(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2002 年10 月10 日 (10.10.2002)

PCT

(10) 国際公開番号
WO 02/080448 A1

(51) 国際特許分類[7]: H04L 9/08, G06F 17/60

(21) 国際出願番号: PCT/JP02/02958

(22) 国際出願日: 2002 年3 月27 日 (27.03.2002)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2001-94809 2001 年3 月29 日 (29.03.2001) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都 品川区 北品川６丁目７番３５号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 石黒 隆二 (ISHIGURO,Ryuji) [JP/JP]; 〒141-0001 東京都 品川区 北品川６丁目７番３５号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 稲本 義雄 (INAMOTO,Yoshio); 〒160-0023 東京都 新宿区 西新宿７丁目１１番１８号 ７１１ビルディング４階 Tokyo (JP).

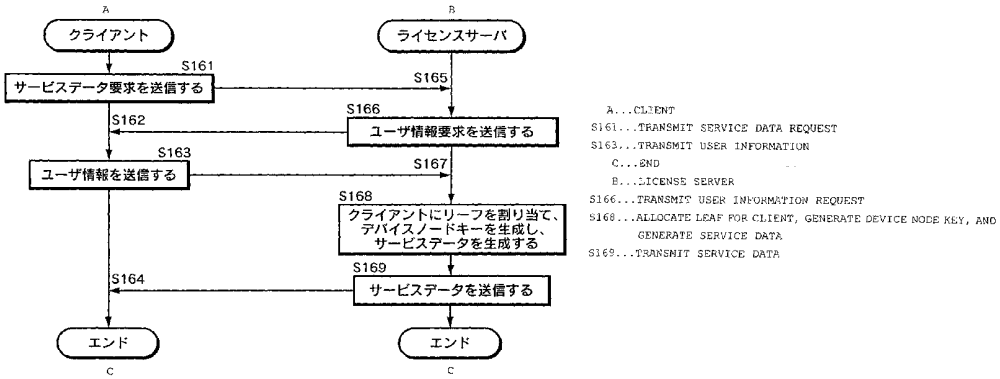
(81) 指定国 (国内): JP, KR, US.

(84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).

添付公開書類:
— 国際調査報告書

(54) Title: INFORMATION PROCESSING APPARATUS

(54) 発明の名称: 情報処理装置



(57) Abstract: A license server providing a key for decoding a content transmits a user information request to a client upon reception of a service data request. When the client receives the user information request, the client displays a message prompting input of user information. When personal information of the user and user information such as a transaction are input, they are transmitted to the license server. Upon reception of the user information, the license server allocates a leaf of a key management hierarchical tree structure, generates a set of node keys as a device node key, transmits it together with a leaf ID, a client secret key, and the like to the client, and records the user information by relating it to the leaf ID.

[続葉有]

WO 02/080448 A1

(57) 要約:

コンテンツを復号するための鍵を提供するライセンスサーバは、サービスデータ要求を受信すると、ユーザ情報要求をクライアントに送る。 クライアントは、ユーザ情報要求を受信するとユーザ情報の入力を促すメッセージを表示し、ユーザ本人の個人情報や決済情報等のユーザ情報が入力されると、それをライセンスサーバに送信する。ライセンスサーバは、ユーザ情報を受信すると、鍵管理階層ツリー構造のリーフをクライアントに割り当て、ノードキーの組をデバイスノードキーとして生成し、リーフＩＤ、クライアントの秘密鍵等と共にクライアントに送信し、ユーザ情報をリーフＩＤと対応づけて記録する。

明細書

情報処理装置

技術分野

本発明は、情報処理装置に関し、特に、コンテンツの流通を妨げることなく、コンテンツの著作権を確実に管理することができるようにした情報処理装置に関する。

背景技術

従来、音楽や映像等のコンテンツの著作権を管理する場合、コンテンツのコピーを制限するなどして、コンテンツそのものの不正なコピーを防止するようにしていた。

しかしながら、コンテンツそのものの不正なコピーを防止するようにすると、コンテンツそのものの配布方法が制限されるため、コンテンツを大量に、かつ効率的に多くのユーザに配布すること自体が困難となる課題があった。

発明の開示

本発明は、このような状況に鑑みてなされたものであり、コンテンツを大量、かつ効率的に配布することができるようにするとともに、コンテンツの著作権を確実に管理することができるようにするものである。

本発明の情報処理装置は、他の情報処理装置からキー要求を受信する受信手段と、他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成手段と、生成手段により生成されたデバイスノードキーを他の情報処理装置に送信する送信手段とを備えることを特徴とする。

前記送信手段は、前記他の情報処理装置に割り当てられたリーフを識別するリーフ識別情報をさらに送信するようにすることができる。

前記送信手段は、他の情報処理装置に割り当てられた秘密鍵をさらに送信するようにすることができる。

前記ライセンス毎に使用条件を記憶する記憶手段をさらに備えるようにすることができる。

前記受信手段は、他の情報処理装置の利用者に関する利用者情報をさらに受信するようにすることができる。

前記受信手段により受信された利用者情報を、他の情報処理装置に割り当てられたリーフを識別するリーフ識別情報と対応付けて記録する記録手段をさらに備えるようにすることができる。

本発明の情報処理方法は、他の情報処理装置からキー要求を受信する受信ステップと、他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、生成ステップの処理により生成されたデバイスノードキーを他の情報処理装置に送信する送信ステップとを含むことを特徴とする。

本発明の記録媒体のプログラムは、他の情報処理装置からキー要求を受信する受信ステップと、他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、生成ステップの処理により生成されたデバイスノードキーを他の情報処理装置に送信する送信ステップとを含むことを特徴とする。

本発明のプログラムは、他の情報処理装置からキー要求を受信する受信ステップと、他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、生成ステップの処理により生成されたデバイスノードキーを他の情報処理装置に送信する送信ステップとをコンピュータに実現させる。

本発明においては、他の情報処理装置からキー要求が受信され、他の情報処理装置が鍵管理階層ツリー構造のリーフに割り当てられ、デバイスノードキーが生成され、その生成されたデバイスノードキーが他の情報処理装置に送信される。

図面の簡単な説明

図１は、本発明を適用したコンテンツ提供システムの構成を示すブロック図である。

図２は、図１のクライアントの構成を示すブロック図である。

図３は、図１のクライアントのコンテンツのダウンロード処理を説明するフローチャートである。

図４は、図１のコンテンツサーバのコンテンツ提供処理を説明するフローチャートである。

図５は、図４のステップＳ２６におけるフォーマットの例を示す図である。

図６は、図１のクライアントのコンテンツ再生処理を説明するフローチャートである。

図７は、図６のステップＳ４３のライセンス取得処理の詳細を説明するフローチャートである。

図８は、ライセンスの構成を示す図である。

図９は、図１のライセンスサーバのライセンス提供の処理を説明するフローチャートである。

図１０は、図６のステップＳ４５におけるライセンス更新処理の詳細を説明するフローチャートである。

図１１は、図１のライセンスサーバのライセンス更新処理を説明するフローチャートである。

図１２は、キーの構成を説明する図である。

図１３は、カテゴリノードを説明する図である。

図１４は、ノードとデバイスの対応の具体例を示す図である。

図１５Ａは、有効化キーブロックの構成を説明する図である。

図１５Ｂは、有効化キーブロックの構成を説明する図である。

図１６は、有効化キーブロックの利用を説明する図である。

図１７は、有効化キーブロックのフォーマットの例を示す図である。

図１８は、有効化キーブロックのタグの構成を説明する図である。

図１９は、DNKを用いたコンテンツの復号処理を説明する図である。

図２０は、有効化キーブロックの例を示す図である。

図２１は、複数のコンテンツの１つのデバイスに対する割り当てを説明する図である。

図２２は、ライセンスのカテゴリを説明する図である。

図２３は、登録処理を説明するタイミングチャートである。

図２４は、クライアントのリッピング処理を説明するフローチャートである。

図２５は、ウォーターマークの構成を説明する図である。

図２６は、コンテンツのフォーマットの例を示す図である。

図２７は、公開鍵証明書の例を示す図である。

図２８は、コンテンツの配布を説明する図である。

図２９は、クライアントのコンテンツのチェックアウト処理を説明するフローチャートである。

図３０は、タグによる有効化キーブロックをたどる例を説明する図である。

図３１は、有効化キーブロックの構成例を示す図である。

図３２は、マークの構成を説明する図である。

図３３は、クライアントのライセンス買い取り処理を説明するフローチャートである。

図３４は、ライセンスサーバのライセンス買い取り処理を説明するフローチャートである。

図３５は、マークの構成例を示す図である。

図３６は、クライアントの証明書の登録処理を説明するフローチャートである。

図３７は、コンテンツサーバの証明書登録処理を説明するフローチャートである。

図３８は、グループの証明書の例を示す図である。

図３９は、グルーピングが行われている場合におけるコンテンツサーバの処理を説明するフローチャートである。

図４０は、コンテンツキーの暗号化の例を示す図である。

図４１は、グループに属するクライアントの処理を説明するフローチャートである。

図４２は、他のクライアントにライセンスをチェックアウトするクライアントの処理を説明するフローチャートである。

図４３は、他のクライアントからライセンスのチェックアウトを受けるクライアントの処理を説明するフローチャートである。

図４４は、ライセンスのチェックアウトを受けたクライアントの再生処理を説明するフローチャートである。

図４５は、他のクライアントからライセンスのチェックインを受けるクライアントの処理を説明するフローチャートである。

図４６は、他のクライアントにライセンスをチェックインするクライアントの処理を説明するフローチャートである。

図４７は、MAC の生成を説明する図である。

図４８は、ICV 生成キーの復号処理を説明するフローチャートである。

図４９は、ICV 生成キーの他の復号処理を説明する図である。

図５０Ａは、ICV によるライセンスのコピーの管理を説明する図である。

図５０Ｂは、ICV によるライセンスのコピーの管理を説明する図である。

図５１は、ライセンスの管理を説明する図である。

発明を実施するための最良の形態

図１は、本発明を適用したコンテンツ提供システムの構成を示している。インターネット２には、クライアント１－１，１－２（以下、これらのクライアントを個々に区別する必要がない場合、単にクライアント１と称する）が接続されて

いる。この例においては、クライアントが２台のみ示されているが、インターネット２には、任意の台数のクライアントが接続される。

また、インターネット２には、クライアント１に対してコンテンツを提供するコンテンツサーバ３、コンテンツサーバ３が提供するコンテンツを利用するのに必要なライセンスをクライアント１に対して付与するライセンスサーバ４、およびクライアント１がライセンスを受け取った場合に、そのクライアント１に対して課金処理を行う課金サーバ５が接続されている。

これらのコンテンツサーバ３、ライセンスサーバ４、および課金サーバ５も、任意の台数、インターネット２に接続される。

図２はクライアント１の構成を表している。

図２において、CPU（Central Processing Unit）２１は、ROM（Read Only Memory）２２に記憶されているプログラム、または記憶部２８からRAM（Random Access Memory）２３にロードされたプログラムに従って各種の処理を実行する。タイマ２０は、計時動作を行い、時刻情報をCPU２１に供給する。RAM２３にはまた、CPU２１が各種の処理を実行する上において必要なデータなども適宜記憶される。

暗号化復号部２４は、コンテンツデータを暗号化するとともに、既に暗号化されているコンテンツデータを復号する処理を行う。コーデック部２５は、例えば、ATRAC（Adaptive Transform Acoustic Coding）３方式などでコンテンツデータをエンコードし、入出力インタフェース３２を介してドライブ３０に接続されている半導体メモリ４４に供給し、記録させる。あるいはまた、コーデック部２５は、ドライブ３０を介して半導体メモリ４４より読み出した、エンコードされているデータをデコードする。

半導体メモリ４４は、例えば、メモリスティック（商標）などにより構成される。

CPU２１、ROM２２、RAM２３、暗号化復号部２４、およびコーデック部２５は、バス３１を介して相互に接続されている。このバス３１にはまた、入出力インタフェース３２も接続されている。

入出力インタフェース３２には、キーボード、マウスなどよりなる入力部２６、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部２７、ハードディスクなどより構成される記憶部２８、モデム、ターミナルアダプタなどより構成される通信部２９が接続されている。通信部２９は、インターネット２を介しての通信処理を行う。通信部２９はまた、他のクライアントとの間で、アナログ信号またはデジタル信号の通信処理を行う。

入出力インタフェース３２にはまた、必要に応じてドライブ３０が接続され、磁気ディスク４１、光ディスク４２、光磁気ディスク４３、或いは半導体メモリ４４などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部２８にインストールされる。

なお、図示は省略するが、コンテンツサーバ３、ライセンスサーバ４、課金サーバ５も、図２に示したクライアント１と基本的に同様の構成を有するコンピュータにより構成される。そこで、以下の説明においては、図２の構成は、コンテンツサーバ３、ライセンスサーバ４、課金サーバ５などの構成としても引用される。

次に、図３のフローチャートを参照して、クライアント１がコンテンツサーバ３からコンテンツの提供を受ける処理について説明する。

ユーザが、入力部２６を操作することでコンテンツサーバ３に対するアクセスを指令すると、CPU２１は、ステップＳ１において、通信部２９を制御し、インターネット２を介してコンテンツサーバ３にアクセスさせる。ステップＳ２において、ユーザが、入力部２６を操作して、提供を受けるコンテンツを指定すると、CPU２１は、この指定情報を受け取り、通信部２９から、インターネット２を介してコンテンツサーバ３に、指定されたコンテンツを通知する。図４のフローチャートを参照して後述するように、この通知を受けたコンテンツサーバ３

は、暗号化されたコンテンツデータを送信してくるので、ステップＳ３において、CPU２１は、通信部２９を介して、このコンテンツデータを受信すると、ステップＳ４において、その暗号化されているコンテンツデータを記憶部２８を構成するハードディスクに供給し、記憶させる。

次に、図４のフローチャートを参照して、クライアント１の以上の処理に対応するコンテンツサーバ３のコンテンツ提供処理について説明する。なお、以下の説明において、図２のクライアント１の構成は、コンテンツサーバ３の構成としても引用される。

ステップＳ２１において、コンテンツサーバ３のCPU２１は、インターネット２から通信部２９を介してクライアント１よりアクセスを受けるまで待機し、アクセスを受けたと判定したとき、ステップＳ２２に進み、クライアント１から送信されてきたコンテンツを指定する情報を取り込む。このコンテンツを指定する情報は、クライアント１が、図３のステップＳ２において通知してきた情報である。

ステップＳ２３において、コンテンツサーバ３のCPU２１は、記憶部２８に記憶されているコンテンツデータの中から、ステップＳ２２の処理で取り込まれた情報で指定されたコンテンツを読み出す。CPU２１は、ステップＳ２４において、記憶部２８から読み出されたコンテンツデータを、暗号化復号部２４に供給し、コンテンツキーKcを用いて暗号化させる。

記憶部２８に記憶されているコンテンツデータは、コーデック部２５により、既にATRAC３方式によりエンコードされているので、このエンコードされているコンテンツデータが暗号化されることになる。

なお、もちろん、記憶部２８に予め暗号化した状態でコンテンツデータを記憶させることができる。この場合には、ステップＳ２４の処理は省略することが可能である。

次に、ステップＳ２５において、コンテンツサーバ３のCPU２１は、暗号化したコンテンツデータを伝送するフォーマットを構成するヘッダに、暗号化され

ているコンテンツを復号するのに必要なキー情報（図５を参照して後述するEKB（Enabling Key Block）とK_{EKBC}（Kc））と、コンテンツを利用するのに必要なライセンスを識別するためのライセンスIDを付加する。そして、ステップＳ２６において、コンテンツサーバ３のCPU２１は、ステップＳ２４の処理で暗号化したコンテンツと、ステップＳ２５の処理でキーとライセンスIDを付加したヘッダとをフォーマット化したデータを、通信部２９から、インターネット２を介して、アクセスしてきたクライアント１に送信する。

図５は、このようにして、コンテンツサーバ３からクライアント１にコンテンツが供給される場合のフォーマットの構成を表している。同図に示されるように、このフォーマットは、ヘッダ（Header）とデータ（Data）とにより構成される。

ヘッダには、コンテンツ情報（Content information）、URL（Uniform Resource Locator）、ライセンスID（License ID）、イネーブリングキーブロック（有効化キーブロック）（EKB（Enabling Key Block））および、EKBから生成されたキーK_{EKBC}を用いて暗号化されたコンテンツキーKcとしてのデータK_{EKBC}（Kc）が配置されている。なお、EKBについては、図１５Ａおよび図１５Ｂを参照して後述する。

コンテンツ情報には、データとしてフォーマット化されているコンテンツデータを識別するための識別情報としてのコンテンツID（CID）、そのコンテンツのコーデックの方式などの情報が含まれている。

URLは、ライセンスIDで規定されるライセンスを取得するときアクセスするアドレス情報であり、図１のシステムの場合、具体的には、ライセンスを受けるために必要なライセンスサーバ４のアドレスである。ライセンスIDは、データとして記録されているコンテンツを利用するとき必要とされるライセンスを識別するものである。

データは、任意の数の暗号化ブロック（Encryption Block）により構成される。各暗号化ブロックは、イニシャルベクトル（IV（Initial Vector））、シー

ド（Seed）、およびコンテンツデータをキーK'cで暗号化したデータEK'c(data)により構成されている。

キーK'cは、次式により示されるように、コンテンツキーKcと、乱数で設定される値Seedをハッシュ関数に適用して演算された値により構成される。

K'c＝Hash(Kc,Seed)

イニシャルベクトルIVとシードSeedは、各暗号化ブロック毎に異なる値に設定される。

この暗号化は、コンテンツのデータを８バイト単位で区分して、８バイト毎に行われる。後段の８バイトの暗号化は、前段の８バイトの暗号化の結果を利用して行われるCBC（Cipher Block Chaining）モードで行われる。

CBCモードの場合、最初の８バイトのコンテンツデータを暗号化するとき、その前段の８バイトの暗号化結果が存在しないため、最初の８バイトのコンテンツデータを暗号化するときは、イニシャルベクトルIVを初期値として暗号化が行われる。

このCBCモードによる暗号化を行うことで、１つの暗号化ブロックが解読されたとしても、その影響が、他の暗号化ブロックにおよぶことが抑制される。

なお、この暗号化については、図４７を参照して、後に詳述する。

また、暗号方式についてはこれに限らず、単にコンテンツキーKcでコンテンツデータを暗号化しても良い。

以上のようにして、クライアント１は、コンテンツサーバ３からコンテンツを無料で、自由に取得することができる。従って、コンテンツそのものは、大量に、配布することが可能となる。

しかしながら、各クライアント１は、取得したコンテンツを利用するとき、ライセンスを保持している必要がある。そこで、図６を参照して、クライアント１がコンテンツを再生する場合の処理について説明する。

ステップＳ４１において、クライアント１のCPU２１は、ユーザが入力部２６を操作することで指示したコンテンツの識別情報（CID）を取得する。この識

別情報は、例えば、コンテンツのタイトルや、記憶されている各コンテンツ毎に付与されている番号などにより構成される。

そして、CPU２１は、コンテンツが指示されると、そのコンテンツに対応するライセンスID（そのコンテンツを使用するのに必要なライセンスのID）を読み取る。このライセンスIDは、図５に示されるように、暗号化されているコンテンツデータのヘッダに記述されているものである。

次に、ステップＳ４２に進み、CPU２１は、ステップＳ４１で読み取られたライセンスIDに対応するライセンスが、クライアント１により既に取得され、記憶部２８に記憶されているか否かを判定する。まだ、ライセンスが取得されていない場合には、ステップＳ４３に進み、CPU２１は、ライセンス取得処理を実行する。このライセンス取得処理の詳細は、図７のフローチャートを参照して後述する。

ステップＳ４２において、ライセンスが既に取得されていると判定された場合、または、ステップＳ４３において、ライセンス取得処理が実行された結果、ライセンスが取得された場合、ステップＳ４４に進み、CPU２１は、取得されているライセンスは有効期限内のものであるか否かを判定する。ライセンスが有効期限内のものであるか否かは、ライセンスの内容として規定されている期限（後述する図８参照）と、タイマ２０により計時されている現在日時と比較することで判断される。ライセンスの有効期限が既に満了していると判定された場合、CPU２１は、ステップＳ４５に進み、ライセンス更新処理を実行する。このライセンス更新処理の詳細は、図１０のフローチャートを参照して後述する。

