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

特許第5192556号
(P5192556)

(45)発行日　平成25年5月8日(2013.5.8)　(24)登録日　平成25年2月8日(2013.2.8)

(51)Int.Cl.		F I		
G06F 21/64	(2013.01)	G06F	21/24	167
H04L 9/32	(2006.01)	H04L	9/00	675A
H04N 7/167	(2011.01)	H04L	9/00	675B
G11B 20/10	(2006.01)	H04N	7/167	Z
		G11B	20/10	H

請求項の数 17　外国語出願　(全 98 頁)

(21)出願番号　特願2011-10472(P2011-10472)
(22)出願日　平成23年1月21日(2011.1.21)
(62)分割の表示　特願2006-518845(P2006-518845)の分割
原出願日　平成16年7月7日(2004.7.7)
(65)公開番号　特開2011-86313(P2011-86313A)
(43)公開日　平成23年4月28日(2011.4.28)
審査請求日　平成23年2月21日(2011.2.21)
(31)優先権主張番号　10/614,765
(32)優先日　平成15年7月7日(2003.7.7)
(33)優先権主張国　米国(US)
(31)優先権主張番号　60/537,421
(32)優先日　平成16年1月16日(2004.1.16)
(33)優先権主張国　米国(US)

前置審査

(73)特許権者　597095197
ロヴィ・ソリューションズ・コーポレーション
アメリカ合衆国　カリフォルニア州　９５０５０　サンタクララ　デ・ラ・クルーズ・ブールバード　２８３０
(74)代理人　100079108
弁理士　稲葉　良幸
(74)代理人　100109346
弁理士　大貫　敏史
(72)発明者　ポール　シー．コッヘル
アメリカ合衆国　９４１１７　カリフォルニア州　サンフランシスコ　ピアス　ストリート　４８

最終頁に続く

(54)【発明の名称】海賊行為を規制し、インタラクティブコンテンツを使用可能にするための再プログラマブルなセキュリティ

(57)【特許請求の範囲】
【請求項１】
デジタルコンテンツと、前記デジタルコンテンツに固有でかつ前記デジタルコンテンツの少なくとも一部分の再生に影響を与えるように再生デバイスのコンピュータ言語インタープリタによって実行可能であるデータ処理命令とを含むデータを受信することと、
前記再生デバイスの前記コンピュータ言語インタープリタを用いて前記データ処理命令を実行することと、
を備え、
前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、
失効ステータスのデジタル署名にアクセスし、前記失効ステータスの前記デジタル署名に基づいて前記失効ステータスが真正であることを判断し、
前記再生デバイスの、前記デジタルコンテンツに対応しかつ前記デジタルコンテンツに関連する認証が失効したことを示す前記失効ステータスを格納するメモリにアクセスし、ここで、前記失効ステータスを格納することは、前記失効ステータスが真正であるとの判断に応答したものであり、
前記失効ステータスに基づいて、前記再生デバイスによる、前記デジタルコンテンツに関連し、かつ、前記デジタルコンテンツの低下した解像度、すなわち、前記受信したデータ内に含まれる前記デジタルコンテンツの最大解像度より低い解像度での前記デジタルコンテンツの前記少なくとも一部分の前記再生を可能にすることを含むオペレーションの実行を開始する

ように、前記コンピュータ言語インタープリタを構成する、方法。
【請求項２】
前記オペレーションは、
前記再生デバイスのユーザに前記デジタルコンテンツが違法であることを通知することと、
前記ユーザが前記デジタルコンテンツに関する購入トランザクションを開始することを可能にすることと、
前記デジタルコンテンツの更なる部分の再生を防止することと、
のうちから少なくとも１つを含む、請求項１に記載の方法。
【請求項３】
前記デジタルコンテンツに固有の前記データ処理命令は、第１のデータ処理命令であり、
前記再生デバイスの前記メモリは、前記デジタルコンテンツに関連する前記オペレーションを実行するように前記再生デバイスの前記コンピュータ言語インタープリタによって実行可能な第２のデータ処理命令を格納し、前記第２のデータ処理命令は前記失効ステータスに対応しかつ、実行されると、前記オペレーションを実行するように前記再生デバイスを構成する、請求項１に記載の方法。
【請求項４】
前記データは、前記デジタルコンテンツに対応する識別子を含み、
前記データ処理命令は、前記オペレーションの前記実行を開始するように前記コンピュータ言語インタープリタを構成する際に、前記識別子の認証性に基づいて前記実行を開始するように前記コンピュータ言語インタープリタを構成し、
前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、前記識別子の前記認証性を判断するように、前記コンピュータ言語インタープリタを構成する、請求項１に記載の方法。
【請求項５】
前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、前記識別子のデジタル署名にアクセスするように前記コンピュータ言語インタープリタを構成し、
前記データ処理命令は、前記識別子の前記認証性を判断するように前記コンピュータ言語インタープリタを構成する際に、前記識別子の前記デジタル署名に基づいて前記認証性を判断するように前記コンピュータ言語インタープリタを構成する、請求項４に記載の方法。
【請求項６】
前記再生デバイスの前記メモリは不揮発性メモリである、請求項１に記載の方法。
【請求項７】
前記デジタルコンテンツの前記失効ステータスを受信することと、
前記失効ステータスを前記再生デバイスの前記メモリ内に格納することと、
をさらに備える、請求項１に記載の方法。
【請求項８】
前記データを受信することは、メディアドライブからまたはネットワークを介して前記データの少なくとも一部を受信することを含む、請求項１に記載の方法。
【請求項９】
デジタルコンテンツと、
前記デジタルコンテンツに固有でかつ前記デジタルコンテンツの少なくとも一部分の再生に影響を与えるように再生デバイスのコンピュータ言語インタープリタによって実行可能であるデータ処理命令と、
を備え、
前記データ処理命令は、前記再生デバイスの前記コンピュータ言語インタープリタによって実行されると、

失効ステータスのデジタル署名にアクセスし、前記失効ステータスの前記デジタル署名に基づいて前記失効ステータスが真正であることを判断し、

前記再生デバイスの、前記デジタルコンテンツに対応しかつ前記デジタルコンテンツに関連する認証が失効したことを示す前記失効ステータスを格納するメモリにアクセスし、ここで、前記失効ステータスを格納することは、前記失効ステータスが真正であるとの判断に応答したものであり、

前記失効ステータスに基づいて、前記再生デバイスによる、前記デジタルコンテンツに関連しかつ前記デジタルコンテンツの低下した解像度、すなわち、前記デジタルコンテンツの最大解像度より低い解像度での前記デジタルコンテンツの前記少なくとも一部分の前記再生を可能にすることを含むオペレーションの実行を開始するように、前記コンピュータ言語インタープリタを構成する、非一時的マシン可読媒体。

【請求項１０】

前記オペレーションは、

前記再生デバイスのユーザに前記デジタルコンテンツが違法であることを通知することと、

前記ユーザが前記デジタルコンテンツに関する購入トランザクションを開始することを可能にすることと、

前記デジタルコンテンツの更なる部分の再生を防止することと、

のうちから少なくとも１つを含む、請求項９に記載の非一時的マシン可読媒体。

【請求項１１】

前記デジタルコンテンツに固有の前記データ処理命令は、第１のデータ処理命令であり、前記再生デバイスの前記メモリにアクセスするように前記コンピュータ言語インタープリタを構成する際に、前記メモリ内に格納されかつ前記デジタルコンテンツに関連する前記オペレーションを実行するように前記再生デバイスの前記コンピュータ言語インタープリタによって実行可能な第２のデータ処理命令にアクセスするように前記コンピュータ言語インタープリタを構成し、前記第２のデータ処理命令は前記失効ステータスに対応しかつ、実行されると、前記オペレーションを実行するように前記再生デバイスを構成する、請求項９に記載の非一時的マシン可読媒体。

【請求項１２】

前記デジタルコンテンツに対応する識別子をさらに備え、

前記データ処理命令は、前記オペレーションの前記実行を開始するように前記コンピュータ言語インタープリタを構成する際に、前記識別子の認証性に基づいて前記実行を開始するように前記コンピュータ言語インタープリタを構成し、

前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、前記識別子の前記認証性を判断するように前記コンピュータ言語インタープリタを構成する、請求項９に記載の非一時的マシン可読媒体。

【請求項１３】

前記識別子のデジタル署名をさらに備え、

前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、前記識別子の前記デジタル署名にアクセスするように前記コンピュータ言語インタープリタを構成し、

前記データ処理命令は、前記識別子の前記認証性を判断するように前記コンピュータ言語インタープリタを構成する際に、前記識別子の前記デジタル署名に基づいて前記認証性を判断するように前記コンピュータ言語インタープリタを構成する、請求項１２に記載の非一時的マシン可読媒体。

【請求項１４】

前記識別子は、前記非一時的マシン可読媒体のシリアルナンバである、請求項１２に記載の非一時的マシン可読媒体。

【請求項１５】

前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、

ネットワークを介して前記失効ステータスのアップデートを受信するように前記コンピュータ言語インタープリタを構成し、
前記データ処理命令は、前記オペレーションの前記実行を開始するように前記コンピュータ言語インタープリタを構成する際に、前記失効ステータスの前記アップデートに基づいて前記実行を開始するように前記コンピュータ言語インタープリタを構成する、請求項９に記載の非一時的マシン可読媒体。
【請求項１６】
メモリと、
前記メモリに通信可能に結合されたコンピュータ言語インタープリタと、
前記コンピュータ言語インタープリタに通信可能に結合されたメディアインターフェイスと、
を備え、
前記メディアインターフェイスは、デジタルコンテンツとデータ処理命令とを含むデータを受信するように構成され、前記データ処理命令は前記デジタルコンテンツに固有でかつ前記デジタルコンテンツの少なくとも一部分の再生に影響を与えるように前記コンピュータ言語インタープリタによって実行可能であり、前記データ処理命令は、前記コンピュータ言語インタープリタによって実行されると、
失効ステータスのデジタル署名にアクセスし、前記失効ステータスの前記デジタル署名に基づいて前記失効ステータスが真正であることを判断し、
前記再生デバイスの、前記デジタルコンテンツに対応しかつ前記デジタルコンテンツに関連する認証が失効したか否かを示す前記失効ステータスを格納するメモリにアクセスし、ここで、前記失効ステータスを格納することは、前記失効ステータスが真正であるとの判断に応答したものであり、
前記失効ステータスに基づいて、前記再生デバイスによる、前記デジタルコンテンツに関連しかつ前記デジタルコンテンツの低下した解像度、すなわち、前記受信したデータ内に含まれる前記デジタルコンテンツの最大解像度より低い解像度での前記デジタルコンテンツの前記少なくとも一部分の前記再生を可能にすることを含むオペレーションの実行を開始するように、前記コンピュータ言語インタープリタを構成する、装置。
【請求項１７】
不揮発性メモリと、
デジタルコンテンツとデータ処理命令とを含むデータを受信する手段であって、前記データ処理命令は前記デジタルコンテンツに固有でかつ前記デジタルコンテンツの少なくとも一部分の再生に影響を与えるように実行可能である、手段と、
前記データ処理命令を実行する手段と、
を備え、
前記データ処理命令は、実行されると、
失効ステータスのデジタル署名へのアクセス、及び、前記失効ステータスの前記デジタル署名に基づいて前記失効ステータスが真正であることの判断と、
前記デジタルコンテンツに対応しかつ前記デジタルコンテンツに関連する認証が失効したことを示す前記失効ステータスを格納する前記不揮発性メモリのアクセス、ここで、前記失効ステータスを格納することは、前記失効ステータスが真正であるとの判断に応答したものであること、と、
前記失効ステータスに基づいた前記再生デバイスによる、前記デジタルコンテンツに関連しかつ前記デジタルコンテンツの低下した解像度、すなわち、前記受信したデータ内に含まれる前記デジタルコンテンツの最大解像度より低い解像度での前記デジタルコンテンツの前記少なくとも一部分の前記再生を可能にすることを含むオペレーションの開始と、
をもたらす、システム。
【発明の詳細な説明】
【技術分野】

【０００１】
本願は一般に、デジタルコンテンツおよび他のデータの配布を、海賊行為、および、他の無許可の使用または再配布から保護することに関する。
【背景技術】
【０００２】
幅広い種類のシステムが、デジタルコンテンツを保護するために提案されている。大部分のこのようなスキームは、コンテンツがメディア上に格納される間、または、コンテンツが信頼できない通信チャネルを介して送信される間、コンテンツを暗号化して、無許可の使用およびコピーからコンテンツを保護する。復号化アルゴリズムおよびキーは次いで、耐タンパ性を有する（tamper-resistant）信頼できるソフトウェアまたはハードウェアモジュールによって管理され、これらのモジュールは、コンテンツをどのように使用することができるかを指定するアクセス制御ルール（固定であっても構成可能であってもよい）を実施するように設計される。
【０００３】
コンテンツ保護スキームは一般に、特定の再生環境に合わせてカスタマイズされる。例えば、パーソナルコンピュータに合わせて設計されたソフトウェア専用ストリーミングコンテンツプレイヤにおけるアンチパイラシ（anti-piracy）システムは、耐タンパ性を有するハードウェアのセキュリティの利点を欠いているが、一般に大きな困難なしにアップグレードすることができる（例えば、ユーザがプレイヤをアンインストールし、最新のバージョンをメーカのウェブサイトからダウンロードする場合）。結果として、このようなシステムは、ハードウェアベースのプレイヤよりも堅牢でないセキュリティを提供する可能性があるが、コンテンツストリームを修正すること、および、ユーザが自分のソフトウェアをアップグレードすることを要求することにより、アップグレードされたセキュリティ機能を利用する（deploy）ことができるため、攻撃の影響は比較的小さい。
【０００４】
反対に、光メディアを再生する家電ハードウェアデバイスに組み込まれた保護方法は、知ってのとおりアップグレードすることが困難である。セキュリティの課題には、（後方互換性のないセキュリティアップグレードを妨げる）光メディアの長い耐用期間、アップデートをプレイヤに配信するための好都合で信頼性のある方法の欠如、および、複数のプレイヤ実装の間の標準化の欠如が含まれる。これらの難点は、再生デバイスの長い耐用期間、および、すべての新しいコンテンツが古いプレイヤ上で再生されるようになるという消費者の期待と相まって、セキュリティアップグレードの導入を非常に困難にする。結果として、大部分の家電デバイスは、コピーに対する実際上の保護をほとんど、あるいはまったく提供せず、家電デバイスにおいて利用される少数のコンテンツ保護標準は、柔軟性および更新可能性（renewabilty）をほとんど提供しない、単純で柔軟性のないスキームとなる傾向がある。図１は、背景技術の典型的なコンテンツ保護システムを示している。コンテンツプレイヤ１００は、ソフトウェアを不揮発性プログラムメモリ１０５内に含み、このソフトウェアは、プレイヤのセキュリティポリシ１１０、復号化コード１２０、およびプレイヤキー１３０を実装する。このコードおよびキーは、プロセッサ１４０によって使用されて、メディア１５０から読み取られたコンテンツが有効であるかどうかの妥当性検査を行い、コンテンツが有効である場合、このコードおよびキーは、コンテンツを復号化し、その結果を出力インターフェース１６０に提供する。図１に示したような保護システムの実施例には、デジタルオーディオテープによって使用されるコピー制御スキーム、ＤＶＤビデオを保護するように意図されたコンテンツスクランブリングシステム（ＣＳＳ）、および、ＤＶＤオーディオを保護するために提案されたＣＰＰＭスキームが含まれる。
【０００５】
様々な異なる技術が背景技術において知られている。
【０００６】
アクセス制御ポリシ：幅広い種類のアクセスポリシ、および、このようなポリシを指定す

るための方法が、背景技術において知られている。例えば、Ｈｅｌｌｍａｎの米国特許第４６５８０９３号明細書に記載のソフトウェア保護システムは、パブリッシャによって発行された単純な認証コードを使用する。反対に、Ｇｉｎｔｅｒらの米国特許第５９８２８９１号明細書には、大多数の関係者（participant）を含む種々の非常に複雑なアクセスルールが説明されている。ＰｏｌｉｃｙＭａｋｅｒおよびＸ．５０９証明書フォーマットなどのアクセスポリシをエンコードするための標準規格（コンテンツ配布による使用および他のアプリケーションによる使用の双方のため）も提案されている。

【０００７】

アンチウィルスソフトウェア：既知のウィルス、トロイの木馬、および、他の悪意のあるコードを検出し、かつブロックするための方法が、背景技術においてよく知られている。これらの方法は一般に、既知の命令シーケンスなどの既知のウィルスの属性をスキャンすることを含む。これらのプログラムは、起動中にファイルをスキャンする、ファイルをオンザフライでスキャンする、プログラムを実行する際にプログラムをスキャンする、メモリをスキャンする、新しいメディアをスキャンする、ネットワーク通信をスキャンするなど、様々な方法で動作することができる。

【０００８】

コンテンツ保護システムおよびＤＲＭ：幅広い種類の保護システム（しばしば、デジタル著作権管理（ＤＲＭ）システムとも呼ばれる）が提案されている。背景技術におけるＤＲＭシステムは一般に、コンテンツが暗号化形式で配布されるように準備をし、次いで、正当な購入者のために復号化キーを供給するか、または復号化オペレーションを実行する。多数の機能が提案されている、または多数の機能が市販のＤＲＭに含まれてきているが、これらの機能には、（暗号化されたコンテンツをユーザ間で交換することができる）スーパーディストリビューションのサポート、ペイパーユースビリング（電話回線を介したレポーティングを伴うオフラインのペイパーユースを含む）、可変ビリングレート（プロモーション、使用回数または使用期間、要求されたユーザオペレーション、ユーザ履歴などに基づいて、異なる金額を請求する）、様々なデータタイプ（オーディオ、ビデオ、テキスト、ソフトウェアなど）の保護、様々なフォーマットのサポート、および、様々な再生デバイスタイプ（ポータブル、セットトップ、ハードウェア支援付きのコンピュータベース、ソフトウェア専用など）のサポートが含まれる。

【０００９】

コピープロテクト：パーソナルコンピュータソフトウェアをコピープロテクトするための複数の方法が知られており、これらの方法が、コンピュータゲームなどの特定の種類のソフトウェアのために広く利用されている。これらの方法にはしばしば、（例えば、故意にエラーを組み込むことによって、または、複製することが困難である非標準的なフォーマッティングによって）コピーすることが困難となるように設計される物理的メディアにソフトウェアプログラムをバインドすることが含まれる。他のコピープロテクトシステムには、例えば、認証コードをサーバから取得するようユーザに要求することにより、インストレーションプロセスを保護することが含まれる。場合によっては、コピープロテクト機能がシステム内に設計される。他の場合には（コンピュータソフトウェア、ビデオカセットテープ、およびオーディオＣＤのために使用されたコピープロテクトシステムを含む）、大部分のプレイヤ上で再生が可能であるが、メディアをコピーしようとする大部分の試みを混乱させる非標準的なエンコードによりメディアを作成することによって、コピープロテクトは実装される。コピープロテクトシステムに関する主要な設計課題は、正当なユーザへの影響を最小限にする（すなわち、高いプレイヤビリティ（playablity）およびユーザアクセプタンス（user acceptance）を得る）と同時に、望ましくないアクションをできるだけ効果的に防止すること（すなわち、十分なセキュリティを得ること）である。

【００１０】

暗号化機能：幅広い種類の基本的な暗号化機能が知られており、これらの機能には、ブロック暗号、ハッシュ関数、デジタル署名システム（および他の公開キーシステム）、キー管理システムなどが含まれる。基本的な暗号化についてのさらなる情報については、Bruc

e SchneierによるApplied Cryptographyを参照されたい。

【００１１】

暗号化オラクル：ブロック暗号または他の暗号化機能を使用して、「暗号化オラクル」を構築することができ、暗号化オラクルは、任意の外部供給された入力メッセージに秘密暗号化変換（secret cryptographic transformation）を適用し、結果を返す。オラクルのアルゴリズムおよびプロトコルを知っている攻撃者がオラクルのキーを決定することが計算的にできないように、暗号化オラクルを構築することができる。加えて、オラクルへの可能な入力の数を極度に大きくすることができる（例えば、２５６ビットのブロック暗号から構築されたオラクルでは、２＾２５６）ので、攻撃者は、ランダムな問い合わせに対する応答を予想または事前計算することができない。

【００１２】

インタープリタ、エミュレータおよび仮想マシン：様々なインタープリタコンピュータ言語が背景技術において知られている。Ｊａｖａ（登録商標）などの一部のインタープリタ言語では、ソースコードを実行可能または解釈可能な形式に変換するコンパイルプロセスが必要となる。反対に、大部分のＢＡＳＩＣインタープリタは、直接ソースコードに基づいて動作する。一部のインタープリタは、自己書き換えコードを可能にするが、他のインタープリタは自己書き換えコードを可能にしない。インタープリタを実行するための技術、および、アセンブリ言語をエミュレートするための技術も、背景技術において知られている。例えば、Ｖｉｒｔｕａｌ ＰＣ（登録商標）およびＳｏｆｔＷｉｎｄｏｗｓ（登録商標）などの高度なエミュレータは、Ｍｉｃｒｏｓｏｆｔ Ｗｉｎｄｏｗｓ（登録商標）用に設計されたプログラムをＡｐｐｌｅ Ｍａｃ（登録商標）コンピュータ上で実行させることができる。Ｊａｖａ（登録商標）およびＪａｖａＣａｒｄ（登録商標）用に使用されるものなどの仮想マシン（ＶＭ）設計が知られており、ＶＭがコンピュータ上のネイティブコードと対話すること、または、異なるメモリ空間内の他のＶＭ関数を呼び出すことができることも知られている（多数のＪａｖａ（登録商標）実装によって、これらの機能が提供される）。インタープリタ言語は通常、アプリケーションのため、または、クロスプラットフォーム互換性が必要とされるところで、プロセッサ非依存デバイスドライバフォーマット（processor-independent device driver format）を作成するためなどに使用される（例えば、Writing FCode 2.x Programs, Sun Microsystems, 1993, page 5参照）。

【００１３】

キー管理：暗号化キーを割り当て、かつ管理するための幅広い種類の方法が提案されている。デバイスがデバイス固有のキー、グループキー、公開キー、秘密キー、証明書などを有することができることが知られている。キーを、個々のデバイスに、（例えば、Ｆｉａｔの米国特許第５５９２５５２号明細書に記載されたような）選択されたデバイスのグループに、すべてのデバイスになど、割り当てることができる。デバイスは、異なるタイプの様々なキーを含むことができ、これらのキーには、対称キー、公開キー（例えば、証明書およびデジタル署名を検証するため）、および非対称秘密キーが含まれる。

【００１４】

メディア：多大なストレージ容量、低製造コスト、および十分な耐久性を提供することができる、メディア技術が知られている。現在のメディア技術の例には、光ディスク（ＣＤ、ＤＶＤなど）、磁気メディア、フラッシュメモリおよびＲＯＭが含まれる。ホログラフィックメモリなどのより新しい技術も開発されている。単一のメディアが多数の異なるタイプのデータを含むことができることが知られている。例えば、コンパクトディスクには、標準のレッドブックのオーディオトラック、ならびに、パーソナルコンピュータ上で使用するためのデータセッション（例えば、ソフトウェア、圧縮されたボーナストラック、イメージ、ビデオ、歌詞などを含む）を含めることができる。パーソナルコンピュータで使用するためのコンパクトディスクには、暗号化されたコンテンツ、ならびに、コンテンツを再生するために必要とされる再生ソフトウェアの両方を含めることができる。

【００１５】

ネットワーク通信：インターネットを含む高度なデータネットワークが知られている。これらのネットワークは、柔軟性のある、信頼性のある、高帯域データ通信を提供することができる。物理的接続を伴うネットワークは通常、より高い帯域幅を提供することができるが、無線通信チャネルもまた普及している。

【００１６】

更新可能セキュリティ（renewable security）：場合によっては、可能性のある攻撃のすべてを確実に防止することができるセキュリティシステムを作成することは、現実的ではない。結果として、例えば、任意の危殆化されたキーの使用を中断し、脆弱性を是正することによって、攻撃の後にセキュリティを更新することができることが望ましい。更新可能なセキュリティは望ましいが、多数の利用されているシステムおよび提案されているシステムは、多数の種類の攻撃に対する効果的な復旧メカニズムを欠く。

【００１７】

サンドボクシング：サンドボクシングは、制御された環境内でソフトウェアプログラムを実行することを含み、この環境では、プログラムは、システムに損害を与える可能性のあるいかなるオペレーションにもアクセスすることができない。Ｊａｖａ（登録商標）「仮想マシン」はサンドボクシングをサポートするので、信頼できないアプレット（インターネットを介してダウンロードされるものなど）を実行することができる。

【００１８】

セキュリティモジュール：多数のセキュリティシステムは、リムーバブルなセキュリティモジュールを使用できるので、システムの他の部分を置き換える難点または費用なしに、セキュリティアップグレードを実行することができる。例えば、リムーバブルなセキュリティモジュールは、多数の有料テレビシステムにおいて使用される。

【００１９】

ソフトウェアアップデート：提示されたソフトウェアアップデートを受信し、アップデートの妥当性検査を行うデジタル署名またはメッセージ認証コードを検証し、次いで、（署名が有効である場合、）アップデートを実行することによって、安全なソフトウェアアップデートを実行することができる。例えば、デジタルオーディオプレイヤはコードアップデートを受信し、アップデートにおけるデジタル署名またはメッセージ認証コードを検証し、（有効である場合、）それらのコードをアップデートすることができることが知られている。アップデートを正しい順序で確実に行うための方法（例えば、シーケンスカウンタを使用する）、および、（例えば、以前のソフトウェアバージョンに戻ることによって、または、特別な復旧コードを実行することによって）失敗または不成功のアップデートから復旧するための方法も知られている。インターネット、光メディア、ＲＯＭカートリッジなどの幅広い種類の配布メカニズムを介して、ソフトウェアアップデートを配信することができることも知られている。コードアップデートを信号とともにデスクランブラ（descrambler）へ配布することによって有料テレビに対する海賊行為を防止するために、ソフトウェアアップデートは使用されており、デスクランブラは新しいコードを適用し、首尾よく実行して、次のビデオセグメントのための正しい復号化キーを計算する。これらのアップデートは、通常、無許可のデスクランブラを無効にすることによって、または、さらに言えば破壊することによって、無許可の視聴を防止するために使用される。

【００２０】

ステガノグラフィ：ステガノグラフィは、情報をデータ内に隠すことを含む。例えば、暗号化されたデータをイメージまたは録音の最下位ビットに配置することができることが知られている。このイメージまたは録音を得るが復号化キーを知らない攻撃者は、隠されたデータがあるかどうかを判断することもできない。これは、下位ビットがしばしばランダムに見え、強力な暗号化アルゴリズムによって作成された暗号文を、そのキーなしにランダムデータと区別することができないからである。

【００２１】

耐タンパ性：攻撃に強いデバイスを設計および構築するための多数の方法が知られている。耐タンパ性を有するハードウェアは、通常、攻撃者がデバイスをリバースエンジニアリ

ングすること、または、キーを暗号化モジュールから抽出することを防止することが望ましいシステムにおいて使用される。例えば、Ｗａｖｅ　Ｓｙｓｔｅｍｓは、「Ｅｍｂａｓｓｙ」と呼ばれる、耐タンパ性を有するマイクロプロセッサベースの集積回路製品を市場に出しているが、この製品は、コンテンツプレイヤまたは汎用コンピュータに組み入れることができ、デジタルコンテンツの配布を保護する際に使用するよう宣伝されている。耐タンパ性を有するソフトウェアを実装するための方法もまた提案されている（例えば、Ａｕｃｓｍｉｔｈらの米国特許第５８９２８９９号明細書参照）。

【００２２】

トレイタトレーシング（Traitor Tracing）：通常、無許可のデバイスにおいて使用されたキーを顧客の特定デバイスまたは危殆化されたデバイスに遡ってトレースすることによって危殆化元または攻撃元を特定するために、トレイタトレーシングスキームが提案されている。

【００２３】

ウォータマーキング：ウォータマークは、特別な検出器によって検出することはできるが、コンテンツが再生されるときに人間によるコンテンツの知覚に影響を与えない（または影響を最小限にする）、コンテンツ内に埋め込まれた信号である。ピクチャ、録音およびイメージ内に埋め込まれたウォータマークは、コピーが許可されないことを明示するために、著作権保持者によって使用されている。「堅牢な」ウォータマークが知られており、このウォータマークは、フォーマット間の変換（アナログ出力からの再記録を含む）に抵抗し、ウォータマークを除去しようとする攻撃に対して様々な程度のセキュリティを提供することができる。反対に、「脆弱な」ウォータマークは、フォーマット変換に抵抗する能力をほとんど、またはまったく有していないが、より設計しやすく、より多くの情報を伝達することができる。

【発明の概要】

【発明が解決しようとする課題】

【００２４】

どのアンチパイラシシステムも可能性のある攻撃のすべてを完全に防止することはできないが、背景技術におけるシステムは、保護フォーマット（protected format）から非保護フォーマット（unprotected format）へのデジタル－デジタルコピーを使用する、または保護フォーマットから無保護フォーマットへの高速リッピングを使用するちょっとした(casual)海賊行為などの解決可能な問題に対する実用的な解決策を提供することができない。背景技術における多数のシステムの重大な制限には、以下のものが含まれるが、これらに限定されるものではない。

【００２５】

グローバルシークレットへの依存：多数の保護システムでは、暗号化アルゴリズム、キー、および、復号化するために必要とされる他の情報が秘密に保たれることが必要である。結果として、システムのセキュリティを危殆化させることなく、復号化プロセスをオープンスタンダードのドキュメントの形で文書化することができない。また、多数の実装が使用可能である場合、攻撃者は、最も弱い実装を攻撃することによって、おそらくはスキーム全体を破壊することができる（このような攻撃は、ＤＶＤビデオプロテクトシステムで最近発生した）。このようなシステムは、閉じられた単一ベンダ環境内では有用であるが、このようなシステムを標準化することはできず、このようなシステムは有効な長期のセキュリティを提供しない。

【００２６】

標準化の欠如：コンテンツパブリッシャはすでに、互換性のない様々なデータフォーマットおよび復号化アルゴリズムにコミットしている。異なるコンテンツ保護システムは異なるビジネスモデルを可能にし、あるモデルにコミットしているパブリッシャは、異なるモデルを必要とする任意のセキュリティシステムに反対する可能性が高い。

【００２７】

製品タイプとの非互換性：多数のセキュリティ機能をすべての製品タイプと統合すること

はできない。例えば、パーソナルコンピュータ用のダウンロード可能なソフトウェア専用プレイヤは、耐タンパ性を有するハードウェアを含むことはできない。同様に、インターネット接続性を欠くプレイヤに、頻繁にソフトウェアアップデートを配信することは困難である。

【００２８】

ユーザインターフェース：多数の提案は、複雑なユーザインターフェースを含む。セキュリティは、真正なユーザにとって不可視であるべきである。ユーザは、明示的なユーザの関与（例えば、認証コードを取得する、または認証コードを入力すること）を必要とするスキームを拒否する可能性が高い。一般に、カーステレオおよびビデオディスクプレイヤなどの家電デバイスは使いやすい必要がある。これは、多数のユーザが、文書を読まない場合、技術に及び腰である場合、視力が弱い、もしくは他のハンディキャップを有する場合、または、プレイヤによってサポートされた言語に精通していない場合でも、そのユーザを満足させなければならないからである。

【００２９】

法的な課題：一部のセキュリティシステムは、競業者間の協力を必要とする。このような協力は、反トラスト規制により違法になる可能性がある。

【００３０】

メーカの利点の欠如：メーカは、プレイヤのコストを増し、かつ製品化までの時間を増す、正当な機能の包含を妨げるセキュリティ機能に反対するようになるし、または、それらの製品の効果を弱め、もしくは製品の効果をより望ましくないようにするセキュリティ機能に反対するようになる。半導体技術における進歩は、セキュリティシステムを実装するために必要とされるコストを下げつつあるが、耐タンパ性を有する有効なハードウェアを設計および製造することは困難なままであり、高価なままである。結果として、優れた実装を作成するメーカに依拠するコンテンツ保護システムは、その提供物がより安全であるメーカに実際の市場優位性を提供しない限り、失敗する。

【００３１】

不明確なセキュリティポリシ：有効なセキュリティシステムでは、ユーザが要求した特定のアクションを許可すべきであるか、防止すべきであるかを判断するためのルールまたは他の意思決定プロシージャが指定されなければならない。多数のシステムでは、これらのルールまたはプロシージャがうまく指定されない。

【００３２】

柔軟性のないセキュリティポリシ：コンテンツ保護システムは、異なるパブリッシャ、コンテンツタイプ、権限（jurisdiction）、再生環境などに対して異なるモデルをサポートするための柔軟性を有することが望ましい。システムは、複雑になりすぎることなく、必要な柔軟性を提供すべきである。

【００３３】

脆弱な長期セキュリティ：セキュリティシステムは、長期にわたって有効であり続けるために十分に堅牢であり、柔軟性がなければならない。普及しているフォーマットは３０年を超えて存続することができるが、背景技術におけるコンテンツ保護システムのほとんどは、著名なフォーマットの一部として、数年を超えて存続することができない。

【００３４】

攻撃の非トレース可能性（untraceability）：攻撃者が生じる場合、システムは、危殆化された（または悪用された）デバイスを無効にすることができるように、および、犯人を起訴することができるように、攻撃のソースを特定することができる必要がある。

【課題を解決するための手段】

【００３５】

本願は、幅広い種類の相互運用可能なプラットフォームにわたって柔軟性のある更新可能なコンテンツ保護を提供するように実施することができる、標準化されたコンテンツ保護システムの様々な実施形態および態様に関する。このシステムは、関係者（メーカ、パブリッシャ、アーティスト、および／または消費者など）に、セキュリティおよび機能性に

関する意思決定を行うための比類のない柔軟性を提供する。

【００３６】

このシステムと共に使用可能な例示的プレイヤ（すなわち、保護されたコンテンツを復号化することを望むデバイス、またはそうでない場合は、保護されたコンテンツにアクセスすることを望むデバイス）は、複数のコンポーネントを備える。第１のコンポーネントは、例えば光ディスクドライブ用の、データまたはメディア入力インターフェースである。再生を開始するため、プレイヤは一連のデータ処理コマンドを入力インターフェースからロードし、インタープリタまたは他の実行モジュールを使用することにより、これらのコマンドの実行を開始する。この実行環境は、チューリング完全言語（Turing-complete language）（プレイヤのメモリ、ユーザインターフェースおよびパフォーマンスの制限を受ける、任意のアルゴリズムを実行することができる言語）を提供することが好ましい。この実行環境からコンテンツはプレイヤに問い合わせて、再生環境の構成を決定することができ、プレイヤのキーを使用することにより暗号化オペレーションを実行することができる。したがって、問い合わせに対して満足のいく応答を提供するプレイヤ上でのみ再生が進行することになるように、コンテンツを設計することができる。パブリッシャもまた、制限された再生を提供することができる。例えば、より安全でないプラットフォームでは、ＣＤ音質のステレオオーディオまたは通常の精細度のイメージを提供することができるが、より安全なプラットフォームでは、より多くのオーディオチャネル、高精細度のイメージ、より高いサンプリングレート、および、より高品質の圧縮を提供することができる。再生が開始した後でも、再生はコンテンツのデータ処理コマンドの制御下に残ることができる。１つの例示的実施形態には、堅牢な、本質的にオンザフライのウォータマーキングを実行する機能が含まれる。どのデータ領域が再生されるかをコンテンツ自体が制御できるようにすることで、極めて小さい差異を伴う複数の出力データバージョンから選択することにより、情報を出力内に埋め込むことが可能となる。これらの差異を解析することにより、特定のプレイヤに遡って海賊コピーをトレースすることができる。

【００３７】

コンテンツは、コンテンツ固有のセキュリティポリシを含み、これを実施するので、耐性のある新しいコンテンツを設計し、かつ発行することによって、発生する攻撃に対処することができる。コンテンツがコンテンツ固有のセキュリティポリシを実施することを可能にすることによって得られた柔軟性は、さらに、アーティストのプリファレンス、地域的な「公正使用」の規制などに対するサポートも可能にする。コンテンツにアクセス可能な新しいプレイヤ関数を追加することによって、新しいプレイヤ機能を容易に追加することができる。

【００３８】

ビジネスの観点から、コンテンツパブリッシャおよび家電メーカのビジネスおよびオペレーション上の制約と一致する、最良となり得るセキュリティを提供するという共通の目標においてコンテンツパブリッシャと家電メーカとを結び付けるように、任意のコンテンツ保護システムが使用できることが望ましい。本明細書で開示するシステムにより、パブリッシャはパブリッシャ固有のセキュリティ要件を決定することができるようになり、次いで、コンテンツ自体が、幅広い種類の要素を考慮するポリシを実装して、各環境内で再生されるべきであるかどうか（またはどのように再生されるべきであるか）を決定することができるようになる。さらに、優れたセキュリティを提供し、海賊行為を容易にしない製品を設計するようメーカを動機付けることができるので、それらの顧客はコンテンツに対して幅広くアクセスできるようになる。

【図面の簡単な説明】

【００３９】

【図１】背景技術のコンテンツ保護方法を使用するメディアプレイヤを示す図である。

【図２】本明細書で開示したコンテンツ保護方法を使用する例示的メディアプレイヤを示す図である。

【図３】例示的実施形態の復号化部分を例示する図である。

【図４】集中コード署名局を必要とすることなく、不揮発性メモリへのアクセスを保護するための、プレイヤ方法の例示的実施形態を示す図である。
【図５】不揮発性メモリスロットにアタッチするときの例示的妥当性検査プロセスを示す図である。
【発明を実施するための形態】
【００４０】
図２は、物理的メディア２００を使用するプレイヤの例示的実施形態を示している。再生プロセスはプロセッサ２１０によって制御され、プロセッサ２１０はメディアインターフェース２０５を介してメディア２００にアクセスすることができる。メディア２００がマウントされるとき（例えば、メディアが最初に挿入される、または、システムが再初期化されるときなど）、プロセッサ２１０は、メディアインターフェースを初期化すること、メディアの目次を読み取ること、および、サポートされた保護システムを認識することによって、開始する。そのような場合、プロセッサはメディア２００の小さな初期部分を実行およびデータＲＡＭ２２０にロードする。
【００４１】
インタープリタ２１５を使用することにより、プロセッサ２１０は、ロードされたメディア部分によって指定されたデータ処理オペレーションの実行を開始する。インタープリタ２１５は所定のデータ処理オペレーションのセットを提供し、このセットによって、より複雑なタスクを実行することができる。インタープリタ言語は、好ましくはチューリング完全である。チューリング完全プログラミング言語は、１つのこのような言語において実装可能なアルゴリズムを、他の任意の言語においても実装することができ、これらの実装が類似の漸近パフォーマンス特性（asymptotic performance characteristic）を有するようになることを特徴とする。チューリング完全プログラミング言語の例には、Ｃ、Ｃ＋＋、ＢＡＳＩＣ、Ｆｏｒｔｒａｎ、Ｐａｓｃａｌ、Ｊａｖａ（登録商標）および実質的にすべてのアセンブリ言語が含まれるが、これらに限定されるものではない。
【００４２】
ロードされた部分は、インタープリタ２１５によって提供されるプロシージャコールを呼び出すことによって進行する。ＲＡＭ２２０にロードされた初期データは比較的小さい場合があるが、インタープリタ２１５上で実行されるコードは、追加のデータ（コードを含む）を、プロシージャコールを介してメディアからロードすることができ、それによってより複雑なオペレーションを実行することができる。
【００４３】
他のプロシージャコールによって、コンテンツは再生環境構成２２５を決定することができる。したがって、コンテンツは再生環境特性（例えば、プレイヤのタイプ、要求されたユーザクションなど）を解析して、再生が進行すべきであるかどうかを判断することができる。例示的実施形態では、是正可能な問題が検出される場合（例えば、メディアがプレイヤ用のセキュリティファームウェアアップグレードを含む場合）、これらに対処することができる。サポートされる場合、コンテンツは、出力インターフェース２５０、および、サポートされる場合、送信先プログラム／デバイス２６０（例えば、増幅器、デジタルスピーカ、スピーカドライバなど）に問い合わせて、セキュリティ特性を調べ、暗号化キーをロードし、出力パラメータを指定する（例えば、セキュリティが不確実である場合、低下した出力品質を指定する）ことなどもできる。
【００４４】
例示的実施形態では、コンテンツは、暗号化オラクル２３０に問い合わせることもでき、セキュリティハードウェアアップグレードを可能にするために、（スマートカードなどの）外部のリムーバブルセキュリティモジュール２３５内に暗号化オラクル２３０を実装することができる。プロセッサ２１０内、プレイヤ内の他のハードウェア、メディア内、スピーカなどのアタッチされたデバイス内などに、オラクルを実装することもできる。暗号化オラクル２３０は、プレイヤ識別に関する検証可能な証をコンテンツに提供することができる。オラクル２３０への問い合わせの結果は、後続のコンテンツまたはコード部分を

復号化する際に使用することができ、それによって、オラクル２３０は、有効なキーを欠く（またはそのキーが無効となる）プレイヤがコンテンツを復号化することができないという強力な暗号化の保証を提供することができる。

【００４５】

例示的実施形態では、インタープリタは、「サンドボックス」内のコンテンツによって指定されたデータ処理コマンドを実行するが、これは、コンテンツが暗号化秘密（cryptographic secret）（オラクルキーなど）に対してアクセスしないことを意味し、そうでない場合、コンテンツは、プレイヤを危殆化させるおそれがある。サンドボクシングは、すべてのコンテンツが必ずしも信頼できるとは限らない場合に有用である。例えば、攻撃者は、プレイヤから暗号化キーを抽出しようと試みた悪意のあるコンテンツを作成しようとする可能性があった。（以下で、例示的暗号化オラクルおよびそれらのオペレーションについてのさらなる情報を提供する。）

【００４６】

コンテンツが、再生が進行するべきでないと判断する場合（例えば、ユーザがコピーを作成しようとしていて、コンテンツがコピーを禁止するように構成されている場合）、コンテンツはエラーを報告し、要求されたアクションを拒否することができる。代替として、コンテンツはレンダリングプロセスおよび／または出力プロセスを制御して、出力の品質を低下させることができるので、無許可のコピーの品質が落ち、このため無許可のコピーは魅力的でなくなる。

【００４７】

コンテンツが、再生が進行するべきであると判断する場合、コンテンツは次いで、再生がメディア上の特定の位置（例えば、特定のトラック）から開始すべきであると指示するプレイヤからの信号を待つ。インタープリタ２１５は、メディアがマウントされたとき、実行／データＲＡＭ２２０にロードされたデータ処理命令を使用して、要求を処理する。コンテンツは、再生が進行するべきであると決定する場合、プロシージャコールを使用して、暗号化されたコンテンツをメディア２００上の適切な位置からロードすることを開始するようメディアインターフェース２０５に命令する。コンテンツは有効な復号化キーおよびパラメータをバルク復号化モジュール２４０に対して指定し、バルク復号化モジュール２４０は、暗号化されたコンテンツをＲＡＭ２２０から（または代替として、メディアインターフェース２０５から直接）読み出して、これを復号化する。復号化されたコンテンツは次いで出力インターフェース２５０に供給され、出力インターフェース２５０はこのコンテンツを、送信先プログラムまたはデバイス２６０に合わせて適切なアナログまたはデジタルフォーマットに変換する。再生が継続するとき、インタープリタ２１５によって処理されているデータ処理命令は新しい復号化パラメータをロードし、新しいデータのブロックを指定してメディア２００から読み取ることなどができる。再生が完了するとき、コンテンツはＲＡＭ２２０を再初期化することができる。

【００４８】

以下のセクションにおいて、インタープリタ、再生システム、ならびに、他の実施形態および態様について、追加の情報を提供する。

【００４９】

（攻撃への対処）

ソフトウェアにおいて、および、低コスト家電デバイスにおいて幅広く実装されているアンチパイラシシステムは、可能性のある攻撃のすべてを防止することはできない。本明細書に開示する技術は、ある攻撃の後に続いて、既存の攻撃を実質的にブロックする方法で新しいコンテンツをマスタリングすることを容易にするように使用することができる。専門的な海賊行為者は継続的に新しい回避システムを探し出してインストールしようと試みる可能性がある一方で、ちょっとした海賊行為は、攻撃ツールを開発および維持するための継続的な努力を必要とするため、したがって、うまくいけば単にコンテンツを正当に購入するよりも困難となる。以下のセクションでは、本明細書で説明する技術をどのように使用することにより、一部の一般的な攻撃に対処することができるかを説明する。

