(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2013-84294
(P2013-84294A)
(43)公開日　平成25年5月9日(2013.5.9)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/62	(2013.01)	G06F	21/24	166A	5J104
H04L 9/08	(2006.01)	H04L	9/00	601B	
G06F 21/64	(2013.01)	G06F	21/24	167A	

審査請求 有　請求項の数 1　ＯＬ　外国語出願　（全 36 頁）

(21)出願番号　特願2012-276507 (P2012-276507)
(22)出願日　平成24年12月19日 (2012.12.19)
(62)分割の表示　特願2009-552649 (P2009-552649)の分割
原出願日　平成19年3月6日 (2007.3.6)

(71)出願人　509249690
オックスフォード，　ウィリアム　ブイ．
アメリカ合衆国　テキサス　７８７５５，
オースティン，　ピー．オー．　ボックス　２８１６１
(74)代理人　100078282
弁理士　山本　秀策
(74)代理人　100113413
弁理士　森下　夏樹
(72)発明者　オックスフォード，　ウィリアム　ブイ．
アメリカ合衆国　テキサス　７８７５５，
オースティン，　ピー．オー．　ボックス　２８１６１

Ｆターム(参考)　5J104 AA08　AA12　AA13　LA03　NA02　NA37　NA38

(54)【発明の名称】デジタル著作権制御用再帰的セキュリティプロトコルのための方法およびシステム

(57)【要約】
【課題】デジタルデータの保護のために再帰的セキュリティプロトコルを利用するシステムおよび方法を提供すること。
【解決手段】これらは、第１の暗号化アルゴリズムによってビットストリームを暗号化するステップと、第１の暗号解読アルゴリズムを暗号化されたビットストリームに関連付けるステップとを含んでもよい。次いで、得られるビットストリームは、第２のビットストリームを生じるように、第２の暗号化アルゴリズムによって暗号化されてもよい。次いで、この第２のビットストリームは、第２の暗号解読アルゴリズムに関連付けられる。次いで、この第２のビットストリームは、関連キーを使用して意図された受信者によって解読されてもよい。
【選択図】なし

【特許請求の範囲】
【請求項１】
明細書に記載の発明。
【発明の詳細な説明】
【技術分野】
【０００１】
（関連出願の相互参照）
本願は、米国特許出願第６０／３９０，１８０号（名称「Ｒｅｃｕｒｓｉｖｅ Ｓｅｃｕｒｉｔｙ Ｐｒｏｔｏｃｏｌ Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｆｏｒ Ｄｉｇｉｔａｌ Ｃｏｐｙｒｉｇｈｔ Ｃｏｎｔｒｏｌ」、２００２年6月２０日出願、出願人「Ｗｉｌｌｉａｍ Ｖ．Ｏｘｆｏｒｄ」）、米国特許出願第１０／４６５，２７４号（名称「Ｍｅｔｈｏｄ ａｎｄ Ｓｙｓｔｅｍ ｆｏｒ ａ Ｒｅｃｕｒｓｉｖｅ Ｓｅｃｕｒｉｔｙ Ｐｒｏｔｏｃｏｌ ｆｏｒ Ｄｉｇｉｔａｌ Ｃｏｐｙｒｉｇｈｔ Ｃｏｎｔｒｏｌ」、２００３年6月１９日出願、出願人「Ｗｉｌｌｉａｍ Ｖ．Ｏｘｆｏｒｄ」）、および米国特許出願第　不明　号（名称「Ｍｅｔｈｏｄ ａｎｄ Ｓｙｓｔｅｍ ｆｏｒ ａ Ｒｅｃｕｒｓｉｖｅ Ｓｅｃｕｒｉｔｙ Ｐｒｏｔｏｃｏｌ ｆｏｒ Ｄｉｇｉｔａｌ Ｃｏｐｙｒｉｇｈｔ Ｃｏｎｔｒｏｌ」、２００７年2月２３日出願、出願人「Ｗｉｌｌｉａｍ Ｖ．Ｏｘｆｏｒｄ」）に関連する。本パラグラフにおいて引用された全ての出願は、その全体が本明細書に参考として援用される。
【０００２】
（発明の技術分野）
本発明は、一般に、デジタルコンテンツの保護に関し、より具体的には、暗号化の使用を通したデジタルデータの保護に関する。さらに具体的には、本発明は、現在利用されている方法よりも優れたセキュリティおよび融通性の両方を提供する、再帰的セキュリティプロトコルで、デジタルコンテンツを保護することに関する。
【背景技術】
【０００３】
（発明の背景）
過去の著作権法の履行は、印刷された本であろうが、記録されたディスクまたはテープであろうが、いくつかの物理的対象を複製することの困難に依存していた。正確には、個人がそのような対象を複製することは常に可能であったが、物理的対象に含有される情報を別のものに転写する時間費用または機会費用のいずれかを考慮すると、一般的には経済的に実行可能ではなかった。また、この工程に基づくコピーは、現在まで、原本よりも質が低くなる傾向があった。しかしながら、情報のデジタル保存の出現とともに、この費用バランスが混乱している。
【０００４】
著作権のある作品の価値は、創作を含有する物理的対象に必ずしも吹き込まれているとは限らないが、むしろ作品自体を構成する情報に吹き込まれている。したがって、作品を複写する機械費用が無視できるほど小さくなると（現在の多くのデジタルメディアストリームの場合のように）、著作権保護工程は、新しい方式でその問題点に対処しなければならない。本質的には、デジタル形式にカプセル化することができる全ての創作物は、何らかの時点でこの懸念の対象となる。
【０００５】
この時点で所与の多大なデジタルデータストリームを複写することが非現実的となる場合があっても、そのようなデータストリームの複写および保存の費用は、長距離にわたってそれらのデータを伝送する費用と同様に、常に低下している。また、デジタル保存は、経時的に、または反復使用を介して劣化しない完璧なコピーを作製することを可能にする。そのようなものとして、これらの多大なデータセットの寿命は、それらの経済的な実行可能性を潜在的に克服することができ、その時点で、ストリームが無償配給に対応しているか否かにかかわらず、はるかに重要性が低い。この実行可能性の長さは、データへのア

クセスを制御する際に使用するために適切であるセキュリティの量に上限値を設置する。
【０００６】
デジタルセキュリティアルゴリズムにおける公知の技術の現状は、オンライン情報の熟読を介して、または、参照することに完全に本明細書に含まれる、特許文献１、特許文献２、特許文献３、特許文献４、特許文献５、および非特許文献１を含む、本主題を検討する種々の出版物および特許を介して容易に収集することができる。
【０００７】
従来技術のシステムは、デジタルデータ暗号化および暗号解読技術のいくつかの基本的動作カテゴリを利用する。これらのカテゴリは、セキュリティアルゴリズム自体の使用に基づき、実際のデータを暗号化または解読するための実際の機構と無関係である。これらの周知の技術、ならびに幅広く説明されている分類および技術は、以下のとおりである。
一方向ハッシング機構および／またはメッセージダイジェスト
メッセージ認証システム
デジタル署名
秘密キー暗号化システム
公開キー暗号化システム
所与のセキュリティシステムでこれらの技術が使用される手段は、セキュリティプロトコルとして知られている。セキュリティプロトコルは、種々の機能がどのように実施されるかという実際の基礎機構と無関係であることに留意されたい。そのようなものとして、完璧に安全な暗号化アルゴリズムさえ、暗号化技術の安全な側面を破るような方法で全体的なセキュリティを損なうセキュリティプロトコルの内側で、潜在的に使用される場合がある。その結果として、任意の所与のセキュリティシステムの全体的なセキュリティは、基礎セキュリティ技術の相対的強度だけでなく、これらのセキュリティ技術が使用される方法にも依存している。セキュリティシステムを実装することへの過去の試行は、保護される種々の種類のビットストリームの（人為的な）区別を行ってきた。
【先行技術文献】
【特許文献】
【０００８】
【特許文献１】米国特許第６，３２７，６５２号明細書
【特許文献２】米国特許第６，３３０，６７０号明細書
【特許文献３】米国特許出願公開第２００２００１３７７２号明細書
【特許文献４】米国特許第６，２２６，７４２号明細書
【特許文献５】米国特許第６，１０１，６０５号明細書
【非特許文献】
【０００９】
【非特許文献１】Ｄａｖｉｄ Ｌｉｅ，ｅｔ ａｌ．、「Ａｒｃｈｉｔｅｃｔｕｒａｌ Ｓｕｐｐｏｒｔ ｆｏｒ Ｃｏｐｙ ａｎｄ Ｔａｍｐｅｒ－Ｒｅｓｉｓｔａｎｔ Ｓｏｆｔｗａｒｅ」、Ｐｒｏｃｅｅｄｉｎｇｓ ｏｆ ｔｈｅ ９ｔｈ Ａｎｎｕａｌ Ｃｏｎｆｅｒｅｎｃｅ ｏｎ Ａｒｃｈｉｔｅｃｔｕｒａｌ Ｓｕｐｐｏｒｔ ｆｏｒ Ｐｒｏｇｒａｍｍｉｎｇ Ｌａｎｇｕａｇｅｓ ａｎｄ Ｏｐｅｒａｔｉｎｇ Ｓｙｓｔｅｍｓ ａｋａ ＡＳＰＬＯＳ－ＩＸ、Ｃａｍｂｒｉｄｇｅ、ＭＡ、２０００
【発明の概要】
【発明が解決しようとする課題】
【００１０】
したがって、デジタルコンテンツをより良好かつ効率的に保護するために業界標準セキュリティ技術および他の種類のセキュリティ標準を利用してもよい、再帰的セキュリティプロトコルに対する必要性がある。
【課題を解決するための手段】
【００１１】
デジタルコンテンツを良好に保護するために種々の暗号化技法を利用してもよい、セキ

ュリティプロトコル用のシステムおよび方法を開示する。これらのシステムおよび方法は、ユーザが希望どおりに元のデータのバックアップコピーを作製できるようにするが、そのようなコピーを使用するために著作権所有者の許可を依然として必要としてもよい方式で、任意のビットストリーム（オーディオ／ビデオストリームまたはソフトウェアプリケーション等の他のデジタルデータ等）を符号化できるようにする。多くの実施形態では、ビットストリームは、暗号化され、この結果は、暗号解読アルゴリズムに関連付けられる。順に、この組み合わせが暗号化され、この第２の暗号化の結果が第２のビットストリームを生じ、順に、第２の暗号解読アルゴリズムに関連付けられる。

【００１２】

加えて、これらのセキュリティプロトコルを実装するのに好適なコンピュータシステム、ハードウェア、およびソフトウェアでこれらの種類の方法論を具体化する、システムを提示する。

【００１３】

いくつかの実施形態では、各ビットストリームは、関連暗号解読アルゴリズムおよび１つ以上のキーを使用して解読される。

【００１４】

別の実施形態では、これらのキーは、サーバ上に存在してもよく、またはキーは、標的機械上のハードウェア中に存在してもよい。

【００１５】

さらに他の実施形態では、これらのキーは、キーデータ構造に含有される。

【００１６】

依然として他の実施形態は、１つ以上のキーデータ構造を含有するキーリストデータ構造を含む。

【００１７】

さらに具体的な実施形態は、中央サーバ上に存在する、このキーリストデータ構造を含む。

【００１８】

別の１組の実施形態では、暗号化されたビットストリームが真性であるかどうかを判定するために、メッセージダイジェストが使用される。

【００１９】

他の同様の実施形態では、解読されたビットストリームが真性であるかどうかを判定するために、メッセージダイジェストが使用される。

例えば、本発明は以下の項目を提供する。

（項目１）

デジタルコンテンツを保護する再帰的セキュリティプロトコルのための方法であって、

第１の暗号化アルゴリズムによってビットストリームを暗号化することと、

第１の暗号解読アルゴリズムを該暗号化されたビットストリームと関連付けることと、

第２のビットストリームを生じるように、第２の暗号化アルゴリズムによって該暗号化されたビットストリームおよび該第１の暗号解読アルゴリズムの両方を暗号化することと、

第２の暗号解読アルゴリズムを該第２のビットストリームに関連付けることと

を含む、方法。

（項目２）

前記第１の関連暗号解読アルゴリズムおよび前記第２の関連暗号解読アルゴリズムによって前記第１のビットストリームおよび前記第２のビットストリームを解読することをさらに含み、該暗号解読は、標的ユニットによって達成される、項目１に記載の方法。

（項目３）

前記解読することは、各暗号解読アルゴリズムと関連付けられるキーを使用して行われる、項目２に記載の方法。

（項目４）

前記キーは、前記標的ユニットのハードウェア内に常駐するか、または該キーは、サーバから取り出される、項目３に記載の方法。
（項目５）
前記キーは、キーデータ構造に含有される、項目４に記載の方法。
（項目６）
前記キーデータ構造はまた、タイムスタンプおよびカウントダウン値を含有する、項目５に記載の方法。
（項目７）
前記キーデータ構造は、一時キーまたは一次秘密キーを使用して暗号化される、項目５に記載の方法。
（項目８）
前記キーデータ構造は、キーリストデータ構造内に含有される、項目７に記載の方法。
（項目９）
前記標的ユニットは、前記キーリストデータ構造のコピーを維持する、項目８に記載の方法。
（項目１０）
前記一時キーまたは前記一次秘密キーを使用して前記キーリストデータ構造を暗号化することをさらに含む、項目９に記載の方法。
（項目１１）
各暗号化アルゴリズムは、対称キーシステムまたは非対称キーシステムである、項目３に記載の方法。
（項目１２）
第１のメッセージ認証コード（ＭＡＣ）または第１のデジタル署名を各暗号化されたビットストリームと関連付けることをさらに含む、項目３に記載の方法。
（項目１３）
ハッシング関数および公開キーを使用して前記第１のＭＡＣまたは第１のデジタル署名を生成することをさらに含む、項目１２に記載の方法。
（項目１４）
各暗号化されたビットストリームが真性であることを検証することをさらに含む、項目１３に記載の方法。
（項目１５）
各暗号化されたビットストリームを検証することは、
第２のＭＡＣまたは第２のデジタル署名を生成することと、
該第２のＭＡＣまたは第２のデジタル署名を前記第１のＭＡＣまたは第１のデジタル署名と比較することと
をさらに含む、項目１４に記載の方法。
（項目１６）
解読されたビットストリームを検証することをさらに含む、項目１５に記載の方法。
（項目１７）
前記解読されたビットストリームを検証することは、第３のＭＡＣまたは第３のデジタル署名を使用して行われる、項目１６に記載の方法。
（項目１８）
前記キーの暗号化されたバージョンを前記サーバに返送することをさらに含む、項目１７に記載の方法。
（項目１９）
デジタルコンテンツを保護する再帰的セキュリティプロトコル用のシステムであって、
第１の暗号化アルゴリズムによってビットストリームを暗号化することと、
第１の暗号解読アルゴリズムを該暗号化されたビットストリームに関連付けることと、
第２のビットストリームを生じるように、第２の暗号化アルゴリズムによって該暗号化されたビットストリームおよび該第１の暗号解読アルゴリズムの両方を暗号化することと