ステップＳ４４において、ライセンスはまだ有効期限内であると判定された場合、または、ステップＳ４５において、ライセンスが更新された場合、ステップＳ４６に進み、CPU２１は、暗号化されているコンテンツデータを記憶部２８から読み出し、RAM２３に格納させる。そして、ステップＳ４７において、CPU２１は、RAM２３に記憶された暗号化ブロックのデータを、図５のデータ

に配置されている暗号化ブロック単位で、暗号化復号部２４に供給し、コンテンツキーKcを用いて復号させる。

コンテンツキーKcを得る方法の具体例は、図１５Ａおよび図１５Ｂを参照して後述するが、デバイスノードキー（DNK(Device Node Key)）を用いて、EKB（図５）に含まれるキーK_{EKBC}を得ることができ、そのキーK_{EKBC}を用いて、データK_{EKBC}（Kc）（図５）から、コンテンツキーKcを得ることができる。

CPU２１は、さらに、ステップＳ４８において、暗号化復号部２４により復号されたコンテンツデータをコーデック部２５に供給し、デコードさせる。そして、コーデック部２５によりデコードされたデータを、CPU２１は、入出力インタフェース３２から出力部２７に供給し、D/A変換させ、スピーカから出力させる。

次に、図７のフローチャートを参照して、図６のステップＳ４３で行われるライセンス取得処理の詳細について説明する。

クライアント１は、事前にライセンスサーバにアクセスして登録処理を行うことにより、リーフID、DNK(Device Node Key)、クライアント１の秘密鍵・公開鍵のペア、ライセンスサーバの公開鍵、及び各公開鍵の証明書を含むサービスデータを取得しておく。クライアントの登録処理の詳細は図２３を参照して後述する。

リーフIDは、クライアント毎に割り当てられた識別情報を表し、DNKは、そのライセンスに対応するEKB（有効化キーブロック）に含まれる暗号化されているコンテンツキーKcを復号するのに必要なデバイスノードキーである（図１２を参照して後述する）。

最初にステップＳ６１において、CPU２１は、いま処理対象とされているライセンスIDに対応するURLを、図５に示すヘッダから取得する。上述したように、このURLは、やはりヘッダに記述されているライセンスIDに対応するライセンスを取得するときアクセスすべきアドレスである。そこで、ステップＳ６２において、CPU２１は、ステップＳ６１で取得したURLにアクセスする。

具体的には、通信部２９によりインターネット２を介してライセンスサーバ４にアクセスが行われる。このとき、ライセンスサーバ４は、クライアント１に対して、購入するライセンス（コンテンツを使用するのに必要なライセンス）を指定するライセンス指定情報、並びにユーザIDとパスワードの入力を要求してくる（後述する図９のステップＳ１０２）。CPU２１は、この要求を出力部２７の表示部に表示させる。ユーザは、この表示に基づいて、入力部２６を操作して、ライセンス指定情報、ユーザID、およびパスワードを入力する。なお、このユーザIDとパスワードは、クライアント１のユーザが、インターネット２を介してライセンスサーバ４にアクセスし、事前に取得しておいたものである。

CPU２１は、ステップＳ６３，Ｓ６４において、入力部２６から入力されたライセンス指定情報を取り込むとともに、ユーザIDとパスワードを取り込む。CPU２１は、ステップＳ６５において、通信部２９を制御し、入力されたユーザIDとパスワード、ライセンス指定情報、並びにサービスデータ（後述する）に含まれるリーフIDを含むライセンス要求を、インターネット２を介してライセンスサーバ４に送信させる。

ライセンスサーバ４は、図９を参照して後述するように、ユーザIDとパスワード、並びにライセンス指定情報に基づいてライセンスを送信してくる（ステップＳ１０９）か、または、条件が満たされない場合には、ライセンスを送信してこない（ステップＳ１１２）。

ステップＳ６６において、CPU２１は、ライセンスサーバ４からライセンスが送信されてきたか否かを判定し、ライセンスが送信されてきた場合には、ステップＳ６７に進み、そのライセンスを記憶部２８に供給し、記憶させる。

ステップＳ６６において、ライセンスが送信されて来ないと判定した場合、CPU２１は、ステップＳ６８に進み、エラー処理を実行する。具体的には、CPU２１は、コンテンツを利用するためのライセンスが得られないので、コンテンツの再生処理を禁止する。

以上のようにして、各クライアント１は、コンテンツデータに付随しているライセンスIDに対応するライセンスを取得して、初めて、そのコンテンツを使用することが可能となる。

なお、図７のライセンス取得処理は、各ユーザがコンテンツを取得する前に、予め行っておくようにすることも可能である。

クライアント１に提供されるライセンスは、例えば、図８に示されるように、使用条件、リーフID等を含んでいる。

使用条件には、そのライセンスに基づいて、コンテンツを使用することが可能な使用期限、そのライセンスに基づいて、コンテンツをダウンロードすることが可能なダウンロード期限、そのライセンスに基づいて、コンテンツをコピーすることが可能な回数（許されるコピー回数）、チェックアウト回数、最大チェックアウト回数、そのライセンスに基づいて、コンテンツをCD-Rに記録することができる権利、PD（Portable Device）にコピーすることが可能な回数、ライセンスを所有権（買い取り状態）に移行できる権利、使用ログをとる義務等を示す情報が含まれる。

次に、図９のフローチャートを参照して、図７のクライアント１のライセンス取得処理に対応して実行されるライセンスサーバ４のライセンス提供処理について説明する。なお、この場合においても、図２のクライアント１の構成は、ライセンスサーバ４の構成として引用される。

ステップＳ１０１において、ライセンスサーバ４のCPU２１は、クライアント１よりアクセスを受けるまで待機し、アクセスを受けたとき、ステップＳ１０２に進み、アクセスしてきたクライアント１に対して、ユーザIDとパスワード、並びに、ライセンス指定情報の送信を要求する。上述したようにして、クライアント１から、図７のステップＳ６５の処理で、ユーザIDとパスワード、リーフID並びにライセンス指定情報（ライセンスID）が送信されてきたとき、ライセンスサーバ４のCPU２１は、通信部２９を介してこれを受信し、取り込む処理を実行する。

そして、ライセンスサーバ４のCPU２１は、ステップＳ１０３において、通信部２９から課金サーバ５にアクセスし、ユーザIDとパスワードに対応するユーザの与信処理を要求する。課金サーバ５は、インターネット２を介してライセンスサーバ４から与信処理の要求を受けると、そのユーザIDとパスワードに対応するユーザの過去の支払い履歴などを調査し、そのユーザが、過去にライセンスの対価の不払いの実績があるか否かなどを調べ、そのような実績がない場合には、ライセンスの付与を許容する与信結果を送信し、不払いの実績などがある場合には、ライセンス付与の不許可の与信結果を送信する。

ステップＳ１０４において、ライセンスサーバ４のCPU２１は、課金サーバ５からの与信結果が、ライセンスを付与することを許容する与信結果であるか否かを判定し、ライセンスの付与が許容されている場合には、ステップＳ１０５に進み、ステップＳ１０２の処理で取り込まれたライセンス指定情報に対応するライセンスを、記憶部２８に記憶されているライセンスの中から取り出す。記憶部２８に記憶されているライセンスは、あらかじめライセンスID、バージョン、作成日時、有効期限等の情報が記述されている。ステップＳ１０６において、CPU２１は、そのライセンスに受信したリーフIDを付加する。さらに、ステップＳ１０７において、CPU２１は、ステップＳ１０５で選択されたライセンスに対応づけられている使用条件を選択する。あるいはまた、ステップＳ１０２の処理で、ユーザから使用条件が指定された場合には、その使用条件が必要に応じて、予め用意されている使用条件に付加される。CPU２１は、選択された使用条件をライセンスに付加する。

ステップＳ１０８において、CPU２１はライセンスサーバの秘密鍵によりライセンスに署名し、これにより、図８に示されるような構成のライセンスが生成される。

次に、ステップＳ１０９に進み、ライセンスサーバ４のCPU２１は、そのライセンス（図８に示される構成を有する）を、通信部２９からインターネット２を介してクライアント１に送信させる。

ステップＳ１１０においてライセンスサーバ４のCPU２１は、ステップＳ１０９の処理で、いま送信したライセンス（使用条件、リーフIDを含む）を、ステップＳ１０２の処理で取り込まれたユーザIDとパスワードに対応して、記憶部２８に記憶させる。さらに、ステップＳ１１１において、CPU２１は、課金処理を実行する。具体的には、CPU２１は、通信部２９から課金サーバ５に、そのユーザIDとパスワードに対応するユーザに対する課金処理を要求する。課金サーバ５は、この課金の要求に基づいて、そのユーザに対する課金処理を実行する。上述したように、この課金処理に対して、そのユーザが支払いを行わなかったような場合には、以後、そのユーザは、ライセンスの付与を要求したとしても、ライセンスを受けることができないことになる。

すなわち、この場合には、課金サーバ５からライセンスの付与を不許可とする与信結果が送信されてくるので、ステップＳ１０４からステップＳ１１２に進み、CPU２１は、エラー処理を実行する。具体的には、ライセンスサーバ４のCPU２１は、通信部２９を制御してアクセスしてきたクライアント１に対して、ライセンスを付与することができない旨のメッセージを出力し、処理を終了させる。

この場合、上述したように、そのクライアント１はライセンスを受けることができないので、そのコンテンツを利用すること（暗号を復号すること）ができないことになる。

図１０は、図６のステップＳ４５におけるライセンス更新処理の詳細を表している。図１０のステップＳ１３１乃至ステップＳ１３５の処理は、図７のステップＳ６１乃至ステップＳ６５の処理と基本的に同様の処理である。ただし、ステップＳ１３３において、CPU２１は、購入するライセンスではなく、更新するライセンスのライセンスIDを取り込む。そして、ステップＳ１３５において、CPU２１は、ユーザIDとパスワードとともに、更新するライセンスのライセンスIDを、ライセンスサーバ４に送信する。

ステップＳ１３５の送信処理に対応して、ライセンスサーバ４は、後述するように、使用条件を提示してくる（図１１のステップＳ１５３）。そこで、クライ

アント１のCPU２１は、ステップＳ１３６において、ライセンスサーバ４からの使用条件の提示を受信し、これを出力部２７に出力し、表示させる。ユーザは、入力部２６を操作して、この使用条件の中から所定の使用条件を選択したり、所定の使用条件を新たに追加したりする。ステップＳ１３７でCPU２１は、以上のようにして選択された使用条件（ライセンスを更新する条件）を購入するための申し込みをライセンスサーバ４に送信する。この申し込みに対応して、後述するようにライセンスサーバ４は、最終的な使用条件を送信してくる（図１１のステップＳ１５４）。そこで、ステップＳ１３８において、クライアント１のCPU２１は、ライセンスサーバ４からの使用条件を取得し、ステップＳ１３９において、その使用条件を記憶部２８にすでに記憶されている対応するライセンスの使用条件として更新する。

図１１は、以上のクライアント１のライセンス更新処理に対応して、ライセンスサーバ４が実行するライセンス更新処理を表している。

最初に、ステップＳ１５１において、ライセンスサーバ４のCPU２１は、クライアント１からのアクセスを受けると、ステップＳ１５２において、クライアント１がステップＳ１３５で送信したライセンス指定情報をライセンス更新要求情報とともに受信する。

ステップＳ１５３において、CPU２１は、ライセンスの更新要求を受信すると、そのライセンスに対応する使用条件（更新する使用条件）を、記憶部２８から読み出し、クライアント１に送信する。

この提示に対して、上述したように、クライアント１から使用条件の購入が図１０のステップＳ１３７の処理で申し込まれると、ステップＳ１５４において、ライセンスサーバ４のCPU２１は、申し込まれた使用条件に対応するデータを生成し、ステップＳ１５４において、クライアントと１に送信する。クライアント１は、上述したように、ステップＳ１３９の処理で受信した使用条件を用いて、すでに登録されているライセンスの使用条件を更新する。

本発明においては、図１２に示されるように、ブロードキャストインクリプション（Broadcast Encryption）方式の原理に基づいて、デバイスとライセンスのキーが管理される（特開 2001-352321 号公報参照）。キーは、階層ツリー構造とされ、最下段のリーフ（leaf）が個々のデバイスのキーに対応する。図１２の例の場合、番号０から番号１５までの１６個のデバイスまたはライセンスに対応するキーが生成される。

各キーは、図中丸印で示されるツリー構造の各ノードに対応して規定される。この例では、最上段のルートノードに対応してルートキーKR が、２段目のノードに対応してキーＫ０，Ｋ１が、３段目のノードに対応してキーＫ００乃至Ｋ１１が、第４段目のノードに対応してキーＫ０００乃至キーＫ１１１が、それぞれ対応されている。そして、最下段のノードとしてのリーフ（デバイスノード）に、キーＫ００００乃至Ｋ１１１１が、それぞれ対応されている。

階層構造とされているため、例えば、キーＫ００１０とキー００１１の上位のキーは、Ｋ００１とされ、キーＫ０００とキーＫ００１の上位のキーは、Ｋ００とされている。以下同様に、キーＫ００とキーＫ０１の上位のキーは、Ｋ０とされ、キーＫ０とキーＫ１の上位のキーは、KR とされている。

コンテンツを利用するキーは、最下段のリーフから、最上段のルートノードまでの１つのパスの各ノードに対応するキーで管理される。例えば、番号３のノード（リーフ ID）に対応するライセンスに基づき、コンテンツを利用するキーは、キーＫ００１１，Ｋ００１，Ｋ００，Ｋ０，KR を含むパスの各キーで管理される。

本発明のシステムにおいては、図１３に示されるように、図１２の原理に基づいて構成されるキーシステムで、デバイスのキーとライセンスのキーの管理が行われる。図１３の例では、８＋２４＋３２段のノードがツリー構造とされ、ルートノードから下位の８段までの各ノードにカテゴリが対応される。ここにおけるカテゴリとは、例えばメモリスティックなどの半導体メモリを使用する機器のカテゴリ、デジタル放送を受信する機器のカテゴリといったカテゴリを意味する。

そして、このカテゴリノードのうちの１つのノードに、ライセンスを管理するシステムとして本システム（Ｔシステムと称する）が対応する。

すなわち、このＴシステムのノードよりさらに下の階層の２４段のノードに対応するキーにより、ライセンスが対応される。この例の場合、これにより、２の24乗（約１６メガ）のライセンスを規定することができる。さらに、最も下側の３２段の階層により、２の32乗（約４ギガ）のユーザ（あるいはクライアント１）を規定することができる。最下段の３２段のノードに対応するリーフからルートノードまでのパスの各ノードに対応するキーが、DNK（Device Node Key）を構成し、最下段のリーフに対応するIDがリーフIDとされる。

各デバイスやライセンスのキーは、６４（＝８＋２４＋３２）段の各ノードで構成されるパスの内の１つに対応される。例えば、コンテンツを暗号化したコンテンツキーは、対応するライセンスに割り当てられたパスを構成するノードに対応するキーを用いて暗号化される。上位の階層のキーは、その直近の下位の階層のキーを用いて暗号化され、ＥＫＢ（図１５Ａおよび図１５Ｂを参照して後述する）内に配置される。ＤＮＫは、ＥＫＢ内には配置されず、サービスデータに記述され、ユーザのクライアント１に与えられる。クライアント１は、サービスデータに記述されているＤＮＫを用いて、コンテンツデータとともに配布されるＥＫＢ（図１５Ａおよび図１５Ｂ）内に記述されている直近の上位の階層のキーを復号し、復号して得たキーを用いて、ＥＫＢ内に記述されているさらにその上の階層のキーを復号する。以上の処理を順次行うことで、クライアント１は、そのパスに属するすべてのキーを得ることができる。

図１４に階層ツリー構造のカテゴリの分類の具体的な例を示す。図１４において、階層ツリー構造の最上段には、ルートキーKR２３０１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。

最上段から第M段目（図１３の例では、M＝８）の所定のノードがカテゴリノード２３０４として設定される。すなわち第M段目のノードの各々が特定カテゴリのデバイス設定ノードとされる。第M段の１つのノードを頂点として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される。

例えば図１４の第M段目の１つのノード２３０５にはカテゴリ［メモリスティック（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以下が、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される。

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。図１４の例では、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２３０６が設定されている。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が設定されている。

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。

このように、１つのノードを頂点として、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。

例えば、図１２に示されるツリー構造において、１つのグループに含まれる４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、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。

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）０

０１,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。

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１５Ａ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、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１２に示されるようなツリー構造を構成する各リーフ（最下段のノード）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。

図１５Ａに示す有効化キーブロック（ＥＫＢ）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１５Ａの例は、図１２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１２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。

図１５Ａ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。図１５Ａの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Ｋ００１０によってこの暗号化キーを復号し、更新ノードキーＫ（ｔ）００１を得ることができる。また、復号により得た更新ノードキーＫ（ｔ）００１を用いて、図１５Ａの下から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００１，Ｋ（ｔ）００）が復号可能となり、更新ノードキーＫ（ｔ）００を得ることができる。

以下順次、図１５Ａの上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号することで、更新ノードキーＫ（ｔ）０が得られ、これを用いて、図１５Ａの上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号することで、更新ルートキーＫ（ｔ）Ｒが得られる。

一方、ノードキーＫ０００は更新する対象に含まれておらず、ノード０，１が、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。ノード０，１は、デバイスノードキーに含まれるノードキーＫ０００を用いて、図１５Ａの上から３段目の暗号化キーＥｎｃ（Ｋ０００，Ｋ（ｔ）００）を復号することで更新ノードキーＫ（ｔ）００を取得し、以下順次、図１５Ａの上から２段目の暗号化キーＥｎｃ（Ｋ（ｔ）００，Ｋ（ｔ）０）を復号することで、更新ノードキーＫ（ｔ）０を得、図１５Ａの上から１段目の暗号化キーＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）を復号することで、更新ルートキーＫ（ｔ）Ｒを得る。このようにして、デバイス０，１，２は更新したキーＫ（ｔ）Ｒを得ることができる。

なお、図１５Ａのインデックスは、図の右側の暗号化キーを復号するための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。

図１２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１５Ｂ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。

図１５Ｂ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１２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）００，Ｋ（ｔ）ｃｏｎ）が、図１５Ｂに示されるＥＫＢとともに配布される。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が復号することができないデータとしての配布が可能となる。

すなわち、デバイス０，１，２はＥＫＢを処理して得たキーＫ（ｔ）００を用いて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。

図１６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１５Ｂ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。すなわちこの例は、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。

図１６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DNKに含まれるノードキーＫ０００を用いて、上述した場合と同様のＥＫＢ処理により、ノードキーＫ（ｔ）００を生成する。さらに、デバイス０は、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。

図１７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６０６の位置を示すポインタであり、タグポインタ６０４はタグ部６０７の位置、署名ポインタ６０５は署名６０８の位置を示すポインタである。

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１６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。

タグ部６０７は、データ部６０６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１８を用いて説明する。

図１８では、データとして先に図１５Ａ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１８のテーブルに示すようになる。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。この例の場合は、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１８に示す階層ツリーに示す位置Ｐ０に対応する。次の段のデータは、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Ｐ００に対応する。ツリー構造の所定の位置から見て、その下に、データがある場合は、タグが０、ない場合はタグが１に設定される。タグは｛左（Ｌ）タグ，右（Ｒ）タグ｝として設定される。図１８のテーブルの最上段のデータＥｎｃ（Ｋ（ｔ）０，Ｋ（ｔ）Ｒ）に対応する位置POの左下の位置POO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１８に示すデータ列、およびタグ列が構成される。

タグは、対応するデータ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ものである。データ部６０６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が、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が判別可能となる。上述したタグを用いずに、先の図１５Ａおよび図１５Ｂ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）Ｒ）

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。

図１７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）６０８は、有効化キーブロック（ＥＫＢ）を発行した例えば鍵管理センタ（ライセンスサーバ４）、コンテンツロバイダ（コンテンツサーバ３）、決済機関（課金サーバ５）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。