【００５０】
第１の攻撃のカテゴリは、危殆化されていないプレイヤを使用して、無許可のアクションを実行しようとする試行を含む。例えば、コンテンツをマスタリングして、オリジナルのメディアからのコピーを可能にするが、コピーからのコピーを許可しないようにすることができる。このようなコンテンツをコピーからコピーしようとする試行がなされる場合（コンテンツは、例えば、コピープロセス中に挿入された変更を検出することによって、または、現在のメディアのシリアルナンバおよび／またはタイプをオリジナルと比較することによって、認識することができる）、再生をインタープリタコードによってブロックすることができる。代替として、インタープリタは、コンテンツが低下した品質で（より高いサンプルレートのマルチチャネルオーディオが使用できる可能性があるにもかかわらず、４４．１キロヘルツのサンプルレートのステレオオーディオを再生するなど）、または、追加のアンチパイラシ警告を挿入することによって、再生を認めることができる。したがって、インタープリタに提供された情報を解析することによって、不適切なユーザ要求を、危殆化されていないプレイヤ上で検出し、処理することができる。
【００５１】
第２の攻撃のカテゴリは、プレイヤの暗号化キーの危殆化を含む。プレイヤの暗号化キーが危殆化されている場合、攻撃者は、（少なくとも理論的には、）暗号化オラクルをエミュレートすることによって、および（任意で）再生環境についての問い合わせに対して偽の応答を提供することによって、危殆化された再生環境を完全にエミュレートすることができる。このような攻撃の後、それ以後のコンテンツ内のインタープリタコードが危殆化されたデバイス内に存在しなかった少なくとも１つの暗号化キーを要求するようにすることによって、セキュリティを再確立することができる。特定のプレイヤモデルまたはメーカが多数の攻撃のソースである場合（例えば、プレイヤ実装が不十分なセキュリティしか有さないため）、パブリッシャは、このようなプラットフォーム上で再生されないようになる（または、低下した品質で再生されるようになる）コンテンツを作成することができる。
【００５２】
第３の攻撃のカテゴリは、類似のインタープリタセキュリティコードを含む、コンテンツの特定の部分またはタイトルのグループを危殆化させることを含む。セキュリティチェックを回避するようにコンテンツ自体を変更することによって、または、ターゲットタイトルを再生するように調整された悪意のあるインタープリタを作成することによって、おそらくはこのような攻撃を開始することができる。以後のコンテンツにおいて、異なる保護ソフトウェア、またはより良い保護ソフトウェアを利用することによって、このような攻撃に対処することができる。
【００５３】
第４の攻撃のカテゴリは、コンテンツを保護メディア（protected media）から非保護フォーマットへコピーし、次いでコンテンツをその新しいフォーマットで再配布することを含む。どのコンテンツ保護システムもこのような攻撃を完全に防止することはできないが、本明細書で開示する技術およびシステムは、強力で柔軟性のあるウォータマーキング機能を備え、この機能を使用して、特定のデバイスに遡って危殆化をトレースすることができ、次いで、以後の攻撃を防止するために、この特定のデバイスを失効させることができる。海賊行為目的で活発にコンテンツをアップロードするユーザの数は比較的少ないので、これらのユーザのプレイヤを特定し、失効することによって、海賊行為を大幅に減らすことができる。暗号文の諸部分を選択的にスキップすることによって、知覚できない差異を復号化出力内に導入することができる。例えば、例示的実施形態では、第１の暗号文の一部分を復号化および出力し、次いで第２の暗号文の一部分をスキップするよう、コンテンツがプレイヤの復号化モジュールに命令することによって、「０」ビットにウォータマークを入れることができる。「１」ビットにウォータマークを入れるため、第１の暗号文の一部分をスキップし、第２の暗号文の一部分を出力するよう、コンテンツはモジュールに命令することができる。一連のこのようなビットをエンコードすることにより、インタ

ープリタコードにとって使用可能な任意のデータによって、コンテンツにウォータマークを入れることができる。このデータには、プレイヤの識別、暗号化オペレーションの結果、ユーザアクション説明、出力デバイス情報などが含まれるが、これらに限定されるものではない。コンテンツの海賊コピーが発見される場合、ウォータマークを解析して、不法コピーを単一のプレイヤまで遡ってトレースすることができ、次いで、このプレイヤを以降のコンテンツリリースにおいて失効させることができる。この機能は、法的処置（law enforcement）および科学捜査（forensic）の目的においても有用である。なぜならば、特定のコピーが特定のプレイヤから生じたことを、確信をもって証明することができるからである。コピーをトレースするための機能は、海賊行為に対する阻害要因としての機能を果たすこともできる。なぜならば、違法コピーの作成を検討する人は、自分が特定され、逮捕され、起訴される可能性があることを知ることによって、コピーする気がそがれるようになるならである。

【００５４】

言うまでもなく、すべての環境における可能性のある攻撃のすべてを確実に防止することができる、消費者が扱いやすい（consumer-friendly）アンチパイラシシステムはない。例えば、オーディオおよびビデオは、アナログ出力から記録することができる（ウォータマークがコンテンツ内に埋め込まれる場合でも、ウォータマーク検出器を有していないレコーダを使用できる）。アナログ出力から取り込まれたデータは次いで、新しいデジタルまたはアナログメディア上にリマスタリングされ、元々のセキュリティ機能を有することなく再配布される可能性がある。同様に、メディアの正確なコピーを作成するために必要とされる機器を有する専門の海賊行為者によって作成されたコピーを、プレイヤによって検出することはできないが、本明細書で開示する技術およびシステムは、メディアのコピーを防止する上で役立つことができる。例えば、メディア上のディスクメーカ識別子はコンテンツによって調べられ、真正の、または不注意な複製施設が海賊行為者によってだまされないようにすることを確実にすることができる。メディアタイプ識別子は、読み取り専用メディアの形態で販売されたコンテンツが、記録可能メディアの形態で再配布されることを防止できる。インターネット、電話／モデム、または他のネットワークサポートを使用するプレイヤでは、コンテンツは（例えば）再生（または最初の再生）に先立って、サーバから認証を受け、そのメディアが有効であることを検査することができる。不揮発性ストレージを有するプレイヤは、既知の不良メディアシリアルナンバのテーブルを格納することもでき、コンテンツおよび／またはプレイヤがこのテーブルに問い合わせて、メディアが失効しているどうかを判断することができる。

【００５５】

（再生環境への問い合わせおよび再生環境の制御）

コンテンツを、コンテンツによってコンテンツ自体が復号化されることが可能になるかどうかを決定するように、構成することができる。この決定を支援するため、プレイヤは再生環境についての情報をコンテンツに提供することができる。多くの場合は非常に限られた情報（ユーザにより要求されたアクションおよびプレイヤモデルなど）だけで十分である可能性が高いが、再生が進行すべきであるかどうかについて、コンテンツがより確かな情報に基づいて評価することができるように、より詳細で正確な情報が望ましい。コンテンツに提供される特定の情報および機能はプレイヤ実装に依存するが、以下では、コンテンツに提供することができる一部の例示的関数および機能（ただし、以下に限定はされない）を説明する。複数の接続されたコンポーネント（出力ポート、接続された出力デバイス、オペレーティングシステムデバイスドライバ、セキュリティモジュールなど）から構築されたプレイヤでは、これらの接続されたデバイス、ならびに、インタープリタを含むプレイヤの主要部分が、以下の情報の一部またはすべてを含むことができることに留意されたい。

【００５６】

セキュリティサポート情報：セキュリティ仕様バージョン、サポートされた問い合わせ機能、および／または、セキュリティモジュール形状因子（form factor）（置換可能なハ

ードウェア、組み込みハードウェア、アップデート可能なファームウェア、ＲＯＭファームウェア、ＰＣソフトウェアなど）など。（例示的暗号化処理機能および再生制御／復号化機能については、以下で詳述する。）

【００５７】

メーカ情報：名前、識別子、ウェブサイト、公開キー／証明書、製造バッチ、製造日／時間、製造地域、製造国、メーカ住所、技術サポート連絡先情報、および／または、メーカ保証情報など。

【００５８】

デバイス情報：製造ライン、シリアルナンバ、モデルナンバ、ファームウェア／ソフトウェアバージョン、デバイス公開キー／証明書識別子、ＧＰＳ位置または他の物理的位置／地域、コンテンツによりサポートされたコーデックタイプ、ネットワーク／インターネットサポート情報、ネットワークアドレス、デバイス電話番号、ＩＰアドレス、ウォータマークサポート、インタープリタパフォーマンス評価、セキュリティ証明評価、デバイスディストリビュータ、デバイス小売業者、デバイス形状因子、および／または、セキュリティ仕様など。

【００５９】

ユーザ情報：ユーザ名、地理的地域、国、住所、ＧＰＳ位置または他の物理的位置／地域／国など、ユーザ電話番号、ＩＰアドレス、電子メールアドレス、ウェブアドレス、希望する言語、問題となるマテリアルに対する許容範囲、希望する支払い方法／口座、支払い制限、購入履歴、および／または、プライバシプリファレンスなど。

【００６０】

メディア制御：問い合わせメディアフォーマット、記録可能または記録不能、メディアシリアルナンバ、記録デバイスタイプ、記録デバイス所有者、記録デバイスシリアルナンバ、記録デバイスセキュリティ情報、および／または、記録デバイスウォータマークチェック機能など。関数は、メディアからの読み取り、メディアへの書き込み、メディアのフォーマット、メディアのテスト、および／または、メディアの排出なども可能にする。追加の関数は、特定のメディアフォーマットによってサポートされる暗号化機能または他の特別な機能へのアクセスを提供することができる。

【００６１】

要求されたユーザオペレーション：例えば、再生、記録、新しいフォーマットへの変換、ポータブルデバイスへのロード、最初のコピーの作成、複数のコピーの作成、および／または、同時再生／記録など。要求されたオペレーションを開始または変更する能力をコンテンツに与えることもできる。

【００６２】

出力情報：出力ポート、出力ポート構成、出力ポートセキュリティ特性、接続されたデバイス、出力データフォーマット、および／または、出力データ品質／解像度などについての情報。サポートされる場合、コンテンツは出力デバイスに直接問い合わせて、デバイスについての追加の情報を得ること、および／または、暗号化オペレーションを要求することなどができる。プレイヤは、コンテンツがこれらのパラメータを変更して、例えば、セキュリティが不十分である場合には低下した品質の出力を指定することを可能にすることもできる。

【００６３】

環境：プラットフォーム上の他の実行中のプログラムおよびデバイスドライバの識別／ハッシュ／バージョン、メモリのコンテンツまたはハッシュ、インストールされた攻撃検出モジュールのバージョン、攻撃に関するシステムスキャンの結果、および／または、改ざん検出器の状況など。これらの機能によって、コンテンツはメモリを修正して、例えば、他のプログラム内のセキュリティ弱点を是正することもできる。

【００６４】

時間：日付、時間、時間帯、経過クロックサイクル数、最終リセットからの時間、製造からの時間、最終セキュリティアップグレードからの時間、最終バッテリ交換からの時間、

および／または、推定の残りバッテリ寿命など。

【００６５】

接続性：プレイヤ通信機能の決定、現在の接続状況のチェック、ネットワーク接続の確立、モデム接続の確立、ネットワーク接続を確立する臨界の指定、接続セキュリティ特性のチェック／指定、データ送信、データ受信、接続を閉じること、および／または、接続のアイドリングなど。

【００６６】

ユーザインターフェース：ユーザメッセージの表示、歌詞の表示、グラフィックスイメージの表示、グラフィックスイメージの印刷、広告／プロモーションメッセージの表示、使用可能なユーザインターフェース制御の識別、ユーザ入力の取得、プレイヤの音声シンセサイザを使用して音声をユーザに再生すること、および／または、エラーの報告など。

【００６７】

ウォータマーク制御：出力すべきコンテンツ領域の選択、外部ウォータマーキングアルゴリズムの選択、外部ウォータマーク検出器の制御、および／または、マーク検出器状況のチェックなど。

【００６８】

その他：プレイヤ／再生状況情報、ペイパープレイビリング制御（例えば、プレイヤベースの資金源）、エラー処理、再生終了、安全な不揮発性メモリサポート（下記参照）、プレイヤファームウェアアップデートの適用、および／または、外部モジュール（動的にリンクされたライブラリなど）の呼び出しなど。

【００６９】

関数およびパラメータの標準化の一部は、複数の実装の間の相互運用性を確実にするため（例えば、コンテンツが最初に発行された後に設計されたプレイヤ環境内で、コンテンツが効果的に機能することができるようにする）、および、安全なコンテンツを作成するタスクを単純化するために有用である。標準化は、様々な異なるメーカの製品が同じタイプの情報またはオペレーションを提供すべきである機能に関して特に有用である。例えば、コンテンツがプレイヤ形状因子（ホームオーディオ／ビデオ、ポータブル、自動車用、パーソナルコンピュータソフトウェア専用、ハードウェア支援を有するパーソナルコンピュータソフトウェア、専門スタジオ、映画館など）を決定することを可能にするための関数および応答コードを、標準化することができる。標準化は、以前から存在するコンテンツが理解できない非標準フォーマットで適切なリスク関連情報を報告することによって、メーカがセキュリティ制御を回避しようと試みることを防止するという、さらなる利点を有する。

【００７０】

言うまでもなく、メーカがさらなるメーカ独自の関数を追加して、それらを使用することを選択するコンテンツ作成者によって使用されるように、このシステムをさらに構成することもできる。新しい関数を追加するための能力は、新しい機能を自社製品に追加することを望むメーカにとって特に貴重である。なぜならば、メーカがこれらの機能を追加し、次いでコンテンツパブリッシャと共に協調的なビジネス関係を確立して、これらの機能をサポートすることができるからである。このような一実施形態は、（望まれる場合は）後方互換性を維持しながら、容易に拡張することができる。

【００７１】

メーカは、正確な情報をコンテンツに提供する責任がある。コンテンツは一般に、受信する情報の多くの精度を直接検証することはできないが、メーカが、この情報が正しいことを保証するための強力なインセンティブを有する場合には、これは厳密には必要ではない。例えば、パブリッシャは、パブリッシャ自身の将来のコンテンツが、不正なメーカによって作成された製品上で再生されることを防止することができる。

【００７２】

プレイヤが、コンテンツに提供する情報の暗号化認証を提供する（例えば、認証されたプレイヤまたはメーカキーを使用することによって発行されたデジタル署名を含むことによ

る）場合、有効となり得るが、このような認証は大部分のデータにとって必須ではない。出力デバイス（高品質デジタルオーディオデータを要求するデジタルスピーカなど）、または、潜在的に信頼できないインターフェースを介して接続するシステムの他の部分にとって、信頼できるデバイスとして装う悪意のあるデバイスを検出し、かつ、回避することができるようにするため、暗号認証はより重要である。

【００７３】

（暗号化処理）

再生環境を説明する情報を提供することに加えて、例示的プレイヤは、さらに、コンテンツによって呼び出すことができる暗号化オペレーションを実装する。これらのオペレーションは暗号化オラクルのように振る舞うことができ、コンテンツが入力データ（例えば、６４ビット平文ブロック）を供給することを可能にし、暗号化計算の結果を返す。例示的実施形態では、暗号化計算への入力には、少なくとも１つのキー（その値は通常、コンテンツにとって未知であり、コンテンツによってアクセスできない）、および、コンテンツにより指定された入力データが含まれる。

【００７４】

以下は、再生環境を認証すること、コンテンツ復号化キーを導出することなど（ただし、これらに限定されない）に使用するために、コンテンツに提供することができる暗号化プリミティブの例（ただし、以下のものに限定されない）である。

【００７５】

ブロック暗号オラクル：このオラクルは、秘密キーを使用して入力メッセージを暗号化（または復号化）し、暗号文（または平文）の結果を作成する。

【００７６】

ハッシュ関数オラクル：入力メッセージが通常は秘密キーにより（例えば、ＨＭＡＣ－ＳＨＡなどのアルゴリズムを使用して）ハッシュされ、結果が作成される。

【００７７】

デジタル署名オラクル：秘密（公開）キーを使用して入力メッセージにデジタル署名が付けられ、結果が作成される。この関数は、秘密キーおよび（１つまたは複数の）秘密キーの証明書をコンテンツに提供することもできる。

【００７８】

乱数生成器：乱数生成器はコンテンツに予測不能な情報を提供して、例えば、オンライン接続におけるリプレイアタックを防止する上で使用することができる。

【００７９】

数学関数：基本的な数学演算を提供して、コンテンツがその計算プロセスを最適化することを助けることができる。例えば、コンテンツが最適化されたモジュラ乗算または指数関数を使用して、Ｒｉｖｅｓｔらの米国特許第４４０５８２９号明細書に記載のＲＳＡアルゴリズムを実行し、デジタル署名を作成かつ検証して、メッセージを暗号化および復号化することができる。

【００８０】

最適化された暗号化プリミティブ：標準暗号化アルゴリズムの最適化された実装は、パフォーマンスを向上させるのに役立つことができる。これらのオペレーションを使用して、データのブロックを復号化すること、またはハッシュすることを助けることができる。これらのデータのブロックには、インタープリタコード空間の領域、または、メディアからロードされたコンテンツのセクタが含まれるが、これらに限定されるものではない。

【００８１】

復号化制御：コンテンツが、再生が許可されると決定する場合、インタープリタコードはコンテンツ復号化モジュールを、コンテンツの各セグメントのための正しい復号化キーによって初期化することができる。加えて、インタープリタコードは、レンダリングされるべきであるか、またはスキップされるべきであるコンテンツの諸部分を指定することができる（例えば、再生中にリアルタイムウォータマーク挿入を可能にするため）。インタープリタと、メディアからのコンテンツストリーミングとの間の同期を確実にするために、

キー変更（またはスキップされた領域）をあらかじめ指定し、次いで、コンテンツ内の信号によってキー変更を引き起こすことができる。例えば、例示的実施形態では、コンテンツは、暗号文内で発見したときにキー変更を引き起こす６４ビット値、キー変更の後に続いてスキップすべき暗号文バイトの数、および、使用すべき新しいキー値を指定することができる。

【００８２】

キー管理：これらの機能によって、どのキーがプレイヤに知られるかをコンテンツは決定することができる。

【００８３】

オペレーションがランダムなパラメータまたは他のこのような可変データを組み込まない暗号化オラクルに関する例示的実施形態では、特定の入力に対して期待される結果をあらかじめ（例えば、コンテンツがマスタリングされるときに）計算することができるように、このシステムを構成することができる。したがって、パブリッシャはコンテンツをプログラムして、選択された入力をオラクルに提供し、次いで、期待された結果が得られることを検証することができる。許可された暗号化キーを欠く悪意のあるプレイヤは、正しいオラクル応答を計算することができないようになる。可能なオラクル入力の数は莫大である（例えば、１２８ビットのブロックサイズを有するブロック暗号を使用したオラクルでは、２＾１２８である）ので、攻撃者が可能な問い合わせのすべてに対する結果を事前計算すること、または格納することは、実際的に実現可能でない。

【００８４】

有効なプレイヤを検査することに加えて、暗号化オラクルを使用して、無効なプレイヤを識別することもできる。例えば、正当なプレイヤから抽出されたキーが無許可の目的のために使用されている場合、コンテンツをマスタリングし、これによって、失効したオラクルを含むプレイヤ上でコンテンツが再生されることを拒否することができる。コンテンツは、有効なキーがないと再生されないので、無許可のプレイヤは盗んだキー（stolen key）を含まなければならない。しかし、これらの盗んだキーを使用することによって、無許可のデバイスはそれらの状況を、危殆化を認識している新しいコンテンツに対して示すことになる。

【００８５】

オラクル結果を組み込むための、または、特定のオラクル問い合わせ応答が有効であるかどうかを調べるための幅広い種類の方法を使用することができる。最も単純な方法は、単に期待値と比較することである。これは、（少なくとも理論的には、）すべての比較が合致するかのように振る舞う、悪意をもって設計されたインタープリタによって回避することができるので、コンテンツは、失敗すると期待される「ダミー」比較、または、悪意のあるインタープリタを防止するように設計された他のそのようなテストを含むことができる。さらに、オラクル自体を使用して、コードを復号化することもできるし、または自己書き換えコードに影響を与えることもできる。例えば、オラクルへの入力を、所望のコードの暗号化されたバージョンとすることができる。したがって、それらの構成に応じて、このようなオラクルによって、コンテンツパブリッシャは、許可されたプレイヤまたはプレイヤのサブセットによってのみ復号化することができるコードをメディア上に含むことができ、それにより、このようなオラクルは、コンテンツコードを攻撃者となり得る者から離しておく助けとなる。オラクルを使用するもう１つの方法は、それらの出力を、暗号化キーとして、またはキーを導出するために、使用することである。次いで、例えば、コード、コンテンツ、他のキー、または他の任意のデータを復号化するために、これらのキーを使用することができる。この柔軟性のある復号化機能を使用して、幅広い種類のプロトコルおよびポリシをコンテンツ内に実装することができる。例えば、プレイヤが適切なキーの組合せを有する場合、ＦｉａｔおよびＮａｏｒの方法（A. Fiat and M. Naor, "Broadcast Encryption," Advances in Cryptology, Douglas Stinson, editor, p.480; Springer Verlag, 1993 参照）などのスキームを使用するように、コンテンツをプログラムすることができる。望まれる場合、Ｇｉｎｔｅｒらの米国特許第５９８２８９１号明細書

で説明されているシステムなどの高度なアクセス制御システムを実装することもできる（言うまでもなく、プレイヤが必要なユーザインターフェース、ネットワーク、データストレージおよび暗号化機能を提供するならば）。

【００８６】

コンテンツのマスタリングに関して、パブリッシャは、オラクル入力／出力のペアへのアクセスを有することから利点を得ることができる。オラクルがＲＳＡなどの非対称暗号化システムに対して秘密キーを使用する場合では、パブリッシャは単に公開キーを取得し、かつ、これを使用して、オラクルオペレーションとは逆のオペレーションを実行する。ブロック暗号を使用して構築された対称オラクルでは、プレイヤのメーカは、パブリッシャのために、各プレイヤ内に提供された対称オラクルの逆を計算することができる。例えば、プレイヤオラクルがブロック暗号を使用して、秘密キーの下で２５６ビットデータブロックを復号化する場合、メーカはパブリッシャに、対応する暗号化関数へのアクセスを提供することができる。逆オラクルへのアクセスは、オラクルが危殆化されることを可能にしないので、メーカは、（例えば、）ＳＳＬを使用して公的にアクセス可能なウェブサーバにアクセスすることによって、逆オラクル計算を提供することができる。メーカはまた、ランダムに選択されたオラクル入力からの出力をパブリッシャに提供することもできる。（メーカは、プレイヤ内に実装された実際のオラクル関数をパブリッシャに提供することができるが、正当なプレイヤをエミュレートする無許可のプレイヤを構築するために、これらの関数が悪用されるおそれがある。）

【００８７】

キーをプレイヤおよびメーカに割り当てるために使用される特定の方法は、特定の実施形態およびセキュリティ目標によって決まる。例えば、１つの例示的実施形態では、プレイヤには様々な対称暗号化オラクルキーが割り当てられ、これらのキーには、次のようなものが含まれる（ただし、以下に限定されるものではない）：このようなキーのより大きなグローバルプールから（擬似）ランダムに選択されたプレイヤ対称キー、メーカによって（擬似）ランダムに生成されたプレイヤ固有の対称キー、メーカ、プレイヤモデルなどに一意の対称キー、および／または、プレイヤが特定の特性を有していない（例えば、特定のメーカによって作成されなかった）ことを認証する対称キー。この例示的実施形態では、コンテンツは、サポートされたキーのリストを返す別の関数を呼び出すことによって、どのキーがプレイヤ内に実装されるかを識別することができる。プレイヤには、非対称キーを含めることもできる。例えば、例示的実施形態では、プレイヤは、プレイヤ固有の公開／秘密キーペア、メーカの秘密キーを使用してプレイヤ公開キーに署名することによってメーカにより発行されたプレイヤ証明書、メーカの公開キーの妥当性を検査するルートキー発行局（root key issuing authority）によって発行された証明書、プレイヤの安全なメモリ領域（下記参照）にアクセスする要求の妥当性を検査するために使用される公開キー、および／または、プレイヤファームウェアアップデートの妥当性を検査するために使用される公開キーを有する。

【００８８】

複数のプレイヤメーカを含むインフラストラクチャでは、１つまたは複数の中央管理組織にプレイヤ、メーカなどのためのキーを管理させることが有効である場合がある。中央管理者はまた、最小限のセキュリティ標準を実装すること、プレイヤが正確な情報をコンテンツコードに提供することを確実にすること、（メーカの製品が古いコンテンツを再生することができるように）キーを新しいメーカのために予約すること、危殆化されたキーを追跡すること、暗号化オラクルオペレーションをコンテンツパブリッシャのために実行することなどのために有効である場合もある。

【００８９】

（安全なメモリおよびカウンタ）

コンテンツにとって使用可能なメモリは通常、揮発性であり、コンテンツが実行されるたびに、コンテンツに「クリーン」な実行環境を提供する。しかし、一部の機能にとっては、コンテンツが複数の再生の間および複数のタイトルの間でデータを格納できることが有

効である。この必要性を満たすため、プレイヤは、複数の再生の間で状態を維持するための安全な不揮発性ストレージをコンテンツに提供することができる。このようなストレージは、許可されたインタープリタコードのみが不揮発性メモリコンテンツを読み取ることができること、または変更することができることを確実にするために、さらなるセキュリティ保護を必要とする可能性がある。不揮発性メモリのセキュリティを確実なものにすることは、このメモリを信頼して、例えば、後の支払請求のためにオフラインのペイパービュー視聴履歴を追跡することができるようにするために、パブリッシャにとって重要である。各メモリスロットのロックを解除するために、単にキーをメディア上に有するだけでは十分ではない。なぜならば、このキーがまもなく海賊行為者によって発見され、すべてのプレイヤのメモリスロットを危殆化させるようになるからである。したがって、一実施形態は、これらの安全な不揮発性メモリ領域にアクセスするコードの明示的な暗号化認証を提供する。

【００９０】

この実施形態では、プレイヤは複数の不揮発性メモリのブロックを含み、これらのブロックはデフォルト設定ではロックされる（すなわち、読み取りおよび書き込みが許可されない）。プレイヤはまた、メモリブロックのロックを解除する要求を認証するために使用される公開キーも含む。このメモリブロックへのアクセスを得るために、コンテンツは、メモリにアクセスすることを許可されるコードのブロック上で、デジタル署名を入力として受け取る関数を呼び出す。この署名は、プレイヤ内に埋め込まれた公開キーを使用することによって検証可能であり、ロックを解除すべきメモリブロック、および、ブロックの各部分内で許可されるアクセス特権（任意の読み取り、任意の書き込み、インクリメント、デクリメント、ゼロ化など）を指定する。インタープリタはデジタル署名を検証し、署名が有効である場合、メモリのロックを解除し、デジタル署名付きコードを実行する。以下に、時折（例えば、毎月）の監査によってオフラインのペイパーユースコンテンツの支払請求をする際に使用するための、このプロセスの一例を示す。

【００９１】

（ａ）パブリッシャＸは、プレイヤのメーカＹと、メーカＹのプレイヤの不揮発性メモリ内の４バイトカウンタを制御するための権利について協議する。

（ｂ）パブリッシャＸは、メモリコンテンツをチェックするインタープリタ用の関数を記述する。値が支出制限より小さい場合、関数はカウンタをインクリメントする。値が支出制限以上の場合、関数はパブリッシャとのインターネット接続を確立し、カウンタ値、乱数、および支払情報（クレジットカード番号、または、プレイヤ内に格納された他の資金源など）を含む支払要求を送信する。パブリッシャが、カウンタによって示された過去の購入に加えて現在の購入のための支払を受け入れる場合、パブリッシャは、カウンタを消去するための暗号化認証をプレイヤに送信し、この暗号化認証をプレイヤが検証し、（有効な場合）カウンタをゼロにする。プレイヤは、メモリを再ロックすること、および、成功または失敗を示すコードを返すことによって、終了する。

（ｃ）メーカＹは、パブリッシャＸのメモリ領域、アクセス特権などを識別するパラメータを含むメモリアップデートコードにデジタル署名する。

（ｄ）パブリッシャＸは、署名付きコードを含むコンテンツを作成し、これをユーザに配布する。

（ｅ）ユーザのプレイヤはコンテンツのロードを開始し、これはユーザに購入オプションを提示する。ユーザが購入を断る場合、再生は進行しない。

（ｆ）コンテンツは、ステップ（ｂ）で記述されたコードへのポインタ、および、ステップ（ｃ）で作成されたデジタル署名をもってメモリアンロック関数を呼び出す。

（ｇ）メモリアンロック関数は、ステップ（ｂ）で説明したように購入しようと試み、成功または失敗を報告する。

（ｈ）購入が成功した場合、コンテンツはユーザのために再生される。購入が失敗した場合、再生は終了する。

【００９２】

言うまでもなく、上述の安全なカウンタメカニズムを使用して、はるかに高度な購入メカニズムを利用することができる。コンテンツ内で何を実装することができるかについての実際の制限は、プレイヤの機能およびパブリッシャの創造性のみによって決まる。

【００９３】

様々なストレージ技術を、本明細書で開示するシステムおよび技術と共に使用することができる。これらのストレージ技術には、フラッシュメモリ、磁気ストレージ（例えば、ハードディスク）、バッテリバックＲＡＭ（battery-backed RAM）などが含まれるが、これらに限定されるものではない（不揮発性ストレージを提供するための、および、このようなストレージを暗号化するか、またはそうでない場合はそのようなストレージを保護するための、幅広い種類の方法が背景技術において知られている）。安全なストレージをプレイヤの外部に配置することができ、例えば、これらに限定されるものではないが、リムーバブルモジュール内（スマートカードなど）、アタッチされた出力ペリフェラル内（スピーカ、ディスプレイ、ホームネットワーク内のリモートデバイスなど）、コンピュータネットワークを介してリモートに安全なストレージを配置することができる。メモリブロック割り当てを、例えば使用可能な空間に基づいて提供し、確実にし（例えば、スロット番号により）、または、優先順位に基づいて割り振り／リサイクルすることができる。メモリスロットを消去または解放することは、報告されないペイパービュー記録を失う結果となる可能性があるので、スロットを上書きすることができる条件を指定する能力をコンテンツに与えることができる。複数のタイトルを同時に再生することができるが、１セットの不揮発性メモリスロットしか有さないプレイヤでは、ロッキングメカニズムは、コンテンツの一部分が、コンテンツの別の部分によって変更されているスロットにアクセスすることを確実にするよう要求される場合がある。

【００９４】

一実施形態では、プリペイドスマートカードが消費者によって購入され、プレイヤ上のスロットに挿入される。このカードは複数の追記型メモリスロットを含み、プレイヤは、このメモリスロットへペイパービューコンテンツタイトルに対応する識別子を書き込むことができる。識別子が書き込まれると、コンテンツ識別子は、カード内に実装された暗号化オラクル計算に組み込まれる。したがって、コンテンツは、再生可能となる前に正しいオラクルが存在することを検証することによって、購入が完了していることを検証することができる。

【００９５】

上述の、プレイヤ関数へのコールを認証するための全般的な手法は、安全なカウンタと共に使用することに限定されないことに留意されたい。例えば、同じ手法を使用して、許可されたパブリッシャにのみ使用可能な特別なプレイヤ機能へのアクセスを保護することができる。この手法は、さらに、本明細書で開示する技術およびシステムの他の態様とは異なる適用可能性を有する。なぜならば、この手法は、計算関数へのアクセスを保護するための、汎用的ではあるが非常に柔軟性のある方法を提供するためである。

【００９６】

（暗号および言語ベースのセキュリティ機能の比較）

セキュリティポリシを複数の異なる方法で実装することができる。暗号化保護は、失効したプレイヤ、または無許可のプレイヤがコンテンツを復号化するために必要な暗号化キーを欠くようなコンテンツの構築を可能にする。無許可のプレイヤは、キーを欠くコンテンツにアクセスすることはできない（言うまでもなく、優れた暗号が使用されるならば）。この手法は、コンテンツ所有者に、特定のデバイス上での再生をブロックする能力のみを提供するので、比較的柔軟性がない。（より高度な実施形態では、異なるキーセットを使用して、幾分より詳細な制御を提供することができる一方で、キーベースの制御は、より複雑なアクセス制御の課題を解決するために必要とされる柔軟性を欠く。）それにもかかわらず、特定のプレイヤが危殆化される場合、または、コンテンツを復号化する能力を有するために信頼できないと見なされる場合に対処する点で、これは、極めて有効である。

【００９７】

反対に、言語ベースの制御は、プレイヤが危殆化される（または、何らかの他の理由のためにまったく信頼できない）場合にそれほど有効ではないが、極めて高度なセキュリティポリシを実装することができる。前述のように、コンテンツは再生環境を解析することができ、暗号化オラクルへコールすることができ、結果が不満足であると見なされる場合、再生を拒否することができる。この手法は実質的に無制限の柔軟性を提供し、一般に真正に振る舞うが、一部のパブリッシャが所定のコンテンツにおいて防止することを望む可能性のあるオペレーション（非保護フォーマットへのリッピングなど）をサポートする場合のある、プレイヤ上での再生に関連するリスクを管理することに、理想的に適するようになる。攻撃者は、少なくとも理論的には、コンテンツの個々の部分を解析し、解読することができるが（特に、コンテンツのコードが不十分に書かれる場合）、これらの攻撃を一般化することはできず、暗号化オラクルを慎重に使用して、これらの攻撃に確実に対処することができる。さらに、本明細書で説明した復号化制御機能によって自らのコンテンツの海賊コピーを監視するパブリッシャは、危殆化されたデバイスを識別し、脆弱でない新しいコンテンツを作成することができる。

【００９８】

（進化）

長期にわたって安全であり続ける配布インフラストラクチャをコンテンツ所有者に提供することが望ましい。以前のコンテンツ保護システムはこの点でひどく失敗しており、実装者はコンテンツ所有者を新しいフォーマットへと勧誘するとき、最初はセキュリティに熱心である可能性が高いが、セキュリティレベルは、フォーマットの成功が確実になった後で著しく低下する傾向がある。様々な要因がこの低下を導くが、これらの要因には、攻撃するためのより多くの実装が入手可能であること（容易に解読される製品が販売されるようになる可能性が増す）、より保護されたコンテンツが入手可能になるために海賊行為に対する需要が増すこと、および、攻撃者の高度な知識が増すことが含まれる。本明細書で開示するシステムおよび技術の例示的実施形態を構成して、コンテンツ所有者が、メディアフォーマットが標準化された後でも自分のコンテンツがどのように保護されるかを指定し続けることを可能にすると同時に、攻撃が発見される場合にセキュリティが永遠に失われないように、実質的に無制限の更新可能性を可能にすることができる。

【００９９】

セキュリティポリシが静的でない場合、メーカは、有効なセキュリティを提供するための継続的な長期インセンティブを有する。例えば、コンテンツ所有者は、キーが危殆化されるデバイス上で、または、通常海賊行為のために使用される製品上で、再生をブロックする（または、高品質再生を防止する）能力を有することができる。結果として、従来のシステムとは異なり、製品メーカは、自社製品を可能な限りの低価格で提供するために競争するとき、セキュリティを犠牲にすることはできない。なぜならば、消費者もまた、堅牢なセキュリティを有する製品を探し出すようになるからであり、これらの製品は最良で最も信頼性の高い再生体験を提供するようになるからである。

【０１００】

善意のメーカでさえ、後にセキュリティ欠陥を有すると認められる製品を、偶然に製造する可能性がある。したがって、発明者らは、危殆化およびセキュリティ弱点に対処するために使用することができる様々な方法を開示する。例えば、プレイヤ暗号化キーおよびソフトウェアを、デジタル署名付きコードまたはキーアップデートを使用してアップデートすることができる。これらのアップデートを、キーアップデートを実行するソフトウェアを含むメディアの形態でプレイヤに配信することができる。例えば、正当なユーザのプレイヤが、以前の所有者がそのセキュリティを危殆化させたために、最終的に失効する場合、新しい所有者は製品のテクニカルサポートラインに電話して、新しいキーを取得することができる。（言うまでもなく、カスタマサービス担当者は、海賊行為者が無許可の目的のために新しいキーを要求する電話をかけさせないように、名前、住所、クレジットカード番号、電話番号、電子メールアドレス、ＩＰアドレスなど、何らかのユーザ情報を得ることを望む場合がある。）アップデートを、インターネット（または他のネットワーク接

続）、モデムコール、リモートコントロールまたはキーボードを介した入力などを介して配布することもできる。言うまでもなく、攻撃者がアップデートプロセスを使用して、危殆化されたキーを投入すること、またはそうでない場合はプレイヤを攻撃することができないように、アップデートは、可能なときは常に、暗号的に保護されるべきである。

【０１０１】

メーカが危殆化の影響を低減させることができるもう１つの方法は、スマートカードなどのリムーバブルなセキュリティモジュールを含めることである。スマートカードは、暗号化オラクルの一部またはすべて、ならびに、コンテンツに提供された他のセキュリティ関連機能を実装するようになる。危殆化が発生する場合、またはセキュリティ欠陥が発見される場合、プレイヤ全体を取り替えるか、またはアップグレードするのではなく、スマートカードを取り替えることができる。単にスマートカードスロットを提供するが、スマートカードがセキュリティ上の理由で必要になるようなときまでスマートカードを利用しなくても十分である場合があることに留意されたい。スマートカードが正当なプレイヤから取り除かれないように、および、悪意のあるプレイヤで使用されないようにするため、プレイヤおよび／またはカードが消費者に送られる前に、（例えば、これらに対称キーを共有させることにより）スマートカードを受信器へ暗号的にリンクさせることができる。

【０１０２】

（マスタリングおよびＤＲＭ）

コンテンツのマスタリングに伴うすべての新しいコストが、コンテンツ所有者にとって当然の関心事である。単純なセキュリティ手段が使用される場合、マスタリングプロセスに対する莫大な新しいコストを回避するために、本明細書で開示する技術およびシステムを利用することができる。複雑なセキュリティポリシを実装するコンテンツを開発するには明らかにより多くの開発およびテスト活動が必要となるが、この費用は完全に任意である。（他の保護システムは単にこの選択を排除し、すべてのコンテンツパブリッシャに強制的に同じセキュリティシステム、ポリシなどを使用させる。）

【０１０３】

言うまでもなく、パブリッシャはセキュリティシステム自体を開発する必要はない。なぜならば、本明細書に開示するシステムおよび技術は、サードパーティＤＲＭベンダがセキュリティモジュールおよびマスタリングシステムを提供することも可能にするからである。これらのベンダは、最良の機能、最良のセキュリティ、最低のコスト、最大の柔軟性、最良の使いやすさ、最良のパフォーマンス、最小のコードサイズ、最も広範な失効リストなどを提供することによって、パブリッシャのビジネスを求めて競争するようになる。本明細書で開示する技術およびシステムは、コンテンツ所有者がセキュリティについて自分自身で意思決定する能力を有するプラットフォームとしての機能を果たすことができる。

【０１０４】

（ウォータマーキングおよび危殆化トレーシング）

大部分の従来のウォータマーキング方法によって、マーク検出プロセスが、多数の幅広く利用された製品において標準化および実装される。この静的なアルゴリズムは不運にも重大なリスクをもたらす。なぜならば、検出アルゴリズムを知ることで、一般に、攻撃者がコンテンツの品質をひどく低下させることなく、ウォータマークを除去することが可能となるからである。例示的実施形態において、本明細書で開示するシステムおよび技術は、マークフォーマット、エンコードプロセスおよび検出プロセスがすべてパブリッシャによって選択されるために一般的なマーク除去攻撃の影響を受けにくい、オンザフライのウォータマーク挿入を含むことができる。

【０１０５】

１つの例示的実施形態では、パブリッシャ（または、より正確には、パブリッシャによって記述された制御プログラム）は、所定の情報を所定の出力コンテンツ内に埋め込むことを望む。第１のコンテンツ部分または第２の部分の一方を復号化し、出力することによって、この情報の各ビットをエンコードすることができる。これらの部分はメディア上の異なる暗号化された領域とすることができ、これらの部分を異なるキーにより暗号化するこ

とができる。コンテンツがマスタリングされるときにパブリッシャはこれらの部分の間の差異を選択することができ、これらの部分の間の差異は、知覚できないほど微妙なバリエーションからまったく類似しないものまで、どのようなものにもすることができる。２つの部分の間には所定の関係がないので、一方の部分（その部分用の復号化キーを含む）のみを知っている海賊行為者が他方の部分を決定するための方法はない。

【０１０６】

暗号化およびプログラムベースの制御を使用して、どの領域が復号化されるかを選択することができるので、攻撃者は、代替領域が何を含むか判断することができない。実際、攻撃者が代替領域が存在するかどうかを確認することすらできないように、例えば、（異なるプレイヤが異なるコードを使用するように）制御コードを暗号化することにより、および、どのプレイヤも復号化することができない、または非常に少数のプレイヤのみが復号化することができるダミー領域を含めることにより、コンテンツを設計することができる。

【０１０７】

１つの例示的実施形態では、すべてのプレイヤのサブセットのみがコンテンツの領域の各バージョンを復号化するために必要なキーを有するが、実質的にすべてのプレイヤがその領域の少なくとも１つのバージョンを復号化するために必要なキーを有するように、コンテンツは作成される。したがって、この領域の無許可のコピーを解析することによって、パブリッシャは攻撃者に関する情報を割り出すことができる。このことは、攻撃者が何とか（脆弱な）プログラムを解析し、複数の代替領域を復号化することができる場合にも当てはまり、これは、結果として生じる複数の領域の組合せがなお、どのバージョンが復号化されたかをパブリッシャに対して示すからであることに留意されたい。最終的に、ユーザが自分の識別（または自分のプレイヤの識別）をパブリッシャのアンチパイラシ実施専門家に明らかにすることを回避できる唯一の確実な方法は、まず第一に海賊行為に参加することをやめることである。

【０１０８】

マーク検出プロセスを標準化する必要はないので、この汎用マーキング手法は、従来のウォータマーキングとは異なる。この違いは非常に強化されたセキュリティを可能にし、実際には、このマーキングスキームに対する一般的な攻撃はないことを示すことができる。さらに、ウォータマーク付きビットは出力において差異を生じるので、これらのウォータマークは極めて堅牢となることができ、デジタル／アナログ変換、編集、フォーマット変換、悪意のある攻撃などを切り抜けて存続するように設計することができる。

【０１０９】

どのようにコンテンツマーキング機能を構成し、かつ使用するかを決定することは通常、パブリッシャによってなされる。一部のアーティストは、ウォータマーキング機能の使用を除いて、（変更がどんなに小さくても）自分の作品に対して変更を加える可能性のある技術を回避することを望む場合がある。他の場合には、所定のタイプのコンテンツは幅広く海賊行為を受け、マーキング機能を非常に積極的に使用するためのよい候補である。諸部分は通常、知覚できないほどの差異のみを有するように選択されるが、どの代替バージョンをエンコードするかを選択すること、可能性のある複数の出力バージョンの間でどのように選択するか、および、これらの部分のための復号化キーの管理は、コンテンツによって制御される。マーキング機能は、コンテンツと統合されたデータ処理命令によって制御されるので、この技術を他の機能のために使用することができる。これらの機能には、勝者のプレイヤが祝辞を出力するくじを実施すること、プレイヤが不十分なセキュリティを提供するユーザにセキュリティアラートを配信すること、および、ボーナスコンテンツをあるユーザに提供することが含まれるが、これらに限定されるものではない。

【０１１０】

言うまでもなく、他のウォータマーキングスキームもまた、本明細書で開示する技術およびシステムと共に使用することができる。例えば、従来のウォータマーク（ウォータマークに対してマーク検出アルゴリズムが標準化されるもの）を、コンテンツのコード、また

は、外部ウォータマーク埋め込み回路（コンテンツの制御下にあっても、なくてもよい）のいずれかによって、出力に埋め込むこともできる。同様に、（やはり、コンテンツのコード、または外部の検出器のいずれかによって）受信コンテンツ内のウォータマークを感知して、例えば、無許可コピーの作成または無許可コンテンツの導入の試行を検出することができる。どのウォータマークを埋め込むか選択すること、および、検出されたウォータマークにどのように応答するかを、プレイヤ内および／またはコンテンツ内に実装することができる。

【０１１１】

（移行パスの実施例：ＣＤ－オーディオ）

デジタルコンテンツの大部分は今日、非保護フォーマット、または最小限に保護されたフォーマットで配布される。例えば、ＣＤオーディオ規格はアンチコピー機能を含んでおらず、ＤＶＤビデオにおける保護スキームは幅広く解読されている。レガシーメディアプレイヤは十分なセキュリティをサポートしていないので、アップグレードまたは取り替えの必要がある。新しいセキュリティシステムの成功は、互換性のあるプレイヤの臨界数量（critical mass）を確立することにかかっている。

【０１１２】

本明細書で開示する技術およびシステムと、コピープロテクトされたＣＤを作成するための既存の方法とを組み合わせることによって、後方互換性のあるＣＤを作成することができる。このようなＣＤは非標準ＣＤフォーマッティングを利用して、大部分のオーディオＣＤプレイヤ上で正しく再生されるがコンピュータベースのリッピングソフトウェアを混乱させるディスクを作成するようになる。許可された（例えば、ライセンスを受けた）パーソナルコンピュータソフトウェアもまた、不正確に読み取られるか、またはそうでない場合はコンピュータを混乱させる部分を是正することによって、このディスクを再生することができる。したがって、再生は、（大部分の）レガシーオーディオプレイヤ上で可能になり、これは、これらのレガシーオーディオプレイヤが非標準（コピープロテクトされた）レッドブックオーディオ部分を再生することができるからであり、また、再生は、（例えば、ＣＤ上に含めることができ、または、インターネットを介してダウンロードすることができる）適切なプレイヤソフトウェアを有するパーソナルコンピュータ上で可能になる。既存のＣＤオーディオプレイヤとの後方互換性に対する長期サポートはさらなるセキュリティリスクをもたらす可能性があるが、新しい安全なフォーマットを再生することができるオーディオプレイヤの利用を奨励し、（最終的には）コンテンツを安全なフォーマットでのみ販売することができるようにすることは、長期戦略の一部として、有益となり得る。