、
第２の暗号解読アルゴリズムを該第２のビットストリームと関連付けることと
のために動作可能である、システム。
（項目２０）
前記第１の関連暗号解読アルゴリズムおよび前記第２の関連暗号解読アルゴリズムによって、前記第１のビットストリームおよび前記第２のビットストリームを解読するためにさらに動作可能であり、該暗号解読は、標的ユニットによって達成される、項目１９に記載のシステム。
（項目２１）
前記解読することは、各暗号解読アルゴリズムと関連付けられるキーを使用して行われる、項目２０に記載のシステム。
（項目２２）
前記キーは、前記標的ユニットのハードウェアに常駐するか、または該キーは、サーバから取り出される、項目２１に記載のシステム。
（項目２３）
前記キーは、キーデータ構造に含有される、項目２２に記載のシステム。
（項目２４）
前記キーデータ構造はまた、タイムスタンプおよびカウントダウン値を含有する、項目２３に記載のシステム。
（項目２５）
前記キーデータ構造は、一時キーまたは一次秘密キーを使用して暗号化される、項目２３に記載のシステム。
（項目２６）
前記キーデータ構造は、キーリストデータ構造に含有される、項目２５に記載のシステム。
（項目２７）
前記標的ユニットは、前記キーリストデータ構造のコピーを維持する、項目２６に記載のシステム。
（項目２８）
前記一時キーまたは前記一次秘密キーを使用して前記キーリストデータ構造を暗号化するためにさらに動作可能である、項目２７に記載のシステム。
（項目２９）
各暗号化アルゴリズムは、対称キーシステムまたは非対称キーシステムである、項目２１に記載のシステム。
（項目３０）
第１のメッセージ認証コード（ＭＡＣ）または第１のデジタル署名を各暗号化されたビットストリームに関連付けるためにさらに動作可能である、項目２１に記載のシステム。
（項目３１）
ハッシング関数および公開キーを使用して前記第１のＭＡＣまたは第１のデジタル署名を生成するためにさらに動作可能である、項目３０に記載のシステム。
（項目３２）
各暗号化されたビットストリームが真性であることを検証するためにさらに動作可能である、項目３１に記載のシステム。
（項目３３）
各暗号化されたビットストリームを検証することは、
第２のＭＡＣまたは第２のデジタル署名を生成することと、該第２のＭＡＣまたは第２のデジタル署名を前記第１のＭＡＣまたは第１のデジタル署名と比較することとをさらに含む、項目３２に記載のシステム。
（項目３４）
解読されたビットストリームを検証するためにさらに動作可能である、項目３３に記載

のシステム。
（項目３５）
前記解読されたビットストリームを検証することは、第３のＭＡＣまたは第３のデジタル署名を使用して行われる、項目３４に記載のシステム。
（項目３６）
前記キーの暗号化されたバージョンを前記サーバに返送するためにさらに動作可能である、項目３５に記載のシステム。
（項目３７）
第１の暗号化アルゴリズムによってビットストリームを暗号化することと、
第１の暗号解読アルゴリズムを該暗号化されたビットストリームに関連付けることと、
第２のビットストリームを生じるように、第２の暗号化アルゴリズムによって該暗号化されたビットストリームおよび該第１の暗号解読アルゴリズムの両方を暗号化することと、
第２の暗号解読アルゴリズムを該第２のビットストリームと関連付けることと
のために翻訳可能な命令を含有する、デジタルコンテンツを保護する再帰的セキュリティプロトコル用のソフトウェアシステムまたはコンピュータプログラム。
（項目３８）
前記第１の関連暗号解読アルゴリズムおよび前記第２の関連暗号解読アルゴリズムによって前記第１のビットストリームおよび前記第２のビットストリームを解読するためにさらに翻訳可能であり、該暗号解読は標的ユニットによって達成される、項目３７に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目３９）
前記解読することは、各暗号解読アルゴリズムと関連付けられるキーを使用して行われる、項目３８に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４０）
前記キーは、前記標的ユニットのハードウェア内に常駐するか、または該キーは、サーバから取り出される、項目３９に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４１）
前記キーは、キーデータ構造に含有される、項目４０に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４２）
前記キーデータ構造はまた、タイムスタンプおよびカウントダウン値も含有する、項目４１に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４３）
前記キーデータ構造は、一時キーまたは一次秘密キーを使用して暗号化される、項目４１に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４４）
前記キーデータ構造は、キーリストデータ構造内に含有される、項目４３に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４５）
前記標的ユニットは、前記キーリストデータ構造のコピーを維持する、項目４４に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４６）
前記一時キーまたは前記一次秘密キーを使用して前記キーリストデータ構造を暗号化するためにさらに翻訳可能である、項目４５に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４７）
各暗号化アルゴリズムは、対称キーシステムまたは非対称キーシステムである、項目３９に記載のソフトウェアシステムまたはコンピュータプログラム。

（項目４８）
第１のメッセージ認証コード（ＭＡＣ）または第１のデジタル署名を各暗号化されたビットストリームと関連付けるためにさらに翻訳可能である、項目３９に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目４９）
ハッシング関数および公開キーを使用して前記第１のＭＡＣまたは第１のデジタル署名を生成するためにさらに翻訳可能である、項目４８に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目５０）
各暗号化されたビットストリームが真性であることを検証するためにさらに翻訳可能である、項目４９に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目５１）
各暗号化されたビットストリームを検証することは、
第２のＭＡＣまたは第２のデジタル署名を生成することと、
該第２のＭＡＣまたは第２のデジタル署名を前記第１のＭＡＣまたは第１のデジタル署名と比較することと
をさらに含む、項目５０に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目５２）
解読されたビットストリームを検証するためにさらに翻訳可能である、項目５１に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目５３）
前記解読されたビットストリームを検証することは、第３のＭＡＣまたは第３のデジタル署名を使用して行われる、項目５２に記載のソフトウェアシステムまたはコンピュータプログラム。
（項目５４）
前記キーの暗号化されたバージョンを前記サーバに返送するためにさらに翻訳可能である、項目５３に記載のソフトウェアシステムまたはコンピュータプログラム。
【００２０】
本発明のこれらの側面および他の側面は、以下の説明および添付図面と併せて考慮すると、より良好に認識および理解されるであろう。しかしながら、以下の説明は、本発明の種々の実施形態およびその多数の具体的詳細を示す一方で、限定ではなく例証として挙げられることを理解されたい。その精神から逸脱することなく、本発明の範囲内で、多くの置換、修正、追加、および／または再配設を行ってもよく、本発明は、全てのそのような置換、修正、追加、および／または再配設を含む。
【図面の簡単な説明】
【００２１】
本明細書に付随し、その一部を形成する図面は、本発明のある側面を示すように含まれる。本発明、および本発明に提供されるシステムの構成要素および動作のより明確な概念は、図面に図示された、例示的な、したがって非限定的な実施形態を参照することによってより容易に明白となり、図中、同一参照数字は同じ構成要素を指定する。本発明は、本明細書で提示される説明と組み合わせて、これらの図面のうちの１つ以上を参照することによって、より良好に理解することができる。図面に図示された特徴は、必ずしも一定の縮尺で描かれていないことに留意されたい。
【図１】図１は、セキュリティプロトコルエンジンの実施形態のブロック図である。
【図２】図２は、暗号解読キーデータ構造の実施形態の表示である。
【図３】図３は、セキュリティプロトコルの暗号化および配布工程の実施形態の図である。
【図４】図４は、セキュリティプロトコルの実施形態に対する暗号解読およびローディング工程の図である。

【図５】図５は、セキュリティプロトコルの暗号化／暗号解読工程の一実施形態の図である。
【図６】図６は、キーリストデータ構造の実施形態の表示である。
【図７Ａ】図７Ａは、一時キー所有権譲渡手順の実施形態の図である。
【図７Ｂ】図７Ｂは、一時キー所有権譲渡手順の実施形態の図である。
【発明を実施するための形態】
【００２２】
添付図面に図示され、以下の説明で詳述される非限定的な実施形態を参照して、本発明ならびにその種々の特徴および有利な詳細をより全体的に説明する。不必要に本発明を細部にわたって判り難くしないように、周知の出発材料、処理技法、構成要素、および同等物の説明は省略する。しかしながら、詳細な説明および具体例は、本発明の好ましい実施形態を示す一方で、限定ではなく例証として挙げられることを理解されたい。基本的な発明概念の精神および／または範囲内である種々の置換、修正、追加、および／または再配設は、本開示から当業者にとって明白となるであろう。
【００２３】
ここで、デジタルコンテンツを保護することを目的とする、セキュリティプロトコル用のシステムおよび方法に注意を向ける。これらのセキュリティプロトコルは、任意のデジタルコンテンツに使用可能であり、また、実際のデジタルコンテンツが改変されることを必要とせずに従来の透かし方式に通常は関連付けられる、同一性追跡の概念を支援することもできる。これらのプロトコルは、全てのデジタルビットストリームが平等であるという前提に基づくため、プロトコル自体の更新へのアクセスを制御するために、再帰的様式で使用することさえできる。言い換えれば、データが保護されるメディアストリームであろうと、それらのストリームを再生する必要がある実行可能コードであろうと、それらのストリームを再生する必要がある暗号化された実行可能コードであろうと、それらのストリームを再生する必要がある暗号化されたコードを解読する必要がある、実行可能コードであろうと、暗号解読コードとともに使用されるキー等であろうと、プロトコルは、デジタルデータの種類を区別しない。これらのデータのデジタル性質のみが、プロトコルにとって重要なものである。したがって、デジタルデータの性質および／または使用は、セキュリティプロトコルには関係ないため、プロトコルは、自らを保護することが可能である。
【００２４】
この能力は、実行中でさえも、それが作動しているハードウェアの変更を必要とせずに、（例えば、最近発見されたセキュリティホールを修繕するように）セキュリティプロトコルを更新できることを意味する。「古いほうの」セキュリティシステムは、新しいほうのセキュリティシステムの一部として「包含される」（すなわち、システム全体の新しくより安全なレベルの保護を追加するために、古い保護「包装物」を決して剥がさなくてもよい）。したがって、システム全体は、最新で最も安全な暗号化および／またはアクセス制御システムにカプセル化される。新しいキーが追加されてもよいだけではなく、既存のシステムの最上部にも同様に、完全に新しいセキュリティおよび／または暗号化アルゴリズムを追加することができる。
【００２５】
この融通性は、プロトコルが、時間限定レンタル、ペイパービュー、多重バージョニング、機械依存性ライセンス取消、あるユーザから別のユーザへの所有権の永久譲渡を含む、多数のビジネスモデルを支援できるようにする。
【００２６】
著作権のあるソフトウェアアプリケーションが例示的実施形態で利用されるものの、ビデオ、音声データ、ソース、およびオブジェクトコードを含む、あらゆるビットストリームにセキュリティを提供するために、同じ方法およびシステムを使用できることが、当業者によって理解されるであろう。
【００２７】

セキュリティプロトコルの実施形態が提供するように設計される基本機能は、以下を含む（しかし、それらに限定されない）。
公正使用（「タイムシフト」、「スペースシフト」、およびアーカイブバックアップ）
増分アップグレード
所有権の一時譲渡
所有権の永久譲渡
時間限定アクセス
使用限定アクセス（使用される回数）
デバイス特有のライセンス取消
データまたはストリーム特有のライセンス取消
上記のように、（少なくとも最新の著作権コントロールシステム用の）著作権のある作品に含有される知的財産の保護のための一次機構は、単純に、アクセス制御である。しかしながら、そのような機構が回避された場合、最も精巧なアクセス制御機構によって与えられる保護さえ、少ししか価値がない。これは、アクセス制御が役に立たない機構であると言うことではなく、それ自体では全セキュリティシステムではないと言うことである。多数の著作権のあるメディアストリームがインターネット上で公共消費に自由に利用可能であるという事実は、そのようなセキュリティシステムをほとんど常に回避できるという事実の証拠である。この種類のアクセス制御はまた、原本が破壊される危険がある場合に必要な、合法的に購入された著作権のある作品のバックアップコピーを作製する機構を確立することをより困難にする。したがって、本明細書で説明されるセキュリティプロトコルは、有用にするために、いかなる種類のアクセス制御システムも必要としない。

【００２８】

説明されるセキュリティプロトコルは、作品自体を構成するデジタルデータではなく、著作権のある作品の表現を制御することに専念する。そのようなものとして、プロトコルは、著作権のある作品、またはその作品がどのように解釈されるかを説明するために使用される他のデジタルデータをカプセル化するために使用される、デジタルデータを区別しない。結果として、プロトコルは、他のセキュリティプロトコルをカプセル化するためにさえ使用することができる。

【００２９】

（基本的動作説明：）

セキュリティプロトコルの実施形態は、それらのコードが、そのアルゴリズムをコピーあるいは不正流用しようとする者による分解から保護されるという高い信頼度を、１つのソフトウェアの作者が有することを可能にするように設計される。それらはまた、その機能性を改変しようと試みる者による修正からこのコードを保護するようにも設計される。他の汎用コンピューティングシステムにおいてこれらの主要な特性を実装することができる方法のうちの１つを、以下の項で論議する。これら２つの主要機能の副産物として発生する、付加的財産は、ソフトウェアを作動させることができる条件（すなわち、いつ、どのように、およびどの１つまたは複数の機械上で、コードが実行されることが可能となるか）を制御する能力である。これらの機能のうちの第１は、システムに改ざん防止タイマ要素を追加することによって達成されてもよい。他は、問題のブロックコードを実行するために満たされなければならない、所望の条件を示すために使用される、安全なデータ構造を実装する手段によって達成される。このデータ構造はハードウェア特有ではないため、種々の方法で使用することができ、それを解釈するために使用されるソフトウェアを更新することによって修正されることが可能である。より効率的にプロトコルを実装するために利用される、ハードウェア特有の特徴を論議し、プロトコルを支援するために、これらの特徴をどのように使用できるかという例を挙げる。最後に、著作権のある作品を保護するためにプロトコルをどのように使用できるかを示す。

【００３０】

セキュリティプロトコルの実施形態は、その意図された受信者によって解読されることのみを可能にするような方法で、コードのブロックを暗号化する能力に依存する。これは

、よく理解された問題であり、多数の業界標準暗号化アルゴリズムの基礎である。しかしながら、プロトコルのコアが典型的なオンチップ命令キャッシュ（Ｉ－キャッシュ）の（比較的）小さい範囲内に適合できれば役立つという事実、および半自律的方式で作動することが可能であるという事実といった、セキュリティプロトコルの実施形態とともに使用するために考慮されるべき２つの付加的な要因がある。言い換えれば、プロトコルが小さく、通常の日常動作に中央セキュリティサーバの使用を必要としなければ、有用である。

【００３１】

（ハードウェア：）

セキュリティプロトコルシステムの要素は、プロトコルエンジン（「標的ユニット」としても知られる）１００上で安全な方式でプロトコルを実装する、１組のハードウェアブロックを含んでもよい。このセキュリティプロトコルを実行することが可能なデバイスの一例の全体的なブロック図を図１に示す。これらのブロックは、プロトコルが正しく動作するために、ハードウェアに投入される必要はないが、下記のハードウェア要素の全てを含むデバイスは、最小限の諸経費でプロトコルを実装することが可能となる。

【００３２】

これらのハードウェアブロックのうちの第１は、リアルタイムクロック１０２である。これは、中央サーバとの安全な相互作用によって設定またはリセットされることが可能な、自由作動タイマである。これは完全に必須のブロックではないが、安全な時間標準のクエリを行うことによって時間が確立されてもよいため、この機能をオンチップにすることがより便利となる。これは、時間依存性ソフトウェアライセンスと関係があり、そのようなものの例を本書の以降の項で挙げる。