以上のようにして、ライセンスサーバ４から供給されたライセンスに基づいて、コンテンツサーバ３から供給されたコンテンツを利用する処理をまとめると、図１９に示されるようになる。

すなわち、コンテンツサーバ３からクライアント１に対してコンテンツが提供されるとともに、ライセンスサーバ４からクライアント１にライセンスが供給される。コンテンツは、コンテンツキーKc により、暗号化されており（Enc（Kc, Content））、コンテンツキーKc は、ルートキーKR(EKB から得られるキーであって、図５におけるキーK_{EKBC}に対応する)で暗号化され（Enc（KR, Kc））、EKB とともに、暗号化されたコンテンツに付加されてクライアント１に提供される。

図１９の例における EKB には、例えば、図２０に示されるように、DNK で復号可能なルートキーKR が含まれている（Enc（DNK, KR））。従って、クライアント１は、サービスデータに含まれる DNK を利用して、EKB からルートキーKR を得ることができる。さらに、ルートキーKR を用いて、Enc（KR, Kc）からコンテンツキーKc を復号することができ、コンテンツキーKc を用いて、Enc（Kc, Content）からコンテンツを復号することができる。

このように、クライアント１にDNKを個別に割り当てることにより、図１２、並びに図１５Ａおよび図１５Ｂを参照して説明した原理に従って、個々のクライアント１のリボーク（revoke）が可能になる。

また、ライセンスにリーフIDを付加して配布することにより、クライアント１において、サービスデータとライセンスの対応付けが行われることになり、ライセンスの不正コピーを防止することが可能になる。

また、クライアント用の証明書と秘密鍵をサービスデータとして配信するようにすることで、エンドユーザも、これらを用いて不正コピーを防止可能なコンテンツを作成することが可能になる。

証明書と秘密鍵の利用については、図２９のフローチャートを参照して後述する。

本発明においては、図１３を参照して説明したように、カテゴリノードにライセンスを管理する本発明のコンテンツ配信システムと、各種のコンテンツを利用するデバイスのカテゴリが対応づけられるので、複数のDNKを同一のデバイスに持たせることができる。その結果、異なるカテゴリのコンテンツを１つのデバイスで管理することが可能となる。

図２１は、この関係を表している。すなわち、デバイスＤ１には、コンテンツ配信システムに基づいて、DNK１が割り当てられている、コンテンツ１を利用するライセンス及びサービスデータが記録される。同様に、このデバイスＤ１には、例えば、DNK２が割り当てられた、メモリスティックにCDからリッピングしたコンテンツ２を記録することができる。この場合、デバイスＤ１は、コンテンツ１とコンテンツ２という、異なるシステム（コンテンツ配信システムとデバイス管理システム）により配信されたコンテンツを同時に扱うことが可能となる。新たなDNKを割り当てるとき、既に割り当てられているDNKを削除するなどして、デバイスに１個のDNKだけを対応させるようにした場合、このようなことはできない。

また、図１３における、例えば、下側の３２階層の各三角形の１つ１つに、図２２に示されるライセンスカテゴリ１とライセンスカテゴリ２を割り当てることにより、同一のカテゴリ内を、サブカテゴリを利用して、コンテンツのジャンル、レーベル、販売店、配信サービス、コンテンツの出所、提供方法等の小さな集まりに分類して、管理することが可能となる。

図２２の例においては、例えば、ライセンスカテゴリ１は、ジャズのジャンルに属し、ライセンスカテゴリ２は、ロックのジャンルに属する。ライセンスカテゴリ１には、ライセンス ID が１であるコンテンツ１とコンテンツ２を対応させ、それぞれユーザ１乃至ユーザ３に配布されている。ライセンスカテゴリ２は、ライセンス ID２のコンテンツ３、コンテンツ４、およびコンテンツ５が含まれ、それぞれユーザ１とユーザ３に提供されている。

このように、本発明においては、カテゴリ毎に独立したキー管理が可能になる。

また、DNK を、機器やメディアに予め埋め込むのではなく、ライセンスサーバ４により、登録処理を行う際に、各機器やメディアにダウンロードするようにすることで、ユーザによるキーの取得が可能なシステムを実現することができる。

この場合のクライアント１の登録処理について、図２３を参照して説明する。

ステップＳ１６１において、クライアント１のＣＰＵ２１は通信部２９を制御してライセンスサーバ４にサービスデータ要求を送信する。ライセンスサーバ４のＣＰＵ２１は、ステップＳ１６５において、通信部２９を介して入力されるサービスデータ要求を受信すると、Ｓ１６６において、通信部２９を介してユーザ情報要求をクライアント１に送信する。

クライアント１のＣＰＵ２１は、ステップＳ１６２において、通信部２９を介してユーザ情報要求を受信すると、出力部２７を制御しディスプレイなどにユーザ情報の入力を促すメッセージを表示させる。ユーザがキーボードなどを操作することにより、入力部２６からユーザ本人の個人情報や決済情報等のユーザ情報を入力すると、Ｓ１６３においてクライアント１のＣＰＵ２１は、入力されたユーザ情報を、通信部２９を介してライセンスサーバ４に送信する。

ライセンスサーバ４のＣＰＵ２１は、ステップＳ１６７において、通信部２９を介してユーザ情報を受信すると、ステップＳ１６８において、そのライセンスサーバ４に割り当てられたカテゴリのノード以下のリーフのうち、まだ割り当てられていないリーフをクライアント１に割り当て、そのリーフからライセンスサーバ４に割り当てられたカテゴリのノードまでのパス上のノードに割り当てられたノードキーの組をデバイスノードキーとして生成し、生成されたデバイスノードキー、クライアント１に割り当てられたリーフのリーフID、クライアント１の秘密鍵、クライアント１の秘密鍵・公開鍵のペア、ライセンスサーバの公開鍵、及び各公開鍵の証明書をまとめてサービスデータとして生成し、Ｓ１６９において通信部２９を介してクライアントに生成されたサービスデータを送信すると共に、ドライブ３０を制御してユーザ情報をリーフIDと対応付けてハードディスク等の記録メディアに記録させる。

クライアント１のＣＰＵ２１は、ステップＳ１６４において、通信部２９を介してサービスデータを受信すると、暗号化復号部２４を制御して受信したサービスデータを暗号化し、ドライブ３０を制御してハードディスク等の記録メディアに記録させる。

以上のようにして、ライセンスサーバ４はクライアント１及びそのユーザを登録し、クライアント１は所望のコンテンツ配信サービスを利用するために必要な、デバイスノードキーを含むサービスデータを受け取ることができる。

コンテンツは、それが作成された後、どのような使われ方をされようとも、その使われ方に関わりなく、全ての用途において、使用可能であるのが望ましい。例えば、異なるコンテンツ配信サービス、あるいはドメインの使用状況が異なる場合においても、同一のコンテンツが使えることが望ましい。本発明においては、このため、上述したように、各ユーザ（クライアント１）に、認証局としてのライセンスサーバ４から秘密鍵と、それに対応する公開鍵の証明書（certificates）が配布される。各ユーザは、その秘密鍵を用いて、署名

（signature）を作成し、コンテンツに付加して、コンテンツの真正さ(integrity)を保証し、かつコンテンツの改竄防止を図ることができる。

この場合の処理の例について、図２４のフローチャートを参照して説明する。図２４の処理は、ユーザがCDから再生したデータを記憶部２８に記憶させるリッピング処理を説明するものである。

最初に、ステップＳ１７１において、クライアント１のCPU２１は、通信部２９を介して入力されるCDの再生データを記録データとして取り込む。ステップＳ１７２において、CPU２１は、ステップＳ１７１の処理で取り込まれた記録データにウォーターマークが含まれているか否かを判定する。このウォーターマークは、３ビットのコピー管理情報（CCI）と、１ビットのトリガ（Trigger）とにより構成され、コンテンツのデータの中に埋め込まれている。CPU２１は、ウォーターマークが検出された場合には、ステップＳ１７３に進み、そのウォーターマークを抽出する処理を実行する。ウォーターマークが存在しない場合には、ステップＳ１７３の処理はスキップされる。

次に、ステップＳ１７４において、CPU２１は、コンテンツに対応して記録するヘッダのデータを作成する。このヘッダのデータは、コンテンツID、ライセンスID、ライセンスを取得するためのアクセス先を表すURL、およびウォーターマークに含まれていたコピー管理情報（CCI）と、トリガ（Trigger）により構成される。

次に、ステップＳ１７５に進み、CPU２１は、ステップＳ１７４の処理で作成したヘッダのデータに基づいたデジタル署名を、自分自身の秘密鍵を用いて作成する。この秘密鍵は、ライセンスサーバ４から取得したものである（図７のステップＳ６７）。

ステップＳ１７６で、CPU２１は、暗号化復号部２４を制御し、コンテンツキーでコンテンツを暗号化させる。コンテンツキーは、乱数等を用いて生成される。

次に、ステップＳ１７７において、CPU２１は、ファイルフォーマットに基づき、データを、例えば、ミニディスク等により構成される光磁気ディスク４３に記録させる。

なお、記録媒体がミニディスクである場合、ステップＳ１７６において、CPU２１は、コンテンツをコーデック部２５に供給し、例えば、ATRAC３方式によりコンテンツを符号化させる。そして、符号化されたデータが暗号化復号部２４によりさらに暗号化される。

図２５は、以上のようにして、記録媒体にコンテンツが記録された状態を模式的に表している。暗号化されているコンテンツ（E（At３））から抽出されたウォーターマーク（WM）が、コンテンツの外（ヘッダ）に記録されている。

図２６は、コンテンツを記録媒体に記録する場合のファイルフォーマットのより詳細な構成を表している。この例においては、コンテンツID（CID）、ライセンスID（LID）、URL、およびウォーターマーク（WM）を含むヘッダが記録されている他、EKB、コンテンツキーKcをルートキーKRで暗号化したデータ（Enc（KR, Kc））、証明書（Cert）、ヘッダに基づき生成されたデジタル署名（Sig（Header））、コンテンツをコンテンツキーKcで暗号化したデータ（Enc（Kc, Content））、メタデータ（Meta Data）およびマーク(Mark)が記録されている。

ウォーターマークは、コンテンツの内部に埋め込まれているものであるが、図２５と図２６に示されるように、コンテンツの内部とは別に、ヘッダ内に配置するようにすることで、ウォーターマークとしてコンテンツに埋め込まれている情報を迅速に、かつ簡単に検出することが可能となる。従って、そのコンテンツを、コピーすることができるか否かを、迅速に判定することができる。

なお、メタデータは、例えば、ジャケット、写真、歌詞等のデータを表す。マークについては、図３２を参照して後述する。

図２７は、証明書としての公開鍵証明書の例を表している。公開鍵証明書は、通常、公開鍵暗号方式における認証局（CA：Certificate Authority）が発行す

る証明書であり、ユーザが、認証局に提出した自己のIDや公開鍵などに、認証局が有効期限等の情報を付加し、さらに、認証局によるデジタル署名を付加して作成される。この発明においては、ライセンスサーバ４（またはコンテンツサーバ３）が、証明書と秘密鍵、従って公開鍵も発行するので、ユーザは、ユーザID、パスワード等をライセンスサーバ４に提供し登録処理を行うことによって、この公開鍵証明書を得ることができる。

図２７における公開鍵証明書は、証明書のバージョン番号、ライセンスサーバ４が証明書の利用者（ユーザ）に対して割りつける証明書の通し番号、デジタル署名に用いたアルゴリズム、およびパラメータ、認証局（ライセンスサーバ４）の名前、証明書の有効期限、証明書利用者のID（ノードIDまたはリーフID）、並びに証明書利用者の公開鍵が、メッセージとして含まれている。さらに、このメッセージには、認証局としてのライセンスサーバ４により作成されたデジタル署名が付加されている。このデジタル署名は、メッセージに対してハッシュ関数を適用して生成されたハッシュ値に基づいて、ライセンスサーバ４の秘密鍵を用いて生成されたデータである。

ノードIDまたはリーフIDは、例えば、図１２の例の場合、デバイス０であれば「００００」とされ、デバイス１であれば「０００１」とされ、デバイス１５であれば「１１１１」とされる。このようなIDに基づいて、そのデバイス（エンティティ）がツリー構成のどの位置（リーフまたはノード）に位置するエンティティであるのかが識別される。

このように、コンテンツを利用するのに必要なライセンスを、コンテンツとは分離して配布するようにすることにより、コンテンツの配布が自由に行われることになる。任意の方法、あるいは経路で入手されたコンテンツは、一元的に取り扱うことが可能である。

また、ファイルフォーマットを図２６に示されるように構成することで、そのフォーマットのコンテンツを、インターネットを介して配信する場合は勿論、

SDMI（Secure Digital Music Initiative）機器に提供する場合においても、コンテンツの著作権を管理することが可能となる。

さらに、例えば、図２８に示されるように、コンテンツが記録媒体を介して提供されたとしても、インターネット２を介して提供されたとしても、同様の処理により、SDMI（Secure Digital Music Initiative）機器としての所定のPD（Portable Device）等に、チェックアウトしたりすることが可能となる。

次に、図２９のフローチャートを参照して、クライアント１が他のクライアント（例えば、PD）に対してコンテンツをチェックアウトする場合の処理について説明する。

最初に、ステップＳ１９１において、CPU２１は、コンテンツにデジタル署名が付加されているか否かを判定する。デジタル署名が付加されていると判定された場合、ステップＳ１９２に進み、CPU２１は、証明書を抽出し、認証局（ライセンスサーバ４）の公開鍵で検証する処理を実行する。すなわち、クライアント１は、ライセンスサーバ４からライセンスサーバ４の秘密鍵に対応する公開鍵を取得し、その公開鍵で公開鍵証明書に付加されているデジタル署名を復号する。図２７を参照して説明したように、デジタル署名は、認証局（ライセンスサーバ４）の秘密鍵に基づいて生成されており、ライセンスサーバ４の公開鍵を用いて復号することができる。さらに、CPU２１は、証明書のメッセージ全体に対してハッシュ関数を適用してハッシュ値を演算する。そしてCPU２１は、演算されたハッシュ値と、デジタル署名を復号して得られたハッシュ値とを比較し、両者が一致すれば、メッセージは改竄されたものではないと判定する。両者が一致しない場合には、この証明書は、改竄されたものであるということになる。

そこで、ステップＳ１９３において、CPU２１は、証明書が改竄されていないか否かを判定し、改竄されていないと判定された場合、ステップＳ１９４に進み、証明書をEKBで検証する処理を実行する。この検証処理は、証明書に含まれるリーフID（図２７）に基づいて、EKBをたどることができるか否かを調べることにより行われる。この検証について、図３０と図３１を参照して説明する。

いま、図３０に示されるように、例えば、リーフキーＫ１００１を有するデバイスがリボークされたデバイスであるとする。このとき、図３１に示されるようなデータ（暗号化キー）とタグを有するEKBが、各デバイス（リーフ）に配布される。このEKBは、図３０におけるデバイス「１００１」をリボークするために、キーKR,Ｋ１,Ｋ１０,Ｋ１００を更新するEKBとなっている。

リボークデバイス「１００１」以外の全てのリーフは、更新されたルートキーＫ（ｔ）Ｒを取得することができる。すなわち、ノードキーＫ０の下位に連なるリーフは、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、暗号化キーEnc（Ｋ０，Ｋ（ｔ）Ｒ）を、キーＫ０によって復号することで、更新ルートキーＫ（ｔ）Ｒを取得することができる。

また、ノード１１以下のリーフは、更新されていないノードキーＫ１１を用いて、Enc（Ｋ１１，Ｋ（ｔ）１）をノードキーＫ１１によって復号することで、更新ノードキーＫ（ｔ）１を取得することができる。さらに、Enc（Ｋ（ｔ）１，Ｋ（ｔ）Ｒ）をノードキーＫ（ｔ）１によって復号することで、更新ルートキーＫ（ｔ）Ｒを取得することが可能となる。ノードキーＫ１０１の下位リーフについても、同様に更新ルートキーＫ（ｔ）Ｒを取得することが可能である。

さらに、リボークされていないリーフキーＫ１０００を有するデバイス「１０００」は、自己のリーフキーＫ１０００でEnc（Ｋ１０００，Ｋ（ｔ）１００）を復号して、ノードキーＫ（ｔ）１００を取得することができ、これを用いてさらに、上位のノードキーを順次復号し、更新ルートキーＫ（ｔ）Ｒを取得することができる。

これに対して、リボークされたデバイス「１００１」は、自己のリーフの１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。

リボークされていない正当なデバイス（クライアント１）には、図３１に示されるデータとタグを有するEKBが、ライセンスサーバ４から配信され、格納されている。

そこで、各クライアントは、そのタグを利用して、EKB 追跡処理を行うことができる。この EKB 追跡処理は、上位のルートキーからキー配信ツリーをたどれるか否かを判定する処理である。

例えば、図３０のリーフ「１００１」の ID（リーフ ID）である「１００１」を、「１」「０」「０」「１」の４ビットとして把握し、最上位ビットから順次、下位ビットに従って、ツリーをたどることができるか否かが判定される。この判定では、ビットが１であれば、右側に進み、０であれば、左側に進む処理が行われる。

ID「１００１」の最上位ビットが１であるから、図３０のルートキーKR から右側に進む。EKB の最初のタグ（番号０のタグ）は、０：｛０，０｝であり、両枝にデータを有するものであると判定される。この場合、右側に進むことができるので、ノードキーＫ１にたどり着くことができる。

次に、ノードキーＫ１の下位のノードに進む。ID「１００１」の２番目のビットは０であるから左側に進む。番号１のタグは、左側のノードキーＫ０の下位のデータの有無を表すものであり、ノードキーＫ１の下位のデータの有無を示すタグは、番号２のタグである。このタグは、図３１に示されるように、２：｛０，０｝であり、両枝にデータを有するものとされる。従って、左側に進み、ノードキーＫ１０にたどり着くことができる。

さらに、ID「１００１」の３番目のビットは０であり、左側に進む。このとき、Ｋ１０の下位のデータの有無を示すタグ（番号３のタグ）は、３：｛０，０｝であり、両枝にデータを有するものと判定される。そこで、左側に進み、ノードキーＫ１００にたどり着くことができる。

さらに、ID「１００１」の最下位ビットは１であり、右側に進む。番号４のタグは、ノードキーＫ１１に対応するものであり、Ｋ１００の下位のデータの符号を表すタグは、番号５のタグである。このタグは、５：｛０，１｝である。従って、右側には、データが存在しないことになる。その結果、ノード「１００１」にはたどり着けないことになり、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。

これに対して、例えば、リーフキーＫ１０００を有するデバイスIDは、「１０００」であり、上述した場合と同様に、EKB内のタグに基づくEKB追跡処理を行うと、ノード「１０００」にたどり着くことができる。従って、ID「１０００」のデバイスは、正当なデバイスであると判定される。

図２９に戻って、CPU２１は、ステップＳ１９４の検証処理に基づき、証明書がリボークされていないか否かをステップＳ１９５で判定し、証明書がリボークされていない場合には、ステップＳ１９６に進み、デジタル署名を証明書に含まれる公開鍵で検証する処理を実行する。

すなわち、図２７に示されるように、証明書には、証明書利用者（コンテンツ作成者）の公開鍵が含まれており、この公開鍵を用いて、図２６に示される署名（Sig（Header））が検証される。すなわち、この公開鍵を用いて、デジタル署名Sig（Header）を復号して得られたデータ（ハッシュ値）と、図２６に示されるHeaderにハッシュ関数を適用して演算されたハッシュ値とを比較することで、両者が一致していれば、Headerが改竄されていないことを確認することができる。これに対して、両者が一致しなければ、Headerは改竄されているということになる。

ステップＳ１９７において、CPU２１は、Headerが改竄されているか否かを判定し、改竄されていなければ、ステップＳ１９８に進み、ウォーターマークを検証する。ステップＳ１９９において、CPU２１は、ウォーターマークの検証の結果、チェックアウトが可能であるか否かを判定する。チェックアウトが可能である場合には、ステップＳ２００に進み、CPU２１は、チェックアウトを実行する。すなわち、チェックアウト先のクライアント１に対してコンテンツを転送し、コピーさせる。