【０１１３】

（実施例：高精細度ＤＶＤ）

現在のＤＶＤビデオプレイヤによって使用されるコピープロテクトシステムは、幅広く解読されている。何百万のＤＶＤプレイヤがすでに販売されており、新しいプロテクトシステムにアップグレード可能ではないので、これらのレガシーユーザのためのサポートを中止することなく、現在のＤＶＤフォーマットをアップグレードする直接的な方法はない。幸いにも、インストールベースのＤＶＤプレイヤは、「標準」の精細度のテレビ（例えば、ＮＴＳＣでは５２５本、ＰＡＬでは６２５本など）のみをサポートするが、高精細度テレビ（ＨＤＴＶ）フォーマットによって提供されるはるかに高品質の信号をサポートしないように設計される。レガシープレイヤはＨＤＴＶをサポートしないので、本明細書で開示する新しいセキュリティ機能を、ＨＤＴＶをサポートするＤＶＤ上に組み込むことができる。

【０１１４】

１つの例示的実施形態では、プレイヤは、（１つまたは複数のディスク用の機械化されたトレイからなる）ユーザがアクセスできるメディア入力を有し、このメディア入力はメディアをスピンドルへロードし、そこでメディアが回転され、レーザを使用して読み取られる。メディアから読み取られたデータはマイクロプロセッサベースの回路に提供され、こ

の回路はディスクエンコーディングを解析して、ディスクの容量、フォーマッティングタイプ、およびセキュリティ方法を決定する。ディスクが、レガシーセキュリティスキーム（ＣＳＳ）を使用するレガシー（低解像度）ＤＶＤである場合、このディスクは、背景技術で知られている方法を使用して再生される。ディスクが、本明細書で開示されるようなプログラマブルなセキュリティ方法を使用する高密度ＤＶＤである場合、コンテンツのセキュリティポリシのためのプログラムコード（データ処理命令）がディスクからロードされ、プレイヤによって実行される。プレイヤはまた、任意で、改良されたセキュリティを使用する低密度ＤＶＤ、ならびに、レガシーな保護方法を使用する高密度ＤＶＤをサポートすることもできる（しかし、新しいコンテンツのために幅広く解読されたセキュリティスキームを使用することは、一般にほとんど利点を提供しない）。ＤＶＤプレイヤからの出力の品質を、コンテンツによって制御することができる。例えば、コンテンツは、プレイヤおよび／またはＨＤＴＶ出力デバイスが十分なセキュリティを提供しない場合、より低い解像度の出力を出力するように選択することができる。この場合、コンテンツは、ＨＤＴＶ信号をより低い解像度へダウンコンバートする（例えば、このために特に設計された品質低下モジュール（degradation module）を使用する）ようプレイヤに命令すること、信号のより低い解像度部分を復号化するために必要とされるキーのみをプレイヤに供給する（および、より高い解像度部分のために必要とされるキーを与えないでおく）こと、または、より高い解像度バージョンとは分離して、メディア上でエンコードされるコンテンツの低解像度バージョンを出力するようプレイヤに命令することができる（これらは例であり、これらに限定されるものではない）。

【０１１５】

（インタープリタキテクチャ）

１つの例示的実施形態では、インタープリタ言語はＤＬＸアセンブリ言語に基づく。基本的なＤＬＸプロセッサアーキテクチャは背景技術においてよく知られている（例えば、Computer Architecture: A Quantitative Approach by Hennessy et al., Second Edition参照）。インタープリタのメモリ空間（１つの例示的実施形態では、８メガバイトのＲＡＭからなる）内で実行するコードは、このメモリおよびプロセッサのレジスタセットのみにアクセスすることができるようにサンドボックスされる。無効な命令（または、他のオペレーション）は、ＮＯＰ（すなわち、何もしない）として処理される場合があり、または例外を引き起こす場合がある。同様に、区域外のメモリアクセスは、例外を引き起こす場合があり、または（例えば、８メガバイトのアドレス空間からの３２ビット読み取りの場合、アドレスと、１６進の０ｘ００７ＦＦＦＦＣとでＡＮＤ演算を行い、区域外のアクセスをメモリの開始にラップアラウンドして、３２ビットのアラインメントを確実にすることによって）是正される場合がある。

【０１１６】

ＤＬＸ「トラップ」命令は、外部プロシージャコールへのアクセスを提供するために使用される。「トラップ」命令はプレイヤ内で、サンドボックスの外部に拡張する（すなわち、標準の命令がアクセス可能なレジスタおよびメモリを越える）ことができるオペレーションを呼び出す。このようなオペレーションの説明については、「再生環境への問い合わせおよび再生環境の制御」のセクションを参照されたい。

【０１１７】

専用ハードウェア実施形態では、ソフトウェアベースのインタープリタ／エミュレータではなく、ＤＬＸプロセッサのＡＳＩＣまたはＦＰＧＡ（または他のハードウェア）実装を使用することも可能であり、この場合、（例えば、）プロセッサがより高い特権レベルに入るように、（例えば、追加のアドレス行を使用可能にすることによって）アドレススペースを拡大してＲＯＭまたはＥＥＰＲＯＭ領域を含むように、リターンプログラムカウンタを格納するように、および、さらなる処理のために、拡大されたアドレススペース内の所定のアドレスにジャンプするように、「トラップ」命令を構成することができる。より高い特権レベルは、外部ペリフェラル（例えば、不揮発性メモリ、暗号化アクセラレータ、キー管理コンポーネント、光メディアドライブ、データネットワーク、衛星受信器など

）と対話する能力などのさらなる命令または機能をプロセッサコア内で使用可能にすることもできる。ハードウェア実装におけるメモリ保護機能には、アドレス行の数を制限すること（それにより、区域外アクセスを防止する）、または、背景技術において知られている他のメモリ保護方法を使用することを含めることができる。「トラップ」コールの完了時に、プロセッサは特権レベルを下げて、コンテンツコードの実行を進める。

【０１１８】

高密度光ディスク上に格納されて配布されたビデオを復号化するための例示的ＤＬＸ実装では、「トラップ」オペレーションは、コンテンツがデータをディスクから読み取ることを可能にするために提供される。データを物理的メディアからロードするため、コンテンツコードは通常、アドレス（例えば、光ディスク上のセクタ番号）、ＤＬＸメモリ空間内のデータのための宛先アドレス、ロードすべきデータの量、および、任意でデコードパラメータ（エラー訂正多項式／パラメータ、デコードキーなど）を指定する。コンテンツは、データをコードとして実行することも含めて、データを使用する任意の方式の処理ステップを実行することができる。光ドライブ、ハードドライブ、および他のデータソースはしばしばかなりの待ち時間を有するので（特に、新しいトラックにシークするなどのオペレーションを実行中であるとき）、別の「トラップ」オペレーションを使用して、必要とされると予想される領域を事前に指定し、データを要求し、保留中の要求の状況をチェックし、および／または、実際にデータをＤＬＸメモリ空間にロードすることができる。

【０１１９】

コンテンツコードは、高速暗号オペレーションをメモリ上で実行するためにトラップオペレーションを呼び出すこともできる。例えば、例示的バルク復号化「トラップ」オペレーションはＡＥＳ暗号化アルゴリズムを利用し、（ａ）プレイヤ内に（またはプレイヤがアクセスできる）格納された複数の秘密キーの中から選択するインデックス、（ｂ）暗号化されたキー、（ｃ）復号化すべきデータのためのＤＬＸメモリ空間内のアドレス、および（ｄ）復号化すべきデータの長さを、コンテンツが指定することができるようにする。トラップは（ａ）インデックス値によって識別されたキーを使用して、コンテンツからの受信した暗号化されたキーを復号化し、（ｂ）復号化されたキーをＡＥＳアルゴリズムと共にＥＣＢモードで使用して、指定された数のデータブロックを示されたアドレスで復号化する。インデックスによって選択することができるキー復号化キーには、プレイヤ内に格納されたキー（プレイヤ固有のキー、メーカキー、グループキー、メディアキーなどが含まれ（ただし、これらに限定されない）、任意で、これらを暗号機能および内部不揮発性メモリを備える耐タンパ性を有するチップに格納することができる）、外部デバイス（暗号モジュール、ディスクドライブ、リモートネットワークアクセス可能デバイス／サーバ、ディスプレイ、スピーカなど（ただし、これらに限定されない））内に格納されたキーを含めることができる。どのキーが使用可能であるかを判断するため、コンテンツコードは、プレイヤのタイプ、プレイヤのシリアルナンバ、プレイヤ内に含まれる（および、任意で、プレイヤのメーカまたは信頼できるパーティによってデジタル署名された）キーリスト／説明、ネットワークを介して得られたデータ、および、プレイヤまたは再生環境についての他の任意の入手可能なデータなどの情報を解析することができる（「再生環境への問い合わせおよび再生環境の制御」のセクションを参照）。コンテンツコードは、任意の方法の他の暗号化機能をプレイヤに提供することもできる。これらの暗号化機能には、暗号化、復号化、対称アルゴリズム（いずれかのオペレーションのモードによる、ストリーム暗号、ブロック暗号、ハッシングアルゴリズム、メッセージ認証コードなど）、公開キーアルゴリズム（署名、検証、暗号化、復号化、キー合意、ゼロ知識オペレーションなど）、キーおよび証明書管理などが含まれるが、これらに限定されるものではない。

【０１２０】

コンテンツコードは、追加の処理（または前処理）オペレーションを復号化結果に対して実行することができる。例えば、ＸＯＲ演算を実行して、ＥＣＢ復号化結果をＣＢＣ復号化結果に変換することができる。スクランブル解除ステップを適用して、敵対者がプレイヤから抽出したキーを使用してそのインタープリタコードを実行することなくコンテンツ

を復号化することを、防止することができる。適用することができるステップの例には、ビットをトグルすること、単純な２項演算を使用して変更を行うこと、ブロックを再配列すること、（例えば、ＭＰＥＧ－２または他のビデオ圧縮規格に不服のあるデータストリームを組み立てるために）オフセット／アドレスを修正または挿入すること、公開キー演算を適用すること（合成数を法とするモジュラ平方または立方など）、対称暗号化オペレーションを適用すること、および、内部チェックサムを更新することが含まれるが、これらに限定されるものではない。後処理ステップを使用して、例えば、コピーを特定のデバイスまでトレースできるように、フォレンシックウォータマーク（forensic watermark）を導入または修正することもできる。インタープリタを使用して、デコード／処理済みデータを実行し、復号化／再生コード自体の諸部分を暗号化された形式で配布することを可能にし、自己書き換えコードの使用などの幅広い種類のコード隠蔽および難読化（obfuscation）技術をコンテンツが利用することを可能にすることもできる。多機能データ、例えば、実行されるときに有用なタスクを実行し、有効な圧縮ビデオも表すデータを構築することも可能である。

【０１２１】

処理が完了し、データが出力される準備ができているとき、データをユーザに出力するために、コンテンツコードはさらなるプロシージャコール（例えば、ＤＬＸ「トラップ」オペレーション）をプレイヤ内で呼び出すことができる。このようなコールは例えば、データを１つまたは複数のビデオデコーダ（例えば、ＭＰＥＧ－２デコーダ）、オーディオデコーダ（例えば、ＭＰ３またはＡＣ－３デコーダ）、または、グラフィックスオーバーレイシステム（例えば、透過性／オーバーレイ機能を有し、さらに、ＧＬ、ＳｈｏｃｋＷａｖｅまたはＦｌａｓｈなどのイメージおよび／またはアニメーションエンジンをサポートする）に転送することができる。データは、適切な場合、変換（例えば、圧縮解除）され、次いで提示される。提示には、オーディオスピーカまたはビデオディスプレイなどの１つまたは複数の物理的に分離したデバイスにデータを転送することを含めることができる。

【０１２２】

別々のＡＰＩコールとしての復号化機能および圧縮解除／出力機能を有する実施形態は、コンテンツによるより大きな制御を可能にするという利点を有するが、コンテンツがメモリから読み取られ、メモリに書き込まれる必要のある回数を増すという、潜在的な欠点を有する。しかし、実際には、ＲＡＭは通常十分に高速であり、映画館品質の高精細度ビデオなどの非常に高ビットレートのコンテンツに対しても、追加の待ち時間を管理できる。高速の実装では、プレイヤコーデックは不必要である場合がある。なぜならば、圧縮解除をインタープリタコード内に実装することができるからである。プレイヤはまた、一命令複数データ並列処理機能（single-instruction, multiple-data parallel processing capability）を提供して（例えば、ｘ８６プロセッサで見られたＭＭＸ、ＳＳＥおよびＳＳＥ２命令にほぼ類似した、プロシージャコールを介してアクセス可能な一命令複数データ数学演算を提供することによる）、インタープリタコードを使用して実行されたコーデック、グラフィックス処理演算などのパフォーマンスを向上させることもできる。

【０１２３】

様々なインタープリタ実装戦略が可能である。一実施形態では、インタープリタは、標準のマイクロプロセッサ上で実行されるソフトウェア内に実装される。別の実施形態では、インタープリタは、フィールドプログラマブルゲートアレイなどの再構成可能ロジックを使用して実装される。別の実施形態では、専用ハードウェアデバイスがインタープリタの役割を実行する。３つの場合すべてにおいて、プロシージャコールを、ネイティブソフトウェア、ハードウェアアクセラレーション、および、外部デバイスまたはコンポーネントへのコールの任意の組合せを使用して実装することができる（ただし、これらに限定されるものではない）。

【０１２４】

（ネイティブコード）

プレイヤ独立のサンドボックスされたコードを解釈することに加えて、プレイヤは、コンテンツが実行および／または格納のためにネイティブコードを提供することを可能にすることもできる。キーまたは他の特権付きリソースへアクセスすることができるソフトウェアまたはロジックを受け入れることに先立って、プレイヤはコードの妥当性を検査する。例えば、プレイヤメーカまたは別の信頼できるパーティによって発行された有効なＲＳＡデジタル署名をコードが含むことを確認することによって、妥当性検査を行うことができる。現在ロードされているコンテンツによる実行のために、妥当性検査を通過したネイティブコードを揮発性メモリ内に格納することができ、または、他のタイトルが使用可能なプレイヤの不揮発性メモリ内に格納することができる。例えば、可能性のある他のタイトルへの悪影響を回避するため、現在ロードされているタイトルのみが使用できるように、プレイヤにおける表面的な特異な動作を是正するためのパッチ、または、パフォーマンス最適化を提供するためのパッチを揮発性メモリ内に格納することができる。反対に、セキュリティ脆弱性を是正するためのアップグレードは通常、プレイヤの不揮発性メモリ内に永続的に格納されるようになる。

【０１２５】

ネイティブコードは通常、単一のプレイヤプラットフォームまたはプレイヤアプリケーションに固有であり、インタープリタコードよりも移植性が低くなる。その利点は、インタープリタコードを使用して対処することができない必要性が生じる場合、ネイティブコードを使用することができることである。例えば、コンテンツは、正当なプレイヤと、無許可のエミュレータまたは「クローン」とを区別するための方法としてネイティブコードを使用し、攻撃者が製品の設計における主要なセキュリティ脆弱性を発見するたびに、影響を受ける可能性のある任意のデバイスを失効させる必要性を回避することができる。このような攻撃に対する防御として、製品ベンダは、エミュレートまたはリバースエンジニアリングすることが困難となる、ビルトインのネイティブコード機能またはインタープリタオペレーションを含むことができる。プレイヤ特有の攻撃の検出、またはそれに対する対処の助けとなるように設計された機能はベンダ独自および／またはプレイヤに特有である可能性があり、これは、それらの機能がプレイヤ特有の問題に応答してのみ作動されるようになるからであることに留意されたい。特有の手段には、単純な非文書化機能（simple undocumented feature）、タイミング依存ルーチン、ソフトウェアにおいてフルスピードでリバースエンジニアリングまたはエミュレートすることが困難となるように明示的に設計されるオペレーション（例えば、全体を参照することにより本明細書に組み込まれる、Ｋｏｃｈｅｒらの米国特許第６２８９４５５号明細書に記載の擬似非対称関数を参照）、および、完全暗号化オラクル（対称または非対称）を含めることができるが、これらに限定されるものではない。インタープリタが専用ハードウェア内に実装されるデバイスの場合には、「ネイティブ」コードおよびインタープリタコードは類似している、または等しい場合がある（しかし、これらのコードは異なる特権をもって動作する場合があり、この場合、プレイヤはより高い特権レベルでの実行を、特別に認証されたコードに制限することができる）。

【０１２６】

ネイティブコードを使用するコンテンツの例示的実施形態では、メディアは、解釈されるときにさらなる解釈可能コードをロードするインタープリタコードからなる初期ブート部分を含む。コンテンツコード（例えば、ブート部分によってロードされたコード）は次いで、プロシージャコールをプレイヤに発し、結果を解析して、プレイヤタイプを含む再生環境についての情報を決定するようになる。コンテンツは、例えば、プレイヤ（または他のコンポーネント）が、報告された再生環境特性に特有の（またはその特性を示す）キーを使用して暗号化オペレーションを実行するよう要求することによって、再生環境を検証することもできる。悪意のあるインタープリタが妥当性検査の結果を改ざんすることを防ぐため、このようなオペレーションの結果を後続のデコードおよび再生プロセスにおいて使用することができる。（例えば、プレイヤからの暗号化の結果を復号化キー計算に組み込むことによって、コンテンツは、特定のプレイヤタイプまたは再生環境属性を主張する

が対応する暗号化キーを欠く、悪意のあるプレイヤ上での再生を堅牢に防止することができる。結果を期待値と比較し、公開キースキームを使用して検証することなどもできる。）コンテンツは次いで、プレイヤが是正を必要とするセキュリティ欠陥（または、他の問題）をデフォルトで含むタイプであるかどうかを判断する。この判断を行う際、コンテンツコードは、プレイヤ内からの情報（例えば、コンテンツがアクセス可能な不揮発性メモリ内のデータ、プレイヤ内のクロックからの日付／時間情報など）、および／または、外部で得られた情報（例えば、アタッチされたデバイスに問い合わせること、または、ネットワークを介して通信することによる）を解析することができる。問題が特定される場合、（例えば、メディアから、または、インターネットなどのデータネットワークを介して）対応する対策が得られる。問題の性質に応じて、対策はインタープリタコードおよび／またはネイティブコードを含むことができる。ネイティブコード対策が必要である場合には、コンテンツは対策コードをデジタル署名付きデータとしてプレイヤに提供することができ、このデータは、コードが将来の使用のために（例えば、プロシージャコールを通じて）揮発性メモリ内でバッファリングされるべきであるか、（例えば、既存の不揮発性メモリ内のバグを修正するため）永続的に格納されるべきであるか、および／または、即時に実行されるべきであるかを示す命令を含む。ネイティブコードは、識別可能オペレーション（コンテンツ復号化プロセスと統合することができる暗号化計算など）を実行するように構成することができるので、ネイティブコードが実際にプレイヤによって実行されたことを、コンテンツコードに対して保証することができる。例えば、暗号化キーを上書きすることによって、悪意のあるプレイヤを無効にするように、ネイティブコードを構成することもできる。コンテンツはまた、コードおよびコードアップデート（ネイティブまたはインタープリタ）に問い合わせ、これらを解析し、デジタルインターフェースを介して接続されたディスプレイまたはスピーカなどの他のデバイスに配信することもできる。再生環境が受け入れ可能になると、コンテンツコードは次いで、例えば、データのチャンクをメディアからロードすること、フォレンシックウォータマークを挿入しながら復号化オペレーションを実行すること、および、復号化されたデータを圧縮解除および出力のために配信することにより、ユーザによって要求されたように再生を進行する。

【０１２７】

（標準化およびパフォーマンスの考慮事項）

再生環境を十分詳細に定義し、準拠プレイヤのメーカが（セキュリティポリシなどにしたがう）準拠コンテンツをそれらのメーカの製品で再生できることを確信することができる、標準を有することがしばしば必要である。このような標準は通常、インタープリタの基本命令セット（またはその均等物）、および要求されたプロシージャコールを指定する。デコードプロセスのリアルタイム部分に含まれる場合のある任意の命令およびプロシージャコールについてのパフォーマンス要件を定義することも必要である場合がある。（パフォーマンス要件は一般に、起動、シャットダウン、および、他の非リアルタイムオペレーション中にのみ実行されるオペレーションでは、それほど重要ではない。）

【０１２８】

例えば、例示的仕様は、準拠インタープリタが毎秒最低８００万「タイムユニット」（ＴＵ）を実行できることを要求する場合があり、この場合、標準の低レベルインタープリタオペレーションはそれぞれ最大１ＴＵを要する場合があり、乗算および除算演算はそれぞれ４ＴＵを要する場合がある。プレイヤにより供給されたプロシージャ（例えば、ＤＬＸ「トラップ」）へのコールのためのパフォーマンス要件は、要求されるオペレーションによって決まる。例えば、ＡＥＳ暗号化オペレーションは、最大１００ＴＵに加えて、１２ＴＵをブロック暗号計算毎に要する場合がある。ワードアラインされたメモリコピーは、最大５０ＴＵに加えて、１ＴＵを１６バイト（またはその部分）毎に要する場合がある。メディア読み取り要求のバッファリング、または、バッファリングされた要求の完了状況のチェックは、最大１００ＴＵを要する場合がある。読み取り要求は一度に１つ、提示された順序で、しかし、他の処理と並列で実行され、最大１００００ＴＵに加えて、１５００ＴＵを２０４８バイトセクタ毎に要する場合がある。不連続の読み取りは、追加（２０

０００＋６４００００＊セクタ内のシーク距離／メディア毎の最大セクタ）のＴＵをシークオーバーヘッドとして要する場合がある。完了された読み取りからデータをロードすること（すなわち、データをドライブのバッファからインタープリタのメモリ空間に転送すること）は、最大１００ＴＵに加えて、１２８ＴＵを、転送された２０４８バイトセクタ毎に要する場合がある。データをコーデックまたは他の出力に転送することは、最大１００ＴＵに加えて、１ＴＵを、転送された１６バイト（またはその一部）毎に要する場合がある。言うまでもなく、これらのタイミング値は例示のために提供され、特定のパフォーマンスメトリックスは、システムの要件によって決まるようになる。（例えば、命令シーケンスの合計計算時間を指定する）より複雑なパフォーマンス指令を指定して、プレイヤ実装者により大きな柔軟性を提供し、または、コンテンツ作成者によりよいパフォーマンス保証を提供することもできる。

【０１２９】

実際には、多数の実装は、最低限のパフォーマンスよりかなり高速に動作するようになる。このことは単に、データが必要とされる前に準備ができるようになることを意味する。例えば、通常のインタープリタ命令につき４０クロックサイクルを要するソフトウェアベースのインタープリタは、約５０００万ＴＵ／秒を２ＧＨｚマイクロプロセッサ上で実行するようになる。同様に、２５ＭＨｚで実行し、命令につき２クロックサイクルで実行する専用ハードウェア実装もまた、毎秒８００万ＴＵよりも大幅に多く実行するようになる。

【０１３０】

標準の開発者は、システムの簡素化と、パフォーマンスとの間のトレードオフに直面することに留意されたい。具体的には、ソフトウェアベースの実装では、プロシージャコールにおいて実行されるオペレーションは一般に、インタープリタコードにおける同じオペレーションよりもかなり高速に動作すると仮定することができる。他方では、これらのオペレーションは通常、前もって定義されなければならず、また通常は、それらのパフォーマンスの仮定においてエントリ／終了オーバーヘッドを含まなければならない。それにもかかわらず、メモリコピー、検索、大数の計算、および暗号化計算などの一般的なオペレーションのためのプロシージャコールは、大きなパフォーマンス利点を提供することができる。代替のパフォーマンス向上手法は、インタープリタがコンテンツコードを実行前または実行中に解析して、最適化すること（例えば、ネイティブコードに変換すること）ができる領域を識別することである。コンテンツコードには、最適化のための適切な候補である領域をインタープリタに知らせるための「ヒント」を含めることもできる。「ヒント」の手法は、（高速になる傾向があるが、複雑なオペレーションを実行するという難点を有する）ハードウェア実装が、（例えば、ＮＯＰとして処理することによって）ヒントを無視し、後続のコードを通常に処理することができるという利点を有する。（低速になる傾向があるが、より高速なネイティブコード機能を有する）ソフトウェア実装はヒントを使用して、インタープリタコードを、機能的に互換性のあるネイティブコードルーチンと置換することができる。望まれる場合、パフォーマンス基準は共通の構造のためのプレイヤパフォーマンス要件を指定することができる。プレイヤはまた、コンテンツが、最低限のパフォーマンスを満たすよう常に保証されるインタープリタモード（例えば、リアルタイムタスク用）と、よりよい平均の場合のパフォーマンスを有するモード（例えば、非リアルタイムタスク用）とから選択することを可能にすることもできる。

【０１３１】

コンテンツを作成しているとき、コンテンツ開発者は、自分が書いたソフトウェアが、基準によって指定された最低限のパフォーマンスを満たすことを検証する必要がある。タイミング準拠を検証するための専門のテストシステムは、コンテンツコードが実行するときのその最悪の場合のパフォーマンス特性を作表する。このシステムは、実行される各サブオペレーションに対してプレイヤが取ることのできる最大許容時間を作表しながら、再生プロセスをエミュレートすることによって動作する。再生プロセスが低速すぎる場合（例えば、測定された最悪の場合のプレイヤパフォーマンスが、コーデックに供給されている

データにおけるタイムスタンプに遅れを取る場合、または、コーデックがデータ「不足」になる可能性がある場合）、ツールはメディア作成者に通知することができ、メディア作成者は次いで問題を是正することができる。オーサリングツールは同じ手法を使用して、それらの出力が確実に再生されるようにすることができる。

【０１３２】

（不揮発性メモリの保護）

前述のように、プレイヤデバイスは、コンテンツによる使用のために不揮発性（ＮＶ）ストレージ機能をコンテンツに提供することができる。コンテンツを作成するエンティティの数が大きい場合があるので（場合により、少数のアーティスト、学生、ホームユーザなど、ならびに大規模のスタジオが含まれる）、あるコンテンツが不十分に、または悪意をもってさえ書かれる場合があるという仮定の下で、コンテンツおよびプレイヤがＮＶストレージ使用において制限を実施することは、有効である場合がある。結果として、プレイヤは、各タイトルがＮＶメモリを予約し、格納されたデータを読み取り、修正し、上書きする能力を制限することを望む場合がある。「安全なメモリおよびカウンタ」というタイトルのセクションでは、デジタル署名を使用して不揮発性メモリにアクセスするコードの妥当性を検査することを、説明する。しかし、状況によっては、コンテンツタイトルが不揮発性メモリ領域を割り振りおよび／または制御することを可能にしながら、不揮発性メモリセキュリティが集中認証局なしに動作することが望ましい場合がある（例えば、政治的および／または技術的な理由のため）。

【０１３３】

以下のセクションでは、集中署名局を必要とすることなく、不揮発性メモリへの安全なアクセスをコンテンツに提供する、プレイヤの例示的実施形態を説明する。図４を参照すると、例示的メモリマネージャは１２８キロバイトのフラッシュメモリへのアクセスを制御し、この１２８キロバイトは、各々が２５６バイトである５１１のスロット、および、追加のデータのための２５６バイトに分割される。スロット０［４１０］、スロット１［４１２］、スロット２［４１４］、および、その後の各スロット４４０は、最初にスロットを割り振ったタイトルのメディアＩＤを識別する１２８ビットの作成者メディアＩＤ４２０、最後にスロットを変更したタイトルのメディアＩＤを識別する１２８ビットの最終アップデートメディアＩＤ４２２、スロットが最後にアップデートされたときを識別する４０ビットの最終アップデートシーケンスカウンタ４２４、スロットが上書きされる必要がある場合にスロットのランクを示す８ビットのスロット優先順位値４２６、許可されたコードによってのみアクセス可能である１６バイトのプライベートデータ４２８、スロットにアクセスすることを許可されるコードの１６０ビットのハッシュ４３０、および、メインスロットペイロードデータ４３２を含む。プレイヤが工場で初期化されるとき、これらの値をすべてゼロに初期化して、スロットが空であることを示すことができる。

【０１３４】

１２８キロバイトの不揮発性メモリの最後の２５６バイトを使用して、秘密プレイヤキー４４４、上書きされている優先順位６スロットの数を含むカウンタ４４５、上書きされている優先順位７スロットの数を含むカウンタ４４６、および、高部分４４７および低部分４４８として格納されたスロット書き込みカウンタを含む値を格納する。スロット書き込みカウンタは頻繁に更新され、一部の不揮発性メモリ技術は、非常に多くの書き込みサイクル後に消耗するので、このカウンタは、いずれかの特定のメモリセルが更新される回数を制限する形態で格納される。低部分における１０２４ビットのうち１０２３が満たされていない限り、１ビットを低部分４４８に設定することによって、このカウンタはインクリメントされ、低部分における１０２４ビットのうち１０２３が満たされる場合、高部分４４９がインクリメントされ、低部分４４８のすべての１０２４ビットは消去される。高部分４４７を１０２４で乗算し、次いで低部分４４９に設定されたビットの数を加算することによって、カウンタ値は読み取られる。プレイヤが工場で初期化されるとき、これらの値をすべてゼロに初期化することができるが、ただし、プレイヤキーはシークレット（擬似）ランダム値により初期化されるべきである。

【０１３５】
プレイヤはまた、揮発性メモリ（例えば、従来のＲＡＭ）内に格納することができる、複数の値も維持する。これらには、メディアキー４５０、メディア識別子４５２、どのスロット（すなわち、番号０ないし５１０）が現在アタッチされているかを示す値４５６、および、現在のタイトルによってこれまでに書かれたＮＶメモリ内の最高優先順位スロットを示す値が含まれる。タイトルが初期化されるとき（例えば、メディアが挿入されるか、またはプレイヤがリセットされるとき）、アタッチされたスロット識別子４５４、アタッチされたスロット優先順位４５６、および、最大作成優先順位４５８がリセットされる。メディアキー４５０は好ましくは、コンシューマ記録可能メディア上でコンシューマ記録可能デバイスにより書き込み可能ではないメディアの一部分からロードされる。メディア識別子４５２は、次いで、背景技術においてよく知られているセキュアハッシュアルゴリズム（ＳＨＡ－１）などの一方向の暗号化変換を適用することによって、メディアキーから導出される。追加の保証を提供するために、メディアは暗号化署名認証メディアキー４５０および／またはメディア識別子４５２を運ぶことができ、次いで、これらをプレイヤおよび／またはコンテンツコードによって認証することができる。代替実施形態は、他の値（メディアを製造またはプレスした施設の識別、および／または、代わりにメディアの特定の部分に特有の識別子など）を含むことができ、必ずしも識別子とキーの間の固定関係を有する必要はない。

【０１３６】
一般に、コンテンツコードは、各スロットのプライベートデータ４２８を除く不揮発性メモリコンテンツに対する実質的に自由な読み取りアクセスが認められる可能性がある。この読み取りアクセスは、コンテンツがスロット番号を指定してコンテンツを検索することを可能にするプロシージャコール（例えば、ＤＬＸ「トラップ」オペレーション）を使用して、実施することができる。要求されたスロットが現在アタッチされていない（すなわち、アタッチされたスロット識別子４５４によって識別されない）場合、スロットプライベートデータ４２８は返されない（例えば、ゼロがこれらの位置に返される）。

【０１３７】
例示的実施形態では、コンテンツには、不揮発性メモリスロットから読み取り、不揮発性メモリスロットへのアクセス（アタッチ）を要求し、および、不揮発性メモリスロットの修正を行うための、以下の基本的オペレーションが提供される。

【０１３８】
ＳｌｏｔＲｅａｄ：このプロシージャコールは、指定されたスロットのコンテンツを、コンテンツコードによってアクセス可能なメモリ空間に読み込む。このプロシージャの入力パラメータには、スロット番号、および、結果が格納されるべきコンテンツのメモリ内のポインタが含まれる。スロットコンテンツ全体は、プライベートデータフィールド４２８を除いて返され、プライベートデータフィールド４２８は通常、読み取り結果においてゼロにされる。指定されたスロット番号が（－１）である場合、アタッチされたスロット識別子４５４によって識別されたスロットが読み取られ、完全なコンテンツ（プライベートデータ４２８を含む）が検索および格納される。オペレーションの戻り値は、読み取られたスロット番号（例えば、スロット（－１）が指定された場合、アタッチされたスロット識別子４５４）、または、なぜ要求が失敗したかを示すエラーコードのいずれかを含む整数である。

【０１３９】
ＳｌｏｔＡｔｔａｃｈ：このプロシージャコールは、指定されたスロットへの特権付きアクセスを要求するために使用される。このようなアクセスを付与することに先立って、要求を行うコードが認証される。プロシージャの入力パラメータは、スロット番号、コードの長さ、および要求された優先順位レベルを指定する。プロシージャは、アクセスを付与されるべきコードの開始アドレス（例えば、ＳｌｏｔＡｔｔａｃｈオペレーションを呼び出す命令の後に続く、コンテンツのメモリ内のアドレス）を割り出す。このアドレスおよび指定された長さを使用して、プロシージャは次いで、（例えば、ＳＨＡ－１を使用して

）コードの暗号化ハッシュを計算する。ハッシュ結果が、スロット内に格納された許可ハッシュ４３０の値に合致しない場合、または、（例えば、図５に関連して後述するように）要求された優先順位が無効であると判定される場合、スロットゼロがアタッチされ（すなわち、アタッチされたスロット識別子４５４およびアタッチされたスロット優先順位４５６がゼロに設定され）、エラーが返される。そうでない場合、要求されたスロット番号は、現在アタッチされているスロットになる（すなわち、アタッチされたスロット識別子４５４が、要求されたスロット番号に設定され、アタッチされたスロット優先順位４５６が設定される）。特別な場合として、呼び出し元コードは（－１）のスロット番号を指定して、新しいスロットが割り振られるよう要求することができる。この場合、プレイヤは（例えば、図５に関連して後述するように）要求された優先順位の妥当性を検査し、優先順位が無効である場合、エラーを返す。そうでない場合、プレイヤは（後述のように）上書きすべきスロットを選択し、（例えば、作成者メディアＩＤ４２０を現在のメディアＩＤ４５２に設定し、他のスロットフィールドをゼロにし、書き込みカウンタ４４７／４４８をインクリメントすることにより）これを消去して、（例えば、アタッチされたスロット識別子４５４をこのスロットの番号に設定し、優先順位４５６を要求された優先順位に設定することにより）このスロットにアタッチし、最大作成優先順位４５８を、その現在値および要求された優先順位４５６のうち大きい方に設定する。潜在的に信頼できないコードを不意に実行させる可能性のある他の機能、または、割り込みをインタープリタがサポートする場合、スロットがアタッチされる間に悪意のあるコードが開始されることを回避するために、これらは無効にされるべきである。戻り値は、アタッチされたスロット番号４５４であり、または、オペレーションが失敗した場合（例えば、コードハッシュのミスマッチのため、または要求された優先順位が無効であるため）、エラーコードである。

【０１４０】

ＳｌｏｔＷｒｉｔｅ：このプロシージャコールは、現在アタッチされているスロットにデータを書き込む。このプロシージャの入力パラメータは、スロットプライベートデータ４２８、認証ハッシュ４３０、およびペイロード４３２のための新しいコンテンツを示し、これらは、他のスロットフィールドのための更新値と共に書き込まれる。（詳細には、作成者ＩＤ４２０は不変のまま残され、最終アップデートメディアＩＤ４２２は現在のメディアＩＤ４５２に設定され、最終アップデートシーケンスカウンタ４２４はスロット書き込みカウンタ４４７／４４８に設定され、スロット優先順位４２６は、アタッチされたスロット優先順位４５６に設定される。）スロット書き込みに先立って、スロット書き込みカウンタは、その低部分４４８、および、必要な場合は高部分４４７を更新することによって、インクリメントされる。有限のライフタイム（例えば、多数のフラッシュおよびＥＥＰＲＯＭメモリは、１００万書き込みサイクルと見積もられる）を有する不揮発性メモリを使用するプレイヤは、電源投入／メディア挿入後に、（例えば、１２８を超える）非常に多くの書き込みが実行されている場合、書き込みを拒否することができる。書き込みオペレーションは、アタッチされたスロット識別子４５４およびアタッチされたスロット優先順位４５６を共にゼロにリセットする。戻り値は、書き込みが成功したかどうかを示すステータスコードである。

【０１４１】

スロット優先順位管理サポートは、複数の相反し得る目的のバランスを取るために提供され、これらの目的には、（ａ）コンテンツは、適度に必要とする可能性のある量の不揮発性メモリへアクセスすべきであること、（ｂ）コンテンツは、その不揮発性メモリが不意に上書きされないことを確実にすべきであること、（ｃ）あるタイトルが過度に多くの不揮発性メモリを予約して、それにより他のタイトルが予約できないようにすることが、可能であるべきではないこと、（ｄ）空のスロットが不揮発性メモリ内で使用できない場合、新しいコンテンツに何らかの不揮発性ストレージを提供するために、めったに使用されないスロットがリサイクルされるべきであること、および、（ｅ）タイトルが、正当なスロットをリサイクルさせるように設計された多数の要求を提示することができないようにするべきであることが含まれる。一般に、より高い優先順位を有するスロットは、プレイ

ヤがスロットを使い切っている場合、上書きされる可能性が低い。例示的実施形態では、優先順位ルールは、各メディアタイトルが多くても１つの最高優先順位スロット（優先順位７）を確実に有することができるように設計される。加えて、メディアは、各挿入または各プレイヤパワーサイクルにおいて、２以上の優先順位を有するスロットを１つだけ作成することが認められる。コンテンツは、７より大きい優先順位を有するスロットを作成することはできないが、プレイヤが製造されるときに予約されたスロットは、より高い優先順位レベルを有することができる。

【０１４２】

図５は、アタッチされたスロットのための要求された優先順位が受け入れられるかどうかの妥当性を検査するための例示的プロセスを示している。スロットがアタッチされるとき、または作成されるとき（上記のＳｌｏｔＡｔｔａｃｈを参照）、コンテンツは、要求された優先順位値を指定する。ステップ５００で、プレイヤは、新しいスロットが割り振られている（例えば、スロット番号が－１として指定される）かどうかをチェックする。そうである場合、ステップ５１０で、プレイヤは、要求された優先順位がスロットの既存の優先順位４２６を超えるかどうかをチェックし、超える場合は、要求された優先順位が大きすぎることを示すエラーを返す。要求された優先順位がスロットの既存の優先順位４２６を超えない場合は、ステップ５２０で、アタッチメント要求が認められて、要求された優先順位を使用して進行する。ステップ５００で、要求は新しいスロットを割り振ることであると判断する場合、プレイヤはステップ５３０で、要求された優先順位が７以下であるかどうかをチェックし、そうでない場合、エラーを返す。要求された優先順位が７より大きい場合、ステップ５４０で、プレイヤは、要求された優先順位が２より大きいかどうかをチェックし、そうでない場合、要求された優先順位が、スロットを割り振るために有効であるとして受け入れられる。要求された優先順位が２より大きい場合、ステップ５５０で、プレイヤは、最大作成優先順位４５８をチェックして、メディアが挿入された後、またはプレイヤがリセットされた後、２より大きい優先順位を有する任意のスロットがすでに作成されているかどうかを判定し、そうである場合、要求された優先順位が拒否される。２より大きい優先順位を有するスロットが作成されていない場合、ステップ５６０で、プレイヤは、要求された優先順位が７であるかどうかをチェックし、そうでない場合、要求された優先順位は、スロットを割り振るために有効であるとして受け入れられる。要求された優先順位が７である場合、ステップ５７０で、プレイヤは、その格納された作成者メディアＩＤ４２０が現在のメディアＩＤ４５２に等しい優先順位７スロットがすでに存在しているかどうかをチェックし、そうである場合、無効であるとして、要求された優先順位を拒否する。優先順位７スロットが存在していない場合、要求された優先順位は、スロットを割り振るために受け入れられる。

【０１４３】

新しいスロットが割り振られる（すなわち、コンテンツがＳｌｏｔＡｔｔａｃｈをスロット－１をもって呼び出す）とき、プレイヤは、上書きするために最も低い優先順位４２６を有するスロットを選択する。工場で、空のスロットが、最も低いと考えられる優先順位（ゼロ）で初期化され、したがって、一般に最初に使用されるようになる。複数のスロットが最も低い優先順位を共通して有する場合、最も低い書き込みシーケンスカウンタを有するスロットが選択される。上書きするために優先順位６または７を有するスロットが選択される場合、対応するカウンタ（優先順位６上書きカウント４４５、または、優先順位７上書きカウント４４６）がインクリメントされる。代替として、非常に多数のスロット用のスペースを有するプレイヤでは、要求が高優先順位スロットを上書きすることを必要とする場合、あるいは、高優先順位スロットを上書きすることを伴う場合、単に失敗する可能性がある。

【０１４４】

工場において、認証ハッシュ４３２およびゼロでないスロット優先順位用の所定の値をもって一部のスロットを初期化することができる。スロットのための機能性および／またはセキュリティ要件は、ハッシュされるコードによって決まる。例えば、所定の認証ハッシ

ュを定式化するために使用されるコードは、以下のステップを実行するように構成することができる：（ａ）すべてのレジスタ（例えば、スタックポインタなど）を「安全な」値に初期化し、（ｂ）ＲＳＡ署名値を、認証されたコード領域の外部の所定のアドレスからロードし、（ｃ）ＳｌｏｔＡｔｔａｃｈオペレーションによって認証された領域に埋め込まれた公開キーを使用して、ＲＳＡ署名がインタープリタのメモリの領域にわたって有効な署名であるかどうかを判定し、および、（ｄ）ＲＳＡ署名が無効である場合、（例えば、一般にアクセス可能なスロット０にアタッチすることにより）現在のスロットからデタッチするが、ＲＳＡ署名が有効である場合は、デジタル署名された領域の最初のアドレスにジャンプする。デジタル署名付きコードが特定のプレイヤ上でのみ再生されるよう意図される場合、このコードは、例えば、プレイヤの識別（または、他の属性）をチェックし、および／または、プレイヤキーを使用して、コードの一部を実行する前に復号化するように、構成することができる。

【０１４５】

先の段落の実施例では、（例えば、「安全なメモリおよびカウンタ」というタイトルのセクションで説明したように、）ハッシュベースのスロット妥当性検査スキームを使用して、非対称署名妥当性検査を実施するための方法を例示している。この手法は、スロットが将来の使用のために予約されることを可能にするが、将来の使用は、プレイヤが製造されるとき、または標準が定義されるときに、指定される必要はない。公開キーベースのコード妥当性検査システムを使用して、ハッシュベースの手法を実装するコードに署名することも可能である。

【０１４６】

単一のスロットを複数の目的に使用することが可能である。例えば、複数のコードセグメントにデジタル署名して、上述のものなどの検証プロセスを通過することができる。これらの各コードセグメントを、スロットの異なる部分を修正するように、および、完了時に適切にデタッチするように構成することができる。

【０１４７】

スロットのプライベートデータフィールド４２８も注目すべきである。なぜならば、このフィールドにより、コードがスロットコンテンツを暗号化できるようになるからである。いずれのコンテンツもメインスロットペイロードを読み取ることができるが、許可されているコードしかプライベートデータ４２８を読み取ることができない（例えば、ＳｌｏｔＡｔｔａｃｈプロセスを使用することにより）。したがって、許可されたコードは、プライベートデータ４２８をキーとして（または、キーを導出するために）使用して、スロットコンテンツを暗号化および復号化することができる。このようにして、スロットに格納されたデータのプライバシを確実にすることができる。望まれる場合、コンテンツは、認証コードまたはデジタル署名をスロットコンテンツ上に配置することもできる。このような署名を、コンテンツコードによって（任意で、プレイヤキーを使用することにより）生成することができ、プレイヤによって生成することができ、または、外部のパーティまたはデバイス（タイトルのパブリッシャまたはプレイヤのメーカなど）によって生成することができる。次いで、すべてのスロットを検索し、（例えば）所定のセキュリティチェックが必要であること、または、特定のメディアＩＤが失効されていることを示す、デジタル署名付きの（または、そうでない場合は認証された）値を検索するように、後続のタイトルを構成することができる。

【０１４８】

複数のタイトルがスロットベースの機能を共有できる。例えば、最新の既知の日付をコンテンツに提供するデータマネージャを実装することができる。この機能を使用する各タイトルは、それがマスタリングされた現在の日付のデジタル署名付き表現、および、スロットの認証ハッシュ４３０に合致する所定のコードを含むようになる。スロットを管理するためのコードは、（ａ）コンテンツと共に含まれた日付におけるデジタル署名をチェックし、無効である場合、スロットからデタッチして、停止し、（ｂ）現在のスロットコンテンツを読み取り、（ｃ）タイトルからの新たに検証された日付を、スロットのペイロード

４３２における日付と比較し、（ｄ）タイトルの日付の方が遅い場合、タイトルの日付をスロットコンテンツ内に入れ、ＳｌｏｔＷｒｉｔｅを呼び出して、新しい日付を不揮発性メモリに格納するが、スロットの残りを不変のまま残し、（ｅ）スロットからデタッチし、および、（ｆ）ＲＡＭからスロットプライベートデータ（ならびに、他の不必要な値）を消去するようになる。このオペレーションは、エラーまたは後の日付値（すなわち、タイトルと共に含まれた認証日、および、以前にスロット内に格納された日付のうちの遅い方）のいずれかを返す。損なわれた日付が遠い将来であると誤って解釈されないように、コードは、任意でデジタル署名を日付と共に格納することができる。加えて、（例えば、プライベートデータフィールド４２８の値をキーとして使用することにより）データを暗号化して格納し、格納された日付値への読み取りアクセスを、現在の日付を含むタイトルに制限することもできる。プレイヤにおける日付値を、例えば、コンテンツが使用して、セキュリティアップデートが必要とされるかどうか、オンラインセキュリティチェックが予定されているかどうか、ペイパービュー購入記録が提示される予定であるかどうかなどを判断することができる。

