１２およびサーバー１６は、メッセージの送信に関連して動作を実行することができる。例えば、サーバー１６は、対称暗号技法を用い、メッセージを暗号化するために暗号化動作を実行し、次いで暗号化したメッセージをコンピューター１２に送ることができる。コンピューター１２は、暗号化されたメッセージをサーバー１６から受信し、このメッセージを解読するために、本明細書における技法にしたがって処理を実行することができる。対称暗号技法に関して、メッセージを暗号化および解読するために、同じ鍵（本明細書では、対称鍵とも呼ぶ）が用いられる。つまり、この例を参照すると、メッセージの送信側（この例では、サーバー１６）は鍵を用いてメッセージを暗号化し、このメッセージの受信側（この例では、コンピューター１２）は同じ鍵を用いて、このメッセージを解読する。対称鍵技法は、メッセージ受信側によって、メッセージの改竄を検出し、受信したメッセージが特定の送信元によって送られたことを検証するために用いることができる。

【００１３】

[0019] 本明細書に記載する技法を用いて、コンピューター１２は、解読動作の間鍵を

露出することなく解読を行う鍵特定解読エンジンを用いて、受信したメッセージを解読することができる。鍵特定解読エンジンは、解読動作を行うときに、入力としてこの鍵にアクセスする必要がない。代わりに、以下の節において説明するように、鍵特定解読エンジンは、特定の鍵にしたがって決定される１組のブール関数を用いて実現することができる。当技術分野では周知であるが、ブール関数（論理関数としても知られている）は、表現の評価のような評価を実行する関数として定めることができ、評価の結果を示す真または偽のブール結果即ち論理結果を戻す。鍵特定解読エンジンは、対称暗号化に用いられる特定の鍵にカスタム化された解読エンジンとして特徴付けることができる。鍵特定解読エンジンのインスタンスは、対称鍵毎に発生される。対称鍵は、鍵特定解読エンジンからは容易に決定することはできない。逆に、鍵特定解読エンジンの論理関数は、リバース・エンジニアリングを行い対称鍵を決定する試みを複雑にするように、対称鍵をわかりにくくする。

【００１４】

[0020] 同様に以下の節において説明する別の実施形態では、コンピューター１２上におけるソフトウェア・モジュールは、ソフトウェアの盗用を防止することに関連して、本明細書における技法を用いることができる。例えば、ユーザーはソフトウェア・アプリケーションをコンピューター１２上にインストールすることができる。インストールしたソフトウェア・アプリケーションを活性化するためには、ユーザーは識別子を入手すればよい。識別子は、別のコンピューター・システムから入手することができる。あるいは、コンピューター１２は、サーバー１６に接続せずに単体で動作することができ、識別子は、ユーザーが生の人間または自動化した音声認識システムと通信するときのように、ユーザーによって電話を通じて入手することができる。識別子、またはその一部は、暗号化されたデータ項目であってもよい。コンピューター１２上のソフトウェア・モジュールは、ソフトウェア・アプリケーションを実行しようとする度に、検証処理を実行することができる。この検証処理は、コンピューター１２上においてソフトウェア・モジュールが、本明細書における技法を用いて、識別子またはその一部の解読を実行することを含むことができる。したがって、本明細書における技法は、ソフトウェアの盗用と関連して発生される場合もある偽りの識別子の使用を防止することができる。偽りの識別子は、ソフトウェア・アプリケーションの不正コピーまたは使用許諾を受けていないコピーを活性化するために発生される場合もある。これについては、本明細書において更に詳しく説明する。

【００１５】

[0021] 以上は、本明細書における技法の使用を示す２つの例に過ぎない。尚、本明細書における技法は、同じまたは異なるコンピューター・システムにおいて、更に一般的に、あらゆる送信側からあらゆる受信側に送信される通信と関連付けて用いることができることは、当業者には認められよう。

【００１６】

[0022] 図２を参照すると、そこに示されているのは、本明細書において記載する技法の実行に関して、一実施形態において用いることができるコンポーネントを示す一例である。図２の例１００は、送信側１２２および受信側１２４を含む。送信側１２２は、対称鍵１０４を用いてメッセージ１０２を暗号化する対称暗号技法を用いて暗号化動作を実行するために、対称暗号化エンジン１０６を用いることができる。対称鍵１０４は、送信側１２２による暗号化メッセージ１１０を発生するための対称暗号化を行うことに関連して用いられる鍵である。対称鍵１０４が対称であるというのは、暗号化および解読の暗号動作を行うときのその使用に関して、同じ鍵１０４がメッセージの暗号化および解読の両方に関して用いられるからである。対称暗号化エンジン１０６は、高度暗号化標準（ＡＥＳ：Advanced Encryption Standard）の実施というような、いずれの対称暗号技法でも用いることができる。対称暗号化エンジン１０６は、ＡＥＳの既存の実装というような、選択された対称暗号技法の既存の実装において暗号化動作を実行することと関連付けて用いることができ、対称鍵１０４と暗号化するメッセージ１０２とを含む入力を処理する実行可能コンピューター命令を用いて実現される。つまり、対称暗号化エンジン１０６は、対称

鍵１０４を入力として用いる。

【００１７】

[0023] 本明細書における例に関して、例示の目的のために暗号化メッセージ１１０に言及することがある場合、送信するメッセージ全体を暗号化することもできる。尚、本明細書における技法は、送信するメッセージの一部を暗号化し、それに応じて解読することができる一実施形態と合わせても用いることができることは、当業者には認められよう。

【００１８】

[0024] 一旦送信側が暗号化メッセージ１１０を発生したなら、暗号化メッセージ１１０は受信側１２４に送信される。次いで、受信側１２４は、鍵特定解読エンジン１１２を用いて受信した暗号化メッセージ１１０を解読し、解読メッセージ１１４を発生する。鍵特定解読エンジン１１２を発生するために用いることができる技法については、本明細書において更に詳しく説明する。本明細書において記載する技法を利用する一実施形態にしたがって、選択された対称暗号技法の既存の実装において利用されていたかもしれない解読エンジンの代わりに、鍵特定解読エンジン１１２が用いられる。選択された対称暗号技法の既存の実装において用いられる解読エンジンは、対称鍵１０４と解読する暗号化メッセージ１１０とを含む入力を処理する実行可能コンピューター命令を用いて実現することができる。つまり、暗号化エンジン１０６と同様に、対称鍵が入力として用いられ、例えば、コンピューター上にある記憶位置から得ることができる。本明細書における技法を用いる一実施形態によれば、選択された対称暗号技法のための解読動作を行う鍵特定解読エンジン１１２は、入力を処理する既存のコンピューター命令を用いても実現することができる。しかしながら、鍵特定解読エンジン１１２への入力は、暗号化メッセージ１１０を含むが、対称鍵は含まない。