ステップＳ１９１において、デジタル署名が存在しないと判定された場合、ステップＳ１９３において、証明書が改竄されていると判定された場合、ステップ

Ｓ１９５において、証明書をEKBで検証することができなかったと判定された場合、ステップＳ１９７において、デジタル署名の検証の結果、ヘッダが改竄されていると判定された場合、または、ステップＳ１９９において、ウォーターマークにチェックアウトの禁止が記述されていると判定された場合、ステップＳ２０１に進み、エラー処理が実行される。すなわち、この場合には、チェックアウトが禁止される。

このように、証明書と秘密鍵をライセンスサーバ４からユーザに配布し、コンテンツ作成時に、デジタル署名を付加することにより、コンテンツの作成者の真正を保証することが可能となる。これにより、不正なコンテンツの流通を抑制することができる。

さらに、ウォーターマークをコンテンツ作成時に検出し、その情報をデジタル署名に付することで、ウォーターマーク情報の改竄を防止し、コンテンツの真正を保証することができる。

その結果、一度作成されたコンテンツは、どのような形態で配信されたとしても、元のコンテンツの真正を保証することが可能となる。

さらに、コンテンツは、使用条件を有さず、使用条件は、ライセンスに付加されているので、ライセンス内の使用条件を変更することで、それに関係するコンテンツの使用条件を一斉に変更することが可能となる。

次に、マークの利用方法について説明する。本発明においては、上述したように、使用条件は、コンテンツではなく、ライセンスに付加される。しかしながら、コンテンツによって、使用状況が異なる場合がある。そこで、本発明においては、図２６に示されるように、コンテンツにマークが付加される。

ライセンスとコンテンツは、１対多の関係にあるため、コンテンツの個々の使用状況をライセンスの使用条件にのみ記述するのは困難となる。そこで、このように、コンテンツに使用状況を付加することにより、ライセンスでの管理をしながらも、個々のコンテンツを管理することが可能となる。

このマークには、例えば、図３２に示されるように、ユーザのID（リーフID）、所有権フラグ、使用開始時刻、およびコピー回数等が記述される。

さらに、マークには、リーフID、所有権フラグ、使用開始時刻、およびコピー回数等のメッセージに基づいて生成されたデジタル署名が付加される。

所有権フラグは、例えば、所定の期間だけコンテンツを使用可能とするライセンスを、そのまま買い取ったような場合（使用期間を永久に変更したような場合）に付加される。使用開始時刻は、コンテンツの使用を所定の期間内に開始した場合に記述される。例えば、コンテンツをダウンロードする時期が制限されているような場合において、その期限内にダウンロードが行われたようなとき、その実際にコンテンツをダウンロードした日時がここに記述される。これにより、期間内での有効な使用であることが、証明される。

コピー回数には、それまでにそのコンテンツをコピーした回数が履歴（ログ）として記述される。

次に、図３３のフローチャートを参照して、ユーザがライセンスを買い取った場合に、マークを付加する処理について、マークをコンテンツに付加する例として説明する。

最初に、ステップＳ２２１において、CPU２１は、入力部２６からのユーザの指令に基づいて、インターネット２を介して、ライセンスサーバ４にアクセスする。

ステップＳ２２２において、CPU２１は、ユーザからの入力部２６を介しての入力を取り込み、その入力に対応してライセンスサーバ４に対してライセンスの買い取りを要求する。

この要求に対応して、図３４のフローチャートを参照して後述するように、ライセンスサーバ４は、ライセンスを買い取るために必要な対価を提示してくる（図３４のステップＳ２４２）。そこで、ステップＳ２２３において、クライアント１のCPU２１は、ライセンスサーバ４からの対価の提示を受け取ると、これを出力部２７に出力し、表示させる。

ユーザは、この表示に基づいて、提示された対価を了承するか否かを判断し、その判断結果に基づいて、入力部２６からその判断結果を入力する。

CPU２１は、ステップＳ２２４において、入力部２６からの入力に基づいて、ユーザが提示された対価を了承したか否かを判定し、了承したと判定した場合には、ステップＳ２２５に進み、ライセンスサーバ４に了承を通知する処理を実行する。

この了承通知を受信すると、ライセンスサーバ４は、対価の買い取りを表す情報、すなわち所有権フラグを記述したマークを送信してくる（図３４のステップＳ２４４）。そこで、ステップＳ２２６において、クライアント１のCPU２１は、ライセンスサーバ４からのマークを受け取ると、ステップＳ２２７において、受け取ったマークをコンテンツに埋め込む処理を実行する。すなわち、これにより、買い取られたライセンスに対応するコンテンツのマークとして、図３２に示されるような所有権フラグが記述されたマークがコンテンツに対応して記録されることになる。また、このとき、CPU２１は、メッセージが更新されたことになるので、デジタル署名（図２６）も更新し、記録媒体に記録する。

ステップＳ２２４において、ライセンスサーバ４から提示された対価が了承されていないと判定された場合、ステップＳ２２８に進み、CPU２１は、提示された対価を了承しないことをライセンスサーバ４に通知する。

このようなクライアント１の処理に対応して、ライセンスサーバ４は、図３４のフローチャートに示す処理を実行する。

すなわち、最初に、ステップＳ２４１において、ライセンスサーバ４のCPU２１は、クライアント１からライセンス買い取りの要求が送信されてくると（図３３のステップＳ２２２）、これを受け取り、ステップＳ２４２において、対象とされているライセンスの買い取りに必要な対価を記憶部２８から読み出し、これをクライアント１に送信する。

上述したように、このようにして提示された対価に対して、クライアント１から提示された対価を了承するか否かの通知が送信されてくる。

そこで、ステップＳ２４３において、ライセンスサーバ４のCPU２１は、クライアント１から了承通知を受信したか否かを判定し、了承通知を受信したと判定した場合、ステップＳ２４４に進み、対象とされるライセンスの買い取りを表すメッセージを含むマークを生成し、自分自身の秘密鍵で、デジタル署名を付加して、クライアント１に送信する。このようにして送信されたマークは、上述したように、クライアント１の記憶部２８において、対応するコンテンツに記録される（図３３のステップＳ２２７）。

ステップＳ２４３において、クライアント１から了承通知が受信されていないと判定された場合、ステップＳ２４４の処理はスキップされる。すなわち、この場合には、ライセンスの買い取り処理が最終的に行われなかったことになるので、マークは送信されない。

図３５は、ステップＳ２４４において、ライセンスサーバ４からクライアント１に対して送信されるマークの構成例を表している。この例においては、そのユーザのリーフ ID、所有権フラグ（Own）、並びにリーフ ID と所有権フラグを、ライセンスサーバ４の秘密鍵Ｓに基づいて生成されたデジタル署名 Sigs（LeafID,Own）により、マークが構成されている。

なお、このマークは、特定のユーザの特定のコンテンツに対してのみ有効なものであるので、対象とされるコンテンツがコピーされた場合には、そのコピーされたコンテンツに付随するマークは無効とされる。

このようにして、コンテンツとライセンスを分離し、使用条件をライセンスに対応させる場合においても、個々のコンテンツの使用状況に応じたサービスを実現することが可能となる。

次に、グルーピングについて説明する。複数の機器やメディアを適当に集め、その１つの集合内においては、コンテンツを自由に授受することができるようにすることは、グルーピングと称される。通常、このグルーピングは、個人の所有する機器やメディアにおいて行われる。このグルーピングは、従来、グループ毎にグループキーを設定する等して行われていたが、グループ化する複数の機器や

メディアに、同一のライセンスを対応づけることにより、容易にグルーピングすることが可能となる。

また、各機器を予め登録しておくことで、グルーピングすることも可能である。この場合のグルーピングについて、以下に説明する。

この場合、ユーザは、グルーピング対象とされる機器の証明書を予めサーバに登録しておく必要がある。この証明書の登録処理について、図３６と図３７のフローチャートを参照して説明する。

最初に、図３６のフローチャートを参照して、クライアント（グルーピング対象となる機器）の証明書の登録処理について説明する。ステップＳ２６１において、クライアント１のCPU２１は、グルーピングの対象とされる機器としての自分自身の証明書を作成する。この証明書には、自分自身の公開鍵が含まれる。

次に、ステップＳ２６２に進み、CPU２１は、ユーザの入力部２６からの入力に基づいて、コンテンツサーバ３にアクセスし、ステップＳ２６３において、ステップＳ２６１の処理で作成された証明書をコンテンツサーバ３に送信する処理を実行する。

なお、証明書としては、ライセンスサーバ４から受信したものを、そのまま使用することもできる。

以上の処理は、グルーピング対象とされる全ての機器が行う。

次に、図３７のフローチャートを参照して、図３６のクライアント１の証明書の登録処理に対応して行われるコンテンツサーバ３の証明書の登録処理について説明する。

最初に、ステップＳ２７１において、コンテンツサーバ３のCPU２１は、クライアント１から送信されてきた証明書を受信すると、ステップＳ２７２において、その証明書を記憶部２８に登録する。

以上の処理が、グループ対象とされる機器毎に行われる。その結果、コンテンツサーバ３の記憶部２８には、例えば、図３８に示されるように、グループ毎に、そのグループを構成するデバイスの証明書が登録される。

図３８に示される例では、グループ１の証明書として、証明書Ｃ１１乃至Ｃ１４が登録されている。これらの証明書Ｃ１１乃至Ｃ１４には、対応する公開鍵K_{P11}乃至K_{P14}が含まれている。

同様に、グループ２の証明書として、証明書Ｃ２１乃至Ｃ２３が登録されており、これらは対応する公開鍵K_{P21}乃至K_{P23}が含まれている。

以上のようなグループを構成する各機器毎に、その証明書が登録された状態において、ユーザからそのグループに属する機器にコンテンツの提供が要求されると、コンテンツサーバ３は、図３９のフローチャートに示す処理を実行する。

最初に、ステップＳ２８１において、コンテンツサーバ３のCPU２１は、記憶部２８に記憶されている証明書のうち、そのグループに属する証明書を検証する処理を実行する。

この検証処理は、図３０と図３１を参照して説明されたように、各機器の証明書に含まれるリーフIDに基づいて、タグを利用してEKBをたどることで行われる。EKBは、コンテンツサーバ３にも、ライセンスサーバ４から配布されている。この検証処理により、リボークされている証明書は除外される。

ステップＳ２８２において、コンテンツサーバ３のCPU２１は、ステップＳ２８１の検証処理の結果、有効とされた証明書を選択する。そして、ステップＳ２８３において、CPU２１は、ステップＳ２８２の処理で選択された各機器の証明書の各公開鍵でコンテンツ鍵を暗号化する。ステップＳ２８４において、CPU２１は、対象とされるグループの各機器に、ステップＳ２８３の処理で暗号化されたコンテンツ鍵をコンテンツとともに送信する。

図３８に示されるグループ１のうち、例えば、証明書Ｃ１４がリボークされているとすると、ステップＳ２８３の処理で、例えば、図４０に示されるような暗号化データが生成される。

すなわち、図４０の例においては、コンテンツ鍵Kcが、証明書Ｃ１１の公開鍵K_{P11}、証明書Ｃ１２の公開鍵K_{P12}、または証明書Ｃ１３の公開鍵K_{P13}により、暗号化されている。

コンテンツサーバ３の図３９に示されるような処理に対応して、コンテンツの提供を受ける各グループの機器（クライアント）は、図４１のフローチャートに示す処理を実行する。

最初に、ステップＳ２９１において、クライアント１のCPU２１は、コンテンツサーバ３が図３９のステップＳ２８４の処理で送信してきたコンテンツを、コンテンツ鍵とともに受信する。コンテンツは、コンテンツ鍵Kcにより、暗号化されており、コンテンツ鍵は上述したように、各機器が保持する公開鍵により暗号化されている（図４０）。

そこで、ステップＳ２９２において、CPU２１は、ステップＳ２９１の処理で受信した自分宛のコンテンツ鍵を、自分自身の秘密鍵で復号し、取得する。そして、取得したコンテンツ鍵を用いてコンテンツの復号処理が行われる。

例えば、図４０の例に示される証明書Ｃ１１に対応する機器は、公開鍵K_{P11}に対応する自分自身の秘密鍵を用いて、コンテンツ鍵Kcの暗号を復号し、コンテンツ鍵Kcを取得する。そして、コンテンツ鍵Kcを用いて、コンテンツがさらに復号される。

同様の処理は、証明書Ｃ１２，Ｃ１３に対応する機器においても行われる。リボークされている証明書Ｃ１４の機器は、自分自身の公開鍵を用いて暗号化されたコンテンツ鍵Kcがコンテンツに付随して送られてこないので、コンテンツ鍵Kcを復号することができず、従って、コンテンツ鍵Kcを用いてコンテンツを復号することができない。

以上においては、コンテンツキー（すなわちコンテンツ）に対してグルーピングを行うようにしたが、ライセンスキー（ライセンス）に対してグルーピングを行うことも可能である。

以上のようにして、特別なグループキーや、後述するICV（Integrity Check Value）を用いずにグループ化が可能となる。このグループ化は、小規模のグループに適用するのに向いている。

本発明においては、ライセンスもチェックアウト、あるいはチェックインしたり、ムーブしたり、コピーしたりすることが可能とされる。但し、これらの処理はSDMIで定められたルールに基づいて行われる。

次に、図４２と図４３のフローチャートを参照して、このようなクライアントによるライセンスのチェックアウト処理について説明する。

最初に、図４２のフローチャートを参照して他のクライアントにライセンスをチェックアウトするクライアントの処理について説明する。最初に、ステップＳ３０１において、クライアント１のCPU２１は、チェックアウト対象のライセンスのチェックアウト回数Ｎ１を読み取る。このチェックアウト回数は、図８に示される使用条件に書き込まれているので、この使用条件から読み取られる。

次に、ステップＳ３０２において、CPU２１は、チェックアウト対象のライセンスの最大チェックアウト回数Ｎ２を、やはりライセンスの使用条件から読み取る。

そして、ステップＳ３０３において、CPU２１は、ステップＳ３０１の処理で読み取られたチェックアウト回数Ｎ１と、ステップＳ３０２の処理で読み取られた最大チェックアウト回数Ｎ２とを比較し、チェックアウト回数Ｎ１が最大チェックアウト回数Ｎ２より小さいか否かを判定する。

チェックアウト回数Ｎ１が、最大チェックアウト回数Ｎ２より小さいと判定された場合、ステップＳ３０４に進み、CPU２１は、相手側の装置（チェックアウト先のクライアント）のリーフキーを相手個々の装置から取得し、そのリーフキーを、いまチェックアウト対象とされているライセンスIDに対応して記憶部２８のチェックアウトリストに記憶させる。

次に、ステップＳ３０５において、CPU２１は、ステップＳ３０１の処理で読み取られたライセンスのチェックアウト回数Ｎ１の値を１だけインクリメントする。ステップＳ３０６において、CPU２１は、ライセンスのメッセージに基づいて、ICVを演算する。このICVについては、図４７乃至図５１を参照して後述する。ICVを用いてライセンスの改竄を防止することが可能となる。

次に、ステップＳ３０７において、CPU２１は、チェックアウト対象のライセンスと、ステップＳ３０６の処理で演算されたICVを、自分自身の公開鍵を用いて暗号化して、EKBおよび証明書とともに、相手側の装置に出力し、コピーさせる。さらに、ステップＳ３０８において、CPU２１は、ステップＳ３０６の処理で演算されたICVを、相手側装置のリーフキーと、ライセンスIDに対応して記憶部２８のチェックリスト中に記憶させる。

ステップＳ３０３において、チェックアウト回数Ｎ１が最大チェックアウト回数Ｎ２より小さくない（例えば、等しい）と判定された場合、もはや許容される回数だけチェックアウトが行われているので、これ以上チェックアウトを行うことができない。そこで、ステップＳ３０９に進み、CPU２１は、エラー処理を実行する。すなわち、この場合、チェックアウト処理は実行されないことになる。

次に、図４３のフローチャートを参照して、図４２のチェックアウト処理により、ライセンスのチェックアウトを受けるクライアントの処理について説明する。

最初に、ステップＳ３２１において、相手側装置（ライセンスをチェックアウトするクライアント１）に、自分自身のリーフキーを送信する。このリーフキーは、ステップＳ３０４において、相手側のクライアントにより、ライセンスIDに対応して記憶される。

次に、ステップＳ３２２において、CPU２１は、相手側のクライアント１から暗号化されたライセンスとICVが、EKBおよび証明書とともに送信されてきた場合、これを受信する。すなわち、このライセンス、ICV、EKBおよび証明書は、図４２のステップＳ３０７の処理で相手側の装置から送信されたものである。

ステップＳ３２３において、CPU２１は、ステップＳ３２２の処理で受信したライセンス、ICV、EKBおよび証明書を、記憶部２８に記憶させる。

以上のようにして、ライセンスのチェックアウトを受けたクライアント１は、チェックアウトを受けたそのライセンスを使用して、所定のコンテンツを再生する場合、図４４のフローチャートに示される処理を実行する。

すなわち、最初に、ステップＳ３４１において、クライアント１のCPU２１は、ユーザより入力部２６を介して再生が指定されたコンテンツのICVを演算する。そして、ステップＳ３４２において、CPU２１は、記憶部２８に記憶されている暗号化されているICVを、証明書に含まれている公開鍵に基づいて、復号させる。

次に、ステップＳ３４３において、CPU２１は、ステップＳ３４１の処理により、いま演算されたICVと、ステップＳ３４２の処理により読み出され、復号されたICVが一致するか否かを判定する。両者が一致する場合には、ライセンスは改竄されていないことになる。そこで、ステップＳ３４４にすすみ、CPU２１は、対応するコンテンツを再生する処理を実行する。

これに対して、ステップＳ３４３において、２つのICVが一致しないと判定された場合、ライセンスは改竄されている恐れがある。このため、ステップＳ３４５に進み、CPU２１は、エラー処理を実行する。すなわち、このとき、そのライセンスを用いてコンテンツを再生することができないことになる。

次に、以上のようにして、他のクライアントに一旦チェックアウトしたライセンスのチェックインを受けるクライアントの処理について、図４５のフローチャートを参照して説明する。

最初に、ステップＳ３６１において、CPU２１は、相手側の装置（ライセンスを返却（チェックイン）してくるクライアント１）のリーフキーと、チェックイン対象のライセンスのIDを取得する。次に、ステップＳ３６２において、CPU２１は、ステップＳ３６１で取得されたチェックイン対象のライセンスが、自分自身が相手側装置にチェックアウトしたライセンスであるか否かを判定する。この判定は、図４２のステップＳ３０８の処理で記憶された ICV、リーフキー、およびライセンスIDに基づいて行われる。すなわち、ステップＳ３６１で取得されたリーフキー、ライセンスID、およびICVが、チェックアウトリスト中に記憶されているか否かが判定され、記憶されている場合には、自分自身がチェックアウトしたライセンスであると判定される。

ライセンスが、自分自身がチェックアウトしたものであるとき、ステップＳ３６３において、CPU２１は、相手側の装置のライセンス、EKBおよび証明書の削除を要求する。後述するように、この要求に基づいて、相手側の装置は、ライセンス、EKBおよび証明書の削除を実行する（図４６のステップＳ３８３）。

ステップＳ３６４において、CPU２１は、一旦チェックアウトしたライセンスが再びチェックインされてきたので、そのライセンスのチェックアウト回数N１を１だけデクリメントする。

ステップＳ３６５において、CPU２１は、相手側の装置に他のライセンスをチェックアウトしているか否かを判定し、まだチェックアウトしている他のライセンスが存在しない場合には、ステップＳ３６６に進み、CPU２１は、相手側の装置のチェックイン対象機器としてのチェックアウトリストにおける記憶を削除する。これに対して、ステップＳ３６５において、相手側の装置にチェックアウトしている他のライセンスが存在すると判定された場合には、他のライセンスのチェックインを受ける可能性があるので、ステップＳ３６６の処理はスキップされる。

ステップＳ３６２において、チェックイン対象とされているライセンスが、自分自身が相手側装置にチェックアウトしたライセンスではないと判定された場合、CPU２１は、ステップＳ３６７に進み、エラー処理を実行する。すなわち、この場合には、自分自身が管轄するライセンスではないことになるので、チェックイン処理は実行されない。

ユーザが、ライセンスを不正にコピーしたような場合、記憶されているICVの値と、ステップＳ３６１の処理で取得されたライセンスに基づいて演算されたICVの値が異なるものとなるで、チェックインできないことになる。