【００３３】

別のハードウェア要素は、実行されるコードをオンチップで記憶することができるメモリ１１０のブロックである。これは、典型的にはＩ－キャッシュとして知られているが、いくつかの実施形態では、このＩ－キャッシュ１１０の各部分の重要な特性は、あるブロックに含有されるデータがＣＰＵ実行ユニット１２０のみによって可読であることである。言い換えれば、Ｉ－キャッシュメモリ１３０のこの特定のブロックは、実行専用であり、いずれのソフトウェアによっても、読み取ることも書き込みすることもできない。Ｉ－キャッシュのこの特別なセクションを「保護されたコードブロック」１３０と呼ぶ。実行されるコードが、この保護されたＩ－キャッシュブロック１３０に実際に蓄積される方式は、別のハードウェア要素を介するものであってもよい。

【００３４】

加えて、安全なコードブロックの動作を増進するために使用することができる、他のカテゴリの可能な「強化」がある。これらのうちの１つは、ＣＰＵ１２０が安全なコードを実行している間のみアクセス可能である、および／または安全なコードブロックの実行の完了時（または、何らかの理由で、実行ユニットが「通常の」Ｉ－キャッシュからコードの任意のセクションに飛んだ場合）に消去される、１組（一部）のＣＰＵレジスタ１４０を指定する能力である。たとえ「保護された」コードおよび「保護されていないコード」の混合物を実行するＣＰＵ１２０の可能性がないように思われる場合があるとしても、割り込みルーチンに飛ぶ時にコンテキストを切り替える工程に何が起こり得るか、およびＣＰＵ１２０コンテキストがどこに記憶されるか（ほとんどのＣＰＵはコンテキストを主要メモリに記憶し、そこでは潜在的に、保護されていないコードブロックによって、以降の時点で発見の対象となる）を常に考慮しなければならない。

【００３５】

別の可能性（どのレジスタ１４０が消去されるかを保護されたコードブロックの作者が明示的に識別することを必要とする以外）は、それを自動的に行わせることである。これは、ＣＰＵ実行ユニット１２０が、保護されたコードブロックの内側で実行している間にどのレジスタ１４０が読み取られるか、または書き込まれるかを追跡し、次いで、「安全」モードの終了時に自動的に消去する場合となる。これは、２種類のコードブロックの間で共有されることが許可されているデータのみが原型を保つように、保護されたコードが

自ら迅速に「一掃」できるようにする。「自動」工程は、「明示的」手順よりも潜在的に安全となる場合があるが、コードの作者が保護されたコードブロックと保護されていないコードブロックとの間で情報を共有することを希望する場合をより複雑にする場合がある。

【００３６】

安全なコードセグメントと安全ではないコードセグメントとの間のレジスタ記憶データの「漏出」に対処するための別の潜在的な方式は、ＣＰＵ１２０が保護されたコードを実行している時にのみ使用される、一意の１組のレジスタを識別することである。大型汎用レジスタセット１４０を伴う一部のＣＰＵアーキテクチャにとって、これは、最初はひどく高価に思われるかもしれない。しかしながら、多くの現代ＣＰＵで実践されている、レジスタ名変更およびスコアボード構造の修正されたバージョンを使用することによって、とてつもない量の諸経費を必要とせずに（すなわち、物理的に異なった１組の「安全な」レジスタを実装するステップに関与するシリコンの諸経費なしで）、同じ効果が達成され得る。保護されたコードブロックの実行をアトミック作用（すなわち、割り込み可能ではない）として扱う場合、これらの問題点は対処しやすいが、この利便性は、性能および潜在的な全体のコード複雑性を犠牲にする場合がある。Ｉ－キャッシュの「保護された」部分１３０は、Ｉ－キャッシュの「通常の」部分１５０とは異なるＣＰＵへのデータパスを必ずしも必要としないことに留意されたい。実際、２つは完全に同義となり得る。

【００３７】

一方向ハッシュ関数ブロック１６０も示されている。ハードウェアにおいてこの機能性を実装する必要なく、セキュリティプロトコルの実施形態を実行することが可能となるエンジンを構築することが可能である。しかしながら、ハッシングアルゴリズムのある部分用のハードウェアアクセラレータは、確かに望ましい特徴である。この関数ブロックのハードウェアおよびソフトウェア実装間のトレードオフを以降で論議する。

【００３８】

標的ユニット１００の別の部分は、実行可能コードブロックに翻訳するために、暗号化されたメッセージに影響するように標的ユニット１００の秘密キーおよび公開／プライベートキー（以降で説明）を使用する、ハードウェア補助の暗号解読システム１７０であってもよい。この暗号解読システム１７０は、多数の方法で実装することができる。プロトコル全体の速度およびセキュリティは、このブロックの構造に依存し得るため、それはセキュリティシステム更新に適応するように十分に融通性があり、かつ、システムがリアルタイムコード更新を行うことを可能にするように十分に高速となるべきである。

【００３９】

これら２つの制約を念頭に置くと、正確にどの暗号化アルゴリズムがこのハードウェアブロック１７０に使用されるかは、プロトコルにとって重要ではない。最大限の融通性を促進するためには、実際のハードウェアは、非アルゴリズム特有の方式で使用されるように十分に汎用であるが、この機構を実装することができる多くの異なる手段があると想定される。

【００４０】

また、点線でブロック図に示されたオンチップ乱数発生器１８０があることにも留意されたい。このブロックはオプションである。加えて、それは、ソフトウェアを用いた疑似乱数発生システムのシード値を供給するために使用することができる、十分な乱数の配列を産生する好適なオンチップ方法に置換することができる。この疑似乱数発生器はまた、ハードウェアまたは「安全な」ソフトウェアに潜在的に実装することもできる。当然ながら、ハードウェア実装と対比した、ソフトウェアを用いたシステムの融通性の同じ原理のトレードオフが、この場合にも該当する。しかしながら、標的デバイス１００が乱数を生成しなければならない場合は、このプロトコルで頻繁に発生しないため、この特定の機能がハードウェア加速型ではない場合に、全体的な性能に影響を及ぼす可能性は低い。

【００４１】

（秘密キー：）

各プロトコルエンジン（「標的」ユニット）１００は、オンチップで記憶される、２組の秘密キー定数１０４を有してもよく、そのどちらの値もソフトウェア可読ではない。これらのキーの第１（一次秘密キー）は、１組の秘密キーとして実際に編成されてもよく、その１つのみが、任意の特定の時に可読である。ユニットの「所有権」が変更される（例えば、チップを含有する機器が販売あるいは譲渡される）場合、現在動作中の一次秘密キーは、「消去される」か、あるいは異なる値によって上書きされてもよい。この値は、安全な方式でユニットに転送することができるか、または、この第１のキーが消去される時にしか使用されないような方式でユニットにすでに記憶することができるかのいずれかである。事実上、これは、その所有権が変更される時、またはそのような変更の何らかの他の理由（障害が起きたキー等）がある場合に、その特定のユニットに新しい一次秘密キーを発行するステップと同等である。この一次秘密キー値が記憶される唯一の他の場所は、使用許諾権限における中央サーバ上にある。

【００４２】

一次秘密キーは、中央サーバのデータベース中の特定の標的ユニット１００のシリアル番号１０６と関連付けられてもよい。シリアル番号１０６は、標的デバイス１００上のどこに記憶されてもよく、ソフトウェアアクセス可能であってもよく、一次秘密キーとは他の関係がない。ユニットの動作側面の任意の更新（セッキュリティシステムの更新等）は、一次秘密キーを使用することによって達成されてもよい。このキーの値が、標的ユニット１００および使用許諾権限以外の関係者によって知られていない場合、それは、安全な中央サーバを通るリンクを伴わない、いずれの安全なトランザクションにも使用することができない。しかしながら、この一次キーのセキュリティは、最重要であるため、絶対的に必要なときのみに使用されるべきである。したがって、それはおそらく、例えば、中央使用許諾権限のサーバと標的ユニットとの間の安全なトランザクションのための通信リンクを暗号化するために使用されるべきではない。このリンクは、現在認められている標準的実践に従って、その場で生成される一時キーを使用する標準キー交換プロトコルを使用して保護することができる。

【００４３】

２次秘密キーは、標的ユニット１００自体にのみ知られてもよい（したがって、使用許諾権限には知られない）。標的ユニット１００のＣＰＵ１２０は、１次または２次秘密キーのいずれかの値にアクセスすることが全くできないため、ある意味、標的ユニット１００は、自らの秘密キー１０４さえも「知らない」。これらのキーは、標的ユニット１００のＣＰＵ１２０のセキュリティブロック内でのみ記憶および使用される。標的ユニットの全体的なセキュリティを強化するのは、これらの秘密キーの両方の組み合わせである。それらがどのように使用されるかを以降に説明する。

【００４４】

別の１組のキーが、一時公開／プライベートキーシステム（非対称キーシステムまたはＰＫＩシステムとしても知られる）の一部として動作してもよい。このペアのキー１０８は、その場で生成され、中央サーバの干渉なしに、同様のユニット間の安全な通信リンクを確立するために使用される。そのようなシステムのセキュリティは、典型的には、同等のキー長さの対称キー暗号化システムよりも低いため、これらのキー１０８は、上記の１組の秘密キー１０４よりもサイズが大きくなければならない。これらのキー１０８は、オンチップタイマブロックに存在する値と併用されてもよい。これらのキーはその場で生成されるため、それらが生成される方式は、ある種の乱数発生システム１８０に依存している。最後に、生成されたキー１０８がいわゆる「弱い」キーの部類に含まれないことを確実にするように注意を払わなければならないことに留意されたい。「弱い」と見なされる特定の１組のキーは、使用される特定の暗号化アルゴリズムに依存している。

【００４５】

（動作詳細：）

セキュリティプロトコルの実施形態が動作する方式は、システム初期化、安全なコード生成および大量配布、安全なコードローディングおよび実行、キーリストデータ構造の構

成、一時ライセンス譲渡、永久ライセンス譲渡、システム所有権譲渡、ライセンス取消、およびセキュリティシステム更新といった、いくつかの異なる工程に分類することができる。これらのそれぞれを順に論議する。しかしながら、下記の例は、論議を簡素にする目的で選択され、必ずしもこのプロトコルを実装することができる最も効率的な方式ではない（または唯一の方式でもない）ことに留意されたい。

【００４６】

（システム初期化）

これは、標的ユニット１００の１０４秘密キーが何らかの初期値に設定されるステップである。この手順は、２つの秘密キーのうちのいずれかに対するいくつかの場所のうちの１つで達成することができるが、ロジスティックな理由によって、シリアル番号１０６または秘密キー１０４のいずれかが変更される可能性があり得るのは、組立工程の最後のステップとなるべきである。ユニット１００のシリアル番号１０６がオフチップで記憶される場合、この手順は、最終組立の時点で行われる可能性が最も高い。ユニットのシリアル番号１０６がオンチップに記憶される場合、チップ製造工程の最後の時点で（すなわち、チップが包装された後）この手順を実行することが最も実用的となるため、生産後またはバーンイン副産物には、機能しない部分を選別する機会がある。こうして、安全に保たれなければならないデータの量が最小化される。プロトコル全体のセキュリティは、ユニット１００の秘密キー１０４のセキュリティに基づいてもよいため、初期化手順は、物理的セキュリティが可能な時点で着手されるべきである。

【００４７】

一次秘密キーは、２次秘密キーを供給するために使用されるものとは異なる手順で初期化される（またはデバイスに「焼き付けられる」）べきである。実践では、この２次キーは何らかの時点で知られるが（製造工程の何らかの時点でユニットにプログラムされるため）、一旦、それが標的デバイス１００上に記憶されると、それが関連付けられるユニットはどこにも記録されるべきではない。監査目的で、（配布の無作為性を試験するため、または何らかの他の理由で）どの部分がどのキーを保持するのかを知っていることとは無関係に、全１組の２次秘密キー値が検査されることが潜在的に望ましてくもよい。しかしながら、システムの安全な性質を維持するために、この２次秘密キーをユニットにプログラムするデバイスは、２次秘密キーを第１の秘密キーまたは標的デバイス１００のシリアル番号１０６のいずれかに関連付ける手段を決して持たないことが望ましい。また、これらの秘密キー１０４の両方は、以降で説明する理由により、改ざん防止方式で実装されるべきである。これらの２つの秘密キー１０４がどの順番で初期化されるかは重要ではない。例示的な実施形態で説明される初期化手順後、標的デバイス１００のシリアル番号１０６およびそれらの関連一次秘密キー１０４が共同して位置する唯一の場所（実際のチップ上以外）は、使用許諾権限５１０における安全なサーバ上にあるべきである。

【００４８】

（安全なコード生成および大量配布）

一例のシナリオにおいて、合理的に分解の心配がなく、特定のデバイス上でのみ実行することができる、このプロトコルの下で作動するようにアプリケーションを生産することを開発者５２０が希望すると仮定する。各登録された開発者５２０は、使用許諾権限のサーバ５１０と通信するために、および開発者によって発行されたあらゆる暗号化されたコードブロックを認証するために使用することができる署名されたメッセージ認証コード（典型的には、デジタル署名またはＭＡＣと呼ばれる）を作成するために使用する、あらゆるメッセージを認証するために使用される公開キー／プライベートキーペアを有する。

【００４９】

アプリケーションは、デバッグされた後、本来の開発者のみに知られているアプリケーション特有の暗号化アルゴリズムおよびキーを使用して符号化される。このアプリケーション特有のアルゴリズムおよびキーは、非対称（秘密）キーシステムまたは非対称（ＰＫＩ）キーを用いたシステムのいずれかとなり得る。暗号化されたコードのブロックの端には、それらの発行された公開キー／プライベートキーペアのプライベートキー（したがっ

て、暗号化されたコードブロックに対する一義的デジタル署名を形成する）を使用して、開発者５２０によって署名されるデジタル署名（またはＭＡＣ）が添付される。デジタル署名および対応するコード特有のＩＤ番号を形成するように暗号化された、デジタル署名または本来のＭＡＣのいずれかは、使用許諾権限に供給されてもよい。アプリケーション開発者５２０はまた、同様に適切な復号キーを供給することを選択してもよい（この決定のトレードオフを本書の以降の項で論議する）。

【００５０】

アプリケーション特有のアルゴリズムは、非対称暗号化である場合、メッセージ認証コードを生成するために使用される同じ発行されたＰＫＩキーペアを使用して、必ずしも暗号化される必要がないことに留意されたい。しかしながら、コードブロックの端において記憶されるＭＡＣは、既知のハッシングアルゴリズムを使用して生成されるべきであり、また、開発者の発行された公開キーのうちの１つを使用して署名されなければならない（したがって、デジタル署名を形成する）。これにより、標的が既知のハッシング関数および公開キーを使用してＭＡＣの真正性を検証できるようにする。

【００５１】

全てのアプリケーション特有の暗号化キーデータ構造２１０は、（暗号解読キー自体２２０に加えて）多数の余分なフィールドを含有してもよい。これらのフィールドのうちの１つは、タイムスタンプ２３０と、関連マスク値２４０とを備える。第２は、「カウントダウン値」２５０を含有する。マスク値２４０は、キーが有効である時を判定するために、他の２つのフィールド２３０、２５０と併用される。この構造の一例を図２に示す（しかし、この同じ機能性を実装することができる多数の手段がある）。また、正確にいくつのビットがフィールドのそれぞれに割り付けられるかはプロトコルに関連しないことにも留意されたい。