【００１９】

[0025] 解読メッセージ１１４は、検証処理１２０のための入力として用いることができる。例えば、検証処理１２０は、解読メッセージ１２０が、受信側によってローカルに格納することができる、またはそれ以外で入手することができる、元のメッセージ１０２のコピーと一致するか否か判定することができる。解読メッセージ１２０が元のメッセージ１０２のコピーと一致する場合、検証処理は成功であり、受信した暗号化メッセージが改竄されておらず、送信側１２２によって送られたものであることを示す。それ以外の場合、検証処理は失敗となる。尚、検証処理を実行するためには、種々の異なる技法の内いずれでも用いることができることは、注記してしかるべきである。別の一例として、送信側は、暗号化されたデータの第１部分と第２部分とを含むメッセージを送ることができる。第２部分は、対称鍵１０４と共に対称暗号化エンジン１０６を用いて第１部分を暗号化した結果とすることができる。受信側１２４は、第１および第２部分を含むメッセージを受信し、鍵特定解読エンジン１１２を用いて第２部分を解読して、結果を発生することができる。次いで、受信側は、この解読の結果が、受信したメッセージの第１部分と一致するか否か判定するために、検証処理を実行することができる。

【００２０】

[0026] 鍵特定解読エンジンについて、鍵特定解読エンジンを構成する方法例と共に、これより詳細に説明する。

[0027] 図３を参照すると、本明細書における技法に関連して、一実施形態において用いることができる鍵特定解読エンジンの表現例が示されている。先に説明したように、鍵特定解読エンジン３０２は、対称鍵を用いて解読を実現するために発生された１組のブールまたは論理関数３０４Ａ～３０４ｎとして実現することができる。入力３０６は、暗号化メッセージであり、出力３０８は解読メッセージである。図示のように、ブール関数３０４ａ～３０４ｎの各々は、入力３０８の多数のビットを用いて、出力３０８の１ビットを決定することができる。この例では、関数３０４ａ～３０４ｎの各々は、入力３０６の全てのビットに基づいて、出力の１つのビットを決定する。

【００２１】

[0028] 鍵特定解読エンジン３０２は、一実施形態において用いるために選択された特

定の対称暗号技法にしたがって、所与の入力３０６に対して所望の出力３０８を発生するブール関数を実現するソフトウェアを用いて実現することができる。

【００２２】

[0029] ブール関数３０４ａ～３０４ｎは、ＡＮＤ、ＯＲ、ＮＯＴ、ＸＯＲ、または以上の何らかの組み合わせというような、１つ又は複数の論理演算を実行することができる。

【００２３】

[0030] 図４を参照すると、図３のブール関数３０４ａ～３０４ｎは、選択された対称暗号技法に合わせた解読エンジンのリファレンス実装(reference implementation)４０２を用いて決定された値を有する出力ビット毎に、真理値表を評価することによって決定することができる。リファレンス実装４０２とは、対称鍵４０４と解読する暗号化メッセージ４０６とを含む入力を処理して解読メッセージ４０８を出力として発生する、解読エンジンの一例のことを指し、選択された対称暗号技法の既存の実装(implementation)において利用することができる。リファレンス実装４０２を用いて、出力４０８の期待値を、対称鍵４０４に対する入力値４０６の異なる組み合わせについて決定することができる。入力４０６に対する値の各可能な組み合わせを出力４０８の期待値にマッピングするブール関数を決定することもできる。

【００２４】

[0031] ブール関数をどのように決定することができるか例示するために、簡略化した例を選択し、入力４０６と、２ビット長である出力４０８とを用いて、以下の節において説明する。

【００２５】

[0032] 図５を参照すると、例５００が示されており、鍵特定解読エンジンのブール関数を決定することに関連する真理値表の使用を例示する。真理値表５０２は、異なる可能な入力値５０４ａおよび期待出力値５０４ｂを列挙する。５０４ｂの値は、特定の対称鍵に対する５０４ａにおける入力値の組み合わせ毎に、図４に示すようにリファレンス実装４０２を実行することによって発生することができる。リファレンス実装４０２によって発生された出力値は、５０４ｂにおいて記録することができる。真理値表５０２に基づいて、１組の１つ又は複数のブール関数を、標準形で、出力のビット毎に決定することができる。全ての論理関数は、積の和（ＳＯＰ）および和の積（ＰＯＳ）のような、標準形で表現可能である。また、ＳＯＰは、極小項の論理和（ＯＲ）の正規形としても知られている。関数のＰＯＳ、関数のＳＯＰの論理的同等物は、極大項の論理積（ＡＮＤ）としても知られている。ｎ個の変数の各々が１回現れる変数Ｉｎ０、．．．、Ｉｎｎのブール関数について、相補（例えば、否定）または非相補(uncomplemented)のいずれかは極小項と呼ばれる。つまり、極小項は、論理積（ＡＮＤ）演算子および相補（ＮＯＴ）演算子から成るｎ個の変数の論理表現である。極大項は、論理和（ＯＲ）演算子および相補（ＮＯＴ）演算子から成るｎ個の変数の論理表現である。

【００２６】

[0033] 図５および本明細書における他の図を参照すると、論理ＮＯＴ演算は「～」演算子で表記することができ、論理ＯＲ演算子は「＋」で表記することができ、論理ＡＮＤ演算子は「＊」で表記することができる。

【００２７】

[0034] 本明細書における技法によれば、真理値表５０２を調べて、ＳＯＰまたはＰＯＳを出力のビット毎に決定することができる。出力のビットに対して決定されるＳＯＰまたはＰＯＳは、出力のビットを決定する鍵特定解読エンジンに含まれるブール関数を表すことができる。例示の目的のために、この例ではＳＯＰを用いる。

【００２８】

[0035] エレメント５０６は、Ｏｕｔ０、即ち、出力のビット０を決定する第１ブール関数Ｆ０を表す。エレメント５０６は、真理値表５０５ａおよび５０５ｄを調べることによって決定することができ、出力Ｏｕｔ０は１即ち真となっている。真理値表におけるこ

のような行毎に、極小項を決定する。行５０５ａは、極小項（～Ｉｎ０＊～Ｉｎ１）として表すことができ、行５０５ｄは、極小項（Ｉｎ０＊Ｉｎ１）として表すことができる。ＳＯＰは、５０６において表されているように、以上の極小項の論理ＯＲである。エレメント５０８は、Ｏｕｔ１、即ち、出力のビット１を決定する第２ブール関数Ｆ１を表す。エレメント５０８は、真理値表の行５０５ｂを調べることによって決定することができ、出力Ｏｕｔ１は１即ち真となっている。行５０５ｂは、エレメント５０８に含まれる、極小項（～Ｉｎ０＊Ｉｎ１）として表すことができる。

【００２９】

[0036] ブール関数Ｆ０は、５０６において表されているように実装することができ、ブール関数Ｆ１は、本明細書に記載するように、メッセージの受信側によって用いることができる鍵特定解読エンジンにおいて、５０８において表されるように実装することができる。以上のプロセスを対称鍵毎に実行して、その特定の対称鍵に合わせてカスタム化した鍵特定解読エンジンを決定することができる。