図４６は、図４５のフローチャートに示されるライセンスのチェックイン処理を実行するクライアントに対して、自分自身が有しているライセンスをチェックインさせるクライアントの処理を表している。

ステップＳ３８１において、クライアント１のCPU２１は、相手側の装置（図４５のフローチャートに示す処理を実行するクライアント１）にリーフキーとチェックイン対象のライセンスのIDを送信する。上述したように、相手側の装置は、ステップＳ３６１において、このリーフキーとライセンスIDを取得し、ステップＳ３６２において、それに基づいて、チェックイン対象のライセンスの認証処理を実行する。

ステップＳ３８２において、クライアント１のCPU２１は、相手側の装置からライセンスの削除を要求されたか否かを判定する。すなわち、ライセンスが正当なチェックイン対象のライセンスである場合、上述したように、相手側の装置は、ステップＳ３６３の処理でライセンス、EKBおよび証明書の削除を要求してくる。そこで、この要求を受信した場合、ステップＳ３８３に進み、CPU２１は、ライセンス、EKBおよび証明書を削除する。すなわち、これにより、このクライアント１は、以後そのライセンスを使用できない状態となり、図４５のステップＳ３６４の処理により、チェックアウト回数Ｎ１が、１だけデクリメンドされるので、チェックインが完了したことになる。

ステップＳ３８２において、相手側の装置からライセンスの削除が要求されていないと判定された場合、ステップＳ３８４に進み、エラー処理が実行される。すなわち、この場合には、ICVの値が異なっている等の理由により、チェックインができないことになる。

以上においては、チェックインとチェックアウトについて説明したが、同様に、ライセンスをコピーあるいはムーブさせるようにすることも可能である。

次に、ライセンス（コンテンツも同様）の改竄を防止するためにライセンスのインテグリティ・チェック値（ＩＣＶ）を生成して、ライセンスに対応付けて、ＩＣＶの計算により、ライセンス改竄の有無を判定する処理構成について説明する。

ライセンスのインテグリティ・チェック値（ＩＣＶ）は、例えばライセンスに対するハッシュ関数を用いて計算され、ＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，Ｌ１，Ｌ

２，・・・）によって計算される。ＫｉｃｖはＩＣＶ生成キーである。Ｌ１，Ｌ２はライセンスの情報であり、ライセンスの重要情報のメッセージ認証符号（ＭＡＣ：Message authentication Code）が使用される。

ＤＥＳ暗号処理構成を用いたＭＡＣ値生成例を図４７に示す。図４７の構成に示すように対象となるメッセージを８バイト単位に分割し、（以下、分割されたメッセージをＭ１、Ｍ２、・・・、ＭＮとする）、まず、初期値（IV）とＭ１を、演算部２４－１Ａにより排他的論理和する（その結果をＩ１とする）。次に、Ｉ１をＤＥＳ暗号化部２４－１Ｂに入れ、鍵（以下、Ｋ１とする）を用いて暗号化する（出力をＥ１とする）。続けて、Ｅ１およびＭ２を演算部２４－２Ａにより排他的論理和し、その出力Ｉ２をＤＥＳ暗号化部２４－２Ｂへ入れ、鍵Ｋ１を用いて暗号化する（出力Ｅ２）。以下、これを繰り返し、全てのメッセージに対して暗号化処理を施す。ＤＥＳ暗号化部２４－NB から最後に出てきたＥＮがメッセージ認証符号（ＭＡＣ（Message Authentication Code））となる。

このようなライセンスのＭＡＣ値とＩＣＶ生成キーにハッシュ関数を適用してライセンスのインテグリティ・チェック値（ＩＣＶ）が生成される。例えばライセンス生成時に生成したＩＣＶと、新たにライセンスに基づいて生成したＩＣＶとを比較して同一のＩＣＶが得られればライセンスに改竄のないことが保証され、ＩＣＶが異なれば、改竄があったと判定される。

次に、ライセンスのインテグリティ・チェック値（ＩＣＶ）生成キーであるＫｉｃｖを上述の有効化キーブロックによって送付する構成について説明する。すなわちＥＫＢによる暗号化メッセージデータをライセンスのインテグリティ・チェック値（ＩＣＶ）生成キーとした例である。

図４８および図４９に複数のデバイスに共通のライセンスを送付した場合、それらのライセンスの改竄の有無を検証するためのインテグリティ・チェック値生成キーＫｉｃｖを有効化キーブロック（ＥＫＢ）によって配信する構成例を示す。図４８はデバイス０，１，２，３に対して復号可能なチェック値生成キーＫｉｃｖを配信する例を示し、図４９はデバイス０，１，２，３中のデバイス３をリボ

ーク（排除）してデバイス０，１，２に対してのみ復号可能なチェック値生成キーＫｉｃｖを配信する例を示す。

図４８の例では、更新ノードキーＫ（ｔ）００によって、チェック値生成キーＫｉｃｖを暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）とともに、デバイス０，１，２，３においてそれぞれの有するノードキー、リーフキーを用いて更新されたノードキーＫ（ｔ）００を復号可能な有効化キーブロック（ＥＫＢ）を生成して配信する。それぞれのデバイスは、図４８の右側に示すように、まず、ＥＫＢを処理（復号）することにより、更新されたノードキーＫ（ｔ）００を取得し、次に、取得したノードキーＫ（ｔ）００を用いて、暗号化されたチェック値生成キーＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）を復号して、チェック値生成キーＫｉｃｖを得ることが可能となる。

その他のデバイス４，５，６，７・・・は同一の有効化キーブロック（ＥＫＢ）を受信しても自身の保有するノードキー、リーフキーでは、ＥＫＢを処理して更新されたノードキーＫ（ｔ）００を取得することができないので、安全に正当なデバイスに対してのみチェック値生成キーを送付することができる。

一方、図４９の例は、図１２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，に対してのみ復号可能な有効化キーブロック（ＥＫＢ）を生成して配信した例である。図４９に示す有効化キーブロック（ＥＫＢ）と、チェック値生成キー（Ｋｉｃｖ）をノードキー（Ｋ（ｔ）００）で暗号化したデータＥｎｃ（Ｋ（ｔ）００，Ｋｉｃｖ）を配信する。

図４９の右側には、復号手順を示してある。デバイス０，１，２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またはノードキーを用いた復号処理により、更新ノードキー（Ｋ（ｔ）００）を取得する。次に、Ｋ（ｔ）００による復号によりチェック値生成キーＫｉｃｖを取得する。

図１２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

て更新ノードキー（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、更新ノードキー（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チェック値生成キーを復号して利用することが可能となる。

このように、ＥＫＢを利用したチェック値生成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チェック値生成キーを配信することが可能となる。

このようなライセンスのインテグリティ・チェック値（ＩＣＶ）を用いることにより、ＥＫＢと暗号化ライセンスの不正コピーを排除することができる。例えば図５０Ａに示すように、ライセンスＬ１とライセンスＬ２とをそれぞれのライセンスキーを取得可能な有効化キーブロック（ＥＫＢ）とともに格納したメディア１があり、これをそのままメディア２にコピーした場合を想定する。ＥＫＢと暗号化ライセンスのコピーは可能であり、これを、ＥＫＢを復号可能なデバイスでは利用できることになる。

図５０Ｂに示す例では、各メディアに正当に格納されたライセンスに対応付けてインテグリティ・チェック値（ＩＣＶ（Ｌ１，Ｌ２））を格納する構成とする。なお、（ＩＣＶ（Ｌ１，Ｌ２））は、ライセンスＬ１とライセンスＬ２にハッシュ関数を用いて計算されるライセンスのインテグリティ・チェック値であるＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，Ｌ１，Ｌ２）を示している。図５０Ｂの構成において、メディア１には正当にライセンス１とライセンス２が格納され、ライセンスＬ１とライセンスＬ２に基づいて生成されたインテグリティ・チェック値（ＩＣＶ（Ｌ１，Ｌ２））が格納される。また、メディア２には正当にライセンス１が格納され、ライセンスＬ１に基づいて生成されたインテグリティ・チェック値（ＩＣＶ（Ｌ１））が格納される。

この構成において、メディア１に格納された｛ＥＫＢ，ライセンス２｝をメディア２にコピーしたとすると、メディア２で、ライセンスチェック値を新たに生成すると、ＩＣＶ（Ｌ１，Ｌ２）が生成されることになり、メディア２に格納さ

れているＫｉｃｖ（Ｌ１）と異なり、ライセンスの改竄あるいは不正なコピーによる新たなライセンスの格納が実行されたことが明らかになる。メディアを再生するデバイスにおいて、再生ステップの前ステップにＩＣＶチェックを実行して、生成ＩＣＶと格納ＩＣＶの一致を判別し、一致しない場合は、再生を実行しない構成とすることにより、不正コピーのライセンスの再生を防止することが可能となる。

また、さらに、安全性を高めるため、ライセンスのインテグリティ・チェック値（ＩＣＶ）を書き換えカウンタを含めたデータに基づいて生成する構成としてもよい。すなわちＩＣＶ＝ｈａｓｈ（Ｋｉｃｖ，ｃｏｕｎｔｅｒ＋１，Ｌ１，Ｌ２，・・・）によって計算する構成とする。ここで、カウンタ（ｃｏｕｎｔｅｒ＋１）は、ＩＣＶの書き換えごとに１つインクリメントされる値として設定する。なお、カウンタ値はセキュアなメモリに格納する構成とすることが必要である。

さらに、ライセンスのインテグリティ・チェック値（ＩＣＶ）をライセンスと同一メディアに格納することができない構成においては、ライセンスのインテグリティ・チェック値（ＩＣＶ）をライセンスとは別のメディア上に格納する構成としてもよい。

例えば、読み込み専用メディアや通常のＭＯ等のコピー防止策のとられていないメディアにライセンスを格納する場合、同一メディアにインテグリティ・チェック値（ＩＣＶ）を格納するとＩＣＶの書き換えが不正なユーザによりなされる可能性があり、ＩＣＶの安全性が保てないおそれがある。この様な場合、ホストマシン上の安全なメディアにＩＣＶを格納して、ライセンスのコピーコントロール（例えばcheck-in/check-out、move）にＩＣＶを使用する構成とすることにより、ＩＣＶの安全な管理およびライセンスの改竄チェックが可能となる。

この構成例を図５１に示す。図５１では読み込み専用メディアや通常のＭＯ等のコピー防止策のとられていないメディア２２０１にライセンス１乃至ライセンス３が格納され、これらのライセンスに関するインテグリティ・チェック値（ＩＣＶ）を、ユーザが自由にアクセスすることの許可されないホストマシン上の安

全なメディア２２０２に格納し、ユーザによる不正なインテグリティ・チェック値（ＩＣＶ）の書き換えを防止した例である。このような構成として、例えばメディア２２０１を装着したデバイスが、メディア２２０１の再生を実行する際にホストマシンであるＰＣ、サーバにおいてＩＣＶのチェックを実行して再生の可否を判定する構成とすれば、不正なコピーライセンスあるいは改竄ライセンスの再生を防止できる。

本発明が適用されるクライアントは、いわゆるパーソナルコンピュータ以外に、PDA（Personal Digital Assistants）、携帯電話機、ゲーム端末機などとすることができる。

一連の処理をソフトウエアにより実行させる場合には、そのソフトウエアを構成するプログラムが、専用のハードウエアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、ネットワークや記録媒体からインストールされる。

この記録媒体は、図２に示されるように、装置本体とは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク４１（フロッピディスクを含む）、光ディスク４２（CD-ROM(Compact Disk - Read Only Memory),DVD(Digital Versatile Disk)を含む）、光磁気ディスク４３（MD（Mini-Disk）を含む）、もしくは半導体メモリ４４などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに提供される、プログラムが記録されているROM２２や、記憶部２８に含まれるハードディスクなどで構成される。

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

また、セキュリティに関連する処理を実行させるプログラムは、その処理を解析されるのを防ぐため、そのプログラム自体が暗号化されているのが望ましい。例えば、暗号処理などを行う処理については、そのプログラムをタンパーレジスタントモジュールとして構成することができる。

また、コンテンツを利用許可するライセンスを特定するためにコンテンツのヘッダに記載されている情報はライセンスを一意に識別するライセンスIDでなくてもよい。上記の実施例では、ライセンスＩＤが、コンテンツの利用に必要なライセンスを特定する情報であり、あるライセンスが利用を許可するコンテンツを特定する情報であり、クライアント１からライセンス要求によって要求されるライセンスを識別する情報である。コンテンツにコンテンツのそのコンテンツに関する各種属性情報のリストが記載され、ライセンスに、そのライセンスによって利用許可されるコンテンツの条件式を記載するようにしても良い。この場合では、コンテンツに含まれる属性情報がそのコンテンツの利用を許可するライセンスを特定する情報であり、ライセンスに含まれる条件式がそのライセンスが利用を許可するコンテンツを特定する情報であり、ライセンスＩＤはライセンスを一意に識別する情報となる。このようにした場合には、一つのコンテンツに複数のライセンスを対応付けることが可能になり、ライセンスの発行を柔軟に行うことができる。

また、コンテンツデータは音楽データに限らない。例えばコンテンツは、画像データ、動画データ、テキストデータ、アニメーションデータ、ソフトウェアプログラム、あるいはそれらを組み合わせたものであっても良い。

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

産業上の利用可能性

以上の如く、本発明の情報処理装置によれば、クライアントのキー要求に応じて、クライアント毎に異なるデバイスノードキーを配布するようにしたので、コ

ンテンツ配信システムによって配信されるコンテンツの利用を行うための登録処理を適宜行うことができる。

請求の範囲

1. 他の情報処理装置からのアクセスに基づいて、前記他の情報処理装置に暗号化されたコンテンツを復号するためのキーを提供する情報処理装置であって、

前記他の情報処理装置からキー要求を受信する受信手段と、

前記他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成手段と、

前記生成手段により生成された前記デバイスノードキーを前記他の情報処理装置に送信する送信手段と

を備えることを特徴とする情報処理装置。

2. 前記送信手段は、前記他の情報処理装置に割り当てられた前記リーフを識別するリーフ識別情報をさらに送信する

ことを特徴とする請求の範囲第1項に記載の情報処理装置。

3. 前記送信手段は、前記他の情報処理装置に割り当てられた秘密鍵をさらに送信する

ことを特徴とする請求の範囲第2項に記載の情報処理装置。

4. 前記受信手段は、前記他の情報処理装置の利用者に関する利用者情報をさらに受信する

ことを特徴とする請求の範囲第1項に記載の情報処理装置。

5. 前記受信手段により受信された利用者情報を、前記他の情報処理装置に割り当てられた前記リーフを識別するリーフ識別情報と対応付けて記録する記録手段をさらに備える

ことを特徴とする請求の範囲第1項に記載の情報処理装置。

6. 他の情報処理装置からのアクセスに基づいて、前記他の情報処理装置に暗号化されたコンテンツを復号するためのキーを提供する情報処理装置の情報処理方法であって、

前記他の情報処理装置からキー要求を受信する受信ステップと、

前記他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、

前記生成ステップの処理により生成された前記デバイスノードキーを前記他の情報処理装置に送信する送信ステップと

を含むことを特徴とする情報処理方法。

7． 他の情報処理装置からのアクセスに基づいて、前記他の情報処理装置に暗号化されたコンテンツを復号するためのキーを提供する情報処理装置用のプログラムであって、

前記他の情報処理装置からキー要求を受信する受信ステップと、

前記他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、

前記生成ステップの処理により生成された前記デバイスノードキーを前記他の情報処理装置に送信する送信ステップと

を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

8． 他の情報処理装置からのアクセスに基づいて、前記他の情報処理装置に暗号化されたコンテンツを復号するためのキーを提供する情報処理装置を制御するコンピュータが実行可能なプログラムであって、

前記他の情報処理装置からキー要求を受信する受信ステップと、

前記他の情報処理装置を鍵管理階層ツリー構造のリーフに割り当て、デバイスノードキーを生成する生成ステップと、

前記生成ステップの処理により生成された前記デバイスノードキーを前記他の情報処理装置に送信する送信ステップと

を含むことを特徴とするプログラム。

図1

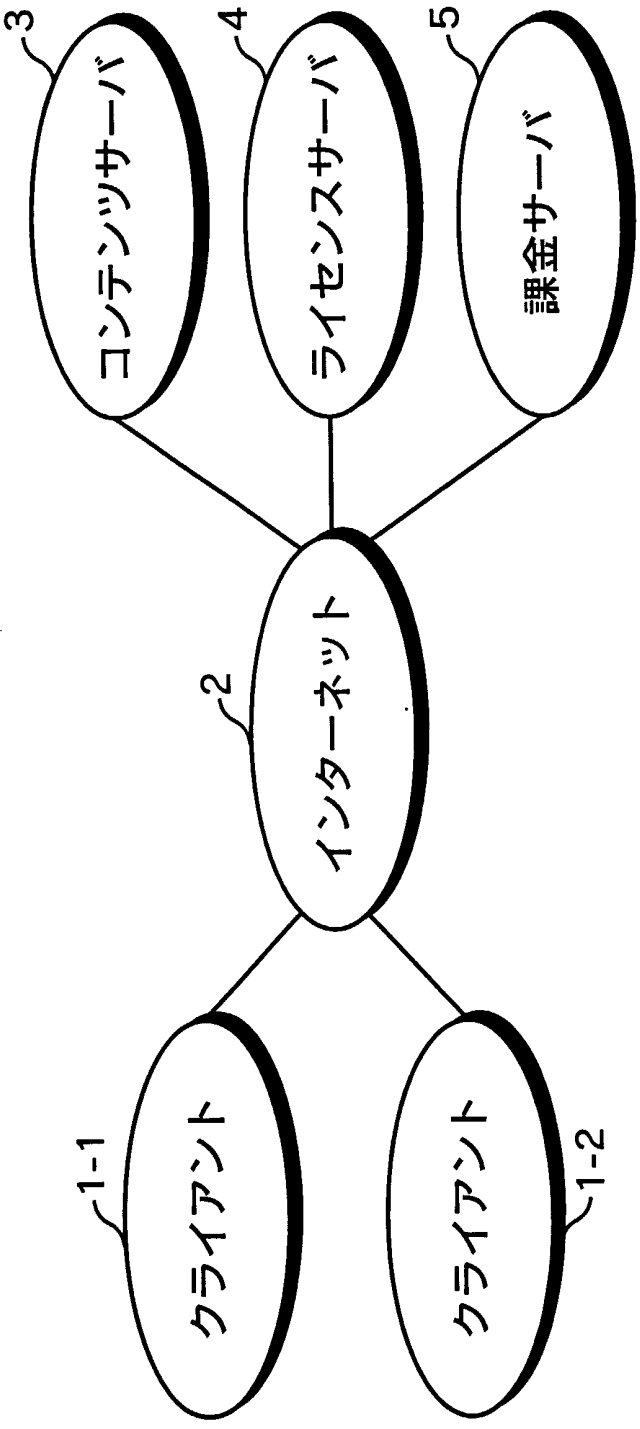


図2

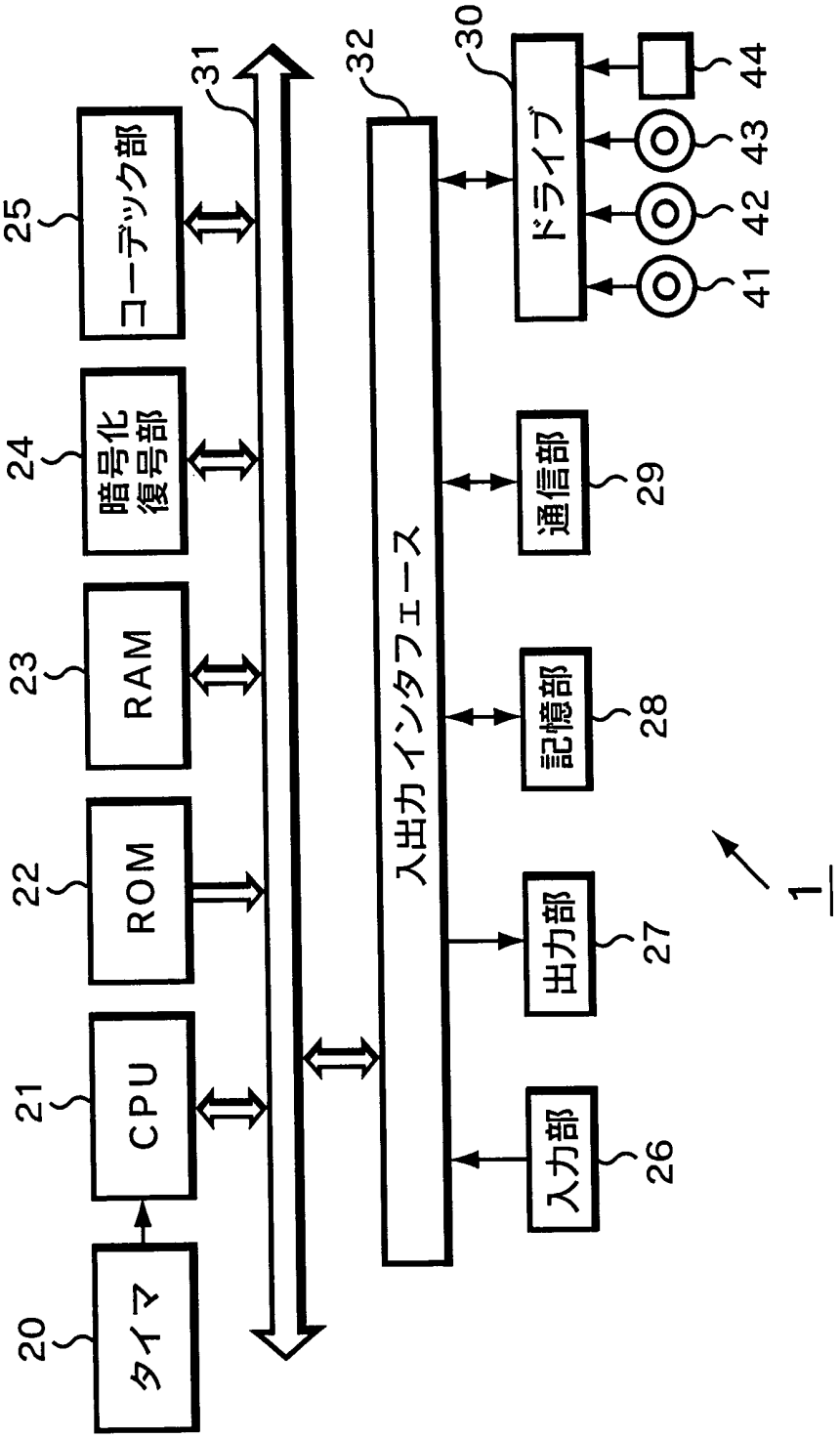

タイマ
20
CPU
21
ROM
22
RAM
23
暗号化
復号部
24
コーデック部
25
入力部
26
出力部
27
記憶部
28
通信部
29
ドライブ
30
31
32
入出力インタフェース
41
42
43
44
1

