(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4622087号
(P4622087)
(45)発行日　平成23年2月2日(2011.2.2)　(24)登録日　平成22年11月12日(2010.11.12)

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

請求項の数 8 (全 27 頁)

(21)出願番号　特願2000-341431(P2000-341431)
(22)出願日　平成12年11月9日(2000.11.9)
(65)公開番号　特開2002-152187(P2002-152187A)
(43)公開日　平成14年5月24日(2002.5.24)
審査請求日　平成19年3月12日(2007.3.12)

(73)特許権者　000002185
ソニー株式会社
東京都港区港南１丁目７番１号
(74)代理人　100101801
弁理士　山田　英治
(74)代理人　100093241
弁理士　宮田　正昭
(74)代理人　100086531
弁理士　澤田　俊夫
(72)発明者　石黒　隆二
東京都品川区北品川６丁目７番３５号　ソニー株式会社内
(72)発明者　浅野　智之
東京都品川区北品川６丁目７番３５号　ソニー株式会社内

最終頁に続く

(54)【発明の名称】情報処理装置、および情報処理方法、並びにプログラム記憶媒体

(57)【特許請求の範囲】
【請求項１】
ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置であり、
前記リーフに対応付けられたエンティティが排除対象エンティティとしてのリボーク・エンティティであるか否かの検証処理を、前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティティの格納キーセットにより復号可能であるか否かの判定により実行する構成を有し、前記復号可能性の判定は、検証対象エンティティの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行する構成を有し、
前記検証対象エンティティの識別子は、該エンティティが対応付けられた前記階層ツリー構造におけるリーフの位置を上位ノードからたどるパス情報を含み、
前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、
前記復号可能性の判定は、検証対象エンティティの識別子に基づいて決定されるパスが、前記キー配置識別タグによって存在が確認され、記検証対象エンティティの対応するリーフ位置まで辿り着けるか否かの判定処理として実行することを特徴とする情報処理装置

。
【請求項２】
前記情報処理装置は、前記検証対象エンティティの識別子に基づく前記タグの追跡処理により、前記検証対象エンティティの対応するリーフ位置に辿り着けるか否かの判定、および、辿りつけない場合において、更新されていないノードキーの下位に属するか否かの判定により前記復号可能性の判定を実行する構成を有することを特徴とする請求項１に記載の情報処理装置。
【請求項３】
前記検証対象エンティティの識別子は、該エンティティの公開鍵証明書に格納された識別子であり、
前記情報処理装置は、
検証対象エンティティの識別子を該エンティティの公開鍵証明書から取得する構成を有することを特徴とする請求項１または２に記載の情報処理装置。
【請求項４】
前記情報処理装置は、
前記階層ツリー構造を構成するノードまたはリーフに対応するエンティティから提供される暗号化コンテンツの復号において、
前記エンティティの公開鍵証明書から該エンティティの識別子を取得し、該取得した識別子に基づく前記有効化キーブロック（ＥＫＢ）のタグによる追跡処理を実行して該エンティティがリボーク・エンティティであるか否かを判定するとともに、前記有効化キーブロック（ＥＫＢ）から取得されるコンテンツ暗号化キーＫｃｏｎに基づく暗号化コンテンツの復号処理を実行する構成を有することを特徴とする請求項１乃至３いずれかに記載の情報処理装置。
【請求項５】
ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置における情報処理方法であり、
前記リーフに対応付けられたエンティティが排除対象エンティティとしてのリボーク・エンティティであるか否かの検証処理を、前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティティの格納キーセットにより復号可能であるか否かの判定により実行する構成を有し、前記復号可能性の判定は、検証対象エンティティの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行し、
前記検証対象エンティティの識別子は、該エンティティが対応付けられた前記階層ツリー構造におけるリーフの位置を上位ノードからたどるパス情報を含み、
前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、
前記復号可能性の判定は、検証対象エンティティの識別子に基づいて決定されるパスが、前記キー配置識別タグによって存在が確認され、記検証対象エンティティの対応するリーフ位置まで辿り着けるか否かの判定処理として実行することを特徴とする情報処理方法。
【請求項６】
前記情報処理方法は、前記検証対象エンティティの識別子に基づく前記タグの追跡処理により、前記検証対象エンティティの対応するリーフ位置に辿り着けるか否かの判定、および、辿りつけない場合において、更新されていないノードキーの下位に属するか否かの判定により前記復号可能性の判定を実行することを特徴とする請求項５に記載の情報処理方法。

【請求項７】
前記情報処理方法は、
前記階層ツリー構造を構成するノードまたはリーフに対応するエンティテイから提供される暗号化コンテンツの復号において、
前記エンティテイの公開鍵証明書から該エンティテイの識別子を取得し、該取得した識別子に基づく前記有効化キーブロック（ＥＫＢ）のタグによる追跡処理を実行して該エンティテイがリボーク・エンティテイであるか否かを判定するとともに、前記有効化キーブロック（ＥＫＢ）から取得されるコンテンツ暗号化キーＫｃｏｎに基づく暗号化コンテンツの復号処理を実行することを特徴とする請求項５または６に記載の情報処理方法。
【請求項８】
ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置における情報処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、
前記リーフに対応付けられたエンティテイが排除対象エンティテイとしてのリボーク・エンティテイであるか否かの検証処理ステップを含み、
前記検証処理ステップは、
前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティテイの格納キーセットにより復号可能であるか否かの判定により実行するステップを有し、
前記復号可能性の判定ステップは、検証対象エンティテイの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行するステップを含み、
前記検証対象エンティテイの識別子は、該エンティテイが対応付けられた前記階層ツリー構造におけるリーフの位置を上位ノードからたどるパス情報を含み、
前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、
前記復号可能性の判定は、検証対象エンティテイの識別子に基づいて決定されるパスが、前記キー配置識別タグによって存在が確認され、記検証対象エンティテイの対応するリーフ位置まで辿り着けるか否かの判定処理として実行することを特徴とするプログラム記憶媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、情報処理装置、および情報処理方法、並びにプログラム記憶媒体に関し、特に、暗号処理を伴うシステムにおける暗号処理鍵を配信するシステムおよび方法に関する。特に、木構造の階層的鍵配信方式を用い、特定のデバイスのリボーク（排除）を効率的に実行することを可能とする情報処理装置、および情報処理方法、並びにプログラム記憶媒体に関する。なお、システムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。
【０００２】
【従来の技術】
昨今、ゲームプログラム、音声データ、画像データ等、様々なソフトウエアデータ（以下、これらをコンテンツ（Content）と呼ぶ）を、インターネット等のネットワーク、あるいはＤＶＤ、ＣＤ等の流通可能な記憶媒体を介しての流通が盛んになってきている。これらの流通コンテンツは、ユーザの所有するＰＣ（Personal Computer）、ゲーム機器によってデータ受信、あるいは記憶媒体の装着がなされて再生されたり、あるいはＰＣ等に付属する記録再生機器内の記録デバイス、例えばメモリカード、ハードディスク等に格納さ

れて、格納媒体からの新たな再生により利用される。

【０００３】

ビデオゲーム機器、ＰＣ等の情報機器には、流通コンテンツをネットワークから受信するため、あるいはＤＶＤ、ＣＤ等にアクセスするためのインタフェースを有し、さらにコンテンツの再生に必要となる制御手段、プログラム、データのメモリ領域として使用されるＲＡＭ、ＲＯＭ等を有する。

【０００４】

音楽データ、画像データ、あるいはプログラム等の様々なコンテンツは、再生機器として利用されるゲーム機器、ＰＣ等の情報機器本体からのユーザ指示、あるいは接続された入力手段を介したユーザの指示により記憶媒体から呼び出され、情報機器本体、あるいは接続されたディスプレイ、スピーカ等を通じて再生される。

【０００５】

ゲームプログラム、音楽データ、画像データ等、多くのソフトウエア・コンテンツは、一般的にその作成者、販売者に頒布権等が保有されている。従って、これらのコンテンツの配布に際しては、一定の利用制限、すなわち、正規なユーザに対してのみ、ソフトウエアの使用を許諾し、許可のない複製等が行われないようにする、すなわちセキュリティを考慮した構成をとるのが一般的となっている。

【０００６】

ユーザに対する利用制限を実現する１つの手法が、配布コンテンツの暗号化処理である。すなわち、例えばインターネット等を介して暗号化された音声データ、画像データ、ゲームプログラム等の各種コンテンツを配布するとともに、正規ユーザであると確認された者に対してのみ、配布された暗号化コンテンツを復号する手段、すなわち復号鍵を付与する構成である。

【０００７】

暗号化データは、所定の手続きによる復号化処理によって利用可能な復号データ（平文）に戻すことができる。このような情報の暗号化処理に暗号化鍵を用い、復号化処理に復号化鍵を用いるデータ暗号化、復号化方法は従来からよく知られている。

【０００８】

暗号化鍵と復号化鍵を用いるデータ暗号化・復号化方法の態様には様々な種類あるが、その１つの例としていわゆる共通鍵暗号化方式と呼ばれている方式がある。共通鍵暗号化方式は、データの暗号化処理に用いる暗号化鍵とデータの復号化に用いる復号化鍵を共通のものとして、正規のユーザにこれら暗号化処理、復号化に用いる共通鍵を付与して、鍵を持たない不正ユーザによるデータアクセスを排除するものである。この方式の代表的な方式にＤＥＳ（データ暗号標準：Deta encryption standard）がある。

【０００９】

上述の暗号化処理、復号化に用いられる暗号化鍵、復号化鍵は、例えばあるパスワード等に基づいてハッシュ関数等の一方向性関数を適用して得ることができる。一方向性関数とは、その出力から逆に入力を求めるのは非常に困難となる関数である。例えばユーザが決めたパスワードを入力として一方向性関数を適用して、その出力に基づいて暗号化鍵、復号化鍵を生成するものである。このようにして得られた暗号化鍵、復号化鍵から、逆にそのオリジナルのデータであるパスワードを求めることは実質上不可能となる。

【００１０】

また、暗号化するときに使用する暗号化鍵による処理と、復号するときに使用する復号化鍵の処理とを異なるアルゴリズムとした方式がいわゆる公開鍵暗号化方式と呼ばれる方式である。公開鍵暗号化方式は、不特定のユーザが使用可能な公開鍵を使用する方法であり、特定個人に対する暗号化文書を、その特定個人が発行した公開鍵を用いて暗号化処理を行なう。公開鍵によって暗号化された文書は、その暗号化処理に使用された公開鍵に対応する秘密鍵によってのみ復号処理が可能となる。秘密鍵は、公開鍵を発行した個人のみが所有するので、その公開鍵によって暗号化された文書は秘密鍵を持つ個人のみが復号することができる。公開鍵暗号化方式の代表的なものにはＲＳＡ（Rivest-Shamir-Adleman）

暗号がある。このような暗号化方式を利用することにより、暗号化コンテンツを正規ユーザに対してのみ復号可能とするシステムが可能となる。

【００１１】

【発明が解決しようとする課題】

上記のようなコンテンツ配信システムでは、コンテンツを暗号化してユーザにネットワーク、あるいはＤＶＤ、ＣＤ等の記録媒体に格納して提供し、暗号化コンテンツを復号するコンテンツキーを正当なユーザにのみ提供する構成が多く採用されている。コンテンツキー自体の不正なコピー等を防ぐためのコンテンツキーを暗号化して正当なユーザに提供し、正当なユーザのみが有する復号キーを用いて暗号化コンテンツキーを復号してコンテンツキーを使用可能とする構成が提案されている。

【００１２】

正当なユーザであるか否かの判定は、一般には、例えばコンテンツの送信者であるコンテンツプロバイダとユーザデバイス間、あるいはコンテンツを送受信するユーザデバイス間において、コンテンツ、あるいはコンテンツキーの配信前に認証処理を実行することによって行なう。

【００１３】

しかしながら、不正なユーザデバイス、例えば自己のデバイスの秘密鍵が露呈してしまい、その秘密鍵をデバイスに格納して正当なデバイスになりすましてコンテンツの受信を行なうなどの事態が発生することがある。このような事態に対処するために、鍵の管理センターが不正者リスト（ブラックリスト）と呼ばれる、不正デバイスのＩＤをリストアップしたリボケーションリストを、正当デバイスに配布して、リボケーションリストによって通信相手のＩＤがリストに含まれるか否かのチェックを行なうことが行なわれる。