【００３０】

[0037] 尚、説明したばかりのような初期ブール関数を一旦決定したならば、この初期ブール関数を更に変換して、論理的には同等であるが更に複雑であり、期待出力が得られるブーリアン関数を決定することもできることは注記してしかるべきである。例えば、Ａ　ＯＲ　Ｂのブール関数における論理演算は、DeMorgan（ドモルガン）の法則にしたがって、ＮＯＴ（ＮＯＴ（Ａ）　ＡＮＤ　ＮＯＴ（Ｂ））として実装することができる。

【００３１】

[0038] 例えば、関数Ｆ０を（～Ｉｎ０＊Ｉｎ１）＋（Ｉｎ０＊Ｉｎ１）として実装する代わりに、DeMorganの法則にしたがってＦ０を更に変換して、ＮＯＴ（Ａ　ＡＮＤ　Ｂ）＝（ＮＯＴ　Ａ）ＯＲ（ＮＯＴ　Ｂ）、およびＡ＝（～Ｉｎ０＊～Ｉｎ１）およびＢ＝（Ｉｎ０＊Ｉｎ１）とし、Ｆ０は（Ｉｎ０＋Ｉｎ１）＊（～Ｉｎ０＋～Ｉｎ１）として実装されるようにする。更なる変換および／または並び替えも、ブール代数の別の論理的等価にしたがって適用することができ、Ａ＝ＮＯＴ（ＮＯＴ　Ａ）であるので、入力Ｉｎ０およびＩｎ１の各々について、～（～Ｉｎ０）および～（～Ｉｎ１）をそれぞれ代用し、関数Ｆ０を実装する際に用いることができる。各関数に異なる変換を適用することもできる。例えば、一実施形態では、ブール関数の内一方をＳＯＰとして決定し、ブール関数の内他方を、ＰＯＳを用いて決定することができる。このような変形および変換は、鍵特定解読エンジンのリバース・エンジニアリングによって鍵を導き出す際に、困難さを増大するために用いることができる。

【００３２】

[0039] 図６を参照すると、図５の５０６および５０８を用いて、鍵特定解読エンジンにおいて実装することができる、ブール関数Ｆ０およびＦ１の別の表現が示されている。ブール関数Ｆ０およびＦ１は、ソフトウェアの符号化によって実装することができる。また、これらのブール関数は、全体的または部分的に、ディジタル回路、フィールド・プログラマブル・ゲート・アレイ等を用いて実装できることも注記してしかるべきである。

【００３３】

[0040] ブール関数を決定するために真理値表の使用を伴う、先に説明したばかりの処理は、ソフトウェアを用いて自動的に実行することもできる。前述の技法はブール関数を決定するために用いることができるが、その用法は実用上、利用されるコンピューター・システムの利用可能なリソースにしたがって限定される場合もある。前述のように全ての入力に基づいて各出力毎に真理値表を発生することに関して、各真理値表のサイズは、２＊＊ｎ個の可能な入力の組み合わせとなる。入力ビットの数が閾値数よりも少ない場合、例えば、３２以下である場合、前述の技法を用いると、出力ビット毎に１つのブール関数が得られ、このブール関数は入力の全てのビットを用いる。選択される閾値は、例えば、処理速度および／または格納限界というような、コンピューター・システムのリソースに基づいて決定するとよい。

【００３４】

[0041] 別の例として、一実施形態では、ＡＥＳの実装を、１２８ビットのブロック・サイズ単位で動作する対称暗号技法として用いることができる。つまり、ＡＥＳを実施する鍵特定解読エンジン３０２は、各々長さが１２８ビットの入力（暗号化メッセージ）および出力（解読メッセージ）を有する。この例に関して、各々１２８個の入力と１つの出力を有する１２８個のブール関数を用いて鍵特定解読エンジンを実現するのは、実現可能でない場合もある。何故なら、ブール関数を発生するために用いられる真理値表は、可能な入力の組み合わせに合わせて２＊＊１２８個のエントリーを含むからである。このため、一実施形態では、鍵特定解読エンジンの実装と関連付けて代わりの区分技法を用いることができる。前述したものと比較して、解読処理をブール関数のグループに区分して、１つの区画の出力を後続の区画の入力にすることによって、用いるブール関数の数は増えるが、各々のブール関数が用いる入力数を少なくして、鍵特定解読エンジンを実装することができる。次いで、これらのブール関数の各々は、例えば、図５に示したような真理値表を用いる、前述の技法を用いて実装することができる。尚、この区分をどのように行えばよいかは、選択された対称暗号技法の処理ステップ、および種々の処理時点における入力と出力との間の依存性に応じて決定されることを注記しておく。

【００３５】

[0042] 図７を参照すると、鍵特定解読エンジン７０２のブール関数を決定するための、前述した代わりの区分技法の図が示されている。例７００では、一実施形態において用いるために選択された対称暗号技法の解読動作の処理を、多数の区画７０２ａ～７０２ｎに分割することができる。各区画は、その区画の中間結果即ち出力を生成するブール関数のグループに対応することができる。例７００を参照すると、区画１の出力は、区画１出力７１０ａと表記されており、区画２の出力は、区画２出力７１０ｂと表記されている等となっている。最後の区画、即ち、区画ｎの出力は、解読されたメッセージ７０６となる。１つの区画の出力は、次の区画の入力として用いられる。選択された対称鍵技法のリファレンス実装は、中間結果即ち区画出力を決定するために用いることができる。

【００３６】

[0043] これより説明するのは、以上の区分技法を示す例である。第１の例として、暗号化メッセージである４ビット入力を用い、解読メッセージを４ビット出力として発生する対称暗号技法の仮説的解読動作に関して、例示を行う。各々、４入力ビット全てを用いて出力のビットの１つを決定する４つの論理関数を用いて鍵特定解読エンジンを実装する場合、１６個のエントリー（例えば、２＊＊４）を有する真理値表を用いればよい。あるいは、各区画が、図７に関して説明したような中間結果を発生するように、解読プロセスを区分するために、区分技法を用いることもできる。更に例示するために、図８を参照する。仮説的解読動作に関して、２つの区画８０２および８０４が決定されるように、解読処理を、処理ステップの２回の繰り返しで実行すると仮定する。各区画は、処理の繰り返しの１つに対応する。暗号化メッセージ８０２ａが区画１　８０２に入力され、中間結果８０２ｂを発生する。本明細書における他のところで説明したように、８０２ｂは、対称鍵を用いる仮説的解読動作に対するリファレンス実装を用いて決定することができる。中間結果８０２ｂは、区画１の出力として発生され、第２区画処理８０４への入力として用いられる。区画８０４は、解読メッセージに対応する出力８０６を発生する。