【００５２】

タイムスタンプ値２３０は、タイムスタンプマスク２４０フィールドに記憶されるビットパターンに応じて、いくつかの方法で使用できることに留意されたい。タイムスタンプマスク２４０値は、標的ユニット１００の現在の時間との比較を行う時に無視される、一部のタイムスタンプ数字を開発者５２０が選択できるようにする。しかしながら、一例として、タイムスタンプフィールド２３０によって支援される最小分解能が１秒であると想定する場合、タイムスタンプデータ２３０の下位５ビットをマスクすることによって、タイムスタンプフィールド２３０に記憶される時間から始まる、約３２秒の経過にわたって使用される時のみに有効な、特定のキーデータ構造２１０を生成することができる。セキュリティプロトコルの全体的な機能性は、タイムスタンプフィールド２３０の最下位ビットの実際の分解能に依存していない。

【００５３】

マスクフィールド２４０に関連付けられる他のビットがあってもよく、そのいくつかは、タイムスタンプ２３０で特定された値の前後でキーが有効かどうかを示すために使用することができる。タイムスタンプ２３０および「カウントダウン」値２５０がどのように関連付けられるかを示すために、さらに別のマスクフィールド２４０ビットを使用することができる。例えば、これは、アプリケーション開発者５２０の意向が、ある日付および時間窓に単純に関係するよりもむしろ、ある日付の前または後のいずれかに、ソフトウェアの使用をある反復回数に限定することであった場合に有用となる。当然ながら、これらの条件の任意の組み合わせを構築することができるため、プロトコルは、この点で極めて融通性がある。加えて、いくつのキーの合法コピーが本来の標的ユニット１００から他者へ同時に配布されてもよいか等、他の特性を示すように、さらなるフラグをこのデータ構造に含むことができる。これは、例えば、デジタルライブラリで見られるような、多重コピーライセンスが所望とされた場合に有用となる。

【００５４】

暗号化工程の一実施形態を表すフロー図を、次のページの図３で見ることができる。デジタルメディアストリームまたはソフトウェアアプリケーション（そのメディアストリー

ムを解釈するために使用される暗号解読命令等）を配布するために使用される工程に実質的な差異がないことに留意されたい。いずれにしても、オンラインサーバを介するか、または直列化されたディスク（標準ＤＶＤ等）上のいずれかで、暗号化されたコードブロック３１０、３２０を配布するためのいくつかの異なるオプションがある。後者の場合、開発者５２０は、使用許諾権限５１０により、大量生産されたディスクの個別シリアル番号を事前登録する（またはしない）ことを選択することができる。そうする場合、シリアル番号は、バーストカッティングエリア（ＤＶＤの場合）に焼き付けることによるか、または標準ＣＤの場合はインクジェットインプリンティングによるかのいずれかによって、ディスクに永久的に取り付けられ得る。同じシリアル番号が大量生産されたディスクの全てにおいて複製されるため、開発者５２０は、ＣＤまたはＤＶＤのデータエリアにこれらのシリアル番号を埋め込むことができないことに留意されたい。ある種の混成形式が使用された場合で、ディスクの一部が大量生産され、別の部分が一度書き込まれ得る場合、これは、個別シリアル番号を伴うディスクを配布する別の潜在的方法となる。どのような場合でも、登録工程でエラーしにくいため、機械可読シリアル番号が確かに好ましい。

【００５５】

開発者５２０が使用許諾権限５１０によりメディアシリアル番号の使用を許諾しないことを選択する場合、適正な暗号化キーをアプリケーションまたはメディアストリームファイルに関連付けることができる、何らかの他の方式があってもよい。したがって、アプリケーション開発者５２０は、コード特有のＩＤまたは関連メディアシリアル番号のいずれかを登録してもよい。前者の場合、アプリケーションを自由に配布することができる（すなわち、特定の公表フォーマットおよびメディアに関係しない）。

【００５６】

個別シリアル番号機構の場合、使用許諾権限５１０には、どのアプリケーション（またはメディアストリーム）がどのシリアル番号に関連付けられるかという指示がないために、エンドユーザのプライバシーが維持される。開発者５２０が、その関連キーとともにアプリケーションＩＤ（またはメディアストリームＩＤ）を登録する場合、使用許諾権限５１０が、どのアプリケーションまたはメディアストリームが特定のエンドユーザによって「所有」されているかを知ることが可能である。一方で、このプライバシーの潜在的な欠如は、開発者５２０が物理的メディアを製造して配布することを必要としないという、付加的な利便性および費用節約によって相殺される。「物理的メディア」という用語は、必ずしもディスクを意味しないことに留意されたい。この機能は、個別シリアル番号ステッカーが貼り付けられた、印刷されたマニュアル（または単純な登録フォームさえ）を使用することによって、同様に達成され得る。必要とされるのは、開発者５２０がエンドユーザによって購入される、一意のシリアル番号を有する何らかの物理的対象を生産しなければならないことだけである。このシリアル番号の目的は、「購入証明」および／またはソフトウェア登録番号の役割を果たすことである。このシリアル番号がプロトコルにおいてどのように使用されるかを以下の項で論議する。

【００５７】

図３に示された例について、暗号化されたソフトウェアアプリケーション（またはメディアストリーム）３１０および機械依存性の暗号解読ソフトウェア３３０の両方は、同じ機構を使用して配布される。これが事実となるべきということはプロトコルの要件ではなく、暗号化されたコードブロック３１０、３３０のいずれかまたは両方は、オンラインで、またはディスクをプレスすることによって配布することができる。しかしながら、デジタルメディアストリームの場合、メディアストリーム自体が２つのブロック３１０、３２０の桁違いに大きい方である可能性が高い。したがって、その場合、大量生産されたディスク形式で、少なくともこのブロックの配布を達成することが最も道理にかなう。多くの場合、随伴の暗号化されたコードブロック３２０（第１のブロックをどのように復号するかという命令を含有するもの）ならびに一次の暗号化されたコードブロックに適合するように、そのようなディスク上に十分な余地があってもよい。また、２つのデータセットのいずれもが、発行後に変更を受ける可能性がないため、オンラインで配布されなければな

らないという基本要件がないことにも留意されたい。そのようなものとして、それらは両方とも、大量生産されたディスクを用いた配布機構によく適している。両方を同じディスク上に有することにより、一義的な様式で一方を他方に関連付けることがより容易となる。

【００５８】

（安全なコードローディングおよび実行）

配布機構が実際のディスクを介する場合、消費者は、従来のソフトウェア購入と正確に同じ方式で、アプリケーションを含有するディスクを購入することができる。当然ながら、エンドユーザは、「標的」ユニット１００のプロセッサ上で修正されていない暗号化されたコードブロックを作動させることができない。ユーザが機械上でアプリケーションを作動させようとすると、ＣＰＵ１２０が、暗号化されたソフトウェアブロックをロードし、問題のコードブロックが真性であることを検証するために、ソフトウェア開発者の公開キーとともにコードブロックの端において保存されたデジタル署名（「署名された」ＭＡＣ）を使用する。これは、他の汎用ＣＰＵ１２０への第１のハードウェア修正が作用し始めてもよい場合である。そのような保護されたコードのブロックをロードして解読するための工程を図４に示す。

【００５９】

ハッシング関数が正しく計算されること（および、さらには、生成されたメッセージダイジェストと「本物の」メッセージダイジェストとの間の比較が有効であること）を確認するために、ＣＰＵ１２０は、安全な方式でこのハッシング関数を実行しなければならない。したがって、ハッシング関数は、デコーダのハードウェアによって直接生成されてもよいか、またはハッシング関数自体は、「安全な」コードのブロックを使用して計算されなければならないかのいずれかであり、その動作は、他の「安全ではない」プログラムによって改ざんすることができない。

【００６０】

ソフトウェアを用いたハッシュの場合、この安全なコードのブロックは、ユニット１００のセキュリティシステムの一部と見なされるべきであり、そのようなものとして、ユニット１００と使用許諾権限５１０との間の安全なトランザクションを介してプレーヤにダウンロードすることしか可能でなくてもよいことに留意されたい。大変興味深いことには、「安全な」ハッシング関数の確立は、本明細書で説明される同じ保護プロトコルを介して達成することができる。セキュリティシステムの前側面に対するこの再帰的挙動は、このプロトコルのソフトウェアを用いたバージョンが、その暗号化／暗号解読アーキテクチャにおいて極度に融通性（したがって、更新可能）となることを可能にするものである。

【００６１】

メッセージダイジェスト計算がハードウェアに固定される場合、ある程度のセキュリティを潜在的に獲得することができるが、これは融通性を犠牲にして成り立つ。ハッシュ値を生成するために専用ハードウェアブロックが使用される場合、チップが製造された後のある時点でハッシングアルゴリズムの多少の脆弱性が発見され（またはその実装に何らかのバグがある場合に）、そのとき事後に問題に対処する機会がない。これは、処理を加速するために、ソフトウェアを用いたハッシング関数のある種のハードウェア加速（プログラム可能なＳ－Ｂｏｘ構造等）を使用できないと言うものではない。しかしながら、その場合、ハードウェアは、理想的には、多種多様の一方向ハッシング関数を支援するように十分に汎用性となるべきである。

【００６２】

しかしながら、このプロトコルのセキュリティは、最終的には、この安全なコードローディング手順の一部として提供される、最下位機能に依存している。下位機能（ハッシング関数で使用される、秘密キーまたは原始的操作等）は、署名されたメッセージダイジェスト等の上位機能性を産生するように、異なる方法で共に組み合わせられる。順に、これらの上位機能ブロックは、同一性確認等の、上位ユーティリティを提供するために使用される。より多くの原始層の最上部に上位機能を構築するこの工程は、「信用の連鎖」を構

築するステップとして知られている。システムの融通性は、セキュリティ関連機能をこの階層内でできる限り低く修正することができる点を設置することにある。しかしながら、ある点で、この連鎖が基づく基本原始的操作は、性質がアトミックでなければならない（すなわち、これは、ハードウェアに実装されなければならない機能性である）。ハードウェアのデータ塊（ｇｒａｎｕｌａｒｉｔｙ）のこの点の正確な選択は、大部分が実装詳細であり、このプロトコルの全体的動作は、上記の条件を考慮すると、この側面に依存していない。

【００６３】

一旦暗号化されたコードブロック３１０が標的１００のメモリ空間１１０にロードされ、メッセージダイジェストが計算されると、結果は、そのとき開発者の公開キーによって、暗号化されたコード３１０とともに記憶されたデジタル署名３４０を解読することによって計算されるメッセージダイジェストと比較される。２つが一致する場合、標的ユニット１００は、暗号化されたコードブロック３１０が真性である（または、少なくとも、公開キーがデジタル署名を解読するために使用された開発者５２０によってコードが配布された）と確信することができる。

【００６４】

この時点で、標的１００は、最近検証された、暗号化されたコードブロック３１０と協調して使用される、暗号解読キーのコピーを要求する使用許諾権限５１０への安全なメッセージを送信する。使用許諾権限５１０との安全な接続を設定するステップの一部として、標的ユニット１００は、一時公開／プライベートキーペア（その公開部分は使用許諾権限５１０サーバに供給される）を生成する。キー交換手順の詳細は、周知であり、これが達成される正確な機構を本論議で詳細に述べる必要はない。どのような場合でも、標的ユニット１００と使用許諾権限５１０における中央サーバとの間の全体的なネットワークトラフィックは、いくつかのキー転送、コード特有のＩＤ、およびそれとともに記憶されたＭＡＣから成るため、適度に小さなデータセットに限定されることに留意されたい。

【００６５】

コード特有のＩＤは使用許諾権限５１０が認識するものであると想定すると、アプリケーションの作者がすでに使用許諾権限５１０に要求された暗号解読キーの「暗号化されていない」コピーを提供しているか否かに応じて、２つの可能な行動方針があってもよい。開発者５２０が使用許諾権限５１０にそのような情報を提供していない場合、中央サーバは、アプリケーション開発者のサーバ５２０に標的デバイス１００の一時公開キーのコピー（ならびに、問題のコード特有のＩＤ）を伝送する。その時点で、開発者のサーバ５２０は、要求暗号解読キー（標的の一時公開キーで暗号化される）を含有するメッセージ、および適正に解読されたコードから生成されるメッセージダイジェストで、使用許諾権限５１０サーバに応答する。こうして、標的デバイス１００のみがメッセージを解読して暗号解読キーを得ることができ、使用許諾権限５１０は、暗号化されていない形の暗号解読キーにアクセスできない。

【００６６】

メッセージダイジェストは、事前計算し、使用許諾権限５２０のサーバ上に記憶することができるが、それがトランザクション中に開発者５２０によって提供されてもよいという事実は、万一ハッシング関数（メッセージダイジェストを生成するために使用される）が変化する場合に潜在的に役立つ。万一このことが起こった場合、開発者５２０は、標的デバイス１００との実際のトランザクションの前または後のいずれかにおいて、解読されたコードのメッセージダイジェストの更新されたバージョンを使用許諾権限５１０に提供する必要がある。使用許諾権限５１０は、本来の（解読された）コードに決してアクセスするべきではないため、開発者５２０は、この情報を提供しなければならない。前述のように、使用許諾権限５１０のサーバと開発者５２０のサーバとの間のネットワークトラフィックの量は、依然として極めて少ない。次いで、開発者５２０から受信された暗号化されたキーは、使用許諾権限５１０から標的への伝送の前に、標的デバイス１００の一次秘密キーでさらにもう一度暗号化される。

【００６７】
アプリケーション開発者５２０が使用許諾権限５１０と標的デバイス１００との間のトランザクションのために「ループ外」でいることを希望する場合、単純に、暗号化されていない（非暗号化）形の関連暗号解読キーのコピー、および解読されたコードブロックに対する関連ＭＡＣ（その値はハッシングアルゴリズムが変更される度に更新されなければならない）を使用許諾権限５１０に提供することができる。したがって、使用許諾権限５１０における中央サーバは、自律的に作用することが可能となり、標的ユニットからのキー要求を履行するために、開発者のサーバ５２０への通信リンクを確立する必要はない。しかしながら、これは、万一「暗号化されていないキー」情報が損なわれた場合、開発者にとって潜在的なセキュリティの危険である。

【００６８】
この場合、暗号化されていないキーは、標的デバイス１００の一時公開キーによる（上記のような）伝送、および再度、標的の一次秘密キーによる伝送の両方の前に、依然として暗号化される。この時点で、標的デバイス１００は、二重に暗号化された形式における適正な暗号解読キーを有する。使用許諾権限５１０が暗号化されていないアプリケーション特有のキー５５０情報にアクセスしない場合、各ユニット１００に対する秘密キーは、使用許諾権限５１０のみに知られるべきであり、伝送のためのプライベートキーは、標的１００のみによって知られるため、意図された標的デバイス１００以外の誰もが、暗号化されていない形でこのキーデータを複製できることが可能となるべきでない。

【００６９】
しかしながら、この時点で、標的１００がアプリケーション開発者５２０から受信する符号化された暗号解読キーは、まだ、標的１００において誰にでも見られる状態で安全に記憶する（例えば、フラッシュＲＯＭに記憶する、またはハードドライブ上にバックアップする）ことができない。問題は、標的デバイス１００が、使用許諾権限５１０から伝送された、符号化された暗号解読キーとともに、一時プライベートキーのコピーも記憶しなければならないことである。使用許諾権限５１０における誰かが、何らかの手段によってこれら２つのデータへのアクセスを獲得した場合には、（標的デバイス１００の一次秘密キーへのアクセスも同様に有する場合があると考えれば）解読されたアプリケーション特有のキー５５０を再構築することが潜在的に可能となる。