【００１４】

リボケーションリストは、不正デバイスのＩＤをリスト化して、改竄防止のためにキー発行センタの署名が付加され、ＣＲＬ（Certificate Revocation List）と呼ばれ、新たな不正デバイスの発生にともない、順次更新され、正当なデバイスに配布される。しかしながら、不正なデバイスが増加するにつれ、リボケーションリストに記録する不正デバイスのＩＤは、単調に増加することになり、リストのサイズ（データ量）が大きくなり、リストデータの配信の負荷が大きくなり、また、配布先である正当デバイス内にリストを格納し保することも記憶スペースの負担となる。

【００１５】

本発明では、上述のようなリボケーションリストのデータの増大に伴う処理負荷、デバイスでのリスト格納における記憶スペースの問題点に鑑みてなされたものであり、不正デバイスのＩＤリストを用いることなく階層ツリー構造のキー配信構成を用いることにより、不正デバイスの検出、排除を可能とした情報処理装置、および情報処理方法、並びにプログラム記憶媒体を提供することを目的とする。

【００１６】

【課題を解決するための手段】

本発明の第１の側面は、

ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置であり、

前記ノードまたはリーフに対応するエンティティが排除対象エンティティとしてのリボーク・エンティティであるか否かの検証処理を、前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティティの格納キーセットにより復号可能であるか否かの判定により実行する構成を有し、前記復号可能性の判定は、検証対象エンティティの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行する構成を有することを特徴とする情報処理装置にある。

【００１７】

さらに、本発明の情報処理装置の一実施態様において、前記検証対象エンティティの識別子は、該エンティティの前記階層ツリー構造における対応するノードまたはリーフの位置情報を含み、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、前記追跡処理は、前記検証対象エンティティの識別子に含まれる、該エンティティの前記階層ツリー構造における位置情報に基づいて、前記タグを追跡する処理として実行する構成であることを特徴とする。

【００１８】

さらに、本発明の情報処理装置の一実施態様において、前記検証対象エンティティの識別子は、該エンティティの前記階層ツリー構造における対応するノードまたはリーフの位置情報を含み、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、前記情報処理装置は、前記検証対象エンティティの識別子に基づく前記タグの追跡処理により、前記検証対象エンティティの対応するノード位置またはリーフ位置に辿り着けるか否かの判定、および、辿りつけない場合において、更新されていないノードキーの下位に属するか否かの判定により前記復号可能性の判定を実行する構成を有することを特徴とする。

【００１９】

さらに、本発明の情報処理装置の一実施態様において、前記検証対象エンティティの識別子は、該エンティティの公開鍵証明書に格納された識別子であり、前記情報処理装置は、検証対象エンティティの識別子を該エンティティの公開鍵証明書から取得する構成を有することを特徴とする。

【００２０】

さらに、本発明の情報処理装置の一実施態様において、前記情報処理装置は、前記階層ツリー構造を構成するノードまたはリーフに対応するエンティティから提供される暗号化コンテンツの復号において、前記エンティティの公開鍵証明書から該エンティティの識別子を取得し、該取得した識別子に基づく前記有効化キーブロック（ＥＫＢ）のタグによる追跡処理を実行して該エンティティがリボーク・エンティティであるか否かを判定するとともに、前記有効化キーブロック（ＥＫＢ）から取得されるコンテンツ暗号化キーＫｃｏｎに基づく暗号化コンテンツの復号処理を実行する構成を有することを特徴とする。

【００２１】

さらに、本発明の第２の側面は、

ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置における情報処理方法であり、

前記ノードまたはリーフに対応するエンティティが排除対象エンティティとしてのリボーク・エンティティであるか否かの検証処理を、前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティティの格納キーセットにより復号可能であるか否かの判定により実行する構成を有し、前記復号可能性の判定は、検証対象エンティティの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行することを特徴とする情報処理方法にある。

【００２２】

さらに、本発明の情報処理方法の一実施態様において、前記検証対象エンティティの識別子は、該エンティティの前記階層ツリー構造における対応するノードまたはリーフの位置情報を含み、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、前記追跡処理は、前記検証対象エンティティの識別子に含まれる、該エンティティの前記階層ツリー構造における位置情報に基づいて、前記タグ

を追跡する処理として実行する構成であることを特徴とする。

【００２３】

さらに、本発明の情報処理方法の一実施態様において、前記検証対象エンティテイの識別子は、該エンティテイの前記階層ツリー構造における対応するノードまたはリーフの位置情報を含み、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグは、有効化キーブロック（ＥＫＢ）中の各々の暗号化キーデータの下位層の暗号化キーデータの有無を識別するタグとして構成されており、前記情報処理方法は、前記検証対象エンティテイの識別子に基づく前記タグの追跡処理により、前記検証対象エンティテイの対応するノード位置またはリーフ位置に辿り着けるか否かの判定、および、辿りつけない場合において、更新されていないノードキーの下位に属するか否かの判定により前記復号可能性の判定を実行することを特徴とする。

【００２５】

さらに、本発明の情報処理方法の一実施態様において、前記情報処理方法は、前記階層ツリー構造を構成するノードまたはリーフに対応するエンティテイから提供される暗号化コンテンツの復号において、前記エンティテイの公開鍵証明書から該エンティテイの識別子を取得し、該取得した識別子に基づく前記有効化キーブロック（ＥＫＢ）のタグによる追跡処理を実行して該エンティテイがリボーク・エンティテイであるか否かを判定するとともに、前記有効化キーブロック（ＥＫＢ）から取得されるコンテンツ暗号化キーＫｃｏｎに基づく暗号化コンテンツの復号処理を実行することを特徴とする。

【００２６】

さらに、本発明の第３の側面は、

ノードおよびリーフの各々に固有のキーを対応付けた階層ツリー構造の各リーフに対応付けられ、各々が前記階層ツリー構造の自己のリーフに対応するリーフキーと、上位層に至るパス上のノードキーからなるキーセットを格納した情報処理装置における情報処理をコンピュータ・システム上で実行せしめるコンピュータ・プログラムを提供するプログラム記憶媒体であって、前記コンピュータ・プログラムは、

前記ノードまたはリーフに対応するエンティテイが排除対象エンティテイとしてのリボーク・エンティテイであるか否かの検証処理ステップを含み、

前記検証処理ステップは、

前記階層ツリー構造の更新ノードキーを下位ノードキーまたはリーフキーによって暗号化した暗号化キーデータを含む有効化キーブロック（ＥＫＢ）を、検証対象エンティテイの格納キーセットにより復号可能であるか否かの判定により実行するステップを有し、

前記復号可能性の判定ステップは、検証対象エンティテイの識別子に基づく、前記有効化キーブロック（ＥＫＢ）中のキー配置識別タグの追跡処理により実行するステップを含むことを特徴とするプログラム記憶媒体にある。

【００２７】

なお、本発明の第３の側面に係るプログラム記憶媒体は、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ・プログラムをコンピュータ可読な形式で提供する媒体である。

【００２８】

このようなプログラム記憶媒体は、コンピュータ・システム上で所定のコンピュータ・プログラムの機能を実現するための、コンピュータ・プログラムと記憶媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、該記憶媒体を介してコンピュータ・プログラムをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発明の他の側面と同様の作用効果を得ることができるのである。

【００２９】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００３０】

【発明の実施の形態】
［システム概要］
図１に本発明の情報処理装置における処理の適用可能なコンテンツ配信システム例を示す。コンテンツの配信側１０は、コンテンツ受信側２０の有する様々なコンテンツ再生可能な機器に対してコンテンツ、あるいはコンテンツキーを暗号化して送信する。受信側２０における機器では、受信した暗号化コンテンツ、あるいは暗号化コンテンツキー等を復号してコンテンツあるいはコンテンツキーを取得して、画像データ、音声データの再生、あるいは各種プログラムの実行等を行なう。コンテンツの配信側１０とコンテンツ受信側２０との間のデータ交換は、インターネット等のネットワークを介して、あるいはＤＶＤ、ＣＤ等の流通可能な記憶媒体を介して実行される。
【００３１】
コンテンツの配信側１０のデータ配信手段としては、インターネット１１、衛星放送１２、電話回線１３、ＤＶＤ、ＣＤ等のメディア１４等があり、一方、コンテンツ受信側２０のデバイスとしては、パーソナルコンピュータ（ＰＣ）２１、ポータブルデバイス（ＰＤ）２２、携帯電話、ＰＤＡ（Personal Digital Assistants）等の携帯機器２３、ＤＶＤ、ＣＤプレーヤ等の記録再生器２４、ゲーム端末等の再生専用器２５等がある。これらコンテンツ受信側２０の各デバイスは、コンテンツ配信側１０から提供されたコンテンツをネットワーク等の通信手段あるいは、あるいはメディア３０から取得する。
【００３２】
［デバイス構成］
図２に、図１に示すコンテンツ受信側２０の情報処理装置の一例として、記録再生装置１００の構成ブロック図を示す。記録再生装置１００は、入出力Ｉ／Ｆ(Interface)１２０、ＭＰＥＧ(Moving Picture Experts Group)コーデック１３０、Ａ／Ｄ，Ｄ／Ａコンバータ１４１を備えた入出力Ｉ／Ｆ(Interface)１４０、暗号処理手段１５０、ＲＯＭ（Read Only Memory）１６０、ＣＰＵ(Central Processing Unit)１７０、メモリ１８０、記録媒体１９５のドライブ１９０を有し、これらはバス１１０によって相互に接続されている。
【００３３】
入出力Ｉ／Ｆ１２０は、外部から供給される画像、音声、プログラム等の各種コンテンツを構成するディジタル信号を受信し、バス１１０上に出力するとともに、バス１１０上のディジタル信号を受信し、外部に出力する。ＭＰＥＧコーデック１３０は、バス１１０を介して供給されるＭＰＥＧ符号化されたデータを、ＭＰＥＧデコードし、入出力Ｉ／Ｆ１４０に出力するとともに、入出力Ｉ／Ｆ１４０から供給されるディジタル信号をＭＰＥＧエンコードしてバス１１０上に出力する。入出力Ｉ／Ｆ１４０は、Ａ／Ｄ，Ｄ／Ａコンバータ１４１を内蔵している。入出力Ｉ／Ｆ１４０は、外部から供給されるコンテンツとしてのアナログ信号を受信し、Ａ／Ｄ，Ｄ／Ａコンバータ１４１でＡ／Ｄ(Analog Digital)変換することで、ディジタル信号として、ＭＰＥＧコーデック１３０に出力するとともに、ＭＰＥＧコーデック１３０からのディジタル信号を、Ａ／Ｄ，Ｄ／Ａコンバータ１４１でＤ／Ａ(Digital Analog)変換することで、アナログ信号として、外部に出力する。
【００３４】
暗号処理手段１５０は、例えば、１チップのＬＳＩ(Large Scale Integrated Curcuit)で構成され、バス１１０を介して供給されるコンテンツとしてのディジタル信号の暗号化、復号処理、あるいは認証処理を実行し、暗号データ、復号データ等をバス１１０上に出力する構成を持つ。なお、暗号処理手段１５０は１チップＬＳＩに限らず、各種のソフトウェアまたはハードウェアを組み合わせた構成によって実現することも可能である。ソフトウェア構成による処理手段としての構成については後段で説明する。
【００３５】
ＲＯＭ１６０は、記録再生装置によって処理されるプログラムデータを格納する。ＣＰＵ１７０は、ＲＯＭ１６０、メモリ１８０に記憶されたプログラムを実行することで、ＭＰＥＧコーデック１３０や暗号処理手段１５０等を制御する。メモリ１８０は、例えば、不揮発性メモリで、ＣＰＵ１７０が実行するプログラムや、ＣＰＵ１７０の動作上必要なデ

ータ、さらにデバイスによって実行される暗号処理に使用されるキーセットを記憶する。キーセットについては後段で説明する。ドライブ１９０は、デジタルデータを記録再生可能な記録媒体１９５を駆動することにより、記録媒体１９５からディジタルデータを読み出し（再生し）、バス１１０上に出力するとともに、バス１１０を介して供給されるディジタルデータを、記録媒体１９５に供給して記録させる。

【００３６】

記録媒体１９５は、例えば、ＤＶＤ、ＣＤ等の光ディスク、光磁気ディスク、磁気ディスク、磁気テープ、あるいはＲＡＭ等の半導体メモリ等のディジタルデータの記憶可能な媒体であり、本実施の形態では、ドライブ１９０に対して着脱可能な構成であるとする。但し、記録媒体１９５は、記録再生装置１００に内蔵する構成としてもよい。

【００３７】

なお、図２に示す暗号処理手段１５０は、１つのワンチップＬＳＩとして構成してもよく、また、ソフトウェア、ハードウェアを組み合わせた構成によって実現する構成としてもよい。