図3

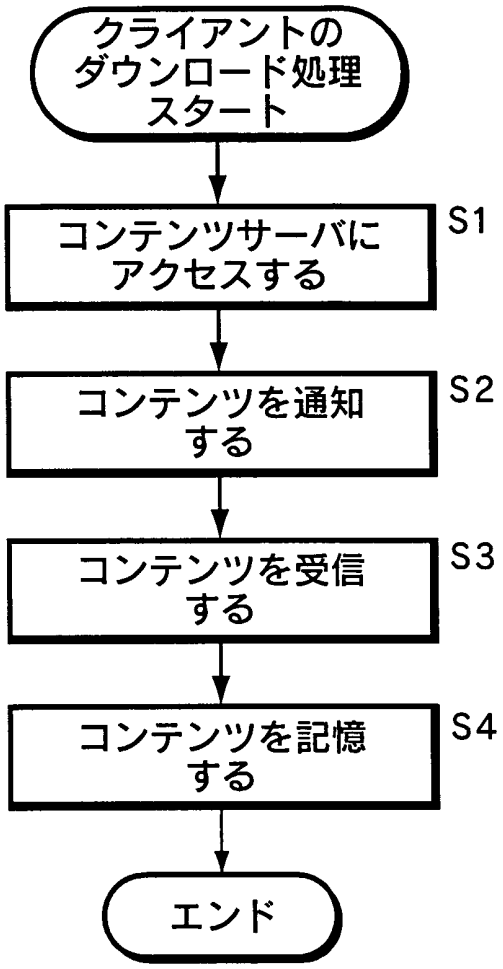

クライアントの
ダウンロード処理
スタート
コンテンツサーバに
アクセスする
S1
コンテンツを通知
する
S2
コンテンツを受信
する
S3
コンテンツを記憶
する
S4
エンド

図4

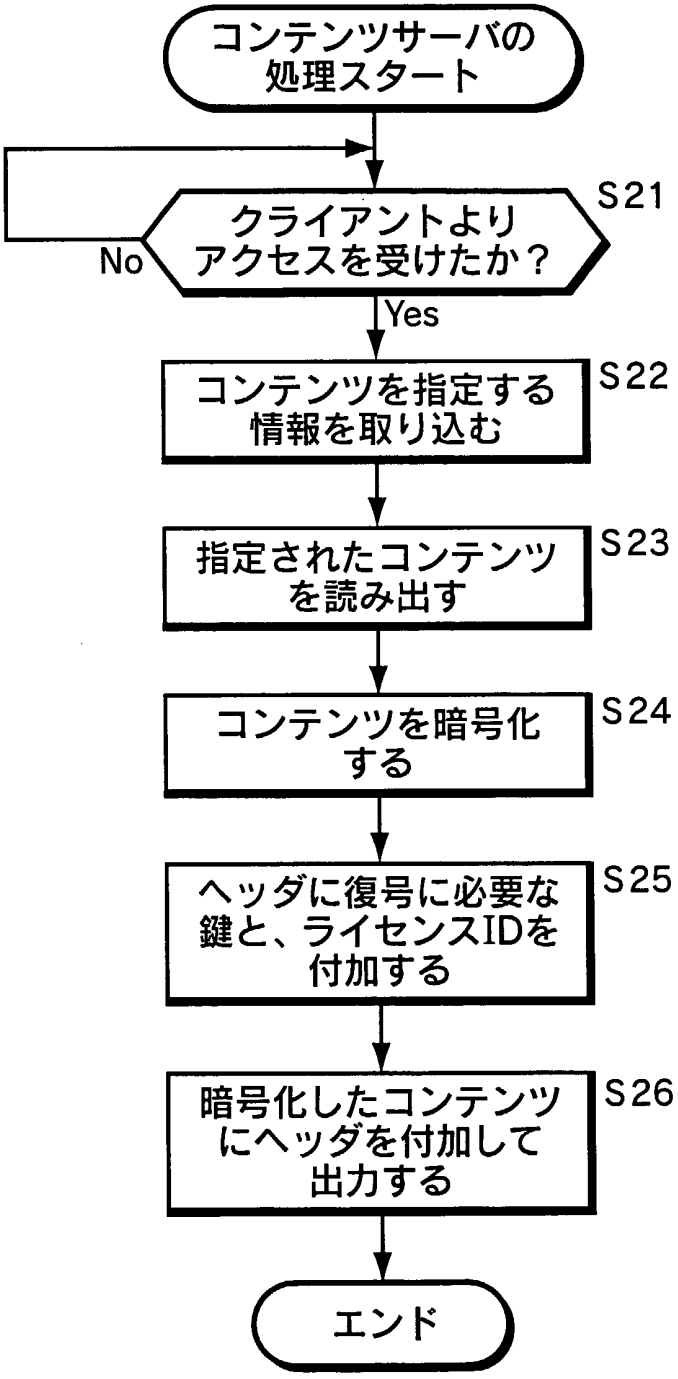

コンテンツサーバの処理スタート
S21
クライアントよりアクセスを受けたか？
No
Yes
S22
コンテンツを指定する情報を取り込む
S23
指定されたコンテンツを読み出す
S24
コンテンツを暗号化する
S25
ヘッダに復号に必要な鍵と、ライセンスIDを付加する
S26
暗号化したコンテンツにヘッダを付加して出力する
エンド

図5

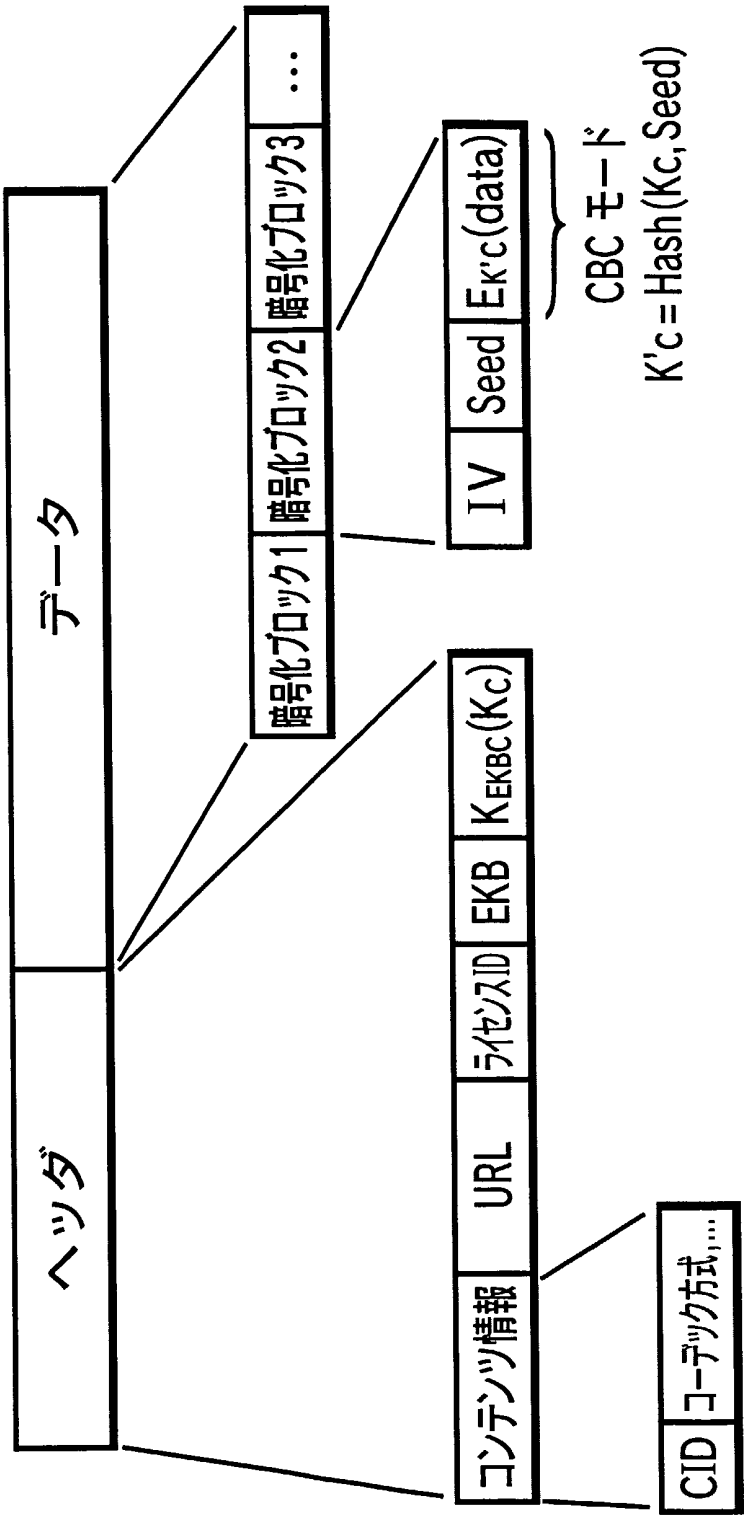

ヘッダ
データ
暗号化ブロック1
暗号化ブロック2
暗号化ブロック3
…
IV
Seed
EK'c(data)
CBC モード
K'c=Hash(Kc,Seed)
コンテンツ情報
URL
ライセンスID
EKB
KEKBC(Kc)
CID
コーデック方式,…

図6

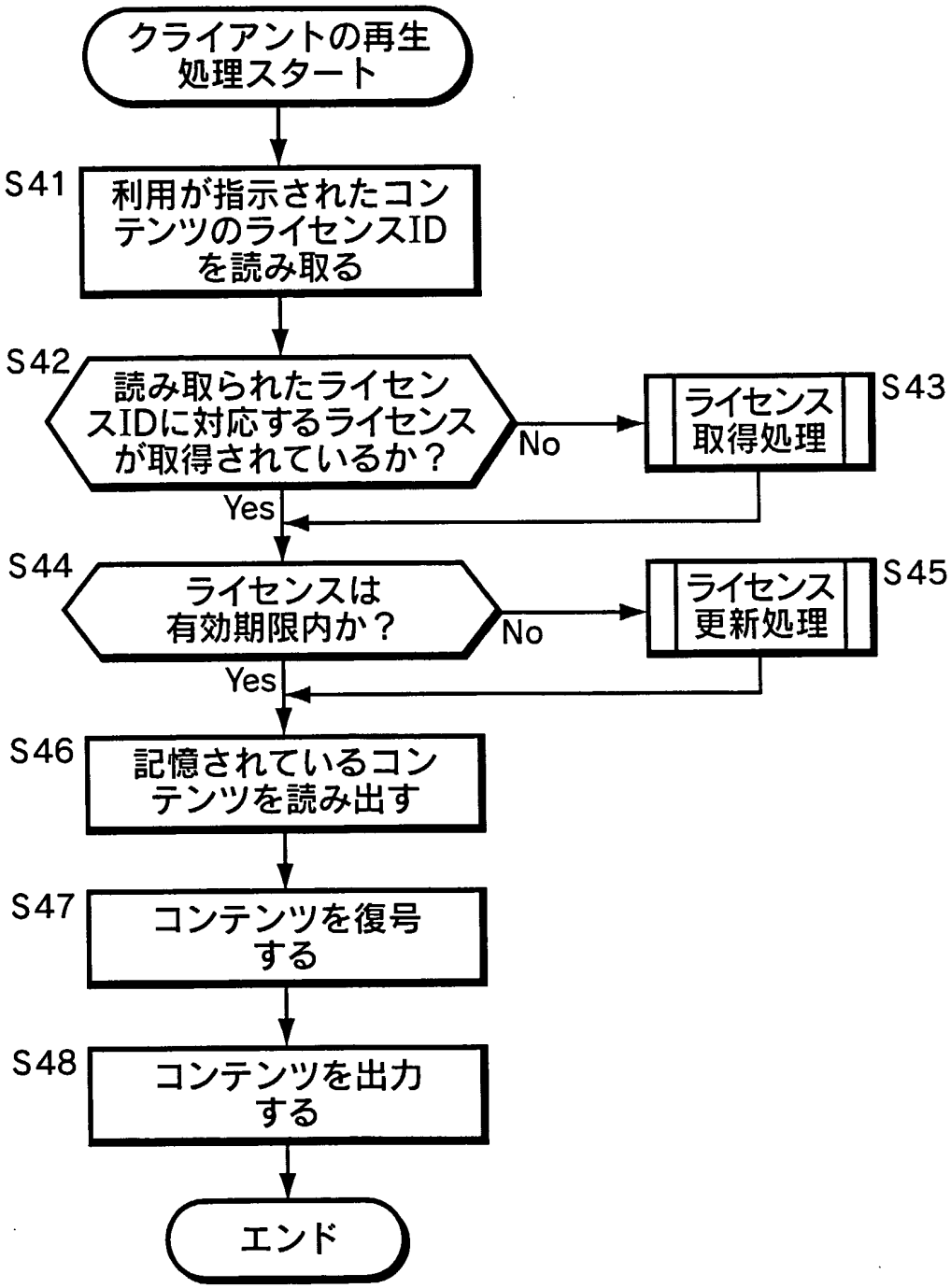

クライアントの再生処理スタート
S41
利用が指示されたコンテンツのライセンスIDを読み取る
S42
読み取られたライセンスIDに対応するライセンスが取得されているか？
No
S43
ライセンス取得処理
Yes
S44
ライセンスは有効期限内か？
No
S45
ライセンス更新処理
Yes
S46
記憶されているコンテンツを読み出す
S47
コンテンツを復号する
S48
コンテンツを出力する
エンド

図7

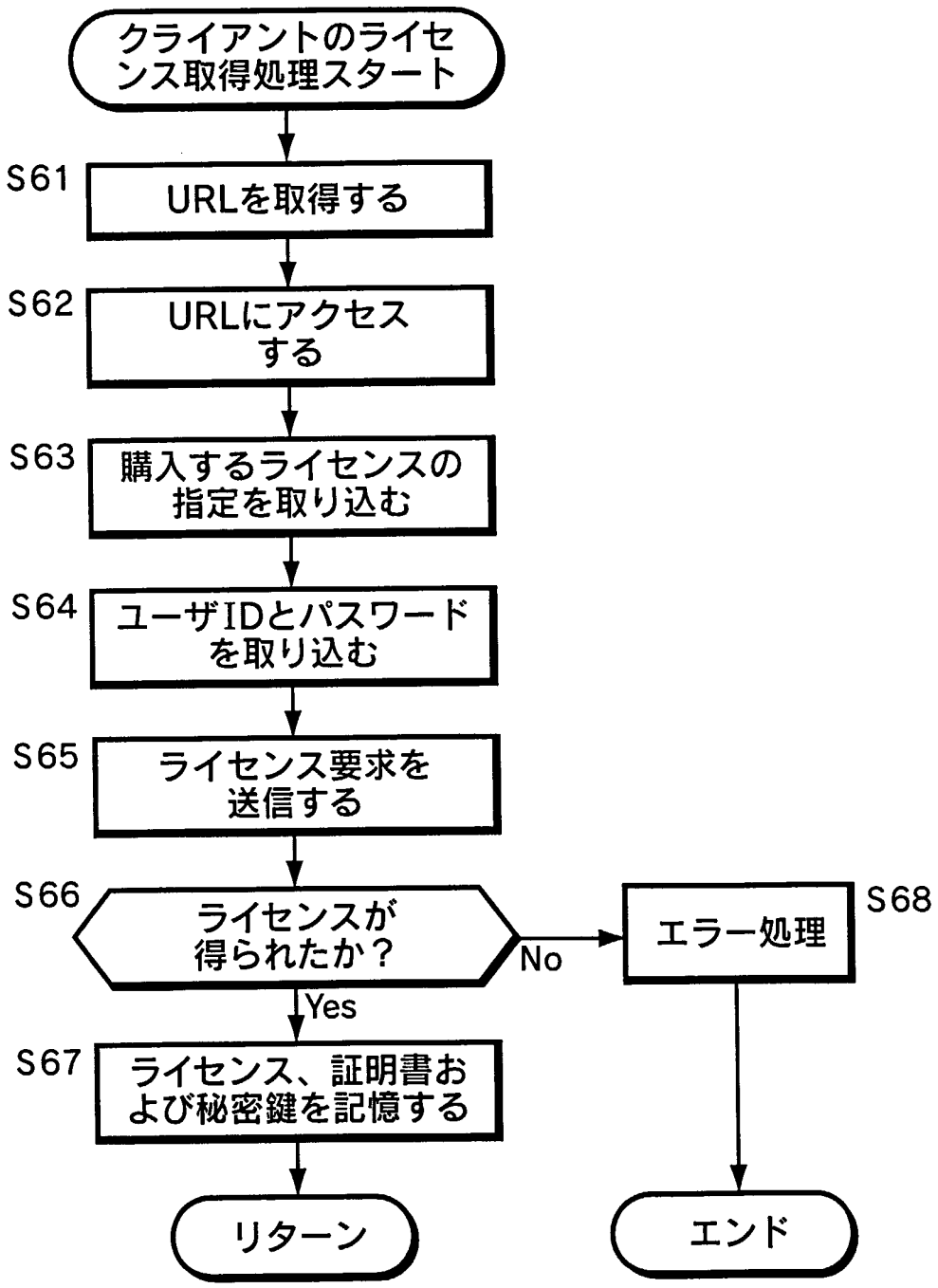

クライアントのライセンス取得処理スタート
S61
URLを取得する
S62
URLにアクセスする
S63
購入するライセンスの指定を取り込む
S64
ユーザIDとパスワードを取り込む
S65
ライセンス要求を送信する
S66
ライセンスが得られたか？
No
S68
エラー処理
Yes
S67
ライセンス、証明書および秘密鍵を記憶する
リターン
エンド