【００７０】
これは、標的デバイス１００の２次秘密キーが使用され始める時点である。２次秘密キーは、標的ユニット２００以外の誰にでも知られていないことを思い出されたい。したがって、使用許諾権限５１０から標的１００に供給されたキーを解読するために、いったん一時プライベートキーが使用されると、その使用（および／または保管）前にアプリケーション特有のキーを再暗号化するために、２次秘密キーが使用される。

【００７１】
次いで、標的は、コードブロック（またはメディアストリーム）を解読するために、アプリケーション特有の（暗号化されていない）キーを使用することができる。したがって、アプリケーションコードが暗号化されていない形で存在する、２つだけの場所は、元の開発者５２０自身、および標的デバイス１００のＩ－キャッシュ１１０の「保護された」部分の内側にある（そこでは実行することしかできず、暗号化されていない形でメモリに再び書き出すことは決してできない）。これは、ユーザと使用許諾権限５１０との間のプライバシーを可能にする。言い換えれば、使用許諾権限５１０は、ユーザがライセンスを有するものが何なのか（大きなプライバシーの問題点）を知らなくてもよいが、ユニット１００が損傷される、または盗まれる、あるいは動作不能である場合に、ユーザのキーリストに対する保存場所（またはバックアップ）の役割を果たすことが依然として可能である。

【００７２】
暗号解読工程が正しく行われたことを検証するチェックとして、適正に解読されたコードのメッセージダイジェストは、元の開発者５２０から使用許諾権限５１０を通して標的

ユニット１００へと転送された、デジタル署名と比較される。前述のように、このデジタル署名は、アプリケーション開発者のプライベートキーで、暗号化されていないコードブロックのメッセージダイジェストを暗号化することによって作成される。代替として、このデジタル署名はまた、接続が確立された時に使用許諾権限５１０に供給された、別の一時公開キー５３０を使用して、再度、開発者５２０によって暗号化することもできる。どのような場合でも、正しいメッセージダイジェストは、開発者の公開キーでデジタル署名を解読することによって、標的デバイス１００によって復号することができる。このメッセージダイジェスト解読されたコードブロックのハッシュに一致する場合、コードは真性であると見なされ、標的１００上で作動することが可能になる。次いで、このメッセージダイジェストは、再暗号化されたアプリケーション特有のキー５５０とともに保管するために、標的ユニット１００の２次キー５４０で再暗号化されてもよい。キー暗号化／暗号解読工程全体のフロー図を以下の図５で概説する。

【００７３】

この手順の最終ステップは、アプリケーション特有のキー５６０の新しく暗号化されたバージョンが、保管目的で使用許諾権限５１０サーバに再伝送されることである。この再伝送は、いくつかの目的を果たす。第１に、標的デバイス１００がコードブロックを適正に解読できたという確認である。第２に、エンドユーザがある種の壊滅的なデータ不具合を被り、アクセスキーの自らのバックアップコピーを作製することを怠っていた場合に対処するために、使用許諾権限５１０が、この暗号化されたキー５６０のコピーを有する必要がある。次いで、使用許諾権限５１０は、任意の特定のユーザに対するバックアップ記憶設備としての役割を果たすことができる。この手段のさらに別の理由は、特定の標的デバイス１００の持ち主が１人のユーザから別のユーザへ変化する場合、または標的デバイス１００をアップグレードすることを希望する場合に対処するためである。この種類の所有権の永久譲渡は、そのユニット１００に対する許諾されたアプリケーションキーの全ての移譲を伴うことができる（その場合、新しい所有者の名前の下でユニットを登録する以外に、何も行われる必要はない）。しかしながら、ユーザが第１のデバイスから第２のデバイスへキーデータの永久所有権を譲渡することを希望する場合、これは、使用許諾権限５１０と標的デバイス１００の両方との間の安全なトランザクションを用いて達成されてもよい。

【００７４】

標的デバイス１００が使用許諾権限５１０サーバに返送する他の情報は、標的デバイス１００の新しく更新されたキーリストデータ構造６１０のメッセージダイジェストである。これは、新しく更新されたキーリストデータ６１０構造の確認であるとともに、使用許諾権限５１０サーバ上および標的デバイス１００上の特定の標的デバイス１００に関連付けられるキーリストデータ構造６１０の等価性を検証するためにも使用される。このデータ構造の正確な構成を、以下の項に説明する。また、特定のキーまたは１組のキーの所有権の永久譲渡が達成される方法を以降の項で論議する。

【００７５】

この時点で、上記で概説される工程は、開発者５２０から標的デバイス１００にアプリケーション特有のキー５５０を譲渡するためにプロトコルを使用することができる、唯一の方式ではないことに留意されたい。例えば、実際のキー譲渡トランザクションは、標的１００とアプリケーション開発者５２０との間でのみ、直接接続を伴うことができる。しかしながら、その場合、接続は、トランザクションにデバイス特有の暗号化情報を寄与するために、開発者５２０のサーバと使用許諾権限５１０のサーバとの間で確立されなければならない。安全な態様でこのプロトコルを動作させることができる多数の機構があり、上記で議論される例は、これらのうちの１つにすぎない。しかしながら、共通することは、標的１００に譲渡されるキーデータが、その標的デバイス１００のみに使用可能であることを確実にするために、３つ全ての関係者が協働しなければならないことである。

【００７６】

キーの構造は、ハードウェア特有の部分ならびにアプリケーション特有の部分といった

、２つの断片を有するように設定できることに留意されたい。これら２つの断片が完全に分離不可能であることは要件ではない。そうである（分離不可能である）ならば、前述の特性を得る。しかしながら、キー断片を独立して動作可能にする方法がある場合、包括的な１組のコピーを有効にし、実際のコードまたは実際の標的デバイス１００とは無関係となり得る制限を使用することができる。言い換えれば、いずれの開発者５２０も、配布の制限がないが、読み取ることができず、実行することしかできない、アプリケーションまたはメディアストリームを発行することができる。これは、使用許諾権限５１０が、製造業者とは無関係に、全てのデバイス上で作動するセキュリティシステム更新を送信したい場合に、有用となり得る。これの別の例は、そのストリームの著作権の制御を依然として維持しながら、公的に利用可能なメディアストリームを放送することである。同様に、発行者は、誰もが読み取る、および／またはコピーすることができるが、１つの特定の標的デバイス１００または１組のデバイス上のみで実行するアプリケーションを配布することができる。これは、例えば、「この特定の部類のデバイスを更新する」というメッセージを送信するために有用となり得る。別の可能なアプリケーションは、どこにおいても作動し、配布に制限がないアプリケーションを送信することである。これは、特定のアプリケーションに対するソースコード（すなわち、オープンソース）を発行するステップと性質が同様である。分離可能なＨ／Ｗ特有およびＳ／Ｗ特有のキー構造によって有効にされる、セキュリティの異なる部類を表１に図示する。

【００７７】

【表１】

ソフトウェアまたはアプリケーション特有のキーセグメント

ハードウェア特有のキーセグメント	「ロック状態」＊	「非ロック状態」＊＊
「ロック状態」＊	制限された動作および制限された配布	無制限の配布、しかし指定ユニット上のみで実行可能（例えば、特定のユニットを標的にしたコード）
「非ロック状態」＊＊	無制限の配布、しかし実行するのみ、すなわち、コードは「可読」ではない	動作または配布に制限がない（すなわち、公開および開放されている）

＊すなわち、特定のシリアル番号に（または一連の番号に）ロックされている

＊＊すなわち、どこでも動作する

表１．分離可能なハードウェア特有およびアプリケーション特有のキー構造

（キーリストデータ構造の構成）

特定の標的デバイス１００に許諾されているアプリケーションまたはメディア特有のキーのリストを含有する、データ構造６１０は、貴重な物品であり、所有者によってバックアップされることが可能となるべきである。個別キーは、（上記のような）標的の２次秘密キーで暗号化されるため、リストは、キーが許諾されるユニットにとって有益であるのみである。しかしながら、このデータ構造６１０が改ざん、破損、および／または完全な損失の恐れがないことを確認できる必要がある。損失したキーリストデータ構造６１０の場合、前述のように、使用許諾権限５１０から、その特定の標的デバイス１００に対するキーリスト６１０の新しいコピーを要求することによって、データ構造６１０全体を回復することができる。キーリストデータ構造６１０に一時的変更が行われた場合（そのよう

なシナリオの理由をこの後の項で論議する）、プロトコルは、一時的であるような変更を識別するための手段に適応してもよい。最終的に、キーリストデータ構造６１０の真正性、適時性、および有効性を立証するための何らかの一義的な機構を含む。

【００７８】

これらの要件を念頭に置いて、次のページの図６に示されるもののような方式で、これらの品質の全てを示す、安全なキーリストデータ構造６１０を構築することができる。例のごとく、以下に示される例は、そのようなデータ構造に所望の特性の全てを含むことができる唯一の方法ではない。それでもなお、図６に図示された特定のデータ構造は、実際に、プロトコルの基本要件の全てを満たす。

【００７９】

上記の図では、留意すべきいくつかの基本的規則がある。第１は、キーリストデータ構造６１０の最上位暗号化が、標的デバイス１００の一次秘密キーで行われるべきことである。この特定のキーを使用することのいくつかの理由があるが、主要な問題点は、このデータ構造のローカルコピーを修復しなければならない場合に、使用許諾権限５１０が、標的デバイス１００とは無関係に、このデータ構造の暗号化された形態を再生できなければならないことである。このデータ構造を暗号化するために、任意の他のキー（例えば、標的の２次秘密キー等）が使用される場合、標的１００がデータ構造の変更を行う必要がある時（キーがリストに追加される時のように）、リスト全体は、バックアップ目的で使用許諾権限５１０に転送されなければならない。これは、使用許諾権限５１０に返送されなければならないネットワークトラフィックの量を大きく増加させる可能性があり、必ずしもチャネル帯域幅の最も効率的な使用ではない。

【００８０】

また、このキーリストデータ構造６１０は、標準アプリケーションまたはメディアストリーム特有のライセンスキーの記憶に使用されることに加えて、セキュリティシステム関連キーの記憶に使用されることが望ましい。このデータ構造は、使用許諾権限５１０によって再生されることができるため、標的デバイス１００上で作動するセキュリティソフトウェアを更新することが望ましい場合、両方の機能に同じキーリストデータ構造６１０が使用され得るならば、より安全で、より効率的であることの両方となる（標的デバイス１００に対するコード記憶の要件の観点から）。

【００８１】

第２の問題点は、キーリストデータ構造６１０の暗号化されたバージョンが本来のキーリストデータ構造６１０のメッセージダイジェストを含むことである。個別キーのそれぞれが暗号化されるものの、リスト自体の他の部分は、メッセージダイジェストが計算される時点で別々に暗号化されないことに留意されたい。メッセージダイジェスト計算の後、キーリストデータ構造６１０の全体（メッセージダイジェストを含む）は、最上位（またはマスター）キーによって識別される、キー値およびアルゴリズムで暗号化される。これは、悪意のある第３者がリストを改ざんし、新しいメッセージダイジェストを計算し、次いで修正されたリストを真性のリストに代用することを防止するために行われる。キーリストデータ構造６１０が標的ユニット１００のメモリスペースに読み込まれると、任意の他の安全な暗号化されたコードブロックにＭＡＣが使用される方法と同じ方式で、キーリスト自体の真正性および有効性を検証するために、この（解読された）メッセージダイジェストが使用される。個別キー以外の要素の全てが、マスターキーで暗号化されるのみであるという事実は、最上位キー以外の任意のキーにアクセスする必要なく、リストを詳しく検討できる（およびリストを維持できる）ことを意味する。また、暗号解読ブロックを通る単一パスのみで、キーリスト目録を編纂することができる。

【００８２】

関心の第３の原則は、個別アプリケーションコードまたはメディアストリーム特有のキーを、各標的デバイス１００に対する個別キーに適応するように十分大きくできることである。コードまたはメディアストリームが大量生産されたディスクを介して配布される場合、これは、アプリケーション開発者５２０が、個別暗号解読キーとともに、新しいコー

ド特有のＩＤを発行する必要があることを意味する。使用許諾工程に関与する関係者の全ての間で転送されなければならないデータの量を最小化しようとする観点より、これはあまり効率的ではないかもしれないが、損なわれた暗号解読キーを追跡する能力を含む（しかし、それに限定されない）機能性をプロトコルに追加する。これはまた、キー取消を扱う以降の項でも論議する。

【００８３】

注目すべき次の問題点は、キーリストデータ構造６１０のヘッダが、リストの残りの部分を構成するアプリケーション特有のキーと同じ１組の特性を共有することである。実際、ヘッダは、キーリストデータ構造６１０自体の残りの部分に対するマスターキー６２０と考えることができる。したがって、リストの残りの部分の管理を判定するために、このキーをどのように使用できるかに関する限りは、動作の同じ原則を適用することができる。これは、標的デバイス１００のセキュリティシステムの時間依存性管理を含む。したがって、標的ユニット１００は、所定の間隔でそのセキュリティシステムを更新せざるを得なくなり、それだけで極度に強力な概念である。

【００８４】

キーリストが多数のセクションを含有することができ、それぞれが独自のマスターキー６２０（リストヘッダ）を有し、したがって、独自の独立暗号化機構を有するという可能性も存在する。任意の他のキーと同様に、リストヘッダは、キーリストデータ構造６１０を解釈するために使用される、暗号化されたコードブロックを指し示すことができる、コード特有のＩＤフィールドを含有する。次いで、リスト全体は、独自のマスターキー（さらに別のリストヘッダ）を含む、さらに別のマスターリスト内に含有され得る。したがって、キーリストデータ構造６１０全体を再帰的に定義することができる。前述のように、この再帰的特性は、新しいキーリストデータ構造６１０を作成して同じデータ構造の以前のバージョンの欠点に対処することによって、セキュリティシステムを更新するために使用することができる。リスト全体のセキュリティが「最外の」（または最新の）セキュリティ層内に含有されるため、キーリストデータ構造６１０全体のセキュリティは、常にセキュリティソフトウェアの最新の反復に基づく。

【００８５】

したがって、キーリストデータ構造６１０の再帰的特性は、強力な特徴である。前の項で論議された、データ構造の正確な実装が最重要ではないということも理由である。上記で提供した説明は、単に、プロトコルの再帰的性質を機能させるために必要とされる最小限の一部の機能性である特徴を含んだ、一例であった。

【００８６】

それがどのように構造化されるかとは無関係に、キーリスト６１０は、いくつかの一般的な状況下で維持および／または更新されてもよい。これらの状況は、リストに含有されるキーのうちの１つ以上の状態が修正される場合を含む（しかし、それに限定されない）。１つのユニットから別のユニットへ特定のキー２１０の所有権を譲渡することができるいくつかの基本的機構があり、これらを以降の項で論議する。しかしながら、どのような場合でも、改訂されたキーリストが維持される機構は、使用許諾権限５１０の干渉を必要とするもの、および独立して実行することができるものといった、２つの部類に分けることができる。