【００３８】

［キー配信構成としてのツリー（木）構造について］

次に、図１に示すコンテンツ配信側１０からコンテンツ受信側２０の各デバイスに暗号データを配信する場合における各デバイスにおける暗号処理鍵の保有構成およびデータ配信構成を図３を用いて説明する。

【００３９】

図３の最下段に示すナンバ０～１５がコンテンツ受信側２０の個々のデバイスである。すなわち図３に示す階層ツリー（木）構造の各葉(リーフ：leaf)がそれぞれのデバイスに相当する。

【００４０】

各デバイス０～１５は、製造時あるいは出荷時、あるいはその後において、図３に示す階層ツリー（木）構造における、自分のリーフからルートに至るまでのノードに割り当てられた鍵（ノードキー）および各リーフのリーフキーからなるキーセットをメモリに格納する。図３の最下段に示すＫ０００００～Ｋ１１１１が各デバイス０～１５にそれぞれ割り当てられたリーフキーであり、最上段のＫＲ（ルートキー）から、最下段から２番目の節（ノード）に記載されたキー：ＫＲ～Ｋ１１１をノードキーとする。

【００４１】

図３に示すツリー構成において、例えばデバイス０はリーフキーＫ００００と、ノードキー：Ｋ０００、Ｋ００、Ｋ０、ＫＲを所有する。デバイス５はＫ０１０１、Ｋ０１０、Ｋ０１、Ｋ０、ＫＲを所有する。デバイス１５は、Ｋ１１１１、Ｋ１１１、Ｋ１１、Ｋ１、ＫＲを所有する。なお、図３のツリーにはデバイスが０～１５の１６個のみ記載され、ツリー構造も４段構成の均衡のとれた左右対称構成として示しているが、さらに多くのデバイスがツリー中に構成され、また、ツリーの各部において異なる段数構成を持つことが可能である。

【００４２】

また、図３のツリー構造に含まれる各情報処理装置（デバイス）には、様々な記録媒体、例えば、デバイス埋め込み型あるいはデバイスに着脱自在に構成されたＤＶＤ、ＣＤ、ＭＤ、フラッシュメモリ等を使用する様々なタイプの情報処理装置が含まれている。さらに、様々なアプリケーションサービスが共存可能である。このような異なるデバイス、異なるアプリケーションの共存構成の上に図３に示すコンテンツあるいは鍵配布構成である階層ツリー構造が適用される。

【００４３】

これらの様々な情報処理装置（デバイス）、アプリケーションが共存するシステムにおいて、例えば図３の点線で囲んだ部分、すなわちデバイス０，１，２，３を同一の記録媒体を用いる１つのグループとして設定する。例えば、この点線で囲んだグループ内に含まれるデバイスに対しては、まとめて、共通のコンテンツを暗号化してプロバイダから送付し

たり、各デバイス共通に使用するコンテンツキーを送付したり、あるいは各デバイスからプロバイダあるいは決済機関等にコンテンツ料金の支払データをやはり暗号化して出力するといった処理が実行される。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、図３の点線で囲んだ部分、すなわちデバイス０，１，２，３を１つのグループとして一括してデータを送付する処理を実行する。このようなグループは、図３のツリー中に複数存在する。コンテンツプロバイダ、あるいは決済処理機関等、各デバイスとのデータ送受信を行なう機関は、メッセージデータ配信手段として機能する。

【００４４】

なお、ノードキー、リーフキーは、ある１つの鍵管理センタによって統括して管理してもよいし、各グループに対する様々なデータ送受信を行なうプロバイダ、決済機関等のメッセージデータ配信手段によってグループごとに管理する構成としてもよい。これらのノードキー、リーフキーは例えばキーの漏洩等の場合に更新処理が実行され、この更新処理は鍵管理センタ、プロバイダ、決済機関等が実行する。

【００４５】

このツリー構造において、図３から明らかなように、１つのグループに含まれる３つのデバイス０，１，２，３はノードキーとして共通のキーＫ００、Ｋ０、ＫＲを保有する。このノードキー共有構成を利用することにより、例えば共通のコンテンツキーをデバイス０，１，２，３のみに提供することが可能となる。たとえば、共通に保有するノードキーＫ００自体をコンテンツキーとして設定すれば、新たな鍵送付を実行することなくデバイス０，１，２，３のみが共通のコンテンツキーの設定が可能である。また、新たなコンテンツキーＫｃｏｎをノードキーＫ００で暗号化した値Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を、ネットワークを介してあるいは記録媒体に格納してデバイス０，１，２，３に配布すれば、デバイス０，１，２，３のみが、それぞれのデバイスにおいて保有する共有ノードキーＫ００を用いて暗号Ｅｎｃ（Ｋ００，Ｋｃｏｎ）を解いてコンテンツキー：Ｋｃｏｎを得ることが可能となる。なお、Ｅｎｃ（Ｋａ，Ｋｂ）はＫｂをＫａによって暗号化したデータであることを示す。

【００４６】

また、ある時点ｔにおいて、デバイス３の所有する鍵：Ｋ００１１,Ｋ００１,Ｋ００,Ｋ０,ＫＲが攻撃者（ハッカー）により解析されて露呈したことが発覚した場合、それ以降、システム（デバイス０，１，２，３のグループ）で送受信されるデータを守るために、デバイス３をシステムから切り離す必要がある。そのためには、ノードキー：Ｋ００１，Ｋ００,Ｋ０,ＫＲをそれぞれ新たな鍵Ｋ（ｔ）００１,Ｋ（ｔ）００,Ｋ（ｔ）０,Ｋ（ｔ）Ｒに更新し、デバイス０，１，２にその更新キーを伝える必要がある。ここで、Ｋ（ｔ）ａａａは、鍵Ｋａａａの世代（Generation）：ｔの更新キーであることを示す。

【００４７】

更新キーの配布処理ついて説明する。キーの更新は、例えば、図４（Ａ）に示す有効化キーブロック（ＥＫＢ：Enabling Key Block）と呼ばれるブロックデータによって構成されるテーブルをたとえばネットワーク、あるいは記録媒体に格納してデバイス０，１，２に供給することによって実行される。なお、有効化キーブロック（ＥＫＢ）は、図３に示すようなツリー構造を構成する各リーフに対応するデバイスに新たに更新されたキーを配布するための暗号化キーによって構成される。有効化キーブロック（ＥＫＢ）は、キー更新ブロック（ＫＲＢ：Key Renewal Block）と呼ばれることもある。

【００４８】

図４（Ａ）に示す有効化キーブロック（ＥＫＢ）には、ノードキーの更新の必要なデバイスのみが更新可能なデータ構成を持つブロックデータとして構成される。図４の例は、図３に示すツリー構造中のデバイス０，１，２において、世代ｔの更新ノードキーを配布することを目的として形成されたブロックデータである。図３から明らかなように、デバイス０，デバイス１は、更新ノードキーとしてＫ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒが必要であり、デバイス２は、更新ノードキーとしてＫ（ｔ）００１、Ｋ（ｔ）００、Ｋ（ｔ

）０、Ｋ（ｔ）Ｒが必要である。

【００４９】

図４（Ａ）のＥＫＢに示されるようにＥＫＢには複数の暗号化キーが含まれる。最下段の暗号化キーは、Ｅｎｃ（Ｋ００１０，Ｋ（ｔ）００１）である。これはデバイス２の持つリーフキーＫ００１０によって暗号化された更新ノードキーＫ（ｔ）００１であり、デバイス２は、自身の持つリーフキーによってこの暗号化キーを復号し、Ｋ（ｔ）００１を得ることができる。また、復号により得たＫ（ｔ）００１を用いて、図４（Ａ）の下から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００１，Ｋ（ｔ）００）を復号可能となり、更新ノードキーＫ（ｔ）００を得ることができる。以下順次、図４（Ａ）の上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号し、更新ノードキーＫ（ｔ）０、図４（Ａ）の上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号しＫ（ｔ）Ｒを得る。一方、デバイスＫ００００．Ｋ０００１は、ノードキーＫ０００は更新する対象に含まれておらず、更新ノードキーとして必要なのは、Ｋ（ｔ）００、Ｋ（ｔ）０、Ｋ（ｔ）Ｒである。デバイスＫ００００．Ｋ０００１は、図４（Ａ）の上から３段目の暗号化キーＥｎｃ（Ｋ０００，Ｋ（ｔ）００）を復号しＫ（ｔ）００、を取得し、以下、図４（Ａ）の上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号し、更新ノードキーＫ（ｔ）０、図４（Ａ）の上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号しＫ（ｔ）Ｒを得る。このようにして、デバイス０，１，２は更新した鍵Ｋ（ｔ）Ｒを得ることができる。なお、図４（Ａ）のインデックスは、復号キーとして使用するノードキー、リーフキーの絶対番地を示す。

【００５０】

図３に示すツリー構造の上位段のノードキー：Ｋ（ｔ）０,Ｋ（ｔ）Ｒの更新が不要であり、ノードキーＫ００のみの更新処理が必要である場合には、図４（Ｂ）の有効化キーブロック（ＥＫＢ）を用いることで、更新ノードキーＫ（ｔ）００をデバイス０，１，２に配布することができる。

【００５１】

図４（Ｂ）に示すＥＫＢは、例えば特定のグループにおいて共有する新たなコンテンツキーを配布する場合に利用可能である。具体例として、図３に点線で示すグループ内のデバイス０，１，２，３がある記録媒体を用いており、新たな共通のコンテンツキーＫ（ｔ）ｃｏｎが必要であるとする。このとき、デバイス０，１，２，３の共通のノードキーＫ００を更新したＫ（ｔ）００を用いて新たな共通の更新コンテンツキー：Ｋ（ｔ）ｃｏｎを暗号化したデータＥｎｃ（Ｋ（ｔ），Ｋ（ｔ）ｃｏｎ）を図４（Ｂ）に示すＥＫＢとともに配布する。この配布により、デバイス４など、その他のグループの機器においては復号されないデータとしての配布が可能となる。

【００５２】

すなわち、デバイス０，１，２はＥＫＢを処理して得たＫ（ｔ）００を用いて上記暗号文を復号すれば、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得ることが可能になる。

【００５３】

［ＥＫＢを使用したコンテンツキーの配布］

図５に、ｔ時点でのコンテンツキーＫ（ｔ）ｃｏｎを得る処理例として、Ｋ（ｔ）００を用いて新たな共通のコンテンツキーＫ（ｔ）ｃｏｎを暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）ｃｏｎ）と図４（Ｂ）に示すＥＫＢとを記録媒体を介して受領したデバイス０の処理を示す。すなわちＥＫＢによる暗号化メッセージデータをコンテンツキーＫ（ｔ）ｃｏｎとした例である。

【００５４】

図５に示すように、デバイス０は、記録媒体に格納されている世代：t時点のＥＫＢと自分があらかじめ格納しているノードキーＫ０００を用いて上述したと同様のＥＫＢ処理により、ノードキーＫ（ｔ）００を生成する。さらに、復号した更新ノードキーＫ（ｔ）００を用いて更新コンテンツキーＫ（ｔ）ｃｏｎを復号して、後にそれを使用するために自分だけが持つリーフキーＫ００００で暗号化して格納する。