【０１４９】

一部の実施形態（例えば、リアルタイムネットワーク機能を提供するものなど）では、ネットワークサポートが可能であるとき、プレイヤに不揮発性メモリへのアクセスを制限させることが有利である場合がある。この制限は、例えば、悪意のあるコンテンツがデータを不揮発性ストレージから抽出し、かつ、ネットワークを介して送信することを防止することによって、ユーザのプライバシを確実にするのに役立つことができる。特定の実施例として、ネットワークアクセス機能を使用するタイトルは、データをスロットから読み取ること（または、ユーザ識別情報を含むと考えられる特定のスロット、もしくは他のタイトルによって作成された所定のスロットを読み取ること）を防止することができる。プレイヤは、タイトルがそれらのネットワークアクセス特権を打ち切ること、および、（例えば、ネットワークアクセスがもはや必要とされなくなった後に、フォレンシックウォータマーク用の値を得るため）より幅広いスロットアクセスを得ることを可能にすることもできる。フォレンシックウォータマークに埋め込まれた情報は同一のプライバシ上の懸念を生じさせることはなく、これは、このデータをコンテンツのコピーからのみ復旧できるためであることに留意されたい。

【０１５０】

上記の特定の不揮発性メモリ管理の説明は、特定の実施形態を例示するよう意図されている。言うまでもなく、多数の変形形態が可能である。例えば、特定のスロット構成、スロット管理オペレーション、および優先順位管理手法は、例示のために提供されている。メモリを固定サイズのスロットに割り振る代わりに、他のメモリ管理手法を使用することができる（スロットに基づかない割り振り方法の使用を含む）。他のアクセス制御メカニズム（メディア識別子に基づかないものなど）も使用することができる。スロット用の不揮発性メモリの総量は可変とすることができる（または、ハードディスクまたは他の大容量ストレージ技術の場合のように、事実上無制限である）。別々の不揮発性ストレージ機能を、メディア／ディスク挿入履歴を追跡するために含めることができる。不揮発性メモリ（または、コンテンツによってアクセス可能なメモリなどの他のメモリ）のコンテンツを暗号化および／または認証し、無許可の読み取り／変更を防止するために、暗号技術を使用することが有効である。ソフトウェア実装においては、様々なコード、アドレス、データ、およびアルゴリズム難読化技術を使用して、プレイヤキーの抽出（または改ざん）を防止することができる。任意の形のデータをスロット内に格納することもでき、これらのデータには、ペイパービュー購入記録、何らかの種類のカウンタ（例えば、購入カウンタ、クレジット／デビットまたは他のバランスカウンタ、障害カウンタ、メディア／ディスク挿入カウンタなど）、オンラインまたは電話ベースのセキュリティチェックの結果、視聴記録／履歴、コード（インタープリタまたはネイティブ）、失効データ（プレイヤ、ペリフェラル用など）、他のタイトルへの信号、リベートまたはディスカウントのための資格を評価するために使用された情報、トランザクションおよび再生履歴データ、デジタル

署名、ならびにキーが含まれるが、これらに限定されるものではない。

【０１５１】

（メディア失効）

１つの注目すべきシナリオは、プレイヤは信頼できるが、メディアが海賊行為を受けている状況である。これは専門的な海賊行為者にとって通常の状況である。なぜならば、海賊行為者は一般に、正当なコピーにできる限り近い「製品」を提供しようとするからである。海賊行為者が、プレイヤが物理的にオリジナルと区別することができない正当なメディアのコピーを作成する方法を開発する場合、すべてのオフラインメディア再生デバイスはおそらく、この性質の攻撃の影響を受けやすい。メディアのコピーが困難な機能（光ディスク上のトラックのウォブル特性の測定など）を使用して、正確な複製をより困難にすることはできるが、決意の固い海賊行為者はなお、コピーを作成する方法を発見する場合がある。同様に、フォレンシックウォータマーキングは、コンテンツ所有者が、（数ある中でも）以前のコンテンツに対して海賊行為を行うために使用された機器を特定することを可能にするが、海賊行為を受けたコンテンツが再生されることを防止しない。

【０１５２】

海賊行為者がメディアの無許可のコピーを作成するというリスクに対処するため、コンテンツ所有者は、コピー毎に一意の識別子を正当なメディア上に配置することができる。光ディスクでは、工場において一意に書き込むことができる領域内（一部の既存のスタンプ付き光ディスク上で発見されるバーストカッティングエリア（burst cutting area）など）に、または、記録可能ディスク部分上に、（例えば、記録可能ＣＤおよびＤＶＤ、磁気ディスクなどを作成するために現在使用されているものなどのストレージ技術を利用して、）このデータを配置することができる。一意に書き込み可能な領域は、少量の情報（例えば、数バイトから数百バイト）のみを保持するだけでよい。具体的には、この領域は、記録されたシリアルナンバ、ユーザ情報（名前、電話番号など）、暗号化キーなど、ならびに、これらの値を認証するデジタル署名を保持することができる。コンシューマ記録可能メディアでは、製造される空のメディアの各々について、一意のメディアシリアルナンバ（および、任意で、関連付けられたデジタル証明書）を記録することができる。

【０１５３】

メディアが再生のために挿入されるときに、その一意のデータの認証性が検証されるように、メディア上のコンテンツコードを構成することができる。メディアタイプおよび記録されたデータに応じて、この検証プロセスは通常、そのメディアに固有のデジタル署名を検証することを含む。敵対者がこの検証チェックを回避することを防止するために、難読化コードを使用して検証を実行することができ、チェックを複数回（例えば、再生中の様々な時間に）実行することができ、検証結果を後続の復号化プロセスと統合することができる。一意のデータが妥当でない場合、コンテンツコードは通常、再生を拒否する。一意のデータが有効である場合、コンテンツコードは、（ａ）メディアが正当である、または（ｂ）そのメディアが、提供された一意のデータを有した正当なメディアの１つからコピーされた、という確信を得る。

【０１５４】

次に、一意のメディアが有効であるか、失効しているかを判断するため、コンテンツは、現在挿入されているメディアの失効状況を示すデータフィールドに関するプレイヤの不揮発性メモリ領域をチェックする。合致する失効情報が発見されない場合、このメディアは有効であると推定される。不注意による失効または悪意のある失効を防止するために、デジタル署名または他の認証データ（暗号ハッシュに対するプレイメージ（pre-image）など）をチェックすることによって、失効データを認証することができる。プレイヤの不揮発性メモリ内のデータは、存在するならば、どのようなアクションが推奨されるかを示すこともできる。アクションには、（ａ）コピーが違法であることをユーザに通知すること、（ｂ）低下した解像度で再生が進行することを可能にすること、（ｃ）再生を完全に防止すること、または、（ｄ）（例えば、電話番号に電話してアンロックコードを入力することにより、または、インターネットを介してサーバと対話することにより）ユーザがコ

ンテンツを正当に購入できるようにすることなどがある。不揮発性メモリは、例えば、望まれる場合、コンテンツが「失効されない」ようにすることができるように、以前の失効メッセージを無効にすることもできる。複数の相反する状況インジケータが存在する場合、シリアルナンバまたは日付を使用して、どれが最新であるかを判定することができる。

【０１５５】

場合によっては、事前に登録されたものなどの特に許可されたデバイス上でのみ再生可能であるメディアを作成することが有用である。この場合、失効させる代わりに、一意のメディア情報を使用して、メディアアクティベーション（media activation）を可能にすることができる。例えば、事前のレビューコピー（review copy）および製造前のコピーは一般に、すべてのプレイヤ上で再生可能である必要はない（そして、そうであるべきではない）。書き込み可能部分に記録されたデータは、各々のメディアの許可された受信者に固有の復号化キーまたは他の情報を含むことができる。一意に記録された領域もまた、名前、電子メールアドレス、口座番号、または、特定の受信者（例えば、フォレンシックウォータマーキングの目的、インタラクティブ機能などのため）または受信者の特性（例えば、視聴のプリファレンス、許可データ、グループ識別子、郵便番号など）を特定する他のデータを含むことができる。メディアの販売中または販売後に、例えば、万引きされたメディアが再生されることを防止する（、それにより、メディアを棚の上に陳列する店における万引きのリスクを減らす）アクティベーションステップとして、これらのフィールドに書き込むこともできる。バックエンドシステムは、決してアクティベートされないメディアについて商店に払い戻すか、または、アクティベートされるときにメディアについて商店に請求することができる。この機能のもう１つの使用は、公式リリース日より前に「ロックされた」メディアを配布し、次いで、再生が許可されるべきであるとき、書き込み可能領域上に再生を可能にするキーを書き込むことである。例えば、小売施設、出荷施設、またはセキュリティエージェント（例えば、プレス施設が完全には信頼できない場合）などのメディアを製造したもの以外の団体または施設が情報を記録できる。エンドユーザデバイスがメディアの諸部分に書き込むことができる場合、メディアが再生またはロック解除されるとき、データ（許可キー、プレイヤ識別子など）を記録することも可能である。メディア上の書き込み可能部分を使用して、例えば、有料でロック解除を行い、次いで複数のプレイヤ上で再生することができる、「ボーナス」マテリアルなどの機能を実装することができる。

【０１５６】

オフラインプレイヤでは、失効通知が通常、後続のタイトル上で配信されるようになる。例えば、新しいタイトルは、失効しているメディアを識別する（認証データを伴う）シリアルナンバのリストを含むことができる。プレイヤが十分な不揮発性ストレージ（例えば、ハードドライブ）を有する場合、プレイヤが失効リスト全体を格納できる。プレイヤが十分な不揮発性ストレージを有さない場合、失効データをプレイヤの挿入履歴および／または不揮発性メモリスロットに対してチェックして、プレイヤによって再生されているいずれかのメディアが失効されるかどうかを判断することができる。失効される場合、対応する失効データはプレイヤの不揮発性メモリ内に格納される。この手法により、海賊行為を受けた「クローン」メディアは、初めて挿入されるときには再生されるが、海賊行為を受けたメディアを失効させるメディアが挿入されるとき、失効される（または、そうでない場合は「フラグが立てられる」）。一般に、メディア失効は有用である。なぜならば、メディア失効により、消費者にとって、海賊行為を受けたメディアは正当なメディアよりも魅力的ではなくなるからである。

【０１５７】

メディアアクティベーション／失効を使用して、様々なプロモーションおよびセキュリティ機能を実装することができる。例えば、（例えば、ワイドスクリーン、パンスキャン、ディレクターズカットなどの）異なるバージョンを含む複数のメディアをもって映画を販売することができる。人々がこのようなメディアを別々に販売またはレンタルすることを防止するために、それらのコンテンツコードは、１つまたは複数の他のメディアがプレイ

ヤの不揮発性メモリ内で表現されることを検証することができる。任意で、他のメディアが、最近（例えば、所定の時間内に、または、スロット書き込みカウンタ４４７／４４８の所定のインクリメント回数以内に）挿入されていることを要求することもできる。別のオプションとして、メディアは、（例えば、他のメディアからのキー値をロードすることによって、）ユーザが再生中に別のメディアを挿入することを要求することもできる。ユーザが所定の組合せの他のメディアを再生している場合、ユーザにボーナスマテリアルへのアクセスが与えられる、プロモーションを作成することができる。言うまでもなく、再生決定は、プレイヤおよび／またはプレイヤ環境の特性などの他の情報にもリンクされる。

【０１５８】

オンライン機能を有するプレイヤは、現在挿入されているタイトル、ならびに、プレイヤのメディア挿入履歴および／または不揮発性メモリスロットで表現された他のタイトルの失効状況をチェックすることができる。このチェックをコンテンツコード内で実施することができ、または、プレイヤによって実行することができる。オンラインチェックを使用して、複数のプレイヤが同時に単一のメディアを再生している（例えば、メディアが海賊行為を受けていることを示す）、または、過剰な数のプレイヤが特定のメディアを使用している（例えば、それが使用許諾契約に違反してレンタルされていることを示す）場合を検出することもできる。

【０１５９】

海賊行為者は、コンテンツのコードを変更することによって、失効チェックを回避するように試みる場合がある。この場合、コードの後続の部分（例えば、再生中に、後に実行されるコード）は、例えば、チェックを繰り返すことによって、または、検証コードを含むメディアの部分をリロードおよび検証することによって、変更を検出することができる。失効チェック結果を復号化プロセスと統合することもできる。

【０１６０】

言うまでもなく、メディア失効手法の変形形態を使用することができる。例えば、プレイヤは、失効したメディアを識別するように構成された、デジタル署名付きのインタープリタコードを格納することができる。これらのコードスニペット（code snippet）を、メディアが挿入されるたびに実行して、新たに挿入されたタイトルが失効されるかどうかを判断することができる。タイトルは失効チェックコードを（好ましくは、プレイヤによって検証されるようになる対応するデジタル署名と共に）格納することができるようになり、プレイヤは、以降のメディアをチェックするためにこのコードを保持するようになる。例えば、プレイヤの不揮発性メモリ内に、失効した光ディスクのセクタ番号およびハッシュ値のテーブルを格納することにより、メディア失効チェックを、プレイヤのＲＯＭ内のコードによって実行することもできる。メディアが書き込み可能である場合、コンテンツはまた、失効データをメディア自体に格納し、得ることもできる（または、ユーザプリファレンス、ユーザ情報など、他の任意の種類および目的のデータをメディア上に格納することもできる）。

【０１６１】

メディアはまた、記録デバイスについての失効データを保持するように使用される場合もある。例えば、コンシューマ記録デバイスが、識別データをそれらの記録上に配置するように構成される場合、プレイヤデバイスは、失効したレコーダの記録を保持することができる。これらの記録は、メディア上に保持された失効データによってアップデートされる場合がある。記録可能メディアをまた、失効したレコーダを識別する情報と共に製造して、失効したレコーダがメディアに書き込むことを防止することもできる。失効関連データフィールドを暗号的に認証して、例えば、正当なデバイスの悪意のある、または不注意の失効を防止することができる。例えば、記録デバイスによって配置された識別データには、メディアシリアルナンバについてのレコーダデジタル証明書およびデジタル署名が含まれる場合がある。記録可能メディア上に配置された識別データを（例えば、サードパーティエージェントの公開キーにより）暗号化して、ユーザのプライバシを保護することがで

きる。閾値暗号技術を使用して、例えば、複数のメディアが記録デバイスを識別することを必要とすることもできる。（本明細書で開示した技術の他の態様においては、閾値暗号技術を、例えば、ある量のコピーされたマテリアルがマークを復旧するために必要とされることを確実にするために、フォレンシックマークと共に使用することもできることに留意されたい。）

【０１６２】

メディア失効に関する１つの他の有用性は、使用可能である他の手法に加えて、ちょっとした海賊行為に対する追加の抑止力を提供することである。フォレンシックウォータマークに埋め込まれたデータは、デバイスによって再生された以前のメディアを識別することができる。実装選択に応じて、そのＩＤが海賊コピーのフォレンシックウォータマークから決定された他のメディアを失効させること、そのシリアルナンバが海賊行為を受けたマテリアルを配布するために使用されたものに近い記録可能メディアを失効させること、問題のあるメディアからの再生に先立って追加の許可ステップを必要とすることなどが有用である場合がある。

【０１６３】

（各種特徴および機能）

セキュリティオーバーライドキー：セキュリティオーバーライドキーを知るプレイヤは、一部またはすべてのセキュリティチェックを回避することができるように、コンテンツを設計することができ、コンテンツの一部分またはすべてへのアクセスを可能にすることができる。このようなキーをサポートすることによって、コンテンツ所有者は作品の限られた部分に対するアクセスを付与することができる（例えば、権限者が、コンテンツ所有者が批評家にフレーム毎のビデオに対する「公正使用」アクセスを付与するよう要求した場合）。これらのオーバーライドキーを使用して、例えば、保護機能が不十分に設計された（再生可能性の問題を生じる）場合に、コンテンツを保護された形態から「リリース」することもできる。必要に応じて、これらのキーを、サードパーティに預託することができる（または、サードパーティの公開キーにより暗号化されたメディア上に格納することができる）。オーバーライドキーを、著作権が切れるときにリリースされるようにスケジュールして、例えば、コンテンツがパブリックドメインに入ることをアンチパイラシメカニズムが防止する可能性があるという、懸念に対処することもできる。

【０１６４】

マルチラウンドコリュージョン解析（Multiple Round Collusion Analysis）：状況によっては、決意の固い敵対者は、フォレンシックウォータマークの復旧を防止しようとして、多数のデバイスからの出力を結合する可能性がある。敵対者が非常に多数のデバイスを危殆化させており、害を及ぼすデバイスをコンテンツが直接識別することができない場合、複数のコンテンツリリースから収集されたフォレンシック情報を結合することが可能である。例えば、第１のタイトルから収集された情報は、敵対者によって使用されている可能性のあるデバイスの範囲を狭める場合があるが、害を及ぼすデバイスのすべてを一意に識別しない場合がある。第２のタイトルがマスタリングされるとき、この知識を使用して、攻撃者および／またはそれらの機器についてさらなる情報を提供するフォレンシックウォータマークを作成することができる。このプロセスは、敵対者が一意に識別されるまで、繰り返すことができる。

【０１６５】

悪意のあるインタープリタへの対策：悪意のあるプレイヤは、コンテンツ内のセキュリティ関連コードを認識しようと試みるように設計される可能性がある。例えば、悪意のあるプレイヤは、どこでＲＳＡ署名検証オペレーションがコンテンツによって実行されているかを特定し、結果を変更して、例えば、無効なＲＳＡ署名が有効であるように見えるようにしようと試みる可能性がある。このようなプレイヤが作成される場合、コンテンツ所有者は新しいコンテンツを作成して、この識別を回避するように設計した異なるＲＳＡ署名計算ルーチンを使用することができる。使用できるオペレーションの例には、信頼できないオペレーションを回避するようにコードを書き換えること、コードを難読化すること、

メッセージ隠し、試算をチェックして攻撃を検出すること、および、中間物および／または結果を他の暗号オペレーション（復号化ステップなど）と統合することが含まれるが、これらに限定されるものではない。

【０１６６】

インタラクティブ性：仮想マシン／インタープリタは、非セキュリティタスク、ならびにセキュリティの目的のためである場合がある。例えば、コンテンツコードを使用して、メニュー、テキスト、グラフィックス、アニメーション、ユーザインターフェース要素などをユーザに表示することができる。同様に、コンテンツはユーザコマンドまたは応答を受信することができる。これらのコマンドまたは応答には、マウス入力（例えば、移動、クリック）、ボタン押下（キーボードまたはリモートコントロール入力など）、ライトペン入力、およびジョイスティックアクションが含まれるが、これらに限定されるものではない。ローカルプレイヤについての情報（カメラ入力、マイクロフォン入力、ユーザの体の位置の変化など）を収集および使用して、例えば、再生を制御することもできる。プレイヤは、ユーザインターフェース実装を支援するための機能を提供することもできる。これらの機能には、ダイアログボックスの表示、表示ウィンドウの管理、音声認識の実施、ユーザプリファレンスの管理などが含まれるが、これらに限定されるものではない。海賊行為に対する抑止力として、インタラクティブなユーザ機能を実装するコードをセキュリティコードと結合し、敵対者があるコードを他のコードから容易に分離することができないようにすることができる。

【０１６７】

（ネットワークを介したコンテンツへのアクセスおよびコンテンツの交換）

大部分のコンシューマビデオおよびオーディオコンテンツは現在、光メディアで配布されているが、ストリーミングダウンロードの人気は時間とともに増すと予測される。物理的メディアの代わりに、または物理的メディアに加えて、ストリーミングまたはローカルでキャッシュされたコンテンツをサポートするように、本明細書で提示したセキュリティ手段を再設計することができる。プレイヤに物理的に位置するメディアからデータをロードする代わりに、コンテンツコードおよびデータはネットワークを介して検索される。例えば、メディアからデータセクタを要求するプロシージャコールを呼び出す代わりに、コンテンツは、ネットワークを介してデータを要求するプロシージャコールを呼び出すようになる。受動的なメディアとは異なり、リモートサーバはそれ自体が処理能力を有することができ、例えば、（例えば、プレイヤに暗号化オペレーションを実行させるために）リモートサーバが要求をコンテンツに送信し、結果の妥当性を検査することが可能となる。ネットワークを介して交換されたデータを保護するためのセキュリティプロトコル（このようなプロトコルにはＳＳＬが含まれるが、これに限定されない）を、コンテンツコード内またはプレイヤ（または他のコンポーネント）内に実装することができる。

【０１６８】

単純なサーバ実装は、妥当性検査されたユーザ（例えば、コンテンツにアクセスするための代金を支払っているユーザ）からの要求を受信し、対応するデータをサーバ自体のローカルストレージから読み取り、結果を配信する。より高度なサーバはデータをリアルタイムで選択および／または修正して、例えば、フォレンシックウォータマークを埋め込み、他のサーバと対話することができる。サーバはまた、リモートプレイヤについての、またはそれに代わる（例えば、プレイヤベースの不揮発性メモリの代替としての）情報を格納し、エンドユーザに合わせてカスタマイズされたセキュリティコードを配信し、リアルタイム失効チェックを実行し、セキュリティアップグレードをコンテンツに自動的に挿入し、インターネット／ウェブプロキシ機能を提供し、他のサービスをコンテンツコードに供給することもできる。例えば、例示的トランザクションには以下のステップが含まれる。（ａ）コンテンツサーバが購入要求をエンドユーザのプレイヤから受信するステップ、（ｂ）コンテンツサーバが支払いを検証するステップ、（ｃ）コンテンツサーバが、ユーザのプレイヤの機能的な、および／またはセキュリティのプロパティを解析するように構成された解釈可能コードの部分を送信するステップ、（ｄ）ユーザのプレイヤがインタープ

リタコードを実行し、そのプロパティについての情報を返すステップ、（ｅ）コンテンツサーバが応答を解析し、（インタープリタコードおよび／またはネイティブコードを含み、カスタム生成される場合がある）セキュリティ検証ロジックをユーザのプレイヤに送信するステップ、（ｆ）ユーザのプレイヤが検証ロジックを処理し、応答をサーバに返すステップ、（ｇ）サーバが応答の妥当性を検査するステップ、（ｈ）コンテンツサーバが、暗号化されたデジタルコンテンツ（例えば、オーディオ、ビデオおよび／またはイメージ）をユーザのプレイヤに送信（例えば、ストリーミング）するステップ、および、（ｉ）ユーザのプレイヤがコンテンツを復号化するステップ（ただし、復号化プロセスの正しいオペレーションは、正しいキー、または、セキュリティ検証ロジックからの結果を必要とする場合がある）。

【０１６９】

サーバプロセス自体は、インタープリタコードによって制御することができ、任意で、プレイヤ側と同じインタープリタキテクチャ（例えば、ＤＬＸ）を使用することができる。これは、サーバの物理的ハードウェアアーキテクチャを意識せずにコンテンツを作成することができるという利点を有する。ホームネットワーク環境では、これは特に魅力的なモデルである。なぜならば、セキュリティおよびデコード「インテリジェンス」はサーバに残るが、認証されたローカルデバイスへコンテンツをストリーミングすることができるからである。同様に、様々な異なるインターネットサービスを介してストリーミングされるコンテンツに関して、サーバ側のインタープリタは、コンテンツが一度作成されると、互換性のあるサービスのいずれからもストリーミングされることを可能にすることができる。

【０１７０】

場合によって、受信デバイスもまた、受信デバイス自体のセキュリティの決定を行う能力を有することがあるが、これは、受信側デバイスがコンテンツをキャッシュして、後にそのコンテンツを、識別が初期転送中に知られていない後続のデバイスへ送信することを望む場合などである。この場合、初期転送には、受信デバイスによってそのセキュリティ関連の決定を行う際に使用するための、解釈可能および／またはネイティブで実行可能なコードが含まれる場合がある。すべてのデバイスが同じインタープリタまたはプログラマブル技術をサポートする必要はない。なぜならば、送信側デバイスによって実行されるコードは必ずしも受信側デバイスによるコードと同じである必要はないからである。

【０１７１】

状況によっては、複数のサーバおよび／またはコンテンツ転送が含まれる場合がある。例えば、コンテンツは、（例えば）プレイヤのメーカおよびコンテンツ所有者の両方によって運営されるサーバを含む、複数のエンティティから取得したセキュリティコードを含む場合がある。一部のパーソナルコンピュータ環境では、複数のインタープリタを使用することも有用である場合がある。例えば、インタープリタは、メディアインターフェース（例えば、光ディスクドライブ）、オペレーティングシステム、アプリケーションソフトウェア（例えば、プレイヤ）、出力デバイス（例えば、増幅器）などに含まれる場合がある。代替として、または加えて、暗号化オラクルもまたコンポーネントにおいて提供される場合がある。

【０１７２】

（プレイヤオペレーションの呼び出しおよび表記法）

プレイヤデバイスは、（通常はソフトウェア実装の）インタープリタ（仮想マシン）を提供し、これによりコンテンツコードは様々な個々のオペレーションを実行することができる。このようなオペレーションには、仮想レジスタを操作すること、および、コンテンツコードに属するメモリにアクセスすることが含まれる。加えて、コンテンツコードはＴＲＡＰ（プレイヤ内に実装された外部オペレーション）を呼び出すこともできる。ＴＲＡＰを使用して、コンテンツコードがそれ自体のリソースを操作することによっては直接実行することができないオペレーション（ペリフェラルへのアクセスなど）を実行すること、または、そうでない場合はコンテンツコードによって保証されることが可能である、より

高いパフォーマンスを提供することができる。

【０１７３】

このセクションで使用される表記法は、Ｃ／Ｃ＋＋プログラマに良く知られているように設計される。型ＵＩＮＴ８、ＵＩＮＴ３２およびＵＩＮＴ６４は８ビット、３２ビットおよび６４ビット符号なし整数をそれぞれ示すために使用される。例示的実施形態では、各プロトタイプは一連の３２ビット値に対応し、これは、適切なＴＲＡＰオペレーションを呼び出すことに先立ってスタック上にプッシュされるべきである。スタック上のすべての値は３２ビットのサイズなので、３２ビットより小さいパラメータは３２ビットに拡張され、より大きい値は、複数の３２ビット値を使用して格納される。各ＴＲＡＰオペレーションにおいて関連する３２ビット値の実際の数は、ゼロである場合がある（パラメータリストがｖｏｉｄである場合）。パラメータの数もまた可変である場合があり、この場合、プロトタイプは「．．．」で終了する。配列は「ｔｙｐｅ ｎａｍｅ［ｓｉｚｅ］」として示され、例えば、「ＵＩＮＴ３２ ｔｅｓｔ［１６］」は、１６個の３２ビットワードの配列を表す。ポインタは「ｔｙｐｅ ＊ｎａｍｅ」として示され、例えば、１つまたは複数の３２ビット符号なし整数に対するｔｅｓｔＰｔｒという名前のポインタは、「ＵＩＮＴ３２ ＊ｔｅｓｔＰｔｒ」となる。

【０１７４】

スタック自体を単に、コンテンツコードによって、および、各ＴＲＡＰを実装するプレイヤの部分によって、アクセス可能なメモリ内の領域にすることができる。例えば、コンテンツコードによってアクセス可能なエミュレートされたレジスタをスタックポインタとして指定することにより、例示的スタックを実装することができる。ＴＲＡＰが呼び出されるとき、このレジスタの値はコンテンツコードによって読み取られて、渡されたパラメータ値を設定する。（スタックポインタまたはスタックコンテンツが無効である場合、ＴＲＡＰオペレーションを無視する、または、有効な値を代用するなどの適切なアクションが取られる。）整数をスタックにプッシュする例示的プロセスは、最初に４をスタックポインタから減算し、次いでこの値を格納して、新しいスタックポインタによって指定されたメモリアドレスにプッシュすることを含む。値をスタックから取り出すことは、最初に、スタックポインタによって指定されたメモリアドレスから値を読み取り、次いで４をスタックポインタに加算することによって、実行される。ＴＲＡＰオペレーションに続いて、ＴＲＡＰを実装するコンテンツコードおよび／またはコードは、例えばスタックポインタ値を元に戻すことによって、パラメータをスタックから消去する。他のスタック、ならびに、より一般には、関数コールおよびパラメータ受け渡し技術が背景技術において知られており、本発明と共に使用することもできる。

【０１７５】

ＴＲＡＰを呼び出すため、呼び出し元は最初に（プロトタイプ定義における最も右側にあるパラメータから開始して）パラメータの各々をプッシュし、適切なＴＲＡＰオペレーションを呼び出し、（例えば、慣例により、レジスタ１内に格納された結果値を読み取ることにより）戻り値を得る。例えば、「ＵＩＮＴ３２ ＴＲＡＰ＿Ｒａｎｄｏｍ（ＵＩＮＴ８ ＊ｄｓｔ，ＵＩＮＴ３２ ｌｅｎ）；」と示される以下のＴＲＡＰを考察する。トラップオペレーションを呼び出すことに先立って、呼び出し元（コンテンツコード）は最初に３２ビット値「ｌｅｎ」をプッシュし、次いで３２ビットポインタ値「ｄｓｔ」（これは、結果が格納されるべきであるメモリアドレスである）をプッシュする。呼び出し元は、次いで適切な低レベルＴＲＡＰオペレーションを呼び出す。通常（必ずではないが）、ＴＲＡＰを呼び出したコンテンツコードスレッドは、要求したオペレーションが実行される間、停止する。ＴＲＡＰの戻り値（この実施例のプロトタイプでは、左側の「ＵＩＮＴ３２」をもって示される）は、コンテンツコードが取得できる位置に入れられる（例えば、戻り値を所定のレジスタ内に入れることによる）。例示的ＴＲＡＰの大部分はステータス値（例えば、成功を示す定数ＳＴＡＴＵＳ＿ＯＫ、または、エラー状態を示すか、またはそうでない場合は、システムまたはオペレーションの結果または状態を説明する値）を返す。簡潔にするために、戻り値の内容は、その値が単にステータス値である場合、以下

のＴＲＡＰの説明において全般的に省略される。一部のＴＲＡＰはいずれのステータス値も返さないが、失敗する場合があり（例えば、不正な入力パラメータが与えられた場合）、その場合、コンテンツコードは、適切な入力が確実に提供されること、および／または、オペレーションが期待通りに完了したことを検証することを要求する場合がある。

【０１７６】

ＴＲＡＰオペレーションを使用することにより交換された値は、例えば、暗号化および／またはデジタル署名を使用して、暗号的に保護することができる。例えば、ＴＲＡＰに渡された値は、プレイヤまたは他のデバイスの公開キーによって暗号化することができる。値はまた、例えばブロックまたはストリーム暗号を使用して、対称暗号技術により暗号化することもできる。値にデジタル署名することもできる。応答もまた暗号的に保護することができる。例えばプレイヤ、プレイヤのメーカ、外部デバイス、フォーマットエンティティなどによって、例えば、応答にデジタル署名することができる。使用される暗号化保護（および、対応するキーなど）の特定の組合せを指定することができ、または、コンテンツコード要求、プレイヤ要求、外部デバイス、プレイヤの状態などに基づくようにすることができる。データを保護するために使用されるキーはプレイヤ内に含まれない（または、プレイヤによってアクセス可能ではない）場合があり、これは例えば、データの終点はコンテンツコードおよび外部デバイスであるが、プレイヤを通過するデータ通信の場合である。

【０１７７】

ＴＲＡＰオペレーションは、しばしばシステム状態を変更する。例えば、コンテンツコードにとって可視のレジスタ値およびメモリコンテンツは変更される場合がある。例えば、上記の「ＴＲＡＰ＿Ｒａｎｄｏｍ」の実施例では、乱数生成器からの出力が、コンテンツコードによる使用のために格納される。コンテンツコードの直接制御の外部にある再生システムの部分も一般に、ＴＲＡＰオペレーションによってアクセスまたは変更される。例えば、ＴＲＡＰ＿ＳｌｏｔＷｒｉｔｅは、プレイヤの不揮発性メモリスロットのコンテンツを変更することができ、ＴＲＡＰ＿ＭｅｄｉａＲｅａｄＲｅｑｕｅｓｔは、保留中のメディア読み取りのキューに要求を追加する。これらの変更は、コンテンツコードによって直接監視可能であっても、そうでなくてもよい。

【０１７８】

（プレイヤのオペレーション）

以下で、例示的実施形態によってサポートされるＴＲＡＰオペレーションを説明する。この設計の制限、制約および他の態様は例示的実施形態を示すもので、必ずしも他の実施形態を示すものではないことに留意されたい。

【０１７９】

void TRAP_Yield(UINT32 sleepTime);

（ａ）ミリ秒単位で指定された時間、または（ｂ）イベントの発生（例えば、メディア読み取り完了など）のうちの早い方まで、インタープリタを制御する。ＴＲＡＰ＿Ｙｉｅｌｄが呼び出されるとき、実装は、インタープリタを中断するよう要求しないが、特に、ポータブルデバイスのバッテリ電力を節約するため、または、マルチスレッドシステム上のプロセッササイクルを節約するために、インタープリタを中断することが望ましい場合がある。実装はまた、任意で、ｓｌｅｅｐＴｉｍｅパラメータを無視する。より小さい値がｓｌｅｅｐＴｉｍｅに指定される場合、または、イベントがより早く発生する場合でも、このオペレーションは１ミリ秒を費やす場合がある。イベントが発生する場合、制御はイベント後１ミリ秒以内に再開する。イベントが発生しない場合、実行は最大でｓｌｅｅｐＴｉｍｅミリ秒にわたって中断される場合がある。ＴＲＡＰは何も返さず、プレイヤの準備ができているとき、次の命令をもって実行が続けられる。

【０１８０】

UINT32 TRAP_Random(UINT8 *dst, UINT32 len);

プレイヤによって生成されたｌｅｎランダム（または強擬似ランダム）バイトをｄｓｔに格納する。

【０１８１】
UINT32 TRAP_Sha(UINT8 *dst, UINT8 *src, UINT32 len, UINT32 op);
ｓｒｃで、ｌｅｎバイトに対してＳＨＡ－１アルゴリズムを実行し、結果をｄｓｔに格納する。ｏｐ用の値には、さらなるデータを既存のＳＨＡ状態に追加するＳＨＡ＿ＵＰＤＡＴＥ（０）、データをハッシングする前に新しいＳＨＡコンテキストを開始するＳＨＡ＿ＩＮＩＴ（１）、ハッシュの更新が完了するときにファイナライズオペレーションを実行するＳＨＡ＿ＦＩＮＡＬ（２）、および、フルブロックオペレーションを実行するＳＨＡ＿ＢＬＯＣＫ（３）（ＳＨＡ＿ＩＮＩＴおよびＳＨＡ＿ＦＩＮＡＬの両方を設定することに相当する）が含まれる。
【０１８２】
ｄｓｔには、少なくとも５１２バイトの使用可能な空間があるべきであり、ファイナライズオペレーション（例えば、ｏｐ＝ＳＨＡ＿ＦＩＮＡＬ）が生じるまで、その空間のコンテンツは確定できない。
【０１８３】
プレイヤ実装は、ｄｓｔにおける５１２バイトを状態の一時ストレージのために使用することができるが、ｄｓｔのコンテンツはコンテンツコードによって悪意をもって選択されたと想定すべきである。成功の場合はＳＴＡＴＵＳ＿ＯＫを返し、失敗の場合は、定義済みエラーコードのうちの１つを返す。
【０１８４】
UINT32 TRAP_Aes(UINT8 *dst, UINT8 *src, UINT32 len, UINT8 *key, UINT32 opOrKeyID);
ＡＥＳ ＥＣＢアルゴリズムをｓｒｃからｌｅｎブロックに対して実行し、（以下で変換されるように）ｋｅｙにおけるキーを使用して、復号化結果をｄｓｔに格納する。ｏｐＯｒＫｅｙＩＤの値は、キーがどのように導出されるべきであるか、および、ＡＥＳ暗号化または復号化を実行するべきであるかどうかを指定する。ｏｐＯｒＫｅｙＩＤについて、以下の値がサポートされる。
【０１８５】
・opOrKeyID=AES_ECB_ENCRYPT（0xFFF10000）－ＥＣＢモードを使用して、ｋｅｙにおける１６バイトキーにより、データを暗号化する。
【０１８６】
・opOrKeyID=AES_ECB_DECRYPT（0xFFF10001）－ＥＣＢモードを使用して、ｋｅｙにおける１６バイトキーにより、データを復号化する。
【０１８７】
・opOrKeyID=AES_ECB_DECRYPT_MEDIA_KEY（0xFFF10002）－ｋｅｙにおける暗号化されたキー値を、現在挿入されているメディアのためのメディアキーを使用して復号化し、次いで、結果をキーとして使用して、ｓｒｃにおけるデータを、ＥＣＢモードを使用して復号化する。
【０１８８】
・opOrKeyID=他の任意の値。ｏｐＯｒＫｅｙＩＤによって特定されるプレイヤキーを使用して、ポインタｋｅｙにおける暗号化されたキーを復号化し、次いで、結果として生じる復号化されたキーを使用して、ポインタｓｒｃにおけるデータを、ＥＣＢモードを使用して復号化する。（コンテンツコードはプレイヤの証明書をチェックして、プレイヤのメインＡＥＳキーセット用のキー範囲を決定することができ、これらのキーは必ずしも０で開始するとは限らず、複数のキーのセットが存在する場合があることに留意されたい。）ｏｐＯｒＫｅｙＩＤによって選択されたキーはプレイヤアプリケーション内に配置される場合があるが、キーはドライブ内、デバイスドライバ内、表示／出力デバイス内、ネットワークを介してリモートに、ユーザリムーバブルなスマートカード（または、他の耐タンパ性を有するチップ）内に、リムーバブルでない耐タンパ性を有するチップ内に、複数のデバイスの間で分割されて、配置される場合もある（ただし、これらに限定されない）。最初の復号化オペレーション（コンテンツにより指定された暗号化されたキーの復号化）を

、識別されたキーを含むデバイスによって実行することができる一方で、バルク処理（すなわち、ｓｒｃにおけるデータの復号化）を他の場所（例えば、キーストレージのための不揮発性メモリを欠く高速暗号モジュール内）で実行することができる。

【０１８９】

暗号化オペレーション、特に外部デバイスに伴って生じる暗号化オペレーションは、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙおよび／またはＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを介して実行することもできる。暗号ブロックチェーン（ＣＢＣ）、カウンタモード、および他のブロック暗号モードを、コンテンツコードを（任意で、ＴＲＡＰ＿Ｘｏｒなどのオペレーションと共に）使用して、ＥＣＢオペレーションから実行することができる。代替実施形態は、ＡＥＳ以外のアルゴリズムおよび／またはＥＣＢ以外のモードを直接提供することもできる。

【０１９０】

UINT32 TRAP_PrivateKey(UINT8 *dst, UINT8 *src, UINT32 srcLen, UINT32 controlWord, UINT32 keyID);

このオペレーションは、ｋｅｙＩＤ（通常は０）によって選択されたＲＳＡ秘密キー、または、プレイヤ内（または、プレイヤによってアクセス可能なデバイス内）の所定の他の非対称キーを使用して、一部のデータを変換する。ｋｅｙＩＤから、結果の長さおよびオペレーションタイプ（例えば、署名または復号化）は暗示される。使用可能なキーについての情報には、対応する公開キーおよび証明書（コンテンツコードが検証できるもの）が含まれ、この情報は、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙおよび／またはＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して取得することができる。どのデータを提供すべきであるかの選択は、コンテンツコード次第である（例えば、メディアおよび／またはＴＲＡＰからのいかなるデータのいかなる機能も提供することができる）。例えば、署名されたデータを生成する際にイベントデータ（ＴＲＡＰ＿ＥｖｅｎｔＧｅｔを参照）を組み込むことによって、署名オペレーションに対して提示された値をユーザインターフェース（および他の）イベントに結合することができる。

【０１９１】

プレイヤのメインＲＳＡ秘密キー（ｋｅｙＩＤ＝０）では、このオペレーションは、ＳＨＡ－１ハッシュの２０４８ビットのＲＳＡ署名を作成する。このハッシュは以下のように計算される。（ａ）４バイト（ＭＳＢが最初）としてエンコードされた、ｓｒｃＬｅｎ値をハッシュし、（ｂ）ユーザにより指定されたバッファのコンテンツ（すなわち、ｓｒｃにおけるｓｒｃＬｅｎバイト）をハッシュし、（ｃ）４バイト（ＭＳＢが最初）としてエンコードされた、ｃｏｎｔｒｏｌＷｏｒｄ値をハッシュし、（ｄ）ｃｏｎｔｒｏｌＷｏｒｄビット３１（ＭＳＢ）が設定される場合、メディアＩＤの値をハッシュし、（ｅ）ｃｏｎｔｒｏｌＷｏｒｄビット３０が設定される場合、宛先ポインタをＰＣ＋４に設定し、ｄｓｔをオーバーライドし、（ｆ）ｃｏｎｔｒｏｌＷｏｒｄビット２９が設定される場合、現在のプログラムカウンタで開始する（２＾１６を法とした制御ワード）コードバイトをハッシュし、次いで、（ｇ）ｃｏｎｔｒｏｌＷｏｒｄビット２８が設定される場合、ＰＣの現在の値をハッシュに組み込む。ＳＨＡ－１ハッシュ結果は次いで、「０ｘ００｜｜０ｘ０１｜｜０ｘＦＦ（２３３回繰り返し）｜｜００」をＳＨＡ－１ハッシュの先頭に追加することによって、パディングされる。パディングされた値は次いで、プレイヤの公開モジュラス（public modulus）を法として、プレイヤの秘密指数（secret exponent）まで引き上げられる。

【０１９２】

署名を検証するため、プレイヤの公開指数は３であり、コンテンツコードにより、システム全体の公開キー（コンテンツコード内に含まれる定数とすることができ、任意で難読化した形態をとる）を使用してプレイヤの証明書を検証することによって、公開モジュラスを得ることができる。

【０１９３】

制御ワードによって提供されたオプションは、コンテンツコードが実行環境についてのア

テステーション（attestation）を得ることを可能にし、これは、インタープリタを含むもの以外の準拠デバイスによって署名が実行されている状況を検出することなどである。インタープリタと、コンテンツコードとの間の検証可能なバインディングは、攻撃者が正当なプレイヤ上で実行される悪意のあるコンテンツを使用して、正当なコンテンツを実行する悪意のあるプレイヤによる使用のために暗号化結果を作成しようと試みる状況に対処するために、有用である可能性がある。

【０１９４】

実施形態は、非対称暗号化アルゴリズム（ＲＳＡ、ＤＳＡ、楕円曲線のバリアント（variant）、Ｄｉｆｆｉｅ－Ｈｅｌｌｍａｎなど）、オペレーション（署名、検証、キー合意など）、および任意の組合せのキーサイズをサポートすることもできる。対称オペレーションを非対称オペレーションと統合することもできる。ＲＳＡ署名検証などの一部の暗号化オペレーションを、いかなる特別なトラップもなしに、または、汎用パフォーマンス加速オペレーション（例えば、ＴＲＡＰ＿ＡｄｄＷｉｔｈＣａｒｒｙなど）のみを使用して、コンテンツコード内に実装できることに留意されたい。より複雑な暗号ＴＲＡＰの一実施例は、以下のうち一部またはすべてを行うものである。（ａ）ＲＳＡ公開キーオペレーションを実行して、データのブロック上の署名を検証すること、（ｂ）署名が有効である場合、ＲＳＡ秘密キーオペレーションを実行して、検証された部分におけるブロックデータを復号化して、対称キーを復旧すること、（ｃ）ＲＳＡ復号化が成功する場合、対称キーを使用して、一部のデータ（例えば、暗号化されたキーに続く署名付きブロックにおけるデータ）を（例えば、ＨＭＡＣ－ＳＨＡを使用して）復号化および検証すること、次いで（ｄ）インタープリタを使用して、復号化されたデータをコードとして処理すること。

【０１９５】

代替実施形態では、任意の方法の他の計算オペレーション（他のＴＲＡＰなど）の入力および出力の署名、検証、復号化、暗号化、またはそうでない場合は処理のために暗号化サポートを提供することができる。

【０１９６】

UINT32 TRAP_AddWithCarry(UINT32 *dst, UINT32 *src, UINT32 len);

このオペレーションは、桁上げ伝搬（carry propagation）によりマルチワード加算オペレーションを実行する。ｓｒｃ［０．．ｌｅｎ－１］における値がｄｓｔ［０．．ｌｅｎ－１］に加算される。ｓｒｃおよびｄｓｔの値を各々、ｌｅｎワードとして格納され、それぞれアドレスｓｒｃ［０］またはｄｓｔ［０］における最上位ワードでエンコードされた大数を示すものとして検証することができる。最下位ワードは、それぞれ、ｓｒｃ［ｌｅｎ－１］およびｄｓｔ［ｌｅｎ－１］に配置される。

【０１９７】

例えば、数０ｘ０８０９０Ａ０Ｂ０Ｃ０Ｄ０Ｅ０Ｆは、ｌｅｎ＝２を有し、ポインタ（例えば、ｓｒｃ［０］）によって指定された位置０ｘ０８０９０Ａ０Ｂ、および、ポインタに４を加えたもの（例えば、ｓｒｃ［１］）によって参照されたバイトオフセットにおける値０ｘ０Ｃ０Ｄ０Ｅ０Ｆと共に格納される。

【０１９８】

ソースおよび宛先領域が重なる場合、正しいオペレーションは、ｓｒｃ＝ｄｓｔである場合にのみ保証される。最後の（最上位または左側の）加算ステップで桁上げが生じた場合、このオペレーションの戻り値は１であり、そうでない場合は、戻り値はゼロである。