【００３７】

[0044] エレメント８０８は、４つのブール関数から成る第１の組を表し、区画８０２の入力と出力との間における依存性というような、仮説的解読のための処理の特殊性にしたがって、第１区画８０２について決定することができる。エレメント８０８は、中間結果８０２ｂの各ビットは、入力の２ビットの関数として決定できることを示す。区画８０２に対して各ブール関数を決定するために、２入力の異なる組み合わせ分の入力を含む真理値表を作成することができる。例えば、中間結果ビットＸ０は、入力Ｉｎ１およびＩｎ２に基づいて真理値表を作成することによって決定され、中間結果ビットＸ１は、入力Ｉｎ１およびＩｎ３に基づいて真理値表を作成することによって決定される等となる。エレメント８０８は、４つのブール関数から成る第１の組を表し、区画８０２の入力と出力と

の間における依存性というような、仮説的解読のための処理の特殊性にしたがって、第１区画８０２について決定することができる。同様に、エレメント８１０は、出力８０６の各ビットは、中間結果即ち区画１の出力８０２ｂの２ビットの関数として決定することができる。区画８０４の各ブール関数を決定するために、８１０において示す８０２ｂの２ビットについて異なる組み合わせ毎にエントリーを含む真理値表を作成することができる。例えば、Ｏｕｔ０は、中間結果のビットＸ２およびＸ３に基づいて真理値表を作成することによって決定され、Ｏｕｔ１は、中間結果のビットＸ１およびＸ３に基づいて真理値表を作成することによって決定される等となる。

【００３８】

[0045] 図示したこの特定の解読動作では、鍵特定解読エンジンを、８関数の全て、つまり、区画毎に４関数を用いて実装することができるように、前述した区分および区画入力の出力に対するマッピングが可能な場合もある。

【００３９】

[0046] これより更に例示するために、サイズが１２８ビットの対称鍵を用いるＡＥＳ対称鍵技法と関連付けて、別の例を説明する。ＡＥＳは、１２８ビットの暗号化メッセージの固定サイズ入力に対して動作する。ＡＥＳ解読プロセスは、暗号ラウンドまたは繰り返しを用いて実施される。繰り返しのことを、ここではラウンドと呼ぶ。各ラウンドは、８ビット・バイトの４×４アレイに対して動作する一連のステップまたは段階から成る。サイズが１２８ビットの対称鍵を用いると、ＡＥＳ解読処理は、１１回処理ラウンドを実行する。特定のＡＥＳの実装によって許容される２５６ビットというような、異なる鍵サイズが用いられる場合、追加の処理ラウンドを実行すればよい。

【００４０】

[0047] ＡＥＳについては、例えば、２００１年１１月２６日付けのFederal Information Processing Standards Publication 197 (FIPS-197：連邦情報処理標準公報１９７)に記載されている。本明細書では、周知のＡＥＳ暗号技法の特定の態様を、本明細書における区分技法を詳しく例示する目的のために引用したが、区分技法は、このようなその他のブロック暗号技法のラウンドを同様に含む他の暗号技法と関連付けた使用にも適用できることは、当業者には認められよう。

【００４１】

[0048] ＡＥＳは、固定長ビットのグループに対して動作する対称鍵を用いるブロック暗号として特徴付けることができる。固定長ビットのグループのことを、ブロックとも呼ぶ。前述の区分技法によれば、解読プロセスの各ラウンドは、図９に示すような区画に対応することができる。例９００では、サイズが１２８ビットの対称鍵を用いる、１１ラウンドのＡＥＳの解読処理を示し、ラウンド１の出力がラウンド２の入力となり、ラウンド２の出力がラウンド３の入力となり、ラウンド１１の出力が解読出力となるまで、同様に続く。ブール関数のグループは、ラウンド毎に、特定の１２８ビット対称鍵について決定することができる。各ラウンドでは、同じ解読処理を実行することができる。ＡＥＳに関しては、各ラウンドは同じ４段階またはステップから成ることができる。ＡＥＳの各ラウンド内における解読処理の各段階は、一度に４×４アレイに含まれるデータの１バイト即ち８ビットに対して動作を実行することを伴う。つまり、鍵特定解読エンジンは、ブール関数を用いて実装することができ、各ブール関数は、１出力ビットを決定するために８ビットの入力を有する。

【００４２】

[0049] 図１０を参照すると、一実施形態の１回のラウンドにおいて実行することができる処理を図示する一例が示されている。例１００は、ＡＥＳ暗号技法に対して解読動作を実行するときのように、４段階またはステップから成るラウンドを示す。更に一般的には、暗号技法に応じて、解読動作のラウンドは、図示とは異なる数の段階またはステップを含むこともでき、段階当たり１２８ビット以外、および図示のようなブール関数毎に８ビット以外の異なる数の入力ビットに対して動作することができる。鍵特定解読エンジンは、段階数、各段階の入力および出力、ラウンド等にしたがって決定されたブール関数を

用いて実装することができる。

【００４３】

[0050] 各段階において実行される処理、ならびに段階数および関連する入力や出力の数は、対称暗号技法毎に様々である。例えば、ＡＥＳに関して言えば、段階１は、代入ステップを実行することを含み、段階１への入力の各８ビットが、参照表にしたがって、別の８ビットと交換され、段階２は転置ステップを含むことができ、４×４アレイの各行のバイトを、ＡＥＳ解読処理の特殊性に応じて等で、循環的に指定された回数だけシフトする。

【００４４】

[0051] 丁度説明したように、区画は、暗号ラウンドに基づいて決定することができる。ラウンド毎のブール関数は、各ラウンドにおける段階数またはステップ数、および段階毎の処理を実行する際に用いられる入力および出力に関する特殊性に応じて決定することができる。次いで、真理値表を用いて、先に説明したような特定の暗号技法に対する解読エンジンのリファレンス実装を用いて、ブール関数を決定することができる。１２８ビットの対称鍵を有するＡＥＳを用いる一例の図示として、１１ラウンドにしたがって解読処理を区分することができ、各ラウンドは４段階を含む。ＡＥＳの解読処理は、前述のように８ビット部分に対して動作を実行するので、ブール関数を段階毎に実装することができ、各関数が、その段階への入力の８ビットに基づいて、当該段階に対する出力の１ビットを決定する。尚、別の暗号技法のための解読処理では、異なるビット数に対して動作することもでき、ブール関数はそれに応じて決定すればよいことは、注記してしかるべきである。図１０に示すように、１２８ビットの対称鍵に基づくＡＥＳに対して解読を行うことに関して、１２８個のブール関数を用いて、１ラウンドの各段階を実施することができ、各関数は、８入力ビットおよび１出力ビットを有し、その結果、ラウンド毎に５１２個のブール関数を実装することになる。つまり、１２８ビット対称鍵に基づくＡＥＳを用いるときには、各鍵特定解読エンジンを実装する際に、合計で５６３２個のブール関数を用いることができる。尚、この例では、各ブール関数を決定する際に用いられる真理値表は、２＊＊８個のエントリーを含み、これらのエントリーは、本明細書における技法と共に用いるために、コンピューター・メモリーに容易に格納できることは注記してしかるべきである。