【００５５】
［ＥＫＢのフォーマット］
図６に有効化キーブロック（ＥＫＢ）のフォーマット例を示す。バージョン６０１は、有効化キーブロック（ＥＫＢ）のバージョンを示す識別子である。なお、バージョンは最新のＥＫＢを識別する機能とコンテンツとの対応関係を示す機能を持つ。デプスは、有効化キーブロック（ＥＫＢ）の配布先のデバイスに対する階層ツリーの階層数を示す。データポインタ６０３は、有効化キーブロック（ＥＫＢ）中のデータ部の位置を示すポインタであり、タグポインタ６０４はタグ部の位置、署名ポインタ６０５は署名の位置を示すポインタである。
【００５６】
データ部６０６は、例えば更新するノードキーを暗号化したデータを格納する。例えば図５に示すような更新されたノードキーに関する各暗号化キー等を格納する。
【００５７】
タグ部６０７は、データ部に格納された暗号化されたノードキー、リーフキーの位置関係を示すタグである。このタグの付与ルールを図７を用いて説明する。図７では、データとして先に図４（Ａ）で説明した有効化キーブロック（ＥＫＢ）を送付する例を示している。この時のデータは、図７の表（ｂ）に示すようになる。このときの暗号化キーに含まれるトップノードのアドレスをトップノードアドレスとする。この場合は、ルートキーの更新キーＫ（ｔ）Ｒが含まれているので、トップノードアドレスはＫＲとなる。このとき、例えば最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）は、図７の（ａ）に示す階層ツリーに示す位置にある。ここで、次のデータは、Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）であり、ツリー上では前のデータの左下の位置にある。データがある場合は、タグが０、ない場合は１が設定される。タグは｛左（Ｌ）タグ，右（Ｒ）タグ｝として設定される。最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）の左にはデータがあるので、Ｌタグ＝０、右にはデータがないので、Ｒタグ＝１となる。以下、すべてのデータにタグが設定され、図７（ｃ）に示すデータ列、およびタグ列が構成される。
【００５８】
タグは、データＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）がツリー構造のどこに位置しているのかを示すために設定されるキー配置識別タグである。データ部に格納されるキーデータＥｎｃ（Ｋｘｘｘ，Ｋｙｙｙ）．．．は、単純に暗号化されたキーの羅列データに過ぎないので、上述したタグによってデータとして格納された暗号化キーのツリー上の位置を判別可能としたものである。上述したタグを用いずに、先の図４で説明した構成のように暗号化データに対応させたノード・インデックスを用いて、例えば、
０：Ｅｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）ｒｏｏｔ）
００：Ｅｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）
０００：Ｅｎｃ（Ｋ（（ｔ）０００，Ｋ（Ｔ）００）
．．．のようなデータ構成とすることも可能であるが、このようなインデックスを用いた構成とすると冗長なデータとなりデータ量が増大し、ネットワークを介する配信等においては好ましくない。これに対し、上述したタグをキー位置を示す索引データとして用いることにより、少ないデータ量でキー位置の判別が可能となる。
【００５９】
図６に戻って、ＥＫＢフォーマットについてさらに説明する。署名（Signature）は、有効化キーブロック（ＥＫＢ）を発行したＥＫＢ発行局、例えば鍵管理センタ、コンテンツロバイダ、決済機関等が実行する電子署名である。ＥＫＢを受領したデバイスは署名検証によって正当な有効化キーブロック（ＥＫＢ）発行者が発行した有効化キーブロック（ＥＫＢ）であることを確認する。
【００６０】
［ＥＫＢを使用したコンテンツキーおよびコンテンツの配信］
上述の例では、コンテンツキーのみをＥＫＢとともに送付する例について説明したが、コンテンツキーで暗号化したコンテンツと、ルートキー、ノードキーなどの暗号キーで暗号

化したコンテンツキーと、ＥＫＢによって暗号化したコンテンツキー暗号鍵を併せて送付する構成について以下説明する。

【００６１】

図８にこのデータ構成を示す。図８（ａ）に示す構成において、Ｅｎｃ（Ｋｃｏｎ，ｃｏｎｔｅｎｔ）８０１は、コンテンツ（Content）をコンテンツキー（Ｋｃｏｎ）で暗号化したデータであり、Ｅｎｃ（Ｋｒｏｏｔ，Ｋｃｏｎ）８０２は、コンテンツキー（Ｋｃｏｎ）をルートキー（Ｋｒｏｏｔ）で暗号化したデータであり、Ｅｎｃ（ＥＫＢ，Ｋｒｏｏｔ）８０３は、ルートキーＫｒｏｏｔを有効化キーブロック（ＥＫＢ）によって暗号化したデータであることを示す。

【００６２】

ここで、ルートキーＫｒｏｏｔは、図３で示すノードキー（Ｋ０００，Ｋ００…）であってもよい。

【００６３】

図８（ｂ）は、複数のコンテンツがメディアに記録され、それぞれが同じＥｎｃ（ＥＫＢ，Ｋｒｏｏｔ）８０５を利用している場合の構成例を示す、このような構成においては、各データに同じＥｎｃ（ＥＫＢ，Ｋｒｏｏｔ）を付加することなく、Ｅｎｃ（ＥＫＢ，Ｋｒｏｏｔ）にリンクするリンク先を示すデータを各データに付加する構成とすることができる。

【００６４】

図９に、図３に示すノードキーＫ００を更新した更新ノードキーＫ（ｔ）００を使用してコンテンツキーＫｃｏｎを暗号化した場合の処理例を示す。この場合、図３の点線枠で囲んだグループにおいてデバイス３が、例えば鍵の漏洩によりリボーク（排除）されているとして、他のグループのメンバ、すなわち、デバイス０，１，２に対して図９に示す（ａ）有効化キーブロック（ＥＫＢ）と、（ｂ）コンテンツキー（Ｋｃｏｎ）を更新ノードキーＫ（ｔ）００で暗号化したデータと、（ｃ）コンテンツ（content）をコンテンツキー（Ｋｃｏｎ）で暗号化したデータとを配信することにより、デバイス０，１，２はコンテンツを得ることができる。

【００６５】

図９の右側には、デバイス０における復号手順を示してある。デバイス０は、まず、受領した有効化キーブロックから自身の保有するリーフキーＫ０００を用いた復号処理により、Ｋ（ｔ）００を取得する。次に、Ｋ（ｔ）００による復号によりコンテンツキーＫｃｏｎを取得し、さらにコンテンツキーＫｃｏｎによりコンテンツの復号を行なう。これらの処理により、デバイス０はコンテンツを利用可能となる。デバイス１，２においても各々異なる処理手順でＥＫＢを処理することにより、コンテンツキーの暗号化キーを取得することが可能となり、同様にコンテンツを利用することが可能となる。

【００６６】

図３に示す他のグループのデバイス４，５，６…は、この同様のデータ（ＥＫＢ）を受信したとしても、自身の保有するリーフキー、ノードキーを用いてＫ（ｔ）００を取得することができない。同様にリボークされたデバイス３においても、自身の保有するリーフキー、ノードキーでは、Ｋ（ｔ）００を取得することができず、正当な権利を有するデバイスのみがコンテンツを復号して利用することが可能となる。

【００６７】

このように、ＥＫＢを利用したコンテンツキーの配送を用いれば、データ量を少なくして、かつ安全に正当権利者のみが復号可能とした暗号化コンテンツを配信することが可能となる。

【００６８】

なお、有効化キーブロック（ＥＫＢ）、コンテンツキー、暗号化コンテンツ等は、ネットワークを介して安全に配信することが可能な構成であるが、有効化キーブロック（ＥＫＢ）、コンテンツキー、暗号化コンテンツをＤＶＤ、ＣＤ等の記録媒体に格納してユーザに提供することも可能である。この場合、記録媒体に格納された暗号化コンテンツの復号に

は、同一の記録媒体に格納された有効化キーブロック（ＥＫＢ）の復号により得られるコンテンツキーを使用するように構成すれば、予め正当権利者のみが保有するリーフキー、ノードキーによってのみ利用可能な暗号化コンテンツの配布処理、すなわち利用可能なユーザデバイスを限定したコンテンツ配布が簡易な構成で実現可能となる。

【００６９】

図１０に記録媒体に暗号化コンテンツとともに有効化キーブロック（ＥＫＢ）を格納した構成例を示す。図１０に示す例においては、記録媒体にコンテンツＣ１～Ｃ４が格納され、さらに各格納コンテンツに対応するの有効化キーブロック（ＥＫＢ）を対応付けたデータが格納され、さらにバージョンＭの有効化キーブロック（ＥＫＢ＿Ｍ）が格納されている。例えばＥＫＢ＿１はコンテンツＣ１を暗号化したコンテンツキーＫｃｏｎ１を生成するのに使用され、例えばＥＫＢ＿２はコンテンツＣ２を暗号化したコンテンツキーＫｃｏｎ２を生成するのに使用される。この例では、バージョンＭの有効化キーブロック（ＥＫＢ＿Ｍ）が記録媒体に格納されており、コンテンツＣ３，Ｃ４は有効化キーブロック（ＥＫＢ＿Ｍ）に対応付けられているので、有効化キーブロック（ＥＫＢ＿Ｍ）の復号によりコンテンツＣ３，Ｃ４のコンテンツキーを取得することができる。ＥＫＢ＿１、ＥＫＢ＿２はディスクに格納されていないので、新たな提供手段、例えばネットワーク配信、あるいは記録媒体による配信によってそれぞれのコンテンツキーを復号するために必要なＥＫＢ＿１，ＥＫＢ＿２を取得することが必要となる。

【００７０】

［認証処理における有効化キーブロック（ＥＫＢ）を使用したリボークエンティテイ（ｅｘ．不正デバイス）の判定］

次に、有効化キーブロック（ＥＫＢ）を使用したリボークエンティテイ（ｅｘ．不正デバイス）の検出処理について説明する。まず、公開鍵暗号方式を用いた相互認証方法を、図１１を用いて説明する。図１１において、Ａは自己の秘密鍵［Ａｐｒｉ－Ｋｅｙ］、公開鍵［Ａｐｕｂ－Ｋｅｙ］、認証局の署名のなされた公開鍵証明書［Ａｃｅｒｔ］を有し、さらに、公開鍵証明書の署名主体である認証局の公開鍵と、ＥＫＢの署名主体であるＥＫＢ発行局の公開鍵を有し、Ｂは、自己の秘密鍵［Ｂｐｒｉ－Ｋｅｙ］、公開鍵［Ｂｐｕｂ－Ｋｅｙ］、認証局の署名のなされた公開鍵証明書［Ｂｃｅｒｔ］、認証局の公開鍵、ＥＫＢ発行局の公開鍵を有する。

【００７１】

Ａ，Ｂ各々の有する公開鍵証明書の構成について図１２を用いて説明する。公開鍵証明書は、公開鍵暗号方式における認証局（ＣＡ：Certificate AuthorityまたはＩＡ：Issuer Authority）が発行する証明書であり、ユーザが自己のＩＤ、公開鍵等を認証局に提出することにより、認証局側が認証局のＩＤや有効期限等の情報を付加し、さらに認証局による署名を付加して作成される証明書である。

【００７２】

図１２に示す公開鍵証明書は、証明書のバージョン番号、認証局が証明書利用者に対し割り付ける証明書の通し番号、電子署名に用いたアルゴリズムおよびパラメータ、認証局の名前、証明書の有効期限、証明書利用者ＩＤ、証明書利用者の公開鍵並びに認証局の電子署名を含む。

【００７３】

電子署名は、証明書のバージョン番号、認証局が証明書利用者に対し割り付ける証明書の通し番号、電子署名に用いたアルゴリズムおよびパラメータ、認証局の名前、証明書の有効期限、証明書利用者の名前並びに証明書利用者の公開鍵全体に対しハッシュ関数を適用してハッシュ値を生成し、そのハッシュ値に対して認証局の秘密鍵を用いて生成したデータである。

【００７４】

公開鍵証明書の証明書利用者ＩＤは、前述のキー配信ツリー構成のノード、リーフ位置を示す識別値としてのリーフＩＤが含まれる。図３のツリー構成であれば、デバイス０は［ＩＤ＝００００］、デバイス１は［ＩＤ＝０００１］、デバイス１５は、［ＩＤ＝１１１

１］などである。このようなＩＤに基づいてそのデバイスなどのエンティテイが、ツリー構成のどの位置（リーフまたはノード）にあるエンティテイ（ｅｘ．デバイス）であるかが識別可能となる。

【００７５】

図１１の相互認証処理は、上述の公開鍵証明書を用いて行われる。まずＢが、Ｂの公開鍵証明書Ｂｃｅｒｔと、乱数Ｒｂを生成し、Ａに送信する。これを受信したＡは、認証局の公開鍵でＢの公開鍵証明書（Ｂ．Ｃｅｒｔ）を検証する。検証がＮＧであれば、公開鍵証明書は無効なものであると判定されるので、この時点で認証処理を中止し、認証不成立となる。Ｂの公開鍵証明書（Ｂ．Ｃｅｒｔ）の検証がＯＫならば、次に、Ｂの公開鍵証明書（Ｂ．Ｃｅｒｔ）内のＢのリーフＩＤで自デバイスに保持したＥＫＢを辿る。

【００７６】

先に説明した図７に関する説明から理解されるようにＥＫＢ内に格納されたタグは自ノードの左及び右のノードの鍵データの有無を０，１で示している。すなわちデータがある場合は０、データがない場合を１として設定される。リーフＩＤに基づくＥＫＢの追跡処理、すなわち辿り方は、このような条件設定に基づくタグを用いて行われる。

【００７７】

リーフＩＤに基づくＥＫＢの追跡（辿り方）について、図１３を用いて説明する。図１３（ａ）に示すようにリーフキーＫ１００１を持つデバイスをリボークデバイス［１００１］とする。このとき、ＥＫＢは、図１３（ｂ）のような暗号化キーとタグの構成を持つ。図１３（ｂ）のＥＫＢは、図１３（ａ）の１つのデバイス［１００１］をリボークするために、ＫＲ，Ｋ１，Ｋ１０，Ｋ１００を更新したＥＫＢとなる。