図8

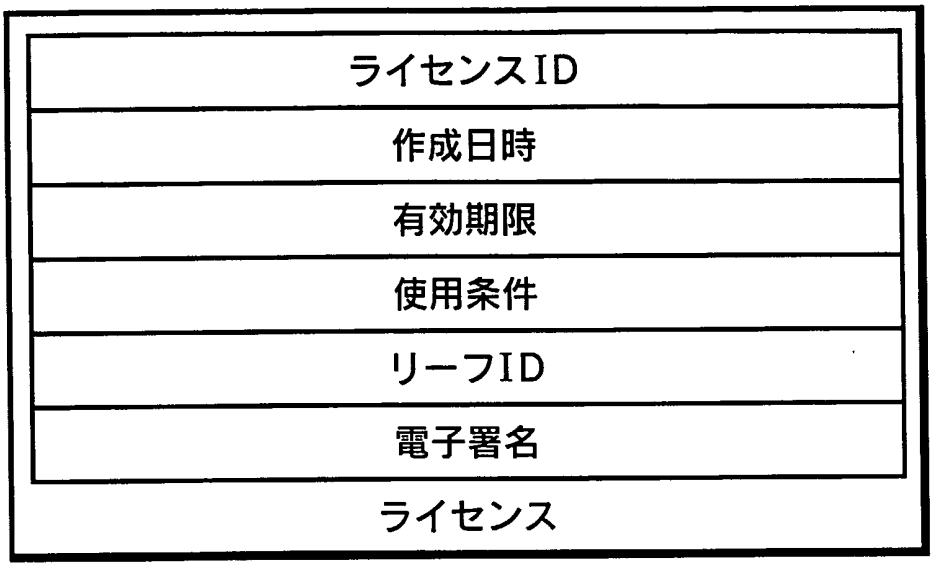


図9

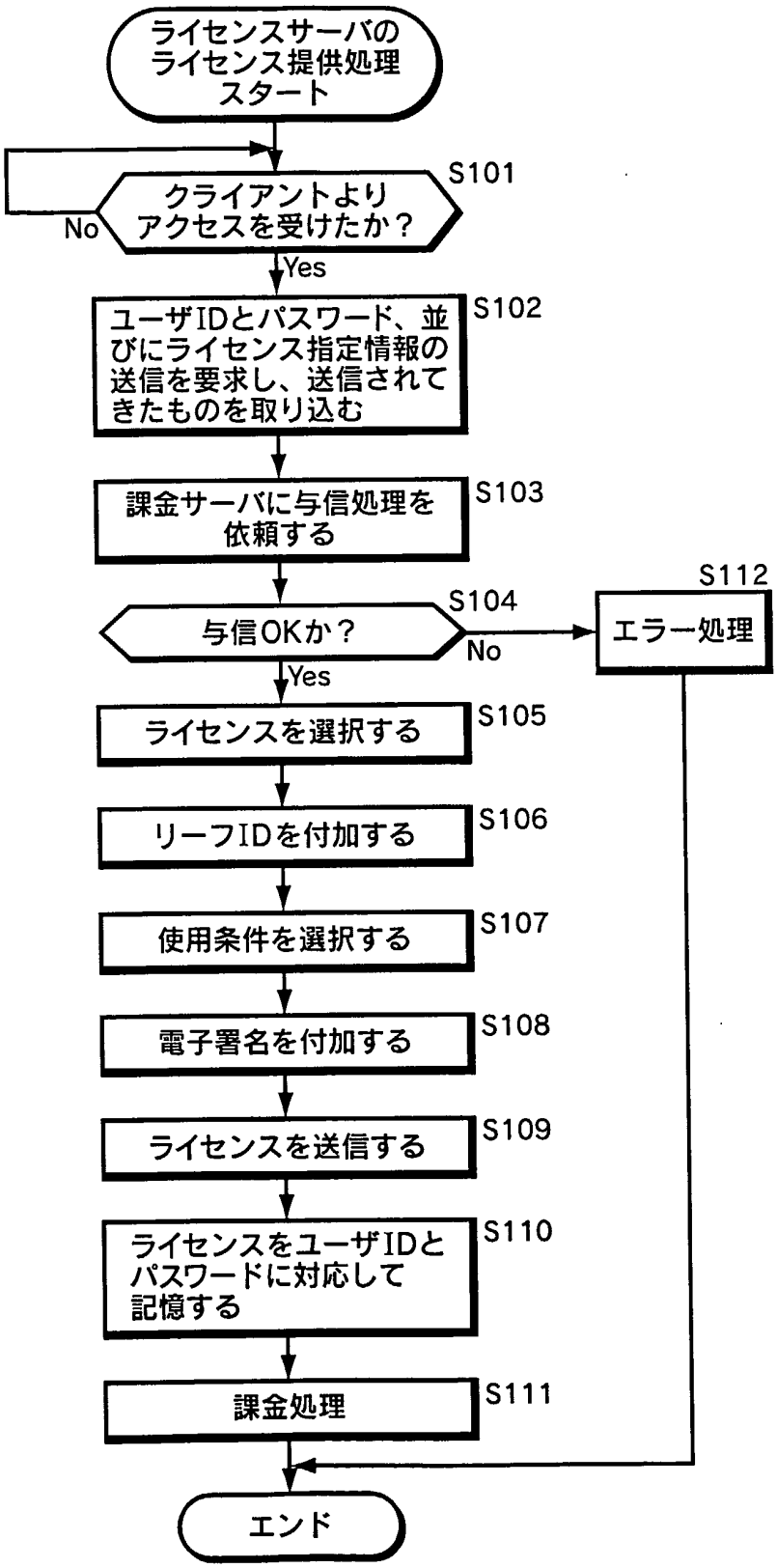

ライセンスサーバの
ライセンス提供処理
スタート
S101
クライアントより
アクセスを受けたか？
No
Yes
S102
ユーザIDとパスワード、並
びにライセンス指定情報の
送信を要求し、送信されて
きたものを取り込む
S103
課金サーバに与信処理を
依頼する
S104
与信OKか？
No
S112
エラー処理
Yes
S105
ライセンスを選択する
S106
リーフIDを付加する
S107
使用条件を選択する
S108
電子署名を付加する
S109
ライセンスを送信する
S110
ライセンスをユーザIDと
パスワードに対応して
記憶する
S111
課金処理
エンド

図10

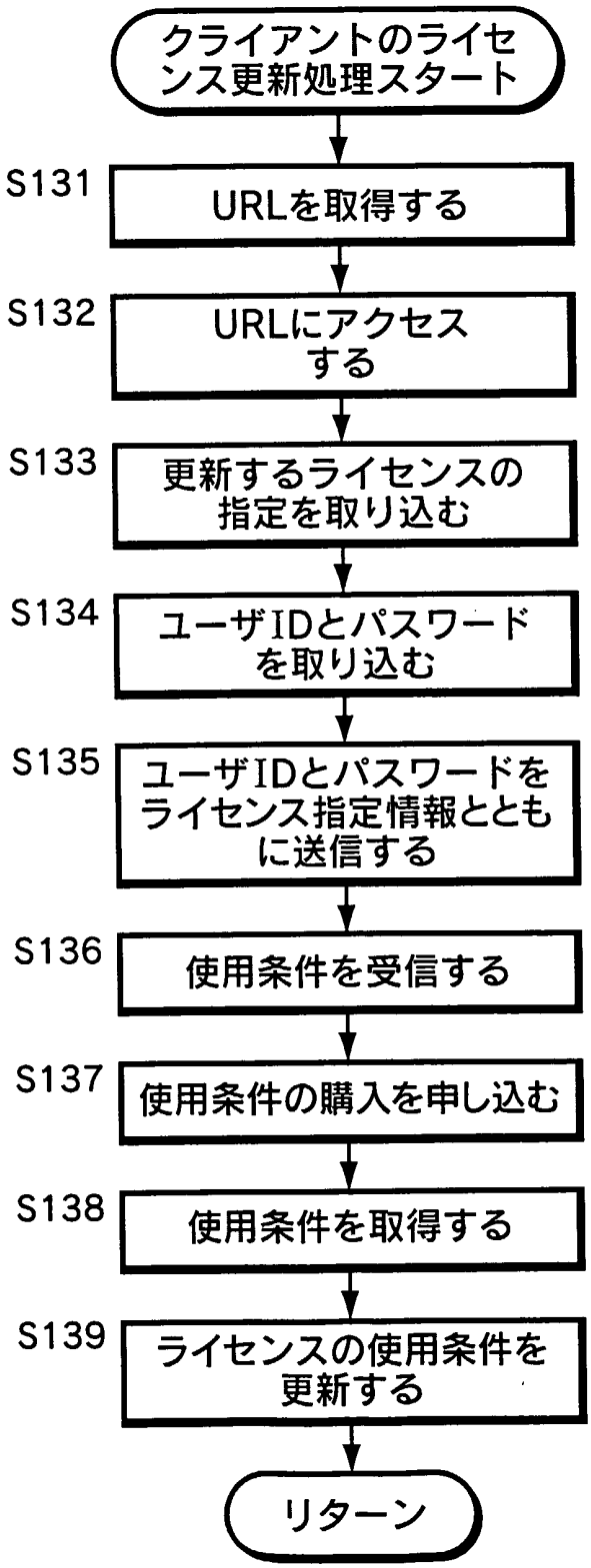

クライアントのライセンス更新処理スタート
S131
URLを取得する
S132
URLにアクセスする
S133
更新するライセンスの指定を取り込む
S134
ユーザIDとパスワードを取り込む
S135
ユーザIDとパスワードをライセンス指定情報とともに送信する
S136
使用条件を受信する
S137
使用条件の購入を申し込む
S138
使用条件を取得する
S139
ライセンスの使用条件を更新する
リターン

図11

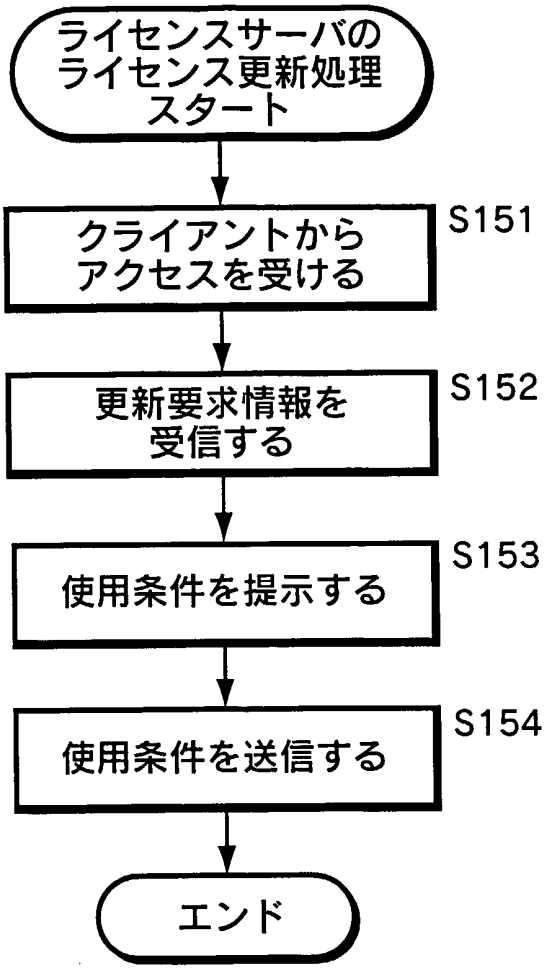

ライセンスサーバの
ライセンス更新処理
スタート
クライアントから
アクセスを受ける
S151
更新要求情報を
受信する
S152
使用条件を提示する
S153
使用条件を送信する
S154
エンド

図12

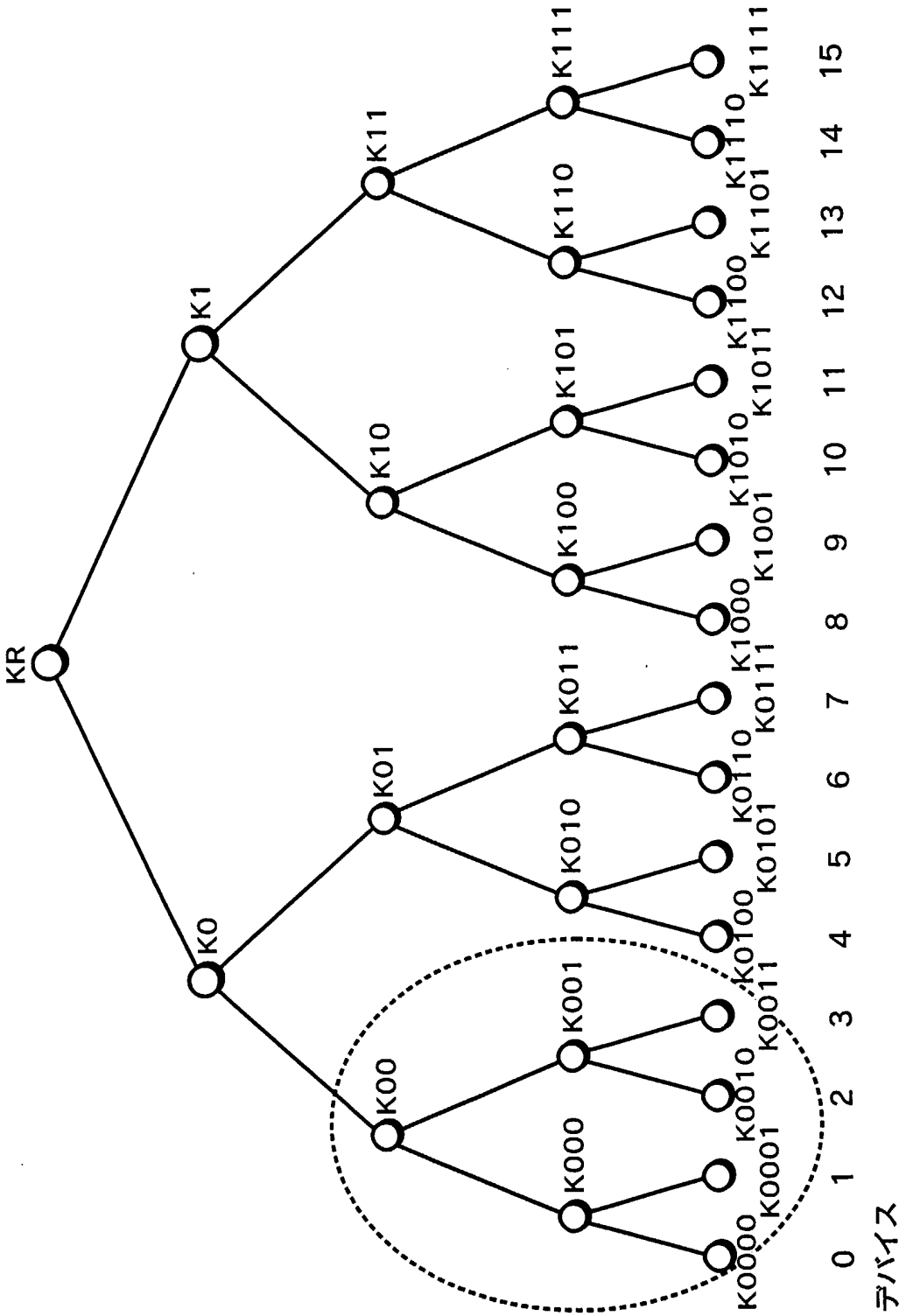

KR
K0
K1
K00
K01
K10
K11
K000
K001
K010
K011
K100
K101
K110
K111
K0000
K0001
K0010
K0011
K0100
K0101
K0110
K0111
K1000
K1001
K1010
K1011
K1100
K1101
K1110
K1111
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
デバイス

図13

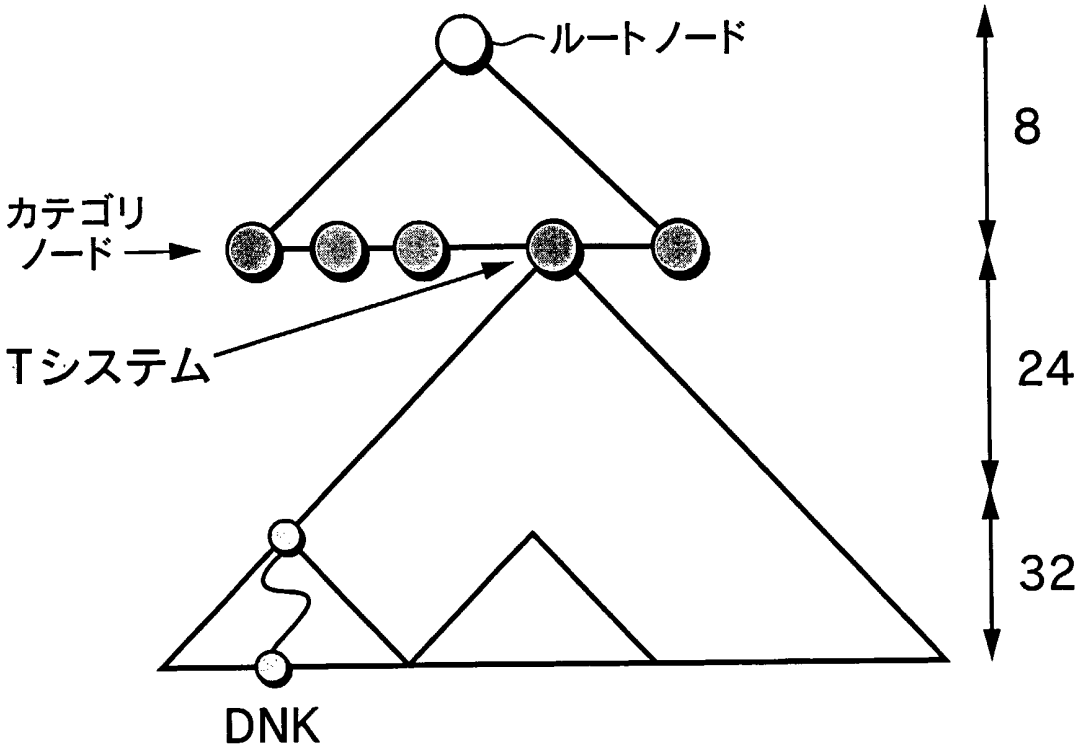


図14

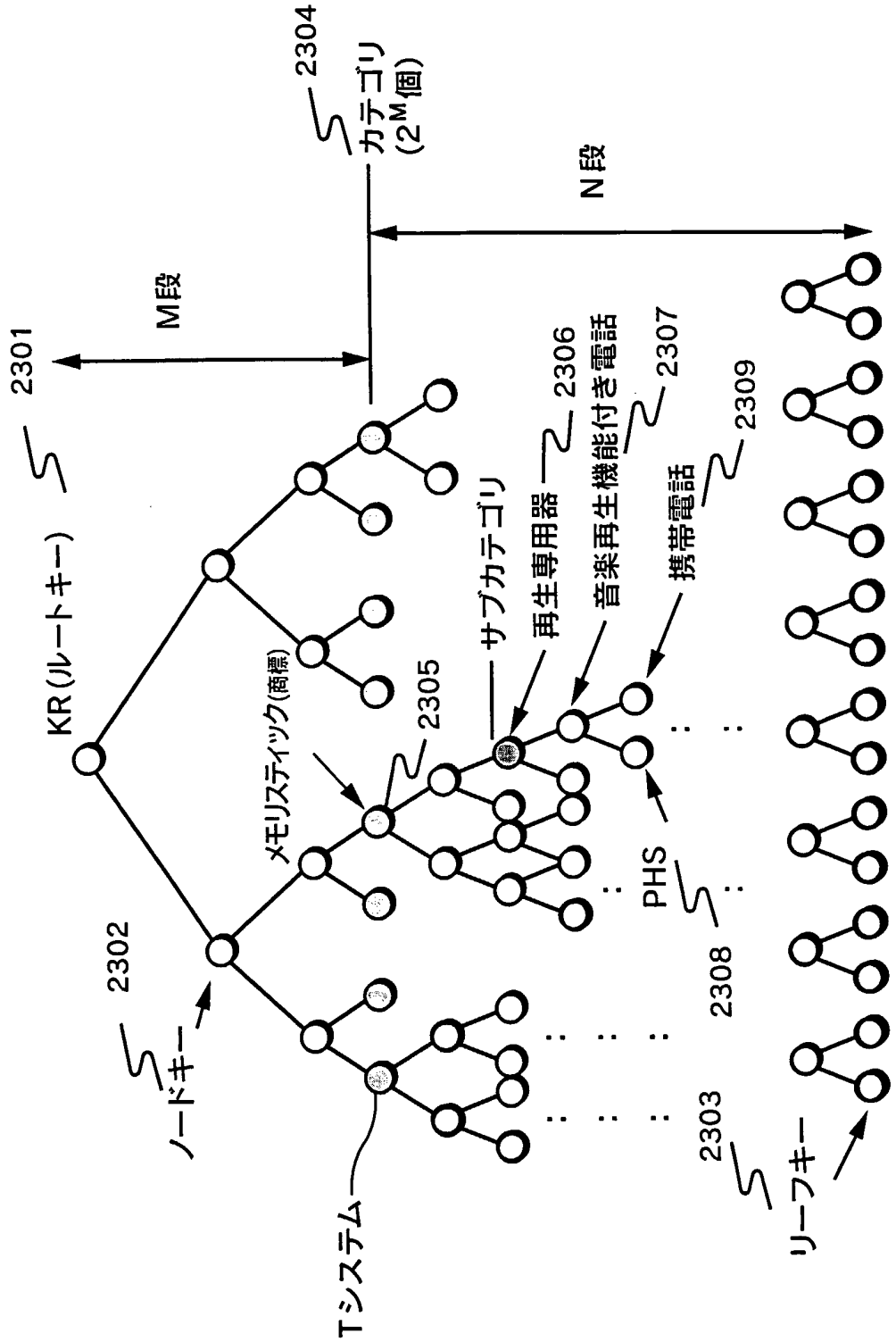

KR(ルートキー)
2301
M段
2304
カテゴリ
(2^M個)
N段
2306
再生専用器
2307
音楽再生機能付き電話
2309
携帯電話
サブカテゴリ
メモリスティック(商標)
2305
2302
ノードキー
Tシステム
PHS
2308
2303
リーフキー