【００８７】

このプロトコルが基づく主要な動作概念のうちの１つは、使用許諾権限５１０の中央サーバと個別標的ユニット１００との間の必要なネットワークトラフィックの量を最小に削減するというものである。したがって、キーリストデータ構造６１０の任意の一時的変更（その理由は以下に説明する）は、標的ユニット１００によって独立して維持されることができるべきである。このことの主要な理由は、これらの変更が、表向きは、デバイスのセキュリティシステムの永久的変更（常に、標的デバイス１００と使用許諾権限との間の相互作用によってのみ達成されるべきである）よりも頻繁に発生するということである。

【００８８】

どのような場合でも、標的デバイス１００が、一義的な方式でマスターキーリストデータ構造６１０の現在の状態を追跡することができる、何らかの機構が存在しなければならない。このことは、２つの「マスター」リストを有することによって達成することができる。これら２つのリストのうちの第１（永久キーリストと呼ぶ）は、使用許諾権限５１０によって維持される。このリストは、問題の標的ユニット１００に関連付けられる、アプリケーション特有のキーの「永久」所有権に関係している。第２のリストは、等しく重要であるが、「永久」キーリストデータ構造６１０の一時的修正に関係しているものである。これらの修正は、リストへの追加となり得るか、または、リストからの削除となり得るかのいずれかであることに留意されたい。２つのリスト自体のデータ構造の実装には必要な差異はなく、主要な差異は、それらがどのように使用されるかによって発生する。これらのリストのうちの一方または他方のいずれかが、損失したイベントから標的ユニット１００を回復するために、何らかの方法が存在すべきことが望ましい。この損失は、何らかの壊滅的な不具合によるもの、またはリストのうちの１つの中に含有された情報がどういうわけか破損した（悪意がなく、または悪意を持ってのいずれかで）場合によるものとなり得る。そのような「キーリスト破損」イベントの含意を以降の項で論議する。使用許諾権限との接続によって、永久リストを修復できることが必要であるものの、使用許諾権限５１０が特定の標的デバイス１００の一時キーリストを回復できることは必要ではない（望ましくさえない）。この見解には多くの理由があるが、主要な理由は、一時キーリストが、永久キーリストよりもはるかに頻繁に更新される可能性が高いことであり、中央使用許諾権限５１０と標的ユニット１００との間の必要なネットワークトラフィックの量を絶対最小値に保つことが望ましい。それでもなお、いくつかの理由で（そのいくつかを以降で論議する）、使用許諾権限５１０が特定の標的１００の一時キーリストに修正を行うことができることが潜在的に望ましくあり得る。この場合、標的デバイス１００の一次秘密キー（使用許諾権限５１０に知られている）を使用して、このリストを暗号化させることが望ましくなる。

【００８９】

前述のように、キーリストデータ構造６１０の両方の完全性は、リスト自体とともに保存される、署名されたメッセージダイジェスト（デジタル署名）を使用することによって検証することができる。このメッセージダイジェストを生成するために使用される安全なコード機構の実装は、以前の項において説明し、手順を再び繰り返す必要はない。また、損失および／または破損の場合に永久キーリストデータ構造６１０を回復するための手順もすでに説明した。対処しなければならない唯一の残っている問題点は、一時キーリストデータ構造６１０の時間依存性部分をどのように解釈するか、および一時キーリストがどういうわけか使用不能になった場合にどのように対処するかである。

【００９０】

（一時ライセンス譲渡）

これは、タイムスタンプフィールド２３０の使用が重要である、セキュリティプロトコルのセクションのうちの１つである。前述のように、一時キーリストデータ構造６１０は、標的デバイス１００の永久キーリストと正確に同じ方式で構築される。しかしながら、２つの間にはいくつかの差異がある。第１の差異は、標的ユニット１００の秘密キーのうちのいずれか１つによって、一時キーリストを潜在的に暗号化できることである。通常の条件下で、使用許諾権限５１０が、このデータ構造を再構築できることは必要ではないため、それを暗号化するためにキーのうちのどれが使用されるかは、表向きは関連性がない。しかしながら、このリストがまた、ユニット１００の一次秘密キーを使用して暗号化された場合、それは潜在的に使用許諾権限５１０の役に立つであろう。このことの理由は、ライセンス取消と関係があり、その状況を以降の項で論議する。

【００９１】

一時および永久キーリストの第２の（かつ最も重要な）区別は、一時キーリストデータ構造に関連するタイムスタンプ値２３０のコピーもまた、標的デバイス１００の内部に記憶されることである。このレジスタは、ソフトウェア可読ではなく、セキュリティブロッ

クの一部であるため、安全なコードによって上書きされることができるのみである。このレジスタにおける値は、一時キーリストデータ構造がどういうわけか損失および／または破損した場合にどうするかを判定するために使用される。その手順をこの項の以降で論議する。

【００９２】

一時キーリストと永久キーリストとのさらに別の区別は、標的ユニット１００が、その永久リストからユニット１００の一時リストへ特定のキーの所有権を（一時的に）譲渡できることであるが、いずれの（同時に動作している）デバイスも、その一時キーリストから任意の他のキーリストへは特定のキーの所有権を譲渡できない。これは、当然ながら、他のユニットの一時キーリストだけでなく、標的１００独自の永久キーリストも同様に含む。これは、誰が（および、いつ）任意の特定のキーを「借用」することができるようになるかを永久所有者のみが決定できることを意味する。しかしながら、この「貸出」期間は無期限にできる（および、このトランザクションは使用許諾権限に連絡する必要なく実行できる）ことに留意されたい。この「永久貸出」特徴は、最も近代的なデジタル著作権コントロール情報（ＣＣＩ）システムの一部である、標準的「コピーワンス」機能要件と同等である。

【００９３】

「キー所有権」譲渡手順は、図書館から本のコピーを借り出す手順といくぶん同様である。「借り手７２０」が永久所有者（「貸し手７１０」）から特定のアプリケーション特有のキー５５０の一時使用を要求する場合、貸し手７１０はまず、キー借出交渉工程の持続時間にわたって特定のキーの使用を禁止する、自らのための更新された一時キーリストを生成する。このアクションは、２つ以上の借り手７２０ユニットが同じキーを要求することを禁止する。貸し手ユニット７１０の一時キーリスト上の「借り出されたキー」の存在は、任意の特定のキーへのアクセスを制御するために、セマフォとして効果的に使用される。しかしながら、キーが「制限」に設置される初期時間量は、比較的短い期間に限定されるべきである。これは、借り手７２０デバイスが、長い期間にわたって特定のキーのアクセスを要求し、次いで、特定のキーの使用を不正に独占することにより、何らかの理由でトランザクションを完了できない場合を防止するためである。この比較的短い借出交渉段階のタイムアウトもまた、貸し手ユニット７１０に対する「サービス妨害」攻撃の同等物を搭載しようとしているかもしれない、悪意のあるデバイスの対策に役立つ。実際、貸し手ユニット７１０は、選択的に、その「承認された借り手」リスト上にないデバイスからの要求、または、万一これらのユニットのうちのいずれか１つがある期間内に過度に多くの要求を行おうとした場合に要求を無視することができる。この一時ブロックがキーに設置される正確な時間の長さは重要ではないが、任意の所与の借出手順が完了することを可能にするように十分長くなるべきである。高ネットワークトラフィックまたは潜時の時には、この期間を延長することができる。一時「キー借り出し」手順を描写する詳細フロー図を図７に示す。

【００９４】

所与のキーの２つ移譲のコピーが同時に借り出されることが可能となる場合、所与のキーのいくつのコピーがいずれか１つの時点で借り出されるかを示すために、貸し手デバイス７１０の一時キーリスト内の適切なフィールドを使用できることに留意されたい。いったん借り手７２０および貸し手７１０が所与のキーの特定の借出期間を交渉すると、貸し手７１０は、借り手７２０にキー７４０の暗号化されたコピーを送信する。この暗号化は、貸し手デバイス７１０のみに知られている、一時秘密キー７３０を使用して実行される。次いで、借り手７２０が暗号化されたキーの正確な受信を確認する（暗号化されたメッセージから計算されるメッセージダイジェストを用いて）と、貸し手７１０は、借り出されたキーの「貸出期間」を延長し、借り手デバイス７２０に一時秘密キー７３０を送信する。この貸出工程の最大持続時間は、プロトコルの動作にとって重要ではなく、この値の選択において行われなければならないいくつかのトレードオフがある。これらの特定の問題点をこの項の以降で繰り返す。上記の例では、「借り手７２０」および「貸し手７１０

」デバイスは、キー毎に借り出し期間の実際の長さを交渉できると想定するが、これは、確かにプロトコルの要件ではない。

【００９５】

借り手７２０または貸し手７１０のいずれかの一時キーリストが更新される時点の直前に、新しい一時リストに関連付けられるタイムスタンプ値２３０のコピーが、標的１００上に不揮発性様式で記憶される。その時点で、一時キーリストデータ構造の暗号化されたバージョンは、安全にメモリに書き出すことができる（または、オンボードＮＶＲＡＭ、フラッシュＲＯＭ、または何らかのハードディク上のどこか７５０のバックアップファイル等の、何らかの他のより永久的な場所に記憶される）。一時キーリストは、潜在的に、永久キーリストよりもはるかに頻繁に読み取られ、更新されるため、このリストは標的ユニットに迅速にアクセス可能であるべきことが望ましく、よって、アクセス潜時が比較的短い少なくとも１つの場所に、このリストが記憶されることが推奨される（しかし、プロトコルの実際の要件ではない）。一方で、電力不具合が不確定時間量にわたってユニット１００の機能性の損失を潜在的に引き起こし得るため、このリストが記憶される唯一の場所が、何らかの揮発性記憶媒体（ＤＲＡＭ等）であることは推奨されない。この問題点について、この項の以降で詳細に述べる。

【００９６】

特定のキーの借り出し期間が満了すると、借り手７２０および貸し手７１０デバイスの両方は、それぞれの一時キーリストデータベースを独立して更新することができる。したがって、「特定のキーを流通に戻す」ために、借り手７２０が貸し手７１０ユニットと連絡を取っていることは要件ではない。これは、借り手７２０および貸し手７１０デバイスが大きく離れている場合に、主要な利便性要因である。当然ながら、この動作のセキュリティは、キータイムスタンプ記録を生成および制御するために使用されるオンチップクロック間の非常に緊密な相関に依存してもよい。したがって、時間／日付クロックは、セキュリティシステムの不可欠な部分でなければならず、そのようなものとして、中央サーバとのトランザクションによって上書きされることができるべきである。また、クロックは、悪意のあるユーザが内部タイムスタンプ値２３０を修正しようとする場合に改ざん抵抗し、また、通常発生するシステム電力不具合を乗り切ることができるように、十分にロバストとなるように設計されなければならない。このクロックがバッテリ電源式であること、およびバッテリが除去され得るか、または経時的に電池切れになり得ることが想定できないわけではないため、システムは、クロックが潜在的に、使用許諾権限との相互作用によって再開およびリセットされ得るような態様で設計されるべきである。

【００９７】

こうして、特定のアプリケーション特有のキー５５０の所有権を１つのユニットから別のユニットへ一時的に譲渡することができる状況を説明した。「貸出期間」の最後に、「借り手７２０」および「貸し手７１０」ユニットの両方は、それらの一時キーリストデータ構造を更新して、本来の所有者へのキーの「返却」を反映することができる。この手順は、両方のユニット上で独立して実行することができ、したがって、２つのデバイス間の相互作用を必要としないことに留意されたい。

【００９８】

ここで、１つ以上のキーが「借り出されている」または「貸出中」である間に、一時キーリストデータ構造の一方または他方が破損および／または損失した場合に対処しなければならない。「貸し手７１０」ユニット側では、キーが借り出されると、それが最初に行うことは、「貸出」期間の終了を判定することである。この値は明らかに、現在の時間／日付フィールドの値に貸出期間の持続時間を加算することによって構築される。次いで、この時間／日付値は、デバイスの一時キーリストが最後に更新された時の結果としてオンチップに記憶されている値と比較される。新しい値が古い値よりも大きい（後である）場合、新しい値は、古い値の代わりに上書きされる。「借り手７２０」側では、この同じ処理が使用されるため、結果は、任意の所与の標的ユニットにおいて、一時キーリストタイムスタンプは、常に特定のユニット１００の一時キーリストの一部として保存されるタイ

ムスタンプのうちのいずれかの最新のものである。

【００９９】

ユニット１００の一時キーリストが損失あるいは不正に修正された場合、この「最新のタイムスタンプ」値が満了する時点まで、一時キーリストおよび永久リストの両方は無効となる。次いで、その時点で、ユニットは、永久キーリストを使用することに再び着手することができ、新しい一時キーリストを再構築する工程を開始することができる。

【０１００】

したがって、デバイスの一時リストが改ざんまたは削除された場合、ユニットは、タイムアウト期間が満了するまで、効果的に動作不能となる。このタイムアウト手順は、不必要に制限的と思われる場合がある一方で、何らかの悪意のある行為の結果として、または１つのユニットから別のユニットへのキーの譲渡中に発生する何らかの故障（停電またはネットワーク接続の停止等）により存在する、任意の特定のアプリケーション特有のキーの複数のコピーの潜在的問題を回避する。また、一時キーリストデータ構造の改ざんの結果としてのそのようないくつかの影響の可能性は、実践の阻止を助長するはずである。

【０１０１】

この点で、プロトコルの動作を強化するために使用され得る、多数のオプションの付加的特徴がある。１つのそのような考えられるオプションは、標的ユニットのオンチップセキュリティセクションに保存されている値に、暗号化されたキーリストデータ構造６１０のうちのいずれか一方（または両方）から生成される、署名されたメッセージダイジェスト（デジタル署名）を追加することである。デジタル署名の暗号解読に起因するＭＡＣ値は、暗号解読工程全体を経過する必要なく、任意の特定のキーリストのうちのいずれかの有効性を迅速に検証するために使用され得る。しかしながら、複数のネスト化されたキーリストの問題点は、暗号化されていないキーを最終的に産出するために、この暗号解読手順をある時点で複数回行わなければならない可能性が全面的に高いことを意味し、よって、これらのデジタル署名がオンチップに保存されることは、プロトコルの動作にとって重大ではない。

【０１０２】

別の強化の可能性は、１つだけよりもむしろ、１対のオンチップタイムスタンプ２３０値を記憶することである。一時キーリストを更新しなければならない最も早い（次の）時間を示すために、付加的なタイムスタンプ２３０が使用され得る。リストを常にチェックする（暗号解読工程を経過することを伴う）必要がないため、これにより、標的デバイス１００が、その一時キーリストを改訂する必要がある時を決定しやすくなる。この特徴は、非常に有用となるが、再度、ユニットがこのプロトコルを実行できるための基本要件ではない。しかしながら、この第２のタイムスタンプを含有するシステムが実装される場合、２つのタイムスタンプが何らかの理由で「同期しなくなる」場合に混乱の可能性を提起する。思い浮かぶ１つのそのような例は、１つのそのようなタイムスタンプが書き込まれた直後であるが、第２のものが更新される前の時点で発生する、電力故障が存在する場合である。