【００４５】

[0052] 図１１を参照すると、本明細書における技法にしたがって鍵特定解読エンジンを発生することに関連して一実施形態において実行することができる処理ステップのフローチャートが示されている。鍵特定解読エンジンは、ブール関数を用いて実装することができ、対称鍵の入力を必要としない。フローチャート１１００のステップは、対称暗号技法のための鍵特定解読エンジンを発生するために実行することができる。次いで、鍵特定解読エンジンを受信側に供給して、例えば、図２と関連して示したように用いることができる。フローチャート１１００は、前述のような処理を要約する。ステップ１１０２において、入力におけるビット数を得る。ステップ１１０４において、入力のビット数が閾値の値を超過するか否かについて判断する。前述のように、この閾値の値は、処理が行われている特定のコンピューター・システムのリソースに応じて選択することができる。何故なら、入力のビット数は、ブール関数を決定するために用いられる真理値表に影響を及ぼすからである。ステップ１１０４において評価が否定になった場合、制御はステップ１１１０に進み、解読エンジンおよび対称鍵のリファレンス実装を用いて、真理値表および対応するブール関数を決定する。制御はステップ１１１２に進み、ステップ１１１０において決定したブール関数の追加の変形を任意で実行する。前述のように、追加の変形は、鍵特定解読エンジンのリバース・エンジニアリングによって対称鍵を判断するのを一層難しくするために行うことができる。ステップ１１１４において、鍵特定解読エンジンを発生することができる。鍵特定解読エンジンは、ステップ１１１０および１１１２の結果として決定されたブール関数を符号化することによって実装することができる。ステップ１１０４における評価が肯定となった場合、制御はステップ１１０６に進み、一実施形態にお

いて可能な区分（１つまたは複数）を決定するために用いられる、選択された対称暗号技法のための解読プロセスに関して、分析を行う。前述のように、決定される区画は、選択された対称暗号技法に応じて様々に変化する。選択された対称暗号技法に対する解読動作の異なる特性を、区画をどのように形成するか決定する際に用いることができる。例えば、各ラウンドが区画に対応するように、暗号ラウンドを用いることができる。ステップ１１０８において、真理値表および各区画に対するブール関数を決定する。本明細書において記載したように、区画は、前述のようにラウンドの段階またはステップに対応するというように、更にサブ区画に分割することもできる。ステップ１１０８から、制御は前述したステップ１１１２および１１１４に進む。

【００４６】

[0053] 尚、本明細書では、真理値表について説明し、ブール関数を決定する際にこれを用いるが、一実施形態では、ブール関数を決定するために他の表現および技法を用いてもよいことは注記してしかるべきである。

【００４７】

[0054] これより説明するのは、本明細書における技法の別の使用例である。前述のように、コンピューター・システム上にインストールしたソフトウェアを活性化することに関して、識別子を入手する場合がある。以下の節に関して、ソフトウェア活性化の一部として入手する識別子を、確認識別子と呼ぶ。ソフトウェア活性化プロセスの一部を、図１２Ａのフローチャート１２００に要約する。図１２Ａでは、ステップ１２０２において、活性化サーバーに連絡することができ、ステップ１２０４において、確認識別子を入手する。確認識別子は、対称鍵によって暗号化したメッセージとすることができ、またはそれ以外では、暗号化データ部分を含むこともできる。確認識別子は、例えば、顧客代表部と連絡を取ったユーザーによって電話で、または自動音声認識システムによって入手することができる。自動音声認識システムは、活性化サーバーと呼ばれるコンピューター・システムから確認識別子を入手する。確認識別子は、種々の異なる技法の内いずれか１つを用いて得られた元のデータを用いて形成された暗号化部分を含むことができる。例示の目的のために、確認識別子は、活性化が実行され、インストールされたソフトウェアを含むコンピューターがネットワーク接続機能(connectivity)を有してないときに、電話の活性化を用いて入手することができる。あるいは、インストールされたソフトウェアの活性化は、オンライン活性化方法を用いても実行することができ、この場合、ユーザー・コンピューターが別のコンピューター・システムとネットワークを通じて通信して、ソフトウェアの活性化に関して用いられる他の情報を入手する。この例に関して、確認識別子は、電話の活性化が行われるときに入手できるのであって、オンラインの活性化が行われるときではない。ステップ１２０６において、インストールされたソフトウェアを含むコンピューター上に確認識別子が格納されるように、ユーザーは、インターフェースを用いて確認識別子を入力することができる。

【００４８】

[0055] コンピューター・システム上にあるソフトウェア・モジュールは、インストールされたソフトウェアを起動する試みがなされる度に検証処理を実行するために、確認識別子を用いることができる。図１２Ｂのフローチャート１２５０は、インストールされたソフトウェアを起動する試みに応答してソフトウェア・プログラムの実行を開始する一部として実行することができる処理を要約したものである。ステップ１２５２において、ユーザーは、マウスまたはその他の入力デバイスを用いて実行するアプリケーションを選択し、ソフトウェア・プログラムを起動することを要求することによる等によって、行為(action)を実行することができる。ステップ１２５４において、インストールされたソフトウェアが活性化されたかについて、電話の活性化またはオンライン活性化方法のようなその他の何らかの活性化方法を用いて判定を行う。電話活性化方法が以前に実行された場合、ステップ１２５４における評価は肯定となり、制御はステップ１２５８に進んで、コンピューター上に既に格納されている確認識別子を引き出す。また、ステップ１２５８は、検証処理を実行することに関連して一実施形態において用いることができる他の情報を引

き出すことも含むことができる。ステップ１２５０において、確認識別子およびその他の情報を引き出すことに成功したか否かについて判定を行う。成功しなかった場合、制御はステップ１２６４に進み、処理を終了する。ステップ１２６０の評価が肯定である場合、制御はステップ１２６２に進み、電話活性化方法にしたがって検証処理を実行する。ステップ１２６２の処理については、以下で詳細に説明するが、確認識別子を用いる。電話活性化方法が以前に実行されなかった場合、ステップ１２５４の評価は否定となり、用いられる特定の活性化方法および既に得られている情報に合わせて、他の検証処理を実行する。他の活性化方法を用いると、得られている確認識別子の他にも、異なる情報を得ることができ、ソフトウェア・プログラムを起動する試みに関して以後用いることができる。