【０１９９】

（別のＴＲＡＰなしにコンテンツコードを使用してＴＲＡＰ＿ＡｄｄＷｉｔｈＣａｒｒｙオペレーションを、様々な他のＴＲＡＰオペレーションと共に実装することもできるが、専用のＴＲＡＰは、プレイヤ設計者が各プラットフォーム上で使用可能な最も効率的な技術を選択することを可能にし、それにより、幅広い種類の可能なプレイヤ設計にわたってよりよいパフォーマンス保証を可能にすることに留意されたい。）

【０２００】

UINT32 TRAP_SubtractWithBorrow(UINT32 *dst, UINT32 *src, UINT32 len);

このオペレーションは、桁借り（borrow）（桁上げ）伝搬によりマルチワード減算オペレーションを実行する。具体的には、ｓｒｃ［０．．ｌｅｎ－１］における値はｄｓｔ［０．．ｌｅｎ－１］から減算される。ｓｒｃおよびｄｓｔの値を各々、ｌｅｎワードとして格納され、それぞれアドレスｓｒｃまたはｄｓｔにおける最上位ワードでエンコードされた、大数を示す。例えば、数０ｘ０８０９０Ａ０Ｂ０Ｃ０Ｄ０Ｅ０Ｆは、ｌｅｎ＝２を有し、ポインタ（例えば、ｓｒｃ［０］）によって指定された位置０ｘ０８０９０Ａ０Ｂ、および、ポインタに４を加えたもの（例えば、ｓｒｃ［１］）によって参照されたバイトオフセットにおける値０ｘ０Ｃ０Ｄ０Ｅ０Ｆと共に格納される。ソースおよび宛先領域が重なる場合、正しいオペレーションは、ｓｒｃ＝ｄｓｔである場合にのみ保証される。最後の（最上位または左側の）減算ステップで桁借りの必要がある場合、このオペレーションの戻り値は１であり、そうでない場合は、戻り値はゼロである。

【０２０１】

UINT32 TRAP_MultiplyWithRipple(UINT32 *dst, UINT32 *src, UINT32 multiplicand, UINT32 len);

このオペレーションは、ｍｕｌｔｉｐｌｉｃａｎｄを、ｄｓｔ［０．．ｌｅｎ－１］内の数の上に乗算する。結果はｌｅｎ＋１ワード長である。結果の最上位ワードが返され、残りはｄｓｔ［０．．ｌｅｎ－１］に格納される。ｄｓｔ値は、ｌｅｎワードとして格納され、ｄｓｔによって示されたアドレスの最上位ワードをもってエンコードされた大数を示すべきである。

【０２０２】

結果の最上位３２ビットが返される（これは、被乗数およびｄｓｔ［０．．ｌｅｎ－１］の積の最上位の３２ビットに等しい）。

【０２０３】

void TRAP_XorBlock(UINT32 *dst, UINT32 *src, UINT32 len);

このオペレーションは、メモリの２つのブロックの排他的ＯＲを計算する。ｓｒｃ［０．．ｌｅｎ－１］に格納されたデータは、ｄｓｔ［０．．ｌｅｎ－１］におけるデータ上にＸＯＲされる。このオペレーションは、例えば、ＣＢＣモードブロック暗号オペレーションを作成するときに、有用である。ｓｒｃおよびｄｓｔブロックが重なる場合のオペレーションは定義されない。戻り値は提供されない。

【０２０４】

void TRAP_Memmove(UINT8 *dst, UINT8 *src, UINT32 len);

このオペレーションは、ｓｒｃからｄｓｔまでのｌｅｎバイトをコピーする。ソースおよび宛先ブロックが重なる場合でも、結果は正しい。戻り値は提供されない。

【０２０５】

UINT32 TRAP_Memcmp(UINT8 *ptr1, UINT8 *ptr2, UINT32 maxlen);

このオペレーションは、ｐｔｒ１におけるメモリとｐｔｒ２におけるメモリとを、最大ｍａｘｌｅｎバイトについて比較する。差異が見つからない場合、戻り値は０であり、最初の差異においてｐｔｒ１における値の方が大きい場合、戻り値は１であり、最初の差異においてｐｔｒ１における値が小さい場合、戻り値は０ｘＦＦＦＦＦＦＦＦである。

【０２０６】

void TRAP_Memset(UINT8 *dst, UINT8 fillvalue, UINT32 len);

このオペレーションは、ｆｉｌｌｖａｌｕｅによって指定されたバイト値でメモリを満たす。宛先アドレスはｄｓｔによって指定され、格納すべきバイトの数はｌｅｎによって指定される。戻り値は提供されない。

UINT8* TRAP_Memsearch(UINT8 *region, UINT32 regionLen, UINT8 *searchData, UINT32 searchDataLen);

このオペレーションは、１つまたは複数のバイトについてメモリを検索する。具体的には、ｒｅｇｉｏｎ（ｒｅｇｉｏｎＬｅｎバイトの長さを有する）内でｓｅａｒｃｈＤａｔａ（ｓｅａｒｃｈＤａｔａＬｅｎバイトの長さを有する）の最初の出現を発見する。それらがｒｅｇｉｏｎ［０．．ｒｅｇｉｏｎＬｅｎ－１］内全体に出現する場合、合致が見つか

る。この範囲内で開始するが最後を越える合致は、カウントされない。このオペレーションは、最初の合致に対するポインタを返す。ｒｅｇｉｏｎ内で合致が見つからない場合、戻り値はＮＵＬＬ（ゼロ）である。

【０２０７】

UINT32 TRAP_SlotAttach(UINT32 slot, UINT32 codeLen, UINT32 reqPriority);

このオペレーションは、既存の不揮発性メモリスロットにアタッチするか、または、（ｓｌｏｔの指定された値がゼロである場合）新しいスロットを割り振る。スロットアタッチオペレーションは、指定されたコード（呼び出し元ＰＣで開始し、ｃｏｄｅＬｅｎバイトまで広がる）がスロットのａｕｔｈｏｒｉｚａｔｉｏｎＨａｓｈに合致しない場合、失敗する（デフォルトのスロットゼロになる）。（不揮発性メモリサポートについてのさらなる情報については、「不揮発性メモリの保護」というタイトルのセクションを参照。）

【０２０８】

UINT32 TRAP_SlotRead(UINT8 *dst, UINT32 slot);

このオペレーションは、不揮発性メモリスロットを読み取ることを試み、成功する場合、指定された位置に結果を格納する。指定されたスロットが、現在アタッチされているスロットではない場合、スロットコンテンツのプライベートデータ部分は返されない。（さらなる情報については、「不揮発性メモリの保護」というタイトルのセクションを参照。）

【０２０９】

UINT32 TRAP_SlotWrite(UINT8 *newContents);

このオペレーションは、現在アタッチされている不揮発性メモリスロットに書き込むことを試みる。（さらなる情報については、「不揮発性メモリの保護」というタイトルのセクションを参照。）スロット書き込みオペレーションはアトミック（atomic）であり、これは、（例えば、不意の停電のため）オペレーションが失敗する場合、オペレーションは全体として完了されるか、またはまったく完了されないかのいずれかであることを、プレイヤが保証することを意味する。

【０２１０】

ＴＲＡＰ＿ＳｌｏｔＷｒｉｔｅオペレーションは、書き込まれたデータがスロットの複数の部分について正しい値を有することを確実にする。これらの部分には、ｃｒｅａｔｏｒＭｅｄｉａＩＤ（コンテンツコードが最初にスロットを作成したメディアの識別子）、ｌａｓｔＵｐｄａｔｅＭｅｄｉａＩＤ（コンテンツコードが直前にスロットに書き込んだメディアの識別子）、ｌａｓｔＵｐｄａｔｅＳｅｑｕｅｎｃｅＣｏｕｎｔｅｒ（書き込みにつき１回単調増加する、グローバル書き込みカウンタの値）、および、ｓｌｏｔＰｒｉｏｒｉｔｙ（スロットのための上書き優先順位を指定する）が含まれる。

【０２１１】

コンテンツコードは、汎用領域に書き込まれたデータを制御する。格納することができる情報の例には、再生状態情報（再生が休止／中断された位置および構成など）、監査情報（監査データのための実行ハッシュまたは他の暗号化認証を含む）、ビデオゲーム用のハイスコア、電子購入を自動的に完了するためのデフォルト情報（ユーザ名、ユーザ住所、クレジットカード番号、クレジットカード期限、請求先住所、出荷先住所、電話番号、電子メールアドレスなど）、実行可能コード（例えば、新しいセキュリティロジック、タイトルまたはプレイヤにおける問題を是正するパッチなど）、キー（および他のセキュリティ情報）などが含まれるが、これらに限定されるものではない。

【０２１２】

戻り値は、書き込みオペレーションが成功したかどうか、および、メディアの全般的状況（例えば、不揮発性メモリが過剰な数の書き込みサイクルにより使い尽くされる寸前である場合の警告）を示す。読み取りオペレーションは、書き込みが成功したという追加の検証のために、書き込みの後に実行される場合がある。

【０２１３】

UINT32 TRAP_MediaEject(void);

このオペレーションは、現在ドライブ内にあるいずれかのリムーバブルメディアを排出す

ることを試みる。ＴＲＡＰ＿ＭｅｄｉａＥｊｅｃｔ（）に対する呼び出しが成功した後、インタープリタは動作を続ける場合があるが、多くの場合、インタープリタは次いでＴＲＡＰ＿Ｓｈｕｔｄｏｗｎ（）を呼び出すことによって、それ自体をシャットダウンする。プレイヤが、自動排出をサポートすることは必要とされず、その場合、オペレーションは失敗する。（コンテンツコードは、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙを使用して、排出サポートおよび他のプレイヤ機能を検出することができる。）

【０２１４】

状況によっては、メディアが交換される間、コンテンツコードのメモリ空間の諸部分を保存することが有用である場合がある。例えば、これは、マルチディスクセットにおいて複数のディスクの間で切り替えるとき、不揮発性メモリスロット内に容易に保存されるよりも多い量のデータを保存するために、有用である場合がある。この状況では、ＴＲＡＰ＿ＭｅｄｉａＥｊｅｃｔを使用して、最初のメディアを排出する。ユーザは次いで、（例えば、オーバーレイにより指示された後）コンテンツの別の部分を挿入する。新たに挿入されたコンテンツの識別を、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙを介して決定することができる。予期しないメディア交換を防止するため、新しいメディアを使用可能にするために、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓに対する明示的なコールが必要とされる。全体的な読み取りについて許可済みとしてマークが付けられた新しいメディアの部分を、次いでロードすることができる。（いくつかの部分は、メディア自体のコンテンツコードによってのみ読み取られるべきであると指定される場合がある。より高度なメディア－メディア（例えば、ディスク－ディスク）チェーニング（chaining）機能を必要とする実施形態は、最初のメディア上のコードが２番目のメディアを認証することを可能にし、また、２番目のメディア上のコードがソースメディアを認証することも可能にすることができる。例えば、ソースおよび宛先ディスク上に保持されたデジタル署名付きルールおよび証明書に基づいて、プレイヤがメディア認証を実行することもできる。）

【０２１５】

UINT32 TRAP_MediaReadRequest(UINT32 offsetHigh, UINT32 offsetLow, UINT32 len);

このオペレーションは、一部のデータの検索を開始するようメディアサブシステムに伝える。例示的実施形態では、ｌｅｎ値は、セクタ境界を含めて１メガバイト（１０４８６７５バイト）を超えることができない。（オフセットがセクタ境界上にない、すなわち、ＳＥＣＴＯＲ＿ＳＩＺＥの正確な倍数でない場合、この長さは１メガバイトからセクタオフセットを引いた値を超えることができず、これは、ＳＥＣＴＯＲ＿ＳＩＺＥを法としたｏｆｆｓｅｔＬｏｗとして計算される。ＳＥＣＴＯＲ＿ＳＩＺＥは現在２０４８バイトとして定義される。）プレイヤは少なくとも２つの未完了の要求（例えば、現在提供されているもの、および、次に提供されるようになるもの）がバッファされることを可能にする。これは、コンテンツがドライブを継続的にビジーに保つことを可能にする。プレイヤは通常、各保留中の要求について別々の１メガバイトの保留バッファを使用する。いずれのエラーもないリターンは、その読み取りが試みられるようになる（、すなわち、要求は明らかに悪くはない）ことを意味するが、その読み取りが実際に成功するかは保証しない（ＴＲＡＰ＿ＭｅｄｉａＲｅａｄＦｉｎａｌを参照）。

【０２１６】

このオペレーションは単に、読み取りが実行されることを要求する（、および、実際にはいかなるデータもコンテンツコードのメモリ領域にロードしない）一方で、代替実施形態は宛先アドレスを指定して、（例えば）イベントを受信することによって、または、宛先バッファが満たされているかどうかをチェックすることによって、読み取りが完了したかどうかを検出することができる。このような手法がコードベースのＮＶＲＡＭセキュリティ機能と共に使用される場合（ＴＲＡＰ＿ＳｌｏｔＡｔｔａｃｈを参照）、セキュリティ手段は、ロック解除されたＮＶＲＡＭスロットに関連付けられたセキュリティコードを破損するか、またはそうでない場合はセキュリティを攻撃するために、敵対者が保留中の読み取りを利用しないことを確実にすることを要求される場合がある。例えば、プレイヤは、ＮＶＲＡＭスロットがロック解除されている間にメディア読み取りを一時停止または禁

止すること、および／または、このようなオペレーションが保留中である間にＮＶＲＡＭスロットアタッチ要求を拒否することができる。

【０２１７】

UINT32 TRAP_MediaReadFinal(UINT8 *dst);

このオペレーションは、最も古い（先頭の）要求された読み取りが完了しているかどうかをチェックする。完了していない場合、ＳＴＡＴＵＳ＿ＮＯＴ＿ＤＯＮＥの戻り値が返される。読み取りが保留中でない場合、ＳＴＡＴＵＳ＿ＮＯＴ＿ＦＯＵＮＤの戻り値が返される。読み取り要求が失敗した場合、ＳＴＡＴＵＳ＿ＥＲＲＯＲが返される。読み取り要求が成功した場合、要求されたデータはドライブのバッファからコンテンツコードのメモリ空間の指定されたアドレスへ転送され、ＳＴＡＴＵＳ＿ＯＫの値が返される。アドレスが０ｘＦＦＦＦＦＦＦＦである場合、データは廃棄される。

【０２１８】

特別な読み取り（バーストカッティングエリアの読み取り、無効にされたエラー検出／修正を伴うデータ読み取りなど）は、このオペレーションではなく、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して構成および実行されることに留意されたい。

【０２１９】

UINT32 TRAP_Mediawrite(UINT32 offsetHigh, UINT32 offsetLow, UINT8 *src, UINT32 len);

このオペレーションは、指定されたデータをメディアに書き込み、指定されたオフセットで開始する（オフセットは、３２ビット上位値および３２ビット下位値としてエンコードされた６４ビット値である）。

【０２２０】

このオペレーションは、書き込みをサポートするメディアフォーマットに対してのみ適用可能である。書き込み可能領域はまた、メディアの所定の部分に制限される場合もある。書き込みはまた、所定の状況では禁止される場合もある（例えば、メディアが交換されている場合、メディア上のレコーダ識別子が現在のデバイスに対応しない場合、メディアが、書き込み機能の使用を許可する有効な証明書を欠く場合など）。

【０２２１】

ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して、書き込みを保護または可能にするために必要とされるいずれかの必要な認証またはキー合意を提供することができる。追記型メディアでは、以前に書き込まれた部分を上書きしようとする試行（、および、その位置でメディアを破損するリスク）を、（例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを介して）明示的な許可が得られない限り、拒否することができる。（通常は望ましくないが、このような書き込みは、ある状況では、例えば、攻撃に応答して、メディアに自己破壊させるために、有用である可能性がある。）ドライブが、書き込みヘッドの調節、トラッキング制御、エラー訂正コード、読み取り／書き込みヘッドの光学特性、または他の構成パラメータの直接制御を可能にする場合、これらもまたＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを介して調整することができる。書き込まれたメディアの特性は、例えば、どの書き込みデバイスが特定の光メディアを作成したかをコンテンツ所有者が割り出すことができるように、科学捜査の目的のために修正することができる。

【０２２２】

UINT32 TRAP_EventGet(UINT8 *evBuf, UINT32 len);

このオペレーションは、（ユーザインターフェースオペレーション、セキュリティ変更などの）いずれかのイベントが保留中であるかどうかをチェックし、保留中である場合、最初のもの（または最高優先順位のもの）についての情報を、ｅｖｅｎｔによって指定されたアドレスに格納する。イベントが発見されない場合、ＳＴＡＴＵＳ＿ＮＯＴ＿ＦＯＵＮＤの値が返され、ｅｖｅｎｔの内容は変わらない。イベントが発見された場合、トラップはＳＴＡＴＵＳ＿ＯＫを返し、イベントを説明する情報をｅｖＢｕｆに格納する（が、ｌｅｎバイトを超えて格納しない）。

【０２２３】

イベントバッファ（ｅｖＢｕｆ）内の返されるデータは、イベントによって決まる。ユーザインターフェースイベントが返される場合、以下のデータ構造は、ｅｖＢｕｆによって指定されたメモリアドレスに格納される。

```
typedef struct{
UINT32 Source; /* Device type that produced event */
UINT32 Action; /* Actual/suggested action for event */
UINT32 Char; /* UTF-8 value of event (if applicable) */
UINT32 Direction; /* Compass values for event */
UINT32 Xpos; /* Pointer X position for event */
UINT32 Ypos; /* Pointer Y position for event */
UINT32 Reserved[10]; /* Do not interpret (future use) */
} EventInfo_UserInterface;
```

【０２２４】

Ｓｏｕｒｃｅフィールドは、イベントが発生したデバイスを示す。Ｓｏｕｒｃｅフィールドの解釈はあらかじめ定められていてもよいし、または、（例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙから得られた情報を介して）コンテンツコードによって決定されてもよい。指定することができる可能なソースの例には、リモートコントロール、キーボード、フェースプレートキー（faceplate key）、マウス（および、他のポインティングデバイス）、メディアドライブ、外部デバイス、タイマ、コーデック、未知／不特定のソースなどが含まれるが、これらに限定されるものではない。

【０２２５】

Ａｃｔｉｏｎフィールドは、イベントのための、または、イベントを引き起こした、推奨されるアクションを示す。この値はモーダル（、すなわち、プレイヤの現在の状態に影響を及ぼす）であっても、一定であってもよい。アクションは再生状態を変更するか、または他の効果を有する場合がある。可能なアクションの例には、再生、休止、停止、巻き戻し（様々な速度で）、早送り（様々な速度で）、スローモーション（様々な速度で）、電源オフ、排出、チャネルアップ、チャネルダウン、選択、エンター、音量調節（または消音）、ガイド／メニュー選択、表示アングルの変更、前方スキップ、シーン選択、ユーザ言語選択、サブタイトル制御、ビデオアングル変更、選択、戻る、進む、イエス、ノー、購入、終了、ヘルプ、エラー、出力デバイス変更通知、再生環境再構成通知、セキュリティアラートなどが含まれるが、これらに限定されるものではない。未知である場合（イベントに対応するアクションがない場合など）、このフィールドはゼロである。（このフィールドは、ユーザインターフェースイベントに対して、ならびに他のイベントに対して有効である。この値の最上位の１６ビットは後続のフィールドの構造、例えば、ＥｖｅｎｔＩｎｆｏ＿ＵｓｅｒＩｎｔｅｒｆａｃｅ、ＥｖｅｎｔＩｎｆｏ＿ＳｅｃｕｒｉｔｙＡｌｅｒｔなどを示す。）

Ｃｈａｒフィールドは、キーに対応する文字（例えば、ＡＳＣＩＩ、ダブルバイトなどを使用してエンコードされた文字）を示す。未知の場合（イベントに対応する文字がない場合など）、このフィールドはゼロである。

【０２２６】

Ｄｉｒｅｃｔｉｏｎフィールドは、イベントが指示する方向を示すか、ない場合はゼロである。この方向はいかなる角度にすることもできるが、方向は所定の値（例えば、北／上、北東／右上、右／東など）に量子化される（quantized）場合がある。

【０２２７】

イベントを標準のＸ／Ｙグリッド上のポインタ移動またはアクションとして解釈することができる場合、ＸｐｏｓおよびＹｐｏｓフィールドは位置を示す。イベントがＸおよび／またはＹ座標に対応しない場合、対応するフィールドはゼロに設定される。

【０２２８】

単一のイベント（リモート入力デバイス上でのボタン押下など）が複数の用途で解釈される場合があることに留意されたい。例えば、一部の入力デバイスでは、「上」用に使用さ

れるボタンは、数字８にも対応する。この場合、一イベントが生成されると、方向は「上」であり、Ｃｈａｒフィールドは数字「８」である。コンテンツコードは、どちらが適切であるかを判断するためにどの入力が望まれるかについて現在知っていることを利用する。（代替実施形態では、プレイヤは別々の機能または情報を提供して、ユーザイベントを解釈することによりコンテンツコードを支援し、または、コンテンツコードが、複数の可能な解釈によりどのようにイベントを解釈すべきであるかを指定することを可能にすることができる。）コンテンツが未知のイベントタイプを受信する場合、（ソースフィールド内で指定された臨界ビット（criticality bit）が別の方法を示さない限り、）コンテンツは安全にこのイベントタイプを無視する場合がある。

【０２２９】

非ユーザインターフェースイベント（デバイス追加、デバイス除去、デバイス再構成、他の再生環境構成の変化、外部コンポーネントからのデータ要求、オペレーションが完了したという通知などを知らせるイベント）では、または、他の種類のユーザインターフェース（音声認識機能、または、ユーザのジェスチャを観察するように構成されたデジタルカメラなど）からのイベントでは、他のデータフィールドを提供することができる。一部のイベントでは、コンテンツコードが追加のＴＲＡＰ（ベンダ固有のＴＲＡＰなど）を呼び出して、追加の情報を得ることが必要である場合もある。

【０２３０】

ＴＲＡＰ＿ＥｖｅｎｔＧｅｔオペレーションはポーリングを使用して変更を検出する一方で、代替実施形態は、割り込み、または、イベントを処理するための他の周知のメカニズムを使用することができることに留意されたい。

【０２３１】

UINT32 TRAP_CodecOutput(UINT32 idx, UINT8 *buf, UINT32 len);

このオペレーションは、データをコーデックに送信する。単純なプレイヤは、１つのコーデック（例えば、ＣＯＤＥＣ＿ＡＵＤＩＯ＿ＡＮＤ＿ＶＩＤＥＯ）のみ、または２つのコーデック（例えば、ＣＯＤＥＣ＿ＶＩＤＥＯ＿ＭＰＥＧ２およびＣＯＤＥＣ＿ＡＵＤＩＯ＿ＭＰＥＧ）をサポートする可能性がある一方で、より複雑なプレイヤは多数をサポートして、例えば、複数の種類のコンテンツストリームをサポートし、ピクチャインピクチャなどの機能を可能にする可能性がある。ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓオペレーションを使用して、コーデックパラメータを設定することができる。

【０２３２】

UINT32 TRAP_CodecSetTime(UINT64 time);

このオペレーションは、メインコーデックタイムクロックを設定する。ゼロの値を指定すると、クロックは、ＴＲＡＰ＿ＣｏｄｅｃＯｕｔｐｕｔに提供されたデータに埋め込まれたビデオタイムスタンプと再同期される。このコマンドはビデオバッファにバッファされ、そのため、このコマンドに出会うまで、効果は遅延される。ＴＲＡＰ＿ＣｏｄｅｃＦｌｕｓｈを使用することにより、このオペレーションを使用して、リアルタイムビデオスプライシングを可能にし、かつ、コンテンツコードがある圧縮されたビデオ／オーディオストリームから別のものへシームレスに切り替えることを可能にする。

【０２３３】

UINT32 TRAP_CodecSetRate(UINT32 rate);

このオペレーションは、コーデックがデータを消費する速度および方向を設定する。ｒａｔｅ値は符号付き整数に変換され、６４の係数をもって縮小される。例えば、０の速度は「休止した」ことを意味し、１２８は、標準の再生速度の２倍の早送りであり、－２５６（０ｘＦＦＦＦＦＦ００）は４倍の巻き戻しであり、１６は、０．２５倍のスロー再生である。要求された値がサポートされない場合、コーデックは、プレイヤの最大速度に関して、符号を無視してより高く丸めることによって、次に最も近い値を選ぶ。例外は部分再生速度であり、ここではコーデックは決して０（休止した）または６４（標準）に丸めるべきではない。負の再生速度（逆）が指定される場合、コーデックは（ＴＲＡＰ＿ＣｏｄｅｃＳｔａｔｕｓを通じて）通常、ビデオＧＯＰを逆の順序で要求する。デフォルトでは

、オーディオは６４以外の再生速度においては消音される。
【０２３４】
UINT32 TRAP_CodecFlush(void);
このオペレーションは、ビデオおよびオーディオコーデックバッファをフラッシュする。このコマンドは通常、コンテンツコードが、あるストリームから別のストリームに高速で切り替えるためにコーデックバッファ内のデータを削除することを望む場合、使用される。例えば、ユーザが１分だけ前方にジャンプすることを要求する場合、コンテンツコードはこのオペレーションを使用して、コーデックバッファ内のいかなるデータも消去し、新たに提供されたデータが即時にデコードされるようにすることができる。
【０２３５】
UINT32 TRAP_CodecStatus(CodecInfo *info);
このオペレーションは、コーデックのＦＩＦＯキュー、および、コーデックが予想するビデオまたはオーディオの次のチャンクについての情報を検索する。返されるデータ値には、現在のタイムスタンプ（現在表示／出力されているオーディオビジュアル情報に対応するタイムスタンプ、現在圧縮されていない情報のタイムスタンプ、および、現在コーデックのバッファ内にある情報のタイムスタンプが含まれる）、コーデックバッファが空になるまでの総時間、コーデックのバッファ内で使用可能なバイト数、次にコーデックバッファに追加されるべきデータ（例えば、これは、ＭＰＥＧ－２ビデオを高速で後方に再生中であるとき、前の「Ｉ」フレームである可能性がある）の位置（例えば、次、前、シーク距離）およびタイプ（例えば、フルＧＯＰまたは「Ｉ」フレーム）が含まれる。適切な場合、別々の情報が各コーデック（オーディオ、ビデオなど）に対して提供される。
【０２３６】
UINT32 TRAP_OverlaySetup(UINT32 idx, UINT32 width, UINT32 height);
このオペレーションは、インデックス値ｉｄｘをもってオーバーレイ面を割り振り、このオーバーレイ面は指定された面積を有し、空白である（すべて透明）。
【０２３７】
UINT32 TRAP_OverlaySetPosition(UINT32 idx, UINT32 x, UINT32 y);
このオペレーションは、ディスプレイ上に指定されたオーバーレイ面の（ｘ，ｙ）位置を設定する（ただし、０，０は左側の角）。このオペレーションは実際にはオーバーレイを表示せず、ＴＲＡＰ＿ＯｖｅｒｌａｙＳｈｏｗ（）が呼び出されるときに、オーバーレイがどこに表示されるかを指定するだけである。
【０２３８】
UINT32 TRAP_OverlayShow(UINT32 idx, UINT64 pts);
このオペレーションは、オーバーレイを表示する。ｐｔｓ値は、例えば、オーバーレイを、基礎となるビデオと同期させる際に使用するために、いつオーバーレイが表示されるべきであるかを指定する。
【０２３９】
ＴＲＡＰ＿ＯｖｅｒｌａｙＳｈｏｗに対する複数のコールを（適切なイメージを実際に生成する、パレットを設定するなどのための、他のオーバレイオペレーションに対するコールと共に）使用して、アニメのメニュー、単純なピクチャインピクチャビデオ、および、任意の方法の他のアニメのオーバーレイを作成することができる。
【０２４０】
UINT32 TRAP_OverlayHide(UINT32 idx, UINT64 pts);
このオペレーションは、オーバーレイを隠すが、消去しない。オーバーレイを隠しても、オーバーレイを後に再度表示させるためにオーバーレイは使用可能なままである。オーバーレイは、指定された時間（ｐｔｓ）隠される。
【０２４１】
UINT32 TRAP_OverlayClear(UINT32 idx);
このオペレーションは、オーバーレイを消去する。オーバーレイを消去すると、そのコンテンツは消去し、オーバーレイを再度表示する前に再度セットアップする（ＴＲＡＰ＿Ｏ

ｖｅｒｌａｙＳｅｔｕｐを参照）必要がある。

【０２４２】

UINT32 TRAP_OverlaySetPalette(UINT32 idx, UINT32 *color, UINT8 *trans);

このオペレーションは、オーバーレイのためのパレットを指定する。パレットには、２５６パレット値の各々に対する色（例えば、ＲＧＢまたはＹＵＶ）および透過度（不透明度）が含まれる。パレットを変更して、オーバーレイをアニメ化または修正することができる（例えば、選択された領域の周囲の境界を強調表示するため、半透明のオーバーレイ枠の不透明性を変更するため、イメージを可視にするためなど）。

【０２４３】

UINT32 TRAP_OverlayBitmapDraw(UINT32 idx, UINT32 x, UINT32 y, UINT32 width, UINT32 height, UINT8 *bmp);

このオペレーションは、指定されたオーバーレイ上にビットマップを描画する。オーバーレイ領域の境界を越えるオーバーレイを描画することでエラーが生じるようになり、または、美的に予測できない結果を生じる場合がある。しかし、プレイヤはバッファをオーバーフローさせるべきではなく、または、このような状況において他のセキュリティ違反を引き起こすべきではない。パラメータｂｍｐは、描画すべきイメージを指定する（圧縮されていても圧縮されていなくてもよい）。代替実施形態は描画オペレーションを提供する場合があり、または、拡大縮小および他のイメージ操作を実行する機能が提供されることが可能であり、それによりこれらの（しばしば計算主体の）プロセスをコンテンツコードから取り除く。

【０２４４】

UINT32 TRAP_OverlayTextDraw(UINT32 idx, UINT32 x, UINT32 y, UINT32 size, UINT8 *text);

このオペレーションは、指定されたオーバーレイ上にテキストを描画する。パレット内の最初の１１個のエントリは、テキストの色付けのために使用される（半透明エッジによるアンチエイリアシング（anti-aliasing）を含む）。コンテンツコードはまた、例えば、プレイヤが必要とされたフォント、文字セットなどを欠く場合に、イメージ描画機能を使用して、テキストを表示することもできる。

【０２４５】

テキスト行がオーバーレイ上に適合しない場合、テキスト行は切り取られる。改行のサポートは提供されず、これは呼び出し元が担う。ｓｉｚｅパラメータは、描画すべきテキストのサイズを指定し、プレイヤのデフォルトにすることができる（例えば、ユーザプリファレンス、現在の表示の文字などを反映する）。

【０２４６】

UINT32 TRAP_OverlayRectDraw(UINT32 idx, UINT32 x1, UINT32 y1, UINT32 x2, UINT32 y2, UINT32 color, UINT32 filled);

このオペレーションは、指定されたオーバーレイ上に長方形を描画する。入力パラメータは、オーバーレイインデックス（ｉｄｘ）、左上側の座標（ｘ１，ｙ１）、右下側の座標（ｘ２，ｙ２）、ｃｏｌｏｒ、および、長方形が塗りつぶされるべきであるかどうかを示すブール値（ｆｉｌｌｅｄ）を指定する。

【０２４７】

UINT32 TRAP_SockConnect(UINT8 *name, UINT32 port);

このオペレーションは、ｐｏｒｔによって指定されたポート上のｎａｍｅによって指定されたアドレスに対して、ソケットベースのネットワーク接続を開く。ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用してソケット設定を構成し、ネットワーク接続が現在使用可能であるか、（例えば、モデムを介してダイヤルしようと試みることにより）ネットワーク接続が潜在的に使用可能であるか、または、確実に使用可能でないかを判断することができる。ネットワーク接続の存在および信頼性は、実施態様によって決まる（例えば、携帯電話と統合されるポータブルデバイスは、ネットワーク接続を有する可能性が高いが、まったく無線接続性サポートを有していないものは、そうでない場合がある）。

【０２４８】
ソケット／ネットワークオペレーションのためのプレイヤサポートは任意であるが、このサポートは一貫性のあるインターフェースを提供するように標準化されて、そのインターフェースを介してコンテンツは、リモートネットワークリソースが使用可能であるときリモートネットワークリソースにアクセスすることができる。ソケットサポートは非同期であり、同期アクセスを必要とするコンテンツは、必要とされたデータが検索されるまでポーリングすべきである。ネットワークパラメータの検出および構成は、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙおよびＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを介して実行される。
【０２４９】
UINT32 TRAP_SockClose(void);
このオペレーションは、現在開いている接続を閉じる。
【０２５０】
UINT32 TRAP_SockRecv(UINT8 *buf, UINT32 len);
このオペレーションは、開いている接続からのｂｕｆ内のデータを受信する。
【０２５１】
コンテンツコードは、受信したデータがどのように使用されるかを制御する。例には、フレッシュセキュリティコードを得ること、失効状況をチェックすること、支払いを処理すること、商品／サービスを見て回ること（および購入すること）、ボーナスコンテンツをダウンロードすること、アップデートされたオファー／広告（価格および購入情報を含む）をダウンロードすること、マルチユーザインタラクティブシステム（ムービーウォッチャチャットなど）を実施すること、ウェブブラウジング（任意で、プレイヤ実装ウェブページレンダリングおよび／またはブランジング機能の支援付き）などが含まれるが、これらに限定されるものではない。
【０２５２】
UINT32 TRAP_SockSend(UINT8 *buf, UINT32 len);
このオペレーションは、ｂｕｆによって示されたデータを、開いている接続に送信する。
【０２５３】
UINT32 TRAP_DeviceDiscovery(UINT32 dev, UINT32 qID, UINT8 *buf, UINT32 *len);
プレイヤ環境ディスカバリプロシージャ（ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙおよびＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓ）は、再生環境についての情報をコンテンツコードに提供し、環境を制御する能力を提供する。例えば、コンテンツは、プレイヤ設定を決定すること（デフォルトプレイヤ言語、音量、輝度、コントラスト、出力解像度など）、どの選択可能なプロシージャコールがサポートされるかを発見すること、再生環境構成を決定すること（アタッチされたデバイス、アクティブなソフトウェアドライバ／プログラム、再生環境コンポーネントの状況など）、出力デバイスを認証すること、コンポーネントを探し出すこと（ローカルに存在するか、直接接続されるか、ネットワークを介して接続されるか、など）、および、オプショナルデバイスまたはリムーバブルデバイスにアクセスすること（ただし、これらに限定されない）などの、オペレーションを実行することを望む場合がある。
【０２５４】
ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙは、ｄｅｖによって指定されたデバイスについての、ｑＩＤによって指定された質問に対する答えを提供する。再生環境についての情報を知るために、コンテンツコードは各デバイスに標準質問のリストを求めることができる。サポートされた質問の例には、どのｑＩＤ（質問ＩＤ）がデバイスによってサポートされるか、デバイスの親および子デバイスがある場合、そのデバイスＩＤは何であるか、システムにおけるデバイスの役割は何であるか（ストレージ、ユーザインターフェース、ビデオディスプレイ、スピーカ、ポータブル、ネットワーキングコンポーネントなど）、デバイスの識別は何であるか（シリアルナンバ、メーカ情報、モデル、バージョン、日付、有効期限など）、デバイスの証明書（および、他の暗号化データおよび機能）は何であ

るか、どの暗号化キーをデバイスは含むか（またはどの暗号化キーへのアクセスを有するか）、デバイスのどの役割（記録、再送信、表示など）が現在使用中であり、どの役割がアクティブでないか、どのプロトコルをデバイスはサポートし、どのプロトコル（およびプロトコルパラメータ）が現在使用可能であるか、デバイスは実行可能コードの受信をサポートするか、およびそうである場合、実行環境特性は何であるか（インタープリタまたはネイティブ、サポートされた仕様バージョン、マルチスレットまたはシングルスレット、セキュリティ認証）、デバイスの現在のタイマ値は何であるか（時間帯などを含む）、どの構成可能パラメータをデバイスはサポートするか、およびそれらの現在の値は何であるか、デバイスのステータスは何であるか、が含まれるが、これらに限定されるものではない。

【０２５５】

標準質問リストは時間とともに拡張する場合があり、ある質問（ｑＩＤ）が標準質問リストの一部になる前に製造されるデバイスは、そのｑＩＤによる問い合わせに対し、エラーコードＳＴＡＴＵＳ＿ＱＵＥＳＴＩＯＮ＿ＵＮＫＮＯＷＮをもって応答するようになる。また、標準質問リスト上の質問は常にｑＩＤの最上位ビット（ビット３１）を消去させることにも留意されたい。この最上位ビットセットを有する質問ＩＤは、拡張されたデバイス固有の質問のために予約される。（特定のデバイスについての拡張された情報を発見するように設計されるコンテンツコードは、そのデバイスの拡張されたｑＩＤリストおよび応答方法を知っていなければならない。）

【０２５６】

呼び出し元ルーチンは、ｌｅｎによってインデックスの付けられた位置として、ｂｕｆによって示されたバッファの最大長（バイト単位）を渡すべきである。ｌｅｎ内の返された値は、実際にｂｕｆに書き込まれたバイト数を示す。

【０２５７】

デバイスがシステムに接続されるとき、プレイヤはデバイスＩＤをそのデバイスに割り当てる。デバイスをシステムから切断しても、そのデバイスＩＤは失われない、または、そのデバイスＩＤは別のデバイスに再び割り当てられない。デバイスＩＤゼロはメインプレイヤ自体のために予約される（しかし、プレイヤは追加のＩＤも有する場合がある）。新しいメディアが挿入されるか、または、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙがゼロのデバイスＩＤ（ｄｅｖ）をもって呼び出されるまで、デバイスＩＤは持続する。ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙが、ゼロに設定されたｄｅｖおよびゼロに設定されたｑＩＤをもって呼び出されるとき、現在割り当てられているデバイスＩＤのすべてが廃棄され、適宜、すべてのデバイスの再スキャンが実行され、デバイスＩＤが新たに割り当てられる。デバイスまたはデバイス構成における変更は通常、（ＴＲＡＰ＿ＥｖｅｎｔＧｅｔによって得られる）イベントを生成する。

【０２５８】

ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙによって提供された情報は、再生するかどうかを判定する際に、どのようにまたは何を再生すべきであるかを判定する際に、復号化キーまたは他の暗号化パラメータを導出する際に、後にコードとして実行されるようになるデータを変換する際に、互換性の問題を識別（または対処）する際に、リスクを査定する際に、プレイヤ機能を決定する際に、ユーザプリファレンスを識別する際に、ならびに、他のセキュリティおよび非セキュリティ役割を実行する際に、コンテンツコードによって使用される場合がある。

【０２５９】

ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙはシステムの状態を変更すべきではなく、デバイスの状態を変更するオペレーションはＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用すべきである。

【０２６０】

UINT32 TRAP_DeviceAccess(UINT32 dev, UINT32 opID, UINT8 *buf, UINT32 *len);

このオペレーションは、ｏｐＩＤによって指定されたデバイスオペレーションが、ｄｅｖ

によって指定されたデバイスによって実行されることを要求する。オペレーションは通常、デバイスに固有である。このＴＲＡＰ（および／またはソケットＴＲＡＰ）は通常、デバイス間またはデバイス内でのメッセージの受け渡しのため、ならびに、構成パラメータを設定し、再生環境を全体的に管理するために使用される。渡されるパラメータの内容は、ｏｐＩＤによって決まる。バッファ（ｂｕｆ）は、ｏｐＩＤに応じて、情報をコンテンツから渡される、もしくはコンテンツへ渡す、またはその両方のために使用することができる。

【０２６１】

ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して、非常に幅広い範囲の機能を実施することができる。ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して、例えば、セキュリティパラメータ（データがどのように物理的メディアからロードされるかに影響を及ぼす、キーおよび他のパラメータなど）を指定することができる。このオペレーションはまた、リモートデバイスおよび他のコンポーネント（ハードウェアおよびソフトウェアコンポーネントの両方を含む）と対話するためにも使用される。この対話には、データの送信、データの受信、キー合意の実行、失効状況の決定、認証の実行、構成状態の変更、電源管理機能の制御、プロセスの終了／制御などが含まれるが、これらに限定されるものではない。

【０２６２】

例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用して、映画のコピーをリモートデバイスに転送することができる。送信元デバイス上で実行されるコンテンツコードは最初に、転送のための送信先デバイスおよび適切なオペレーションＩＤを識別することができる。コンテンツコードはまた、任意のデバイス認証、キー交換、または他の必要とされたセキュリティオペレーションも実行する。（セキュリティ解析は、転送に先立って一般に実行されるが、キー配布および他のセキュリティ関連プロセスが、追加で、または代替として、メインデータ転送中またはその後に実行される場合がある。）次に、コンテンツコードは映画の諸部分を（受信側によってサポートされる場合は、任意で、解釈可能コードと共に）リモートデバイスに提供する。送信されたデータのフォーマットはコンテンツコードおよび送信先デバイスによってネゴシエートされ、最終的にコンテンツコードによって制御される。例えば、送信を行っているコンテンツコードは、送信先デバイスによってサポートされた複数のフォーマットから選択し、コンテンツをスケール変更またはそうでない場合は変換し、フォレンシックマークをコンテンツ内に埋め込み、コンテンツを送信先デバイスに提供することに先立って復号化／再暗号化することができる。コンテンツと共に送信されるコードはまた、送信元デバイスによって現に解釈されているコードとは異なる場合もある（例えば、インタープリタが異なるタイプである場合、異なる役割を実行する場合など）。コード、キー、データおよび他の部分もまた削除または修正される場合があり、これは例えば、後続のコピーを作成する能力などの、受信側デバイスによってサポート（、または、受信側デバイスによって実行されるように許可）されない機能を除外するためである。コードまたはデータを暗号化された形式で提供することもでき、これらは、送信元コンテンツコードによって復号化される必要はない。（場合によっては、送信元デバイスは、復号化キーへのアクセスさえ有することができない。）通常、メディアの諸部分におけるコンテンツは、いずれかの必要なセキュリティ処理または他の変換がなされて、送信元によってロードされ、次いで出力される。受信側デバイスは、インタープリタコードを実行中である場合もある。例えば、デバイス間転送は、最初に所定の初期コードを受信側に送信することを含む場合があり、この初期コードは、受信側においていずれかの必要とされたセキュリティチェックを実行し、次いで受信後続データを管理する。（受信側では、例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを使用してデータを受信し、例えばファイルシステム内に格納することができ、または、ＴＲＡＰ＿ＣｏｄｅｃＯｕｔｐｕｔを使用してデータを表示することができる。）非プログラマブルデバイス（non-programmable device）に送信中であるときでも、送信元コードはセキュリティおよび失効チェックを実行し、インターフェースプロトコルのセキュリティ部分を管理することなどができる。状況によっては（例えば、デバイスおよび転送プロトコルがコンテンツコードよ

りも新しい場合）、コンテンツコードは、必要および／または実行可能な範囲でプロセスを監視しながら、デバイスが（例えば、セキュリティプロセスの詳細にアクセスすること、および、セキュリティ問題が識別される場合に転送を防止することにより）転送プロセスのセキュリティを管理することを可能にすることができる。転送には、ＮＶＲＡＭスロット書き込み、メディア書き込み、外部サーバとのハンドシェーク、または他のプロセスが伴って生じ、例えば、「コピーワンス」ポリシなどの制限を実施することができる。

【０２６３】

UINT32 TRAP_RunNative(UINT8 *signature, UINT32 sigLen, UINT8 *code, UINT32 codeLen);

このオペレーションは、コンテンツコードがネイティブコードをプレイヤ上で実行することを可能にするように意図される。プレイヤは、ネイティブコードがプレイヤのメーカまたは別の信頼できるパーティによってデジタル署名されることを要求する場合がある。（これは、悪意のあるコンテンツが悪意のあるネイティブコードを実行することを防止するために行われる。）ネイティブコードのオペレーションは通常、プレイヤ実装（または、最終的にネイティブコードを実行することになる他のデバイス）に固有である。結果として、コンテンツコードは通常、プレイヤの特性（例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙから決定される）についての情報を使用して、どのネイティブコードがプレイヤによって必要とされるか（または反対に、プレイヤがあるネイティブコードとの互換性を有するか）を判断する。

【０２６４】

ＴＲＡＰ＿ＲｕｎＮａｔｉｖｅの使用の実施例には以下のものが含まれるが、これらに限定されるものではない。

【０２６５】

・（例えば、アップデートされたコードを家電デバイスに含まれる不揮発性メモリに書き込むこと、ＰＣ上にインストールされたソフトウェアプログラムを修正することなどにより、）ソフトウェアアップデートをプレイヤまたはシステムの他の部分にインストールすること。

【０２６６】

・プレイヤ内に格納されたキーまたは暗号化パラメータをアップデートすること。

【０２６７】

・悪意のある（または、潜在的に悪意のある）ソフトウェア（ビデオドライバとして装う（masquerade）ビデオキャプチャソフトウェア、光ディスクドライブまたは他のメディア入力デバイスとして装うメディアエミュレーションソフトウェア、実行中のプログラムを改ざんするために使用されるデバッガ、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙによって正しく報告されないいずれかのアクティブなソフトウェア、有効な失効されていないデジタル署名を欠くいずれかのソフトウェアなど）のために再生環境（メモリ、ストレージなど）をスキャンすること。