【０１０３】

対処されるべき最後の問題点は、これらの一時キーリストタイムスタンプの値に対して、何が最小限度および最大限度であるかという問題である。一方で、最大「一時貸出期間」のより大きい限度は、ユーザが、適度に長期間にわたって１つのユニットから別のユニットへ、特定のデータアプリケーション（またはメディアストリーム）の使用を譲渡できるようにし得る。これは、ユーザが「ホームユニット」から携帯用ユニットへメディアストリームの所有権を譲渡することを希望する場合に、潜在的に役立つ。長い「借り出し期間」を有することにより、本来の「貸し手」ユニット７１０と連絡を取っていることを必要とせずに、ユーザが長期の旅行に携帯用ユニット（その関連一時キーとともに）を持っていくことが可能となる。長い「借り出し」期間の不利な面は、万一、元のユニット上の一時キーリストデータ構造に何かが起こった場合、そのユニットが長期間にわたって潜在的に無効となることである。

【０１０４】
この最後の問題点はまた、１つの悪意のあるコードが、オンチップタイムスタンプレジスタの値をある不確定値に設定できる場合に、標的ユニット１００に対する潜在的危険も指摘する。これは、潜在的に、攻撃の標的を無効化することに等しくなり得るため、このタイムスタンプレジスタの値は、「安全な」コードブロックによって書き込まれることができるのみとなるべきである。再度、各ユニットは、異なる１組の秘密キーを有するため、悪意のあるデバイスが正当なユニットを効果的に装う場合を除いて、１つの特定のユニット１００の秘密キーデータ１０４の発見は、他のユニットへの関心を引き起こすべきではない。この攻撃のモードを、同一性確認に関する問題点を扱う以降の項で論議する。

【０１０５】
（永久ライセンス譲渡）
この手順の要素の多くを、本書の以前の項において論議した。特有のキーが１つのユニットから別のユニットへ永久に譲渡される基本工程を図５に示した。多くの点で、この手順は、この項の直前の項で説明されているような、キー所有権の一時譲渡の手順と本質的に同様である。

【０１０６】
２つの手順の間の主要な差異は、永久譲渡が一時譲渡よりも単純な工程であること、永久キー所有権の譲渡手順が使用許諾権限５１０と標的ユニット１００との間の相互作用を利用してもよいことである。永久譲渡工程がより単純である理由は、一時キー借り出し手順において必須条件である借出期間交渉を必要としないという事実にある。永久譲渡機能が使用許諾権限５１０と標的ユニット１００との間の相互作用を利用する理由は、更新されたキーリストデータ構造が、トランザクションの両端で再構築されることができなければならないという事実によるものである。

【０１０７】
永久ライセンス譲渡は、通常、使用許諾権限５１０との相互作用を用いて発生するため、どのアプリケーションまたはメディアストリーム特有のキーがどの標的ユニットに属するかという記録がある。前述のように、これは、標的ユニット１００のキーリストが何らかの壊滅的なデータ損失状況後に修復されなければならない場合に、または特定の標的ユニット１００の所有権が異なる実体に譲渡される場合に、必要である。使用許諾権限５１０側のこの干渉はまた、特定のキーの永久所有権が１つの標的ユニット１００から別のユニットへ譲渡される場合にも必要である。別の実体から最初に購入された資産を所有者が再販売する能力は、「第１販売権」として知られており、本明細書で説明されているプロトコルがこの特定の機能を支援する能力は重要である。

【０１０８】
標的ユニット１００の永久キーリストが使用許諾権限５１０によって維持されるという事実の別の重要な側面は、ユニット１００がどういうわけか損なわれたことが証明された場合に、またはキーが損なわれているとして識別された場合に、この本体が、個別標的ユニット１００のライセンスキーのいずれかまたは全てを取り消す能力を有することである。（前述のように）あらゆる標的ユニット１００にキーの一意のリストを付与する可能性が存在するため、使用許諾権限５１０が損なわれたキーのソースを追跡する機会も提供し得る。そのような状況では、このプロトコルは、通常は「透かし」特徴に関連付けられる機能を果たし得るが、従来の透かし処理の欠点（透かしがメディアストリームの質に悪影響を及ぼす可能性等）がない。

【０１０９】
たとえ事実と思えないかもしれなくても、アプリケーションコードまたはメディアストリーム特有のＩＤ情報がアプリケーション開発者５２０から発生し、使用許諾権限５１０が、任意の特定のアプリケーションまたはメディアストリームとその許諾所有者との間で関連付けを行うことができるように、十分な情報を必ずしも有するわけではないため、デジタルコンテンツ所有者のプライバシーは、この工程によって依然として維持される。ユーザのプライバシーを保護する能力もまた、このプロトコルの重要な側面である。

【０１１０】
永久キー譲渡工程について留意すべき最後の問題点は、永久キー譲渡が果たす同じ機能の全てを、一時キーライセンス譲渡によって達成することが実際に可能なことである。しかしながら、標的ユニット１００のセキュリティシステムの維持は、中央の安全なサーバによって理想的に制御されるべき機能であるため、連鎖のどこかにそのような機構を設置することが必要である。また、ユーザがプライバシーを維持することを懸念する場合、中央サーバが著作権所有者と標的ユニット１００との間の緩衝の役割を果たすことができるという事実は、大変有用である。最後に、使用許諾権限５１０は、一時キー譲渡機構からこの機能を除外する、特定の標的ユニット１００の永久キーリストに対する中央バックアップ記憶機構の役割を果たすことができるという魅力もある。

【０１１１】
（システム所有権譲渡、ライセンス取消、およびセキュリティシステム更新）
標的ユニット１００のライセンス（またはキー）のうちの１つ以上が取り消されてもよい、いくつかの異なる手段がある。最も単純な方法は、標的１００の一次秘密キーを単純に更新する方法である。この点で、標的１００は、その永久キーリストにアクセスできなくなり、したがって、新しいものを作成する工程を開始しなければならない。一次秘密キーが一時キーリストデータ構造に対する暗号化工程で使用されなかった場合、たとえそうでなければ永久キーリストがアクセス不可能となる場合があるとしても、この一時キーリストは、潜在的に依然としてアクセスされ得ることに留意されたい。この点は、一時キーリストに対する暗号化工程の説明で以前に述べた。この理由により、永久および一時キーリストデータ構造６１０の両方に対する暗号化キーとして、標的ユニット１００の一次秘密キーを使用することがおそらく最善の考えである。

【０１１２】
標的ユニット１００の所有権がある個人から別の個人へ変わる場合、この所有権変更を達成するための最も単純な方式は、ユニット１００の一次秘密キーを何らかの新しい値に設定することである。しかしながら、本来の所有者が標的から永久キーの全てを回復する機会を有する前に、これが発生した場合には、ライセンスを失う。本来の所有者が、標的ユニットとともに関連永久キーリストの所有権を譲渡することを希望する場合には、特定のデバイスに関連付けられる所有権情報（使用許諾権限において記憶される）を変更すること以外に、何も標的ユニット１００になされる必要はない。

【０１１３】
ライセンス取消が発生することができる別の方式は、特定の標的ユニット１００の永久キーリストに対するマスターキーが「満了する」場合である。ユニット１００のセキュリティシステムの更新が永久キーリストの一部として保存されるため、この状況は、潜在的に破滅的な影響を及ぼし得る。

【０１１４】
この窮地から回復することは潜在的に可能となるが、標的１００が、徹底的に構築される全く新しい「信用の連鎖」を必要とすることを要求する。この状況では、新しく初期化されたセキュリティシステムのコアは、標的１００のある部分上でアトミックに作動することができるとして検証され得る計算のみに基づかなければならない。したがって、これは、（潜在的に疑わしくなり得る）最小量の他の汎用コードさえも要求した、任意のハッシング関数の使用を排除する。幸運にも、この状況は、検証可能な形で安全なコード断片の永久コアを永久キーリストデータ構造６１０の一部として常に保つという単純な事柄によって回避することができる。

【０１１５】
ライセンス取消のさらに別の方式は、使用許諾権限５１０が標的ユニット１００の永久または一時キーリストにおける特定のキー入力を無視することを選択した場合に、発生することができる。これは、セキュリティシステムアップグレードが必要とされる場合に、または特定の標的ユニット１００が特定のアプリケーションまたはメディアストリームの無許諾コピーを有していると識別された場合に、使用され得る。標的ユニット１００は通

常、独自のキーリストデータ構造６１０を維持するため、この手順は、使用許諾権限５１０と標的ユニットとの間で、通常よりも大量のネットワークトラフィックを伴う。

【０１１６】

それでもなお、そのような手順は、破壊されたキーを検索して無効にする、および／または古いソフトウェアを更新されたバージョンと交換するように設計される、標的特有のカスタムバージョンで、問題の標的デバイス１００にセキュリティシステムソフトウェアを強制的に改訂させることによって、達成することができる。当然ながら、この手順は、標的デバイス１００が使用許諾権限５１０の中央サーバとの接続を開始する時点でしか発動させることができない。通常の状況下で、任意の特定の標的ユニット１００が、任意の特定のスケジュール通りに使用許諾権限５１０への連絡を開始することは保証できない。幸運にも、問題の標的デバイス１００は、その永久キーリストへの任意の新しい追加を認可するために、使用許諾権限５１０に接続しなければならない（直接または間接的のいずれかで）ため、あらゆるキー取消アクションは、新しいキー使用許諾手順の一部として達成することができる。また、この「リスト監視」アクションを支援するために、前述の「セキュリティシステムタイムアウト」機構が使用され得ることも可能である。しかしながら、これが事実であることは、このプロトコルにとっての要件ではなく、そのようなシステムがユーザのプライバシー権利の侵害をもたらす可能性が高い。

【０１１７】

（他の懸念事項：）

必ずしもプロトコル自体の一部ではないが、それでもなお、本明細書で説明されるプロトコルを適正に実行することができる特定のシステムを作成する工程で対処されなければならない、多数の問題点が存在する。これらの問題点のうちのいくつかは、実際の工程またはデバイスの実装に依存しており、他のものは、大部分はアプリケーションに特有である。この情報は、好適な標的デバイス１００の適正な構成に密接に結び付いているため、これらの問題のいくつかを以下の項で論議する。

【０１１８】

（相互動作することができるユニットの数の限度）

著作権所有者が、主要標的が一時的に所有権を譲渡することができるデバイスの合計数を限定することを希望する場合、これは、どの時点においても有効であってもよい限定数の公開／プライベートキーペアを確立することによって達成されてもよい。これは、以前の項で説明された、同じアプリケーション特有のキーの複数のコピーが同時に「貸出中」であった場合とは異なることに留意されたい。他のシナリオが可能であり、その場合、特定の標的デバイス１００からアプリケーション特有のキーのうちのいずれかを「借り出す」ことができる、デバイスのリストは、ある１組のシリアル番号に限定することができる。使用許諾権限５１０は、標的ユニット１００のセキュリティシステムが管理される正確に同じ方式で、そのような「承認された借り手」リストを管理することができる。したがって、使用許諾権限５１０は、例えば、「承認された借り手」リスト上の１組のシリアル番号を、元の標的デバイス１００と同じ所有権情報を有する者に限定し得る。

【０１１９】

（秘密キー発見および同一性確認の問題点）

特定のプレーヤに対する一次秘密キーが、物理的分解およびチップ金型検査によって発見された場合、これは、各デバイスが異なる１組の秘密キーを有するため、任意の他のデバイスのセキュリティを損なうべきではない。しかしながら、特定のプレーヤに対する一次キーがどういうわけか損なわれた場合、無許諾デバイスが正当な標的ユニットを装うことが潜在的に可能である。この問題が未検出のままとなった場合、この知識を装備した無許諾デバイスが、その特定の標的ユニットへ発行されたアプリケーション特有の暗号解読キーのうちのいずれかを損ない得るという可能性存在する。標的ユニット１００のシリアル番号は、まず第１に、使用許諾権限５１０がデバイスへ暗号解読キーを発行するために登録されなければならないため、この目的の問題は、表向きは、無許諾デバイスによる他の正当な標的ユニット１００の限定に限定される。

【０１２０】
しかしながら、ユニット１００の秘密キー１０４の両方がそのような工程によって発見された場合、暗号化されたキーリストダイジェストの以前にバックアップされたコピーの検査に基づいて、そのユニットに許諾されたアプリケーション特有のキーの全てのセキュリティを損なうことが可能となる可能性がある。この理由により、一次および２次秘密キーの両方は、これらのキーの値を発見しようとする試行がキーデータの損失をもたらすように、「改ざん防止」方式で標的チップ上に実装されるべきである。
【０１２１】
この改ざん防止特徴を標的デバイス１００上に実装することができる、多数の手段があるが、そのようなものの正確な実装は、本書で説明されるプロトコルにとって重要ではない。「秘密キー損失」状況が、ユーザ側の怠慢（または乱用）という何らかの悪意のない行為を通して発生した場合、正当なユーザは、損傷されたユニット１００のアプリケーション特有のキーを新しいデバイスに譲渡させるように手配することができる使用許諾権限５１０に、損傷されたユニットを戻すことができるべきである。しかしながら、元の標的デバイス１００が機能しない場合、新しい標的デバイス１００へのキーの譲渡は、アプリケーション開発者５２０とのトランザクションを伴わなければならない（少なくとも、まず第１に暗号化されていない状態で使用許諾権限５１０に供給されなかったキーに対して）ことに留意されたい。
【０１２２】
しかしながら、他の真性標的ユニット１００に扮することができたデバイスは、表向きは、疑いを持たない合法的に許諾されたデバイスをだまして、そのアプリケーション特有のキーのうちの１つ以上の所有権を一時的に放棄させるか、または（前述のように）動作を一時停止させることが可能となり得たことに留意されたい。後者が発生した場合、それからキーを借用しようとしたユニットの全てを無効にし得る「不正ユニット」を有する可能性が存在する。前者が発生した場合、任意の数のアプリケーションまたはメディア特有のキーが潜在的に損なわれ得る。
【０１２３】
したがって、特定の標的ユニット１００に対する潜在的な「許諾された借り手」の数を、使用許諾権限５１０サーバからの安全な更新を用いて、正当なユニットに供給されることしかできないリストに限定するという、以前に議論された概念は、良好なものである。前者の場合、これは、他の疑いを持たないユニットの所有者が、そのユニットがそもそもハッカーに実際に属しない限り、機能ユニットを分解してその秘密キーへのアクセスを獲得するハッカーによって無効化された正当なデバイスを有することを防止する。後者の場合、これは、アプリケーションまたはメディア特有のキーの譲渡を、使用許諾権限により適正に登録された、ある時点で許諾されたデバイスであったデバイスのみに限定する。それでもなお、断固としたハッカーは、依然として正当なユニットを購入し、それを暴露し、何とかして秘密キーデータへのアクセスを獲得し、次いで、正当なデバイスを装うためにこの情報を使用し得る。
【０１２４】
そのため、この種の偽装イベントをどのように検出しようとするかという問題点が残る。この性質の極度に資金力のある敵を打倒するための唯一の成功する戦略は、少なくとも費用トレードオフの観点から、潜在的利得が必要とされる努力に値しないように、システムを設計することである。
【０１２５】
通信している他の未知のデバイスの真正性を証明しようとする、いくつかの手段が存在する。しかしながら、デバイスが実際に主張するものであることを証明するための最も成功する方法は、このデバイスを他のデバイスから独特にする特性に焦点を当てることである。本明細書によって説明されるもの等の、デジタル暗号解読機構等の特殊目的装置の場合、セキュリティプロトコルを適正に実行すること、および所与の１組の入力変数に基づいて正しい結果を計算することは、デバイスの能力となる。しかしながら、本明細書で説