【００７８】

このＥＫＢを処理することにより、リボークデバイス［１００１］以外のリーフはすべて更新されたルートキーＫ（ｔ）Ｒを取得できる。すなわち、ノードキーＫ０の下位につらなるリーフは、更新されていないノードキーＫ０をデバイス内に保持しているので、Ｅｎｃ（Ｋ０，Ｋ（ｔ）Ｒ）をＫ０によって復号することで更新ルートキーＫ（ｔ）Ｒを取得可能となる。また、Ｋ１１以下のリーフは更新されていないＫ１１を用いて、Ｅｎｃ（Ｋ１１，Ｋ（ｔ）１）をＫ１１によって復号することで更新ノードキーＫ（ｔ）１を取得して、さらに、Ｅｎｃ（Ｋ（ｔ）１，Ｋ（ｔ）Ｒ）をＫ（ｔ）１によって復号することで更新ルートキーを取得できる。Ｋ１０１の下位リーフについても復号ステップが１つ増加するのみで、同様に更新ルートキーを取得できる。

【００７９】

また、リボークされていないリーフキーＫ１０００を持つデバイス［１０００］は、自己のリーフキーで、Ｅｎｃ（Ｋ１０００，Ｋ（ｔ）１００）を復号して、Ｋ（ｔ）１００を取得後、上位のノードキーを順次復号して更新ルートキーを取得できる。

【００８０】

リボークされたデバイス［１００１］のみが、自己のリーフの一段上の更新ノードキーＫ（ｔ）１００をＥＫＢ処理により取得できないので、結局、更新ルートキーＫ（ｔ）Ｒを取得することができない。

【００８１】

リボークされていない正当なデバイスには、図１３（ｂ）に示すデータ部と、タグを有するＥＫＢがＥＫＢ発行局から配信され、デバイス内に格納されている。

【００８２】

相互認証において、図１３の（ａ）に示すリボークデバイス［ＩＤ＝１００１］と、例えばあるコンテンツプロバイダ間で図１１に示す公開鍵方式の相互認証を行なっているとすると、コンテンツプロバイダは、図１３（ａ）のリボークデバイス［ＩＤ＝１００１］から公開鍵証明書を受信し、公開鍵証明書の検証の後、公開鍵証明書からＩＤを取得する。このＩＤは［１００１］であり、ＥＫＢ配信ツリー構成のリーフ位置を示している。

【００８３】

ＩＤ［１００１］を受信したコンテンツプロバイダは、ＩＤ＝１００１のリーフに対応す

るデバイスが、ＥＫＢにおいて有効なリーフデバイスとして設定されているかを検証する。この検証は、すなわち、リーフ［１００１］が更新されたルートキーＫ（ｔ）Ｒを取得できるか否かを判定する処理として実行される。

【００８４】

例えば、非更新ノードキー（ｅｘ．図１３（ａ）のＫ０，Ｋ１１など）の下位に属するリーフであれば、リボークされていないことが明らかであり、正当デバイスであると判定可能であり、更新ノードキーの下位に属するリーフである場合は、その更新ノードキーを取得可能な暗号化データがＥＫＢに格納されているか否かによって、そのエンティティがリボークされているか否かを判定可能となる。

【００８５】

判定処理の一例として、ＥＫＢに格納されたタグに基づいてＥＫＢ追跡処理を行なう例を説明する。ＥＫＢ追跡処理は、上位のルートキーからキー配信ツリーを辿れるか否かを判定する処理である。例えば図１３のリーフ［１００１］のＩＤである［１００１］を［１］、［０］、［０］、［１］の４ビットとして、最上位ビットから順次下位ビットに従ってツリーを辿る。ビットが１であれば右側、０であれば左に進む。

【００８６】

図１３（ａ）のルートから、ＩＤ［１００１］の最上位ビットは１であり、右側に進む。ＥＫＢ内の最初のタグは、０：｛０，０｝であり、両枝にデータを有することが判定され、右側に進みＫ１に辿り着ける。次にＫ１の下位のノードに進む。ＩＤ［１００１］の２番目のビットは０であり、左側に進む。Ｋ１の下位のデータ有無を示すタグは、図１３（ａ），（ｂ）の２：｛０，０｝であり、両枝にデータを有すると判定され、左側に進みＫ１０に辿り着ける。さらに、ＩＤ［１００１］の３番目のビットは０であり、左側に進む。Ｋ１０の下位のデータ有無を示すタグは、図１３（ａ），（ｂ）の３：｛０，０｝であり、両枝にデータを有すると判定され、左側に進みＫ１００に辿り着ける。さらに、ＩＤ［１００１］の最下位ビットは１であり、右側に進む。Ｋ１００の下位のデータ有無を示すタグは、図１３（ａ），（ｂ）の５：｛０，１｝であり、右側にはデータを持たない。従ってノード［１００１］には辿りつけないことが判定され、ＩＤ［１００１］のデバイスはＥＫＢによる更新ルートキーを取得できないデバイス、すなわちリボークデバイスであると判定される。

【００８７】

例えば図１３（ａ）のリーフキーＫ１０００を有するデバイスＩＤは［１０００］であり、上述と同様のＥＫＢ内のタグに基づくＥＫＢ追跡処理、すなわちツリーを辿る処理を実行すると、ノード［１０００］に辿りつくことができるので、ＥＫＢによる更新ルートキーを取得可能なリボークされていない正当なデバイスであると判定される。

【００８８】

また、例えば更新されていないノードキー、例えばＫ０，Ｋ１１などの下位のリーフにも、リーフ自体には、辿り着けないが、この場合は、更新されていない末端ノードに辿りつくことが可能である。更新されていないノードの下位のリーフは、更新されていないノードキーを用いてＥＫＢの処理が可能であり、更新ルートキーを取得できるので正当なデバイスである。更新されていないノードキーであるか否かは、そのノードに対応するタグにより判定することが可能となる。更新されていないノードキーＫ０，Ｋ１１，Ｋ１０１に対応するタグは１：｛１，１｝、４：｛１，１｝、６｛１，１｝となり、これらはさらに下位ノードまたはリーフが存在するが、ＥＫＢ内には暗号化鍵データを持たないことを示しており、これらの下位のリーフのデバイスはリボークされていない有効な正当デバイスであると判定される。

【００８９】

図１３に示す例は、１つのデバイスについてのみのリボーク態様であるが、図１４に示すようにあるノードの下にあるすべてのリーフデバイスを一括してリボークすることも可能である。この場合のＥＫＢのデータ（暗号化キー）、タグは図１４（ｂ）のようになる。

【００９０】

例えば、コンテンツプロバイダがリボークされたＫ１０００に対応するリーフデバイスから公開鍵証明書を受信してＩＤ［１０００］を取得したとすると、このＩＤ［１０００］に基づいてＥＫＢのタグに基づいてツリーを辿る処理を実行する。

【００９１】

図１４（ａ）のルートから、ＩＤ［１０００］の最上位ビットは１であり、右側に進む。ＥＫＢ内の最初のタグ０：｛０，０｝であり、両枝にデータを有することが判定され、右側に進みＫ１に辿り着ける。次にＫ１の下位のノードに進む。ＩＤ［１０００］の２番目のビットは０であり、左側に進む。Ｋ１の下位のデータ有無を示すタグは、図１３（ａ），（ｂ）の２：｛１，０｝であり、左側にはデータを持たない。従ってノード［１０００］には辿りつけない。このときの末端ノードＫ１に対応するタグは｛１，０｝であり、下位のデータ持たない｛１，１｝ではない。

【００９２】

タグ｛１，０｝は、Ｋ１の右側の下位のノードまたはリーフにおいてのみ復号可能な更新されたＫ１（ｔ）を取得するための暗号化鍵データがＥＫＢに格納されていることを示している。

【００９３】

このように、リーフＩＤに基づいて辿り着く最終地点がノードであり、その最終ノードの対応タグが｛１，１｝以外の値を持っている場合は、さらに下位の暗号化鍵データをＥＫＢ内に有することを示している。この場合は、そのＩＤを持つリーフデバイスはＥＫＢの処理によって更新されたルートキーを取得することができないので、リボークされたデバイスであると判定される。

【００９４】

このようにして、認証処理において通信相手から取得した公開鍵証明書に格納されたリーフＩＤに基づいて通信相手がリボークされているか否かを判定することが可能となる。

【００９５】

図１１に戻って認証処理シーケンスについての説明を続ける。Ａは、Ｂから受信した公開鍵証明書から取り出したＢのリーフＩＤに基づいて上述のようなＥＫＢのタグに基づくツリーを辿る処理を実行し、ＩＤの示すリーフ位置がＥＫＢ処理により更新ルートキーを取得可能な位置であるかを判定し、ＥＫＢ処理が可能である位置である場合は、リボークされていない正統デバイスであると判定する。ＥＫＢ処理が不可能なリーフ位置である場合は、リボークされた不正なデバイスであると判定し、認証不成立として処理を中止する。

【００９６】

ＩＤに基づくＥＫＢ処理が可能であるデバイスであると判定された場合は、Ｂから受信した乱数ＲｂにＡの秘密鍵で署名をしてＳｉｇ_Ａ（Ｒｂ）を生成し、さらに乱数Ｒａを生成する。ＡはこれらのＳｉｇ_Ａ（Ｒｂ）、Ｒａに、自己のデバイス内に格納したＥＫＢと公開鍵証明書Ａ．ＣｅｒｔをＢに送信する。

【００９７】

Ｂは、認証局の公開鍵でＡの公開鍵証明書（Ａ．Ｃｅｒｔ）を検証し、検証ＯＫならば、ＥＫＢ配信機関の公開鍵て受信ＥＫＢの検証を行なう。ＥＫＢは前述したように、改竄防止のため、ＥＫＢ配信機関の秘密鍵で署名がなされており、ＢはＥＫＢの公開鍵を用いて検証処理を行なう。検証ＯＫならば、Ａの公開鍵証明書（Ａ．Ｃｅｒｔ）内のＡのリーフＩＤを取得し、前述した図１３，１４を用いた説明と同様リーフＩＤに基づいてＥＫＢを辿る。

【００９８】

辿れなかった場合は、Ａはリボークされたデバイスであると判定され、認証不成立として、その後の処理は中止する。なお、Ａは、デバイスに限らずコンテンツプロバイダ、サービスプロバイダであってもよく、図１３、図１４に示すツリー構成の最下段のリーフではない途中ノードのキーを有するノードであってもよい。例えば、図１３、図１４に示すＫ１０のノードキー位置に対応するノードである場合、そのコンテンツプロバイダまたはサービスプロバイダのＩＤは［１０］となり、ＩＤ［１０］に基づいてＥＫＢのタグを利用