【０２６８】

・悪意のある（または、潜在的に悪意のある）変更がデバイス上でなされているかどうかを検出すること（無許可のファームウェア変更、ＦＰＧＡの再構成、ハードウェアコンポーネントの取り替えなど）。

【０２６９】

・証明書、公開キー、対称キー、および他の暗号化属性（以下参照）が与えられると、デバイスが予想されるタイプであることを検証すること。

【０２７０】

・バグ（プレイヤのオーバーレイ／メニュー機能における欠陥、メモリリーク、メモリ破損の問題、間違ったインタープリタオペレーション、パフォーマンスの問題、セキュリティポリシ解釈の欠陥など）に対する次善策を提供すること。

【０２７１】

・（例えば、別の方法でアクセス不能なシステムについての情報を得るため、または、既

存のＴＲＡＰによってはサポートされない方法でシステム状態を修正するために、）ペリフェラルへの直接アクセスを可能にすること。

【０２７２】

ＴＲＡＰ＿ＲｕｎＮａｔｉｖｅは、敵対者が、暗号化キーを正当なプレイヤのタイプから抽出し、これらのキーを悪意のあるソフトウェアプログラムにより使用するための方法を発見する状況において、セキュリティを再確立する助けとなることができる。このシナリオでは、すべての脆弱なプレイヤのキーを失効させることは、通常は実現できない。なぜならば、それらの機器が失効されてがっかりするような正当ユーザが多数のこのようなプレイヤを所有するようになるからである。

【０２７３】

例示的メディアはキー抽出攻撃への対策を保持し、以下のように動作するコンテンツコードを含む。

【０２７４】

最初に、メディア上のコンテンツコードは、プレイヤの証明書を検証することによって、プレイヤの疑わしいシリアルナンバを判定する。プレイヤシリアルナンバは、プレイヤのタイプ、および、プレイヤが保持する特定の暗号化キー（例えば、ＴＲＡＰ＿Ａｅｓ内）の両方を暗示する。（一部の例示的キー管理方法を、「例示的対称キー管理」のセクションで説明する。）次に、コンテンツコードは、プレイヤが知るべきキーを知っていることを検証する。（例えば、ｓｒｃ、ｄｓｔ、ｋｅｙ、およびｏｐＯｒＫｅｙＩＤをもってＴＲＡＰ＿Ａｅｓを呼び出すことにより、これを実施することができる。ｓｒｃおよびｄｓｔは、再生のために必要な暗号化されたコードまたはデータを含むバッファを示し、ｌｅｎはバッファの長さを含み、ｋｅｙは、関連するプレイヤキーにより暗号化されているバッファの復号化キーの値を示し、ｏｐＯｒＫｅｙＩＤはプレイヤキーを参照する。複数のＴＲＡＰ＿Ａｅｓコールを実行することができ、これには、後続のオペレーションへの入力を定式化するために、以前のオペレーションからの出力が使用される場合が含まれる。他のステップも含まれる場合があり、一定の入力をもってＴＲＡＰ＿Ａｅｓオペレーションを実行して、次いで、「修正（fixup）」値をＸＯＲまたは加算することによって結果を「訂正する」ことなどである。）脆弱なプレイヤのキーでは、キー検証ステップは失敗する（または、少なくとも完全に完了しない）べきである。なぜならば、これらのプレイヤは、再生が脆弱なプレイヤ上で実行されている状況と、脆弱なプレイヤから抽出されたキーを使用して悪意のあるプレイヤ上で再生が実行されている状況とを区別するために、追加の認証を必要とするからである。これは、エミュレートされたプレイヤと正当なプレイヤとを区別し、次いで、（例えば、結果の値を返すこと、インタープリタのメモリ領域の部分を結果に基づいて復号化またはそれ以外の方法で処理することなどにより）結果に従ってインタープリタ状態を修正するように構成されたネイティブコードを実行させるＴＲＡＰ＿ＲｕｎＮａｔｉｖｅを実行することによって、実行することができる。代替として、または加えて、ネイティブコードは、例えば、外部デバイスと通信すること、出力されるものを修正すること（例えば、フォレンシックマークを導入するため）などによって、インタープリタの外部で効果を有することができる。次いで、コンテンツコードは、正しい再生のために、例えば、ビデオまたは必要とされるコード／データのための復号化キーを導出する際の結果を組み込むことにより、またはそうでない場合はその結果を要求することにより、ネイティブコードによって実行されたオペレーションを要求することができる。

【０２７５】

エミュレータが完璧でない限り（現代のコンピューティングデバイスの複雑さを考えると、極度に困難な提案）、ネイティブコードがうまく区別することは可能となる。ネイティブコードが正当なプレイヤをエミュレートされたプレイヤから区別するために使用することができる特性の例には、特定のタイプのネイティブコードを実行する能力、ハードウェアレジスタ内に格納された値、ハードウェアレジスタを修正することに起因する効果、メモリコンテンツおよびチェックサム／ハッシュ、非標準オペレーションを実行するときの

挙動、ベンダ固有のセキュリティ機能の正しい実装（例えば、ビデオデコーダチップにおける文書化されていないセキュリティ機能）、コーデックの丸め誤差、エラー状態の処理、選択可能なユーザインターフェースコンポーネント（ＬＣＤディスプレイまたはキーボード入力など）、正当なデバイス上にない機能の存在、および、オペレーションのパフォーマンスおよびタイミングが含まれるが、これらに限定されるものではない。例示的実施形態では、１つまたは複数のプレイヤセキュリティテストを実行し、テスト結果に基づいてキーを導出し、導出されたキーを使用してインタープリタのメモリ領域の一部を復号化するネイティブコードを、コンテンツは指定する。コンテンツコードは次いで、ネイティブコードのチェックのすべて（または、十分に多数）によりプレイヤが正当であると示される場合、正しく再生するように構成される。プレイヤが正当ではない場合、コンテンツコードおよび／またはネイティブコードは、再生を停止し、エラーを報告し、追加の認証を要求し、プレイヤアップグレードを要求し、映画の終了をデコードすることを拒否し、ボーナス機能を無効にし、低下した解像度で再生し、または、再生環境に関連付けられたより高いリスクを反映する他の方法で応答することができる。

【０２７６】

UINT32 TRAP_VendorSpecific(UINT32 select, ...);

このオペレーションは、プレイヤメーカが自身のセキュリティおよび非セキュリティオペレーションのためのサポートを追加することを可能にするように意図される。例えば（ただし、以下に限定はされない）、一部の実施形態は、ファイルアクセス（開く、読み取る、書き込む、シークする、閉じる、属性を設定するなど）、ベンダ固有のセキュリティオペレーション（主要な暗号化機能が危殆化される場合のバックアップとしての機能を果たすことができる非標準暗号化サポート、補助暗号変換、ハードウェアビデオデコード回路におけるセキュリティ機能など）へのアクセス、ＴＲＡＰ＿ＤｅｖｉｃｅＡｃｃｅｓｓを通じて使用可能ではない特殊機能（３次元表示、香りの出力、ジョイスティックの振動など）へのアクセス、生フレームバッファコンテンツへのアクセスなどを提供することができる。

【０２７７】

（統合されたセキュリティサービス）

例示的実施形態では、ライセンス団体（または他のエンティティ）がコンテンツ作成者のための統合されたセキュリティサービス一式を提供することができる。コンテンツ所有者自身がセキュリティサービスを提供することができるが、これらの機能をサードパーティにアウトソーシングすることは、コンテンツ所有者がプロセスに関与する必要性を低減することができると同時に、例えば、あるセキュリティプロバイダが複数のコンテンツ所有者にサービスを提供することを可能にすることによって、規模の経済（economy of scale）を可能にする。セキュリティサービスが単一のエンティティによって提供されるか、複数のエンティティによって提供されるかにかかわらず、関連のタスクには以下のこと(ただし、これらに限定されない)を含めることができる。

【０２７８】

・オーディオビジュアルコンテンツを、セキュリティ対策、復号化ロジック、フォレンシックマーク埋め込み機能、失効ポリシの実施、不揮発性メモリ機能との統合、および、非セキュリティ機能ユーザインターフェースとの統合と結合する、セキュリティコードまたはサービスを開発すること。

【０２７９】

・自動化された検索ツールおよび／または手動のプロセスを使用して、コンピュータネットワークまたは他の流通経路を介して入手可能な海賊行為を受けたと考えられるマテリアルを探し出し、海賊コピーを手動でおよび／または自動で検査および解析して、フォレンシックマーク内に埋め込まれた情報を復旧し、次いで、復旧された情報を使用して、海賊行為者のデバイスまたは方法についての情報を収集すること。

【０２８０】

・例えば、海賊ソフトウェア／デバイスおよび海賊行為者間の議論（例えば、オンライン

チャットルームにおける）を解析することによって、海賊行為についての他の情報を収集、アーカイブおよび解析すること。

【０２８１】

・海賊デバイスを失効させること、海賊攻撃に対する対策を公式化すること、疑わしい海賊行為者に対する起訴を支援すること（ただし、これらに限定されない）などにより、海賊行為を軽減する助けとするために収集された情報を使用すること。

【０２８２】

・例えば危殆化されている、安全でない、海賊行為に関与していると疑われるか、またはそうでない場合はコンテンツを復号化するために使用されるべきでないキー、プレイヤおよび他のコンポーネントの失効リストを管理すること。このようなリストを電子的に管理することができ、これらのリストは、各エントリの状況に関連付けられた複数のパラメータを含む（ただし、これに限定されない）ので、個々のタイトル、プロセスなどに関連付けられたリスクプロファイルを満たすように失効データをカスタマイズすることができる。

【０２８３】

・例えば、失効していないデバイスによってのみコンテンツを復号化することができるようにコンテンツを作成するために、ブロック復号化オペレーション（ＴＲＡＰ＿Ａｅｓなど）用の入力／出力を生成または取得すること。

【０２８４】

・特定のプレイヤに対する攻撃を含む、攻撃への対策を開発すること、または管理すること。例えば、ネイティブコードベースのセキュリティ対策がプレイヤメーカによって開発およびデジタル署名されなければならないフォーマットで、このようなサービスには、このような対策が必要とされる状況を識別すること、リスクについての情報をメーカに提供すること、対策開発を支援すること、対策のためのテスト（セキュリティおよび互換性のテストを含む）を提供すること、各対策が使用されるべき再生環境を識別するためにコンテンツによって使用するためのコードまたはポリシを開発すること、対策コードを他の復号化プロセスおよび他のセキュリティ機能と統合すること（例えば、対策がスキップまたは回避されないようにするため）、および、メディア上の複数の別々のプレイヤタイプ／メーカからの対策を結合すること、が含まれる場合があるが、これらに限定されるものではない。

【０２８５】

・キー、コード、識別子、セキュリティポリシおよび他の属性を含む、不揮発性メモリスロットに関連付けられたデータを管理すること。

【０２８６】

・オンラインまたは他のネットワークベースまたはインタラクティブなコンポーネントと共に動作すること、または、これらと統合すること（例えば、デコードが、問題のあるセキュリティを有する１つまたは複数のデバイスまたは環境と関連するときに、拡張ユーザ認証ステップまたはプレイヤ認証ステップを提供するため）。

【０２８７】

・完成されたメディアの品質管理を実行すること。

【０２８８】

・コンテンツを、個々のプラットフォーム上にある特別なマーケティング機能、ユーザ機能、非標準機能などと統合すること。

【０２８９】

・プレイヤ互換性テストを実行すること。

【０２９０】

・問題（バグのメニュー化（menuing bug）、コーデック制限、ディスプレイ制限、正常に機能しない機能、是正可能なセキュリティ欠陥などが含まれるが、これらに限定はされない）を有するプレイヤを検出し、適切な次善策（影響を受けたオペレーションを回避すること、影響を受けない簡素化された機能を使用すること、プレイヤパッチをインストー

ルすること、および、ネイティブコードを実行して問題に対処することが含まれる場合がある）を利用するためのコードの開発および統合。

【０２９１】

・レプリケータと統合して、上述の機能を実施するために適切なコードおよびデータと共にメディアが正しく作成されることを確実にすること。および／または、

・いずれかの方法の他のタスクを提供して、コンテンツ所有者、プレイヤ開発者、ユーザ、法執行機関または他のパーティを支援すること。

【０２９２】

（フォレンシックマーキング埋め込みおよび復旧技術）

比較的大規模の共謀攻撃（すなわち、複数の敵対者が、そうでない場合は共謀デバイスを識別および失効させるために使用することができるフォレンシックマークを消し去る意図を通常は有して、複数のデコードデバイスからのデコードされた出力を結合することによる攻撃）に対して最良の可能なセキュリティを得るため、比較的多数のバリエーションを出力において導入する能力を有することが、しばしば有効である。完全なＭＰＥＧ－２ビデオＧＯＰ用のまったく別のバージョンを格納することは確かに可能ではあるが、多数の代替ビデオシーケンスが格納されなければならない場合（映画全体にわたる数百万の代替など）、ストレージ要件は厳しく（prohibitive）なる。このような状況に対応するため、必要とされるストレージの総量を最小限に抑えることが有効である。

【０２９３】

フォレンシックマークのために必要とされる特定の要件および属性は状況に応じて変わるが、望ましい属性には通常、基礎となるバリエーションがコンパクトに表現されること（バリエーションをエンコードするために、比較的わずかなストレージ空間または帯域幅が必要とされることを意味する）、耐久性があること（コンシューマカムコーダ（consumer camcorder）を使用してテレビ画面から記録することによって作成されたコピーなどの品質が低下したコピーからバリエーションを復旧することができることを意味する）、もっともらしい（plausible）こと（出力を調べることによって、それらを自動的に識別および除去することができないことを意味する）、および、これらが芸術的なものとして受け入れ可能（acceptable）であること（バリエーションが過度にコンテンツの品質または履歴（experience）を損なわないことを意味する）が含まれる。これらの要件は（例えば、タイトル、コンテンツ作成者の要件、再生環境特性などに応じて）変わる場合がある。

【０２９４】

フォレンシックマーキングをサポートする例示的光ビデオディスクでは、コンテンツ作成者は、圧縮されたビデオに適用される場合のある複数の代替（または他の修正）を探し出し、または生成する。これらの代替はビデオ圧縮プロセスの一部として識別され、この場合、考えられるバリエーションを識別し、次いで複数の基準（例えば、ストレージサイズ、耐久性、もっともらしさ、芸術的受け入れ可能性など）と照らし合わせてそれらの適合性を評価するように、ビデオ圧縮プロセスが修正される。バリエーションが適切である場合、適切なバリエーションとして出力される。例えば、単一ビット（または単一バイト、単一ワード、単一ブロックなど）の値を変更することが適切な代替ビデオシーケンスを作成する位置を識別するように、圧縮ロジックを構成することができる。

【０２９５】

代替実施形態では、従来の圧縮アルゴリズムを使用することができ、圧縮後ステップとして、代替を生成し、かつ、代替の妥当性を検査することができる。例えば、自動化されたソフトウェアを使用して、（ａ）例えば、（擬似）乱数生成器を使用して、圧縮されたビデオの任意のバイトと、そのバイトのための新しい値とを選択することによって、ビデオストリームへの修正候補を識別すること、（ｂ）ビデオの試験的な圧縮解除を実行し、ストリームが変更によって無効にされた場合、修正候補を無効として拒否すること、および、（ｃ）試験的な圧縮解除の出力を、オリジナルのストリームを圧縮解除した結果と比較し、差異が要件を満たさない場合、修正候補を拒否することができる。例えば、ステップ（ｃ）では、小さすぎる（すなわち、耐久性が十分でない）または大きすぎる（すなわち

、もっともらしくない、および／または、芸術的に受け入れ可能ではない）修正を拒否することができる。追加のテストを実行して、修正が相互に作用しないようにすることを検証することができる（例えば、状況によって、別々に適用された場合、修正は受け入れ可能である場合があるが、共に適用された場合、受け入れ可能でない）。テストを通過する修正は保存される。

【０２９６】

フォレンシックマーキングを使用する例示的メディアを作成するとき、最初の圧縮されたビデオデータストリームが準備される。このデータは、メインの圧縮されたビデオからなり、任意で、すでに適用された一部の修正を伴う（任意で、ビデオを破損する一部の代替または他の修正が含まれる場合がある）。ビデオに対する有効な修正を識別するデータ構造が準備され（例えば、以前に識別されたが適用はされなかった修正を適用する、または、すでに適用されている修正を取り消す）、これらの有効な修正には、ビデオを再生可能にするために必須であるいずれの修正も含まれる。代替を非常にコンパクトにして（例えば、ＭＰＥＧ ＧＯＰにおける２４ビットバイトオフセット、および、８ビット代替値を４バイトで表現することができる）、文字通り（literally）数百万の変更がわずか数メガバイトのデータ内で定義されることを可能にすることができる。

【０２９７】

コンテンツ作成者は次いで、再生環境特性に基づいてバリエーションを選択および適用するためのプログラムロジックを準備する。このプログラムロジックは、変更をビデオシーケンスに適用するように構成されるべきであるが、ただし、適用される変更の組合せは通常、科学捜査の目的のために有用となるであろう情報に基づく（シリアルナンバ、デバイスを暗号的に識別する値、モデル／メーカ情報、接続されたデバイスについての情報など）。加えて、このプログラムロジックは、有効なビデオストリームを作成するために必要ないずれの変更も適用するように構成されることが可能であるべきである（少なくとも、受け入れ可能な再生環境において動作中であるとき）。（「必須」の変更を有することは、変更を適用するコンテンツコード部分を敵対者が無効にしようと試みる攻撃を防止する助けとするために、有用である場合がある。）コンテンツコードロジックには暗号技術の使用が含まれる場合があり、これは例えば、プレイヤキーを使用して、代替を適用すべきプレイヤ上でのみ、代替セットを復号化するためである。同様に、コンテンツコードにはエラー訂正コードの使用が含まれる場合があり、これは例えば、マークが付けられたビデオを復旧するパーティが、（例えば、攻撃または品質低下により）マーク復旧が信頼できない場合でも、プレイヤ情報を復旧することができるようにするためである。埋め込まれる値には、デジタル署名、ＭＡＣ、または、マークを復旧するパーティが復旧された情報の妥当性において確信を有することができるように埋め込まれた情報を認証する、他の値が含まれる場合がある。

【０２９８】

海賊コピーが復旧されるとき、コンテンツ作成者はこのコピーを解析して、現存するバリエーションを識別することができる。この解析を手動で行うことができるが（例えば、海賊ビデオのフレームを、各位置における可能なバリアント（variant）の各々と比較し、次いで、バリエーションのリストを再度再生環境特性にマッピングすることによる）、この解析プロセスを、より効率的にするために、および、より複雑なマークの解析を可能にするために、自動化することができる。例示的な自動フォレンシックマーク解析システムは、カスタマイズされたソフトウェアを備えたコンピュータを使用して実装される。解析ソフトウェアは、オリジナルの圧縮されたビデオ、修正のリスト、および、復旧した海賊ビデオのデジタル表現をもって開始する。次に、解析ソフトウェアは、オリジナルのビデオ内のフレームと海賊ビデオ内のフレームとを照合し、海賊ビデオのフレームとオリジナルのビデオ（および／または、修正が適用されたオリジナル）の対応するフレームとを最も近くマッチさせるための変換（回転、拡大縮小、ベンド、カラーシフト、強度調整、タイムシフトなど）を特定する。解析ソフトウェアは次いで、海賊ビデオのフレーム（またはフレームの諸部分、またはフレームのセットなど）を、オリジナルのビデオのバリアン

トの各々の対応する部分に対して比較する。ソフトウェアは次いで、復旧されたビデオが各バリエーションにどの程度マッチしているかを示す、類似／相違に基づいたスコアを計算する。例えば、（変換された）海賊ビデオイメージとオリジナルのバリエーションとの間の最小二乗誤差を使用して、このスコアを計算することができる。スコアを計算する際、変更によって影響を受けない領域を無視することができ、インターレーシング（interlacing）、フレーム間のぼかしなどのひずみ（distortion）のためにさらなる調整を行うことができる。イメージ解析ソフトウェアはまた、オペレータが海賊ビデオのフレームを（変換の調節の有無にかかわらず）オリジナルのビデオのバリアントに隣接して（バリエーションを視覚的に探し出すことを支援するための選択可能な拡大および強調表示によって）表示することを可能にする「手動モード」ユーザインターフェースを提供することもでき、また、オペレータが、どのバリエーションが海賊ビデオ内にあるかを選択し（または、バリエーションが未知であることを示し）、フレームを各ビデオソース内で前方および後方に移動させ、次のバリエーションまで高速に進め、現在のプログラム状態を保存することを可能にするためのユーザインターフェースオプションを提供する。イメージ解析プロセスの出力は（手動で実行されるか、自動で実行されるかにかかわらず）、ビデオ内に存在する場合のある可能なバリエーションに割り当てられた相対スコアのリストである。（これの簡素化された形態は、出力が単に、各バリアントについての最も可能性の高いオプションを識別する場合である。）自動化された、および／または、手動のプロセスは次いで、復旧したバリエーションデータを再生環境についての情報に変換するために使用される。例えば、マーク埋め込みが、元々はプレイヤシリアルナンバに適用されたエラー訂正コードの出力に基づいていた場合、復旧された選択値を、シリアルナンバを復旧するようになる適切なデコードプロセスへの入力に変換することができる（エラーの数が過剰でないと仮定する）。同様に、選択プロセスが、シリアルナンバの一部をもってシードされた擬似乱数生成器（ＰＲＮＧ）を使用して実行される場合、解析プロセスは、可能なシード値の各々を使用したＰＲＮＧ出力を海賊ビデオからの監視と相関させることを含む場合がある。

【０２９９】

バリエーションはまた、圧縮されたビデオストリームを修正するという手段とは異なる手段を使用して、コンテンツ内に埋め込まれる場合もある。例えば、コンテンツコードは、デコードされたビデオの上にイメージを重ねるようプレイヤに命令することによって、修正を導入することができる。例えば、ＭＰＥＧ－２デコードされたビデオの上に１つまたは複数の半透明オーバーレイフレームを描画することによって（または、同様に、コンテンツコードに出力フレームバッファを直接的または間接的に修正させることによって）、フォレンシックマークを埋め込むことができる。オーバーレイを目立つようにすることもできる。例えば、「名前 名字用スクリーナ。コピーしないでください。（Screener for FirstName LastName's. Do Not Copy.）」と表示される、動的に生成された移動する半透明オーバーレイは、海賊行為への明示的な抑止力を提供することができ、（例えば、光ディスクのバーストカッティングエリア、ＮＶＲＡＭスロット、サーバなどから）視聴者の名前を決定すること、および適切なオーバーレイを表示することによって、これを作成することができる。オーバーレイを使用して、例えば、（意図的に）欠陥のある部分上に描画することによって、圧縮されていないビデオを訂正または修正することもできる。オーバーレイは、メディア上のストレージ空間によって、非常に効率的にすることができる。なぜならば、オーバーレイを描画するためのコードを表すのに必要とされるデータの量を非常に小さくすることができるからである。オーバーレイベースのマークによって、多数のバリアントを有するビデオ部分の生成が効率的になる（例えば、単一のビデオフレームのごく一部を容易に作成して、そのエリア内のプレイヤシリアルナンバの表現を簡単にオーバーレイすることによって、プレイヤを一意に識別することができる）。オーバーレイベースのマーキングは、幅広く変わるマークの作成を簡素化することもできる。なぜならば、（例えば、）淡い半透明なオーバーレイイメージが表示される場合があるとき、スクリーン位置（ｘ，ｙ）および時間のための幅広い範囲のオプションが存在する可能性があ

るからである。海賊行為者が、複数のプレイヤからの出力を結合することによって、それらのコピーのソースを隠そうと試みる場合がある状況においてデバイスを識別する際に、これらの特性は特に有用となる可能性がある。フォレンシックマークを他のデータに埋め込むこともできる。このデータには、オーディオ、静止画像、制御データ、メディア書き込みパターンなどが含まれるが、これらに限定されるものではない。

【０３００】

（メディアおよびコンテンツ認証）

コンテンツコードおよび／またはプレイヤは、メディアが挿入されるとき、および／または、データがその後にロードされるとき、メディアを暗号的に認証することができる。

【０３０１】

例示的実施形態では、メディアの全部または一部の個々のデータ部分（例えば、セクタ）は、ハッシュツリーを使用してデジタル署名される。ルートハッシュは、フォーマットを監督するエンティティ（または別の信頼できるエンティティ）によってデジタル署名され、メディア上に配置される。ルートハッシュ署名はまた、レプリケーションおよび／またはマスタリング施設、著作権所有者、発行日、メディアにアクセスすることを許可される（または許可されない）プレイヤデバイスを記述する基準、および他のこのような情報を識別することもできる。データブロック（例えば、セクタ、ＧＯＰ、トラック、ファイルなど）がメディアからロードされるとき（またはその後）、ロードされたデータを適切な中間ハッシュ値と結合して、ルートハッシュを再作成することにより、これらのデータブロックの妥当性を検査することができ、この妥当性検査はドライブ、コンテンツコード、プレイヤアプリケーション、および／または、他のプレイヤデバイス／部分によってなされる。ルートハッシュの値を、メディア署名が検証された後にキャッシュすることができるので、（比較的遅い）公開キー署名検証ステップを各読み取りごとに繰り返す必要はない。同様に、中間ハッシュ値をデータブロックと共に格納し、キャッシュし、必要に応じて計算し、またはメディアから検索することができる。メディアはまた複数のルートハッシュを含むこともでき、または、他の認証スキームを使用することができる（各セクタ、ＧＯＰ、または他のデータ部分上のデジタル署名を検証することによるなど）。ストレージオーバーヘッドを低減するために、メッセージ復旧を可能にする署名およびパディングスキームを使用することができる。デジタル署名はまた、ディスク上に含まれるブートコードもしくは（再）書き込み可能部分内に格納されたデータ、またはすべてのデータなど、ディスクのコンテンツの部分を認証することもできる。

【０３０２】

署名（または、他の暗号値もしくは非暗号値）はまた、特定のディスクまたはコード部分によって実行される可能性のあるオペレーションを指定または制限することもできる。例えば、コンテンツが特定のプレイヤの特徴または機能にアクセスすることを許可し、可能にする、ライセンス機関によってデジタル署名を発行することができる。このような許可を使用して、ライセンス料が確実に支払われるようにすること、または、無許可の海賊メディアの製造を防ぐことができる。例示的実施形態では、映画を含む光ディスクを大量生産することを望む各コンテンツ作成者（またはそのレプリケーション施設）は、生産されるべきディスクを識別する情報をライセンス団体に提供する。このような情報には、タイトル、合法的な著作権保持者の識別、許可されるべきコピーの数、ディスクコンテンツの少なくとも一部のハッシュ、メディアシリアルナンバ、必要とされる所望の機能、および、要求を認証するデジタル署名を含めることができるが、これらに限定されるものではない。それに応じて、著作権保持者は、メディアの生産を許可する１つまたは複数のデジタル署名を受信する。ライセンス団体はまた、コンテンツ作成者（またはそのエージェント）からの支払いを受け取り、かつ処理することもできる。このように、ライセンス料が、コンテンツによって実際に使用された特定の機能（セキュリティまたはそうでないもの）に直接結合されることが可能である。コンテンツコードのオペレーションに課せられた制限を、（例えば、放送テレビ信号をコピーすることができるかどうかを示すために使用される放送フラグに類似する）非暗号フラグ（non-cryptographic flag）に基づくようにす

ることもできる。パーミッションもまた、（例えば、ユーザのプライバシが確実に維持されるように、プライベートユーザ情報にアクセスしているコンテンツコードによるネットワークリソースへのアクセスを拒否すること、ユーザの許可を得ている、または、リモートサーバからの認証を得ているコンテンツコードに大きなアクセスを付与することなどの）以前のアクションに基づくようにすることができ、異なるスレッド、コンテンツコード部分などでは異なるようにすることができる。

【０３０３】

デジタル署名はコンテンツと共に、またはコンテンツとは別々に配布される場合があり、メディアのいかなる部分上にも配置される場合があり、また、暗号化されたキー、復号化キーを導出するためのロジック、セキュリティポリシ情報などを伴う場合もある。例えば、光メディアの場合では、各ディスクについて別々に書き込むことができるメディアの部分上にデジタル署名を配置することが有用である場合がある（それにより、認証された一意のディスクシリアルナンバを、例えば、メディア失効機能と共に使用するために提供して、どのデバイスがメディアを再生することができるべきであるかなどを明示的に指定する）。

【０３０４】

不揮発性メモリ（ＥＥＰＲＯＭ、フラッシュなど）を含むメディアでは、一意のデータを通常に格納することができるが、他のメディアタイプは他の技術を必要とする場合がある。例えば、スタンプ付き光メディアはバーストカッティングエリア（ＢＣＡ）内のデータを保持することができ、これは通常、強力なレーザを使用して書き込まれる。意図的な欠陥または変更のパターンをメディア表面に導入することもでき、プレイヤおよび／またはコンテンツコードによって読み取ることもできる（例えば、エラー検出およびエラー訂正を無効にすることで得られた読み取り結果を直接処理することによる）。半導体ベースのＲＯＭでは、ヒューズおよび他の追記機能が使用される場合がある。

【０３０５】

署名および証明書に加えて、メディア固有の領域を使用して、ローカリゼーション情報、暗号化されたキー（例えば、特定のプレイヤが現在または将来のメディアを、例えばスクリーナ、デモ、または、規制された配布のために意図された他のディスクとしてデコードすることを可能にするキー、プレイヤ不揮発性メモリデータの読み取り、復号化、および／または書き込みを可能にするキーなど）、暗号化されないキー（例えば、海賊行為者によってレプリケーション施設から盗まれたメディアまたはデータのデコードを防止するために、スタンプ付きメディアの製作の後に書き込まれたもの）、識別情報（例えば、オーバーレイにおける表示のための、および、フォレンシックマーキングにおける使用のための、受信者名など）、マーケティングおよびプロモーションデータ（例えば、トライアル、くじ、オンラインサービスなどへの参加を可能にする値）、電話番号、データネットワークアドレス、実行可能コード、または、コンテンツコードおよび／またはプレイヤによって使用することができる任意の方法の他の暗号または非暗号データも保持することができるが、これらに限定されるものではない。データをディスク固有の領域からロードするとき、ディスク固有の領域のコンテンツを偽造するために必要とされる情報を（例えば、アプリケーションまたはコンテンツコードへ）出力しないように、ドライブは一方向変換（例えば、ＳＨＡハッシュ、コンテンツコードにより指定された値によってキーが付けられた（keyed）ＨＭＡＣ－ＳＨＡ、ＲＳＡ公開キーオペレーションなど）を適用することができる。記録デバイスはまた、ハッシングまたは他の暗号化変換を使用して、敵対者が正当なメディアから抽出された情報を使用して正確な不法コピーを作成することを防止することもできる。メディア固有の領域をまた、記録デバイスによって事前に記録する（例えば、空のコンシューマ記録可能メディア上のシリアルナンバにより）か、または書き込むこともできる（例えば、デジタル署名することができる、記録デバイスの識別子による。この識別子を正当なデバイスによって検証して、レコーダが失効されないことを確実にすることができ、また、違法コピーが復旧されるときにこの識別子を使用して、違法コピーを作成するために使用された記録デバイスを識別し、および／または失効させることが

できる）。
【０３０６】
メディア認証および偽造防止（anti-forgery）技術（例えば、米国特許第６６４６９６７号明細書において概説される、ピット特性またはウォブルトラックコンテンツを調節することなど）をサポートするメディアフォーマットでは、これらの特性から導出された値を、プレイヤによって読み取り（または検出し）、プレイヤおよび／またはコンテンツコードによって認証することもできる。物理的特性を読み取った後、ドライブは、この値を出力する前に、一方向暗号化変換を適用し、修正されないドライブによって受け入れられるような方法で、変換された特性を知る悪意のあるパーティが基礎となる特性を偽造する能力を欠くようにすることができる。コンテンツコード（および／または、ドライブ、プレイヤアプリケーション、または他のコンポーネント）は、（例えば、メディアが信頼できるパーティからの署名を特性値と共に保持することを、検証することにより）変換された特性を認証することができる。
【０３０７】
（ブートストラップ、セキュリティ解析および再生）
例示的光ディスクを伴う例示的プレイヤの実際のオペレーションを説明する。オペレーションは、ドライブにディスクを挿入することで開始する。インタープリタは最初に、コードおよび／またはデータの最初の部分をディスクからロードし、これらを実行することによって、ブートストラップされる（bootstrapped）。この最初の部分を小さく単純にすることができ、例えば、この最初の部分は単に、タイトルがロード中であることをユーザに通知することができ、次いで追加のコードおよびデータをメディアからコンテンツのメモリ領域にロードすることを開始することができる。このコードは追加のコードを順にロードすることができる（例えば、このコードはプレイヤタイプをチェックし、次いでプレイヤタイプに固有のコードをロードすることができる）。プレイヤ設計およびメディアに応じて、任意の数のコードロードチェックを実行することもできる。プレイヤはまた、挿入されたメディアがコード（または、コードを実行するために必要とされた特性）を欠く場合を検出することもでき、そうである場合、プレイヤ内に組み込まれた機能を使用してディスクを再生することもできる。このようなサポートは例えば、ＤＶＤ、ＣＤ、ＤＶＤ－Ａｕｄｉｏなどのレガシーメディアフォーマットからの再生を可能にするために有用である。
【０３０８】
例示的メディアは次いで、ユーザの暗号化インターフェース（例えば、ＴＲＡＰ＿Ａｅｓ）を使用して、１つまたは複数のタイトル固有の復号化キーを導出する。失効していないプレイヤのみが有効な復号化キーを導出することができるように、コンテンツコードは構成される。（これを実施するために、コンテンツコードは、放送暗号化、キー暗号化キー、難読化されたソフトウェアなどの技術を使用することができるが、これらの技術に限定されるものではない。）次いで、これらの復号化キーを使用して、例えば、追加の暗号化されたコード、ビデオまたは他のデータを復号化することができる。ブートストラップにはまた、データをメディアからロードすること、データを必要に応じて圧縮解除すること、および、任意の方法のセキュリティ解析オペレーションを実行することが含まれる場合もある（以下のサブセクションでさらに詳細に説明する）。
【０３０９】
次いで、実際の再生には、通常、様々なステップまたはチェックを繰り返し実行することが含まれ、これらのステップまたはチェックには以下が含まれる場合があるが、以下のものに限定されるものではない。
【０３１０】
・イベントを処理すること。この処理には、ユーザインターフェースイベント（キーストローク、リモートコントロールボタンの押下、マウスの移動、ポインタ／カーソルの移動、選択入力など）および他のイベント（例えば、電源を落とす要求／通知、ハードウェア再構成要求／通知、ソフトウェア変更要求／通知、エラーアラート、メディア排出要求／

通知など）を検出すること、および処理することが含まれる場合がある。（ＴＲＡＰ＿ＧｅｔＥｖｅｎｔを参照。）イベントを処理するための適切な方法は、イベント、タイトル、プレイヤ属性、および、再生プロセスの状態によって決まる場合がある（例えば、メニューが表示されるときにイベントを処理するために使用されるロジックは、ビデオが再生中であるときに使用されるロジックとは異なる場合がある）。一例では、新しいビデオストリームに切り替えること、プレイヤ構成ビットを変更すること、アクションを要求すること（例えば、排出要求に応答して「メディア排出」ＴＲＡＰを呼び出すこと）、メニューをアップデートすること、オーバーレイグラフィックスを変更すること（例えば、アニメ化、アップデートなど）、ネットワーク接続の開始／アップデート／その他を行うこと、再生構成（再生速度など）を変更すること、新しいコンテンツコードをロードし、実行すること、ビデオストリーム内の新しい位置にジャンプすること（これは、バッファされたメディア要求を変更／消去すること、データバッファされたコーデックを消去することなどを必要とする場合がある）、再生を終了すること、セキュリティチェックを実行すること、エラーを表示することなど（ただし、これらに限定はされない）によって、一部のイベントを処理することができる。一部のイベントは、要求されたオペレーションが許可されることをコンテンツコードがチェックすることを、必要とする場合もある。すぐには実行することができないオペレーションは、実行できるようになるまで、バッファされる場合がある。一部のイベントは無視される場合もある。

【０３１１】

・メディアインターフェースを管理すること。例示的実施形態では、メディア処理ルーチン（media handling routine）には、ビデオおよびオーディオコーデックのために、および、他の目的のために、使用可能なデータの安定した供給があることを確実にする責任がある。例えば、メディアが光ディスクである場合、コンテンツコードは、ドライブの状況をチェックし、うまく読み取られているデータを検索し、新しい読み取り要求を提示し、もはや必要ではない読み取り要求を消去し、（例えば、割り込まれない再生を確実にするために、ビデオ内の近づきつつあるブランチの可能な各フォーク（fork）にデータをロードするため）ヒントを提供して機能またはキャッシュを先読みし、コンテンツコードのメモリ領域内（または、他の場所、例えば、コーデック、フレームバッファ、復号化モジュールなど）でデータがロードされるべき場所を指定し、エラーをチェックし、どのようにエラーが処理されるべきであるかを命令し、ドライブまたは他のコンポーネントに対する暗号化キーを指定し、ドライブ読み取り／書き込み要求に関連する認証情報を提供し、デコードパラメータ（エラー訂正情報、キー、セクタアドレスマッピング（sector address mapping）、読み取りヘッド深度／焦点など、ファイルアクセス特権など）を指定することなどができる。

【０３１２】

・セキュリティ処理：このロジックは、キーの導出、ロードされたデータの認証（例えば、ＭＡＣ、ハッシュツリー、デジタル署名などを使用）、および、実行可能部分（例えば、オーディオまたはビデオの特定の部分に関連付けられたセキュリティコード）の実行などの追加の必要とされる復号化または処理ステップのいずれも実行することができる。このステップには、（例えば、ロードされたデータのどの部分を出力するべきかを選択すること、データに修正を加えることなどにより、）フォレンシックマークを埋め込むことが含まれる場合もある。例示的実施形態では、このオペレーションには、ＭＰＥＧ ＧＯＰ用のキーを導出すること、ＧＯＰのデータ上でＡＥＳ復号化オペレーションを実行すること、および、ブロックＡＥＳキーを知っていても敵対者はディスクのコンテンツを復号化できないことを確実にするために、復号化の前／後に前処理および後処理（データの再順序付け（reordering）、ブロックＸＯＲ、置換、バイト修正など）を行うよう、コンテンツコードを使用すること、が含まれる。

【０３１３】

・データをコーデックに転送すること。このコードはまた、エラー状態を検出し、処理することもでき、この状態は、必要とされるメディアからのデータが使用可能ではないため

にコーデックのスターベーション（starvation）を避けることができない状況などである。このコードはまた、コーデック状況を検出して、コーデックバッファがオーバーフローまたはアンダーフローしないことを確実にすること、および、メディアインターフェースコードが正しいデータをロードしていることを確実にすることもできる。

【０３１４】

・オーバーレイおよび特殊な機能を処理すること。このロジックは、メニューの描画および除去、オーバーレイ、サブタイトル、および類似の機能を担う。このロジックはまた、ピクチャインピクチャビデオおよび所定の種類のフォレンシックマーク（オーバーレイを使用して描画されるものなど）も処理する。

【０３１５】

特殊な状況（ゲーム／パズル、メニュー、メニュー選択の処理、隠された「イースターエッグ」など）は、特殊なコードを必要とする場合がある。マルチスレッドインタープリタ（Ｊａｖａ（登録商標）仮想マシンなど）が提供される場合、別々のスレッドが異なるオペレーション（イベント、オーバーレイ、ビデオなどの管理）のために使用される場合があり、スレッドおよび／または共有メモリ領域の間で受け渡されるメッセージのためのインタープリタによるサポートを、スレッド間での同期および制御のために使用することができる。同様に、状況チェックおよびメッセージの受け渡しを使用して、インタープリタが他のコンポーネントと同期されることを確実にすることできる。

【０３１６】

プレイヤはまた、様々なオペレーションのためのデフォルトハンドラを提供して、例えば、ユーザインターフェースの一貫性を改善し、コンテンツ作成努力を低減し、パフォーマンスを向上させることなどもできる。

【０３１７】

完了すると（ユーザがプレイヤ上の排出ボタンを押す場合など）、コンテンツコードに、保留中のシャットダウンを通知することができる。コンテンツコードは次いで、リモートデバイスに通知し、その内部状態を整理し（clean up）（例えば、いずれかの要求された不揮発性メモリ書き込みを完了するなど）、終了することができる。コンテンツコードが所定の期間（例えば、５秒）以内に終了しない場合、プレイヤデバイスはコンテンツコードを終了させ、メディアを排出する。

【０３１８】

（セキュリティ解析オペレーション）

コンテンツコードは再生環境を解析してセキュリティの問題を探すことができる。このプロセスには通常、プレイヤおよび再生環境の他の部分についての情報を得ること、および、それらの情報を処理することが含まれる。この情報には、プレイヤのための、または、コンテンツコードが検証することができる他のコンポーネントのための、デジタル署名および／または暗号化証明書が含まれる場合がある。

【０３１９】

プレイヤから得られたこの情報は、プレイヤおよび再生環境の特性（または、疑わしい特性）を示す。一部の特性（特定のＴＲＡＰの存在など）は直接報告される。他の特性は間接的に推論され、例えば、プレイヤ内の暗号化キーの特定の組合せは、部分的にまたは完全にプレイヤのシリアルナンバによって決定される場合がある。正当なプレイヤとして装うが、正しい暗号化キーを欠くプレイヤは、「嘘で捕まる」可能性がある。なぜならば、プレイヤが有していないキーを使用して暗号化オペレーション（復号化、署名など）を実行することはできないからである。同様に、他の矛盾を利用して、問題を識別することができる。

【０３２０】

セキュリティ解析プロセスは、圧縮されたビデオ、コード、オーディオおよび／または他のデータのためのキーの導出を含む、様々な再生関連プロセスに含まれる場合がある。例えば、特定のタイプの正当なプレイヤがネイティブコード（標準化されたインタープリタ／仮想マシン内で実行されるコードではなく、マシン内のローカルプロセッサ上で実行さ

れるコード）を実行する能力をサポートし、および／または、文書化されていないプレイヤタイプ固有の計算をサポートする場合、コンテンツはこれらの機能を使用して、プレイヤタイプを認証することができる。特定のモデルであると主張するがそのモデルによってサポートされるオペレーションを正しく実行することができないプレイヤ上でコンテンツが実行されていることにそのコンテンツが気付く場合、コンテンツは、例えば、悪意をもって危殆化されたこのタイプのプレイヤから抽出されたキーを使用して、このプレイヤタイプとして装っているデバイス上で実行されていると、合理的に結論を出すことができる。

【０３２１】

セキュリティチェックには、コンポーネントまたは属性の任意の組合せを解析することを含めることができる。例えば、メディア、メディアドライブ、およびハードウェアコンポーネントの特性は、偽造プレイヤを検出するために有用である。事前に記録されたメディア上、または「スタンプ付きの」メディア上の販売されたコンテンツは、ドライブに問い合わせて、そのコンテンツがコンシューマ記録可能メディア上で実行されているかどうかを判断し、コンシューマ記録可能メディア上で実行されている場合、このことは決して起こるべきではないので、再生されることを拒否することができる。ある状況（例えば、支払いが行われる場合）において特定のタイトルがコンシューマ記録可能メディアへの記録を可能にする場合、例えば、特定のシリアルナンバを有するメディア、および／または特定の記録デバイスの識別子を保持するメディアからの再生を明示的に許可する（コンテンツ所有者が許可した担当者によって発行された）有効なデジタル署名が存在する場合にのみ、コンテンツはコンシューマ記録可能メディアから再生することができる。

【０３２２】

出力デバイス上、デバイスドライバ上、および、コンテンツを受信する（または、受信する可能性のある）コンポーネントなどの他の再生コンポーネント上で、セキュリティチェックを実行することができる。この機能は、悪意のある出力デバイス、または危殆化された出力デバイスを使用してコンテンツが記録される場合のある状況に対処するために、特に有用である。出力デバイスのための検証オペレーションは、デバイスの機能によってのみ制限される。例えば、出力デバイス検証には、ネイティブコードを入力または出力デバイスに送信すること（例えば、セキュリティの問題を検出するため、バグを修正するためなど）、他のデバイスとの暗号ハンドシェーク（cryptographic handshake）を実行すること、メディア上に保持された失効リストに対してデバイスのキーをテストすることなどが含まれる場合があるが、これらに限定されるものではない。出力デバイスはまたプレイヤの検証を要求することもでき、この場合、適切な検証は、実際のプレイヤ、コンテンツコード、またはこれら（または他のコンポーネント）の組合せによって提供される場合がある。複数の出力デバイスがチェーン状に（in a chain）接続される場合（例えば、プレイヤデバイスはデジタルオーディオをミキサに送信することができ、ミキサはデジタルオーディオを増幅器に提供し、増幅器はアナログ出力をスピーカに提供する）、各デバイスはそれがコンテンツと共に何を行うことを予定しているかについての情報をプレイヤに提供し、発信元（または他の）デバイスがメッセージを後続のデバイスへ渡すことを可能にすることができる。このように、コンテンツコードは、再生プロセスに関係するデバイスの任意に長いチェーン（またはネットワーク）内の各デバイスを認証することができる。信頼できないデバイス（または、記録デバイスなどの許可されないデバイス）が検出される場合、コンテンツコードは、適切な応答が返されることを確実にすることができる（適切な応答には、再生を拒否すること、信頼できないデバイスへの出力を拒否すること、信頼できないデバイスにコンテンツを送信しないよう中間デバイスに命令すること、信頼できないデバイスを再構成して問題を是正すること、出力品質を低下させること、追加のセキュリティ認証ステップを実行することなどが含まれるが、これらに限定されるものではない）。インターネット接続または別のデータ接続が存在する場合、セキュリティチェックには、追加のセキュリティ関連情報をリモートサイトからダウンロードすることを含めることができる。同様に、リモートデバイスおよびローカルストレージ（例えば、ＮＶＲ