【００４９】

[0056] 図１３を参照すると、電話活性化が以前に行われており、確認識別子が得られたときの検証処理と関連して一実施形態において実行することができる処理ステップのフローチャートが示されている。フローチャート１３００の処理ステップは、単独で、または他の処理と合わせて、インストールされたソフトウェアの実行を進めることを許可するか否か判断する検証処理の一部として実行することができる。フローチャート１３００は、図１２Ｂのステップ１２６２と関連して実行することができるステップを記述する。ステップ１３０２において、確認識別子を引き出す。ステップ１３０２は、電話活性化処理の結果として得られ既に格納されている確認識別子を読み出すことを含むことができる。ステップ１３０４において、確認識別子、またはその暗号化部分を解読する。ステップ１３０６において、ステップ１３０４を実行した結果として得られた解読データが、元のデータのコピーと同じであるか否か判断する処理を実行する。本明細書において記載するように、元のデータのコピーは、例えば、ローカルにユーザー・コンピューター上に格納され、フローチャート１３００の処理を実行する目的で引き出すことができる。ステップ１３０６の評価が否定である場合、制御はステップ１３１０に進む。この場合、検証処理は失敗したことになる。ステップ１３０６の評価が肯定である場合、制御はステップ１３０８に進む。この場合、検証プロセスは成功したことになる。尚、ステップ１３０８から、検証処理が継続してもよく、検証処理の結果が成功するか否かは、ステップ１３０８に続いて実行するステップの結果に依存すればよいことは注記してしかるべきである。ステップ１３０４は、本明細書において説明した鍵特定解読エンジンを用いて実行することができる。

【００５０】

[0057] ソフトウェアの盗用は、例えば、ソフトウェアの非合法コピーを入手すること、および偽の確認識別子を発生することを含むことができる。例えば、図１２Ａ、図１２ｂ、および図１３と関連付けて説明したような実施形態は、本明細書における技法を用いて、ソフトウェアの非合法コピーを活性化することができるように偽の確認識別子を発生するために用いられる対称鍵を入手するのを困難にすることができる。

【００５１】

[0058] オンライン活性化のような他のソフトウェア活性化方法ではなく、電話活性化を実行する場合、ユーザーは電話を通じて確認識別子を得て、次いで手作業で確認識別子を入力することがあるという事実のために、用いられる確認識別子のサイズには、実用上の限度がある場合もある。鍵特定解読エンジンを用いると、ディジタル署名のような他の技法を用いたときに起こり得るメッセージサイズの増大を招くことなく、メッセージの改竄や、有効な確認識別子の認証の検証に備えることができる。更に、ディジタル署名のような他の技法を用いる場合にあり得るような、暗号化メッセージのサイズ増大に関して、余分なオーバーヘッドが生じない。

【００５２】

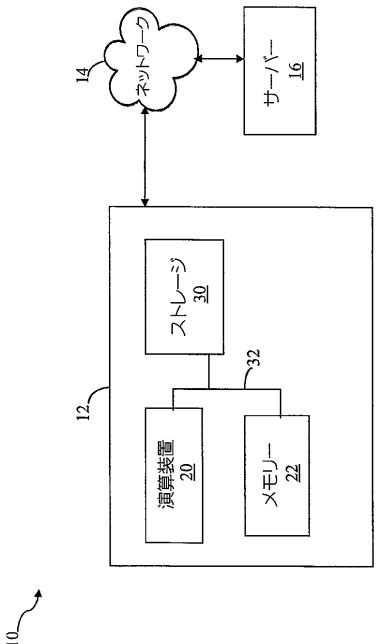
[0059] 尚、本明細書では電話活性化に言及したが、電話活性化は、もっと一般的に、確認識別子を入手することができる一実施形態において実行することができるオフライン活性化処理の一種と呼ぶこともできる。オフライン活性化は、一般に、インストールされたソフトウェアを含むコンピューター・システムが、インストールされたソフトウェアを

活性化するために、他のコンピューター・システムと通信しないときに用いられる活性化方法を指すことができる。むしろ、活性化処理は、インストールされたソフトウェアを含むコンピューター・システムが、ネットワーク、他のコンピューター・システム等との接続機能がなくオフラインである間に実行するとよい。

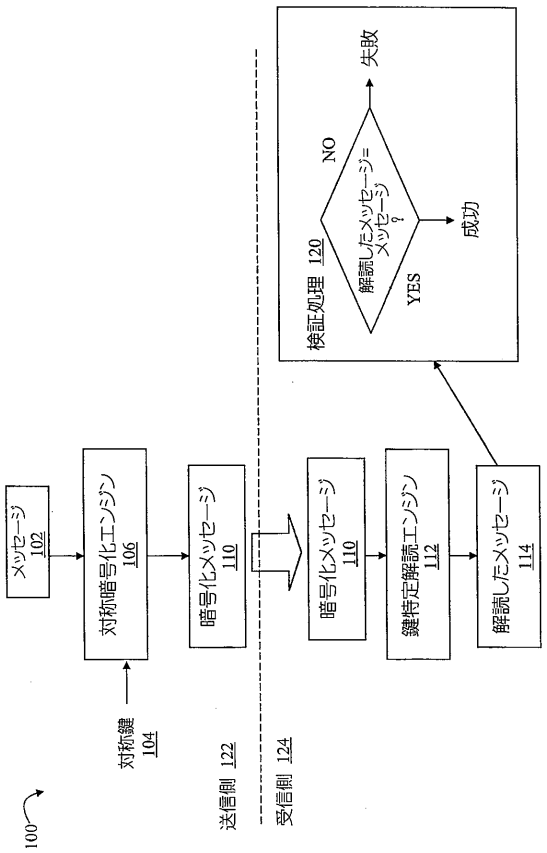
【００５３】

[0060] 以上、構造的特徴および／または方法論的行為に特定的な文言で主題について説明したが、特許請求の範囲に定められる主題は、必ずしも前述した具体的な特徴や行為に限定されるのではないことは言うまでもない。逆に、前述の具体的な特徴および行為は、特許請求の範囲を実現する形態例として開示したのである。