したＥＫＢを辿る処理を実行してリボークされているか否かの判定を行なう。
【００９９】
ＥＫＢを辿る処理によって辿れた場合は、Ａから受信したデータＳｉｇ_Ａ（Ｒｂ）を、Ａの公開鍵証明書（Ａ．Ｃｅｒｔ）内の公開鍵Ａ．Ｐｕｂ-Ｋｅｙで検証する。検証がＯＫならば、ＲａにＢ.pri-.Key（Ｂの秘密鍵）で署名を行なって、Ｓｉｇ_Ｂ（Ｒａ）生成し、生成したＳｉｇ_Ｂ（Ｒａ）をＡに送信する。
【０１００】
Ｓｉｇ_Ｂ（Ｒａ）を受信したＡは、Ｂの公開鍵証明書（Ｂ．Ｃｅｒｔ）から取得したＢの公開鍵を用いてＳｉｇ_Ｂ（Ｒａ）を検証する。検証がＯＫであれば、認証が成立したと判定する。
【０１０１】
図１５にＥＫＢを利用したリボークデバイス判定処理についての処理フローを示す。フローの各ステップについて説明する。ステップＳ１０１において、通信相手（認証相手）の公開鍵証明書からＩＤを取得する。ステップＳ１０２において、取得したＩＤを用いむＥＫＢのタグに基づいて、ＩＤの示すリーフまたはリードを目標とする追跡処理を実行する。
【０１０２】
追跡処理は、前述の図１３,図１４を用いて説明した手順で実行する。追跡処理の結果、ＩＤの示すリーフまたはノードに辿り着くことができたか、辿りつけない場合であってもＩＤの示すリーフまたはノードにおいてＥＫＢ処理が可能であるか否か、すなわち更新ルートキーの取得が可能か否かを判定する（Ｓ１０３）。
【０１０３】
ＥＫＢ処理が可能である位置にあるＩＤであると判定されれば、ステップＳ１０４に進み、ＩＤに対応するデバイスはリボークされていない正当なデバイスであると判定する。一方、ＥＫＢ処理が不可能な位置にあるＩＤであると判定されれば、ステップＳ１０５に進み、ＩＤに対応するデバイスはリボークされている不正なデバイスであると判定する。
【０１０４】
［有効化キーブロック（ＥＫＢ）を使用したリボークデバイス（不正デバイス）の判定処理を伴うコンテンツ利用処理］
次に、有効化キーブロック（ＥＫＢ）を使用したリボークデバイス（不正デバイス）の判定処理を伴うコンテンツ利用処理例について説明する。図１６に示す例は、プロバイダＡがデバイス（ＩＤ＝００ｘｘ）にコンテンツを暗号化して配信する例である。
【０１０５】
コンテンツプロバイダＡは、デバイス［００ｘｘ］に対して、Ａの公開鍵証明書［Ａ．Ｃｅｒｔ］、コンテンツキーを自己の秘密鍵で署名したデータ［Ｓｉｇ＿Ａ（Ｋｃｏｎ）］、有効化キーブロック［ＥＫＢ］、コンテンツキーを更新ルートキーで暗号化したデータ［Ｅｎｃ（Ｋ（ｔ）ｒｏｏｔ，Ｋｃｏｎ）、さらにコンテンツをコンテンツキーで暗号化したデータ［Ｅｎｃ（Ｋｃｏｎ，Ｃｏｎｔｅｎｔ）を送信する。
【０１０６】
これらのデータを受信したデバイス［００ｘｘ］は、まず、受信したＡの公開鍵証明書［Ａ．Ｃｅｒｔ］を認証局の公開鍵で検証する。検証がＯＫであれば、Ａの公開鍵証明書［Ａ．Ｃｅｒｔ］からＡの公開鍵とＡのＩＤを取得する。
【０１０７】
次にコンテンツキーをＡの秘密鍵で署名したデータ［Ｓｉｇ＿Ａ（Ｋｃｏｎ）］をＡの公開鍵証明書［Ａ．Ｃｅｒｔ］から取り出したＡの公開鍵を用いて検証する。検証がＯＫであれば、さらに、公開鍵証明書［Ａ．Ｃｅｒｔ］から取り出したＡのＩＤに基づいて上述したＥＫＢ追跡処理を実行して、ＡのＩＤの示すリーフまたはノード位置においてＥＫＢ処理が可能であるか否かを判定する。
【０１０８】
ＥＫＢの追跡処理により、Ａがリボークされたノードまたはリーフに該当しないことが判

定されると、デバイス［００ｘｘ］は、受領した有効化キーブロックから自身の保有するリーフキー、ノードキーを用いた復号処理により、更新ルートキーＫ（ｔ）ｒｏｏｔを取得する。次に、更新ルートキーＫ（ｔ）ｒｏｏｔによる復号によりさらにコンテンツキーＫｃｏｎを取得する。さらに、取得したコンテンツキーＫｃｏｎによりコンテンツの復号を行なう。これらの処理により、デバイス［００ｘｘ］はコンテンツを利用可能となる。

【０１０９】

上記の処理では、コンテンツの配信者の公開鍵証明書を取得した上で、公開鍵証明書の検証を実行し、コンテンツ配信者の公開鍵と、ＩＤを取得した上で、ＥＫＢの処理、コンテンツの復号を行なうので、コンテンツ配信者の特定がＩＤに基づいて可能であり、配信者が不明瞭なコンテンツが流通するのを防止することが可能となる。

【０１１０】

なお、図１６に示す例は、プロバイダＡがデバイス（ＩＤ＝００ｘｘ）にコンテンツを暗号化して配信する例であり、コンテンツキーに対する署名をプロバイダＡが実行し、デバイスにおいて、プロバイダＡの公開鍵による署名検証処理を行なう例であるが、このように他のプロバイダからの配信コンテンツのデバイスでの記録、再生処理においてではなく、例えば、ユーザの生成したあるいは取得したコンテンツをデバイスの記録媒体に記録する際に、デバイス自身の秘密鍵を用いて署名を行なって記録媒体に記録するようにしてもよい。このように記録媒体に対する格納コンテンツの暗号化キーとしてのコンテンツキーの署名を実行する構成とすれば、コンテンツ再生時に、コンテンツキーの署名検証をデバイスの公開鍵を用いて実行することが必須となり、不正なコンテンツの格納再生の排除が可能となる。

【０１１１】

［階層ツリー構造のカテゴリー分類］

暗号鍵をルートキー、ノードキー、リーフキー等、図３の階層ツリー構造として構成して、コンテンツキーなどを有効化キーブロック（ＥＫＢ）とともに暗号化して配信する構成について説明してきたが、ノードキー等を定義している階層ツリー構造を各デバイスのカテゴリー毎に分類して効率的なキー更新処理を実行する構成について、以下説明する。

【０１１２】

図１７に階層ツリー構造のカテゴリーの分類の一例を示す。図１７において、階層ツリー構造の最上段には、ルートキーＫｒｏｏｔ２３０１が設定され、以下の中間段にはノードキー２３０２が設定され、最下段には、リーフキー２３０３が設定される。各デバイスは個々のリーフキーと、リーフキーからルートキーに至る一連のノードキー、ルートキーを保有する。

【０１１３】

ここで、一例として最上段から第Ｍ段目のあるノードをカテゴリノード２３０４として設定する。すなわち第Ｍ段目のノードの各々を特定カテゴリのデバイス設定ノードとする。第Ｍ段の１つのノードを頂点として以下、Ｍ＋１段以下のノード、リーフは、そのカテゴリに含まれるデバイスに関するノードおよびリーフとする。

【０１１４】

例えば図１７の第Ｍ段目の１つのノード２３０５にはカテゴリ［メモリステッイク（商標）］が設定され、このノード以下に連なるノード、リーフはメモリステッイクを使用した様々なデバイスを含むカテゴリ専用のノードまたはリーフとして設定される。すなわち、ノード２３０５以下を、メモリスティックのカテゴリに定義されるデバイスの関連ノード、およびリーフの集合として定義する。

【０１１５】

さらに、Ｍ段から数段分下位の段をサブカテゴリノード２３０６として設定することができる。例えば図に示すようにカテゴリ［メモリスティック］ノード２３０５の２段下のノードに、メモリスティックを使用したデバイスのカテゴリに含まれるサブカテゴリノードとして、［再生専用器］のノードを設定する。さらに、サブカテゴリノードである再生専用器のノード２３０６以下に、再生専用器のカテゴリに含まれる音楽再生機能付き電話の

ノード２３０７が設定され、さらにその下位に、音楽再生機能付き電話のカテゴリに含まれる［ＰＨＳ］ノード２３０８と［携帯電話］ノード２３０９を設定することができる。

【０１１６】

さらに、カテゴリ、サブカテゴリは、デバイスの種類のみならず、例えばあるメーカー、コンテンツプロバイダ、決済機関等が独自に管理するノード、すなわち処理単位、管轄単位、あるいは提供サービス単位等、任意の単位（これらを総称して以下、エンティティと呼ぶ）で設定することが可能である。例えば１つのカテゴリノードをゲーム機器メーカーの販売するゲーム機器ＸＹＺ専用の頂点ノードとして設定すれば、メーカーの販売するゲーム機器ＸＹＺにその頂点ノード以下の下段のノードキー、リーフキーを格納して販売することが可能となり、その後、暗号化コンテンツの配信、あるいは各種キーの配信、更新処理を、その頂点ノードキー以下のノードキー、リーフキーによって構成される有効化キーブロック（ＥＫＢ）を生成して配信し、頂点ノード以下のデバイスに対してのみ利用可能なデータが配信可能となる。

【０１１７】

このように、１つのノードを頂点としして、以下のノードをその頂点ノードに定義されたカテゴリ、あるいはサブカテゴリの関連ノードとして設定する構成とすることにより、カテゴリ段、あるいはサブカテゴリ段の１つの頂点ノードを管理するメーカー、コンテンツプロバイダ等がそのノードを頂点とする有効化キーブロック（ＥＫＢ）を独自に生成して、頂点ノード以下に属するデバイスに配信する構成が可能となり、頂点ノードに属さない他のカテゴリのノードに属するデバイスには全く影響を及ぼさずにキー更新を実行することができる。

【０１１８】

このようにカテゴリ単位でのＥＫＢによるキー更新とともに、カテゴリ単位、あるいは特定グループでの一括したリボークも可能であり、多くのリボーク・リーフまたはリボーク・ノードが含まれる場合には、上述したＥＫＢ追跡処理によるリボーク判定が特に有効である。なぜなら、すべてのリボークデバイスのＩＤをすべて記録したリストを各デバイスに配信した場合は、リストの格納利用域の問題が発生するとともに、ＩＤの照合処理に費やす負荷も重くなってしまうからである。上述のＩＤに基づくＥＫＢ追跡処理は、ＥＫＢ内のタグに基づく追跡処理であり、その処理負荷は極めて軽く、リボークされているか否かの判定が即座に実行可能となる。

【０１１９】

前述したようにＥＫＢには、ＥＫＢ発行機関の署名がなされており、改竄のチェックが可能であり、正当なＥＫＢであることを署名検証により検証することが可能であり、確実なリボーク判定が実現される。

【０１２０】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０１２１】

【発明の効果】

以上、説明したように、本発明の情報処理装置および方法によれば、コンテンツキー等の配信に適用される階層的鍵配信ツリーを利用した有効化キーブロック（ＥＫＢ）に基づいて、リボーク（排除）エンティティとしてのデバイスやサービスプロバイダなどを判定することを可能としたので、リボークエンティティのＩＤを格納したリボケーションリストをデバイスに配信し、各デバイスがリストを格納する必要がなくなる。

【０１２２】

また、本発明の情報処理装置および方法によれば、公開鍵証明書に階層的鍵配信ツリーの位置識別可能なＩＤを格納し、公開鍵証明書から取得されるＩＤに基づいて有効化キーブ

ロック（ＥＫＢ）のタグを用いた追跡処理を実行する構成としたので、ＩＤの信頼性が公開鍵証明書において保証され、確実なリボークエンティテイ（デバイス）の判定が可能となる。