ＡＭ）を使用して、新たな証明書失効テーブルなどの有用な情報、または、メディア上のコードよりも新しいデジタル署名付きセキュリティコードを取得することもできる。

【０３２３】

セキュリティチェックは通常、再生に先立って実行されるが、コンテンツコードはセキュリティチェックをいつでも実行することができる。再構成される可能性のある再生環境では、または、他の新しいセキュリティ関連情報が入手可能になる可能性のある状況では、追加のセキュリティチェックを定期的または絶えず実行することが有効である場合がある。大きな変更が検出される場合、コンテンツコードは再生を終了するか、または、再生環境における確信を再確立することができるようになるまで、再生を休止することができる。

【０３２４】

テーブルまたは判断ツリー（decision tree）を使用して、特定のプレイヤ属性が与えられるとどのセキュリティ解析コード（もしあれば）が適切であるかを、迅速に選択することができる。現在のプレイヤに適用できないセキュリティコードが実行される必要はなく（または、セキュリティコードがメディアからロードされる必要もなく）、不必要なパフォーマンスへの影響を回避し、それをサポートしないプレイヤ上でセキュリティコードを実行する結果として生じる可能性のある互換性の問題を回避する。前述のように、セキュリティチェックを暗号化キー導出オペレーションに結合することができる（例えば、その場合、特定のセキュリティチェックを必要とするデバイスは実際にチェックを実行して、必要なキーを正しく導出しなければならないが、そのセキュリティチェックを必要としないデバイスは、キーを直接得ることができる）。

【０３２５】

（例示的キー管理）

例示的プレイヤはそのメーカによって、キーの一意の組合せをもって初期化される。これらのキーは、メディアフォーマットを制御するエンティティによって許可されるキー管理局から得られる。キー管理局は、許可されたメーカから要求を受け、その要求の妥当性を検査し、プレイヤキーセットを提供し、それらのキーのための支払い（および、対応するライセンス料）を受ける。例示的プレイヤのためのキーセットは１２８のＡＥＳキーからなり、これらのキーは各々が１２８ビットであり、（キー管理局にのみが知る）トップレベルの２５６ビットＡＥＳキー、（プレイヤのメーカを識別する）３２ビットグループＩＤ、および（プレイヤのシリアルナンバを識別する）３２ビットデバイスＩＤから導出される。グループおよびデバイス識別子は共にプレイヤを一意に識別し、プレイヤのデジタル証明書内に表現される。

【０３２６】

例示的実施形態では、キーセット内の１２８のキー（キーＩＤ値０ないし１２７）の各々は、キー管理局によって、最初にキー選択ブロックをキーＩＤ、グループＩＤおよび／またはデバイスＩＤから計算することにより導出される。キー選択ブロックは次いで、（必要な場合）パディングされ、トップレベルキーを使用して暗号化される。結果として生じる値を後処理し（例えば、切り捨て）、実際のキーを作成することができる。キー選択ブロックを指定されたパラメータから導出するために使用される方法は、キーＩＤによって決まる。以下の表は、キーＩＤ、グループＩＤおよびデバイスＩＤをキー選択ブロックに変換するために使用可能な例示的ルールを示す。

【０３２７】

【表１】

キーID	キーIDの後に続くキー選択ブロックのコンテンツ
0...31	GroupID || (DeviceID >> KeyID)
32..63	(GroupID >> (KeyID - 32)) || 0x00000000
64..79	KeySelector(GroupID, DeviceID, 15, 8)
80..83	KeySelector(GroupID, DeviceID, 16, 8)
84	KeySelector(GroupID, DeviceID, 17, 9)
85	KeySelector(GroupID, DeviceID, 18, 9)
86	KeySelector(GroupID, DeviceID, 19, 10)
87	KeySelector(GroupID, DeviceID, 20, 10)
88	KeySelector(GroupID, DeviceID, 21, 11)
89	KeySelector(GroupID, DeviceID, 22, 11)
90	KeySelector(GroupID, DeviceID, 23, 12)
91	KeySelector(GroupID, DeviceID, 24, 12)
92	KeySelector(GroupID, DeviceID, 25, 13)
93	KeySelector(GroupID, DeviceID, 26, 13)
94	KeySelector(GroupID, DeviceID, 27, 14)
95	KeySelector(GroupID, DeviceID, 28, 14)
96	KeySelector(GroupID, DeviceID, 29, 15)
97	KeySelector(GroupID, DeviceID, 30, 15)
98	KeySelector(GroupID, DeviceID, 31, 16)
99	KeySelector(GroupID, DeviceID, 32, 16)
100	0x00000000 || DeviceID
101..127	GroupID || DeviceID

【０３２８】

キーを（それらのグループＩＤおよびデバイスＩＤに応じて）擬似ランダムデバイスサブセットに割り当てる、例示的関数ＫｅｙＳｅｌｅｃｔｏｒ（ＧｒｏｕｐＩＤ，ＤｅｖｉｃｅＩＤ，ｇ，ｄ）は、以下の通りである。

```
Let hash[0..19] equal the bytes of the SHA hash of KeyID || GroupID.
Let AND_MASK = (1 << g) - 1.
For i = 0 upto d-1:
Let n = hash[i] mod (g - i).
Clear the nth set bit in AND_MASK.
EndFor.
Let XOR_MASK = hash[16..19] AND (AND_MASK XOR ((1 << g) -1)).
Compute deviceHash[0..19] as the SHA hash of KeyID || GroupID || DeviceID.
Return 0x00000000 || ((deviceHash[16..19] AND AND_MASK) XOR XOR_MASK).
```

【０３２９】

「||」は連結（concatenation）を示し、「０ｘ」は、以下に続くものが３２ビット１６進数値であることを示し、「＜＜」は左シフト演算を示し、「－」は減算を示し、「ＡＮＤ」はビット単位の論理ＡＮＤを示し、「ＸＯＲ」はビット単位の論理排他的ＯＲを示すことに留意されたい。

【０３３０】

異なる種類のキーが含まれることに留意されたい。例えば、キー０およびキー１０１ないし１２７は、一意にデバイスに割り当てられるようになる。反対に、キー１ないし３１は、同じグループ内のデバイスのデバイスの次第に大きいセットの間で共有されるようになる。キー３２ないし６３は、複数のグループの間で共有される。最後に、キー６４ないし９９は、デバイスの様々な（この場合、擬似ランダム）サブセットの間で共有される。キー１００は、同じデバイスＩＤを有する異なるグループにわたって複数のデバイスにより共有される。

【０３３１】
デバイスキーが十分に保護されることは重要である。例えば、デバイスの実際のキーをメディア上に配置することは、タイトルのセキュリティが危殆化された場合、プレイヤのキーを危殆化させるおそれがある。同様に、デバイスキーをコンテンツパブリッシャに提供することは、コンテンツパブリッシャが危殆化されるようになる状況においてキーを危険にさらす可能性がある。これらのリスクに対処するため、本発明は、コンテンツを準備し、かつ暗号化するエンティティが信頼できることを必要としない。例えば、キー管理局はサービスを提供することができ、それにより、任意のパーティがプレイヤキーのセットおよび１つまたは複数の入力値を識別できる。それに応じて、このキー管理局は、識別されたキーの各々によって指定された入力を復号化（または暗号化）した結果を提供することができる。代替として、コンテンツパブリッシャは、その選択の平文／暗号文のペアを提供することができる。いずれの場合も、コンテンツを作成するパーティは平文／暗号文のペアを得ることができるが、キー自体を得ることはできない。次いで、安全なメディアを作成する際に、これらのペアを使用することができる。
【０３３２】
例示的メディアの作成には以下のステップが含まれる。（ａ）コンテンツパブリッシャが、いずれの失効したプレイヤにも知られていないプレイヤキーのセットを定義する、ただし各有効なプレイヤはそのセット内の少なくとも１つのキーを含む、（ｂ）コンテンツパブリッシャが任意の平文値を選択する、（ｃ）コンテンツパブリッシャが、所望の平文値を選択されたセット内のキーの識別と共にキー管理局に安全に送信する、（ｄ）キー管理局が、この平文値が以前に別のパーティによって提供されていないことを検証する、（ｅ）キー管理局が、要求されたセット内の各プレイヤキーを導出し、各キーを使用して、平文値を暗号化する、（ｆ）キー管理局が、結果として生じる暗号文をコンテンツパブリッシャに提供する、（ｇ）コンテンツパブリッシャがコンテンツの一部分（または、あるコード、キーもしくは他のデータ）を平文値により暗号化する、および、（ｈ）コンテンツパブリッシャが、暗号文リスト、および、コードを実行するプレイヤがセット内のいずれかのキーを含むかどうかを識別するように構成される対応するコードと共に、暗号化されたコンテンツをメディア上に格納し、コードを実行するプレイヤがセット内のいずれかのキーを含む場合は、プレイヤ内のキーＩＤを決定し、キーＩＤに対応する暗号文を暗号文リスト内で探し出し、例えば、ソースデータとしての暗号化されたコンテンツ部分と、キーパラメータとしての暗号文と、および、選択されたキーＩＤとをもってＴＲＡＰ＿Ａｅｓを呼び出すことによって暗号化されたコンテンツ部分を復号化する。コンテンツコードは次いで、復号化された結果をコーデックに提供し、またはそうでない場合はこのデータを必要に応じて使用することができる。
【０３３３】
代替実施形態は、以下に限定されるものではないが、複数のレベルの復号化オペレーションを使用でき、暗号化結果をフォレンシックマーキングおよび他のオペレーションと統合でき、複数のキー局を使用でき、セキュリティコードまたは他のプロセスを使用して、値（キー、平文、暗号文など）をさらに保護または処理できる。
【０３３４】
公開キー暗号技術が使用される場合、キー管理局はプレイヤの公開キーを提供（または発行）することができる（または、識別ベースの暗号技術を使用して、キーリストを不必要にすることができる）。キーにデジタル署名するため、キー管理局は、参照署名（例えば、メッセージ／署名のペア）をコンテンツコードによる使用のために提供することもできる。（例えば、ＫｅｙＳｅｌｅｃｔｏｒなどのオペレーションを、キーペアを生成するために使用される強力なＰＲＮＧのためのシードとして使用し、キーをランダムに生成して割り当てることなどによって、）公開／秘密キーをデバイスの複数のグループの間で共有することができる。
【０３３５】
言うまでもなく、上記の役割を複数のパーティの間で分割することができる。例えば、キ

ー管理局は、集中失効リスト（centralized revocation list）を管理し、どのキーが有効でありどのキーが有効でないかをコンテンツパブリッシャが判断することを不要にすることもできる。同様に、コンテンツパブリッシャはセキュリティ関連タスクを（キー管理局を含む）他のパーティにアウトソースすることができる。

【０３３６】

プレイヤキーを導出するための上記の例示的プロセスは、決定論的である（deterministic）。他の実施形態は、キーをランダムに生成することを含む場合があり、他の技術を使用する場合がある。キーを導出するために使用されるオペレーション（例えば、ＡＥＳ暗号化）は例示的であり、他のオペレーション（ＭＡＣ、公開キーオペレーション、他のアルゴリズムなど）で代用される場合がある。

【０３３７】

（バージョニング）

コンテンツがデコードされるとき、通常は（ａ）コンテンツがプレイヤよりも古くなるか、または（ｂ）プレイヤがコンテンツよりも古くなるかのいずれかである。両方の場合、プレイヤがすべての必要とされるＴＲＡＰ（必要な暗号化キーのいずれをも含む）を正しく実装し、コンテンツに関連付けられたコードによって実装されるセキュリティルールを満たすならば、再生は正常に発生すべきである。言うまでもなく、コンテンツコードが不十分に書かれる場合、プレイヤに欠陥がある場合、または、重大なセキュリティの問題が作成後に生じている場合、再生は自動的に正常に機能しない場合がある。

【０３３８】

コンテンツコードが正当なプレイヤ上で正しく再生されていない状況に対処するため、（例えば）プレイヤをディスク固有のセキュリティオーバーライドキーについてチェックし、キーが発見される場合、そのキーを使用して、１つまたは複数のセキュリティチェックまたは正常に実行されるであろう他のオペレーションを回避するコンテンツコード部分を復号化するように、コンテンツが作成される場合がある。この「回避」ロジックをメディア上で、暗号化された形式で保持し、復号化キーが実際にリリースされない限りこれを使用することができないことを確実にすることができる。オーバーライドキーが必要とされる場合、そのオーバーライドキーは、例えば、プレイヤの不揮発性メモリ内に（例えば、暗号化された形式で）格納され、ネットワークを介して検索され、ＴＲＡＰを介してアクセスされる場合などがある。セキュリティに影響を及ぼすこと（ディスプレイ制限によりプレイヤのためのグラフィカルメニューを使用できなくすることなど）のない手段は、ユーザによって構成可能であり、安全な認証を必要としない。

【０３３９】

セキュリティ障害またはリスクが検出される状況にユーザが対処できるように、コンテンツコードを構成することもできる。例えば、モデム、インターネット接続、または他のデータチャネルが使用可能である場合、コンテンツは認証サーバと通信して、アップデートされたセキュリティコードを得ることができ、および／または、再生を進めるための許可を得ることができる。アクティベーション値を提供する自動の通話料無料電話サービスにユーザが提供することができる、コードを表示する（または、可聴的に再生する）こともできる。（このサービスは、例えば、音声認識システムを使用することによって、ＤＴＭＦトーンを検出することによって、または、ユーザのスピーカから受話器に伝送された音を検出することによって、電話を介して情報を受信することができる。同様に、情報を、キーボード入力のためにユーザに返すこと、受話器からプレイヤ上のマイクロフォンに伝送することなどができる。）コンテンツコードによって応答を検証し（例えば、デジタル署名、ＭＡＣ、難読化されたチェックサムなどをチェックすることによる）、および／または、後続の暗号化処理において（例えば、キーとして）応答を使用することができる。チャレンジ／レスポンス認証結果をフォレンシックマーキングと共に使用して、例えば、特定のユーザまたは識別子まで遡ってコピーをトレースすることを可能にすることができる。例えば、アクティベーション値が一意であり、フォレンシックマークに含まれる場合、コンテンツ所有者はフォレンシックマークを使用して、復旧された海賊コピーを、所与

のアクティベーション値を作成したトランザクションまで遡ってトレースすることができる。したがって、コンテンツ所有者が、アクティベーション値を提供することに先立って、ユーザについての情報（例えば、ＡＮＩを使用する発呼者の電話番号または発呼者のＩＤ、名前、クレジットカード番号など）を得ることは、有用である場合がある。アクティベーション値はプレイヤの不揮発性メモリ内に格納される場合がある（例えば、将来の使用のため、および、他のタイトルによる使用のため）。

【０３４０】

（メディアのエンコードおよびデコード）

例示的実施形態では、コンテンツコードは、メディアからデータを読み取り、そのデータをデコードするために必要とされるパラメータを（ＴＲＡＰを通じて）指定することができる。光メディアから読み取られたセクタの実際の処理には、以下のいずれかまたはすべてが含まれる場合がある。

【０３４１】

・セクタデコードパラメータが光ドライブ（または、セクタデコードの一部または全部を実行することを担う他のコンポーネント）に提供される。

【０３４２】

・ドライブが、要求されたデータに対応する生データ（例えば、セクタ）を読み取る。実際のロードには、セクタ識別子（または、他のアドレッシング情報）を使用してメディア上のデータを探し出し、次いで実際に適切なデータをメディアからロードすることが含まれる場合がある。ドライブはまた、ロードされたデータと共に含まれた、暗号化されていない部分（ナビゲーションフィールドなど）を除去または処理することもできる。

【０３４３】

・エラー訂正が生セクタデータに適用される。エラー訂正プロセスは、調節可能な多項式または他のパラメータを含む場合がある。コンテンツコードはまた、例えば、これらのオペレーション自体を実行することを望む場合、エラー訂正および／または検出を無効にすることもできる。

【０３４４】

・復号化または他の暗号化変換もまたエラー訂正と共に含まれる場合があり、ＥＣＣの前に実行される場合があり、および／または、後で実行される場合がある。エラー訂正されていないデータ上に適用される復号化プロセスは、エラーを伝える（propagate）ことを回避するように選択される場合がある。例えば、ストリーム暗号（例えば、カウンタモードにおけるＲＣ４またはＡＥＳ）による復号化が、ビットエラーの訂正に先立って実行される場合がある。復号化（または他の処理ステップ）が、ドライブ、コンテンツコード、コーデック、ドライバ、または他の任意のコンポーネントによって適用される場合がある。

【０３４５】

復号化プロセスはまた、一方向プロパティを有するように選択される場合もあるが、これは、例えば、敵対者が、記録された出力を使用して、他のデコードパラメータにより作成されていたであろう出力を計算することができるようにすることを防止するためである。例えば、ドライブはコンテンツ指定（content-specified）のキーを使用して、１６０ビット（または他のサイズ）の（ドライブによって返されない）リードイン値（lead-in value）のＨＭＡＣ－ＳＨＡとして、各セクタのキーを計算することができる。リードイン値を知らなければ、敵対者は、既知のコンテンツ指定のキーを使用して作成された復号化された出力を、他のコンテンツ指定のキー値を使用して作成されていたであろう出力に変換することはできない。結果として、敵対者は、（例えば、ハードドライブおよび悪意のあるデバイスドライバを使用して）メディアをコピーするために必要とされる、またはメディアをエミュレートするために必要とされる情報を欠く。敵対者が、特定のキー値のためのデコードされたセクタ表現を格納することは可能となる一方で、コンテンツコードが、対応する出力が知られていないキーを使用するセクタを要求するときは常に、コピー／エミュレーションは失敗する。例えば、２５６ビットキーについて２＾２５６など、デコ

ードキーの総数を極度に大きくすることができるので、敵対者がすべての可能なセクタデコードを格納することは不可能である。コンテンツコードは、様々な組合せの要求を実行し、それらの一部について、ロードされた実際のデータを使用し、かつチェックすることができる。

【０３４６】

プレイヤ（または他のコンポーネント）は、非対称暗号化変換をセクタデータ上で実行することができる。例えば、ブロック署名を使用して、敵対者が許可なく新しいタイトルをマスタリングすることを防止することができる。一実施形態では、公開キーは、ドライブに埋め込まれた（が、メディアから得られる、証明書から抽出されるなどの可能性もある）２０４８ビットのＲＳＡモジュラスからなる。署名検証オペレーションの効率を高めるために、ＲＳＡモジュラスは、正確な２の累乗に近づくよう、例えば、「１」に等しい最上位の１２８ビットを有して、生成される。メディアをマスタリングする間、各ブロック（例えば、セクタ）は最初に、セクタデータと、メディアＩＤのＳＨＡハッシュ、ブロックのアドレスおよびブロックデータを使用してストリーム暗号をシードすることによって作成されたキーストリームとを排他的ＯＲすることによって、ランダム化される。ブロックデータは次いで、２５６バイト（２０４８ビット）チャンクに分割され、これらのチャンクは各々、ＲＳＡ秘密キーを使用してチャンク上でモジュラ指数計算（modular exponentiation）を実行することによって、デジタル署名される。署名付きチャンクは、ＳＨＡハッシュと共に記録される。よって、このブロックは２０バイト分拡大される（ブロックが２０４８バイトセクタである場合、１％未満）。ストリーム暗号が効果的にランダム化するならば、この確率は無視することができるほど小さく（約２＾（－１２８））、モジュラ指数計算入力は公開モジュラスより大きくなる。結果として、この場合には特別な処理は必要ではない。対応する検証およびデコードプロセスは、データブロック（例えば、セクタ）がメディアから読み取られ、デコードされるときに実行される。このプロセスには、（ａ）２５６バイトチャンクおよびＳＨＡハッシュを読み取ること、（ｂ）公開モジュラスおよび公開指数（例えば、３）を使用して、モジュラ指数計算オペレーションを各チャンク上で実行すること、（ｃ）例えば、チャンクと排他的ＯＲされるキーストリームを計算するために含まれた、ロードされたＳＨＡハッシュ値を使用することによって、ランダム化プロセスをリバースすること、（ｄ）メディアＩＤ、ブロックアドレス、およびブロックデータをハッシュすることによって、ならびに、その結果と、ロードされたハッシュ値とを比較することによって、デコードされたデータの完全性（integrity）を検証すること、および、（ｅ）これらのハッシュが合致する場合、デコードされたデータブロックを返し、これらのハッシュが合致しない場合、エラーが返されることが含まれる。パフォーマンスが問題である場合、検証を確率的に（例えば、ランダムに選択されたセクタ上で）、および／または、より重要なデータを含む領域上でのみ、実行することができる。

【０３４７】

記録可能メディアに書き込む（または、複製のためにマスタを作成する）とき、記録デバイスは、暗号化一方向関数（ＳＨＡ、ＨＭＡＣ－ＭＤ５、ＡＥＳを使用して構築された一方向関数など）を使用して、それが変換する値を受信することができ、メディア上の特別な位置に書き込む（下記参照）。このように、敵対者は正しい入力（例えば、ハッシュプレイメージ）を知るようにならないので、正当な記録機器を使用してメディアをコピーすることはできない。他の認証値（デジタル署名など）もまた格納される場合がある。セキュリティ値は、メディアの任意の部分（またはすべて）に適用可能である場合がある。

【０３４８】

メディアセキュリティ値は通常のデータとして格納される場合があり、または、直接の読み取りからの追加の保護を含む、および／または、特別にエンコードされる、「特別な」領域に格納される場合がある。セキュリティデータの量を比較的小さくすることができるので（例えば、１２８ビット以下）、比較的低い密度およびパフォーマンスを有するストレージ技術が使用される場合がある。例えば、光メディアのピットエンコードまたはトラ

ック位置（「ウォブル」）において、データは様々な形でエンコードされる場合があることが知られている。データはまた、通常はエラー訂正のために予約された冗長ビット内、セクタ期間におけるバリエーション内、ナビゲーションフィールド内に隠される場合もある。データはまた、異なる物理的ディスク位置の上で（例えば、標準可読領域（normally-readable）の内側もしくは外側で、または、異なる焦点深度で）エンコードされる場合もある。

【０３４９】

メディア上のデータには、インジケータビット、ピットエンコードバリエーション、または、所定の処理またはセキュリティステップがデータに関連して実行されるべきであるかどうかを示す他のマーカーが含まれる場合がある。例えば、セクタセキュリティフラグを使用して、そのセクタに含まれるデータが公開キーと共に変換されるべきであり、および／または、最初に受信された暗号許可をドライブが有する場合にのみ、そのセクタに含まれるデータがリリースされるべきであることを、示すことができる。（このような許可には、セクタコンテンツのＳＨＡハッシュを必要とすること、メディアがマスタリングされたときに、キーが失効されなかった受信者により、成功したチャレンジ－レスポンス認証を実行すること、データのリリースを許可するデジタル署名を検証すること、セクタを正しくデコードするキーを受信することなどが含まれる場合があるが、これらに限定されるものではない。）データにマークを付けて、データが暗号化された形式でのみリリースされるようにすることもできる（これは、パーソナルコンピュータ、および、信頼できないバスまたはチャネルを介してデータが移動する場合のある他のシステムにおいて、特に重要である）。

【０３５０】

上記の例示的実施形態は主として、光メディアに関して説明されたが、類似の技術を、磁気メディアなどの他のメディアタイプに適用することができる。ソリッドステートメディア（ＥＥＰＲＯＭ／フラッシュメモリなど）、および、メディア自体が計算を実行することができる他のストレージ技術では、メディアは暗号化処理自体を実行することができ、また、ドライブ、プレイヤアプリケーションなど（これらもまたインタープリタを有する場合がある）と通信するセキュリティインタープリタを内部に含むこともできる。メディアは、圧縮されたコンテンツ（例えば、高精細度の映画では約３０ギガバイト）を実際に格納するために必要な不揮発性メモリの量のみを含むことのみが必要であるが、メディアは、暗号技術または他の技術を使用して、非常に大きい（例えば、２＾２５６バイト）「仮想」または「エミュレートされた」アドレス範囲を作成することができる。このアドレス範囲が作成される場合、敵対者が完全な海賊コピーを作成することはできなくなる。なぜならば、アドレス範囲全体を格納すること（または、読み取ることさえ）は実行不可能であり、アドレス範囲の諸部分がどのように実際に使用されるようになるかを敵対者が決定するための汎用の方法がないからである。例えば、各再生において、コンテンツコードは、異なる領域が使用されることを要求することができる。この汎用的な手法を、メインプレイヤアプリケーションがプログラマブルではないが、メディア（または、メディアドライブ／インターフェース）がプログラマブルである状況においても、使用することができる。

【０３５１】

（条件付きアクセスキーストリームフォレンシック）

有料テレビ受信器は一般に、すべてのセキュリティクリティカルなコンポーネント（security-critical component）をスマートカードまたは他のリムーバブルデバイスに入れるという目的をもって設計されるので、スマートカードを取り替えることによってセキュリティ障害に対処することができる。通常、取り替え可能なモジュールは、復号化キーを導出するために使用され、これらの復号化キーは、セットトップボックス内に含まれる汎用復号化回路に提供される。従来の実施態様の主な欠陥は、敵対者が出力を、許可されたデバイスから記録し、キー（および他の任意の必要とされたデータも）を無許可のデバイスに再送信することができたことである。関連した攻撃には、ビデオ自体を記録および再送

信することが含まれる。

【０３５２】

このような攻撃をトレースすることは、非常に困難であるか、または不可能である可能性がある。例えば、海賊行為者が危殆化したコンピュータを使用して、キーおよびデータを無許可の視聴者に、インターネットを介して匿名でリレーすることができた。ＵＤＰパケットにおいて、偽造された「ｆｒｏｍ」アドレスでキーを送信し、トレーシングを大変困難にすることができる。システムオペレータは、海賊サービスに登録することによって、無許可のデータへのアクセスを受信することができるが、攻撃のソースをトレースする方法はない。他の状況では、データのトレーシングは実用的である場合があるが、ソースは、法執行機関が権限を有さない区域に位置する場合がある。

【０３５３】

現在のシステムでは、ビデオを復号化するために使用される一連の復号化キー（ならびに、復号化されたビデオ自体）は、各サブスクライバについて同一である。これらのキーは（例えば、条件付きアクセススマートカードからビデオデコーダチップへ）移送され、デバイス固有のキーにより暗号化されるが、実際のビデオ復号化キーはなおグローバルである。結果として、あるセットトップボックスを危殆化またはリバースエンジニアリングし（それらの設計はしばしば完全に標準化されるので、これは必ずしも大変困難ではない）、正当にサブスクライブされたアカウントを有する敵対者は、一連のビデオ復号化キーを決定および再送信することができる。この攻撃は通常、実際にセキュリティモジュール自体を危殆化させることを必要とはせず、セキュリティモジュールの置換が効果的な対策とならないようにする。

【０３５４】

このような攻撃に対処するため、システムオペレータは、敵対者によって、再送信されているキーおよび他のデータを抽出するために使用されている特定のアカウント（、および、したがってセキュリティモジュール）を識別することが必要である。このアカウントが識別された後、システムオペレータは、このアカウント、および、任意の関連アカウント（例えば、同じ電話回線に接続されたデバイスからのアカウント、同じ請求先住所を共有するアカウント、物理的位置が近いアカウント、同じクレジットカードまたは他の支払い手段により支払われたアカウントなど）を取り消すことができる（例えば、敵対者のセキュリティモジュールによって利用することができる形態でアクセスのために必要とされたキーを送信することをやめることによる、敵対者のセキュリティモジュールを一時的または永続的に使用できなくする暗号化されたメッセージを無線で（over the air）送信することによる、など）。

【０３５５】

この識別を実施するため、システムのオペレータは各セキュリティモジュールによって出力される実際の一連の復号化パラメータ内にフォレンシックマーキング情報を組み込む。

【０３５６】

一実施形態では、条件付きアクセスモジュールは、従来の復号化キーを、ビデオデコードに先立ってセットトップボックスによって復号化された平文に適用されるべきバイト代替（byte substitution）と共に出力する。ビデオシーケンスを作成するとき、システムオペレータは、バイト置換（byte replacement）が、受け入れ可能なバリエーション（ポリモーフ（polymorph））を作成することができる、ビデオストリーム内の複数の位置を識別する。加えて、有効な（すなわち、快適に視聴できる）ビデオストリームを形成するためにバイト置換が必須であるように、ビデオストリーム内の一部のバイトは破損される。正当にサブスクライブされた条件付きアクセスカードは、視聴可能なストリームを再作成するために必須のバイト代替の十分なセットを含む、一意の組合せのバイト代替を出力するように構成される。各モジュールは一意の組合せの代替を出力し、単一のデバイスに返された、再送信された出力と照合することを可能にすることができる。代替実施形態では必ずしもバイト代替を使用する必要はなく、任意の方法の変換を条件付きアクセスモジュールによって指定することができる。これらの変換には、加算、排他的ＯＲ、ブロック移

動、削除、挿入、ビット反転、および、より複雑なオペレーション（インタープリタまたはマイクロプロセッサ、デコードデバイスによって実行されるべき実際のコードを指定することを含む）が含まれるが、これらに限定されるものではない。

【０３５７】

上述の方法はセットトップボックスにおけるサポートを必要とし、したがって、多数のレガシーシステムと共に使用することはできない。レガシー環境では、２つの異なるキーイング状態（keying state）の各々において解釈されるときに、有効であるストリームをコンテンツプロバイダが構築する（assemble）、異なる実施形態が必要とされる場合がある。

【０３５８】

コンテンツプロバイダは、２つの異なるキー（またはキーイング状態）の各々により復号化されるとき、受け入れられるようにデコードするビデオストリーム部分を構築する。既存の衛星またはケーブルテレビシステムは通常、ビデオの部分（１０秒セグメントなど）の各々を異なるキーにより暗号化する。本発明では、１つまたは複数のセグメントが作成され、これらのセグメントを複数のキーによりデコードすることができる。

【０３５９】

一実施形態では、ビデオ圧縮プロセスは、圧縮されたビデオデータの初期ブロックに対する複数の受け入れ可能な修正を識別するように修正される。次に、このブロックは第１のキーにより暗号化され、第１のキーにより復号化するセットトップボックスへの送信のために適切となる暗号文が作成される。次に、このブロックは第２のキーにより暗号化される。結果として生じる平文もまた、第２のキーによりキーが付けられたデバイス上で受け入れ可能な出力にデコードするようになる場合、このブロックは保存され、このプロセスは後続の各データブロックごとに繰り返される。第２のキーによる復号化が、暗号文は、第２のキーによりキーが付けられたデバイス上での受け入れ可能なストリームの形成の影響を受けにくいことを示す場合、開始ブロックのための新しい値が試される。十分な試行により、システムオペレータは、２つの（または場合によってはより多くの）キーの各々により復号化することができるビデオストリームを徐々に構築することができるようになる。各キーを使用することで得られるビデオの品質は、費やされた計算の試行、ブロック候補を選択するために使用された技術、使用されたビデオ圧縮技術、参照ビデオ、および他の要素によって決まる。

【０３６０】

より計算効率の高いエンコード方法もまた可能である。例えば、大部分のビデオ圧縮スキーム（ＭＰＥＧ－２を含む）においては、フィールド（コメントフィールドなど）を使用して、各キーを使用して暗号化されるデータ領域を交互配置する（interleave）ことができる。この技術は通常、ビデオデータのサイズを増大させるが、計算効率を高くすることができ、出力品質を低下させない。

【０３６１】

例示的ビデオストリームは、従来のようにエンコードされたデータ（例えば、単一のキーにより暗号化されるか、または暗号化されないデジタルビデオおよび／またはオーディオ）が、キー変化が期待される位置まで移動することで開始する。キー変化信号が次いでエンコードされる。（ａ）第１の復号化キーを使用して、圧縮されたビデオストリームの次の８バイトのための適切な値へ復号化するように、および、（ｂ）第２の復号化キーを使用して、（例えば、ＭＰＥＧコメントを開始することによって）後続のバイトを無視するよう圧縮解除デバイスに命令するか、またはそうでない場合は、後続のバイトに、圧縮解除された出力に対して比較的わずかな影響しか及ぼさないようにさせる、ビットシーケンスに復号化するように、キー変化後の第１の暗号文ブロックは選択される。これらのプロパティを有するブロックを発見するため、圧縮デバイスまたはプロセスは（例えば）、最初のキーを使用して復号化する際の所望の平文をもって開始し、次いで、第２のキーによっても適切な結果を出すキーのペアが発見されるまで、第１および／または第２のキーのためのランダムな値を試す。

【０３６２】
（ａ）暗号文が、第１のキーにより復号化された、圧縮されたビデオを生成するように、および、（ｂ）第２のキーにより復号化されるときに暗号文が無視される（または、比較的少ない試行で処理される）ように、ストリームの後続の部分は生成される。ビデオデータを第１のキーにより暗号化することによって、および、（必要な場合）結果として生じるデータを第２のキーにより復号化して、この結果が適切である（例えば、時期を早めてＭＰＥＧコメントを終了させない）ことを検証することによって、このストリームを準備することができる。平文にわずかな調整を施し、かつ、不適切な平文が生じる任意の状況（例えば、時期を早めてデータを無視することを終了するようになるか、または、不適切な長さを伝送するようになるか、またはそうでない場合は、不正なストリームを作成するようになるか、または、ストリームの不快な割り込みを引き起こすようになるなど）を改善することを繰り返し行うことが、必要である場合がある。最終的に（ＭＰＥＧコメントの終了が近づきつつあるとき、第１のキーにより暗号化されているビデオブロックがほとんど完了するときなど）、暗号文ストリームは、第１のキーにより復号化されるときに、圧縮解除プロセスにデータの無視を開始させる、データを含むことができる。この時点で、またはその後間もなく、第２のキーによる「データ無視」状態は終了し、第２のキーにより復号化されるときに有効な圧縮されたビデオを生成するが、第１のキーにより復号化されるときには無視される暗号文をもって、ストリームは継続する。キーの一方により復号化するときに圧縮されたビデオを生成し、他方により復号化するときにコメントデータを生じる、データのセグメントを互い違いにしながら、ストリームは継続する。最終的に、（例えば、ストリーム内に存在するキー変化通知のために）キー変化が引き起こされ、通常のビデオに戻るか、または、新しいキーのペアにより復号化することができるビデオの新しいセグメントを開始する。

【０３６３】
敵対者が、キーシーケンスの再送信のみを行っている場合、これは、２つの復号化結果が、圧縮解除されるときに同じビデオを生成する場合、受け入れ可能である場合がある。しかし、敵対者が、圧縮解除されたビデオを再送信中である場合、キー値の各々により圧縮解除することによって作成されたビデオシーケンスは異なるべきである。

【０３６４】
代替実施形態は、３つ以上のキーにより正しく復号化することができるか、または、複数の暗号化アルゴリズムを使用して（同じかまたは異なるキー値により）デコードすることができる、データを含むことができることに留意されたい。サポートされる復号化アルゴリズムの例には、ＤＥＳ、ＡＥＳ、トリプルＤＥＳ、ＤＶＢブロック暗号、ＩＤＥＡ、いずれかのモード（ＣＢＣ、ＥＣＢなど）におけるいずれかの他のブロック暗号、ＲＣ４、ＳＥＡＬ、いずれかの他のストリーム暗号などが含まれる場合があるが、これらに限定されるものではない。暗号化をセットトップボックス上で無効にすることができる実施形態では、暗号化無効状態をキーイング状態として使用することができる（例えば、送信されたデータを、送信された形式において有効なビデオとしてうまく解釈することができる状態、および、適切なキーにより復号化されるときも）。システムオペレータは、圧縮されたビデオシーケンスまたはシーケンス内で使用可能な部分（有用な暗号文ブロックおよび対応するキーなど）を事前計算することができる。ストリームをリアルタイムで、または前もって構築することができる。例えば、共通のシーケンスのためのビデオ（全黒表示など）を、事前計算し、挿入することができる。キーチェックがデコードデバイスによって実行される場合、システムオペレータは、同じキーチェックを同時に通過することができる複数のキーを探し出すための検索を実行する必要のある場合がある。バースディパラドックス（birthday paradox）を利用する衝突検索技術（collision searching technique）を使用して、このような検索を簡素化することができる。

【０３６５】
ビデオストリームのために必要とされる帯域幅は通常、上述のようにエンコードされたビデオセグメントでは増大するが、トレースされる必要のある海賊攻撃が進行中であると思

われる状況に使用を制限することによって、帯域幅全体にわたる影響は最小限にすることができる。敵対者が、本発明がアクティブである状況においてビデオセグメントを識別し、削除しようと試みる場合（例えば、ストリーム内で無視されたデータ量を解析することによる）、類似の特性を通常の（マークが付いていない）領域に入れて、攻撃者を混乱させるための「レッドヘリング（red herring）」を作成することができる。

【０３６６】

未使用ビット（シングルＤＥＳまたはトリプルＤＥＳキーに通常は含まれるパリティビットなど）と共に暗号を使用するシステムでは、フォレンシック情報をこれらのビットに含めることができるが、敵対者がキーを再送信することに先立ってこれらのビット内の値を上書きすることができる場合、この手法の効果は制限される場合がある。

【０３６７】

デバイス上で適切なキー値を導出するために使用されるロジックは、セキュリティデバイスに内部で実装されるべきである。キー導出は、無線で（例えば、暗号化された形式で、および、ビデオまたは他のコンテンツと混ぜられて）受信されるソフトウェアおよび／またはルールを使用して、実行される場合がある。例えば、複数の方法で復号化することができるブロックを送信することに先立って、視聴ベース（viewing base）の所定のサブセットが各キーを有するように、システムオペレータはキー（またはキー暗号化キー）を（個々に、および／または、グループで）送信することができる。これらのキーを使用して、各デバイス上で適切なビデオキーを復号化、またはそうでない場合は導出することができる。サブセットはランダムに選択される場合があり、および／または、地理的位置（例えば、ローカルテレビチャネルを送信するためにも使用されるスポットビーム衛星信号を受信する能力、ケーブルネットワーク内の位置などに基づく）、サブスクライバＩＤ、電話番号、１つまたは複数のキーの知識、デバイスタイプ、または、他の任意の特性（または特性の組合せ）に基づくようにされる場合がある。実行可能コードはまた、キー導出／選択を支援する（またはこれらを実行する）ために、（任意で、リアルタイムで）導出される場合もある。

【０３６８】

悪意をもって再配布されたキーおよび／またはビデオのソースをトレースするための例示的実施形態に含まれる特定のステップおよびコンポーネントには、以下のいずれかまたはすべてが含まれ、これらのステップおよびコンポーネントはいかなる順序でもよい。（ａ）２つの異なるキーイング状態の各々により復号化されるときに、少なくとも２つの異なる平文を生成し、ただし、有効な圧縮されたコンテンツストリームの一部として各平文を正しくデコードすることができる、圧縮された、暗号化されたビデオの部分を作成すること、（ｂ）制御データを複数の条件付きアクセスデバイスに送信することであって、ただし、前記制御データは、前記条件付きアクセスデバイスの各々に前記少なくとも２つのキーのうちの１つを出力させるように構成されること、（ｃ）前記キーの１つ（または複数）を含むか、または、前記キーの１つ（または複数）に対応する、前記悪意のあるソースによって再送信されたキーおよび／またはビデオ（および／または他のデータ）を受信すること、（ｄ）どのキーが、悪意をもって送信されたデータに含まれるか（または、悪意をもって送信されたデータを作成するために使用されたか）の知識を使用して、例えば、悪意のあるソースが、受信されたキー（またはビデオ）を導出することができるデバイスを有することを推論することにより、悪意のあるソースのセキュリティモジュールについての情報を得ること、（ｅ）海賊グループについてのさらなる知識を使用して、少なくとも１つの悪意のあるデバイスが識別されるまで、上記のステップの一部または全部を繰り返すこと、および、（ｆ）例えば、「キル（kill）」メッセージを送信することによって、または、将来のキーを与えないでおくことによって、前記悪意のあるデバイスを無効にすること。

【０３６９】

上記の実施例は、主としてオーディオビジュアルコンテンツのビデオ部分に関連して説明されるが、同じ技術はオーディオおよび他の種類のコンテンツにも同様に適用可能である

。
【０３７０】
（攻撃および対策の実施例）
このセクションでは、一部の例示的攻撃および対策を説明する。
【０３７１】
敵対者は、オーバーレイベースのフォレンシックマークが出力に埋め込まれないように、プレイヤのオーバーレイ機能を無効にしようと試みる場合がある。この攻撃に対する１つの対策は、コンテンツ所有者が、圧縮されたオーディオビジュアルデータストリームにマークを直接埋め込むことである。
【０３７２】
敵対者は、フォレンシックマークを除去するために、複数のデバイスからの出力を比較し、かつ結合しようと試みる場合がある。例えば、デバイスの出力が異なる位置において、敵対者は、ビデオを結合し、ビデオを削除／品質低下させ、確率論的に（probabilistically）バリアントを選択し、最も一般的なバリアントを選択し、または、他の技術を使用して、トレースがより困難であるコピーを作成するように、試みる場合がある。共謀者の数がかなり少ないことが知られている場合、マーク復旧プロセスは、各マークを使用することによって、共謀者を識別することができ、復旧されたマークを出力することがグループのどのメンバーにもできていなかったであろう、いかなる共謀者のグループも排除する。コンテンツプロバイダはまた、フォレンシックマークを（平均化および選択を困難にする）多数の異なるバリアントと共に含めることもできる。目立つフォレンシックマークを使用して、悪意のある攻撃によって引き起こされた品質低下または難読化に対する強化された耐性を得ることもできる。必要な場合、後続のコンテンツは、改良されたフォレンシックマーキング技術により作成されることが必要である場合がある。コンテンツ所有者はまた多数のマークを埋め込むことができ、個々のデバイスによって埋め込まれたシーケンスと、復旧されたコピーとの間の相関を探すこともできる。
【０３７３】
敵対者は、正当なプレイヤをエミュレートしようと試みる場合がある。関連する攻撃には、（悪意のあるものと思われる）誤った方法で動作するよう正当なプレイヤ内のインタープリタを変更することが含まれる。この場合、正当なプレイヤとエミュレーションの間の差異（これには、文書化されていないオペレーション、ネイティブコードサポート、タイミング／パフォーマンス、暗号化キーなどが含まれるが、これらに限定されない）を使用して、正当な環境と悪意のある環境とを区別することができる。例えば、ネイティブコードプレイヤアップデートを、脆弱性を是正するために利用することもできる。
【０３７４】
敵対者は、コーデック、デバイスドライバ、ディスプレイ／スピーカデバイス、または、圧縮解除の直前または直後のいずれかにおいてコンテンツを受信する他のコンポーネントを危殆化させることによって、デジタルコンテンツを抽出しようと試みる場合がある。同様に、これらまたは他のコンポーネントの悪意のあるバージョンが挿入される場合がある。このような攻撃に対応するため、将来のコンテンツは、出力デバイスを認証することによって、および、危殆化されたデバイスにコンテンツを提供することを拒否することによって、対応することができる。コンテンツは、セキュリティアップグレード（ネイティブコードパッチなど）を脆弱なデバイス、または未知のデバイスに配信することができる。フォレンシックマーキングもまた、このように危殆化されたコピーをトレースするために使用される場合があり、フォレンシックマーキングは心理的および合法的な抑止力、ならびに、失効させるための海賊行為者のデバイスを識別する方法を提供する。
【０３７５】
敵対者は、すべてのＴＲＡＰオペレーションへの入力および出力を「事前記録」し、悪意のあるインタープリタによりこのデータを使用しようと試みる場合がある。ネイティブコードを使用して、必要とされるストレージの総量を過剰にすること、フォレンシックマーキングを使用して危殆化をトレースすること、または、オリジナルデバイスのセキュリテ

ィをアップグレードすることなどの多数の方法によって、この攻撃を防止することができる。ソースデバイスが実際に危殆化されない（例えば、それが悪意のあるコンテンツコードを実行中である）場合、プレイヤ暗号化オペレーションを使用して、プレイヤが実際に、署名が付けられている、復号化されているなどのコンテンツコードを実行中であることを、確実にすることができる。

【０３７６】

悪意のある敵対者は、セキュリティチェックを回避するように意図された方法で誤って動作する、悪意のあるインタープリタを作成しようと試みる場合がある。特定の例は、悪意のあるインタープリタが、例えば、予想される計算上の中間物を検索し、これを置換することによって、コンテンツコードに無効なＲＳＡ署名を受け入れさせようと試みる場合である。このような攻撃を防止するために、プレイヤは、単に（ｎを法として）署名を３乗し、期待値に対して比較するという計算以外の計算を使用して、署名を検証することができる。例えば、コンテンツコードは初期値で乗算をし、次いで、結果が、期待値と初期値の３乗倍にしたものとが等しいことを検証することができる。署名を検証するもう１つの方法には、定数を署名に加算し、次いで、結果が正しいことを検証することが含まれる。署名検証コードを他のオペレーション（対称暗号化キーの修正など）と組み合わせて、結果として生じる暗号値が、署名が有効である場合にのみ正しくなるようにすることもできる。自己書き換えコード、コード難読化技術、ネイティブコード、および、他の任意の対策もまた、必要に応じて使用することができる。