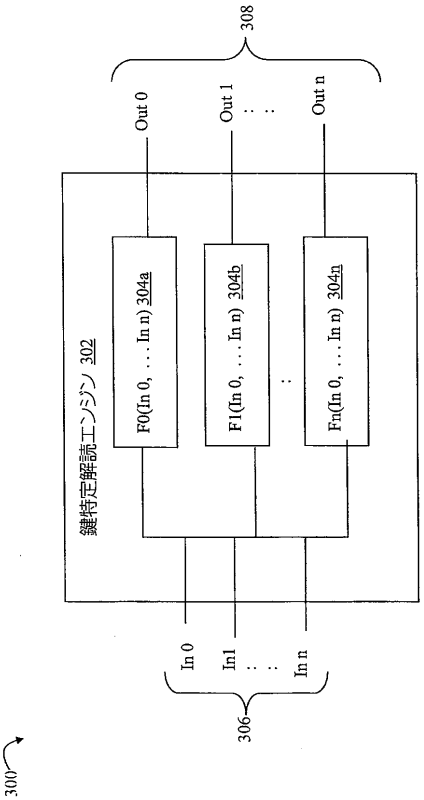
【図１】



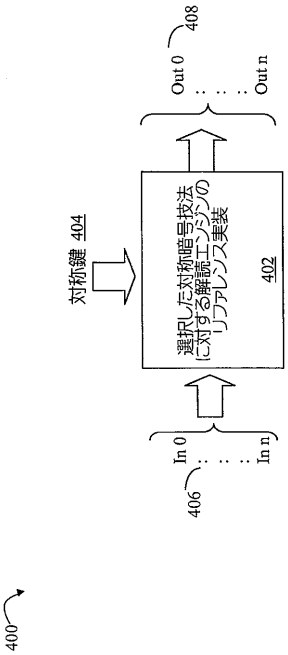
【図２】



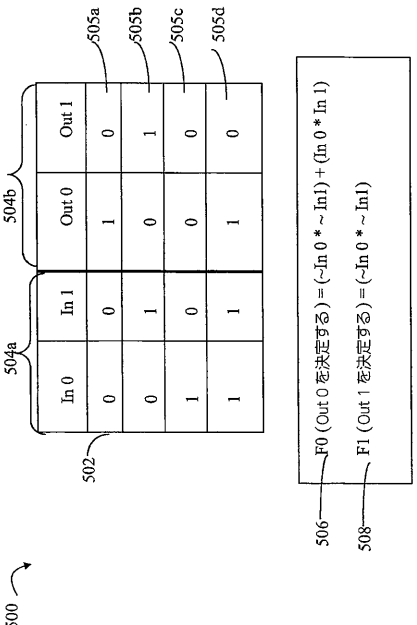
【図３】


308
Out 0
Out 1
Out n
鍵特定解読エンジン 302
F0(In 0, ... In n) 304a
F1(In 0, ... In n) 304b
Fn(In 0, ... In n) 304n
In 0
In 1
In n
306
300

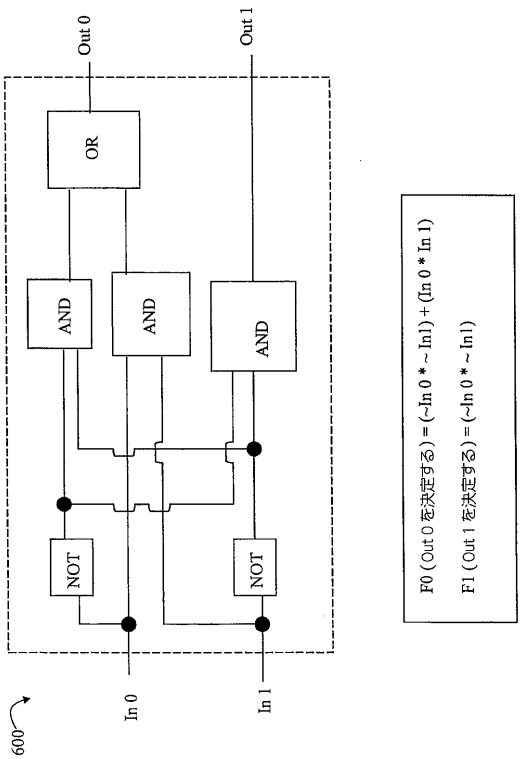
【図４】


408
Out 0
Out n
対称鍵 404
選択した対称暗号技法に対する解読エンジンのリファレンス実装
402
In 0
In n
406
400

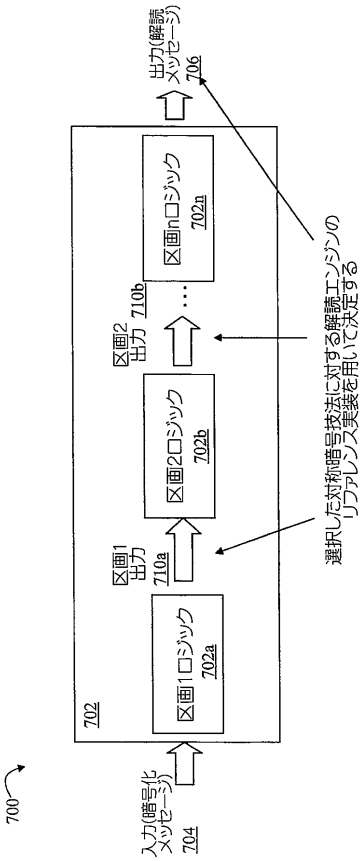
【図５】


505a
505b
505c
505d
504b
504a
In 0
In 1
Out 0
Out 1
0 0 1 0
0 1 0 1
1 0 0 0
1 1 1 0
502
F0 (Out 0 を決定する) = (~In 0 * ~ In1) + (In 0 * In 1)
F1 (Out 1 を決定する) = (~In 0 * ~ In1)
506
508
500

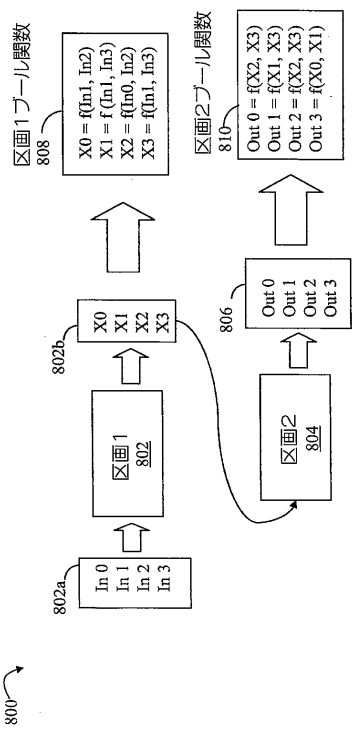
【図６】


Out 0
Out 1
OR
AND
AND
AND
NOT
NOT
In 0
In 1
600
F0 (Out 0 を決定する) = (~In 0 * ~ In1) + (In 0 * In 1)
F1 (Out 1 を決定する) = (~In 0 * ~ In1)

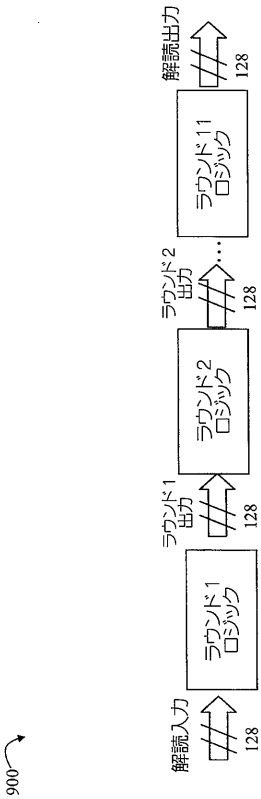
【図７】


700
702
区画1ロジック
702a
区画2ロジック
702b
区画nロジック
702n
区画1
出力
710a
区画2
出力
710b
入力(暗号化
メッセージ)
704
出力(解読
メッセージ)
706
選択した対称暗号技法に対する解読エンジンの
リファレンス実装を用いて決定する