【図面の簡単な説明】

【図１】本発明の情報処理装置の適用可能なシステムの構成例を説明する図である。

【図２】本発明の情報処理装置として適用可能な記録再生装置の構成例を示すブロック図である。

【図３】本発明の情報処理装置において実行される各種キー、データの暗号化処理について説明するツリー構成図である。

【図４】本発明の情報処理装置に対する各種キー、データの配布に使用される有効化キーブロック（ＥＫＢ）の例を示す図である。

【図５】本発明の情報処理装置に対するコンテンツキーの有効化キーブロック（ＥＫＢ）配布例と復号処理例を示す図である。

【図６】本発明の情報処理装置における有効化キーブロック（ＥＫＢ）のフォーマット例を示す図である。

【図７】有効化キーブロック（ＥＫＢ）のタグの構成を説明する図である。

【図８】有効化キーブロック（ＥＫＢ）と、コンテンツキー、コンテンツを併せて配信するデータ構成例を示す図である。

【図９】有効化キーブロック（ＥＫＢ）と、コンテンツキー、コンテンツを併せて配信した場合のデバイスでの処理例を示す図である。

【図１０】有効化キーブロック（ＥＫＢ）とコンテンツを記録媒体に格納した場合の対応について説明する図である。

【図１１】公開鍵暗号方式による認証処理に伴うリボークエンティテイ検証シーケンスを示す図である。

【図１２】公開鍵証明書の構成例を示す図である。

【図１３】リボークエンティテイ判定のためのＥＫＢ追跡処理について説明する図（例１）である。

【図１４】リボークエンティテイ判定のためのＥＫＢ追跡処理について説明する図（例２）である。

【図１５】リボークエンティテイ判定のためのＥＫＢ追跡処理について説明するフロー図である。

【図１６】ＥＫＢ、公開鍵証明書を用いたコンテンツ配信処理について説明する図（例１）である。

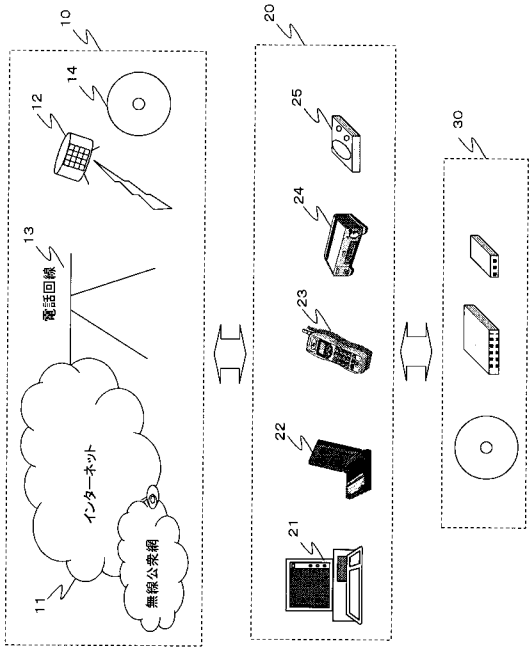
【図１７】階層ツリー構造のカテゴリ分類の例を説明する図である。

【符号の説明】

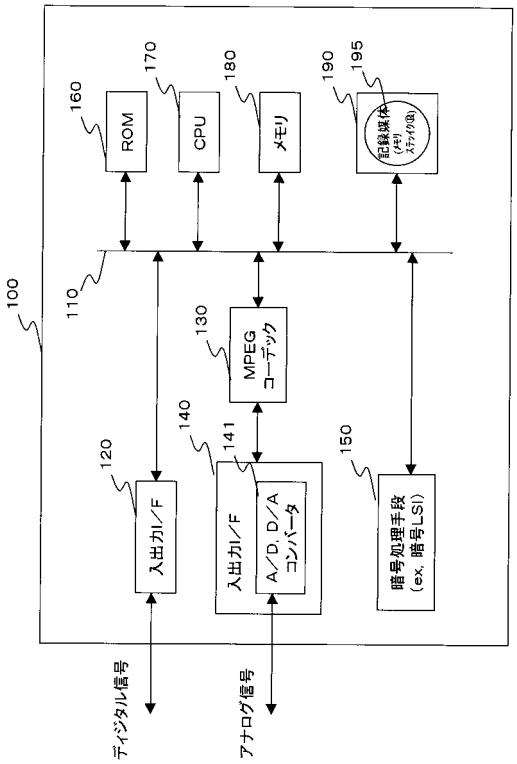
１０　コンテンツ配信側
１１　インターネット
１２　衛星放送
１３　電話回線
１４　メディア
２０　コンテンツ受信側
２１　パーソナルコンピュータ（ＰＣ）
２２　ポータブルデバイス（ＰＤ）
２３　携帯電話、ＰＤＡ
２４　記録再生器
２５　再生専用器
３０　メディア
１００　記録再生装置
１１０　バス
１２０　入出力Ｉ／Ｆ

１３０　ＭＰＥＧコーデック
１４０　入出力Ｉ／Ｆ
１４１　Ａ／Ｄ，Ｄ／Ａコンバータ
１５０　暗号処理手段
１６０　ＲＯＭ
１７０　ＣＰＵ
１８０　メモリ
１９０　ドライブ
１９５　記録媒体
６０１　バージョン
６０２　デプス
６０３　データポインタ
６０４　タグポインタ
６０５　署名ポインタ
６０６　データ部
６０７　タグ部
６０８　署名
２３０１　ルートキー
２３０２　ノードキー
２３０３　リーフキー
２３０４　カテゴリノード
２３０６　サブカテゴリノード

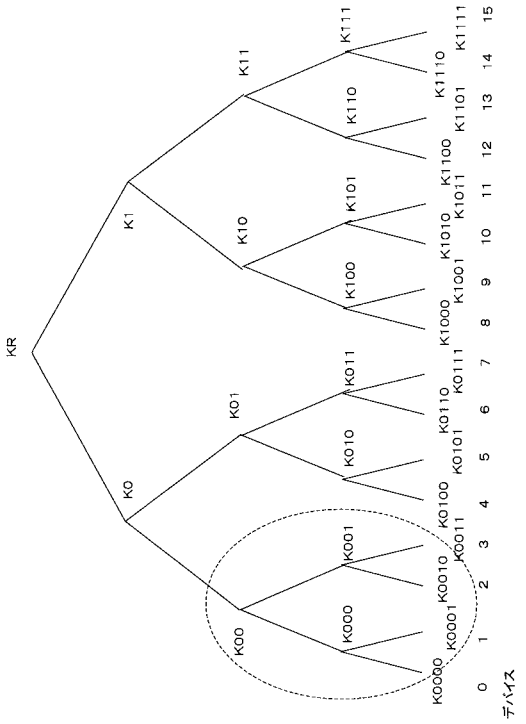
【図１】



【図２】



【図３】



【図４】

（A） 有効化キーブロック(EKB:Enabling Key Block) 例1

デバイス0, 1, 2にバージョン:tのノードキーを送付

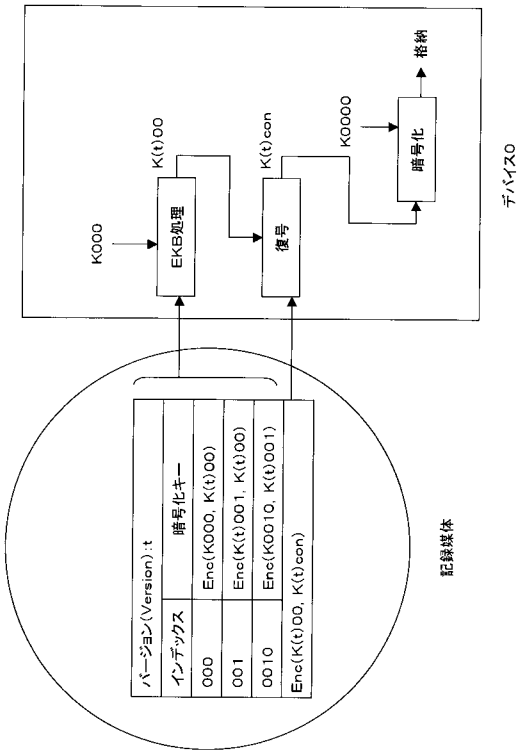
バージョン(Version):t	
インデックス	暗号化キー
0	Enc(K(t)0, K(t)R)
00	Enc(K(t)00, K(t)0)
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

（B） 有効化キーブロック(EKB:Enabling Key Block) 例2

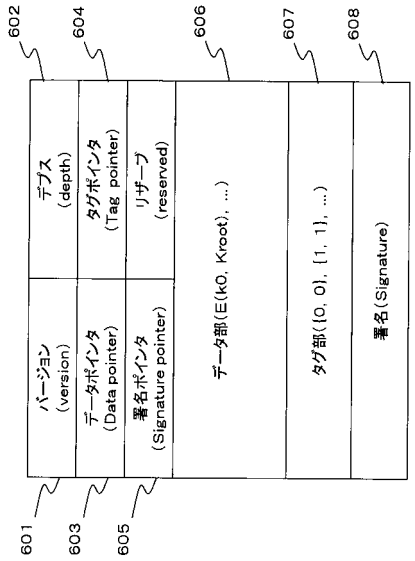
デバイス0, 1, 2にバージョン:tのノードキーを送付

バージョン(Version):t	
インデックス	暗号化キー
000	Enc(K000, K(t)00)
001	Enc(K(t)001, K(t)00)
0010	Enc(K0010, K(t)001)

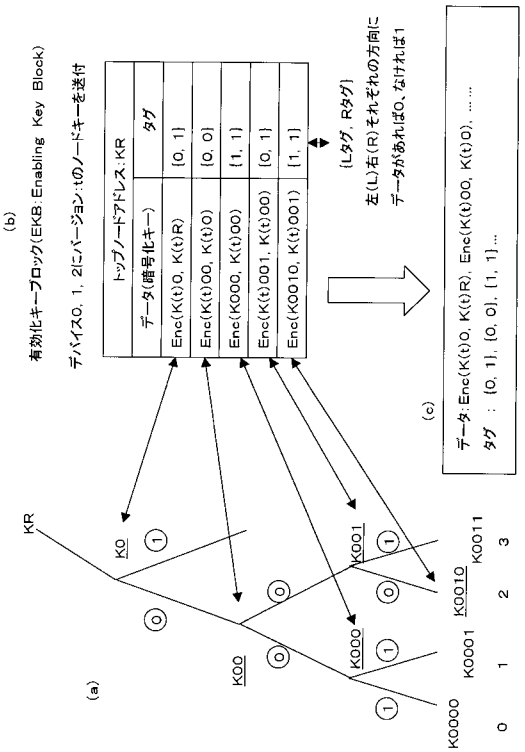
【図５】



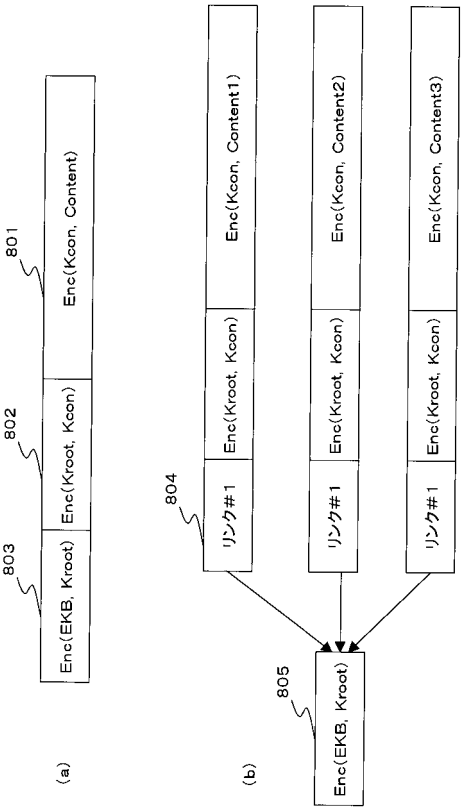
【図６】



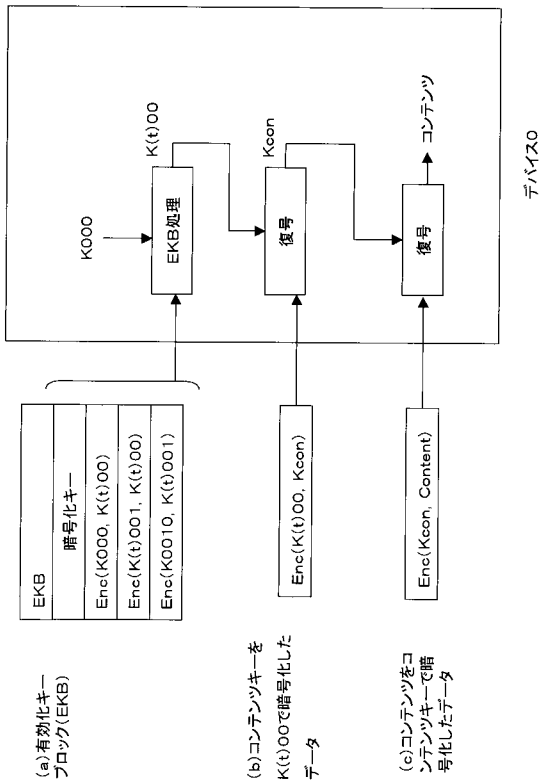
【図７】


(b)
有効化キーブロック(EKB:Enabling Key Block)
デバイス0, 1, 2にバージョン:tのノードキーを送付
トップノードアドレス:KR
データ(暗号化キー)
タグ
Enc(K(t)0, K(t)R)
[0, 1]
Enc(K(t)00, K(t)0)
[0, 0]
Enc(K000, K(t)00)
[1, 1]
Enc(K(t)001, K(t)00)
[0, 1]
Enc(K0010, K(t)001)
[1, 1]
{Lタグ, Rタグ}
左(L)右(R)それぞれの方向に
データがあれば0、なければ1
(c)
データ:Enc(K(t)0, K(t)R), Enc(K(t)00, K(t)0), ...
タグ : [0, 1], [0, 0], [1, 1] ...
KR
K0
K00
K000
K001
K0010
K0011
K0000
K0001
0
1
2
3
(a)

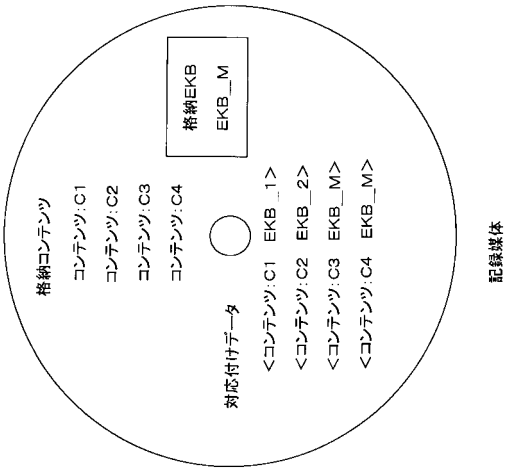
【図８】


801
802
803
Enc(Kroot, Kcon)
Enc(Kcon, Content)
Enc(EKB, Kroot)
(a)
804
リンク#1
Enc(Kroot, Kcon)
Enc(Kcon, Content1)
Enc(Kcon, Content2)
Enc(Kcon, Content3)
805
Enc(EKB, Kroot)
(b)