明されるセキュリティプロトコルは、公知のアルゴリズムに基づくため、計算を完了するまで十分な時間があることを考慮すれば、これは、表向きは、任意の汎用計算デバイスによって達成され得る。実際、デバイスを独特にする秘密キー情報がどういうわけか損なわれた場合に、この問題点は、公的に利用可能な技術に基づく任意のデバイスにとって潜在的な問題となる。したがって、分解およびチップ金型点検に直面しても、正当な標的デバイスの全てに対してオンチップに記憶される秘密キー情報が秘密のままでなければならないという教訓に、最終的には依存しなければならない。

【０１２６】

ある時間量内で検証可能なＭＡＣ値を正しく見つけ出す能力等の要件を、標的識別および検証工程に確かに追加することができる。最終的なＭＡＣ値が複数回に暗号化されることを要求することによって、手順をさらに困難にすることができる。したがって、ライセンスの正当なコピーを自ら単純に購入する費用よりも通常はるかに高価となる、（より一般的な）計算リソースへのアクセスを有することを必要とされる点で、攻撃者が正当なデバイスを模倣する能力を潜在的に制限することができる。メディアストリームプレーヤの場合、プレーヤが表向きは適応するように設計されている、メディアストリームのうちの１つ以上の一部分を正しく復号する能力も含むことができる。

【０１２７】

しかしながら、どのような場合でも、デジタル著作権保護の全工程は、チューリング問題である。したがって、十分な時間およびリソースを考慮すると、いずれのデジタル著作権保護方式も、断固とした敵に打倒される可能性がある。これは、当然ながら、秘密キー情報へのアクセスが間違いなく攻撃未遂者にとって大きな利点となるという事実とは無関係でさえある。したがって、ユニットの秘密キーが損なわれることを防ぐ能力は、このセキュリティプロトコルの重要な部分である。

【０１２８】

（結論：）

上記の著作権保護プロトコルは、いくつかの点で独特である。第１は、ユーザが合法的に購入されたアプリケーションまたはメディア特有のキーデータのバックアップコピーを作製する能力を有することを禁止しようとしないという事実である。第２に、このプロトコルは、いずれの種類のデジタルデータの区別も行わず、したがって、セキュリティプロトコルが、保護するように設計されているデータストリームと同じように容易に更新されることを可能にする。第３に、このプロトコルは、ユーザが、それらのアプリケーションまたはメディア特有のキーの所有権を、プロトコルを実行することが可能な別のユニットに一時的に譲渡できるようにする。また、このプロトコルは、ライセンス取得者が１つの標的ユニット１００から別のユニットへの所有権の永久譲渡を達成する能力を提供する。この最後の特性は、このプロトコルの下で、消費者の合法的な「第１販売権」の実現を可能にする。

【０１２９】

実際、本明細書で説明されるプロトコルと他のコピー保護方式との間の基本的な差異のうちの１つは、このシステムのセキュリティが、特定のデータセットにアクセスする能力を制御することに依存しないが、むしろ、そのデータセット内に含有される着想を表現する行為を制御する能力に依存することである。

【０１３０】

先述の詳細では、具体的な実施形態を参照して本発明を説明した。しかしながら、当業者であれば、以下の請求項で説明されるような本発明の範囲を逸脱することなく、種々の修正および変更を行うことができると理解する。それに応じて、明細書および図は、制限的よりもむしろ例証的な意味で見なされるものであり、全てのそのような修正は、本発明の範囲内に含まれることを目的とする。

【０１３１】

具体的実施形態に関して、便益、他の利点、および問題の解決法を上記で説明した。しかしながら、便益、利点、問題の解決法、および、任意の便益、利点、または解決法を発

生させる、またはより顕著にさせる任意の構成要素は、任意または全ての請求項の重大、必要、または不可欠な特徴として解釈されないものである。

【図１】

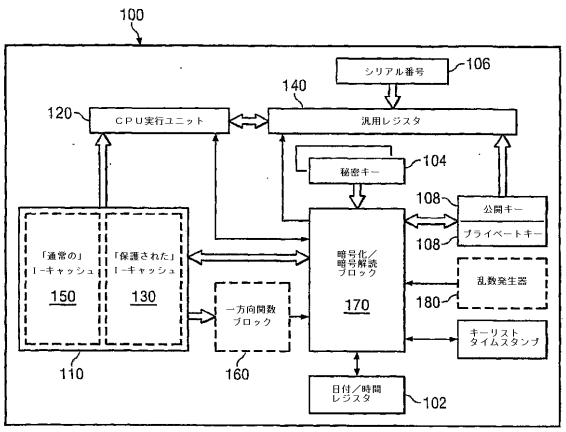


FIG. 1

【図２】

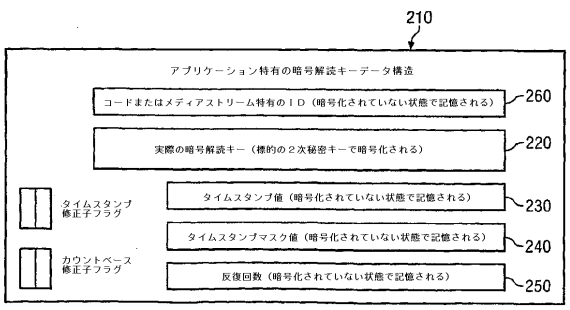


FIG. 2

【図３】

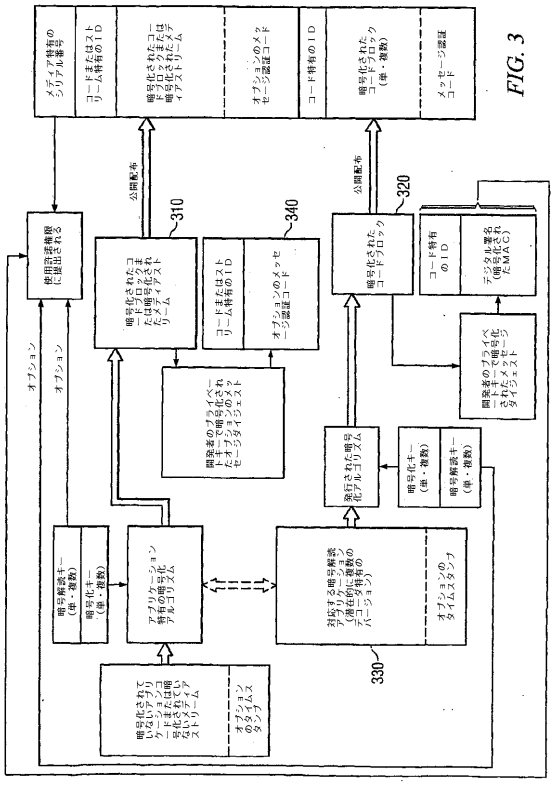


FIG. 3

【図４】

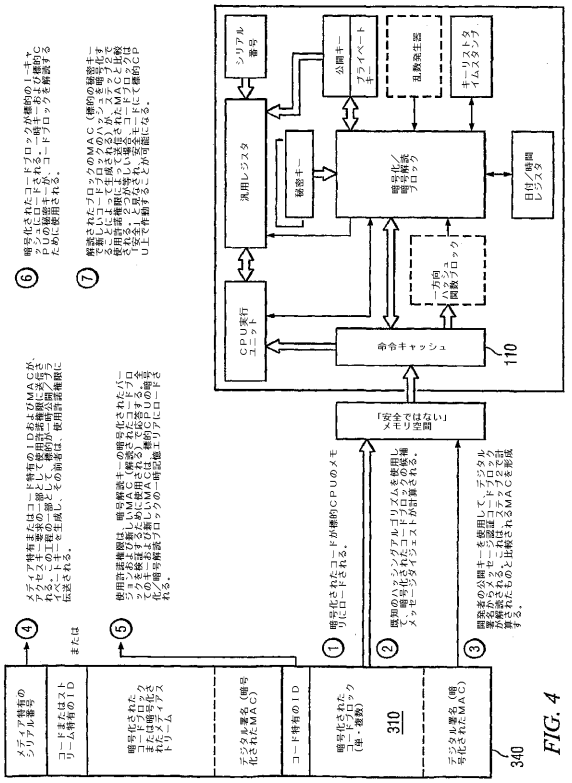


FIG. 4

【図５】

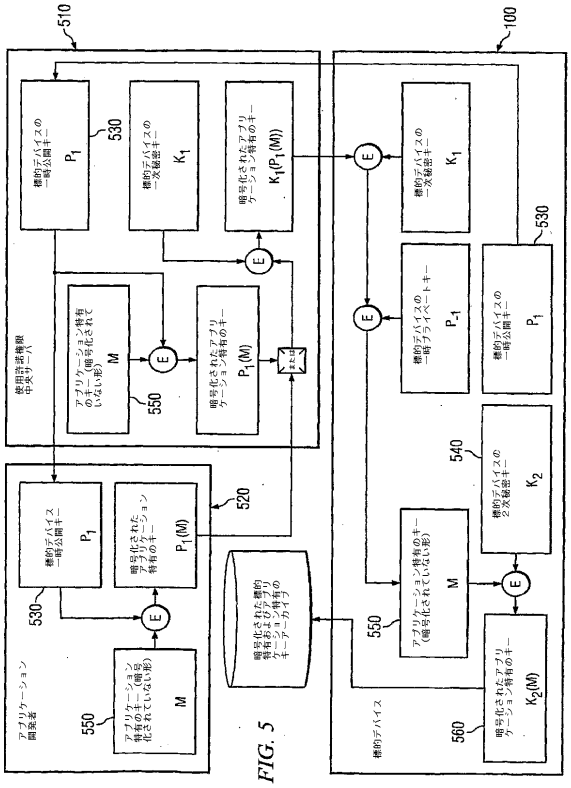


FIG. 5

【図６】

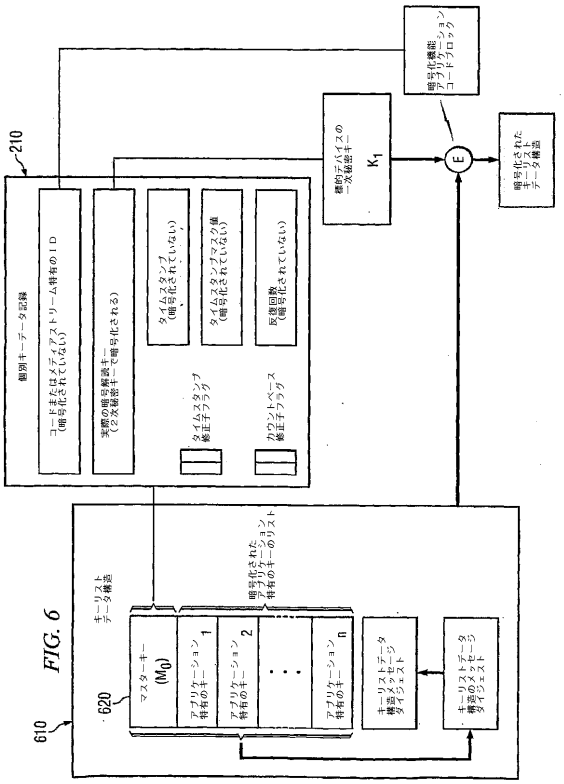


FIG. 6

【図７Ａ】

FIG. 7

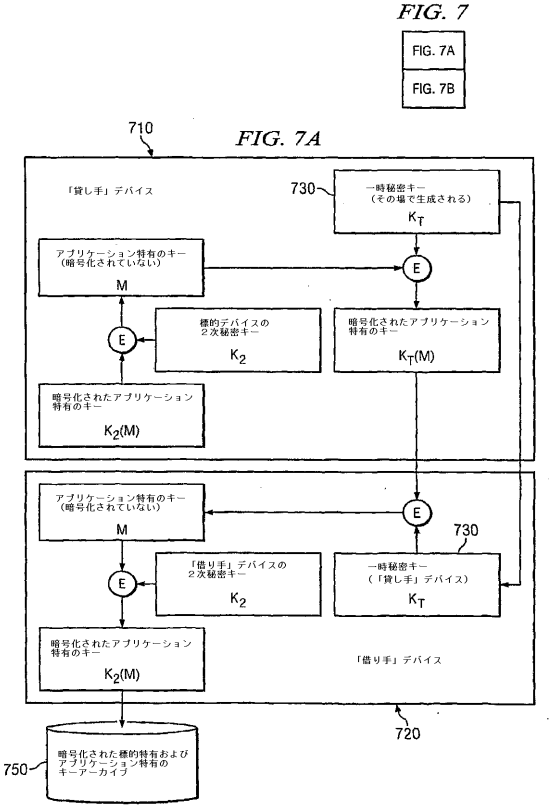
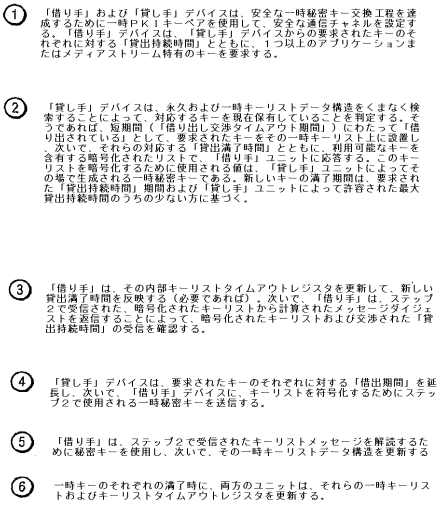


FIG. 7A

【図７Ｂ】

FIG. 7B


①
「借り手」および「貸し手」デバイスは、安全な一時秘密キー交換工程を達成するために一時ＰＫＩキーペアを使用して、安全な通信チャネルを設定する。「借り手」デバイスは、「貸し手」デバイスからの要求されたキーのそれぞれに対する「貸出持続時間」とともに、１つ以上のアプリケーションまたはメディアストリーム特有のキーを要求する。
②
「貸し手」デバイスは、永久および一時キーリストデータ構造をくまなく検索することによって、対応するキーを現在保有していることを判定する。そうであれば、短期間（「借り出し交渉タイムアウト期間」）にわたって「借り出されている」として、要求されたキーをその一時キーリスト上に設置し、次いで、それらの対応する「貸出満了時間」とともに、利用可能なキーを含有する暗号化されたリストで、「借り手」ユニットに応答する。このキーリストを暗号化するために使用される値は、「貸し手」ユニットによってその場で生成される一時秘密キーである。新しいキーの満了期間は、要求された「貸出持続時間」期間および「貸し手」ユニットによって許容された最大貸出持続時間のうちの少ない方に基づく。
③
「借り手」は、その内部キーリストタイムアウトレジスタを更新して、新しい貸出満了時間を反映する（必要であれば）。次いで、「借り手」は、ステップ２で受信された、暗号化されたキーリストから計算されたメッセージダイジェストを返信することによって、暗号化されたキーリストおよび交渉された「貸出持続時間」の受信を確認する。
④
「貸し手」デバイスは、要求されたキーのそれぞれに対する「借出期間」を延長し、次いで、「借り手」デバイスに、キーリストを符号化するためにステップ２で使用される一時秘密キーを送信する。
⑤
「借り手」は、ステップ２で受信されたキーリストメッセージを解読するために秘密キーを使用し、次いで、その一時キーリストデータ構造を更新する
⑥
一時キーのそれぞれの満了時に、両方のユニットは、それらの一時キーリストおよびキーリストタイムアウトレジスタを更新する。

【外国語明細書】
2013084294000001.pdf