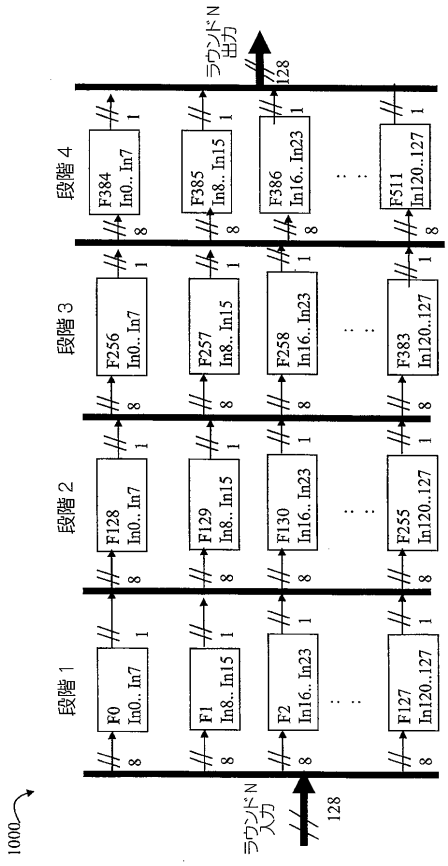
【図８】


800
802a
In 0
In 1
In 2
In 3
区画1
802
802b
X0
X1
X2
X3
区画2
804
806
Out 0
Out 1
Out 2
Out 3
区画1ブール関数
808
X0 = f(In1, In2)
X1 = f (In1, In3)
X2 = f(In0, In2)
X3 = f(In1, In3)
区画2ブール関数
810
Out 0 = f(X2, X3)
Out 1 = f(X1, X3)
Out 2 = f(X2, X3)
Out 3 = f(X0, X1)

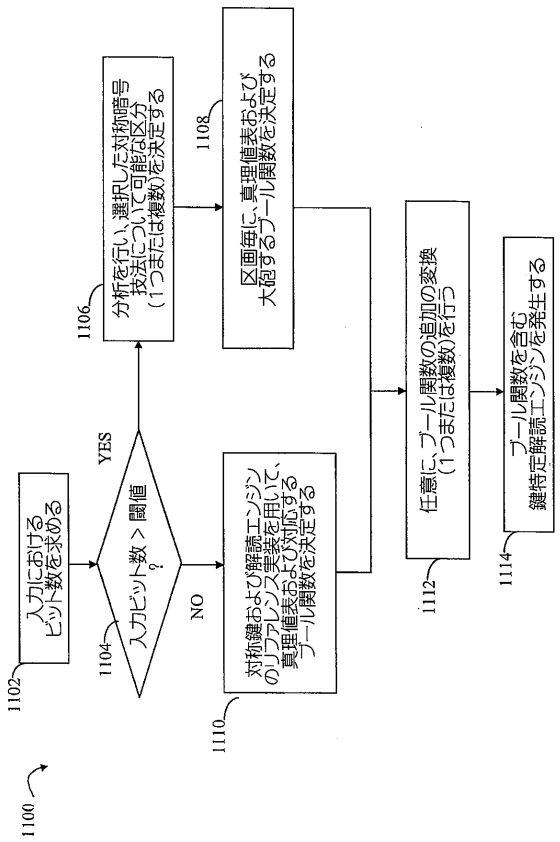
【図９】


900
解読入力
128
ラウンド1
ロジック
ラウンド1
出力
128
ラウンド2
ロジック
ラウンド2
出力
128
ラウンド11
ロジック
解読出力
128

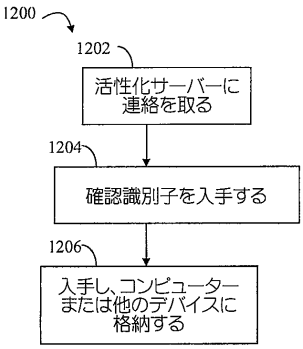
【図１０】


1000
ラウンドN
入力
128
段階 1
段階 2
段階 3
段階 4
F0
In0..In7
F1
In8..In15
F2
In16..In23
F127
In120..127
F128
In0..In7
F129
In8..In15
F130
In16..In23
F255
In120..127
F256
In0..In7
F257
In8..In15
F258
In16..In23
F383
In120..127
F384
In0..In7
F385
In8..In15
F386
In16..In23
F511
In120..127
ラウンドN
出力
128

【図１１】



【図１２Ａ】



【図１２Ｂ】

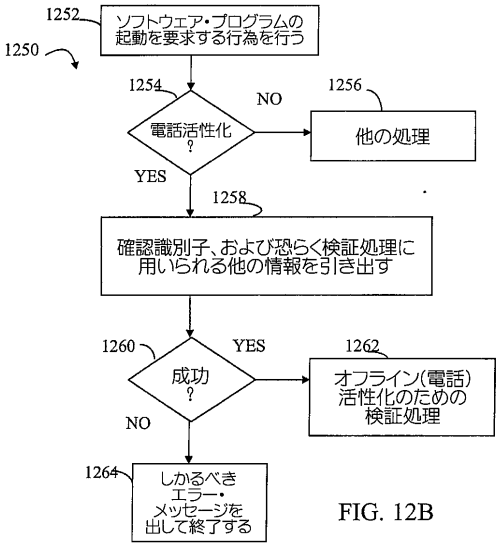
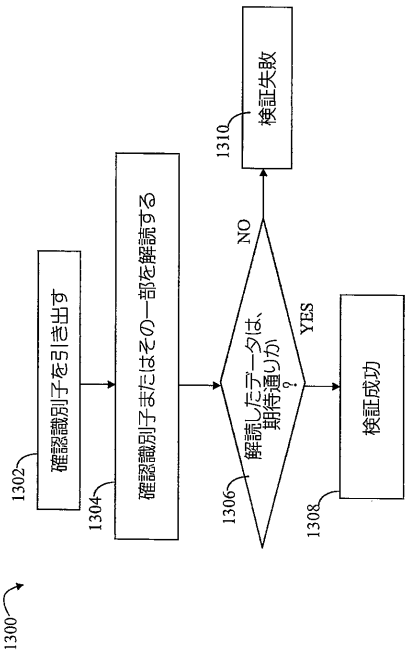


FIG. 12B

【図１３】



フロントページの続き

(72)発明者　アシポヴ，ボリス
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，マイクロソフト　コーポレーション，エルシーエイ－インターナショナル・パテンツ

審査官　松平　英

(56)参考文献　特開２００５－２８６６６１（ＪＰ，Ａ）
特表２００８－５１４９７５（ＪＰ，Ａ）
特開平１０－１０５６２０（ＪＰ，Ａ）
特開２００１－３１８７８６（ＪＰ，Ａ）
特開２００３－２２３０９８（ＪＰ，Ａ）
特開２００３－３０２８９９（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０９Ｃ　　１／００
Ｈ０４Ｌ　　９／００
Ｇ０６Ｆ　２１／２２