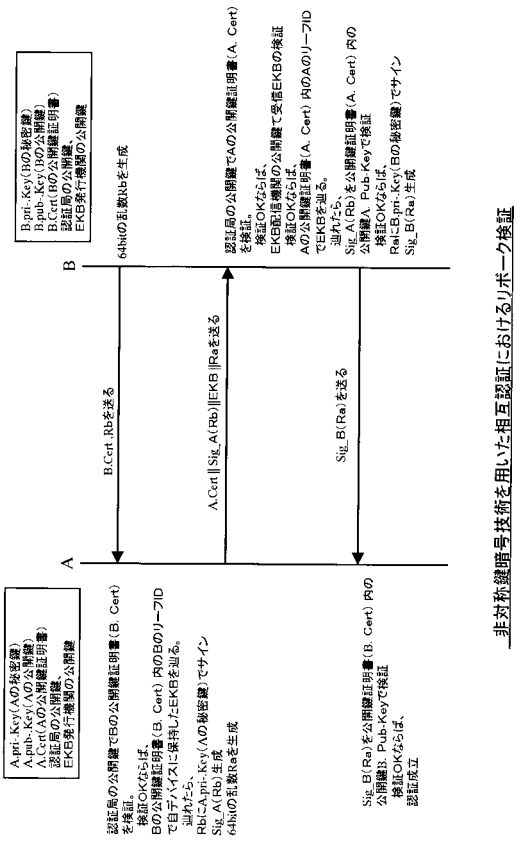
【図９】


EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
K000
EKB処理
K(t)00
Enc(K(t)00, Kcon)
復号
Kcon
Enc(Kcon, Content)
復号
コンテンツ
デバイス0
(a)有効化キーブロック(EKB)
(b)コンテンツキーをK(t)00で暗号化したデータ
(c)コンテンツをコンテンツキーで暗号化したデータ

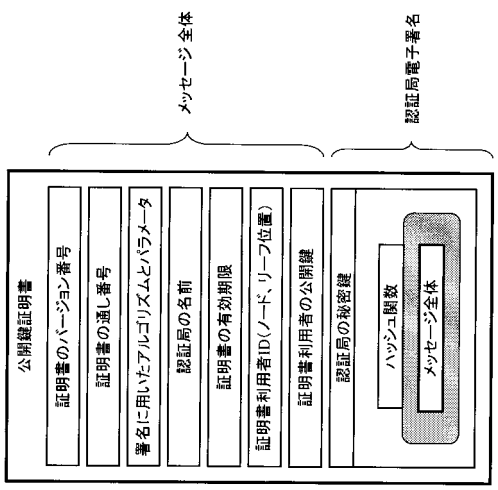
【図１０】


格納コンテンツ
コンテンツ:C1
コンテンツ:C2
コンテンツ:C3
コンテンツ:C4
対応付けデータ
格納EKB
EKB_M
<コンテンツ:C1 EKB_1>
<コンテンツ:C2 EKB_2>
<コンテンツ:C3 EKB_M>
<コンテンツ:C4 EKB_M>
記録媒体

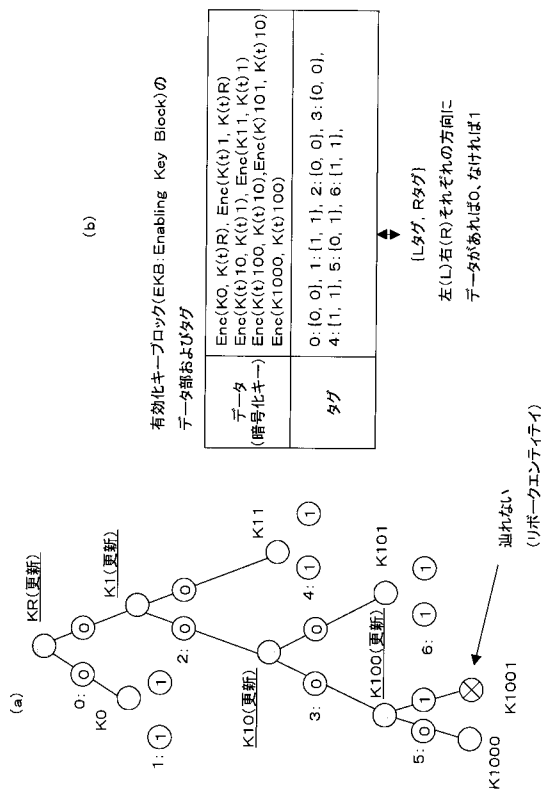
【図１１】



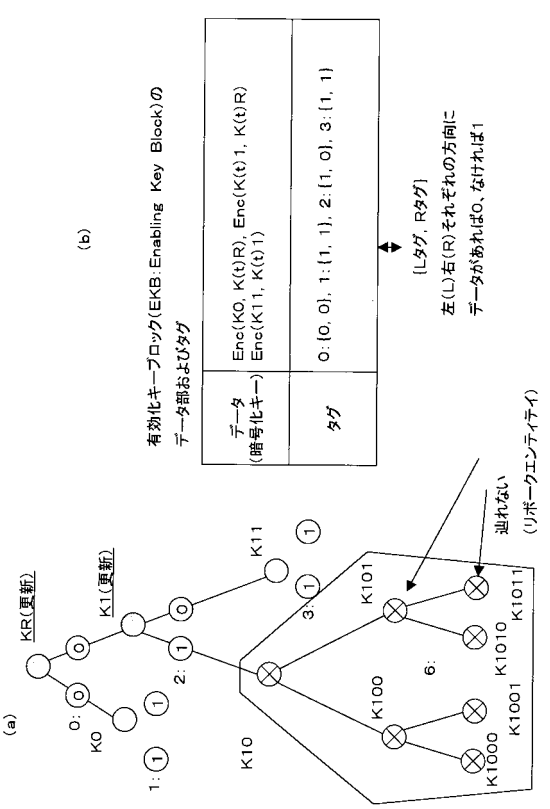
【図１２】



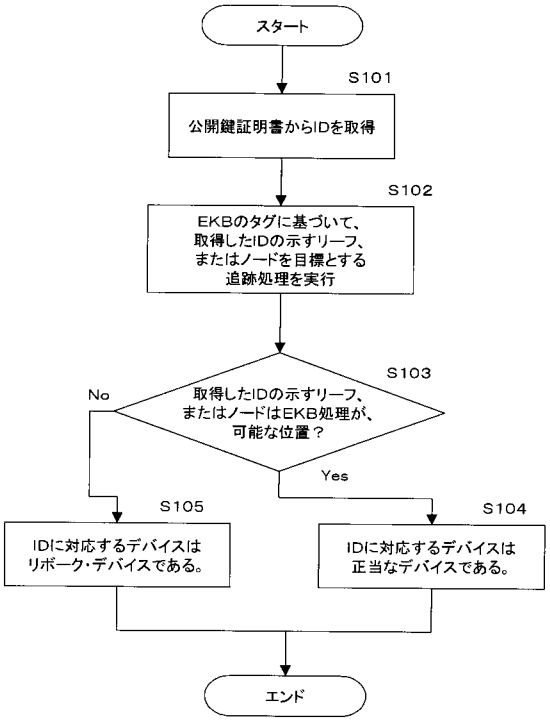
【図１３】



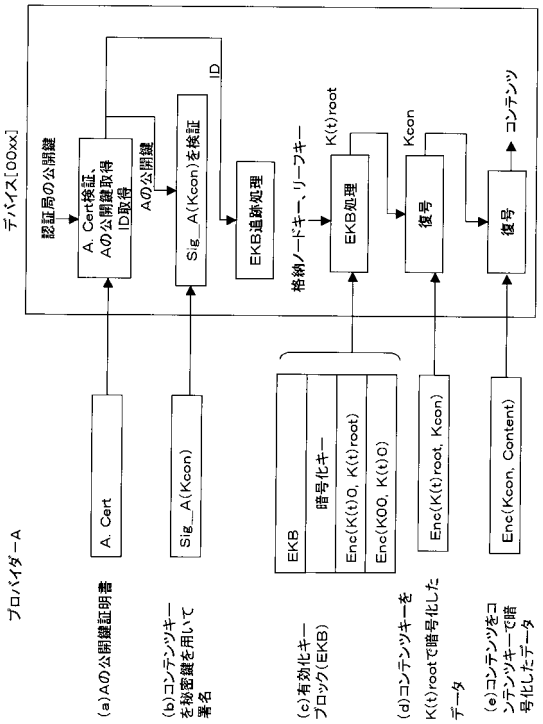
【図１４】



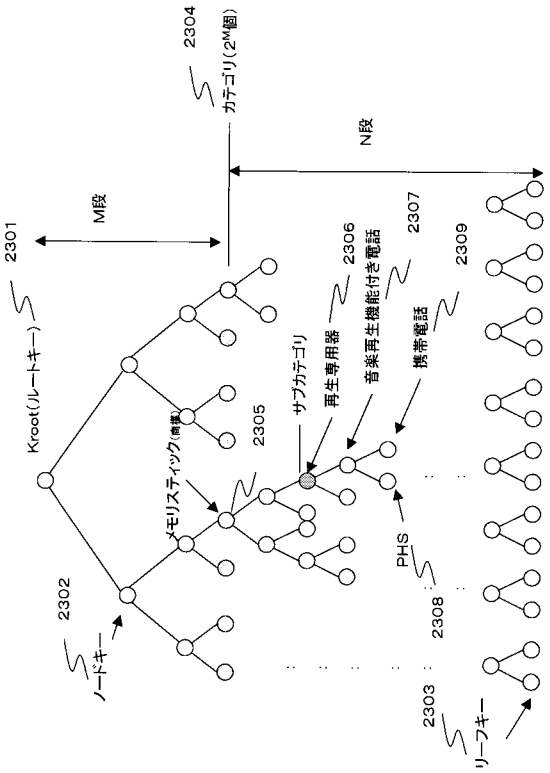
【図１５】


スタート
S101
公開鍵証明書からIDを取得
S102
EKBのタグに基づいて、
取得したIDの示すリーフ、
またはノードを目標とする
追跡処理を実行
S103
取得したIDの示すリーフ、
またはノードはEKB処理が、
可能な位置？
No
Yes
S105
IDに対応するデバイスは
リボーク・デバイスである。
S104
IDに対応するデバイスは
正当なデバイスである。
エンド

【図１６】


デバイス[00xx]
認証局の公開鍵
A. Cert検証、
Aの公開鍵取得
ID取得
Aの公開鍵
ID
Sig_A(Kcon)を検証
EKB追跡処理
格納ノードキー、リーフキー
K(t)root
EKB処理
Kcon
復号
コンテンツ
復号
プロバイダーA
A. Cert
Sig_A(Kcon)
EKB
暗号化キー
Enc(K(t)0, K(t)root)
Enc(K00, K(t)0)
Enc(K(t)root, Kcon)
Enc(Kcon, Content)
(a)Aの公開鍵証明書
(b)コンテンツキーを秘密鍵を用いて署名
(c)有効化キーブロック(EKB)
(d)コンテンツキーをK(t)rootで暗号化したデータ
(e)コンテンツをコンテンツキーで暗号化したデータ

【図１７】


2304
カテゴリ(2M個)
N段
M段
2301
Kroot(ルートキー)
2306
音楽再生機能付き電話
2307
2309
携帯電話
サブカテゴリ
再生専用器
2305
メモリスティック(商標)
PHS
2302
ノードキー
2308
2303
リーフキー

フロントページの続き

審査官　中里　裕正

(56)参考文献　特開平１０－０４０２５５（ＪＰ，Ａ）
特開平１１－１８７０１３（ＪＰ，Ａ）
特開平１１－２０５３０５（ＪＰ，Ａ）
特開２０００－０３１９２２（ＪＰ，Ａ）
米国特許第０５９４９８７７（ＵＳ，Ａ）
米国特許第０６０４９８７８（ＵＳ，Ａ）
5C Digital Transmission Content Protection White Paper, １９９８年　7月１４日, Revision 1.0, ＵＲＬ, http://www.dtcp.com/data/wp_spec.pdf
The VersaKey framework: versatile group key management, IEEE Journal on Selected Areas in Communications, １９９９年　9月, Volume: 17, Issue: 9, p.1614-1631
Secure group communications using key graphs, EEE/ACM Transactions on Networking, ２０００年　2月, Volume: 8, Issue: 1, p.16-30

(58)調査した分野(Int.Cl., ＤＢ名)
H04L　9/08
JSTPlus/JMEDPlus/JST7580(JDreamII)