【０３７７】

敵対者は、悪意のあるファームウェアをメディアインターフェース（光ディスクドライブなど）に挿入しようと試みる場合がある。コンテンツコードに自己チェックを実行させてそれ自体を認証させ、データがメディアから正しくロードされることを検証することによって、この攻撃を検出することができる。ネイティブコードおよび／またはインタープリタコードをドライブによって実行して、この攻撃を検出することもできる。コンテンツはまた、攻撃によって変更されるであろうデータを含むこと、および、それが変更される場合を検出することによって、この攻撃を検出することもできる。

【０３７８】

敵対者は、例えば、データを正当なメディアからではなく、ハードドライブ上のファイルからロードする、悪意のあるデバイスドライバを使用することによって、セキュリティ機能を含むメディアをエミュレートしようと試みる場合がある。ドライブが有効な暗号化キーを有することを検証することによって、この攻撃を検出し、防止することができる。敵対者が、キーを正当なドライブから抽出する方法を発見する（、それにより、悪意のあるエミュレータが、正当なドライブの暗号化オペレーションをエミュレートすることができるようになる）場合、追加のドライブ認証ステップ（ドライブのためのネイティブコードの提供、タイミングチェック、非標準機能性のためのテストなど）が実行される場合がある。加えて、（例えば、ＴＲＡＰ＿ＤｅｖｉｃｅＤｉｓｃｏｖｅｒｙ、ネイティブコードなどを使用して、）悪意のあるデバイスドライバを検出することができる。

【０３７９】

敵対者は、特定のタイトルのセキュリティコードにおける脆弱性を不当に活用しようと試みる場合がある。いくつかの対策が可能である可能性があるが（例えば、アップデートされたセキュリティロジックを配信することによる）、主な解決策は、コンテンツ作成者が、将来のコンテンツがより注意深く作成されて同じ脆弱性を有することのないことを、確実にすることである。

【０３８０】

（追加の考慮事項および変形形態）

メディアは、制限された回数の使用（例えば、いずれかの３つの音楽ビデオを見ることができる）、または、アクセスの継続期間（例えば、最初の視聴後３日間でアクセスが期限切れになる）を可能にするように構成される場合がある。その後、ユーザは、将来の（または無制限の）再生のための許可を得るために支払いを要求される場合がある。視聴カウ

ンタ（および／または、他のセキュリティおよび状態情報）が、メディア自体の上に、および／または、プレイヤ内に、および／または、リモートサーバ上に格納される場合がある。例えば、無制限の再生を許可するデータがメディアの書き込み可能部分に配置され、それにより任意のプレイヤ上で再生できる場合がある。代替として、この許可をプレイヤの不揮発性メモリ内に格納することができ、および／または、ネットワークを介してサーバにアクセスして許可が得られるように、コンテンツコードを構成することができる。

【０３８１】

所定の条件が満たされる（例えば、支払い、所定の組合せの他のタイトルの再生、パズルが解決されること、所定の時間が経過していることなど）まで、メディア上のコンテンツの部分（または全部）にアクセスできなくすることができる。一実施例では、コンテンツコードは、購入またはアクティベートされるまで動作できない。アクティベーションプロセスはローカルで実施される場合があり、または、リモートサーバとの対話を含む場合がある。例えば、コンテンツコードはリモートサーバにアクセスし、ローカル再生デバイスをサーバに対して識別し、ユーザおよびサーバと対話して支払いを生じさせ、一部のキー、解釈可能なセキュリティコード、または、コンテンツのデコードを可能にする他のデータを受信することができる。購入が完了した後、受信されたデータを使用して、コンテンツのデコードをアクティベートする。

【０３８２】

ユーザが新しいプレイヤを購入するとき、古いプレイヤから新しいプレイヤへのプレイヤ不揮発メモリの移行を可能にするために、プロビジョン（provision）が含められる場合がある。一実施形態では、複数のデバイス間で、スマートカードまたは書き込み可能メディアなどの物理的キャリア（carrier）上のデータを移動させることによって、このプロセスは実行される。代替として、この転送は、有線もしくは無線ネットワークまたは赤外線ポートなどのデジタル接続を介して実行される場合がある。最大のセキュリティのために、この転送は、信頼できるサーバデバイスを通じてなされる（または信頼できるサーバデバイスによって仲介される）場合があり、この信頼できるサーバデバイスは送信側デバイスおよび受信側デバイスと通信して、データが正しく転送されること、および、データが受信側デバイス上で使用可能になる前に送信側デバイス上で除去される（または無効にされる）ことを確実にする。幅広い範囲の異なるコンテンツタイトルによって書かれたデータに対応するため、単一のサーバは、どれだけの数のタイトルが不揮発性ストレージを使用するかについて知っている場合があり、または、プレイヤは（例えば、不揮発性メモリスロット自体内で識別されるような）複数のサーバと対話する場合がある。送信側および受信側の両方における暗号化キーを使用して、転送のためのセキュリティを提供することができる。

【０３８３】

メディアは複数のデータエンコード方法を含む場合がある。例えば、単一の光ディスクには、高密度スタンプ付き部分、追記部分、および記録可能部分が含まれる場合がある。

【０３８４】

位置および距離測定技術を使用して、受信側デバイスが、認められない物理的位置内（無許可の国内、家庭内使用専用のコンテンツを提供するホームネットワークサーバから遠すぎるなど）に存在しないことを、確実にすることができる。ラウンド通信の往復時間と光速とを乗算して、デバイスまでの距離について上限を決定することができる。全地球測位システムおよび無線信号の減衰（例えば、デバイスが、８０２．１１ｂ、ブルートゥースなどに関して範囲内である場合、または、デバイスが共通の無線／衛星信号にアクセスすることができる場合など）もまた、位置および／または近接性を推定するために使用される場合がある。共通のワイヤ（電話接続、家庭の電力回路など）を共有するためのデバイスの能力もまた使用することができる。インターネットＩＰアドレスおよび電話番号もまた、位置情報を得るために使用される場合がある（例えば、領域コーディングアプリケーション（region coding application）、局地的なスポーツブラックアウト（blackout）、デフォルト言語／通貨オプションの選択などのため）。位置情報はフォレンシックマー

クに含まれる場合がある。

【０３８５】

プレイヤは、揮発性または「壊れやすい不揮発性」メモリストレージ領域を提供することができ、これらの領域は、挿入される次のディスクへコンテンツがデータを転送することを可能にするが、コンテンツがその直後に削除される、メモリスロットなどである。電源が落とされるときにコンテンツが消去されるかどうかは問題ではない状況において、揮発性メモリを、一時的ストレージのために使用することができる。

【０３８６】

ニューラルネットワークをフォレンシックマーク検出／解析の際に使用して、最適なセキュリティポリシなどを定義することができる。

【０３８７】

コンテンツコードは、適切な認証が存在する場合、記録可能メディアからの再生を可能にするように構成される場合がある。例えば、この機能を使用して、消費者用のメディアを記録することができる光ディスク焼付けキオスク（optical disc burning kiosk）（または他のシステム）を作成して、例えば、小さい店が多大な在庫を有する必要なく、予測できない消費者の需要を満たすことができる。キオスクは、内部ストレージ（ハードディスクなど）およびデータネットワークを含む、任意のソースまたはソースの組合せから、（生）コンテンツを検索することができる。記録されたデータをキオスクによって修正して（任意で、キオスク内に含まれるインタープリタにおいて実行するインタープリタコードを使用して）、例えば、識別フォレンシックマークを導入し、データの部分を再暗号化し、最新のセキュリティロジックをコンテンツに関連付け、消費者によって選択された複数のコンテンツ部分を結合し、ユーザ要求／プリファレンスに対応し、コンテンツを（再）圧縮するかまたはフォーマットを変更して、メディアまたはプレイヤデバイスの容量またはコーデック要件を満たすことなどができる。

【０３８８】

再生の記録および許可を暗号的に認証する値もまた含まれる場合がある。例えば、これを、信頼できるパーティによって発行されたデジタル署名にし、ネットワークを介してキオスクに転送し、メディア上に焼付け、再生中にコンテンツコードによって検証することができる。このデジタル署名には例えば、メディア識別子、ＳＨＡ－１を使用して計算されたコンテンツのハッシュツリーのルート、キオスクの記録ドライブの識別子、発行日、および、宛先メディアのシリアルナンバが含まれる場合がある。

【０３８９】

キオスクにはまた、請求書、クレジットカード、または他の支払いを受けるためのインターフェース、ユーザ選択を受信するためのタッチスクリーンまたは他の入力デバイス、カスタマイズされたボックスインサート、メディア表面、レシートなどを印刷するための機能、新しいコンテンツデータを検索し、新しいユーザインターフェースコンポーネントおよび広告／オファーを受信し、支払いを検証および処理し、エラー状態を報告するためなどのネットワーク機能、および、データを所望の出力フォーマットに変換し、カスタマイズされたメニューを作成するためなどのオーディオ／ビデオ操作機能が含まれる場合もある。

【０３９０】

物理的メディア（光ディスクなど）への書き込みにおいて高パフォーマンスを実現するため、メディア記録デバイスは複数の記録レーザを同時に使用することができる。物理的サポートバリアをメディアの周囲に配置して、遠心力による粉砕または損傷を防止することができる。ミラーまたは他の光要素を使用して、レーザビームをメディアの周りで移動させて、光メディアを物理的に回転させる必要性を低減またはなくすことができる。非レーザベース（non-laser-based）記録技術が使用される場合がある（例えば、インクジェットプリンタで使用されるものに類似した高解像度技術を使用して、円形の基板上にエッチング物質のマイクロ液を滴下し、次いで、基板を保護層によりコーティングすることによって、記録するなど）。

【０３９１】
メディアは、再生デバイスに物理的に存在する必要はない。例えば、メディアは無線または他のネットワークを介してプレイヤデバイスと通信することができる。一実施形態では、メディアはコイルと、少なくとも１つの半導体チップとを含み、これらは（ａ）誘導コイルから電力を受け取り、（ｂ）誘導コイルを使用して、プレイヤ（または他のデバイス）によりデータを送受信し、（ｃ）再書き込み可能部分であっても、そうでなくてもよいローカルメモリからコンテンツ（サポートされる場合、コンテンツコードを含む）を検索し、（ｄ）コンテンツコードまたは他のデバイスによる使用のために暗号化ハンドシェークを実行して、メディアの妥当性を認証するように構成される。複数のこのようなメディアが存在する可能性がある（例えば、複数のメディアを含むことができるプレイヤデバイス内の自由に取り外すことができるトレイに含まれる）場合、アドレッシング方法を使用して、どのメディアが所与の時間に通信中であるべきであるかの選択を可能にすることができる。

【０３９２】
コンテンツコードは、任意の方法の外部デバイスだけでなく、プレイヤデバイス内に含まれる場合のあるコンポーネント（ソフトウェアプログラムなど）と対話することができる。例えば、スマートトイ（smart toy）（または他のデバイス）は、デバイス（またはこのようなデバイスの汎用クラス）をサポートするコンテンツコードと対話することができる。このような対話は、任意の種類のデータ交換を含むことができ、これらのデータ交換には、コマンド（例えば、おもちゃの車が動く方向）、オーディオデータ（例えば、デバイスが発する音、デバイスによって記録された音など）、イメージ／ビデオデータ（例えば、デバイスが表示するデータ、デバイス内のデジタルカメラからのデータなど）、ロジック（例えば、デバイスによる実行のため）、イベント通知などが含まれる。一実施形態では、コンテンツコードは外部電気機器およびデバイス（ライト、暖房、エアコン、ホームセキュリティシステム、電気機器、ペットケア／フィーディング（feeding）デバイス、ビデオレコーダなどが含まれるが、これらに限定されない）と対話し、これらを制御するのに役立つ。プレイヤデバイスと、外部デバイスとの間の通信の方法には、従来のオーディオビジュアル出力を通じてプレイヤデバイスからデータを（例えば、マイクロフォン、ＣＣＤ、または、リモートデバイス上の他のセンサへ）出力すること、および、リモートコントロール（例えば、赤外線）インターフェースを介してデータを受信することが含まれる場合がある。他のサポートされた通信方法には、無線および有線ネットワーキングおよびデータ変換プロトコルが含まれる可能性がある。一部のプレイヤデバイスは、すべての所望の外部対話を可能にするために必要なインターフェースを有していない場合があり、この場合、（例えば、双方向データ通信および電力の物理的インターフェースを通じて、）外部インターフェースモジュールが直接接続される場合がある。外部デバイスはまた、プレイヤデバイスに配信されるコード（インタープリタコードを含む）を格納するための不揮発性メモリまたは他のストレージを備え、例えば、このような外部デバイスによって提供された機能を利用する際にプレイヤデバイスおよび／またはコンテンツコードを支援することもできる。

【０３９３】
再生システムの態様は別々に実装される場合があり、互いに通信する場合がある。例えば、あるプラットフォーム（パーソナルコンピュータなど）上で、発見および／またはセキュリティ評価機能を実装する部分をメインプレイヤアプリケーションから分離することが有用である場合がある。なぜならば、これらのコンポーネントは他のコンポーネントよりも頻繁なアップデートを必要とする場合があるからである。例示的実施形態では、コンテンツコードは、悪意のある、無許可の、および／または疑わしいソフトウェアの存在などの、既知のセキュリティの問題を検出するように構成されたセキュリティスキャナと通信する。スキャナはまた、システムの状態についての情報をコンテンツコードおよび／またはプレイヤアプリケーションに提供するが、この情報は、何のアプリケーションが実行されているか、何のデバイスドライバがインストールされるか、何のデバイスが接続される

ことが知られるか、などである。スキャナはまた、システムが安全な再生のための要件を満たすかどうかなどの現在のシステムのセキュリティ状況の評価を報告することもできる。新しい海賊行為のリスクは瞬時に発生する可能性があるので、例えば、新しいソフトウェアの弱点が公にされる場合、スキャナプログラムは好ましくは、インターネットを介してアップデートをダウンロードすること、または、コンテンツコードにより配信されたデジタル署名付きアップデートを受信することなどによって、頻繁にアップデートされる。ユーザは、アップデートサービスのために課金される場合がある（例えば、ユーザに対応する支払い手段に自動的および定期的に課金すること、支払いがうまくいかなかった場合にユーザに自動的に通知すること、および／または、支払いが最終的に認められない場合、ユーザのアクセスを自動的に打ち切ることによる）。スキャナはまたユーザにその結果を通知し、自動的に問題を是正し、および／または、自動的に対策またはアップデートをダウンロードすることもできる。スキャナはまた、例えば割り込みまたはイベントを登録することによって、コンテンツコードまたは他のコンポーネントに、システム／セキュリティ状況における変化を通知することもできる。コンテンツコードおよびスキャナなどの複数のコンポーネント間の通信を、暗号的に保護することができる。例えば、スキャナはその結果に、チャレンジ値と共にデジタル署名して、リプレイ攻撃を防止することができる。スキャナは、アンチウィルススキャナまたは他の脆弱性スキャナ（vulnerability scanner）と統合される場合があるが、その機能性は従来のウィルス検出器とは異なり、単にコンピュータの所有者またはオペレータの利益を保護することとは対照的に、このスキャナは、コンテンツ所有者によって使用可能であるマシンの状態についてのアテステーションとして、その結果の暗号化認証を提供する。

【０３９４】

コンピュータを検索して無許可の（例えば、海賊行為を受けた）コンテンツを検出するように、自動化されたプログラムを構成することもできる。このようなコンテンツが発見される場合、プログラムは、権利を侵害していると考えられるマテリアルをユーザに通知することができる。プログラムはまたコンテンツコードまたは他のセキュリティコンポーネントと対話（、およびアテステーションをこれらに提供）して、例えば、海賊行為を受けたマテリアルがないと思われるデバイス上でコンテンツの再生を許可することもできる。マテリアルが許可されるかどうかを判断するため、スキャナはライセンスファイル、デジタル署名、既知のファイルハッシュなどを探すことができる。

【０３９５】

本発明の要素（セキュリティ機能および非セキュリティ機能を含む）は、以下に限定されるものではないが、非プログラマブルコンテンツ配布システム、インタープリタまたは仮想マシンの代わりにネイティブコードを使用するコンテンツ保護システムと共に、いずれかのソフトウェアアプリケーションまたはデバイスの一部として、コードが（インタープリタ／仮想マシンを使用して処理されるのではなく）常にネイティブで実行されるプログラマブルシステム上で、使用される場合があり、他のシステム（ビデオゲームコンソールコピー保護およびセキュリティシステムを含む）において使用される場合がある。例えば、個々の態様を、以下に限定はされないが、コンピュータセキュリティアプリケーション、ネットワークセキュリティアプリケーション、非プログラマブルなアンチパイラシシステム、フロードスクリーニング（fraud screening）システム、電子メールフィルタリングなどのために使用することができる。

【０３９６】

本発明は、デジタルデータ放送システム（以下に限定されないが、すべての種類のテレビ放送、衛星送信、ケーブルネットワークを介した通信、インターネットマルチキャストなどが含まれる）と統合される場合がある。放送信号は、その信号または他のコンテンツ内の伝送されたコンテンツと共に使用するために、コード（セキュリティ関連情報および対策を含む）を伝送することができる。放送信号はまた、他のコンテンツタイトルによって使用可能なコンテンツおよび／または他のデータを伝送することもできる。セキュリティ機能はまた、テレビ信号を保護するために使用された従来の条件付きアクセスカードおよ

びシステムと統合される（、および、これらと通信する）場合もある。

【０３９７】

本発明の複数のコンポーネントが対話するとき、これらのコンポーネントは暗号技術を使用して、それらの通信を保護することができる。このようなセキュリティ機能には、対称暗号化および／またはメッセージ認証コード（または他のチェック）をデータ上で使用することが含まれる場合がある。このような通信を保護するために使用される対称キーを、公開キー暗号技術（デジタル署名／証明書、指数キー合意、公開キー暗号化／復号化、チャレンジ－レスポンスプロトコルなどを含む）を使用してネゴシエートすることができ、または、非対称暗号技術をデータに直接適用することができる。（例えば、失効リストをチェックすること、リモートサーバと対話することなどにより、）対称および非対称キーの失効状況を実施することができる。対称暗号化プロトコルを使用して、キーをネゴシエートすることもできる（放送暗号化技術を使用することを含むが、これに限定はされない）。使用されるセキュリティ手段は、すべての通信に対して同じである必要はない。（例えば、制御メッセージ、セキュリティ関連パラメータ、キー、および／または、ＭＰＥＧ「Ｉ」フレームは暗号化される可能性があるが、他の通信は暗号化されない可能性がある。）ハンドシェークを保護するために使用されるキー（各キーまたはキーのセットに関係付けられた、１つまたは複数の関連証明書を含む）を、コンテンツコードを使用して、および／または、デバイスによって直接検証することができる。コンポーネントの通信が暗号的に保護するために適切である可能性のある場合、そのコンポーネントの例には、以下に限定されるものではないが、光ディスクドライブ、他のメディアインターフェース、メディアインターフェースデバイスドライバ、プレイヤアプリケーション、コンテンツコード実行スレッド（例えば、同時に実行されている異なるコンテンツタイトルから、または、同じコンテンツタイトルに属する別のスレッドから）、オペレーティングシステム、ビデオデバイスドライバ、ビデオプロセッサ（またはビデオカードなど）、オーディオデバイスドライバ、オーディオプロセッサ、任意の方法の外部デバイス（ローカルで接続されるか、ネットワークを介してアクセスされるか、バスを介してアクセスされるかなどにかかわらず）、他のアプリケーション（セキュリティスキャナ、オーディオミキサ、ビデオイフェクトプロセッサ（video effect processor）など）、コーデック、およびセキュリティモジュールが含まれる。

【０３９８】

１つまたは複数の証明書を使用して認証されるキーを使用して、デジタル署名を作成することもでき、デジタル署名と他のセキュリティ値とを組み合せることができる。複数のバージョンの署名を提供することができる（例えば、異なるキー／署名者、キーサイズ、アルゴリズムなどにより作成される）。

【０３９９】

実施例は、限定を課すものとして解釈されるべきではない。簡潔にするため、特に、アイテムが類似のリストに含まれる状況では、リストは必ずしも完全に列挙されているとは限らない。「～を含む」という語は、「～を含むが、これに限定されない」ということを意味する。同様に、「例えば」は、説明されているものの一般性を限定しない、例示的な例を示す。同様に、「など」は、追加の可能性があることを意味し、リストは、必ずしもすべての知られている可能性を列挙するものと解釈されるべきではない。

【０４００】

例示的実施形態では、コンテンツを特定のプレイヤに合わせてカスタマイズすることができる。この場合、コンテンツは、単一のプレイヤまたは少数のプレイヤ上でのみ再生可能であるが、受信側デバイス上での再生のために必要とされないコードは、送信される必要はない。したがって、情報をユーザに送信することが困難であるか、高価であるか、または遅いとき、例えば、ストレージスペースが制限される場合、または、コンテンツが低速ネットワーク接続を介して送信されなければならない場合、この手法は特に有用である。しかし、コンテンツはなお、プレイヤに問い合わせて、再生環境が適切に安全であることを検証することができる。

【０４０１】
再生が割り込まれない、またはゆがめられないことを確実にするため、プレイヤのインタープリタのための特定の最低限のパフォーマンス標準を必要とすることが有用である可能性がある。
【０４０２】
例示的実施形態では、コンテンツをあるデバイスから別のデバイスへ交換できるように、システムおよび方法を構成することができる。このような交換の特定のセキュリティ特性は、信頼できる（例えば、パブリッシャにより運営される）サーバとのオンライン通信が利用可能であるかどうかなどの要素によって決まる。コンテンツが転送される形式は、コンテンツによって実装されたセキュリティポリシ、および、デバイスのハードウェア機能によって決まる。例えば、両方のデバイスが安全なインタープリタを含む一実施形態では、送信側デバイスは生の暗号化されたコンテンツ（オリジナルメディア上に格納されるか、または、別のキーにより暗号化され、任意でウォータマークが含められる）を、再生を制御するためのコードと共に送信する。送信側デバイスによって、受信側デバイスに合わせて再生制御コードをカスタマイズすることができる。別の場合では、送信側デバイスは、出力ポートおよび送信先デバイスのセキュリティ特性が受け入れ可能であることを検証し、共有キーについて送信先デバイスとネゴシエートし、コンテンツを復号化およびウォータマーキングし、コンテンツを共有キーにより再暗号化し、再暗号化されたコンテンツを送信先に送信することができる。
【０４０３】
十分な不揮発性ストレージを有するプレイヤを使用して、インタープリタから呼び出されるアップデート可能なコードを格納することができる。例えば、常に特定のパブリッシャ用の最新セキュリティコードを格納するように、プレイヤを構成することができる。この場合、より新しいバージョンのセキュリティコードが見つかる場合、（例えば、新しいコードにおけるデジタル署名を検証した後）古いバージョンはアップデートされる。このように、より古いコンテンツは、新しいコンテンツ上に含まれるセキュリティアップデートから利点を得ることができる。（例えば、前述の安全なメモリ方法を使用して、この手法を実施することができる。）もう１つの実施形態では、現在の日付／時間をプレイヤから取得し、最新の既知のセキュリティアップグレードの日付／時間と比較することによって、コンテンツは、プレイヤが現在のセキュリティアップデートを含むことを要求することができる。このように、コンテンツは、プレイヤが合理的に最新のセキュリティアップグレードを有することを確実にすることができる。
【０４０４】
一般に、コンテンツ保護システムは、正当なユーザによる正当なアクションにおいていかなる可視的な役割を果たすことも回避すべきである。それにもかかわらず、エラーの報告または情報の提供のためなど、一部のユーザインターフェース要素は必要である。コンテンツが複数のサポートされた出力品質から選択することができる（例えば、プレイヤが不十分なセキュリティを提供する場合は「レガシー」品質、セキュリティが十分なものである場合は「高」品質）場合では、ユーザに出力品質を通知するためにインジケータが有用である可能性がある。例えば、一実施形態では、コンテンツの制御下にある緑色の発光ダイオード（ＬＥＤ）は、出力が高品質である（すなわち、セキュリティが十分なものである）ことを示し、オレンジ色のＬＥＤは、低減した品質（すなわち、最低限のセキュリティ）を示し、点滅する赤色のＬＥＤは、プレイヤが失効しているために出力が提供されないことを示すことができる。もう１つの実施形態では、（既知の場合は、ユーザの言語により）短く発話された、または書かれた通知が提供されて、セキュリティの状況が報告される。堅牢なおよび／または脆弱なウォータマークの有無などの他の要因に基づいて、より低い品質出力またはより高い品質を報告および／または使用するかどうかを決定することができる。必要な場合、品質低下モジュールをプレイヤと共に含めて、コンテンツがセキュリティまたは他の理由のために再生の品質を（例えば、レガシーフォーマットの品質へ）低下させることができるようにする。（例えば、高精細度テレビ信号をＮＴＳＣ解像

度へ変換するため、または、高解像度マルチチャネルオーディオを２チャンネルＣＤ品質オーディオに変換するために、品質低下モジュールを含めることができる。）

【０４０５】

メディアインターフェースおよびプレイヤインタープリタが十分なパフォーマンスを提供する場合、別の復号化モジュールにおいてではなく、インタープリタにおいてバルク復号化およびウォータマーク埋め込みを処理することができる。コンテンツがそれ自体を直接復号化できるようにすることは、攻撃者が復号化モジュールに対して攻撃をしかけないようにすることを確実にすることなどの、いくつかのセキュリティの利点を提供することができる。インタープリタのパフォーマンスが十分なものである場合、コンテンツ圧縮解除をインタープリタ内でも実装して、単一のプレイヤコーデックタイプを標準化する必要性を回避することもできる。

【０４０６】

本明細書で開示した技術およびシステムのための特定のハードウェアサポートを有していない（パーソナルコンピュータなどの）プラットフォーム上のインタープリタを使用する実装が好ましいが、多数のインタープリタ関数を専用ハードウェア内に実装することができる。用途（application）によって、専用の実装はコストまたは電力消費量を節約する場合があるが、低減した機能を提供する場合もある。

【０４０７】

物理的メディア上のコンテンツを受信する実施形態では、実質的にいかなるメディアフォーマットも使用することができる。（ＣＤおよびＤＶＤなどの）光ディスクは高ストレージ密度を低コストで提供するが、他のストレージシステムを使用することもでき、これらのシステムには、磁気メディア、ホログラフィックメモリ、バッテリバックＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、およびフラッシュメモリが含まれるが、これらに限定されるものではない。メディアのストレージ容量を、多数の異なるタイプのデータを格納するために使用することができ、これらのデータには、（様々なコンピュータプラットフォームのためのデコード方法を実装する実行可能プログラム、本明細書で開示した方法を使用することにより保護されたコンテンツなどの）本明細書で開示した技術およびシステムに関連する情報、ならびに、（無関係の実行可能プログラム、レッドブックＣＤオーディオなどの保護されていないコンテンツ、他のセキュリティスキームを使用することにより保護されたコンテンツなどの）本明細書で開示した技術およびシステムとは直接的に関連しないデータが含まれる。

【０４０８】

メディアは、暗号化計算を実行して、メディアが無許可のコピーではないことをプレイヤが検証することができるようにするための、耐タンパ性を有する回路を含むことができる。このような機能は、電気的インターフェースを使用するメディアについて実施するために最も単純なものであるが、光メディアでさえ暗号化機能を含むことができる。例えば、（Ｇａｕｍｅｔらの米国特許第５６４０３０６号明細書に記載の非接触スマートカードなどの）非接触暗号化モジュールを光ディスクに加えるか、または埋め込むことができる。暗号化メディア認証が好ましいが、他の認証メカニズムを代わりに使用することもできる。例えば、背景技術において知られる汎用メディア認証方法は、コピーが困難な位置（市販の記録可能メディアまたはドライブを使用する書き込み可能ではない領域など）にシリアルナンバを書き込むこと、および、オリジナルの物理的メディアの様々な特性のデジタル署名付き「記述」を含めることを含む。言うまでもなく、暗号化メカニズムは、既存のメディアを危殆化させるための方法を攻撃者が発見する場合でも、プレイヤに対するいずれの変更を必要とすることなく、将来のメディアを改善されたセキュリティと共に発行することができるという、利点を提供する。

【０４０９】

多数の消費者はすでにレガシーフォーマットのコンテンツに投資をしているので、これらのレガシーフォーマットをサポートするように、本明細書で開示した技術およびシステムを実装するプレイヤを構成することができる。同様に、異なるバージョンのインタープリ

タを特定のプレイヤによってサポートすることができる。この場合、プレイヤはメディアまたはコンテンツを解析して、使用すべき適切なセキュリティシステムを識別する必要がある。例えば、デジタルビデオプレイヤは、ディスクがＣＳＳを使用するレガシーＤＶＤであるか（および、そうである場合、ＣＳＳ復号化システムを選択する）、または、本明細書で開示した技術およびシステムを使用するＤＶＤであるか（および、そうである場合、言語ベースの復号化システムをアクティベートする）を検出する可能性がある。コンテンツに含まれる堅牢なウォータマークを使用して、あるセキュリティシステムにより元は保護されていたコンテンツが、その元々の保護を欠くフォーマットにコピーされているかどうかを検出することができる。例えば、コピーすることができないコンテンツはウォータマークを含み、任意の他のフォーマットのコピー（例えば、保護されていないフォーマット）を発見したいずれのデバイスも、そのコピーを無許可のものとして認識し、（例えば）再生を拒否することができることを、示すことができる。

【０４１０】

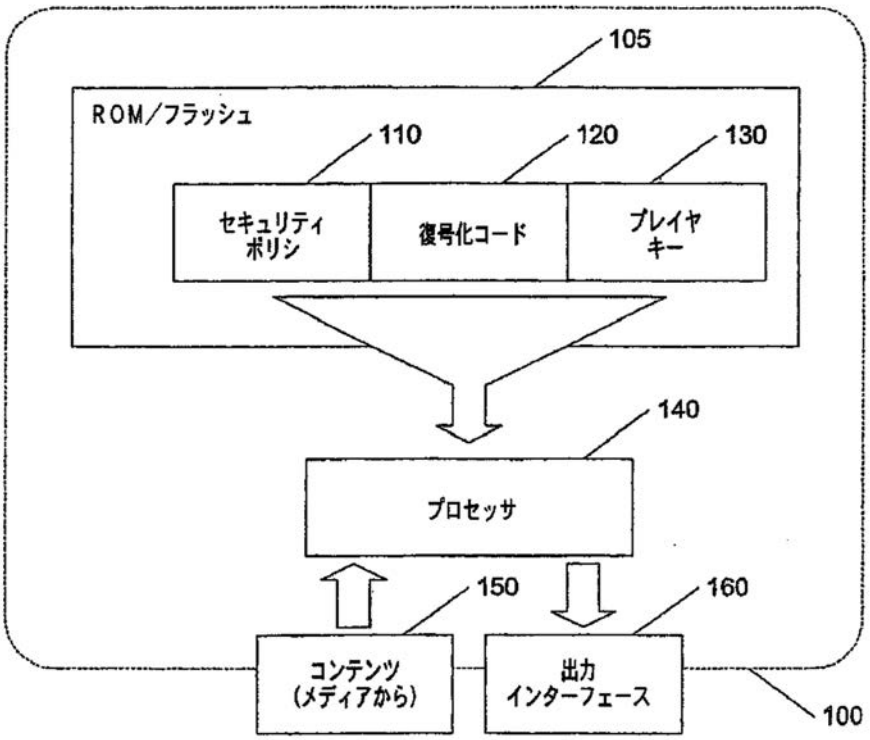
本明細書で開示した技術およびシステムを幅広い種類のコンテンツタイプと共に使用することができる。これらのタイプには、オーディオ、静止画像、ビデオ、３次元イメージ、および３次元ビデオが含まれるが、これらに限定されるものではない。

【０４１１】

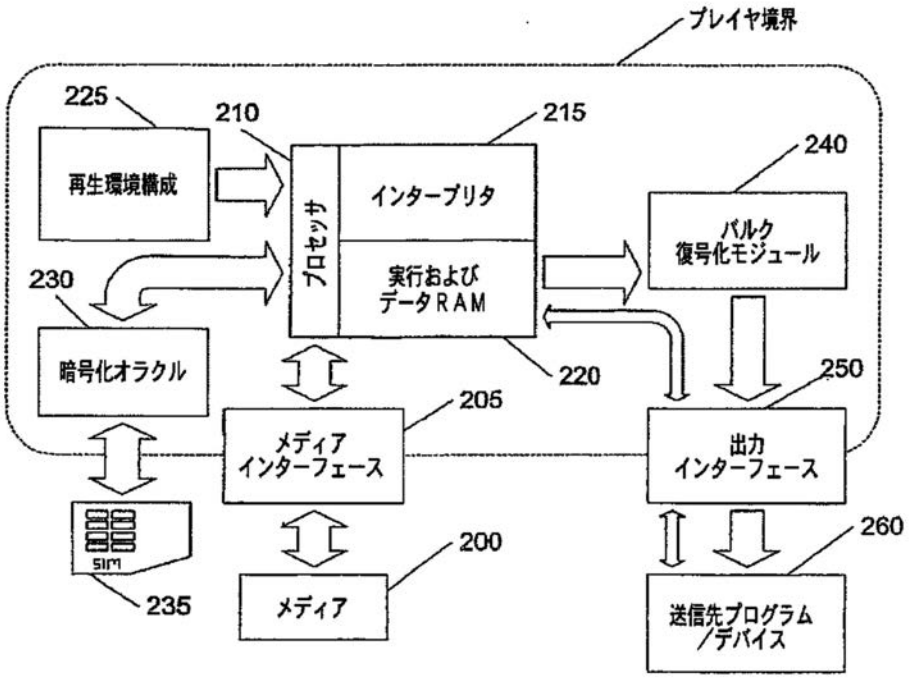
本明細書で開示した技術およびシステムを、様々な物理的デバイスにおいて実施することもできる。１つのデバイスのみが、コンテンツを復号化することを担う場合、セキュリティポリシはそのデバイスにより実装されることが好ましい。しかし、出力デバイスおよび（オーディオイコライザまたはミキサなどの）中間処理デバイスもまた、本明細書で開示した技術およびシステムから、および／または、このような技術およびシステムによってそれらのセキュリティを検証するために使用することができる問い合わせ機能を提供することによって、利点を得ることができる。一実施形態では、ホームエンタテインメントサーバは、コンテンツをダウンロードし、格納し、かつ管理して、セキュリティが成功裡に検証されている（スピーカ、ヘッドフォン、ビデオディスプレイなどの）再生デバイスにコンテンツを転送する。これらのデバイスへの接続は、好ましくは、本明細書で開示した技術およびシステムならびに送信先デバイスの共同制御下で暗号化されて、転送中のコンテンツの盗難を防止する。

【図１】

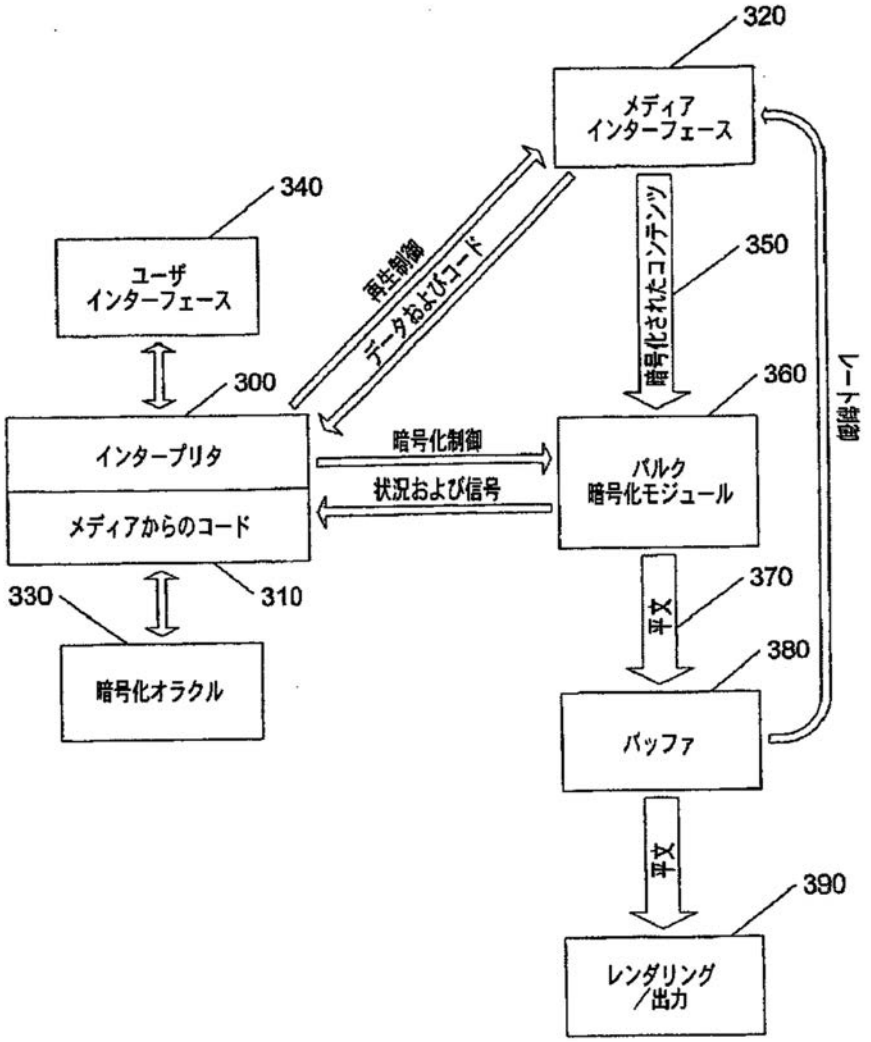
背景技術


105
ROM／フラッシュ
110
120
130
セキュリティポリシ
復号化コード
プレイヤキー
140
プロセッサ
150
160
コンテンツ（メディアから）
出力インターフェース
100

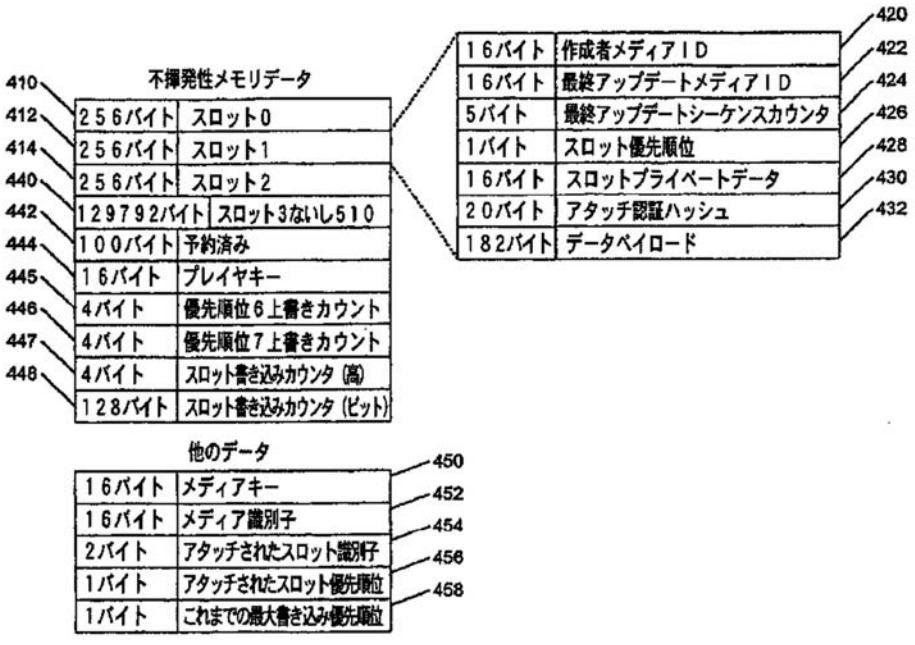
【図２】


プレイヤ境界
225
210
215
240
再生環境構成
プロセッサ
インタープリタ
実行およびデータRAM
バルク復号化モジュール
230
暗号化オラクル
220
250
205
メディアインターフェース
出力インターフェース
SIM
235
200
260
メディア
送信先プログラム／デバイス

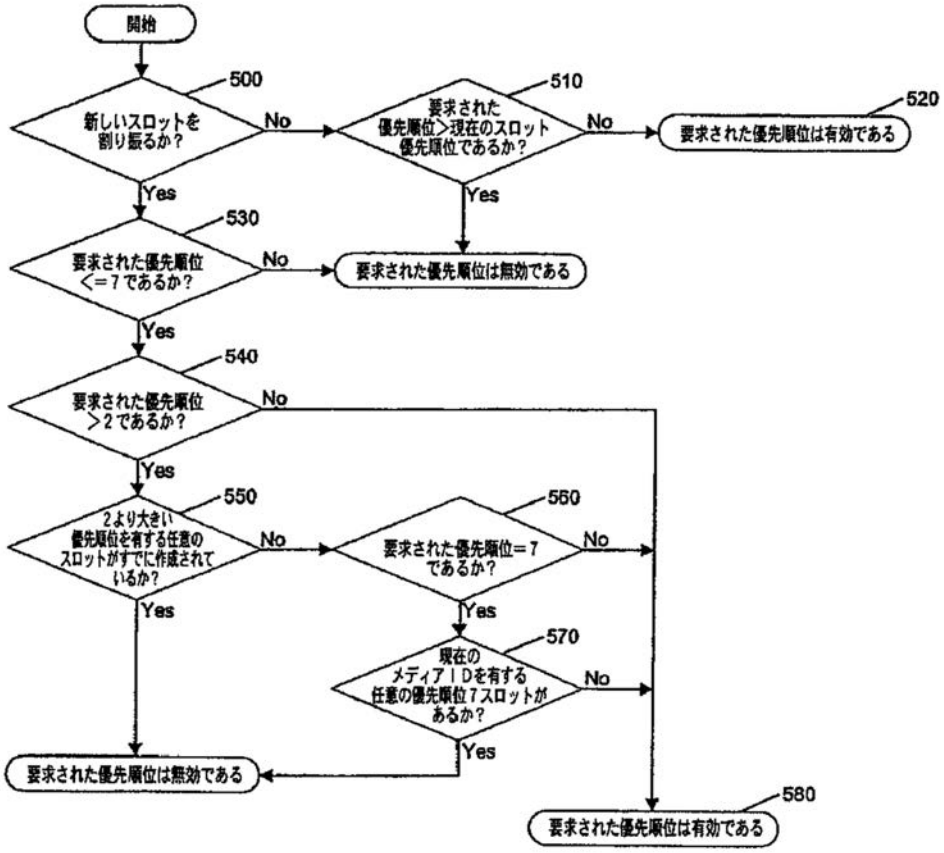
【図３】


320
メディア
インターフェース
340
ユーザ
インターフェース
再生制御
データおよびコード
暗号化されたコンテンツ
350
300
360
インタープリタ
メディアからのコード
暗号化制御
状況および信号
バルク
暗号化モジュール
ライト制御
330
310
370
平文
暗号化オラクル
380
バッファ
平文
390
レンダリング
／出力

【図４】


不揮発性メモリデータ
410
412 256バイト スロット0
414 256バイト スロット1
440 256バイト スロット2
442 129792バイト スロット3ないし510
444 100バイト 予約済み
445 16バイト プレイヤキー
446 4バイト 優先順位6上書きカウント
447 4バイト 優先順位7上書きカウント
448 4バイト スロット書き込みカウンタ（高）
128バイト スロット書き込みカウンタ（ビット）
420 16バイト 作成者メディアID
422 16バイト 最終アップデートメディアID
424 5バイト 最終アップデートシーケンスカウンタ
426 1バイト スロット優先順位
428 16バイト スロットプライベートデータ
430 20バイト アタッチ認証ハッシュ
432 182バイト データペイロード
他のデータ
450 16バイト メディアキー
452 16バイト メディア識別子
454 2バイト アタッチされたスロット識別子
456 1バイト アタッチされたスロット優先順位
458 1バイト これまでの最大書き込み優先順位

【図５】


開始
500
新しいスロットを割り振るか？
No
510
要求された優先順位＞現在のスロット優先順位であるか？
No
520
要求された優先順位は有効である
Yes
Yes
530
要求された優先順位＜＝７であるか？
No
要求された優先順位は無効である
Yes
540
要求された優先順位＞２であるか？
No
Yes
550
２より大きい優先順位を有する任意のスロットがすでに作成されているか？
No
560
要求された優先順位＝７であるか？
No
Yes
Yes
570
現在のメディアＩＤを有する任意の優先順位７スロットがあるか？
No
Yes
要求された優先順位は無効である
580
要求された優先順位は有効である

フロントページの続き

(72)発明者 ジョシュア エム. ジャフィ
アメリカ合衆国 ９４１１４ カリフォルニア州 サンフランシスコ チャーチ ストリート １０７０ アパートメント ２０９
(72)発明者 ベンジャミン シー. ジュン
アメリカ合衆国 ９４６１８ カリフォルニア州 オークランド ボイド アベニュー ５３３２
(72)発明者 カーター シー. ラーレン
アメリカ合衆国 ９４５４６ カリフォルニア州 カストロ バレー アニタ アベニュー ２０２１５
(72)発明者 ピーター ケイ. ピアスン
アメリカ合衆国 ９５００３ カリフォルニア州 アプトス マクドナルド ロード ５３０
(72)発明者 ナサニエル ジェイ. ローソン
アメリカ合衆国 ９４６１１ カリフォルニア州 オークランド プレザント バレー アベニュー １７０４ ナンバー２

審査官 和田 財太

(56)参考文献 国際公開第０２／０７９９０６（WO, Ａ１）
特開平１１－２０５３０５（ＪＰ, Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ ２１／２４
Ｇ１１Ｂ ２０／１０
Ｈ０４Ｌ ９／３２
Ｈ０４Ｎ ７／１６７