図15A

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)

図15B

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16

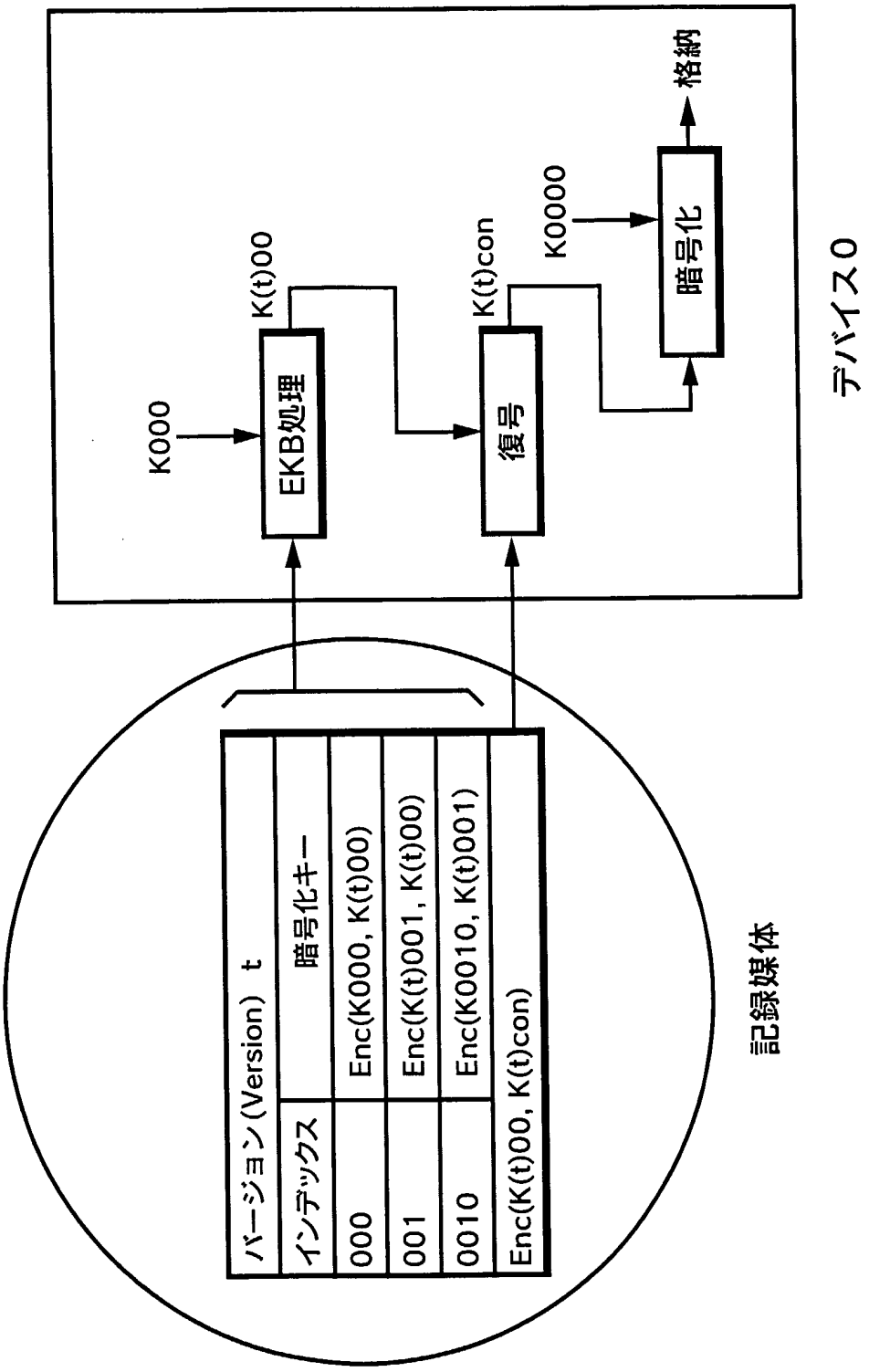

K000
EKB処理
K(t)00
復号
K(t)con
K0000
暗号化
格納
デバイス0
記録媒体
バージョン(Version) t
インデックス
暗号化キー
000
Enc(K000, K(t)00)
001
Enc(K(t)001, K(t)00)
0010
Enc(K0010, K(t)001)
Enc(K(t)00, K(t)con)

図17

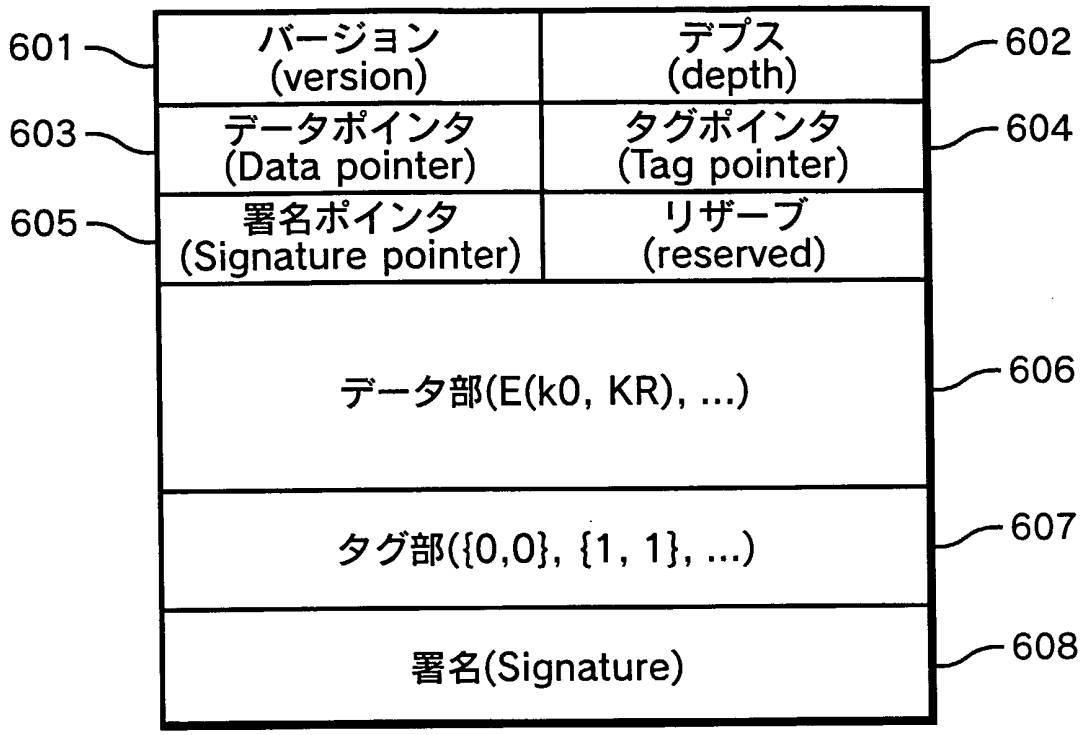

601
バージョン
(version)
デプス
(depth)
602
603
データポインタ
(Data pointer)
タグポインタ
(Tag pointer)
604
605
署名ポインタ
(Signature pointer)
リザーブ
(reserved)
データ部(E(k0, KR), ...)
606
タグ部({0,0}, {1, 1}, ...)
607
署名(Signature)
608

図18

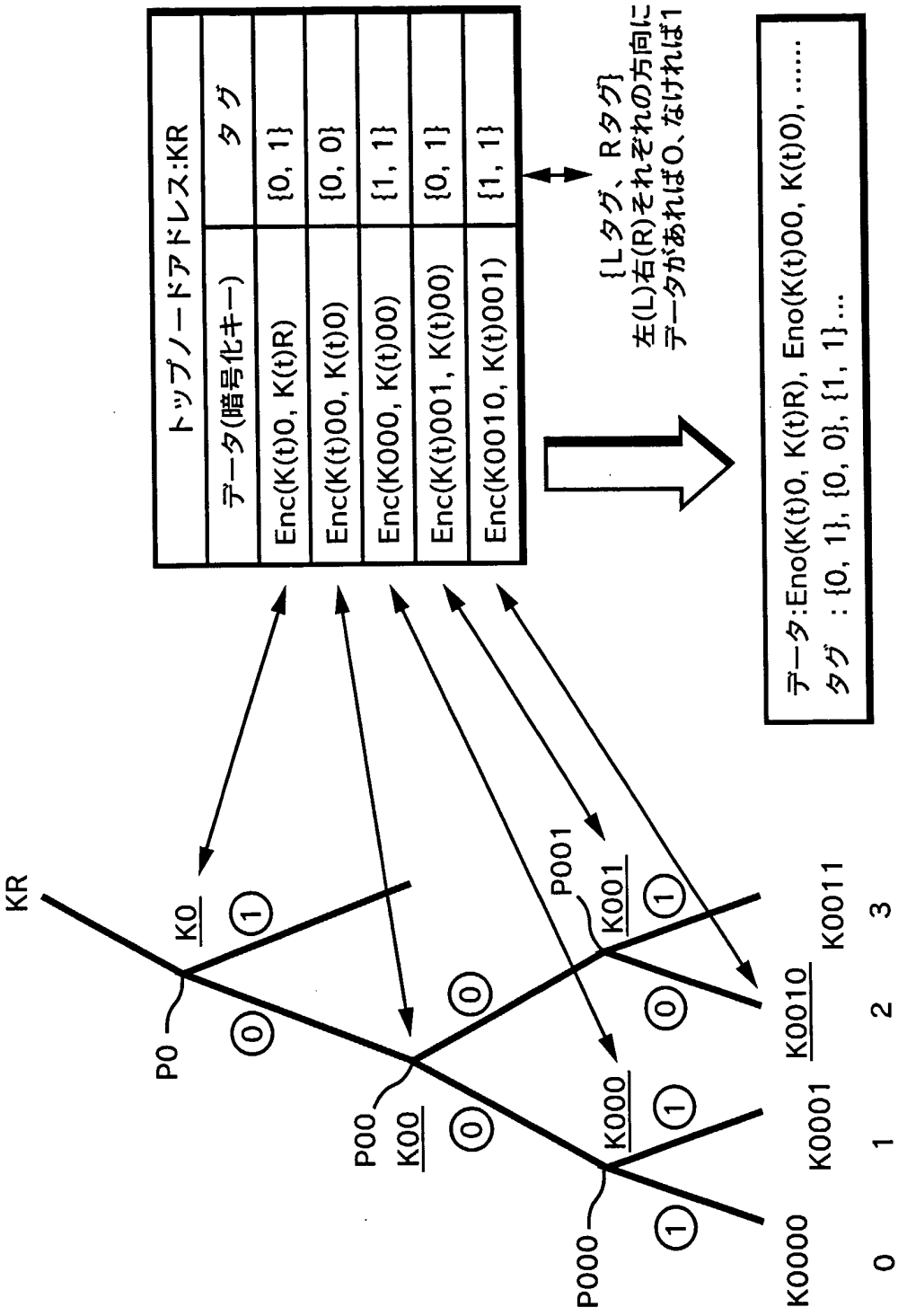

トップノードアドレス:KR
データ(暗号化キー)
タグ
Enc(K(t)0, K(t)R)
{0, 1}
Enc(K(t)00, K(t)0)
{0, 0}
Enc(K000, K(t)00)
{1, 1}
Enc(K(t)001, K(t)00)
{0, 1}
Enc(K0010, K(t)001)
{1, 1}
{Lタグ、Rタグ}
左(L)右(R)それぞれの方向に
データがあれば0、なければ1
データ:Eno(K(t)0, K(t)R), Eno(K(t)00, K(t)0), ...
タグ :{0, 1}, {0, 0}, {1, 1}...
KR
K0
P0
P00
K00
P000
K000
P001
K001
K0010
K0011
K0000
K0001
0
1
2
3

図19

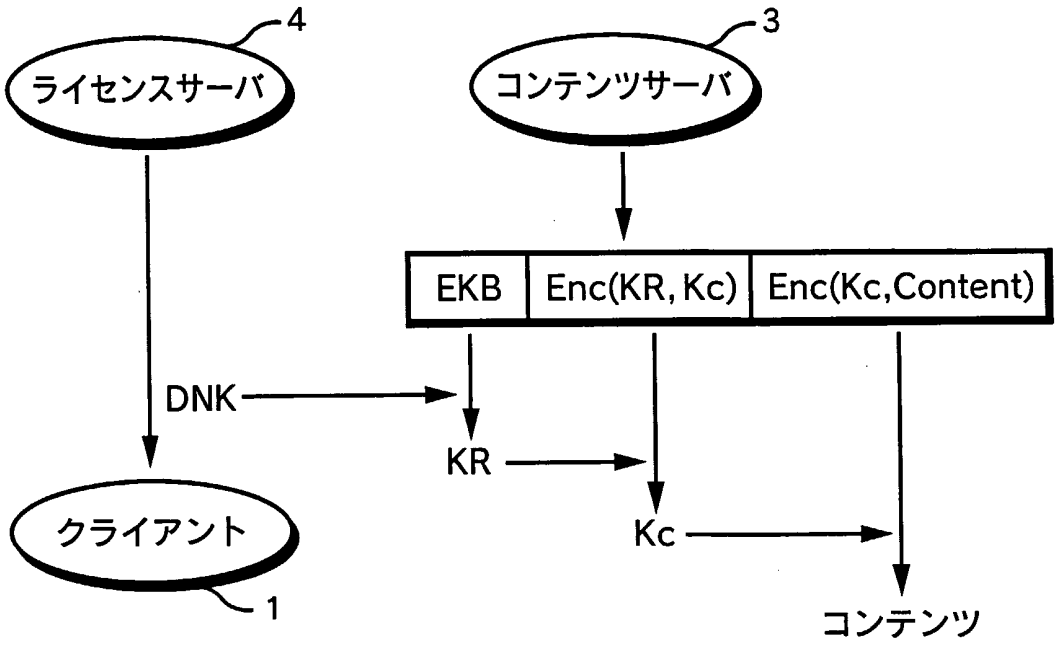

4
ライセンスサーバ
3
コンテンツサーバ
EKB
Enc(KR, Kc)
Enc(Kc,Content)
DNK
KR
Kc
クライアント
1
コンテンツ

図20

EKB


Enc(DNK, KR)

図21

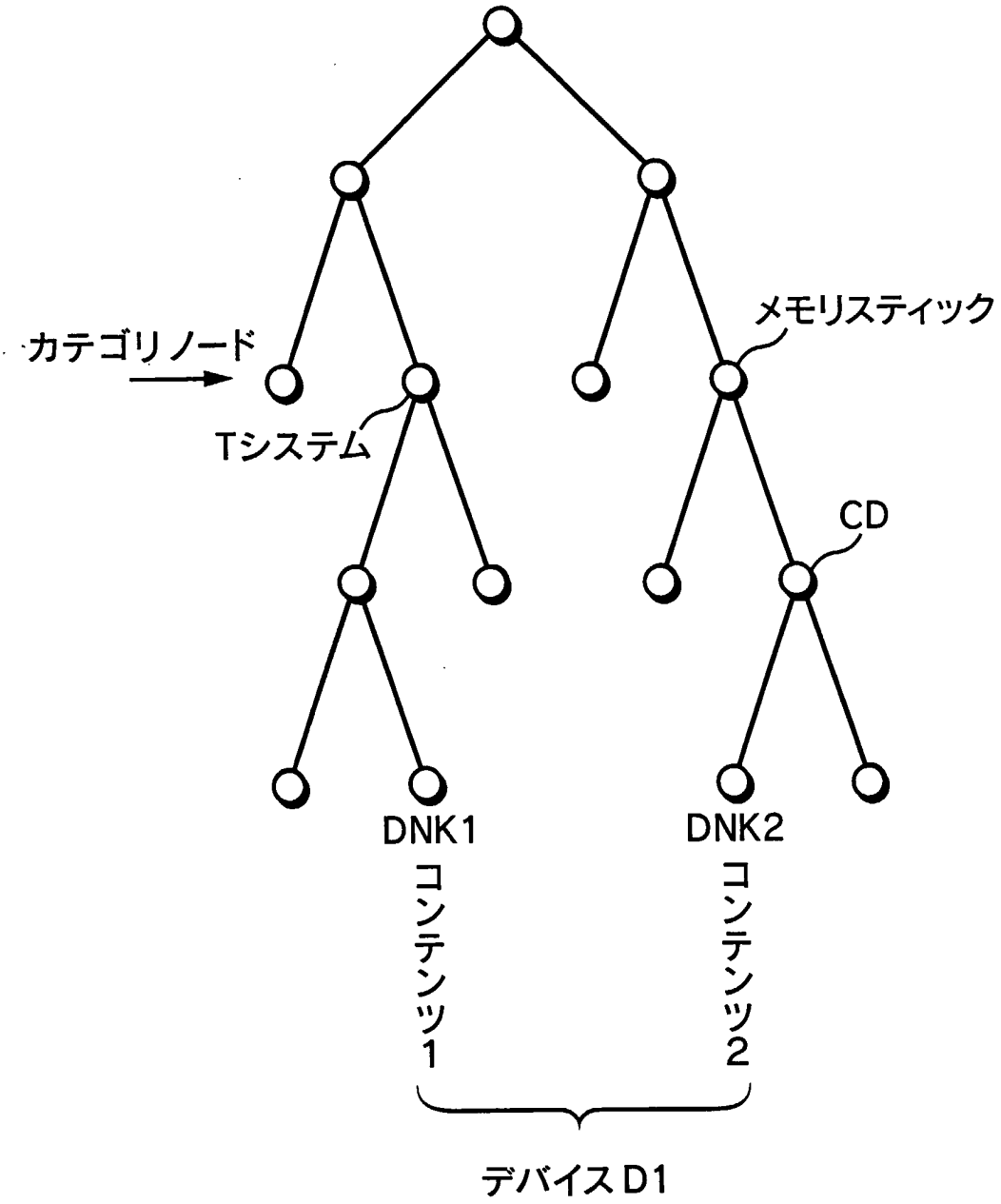


図22

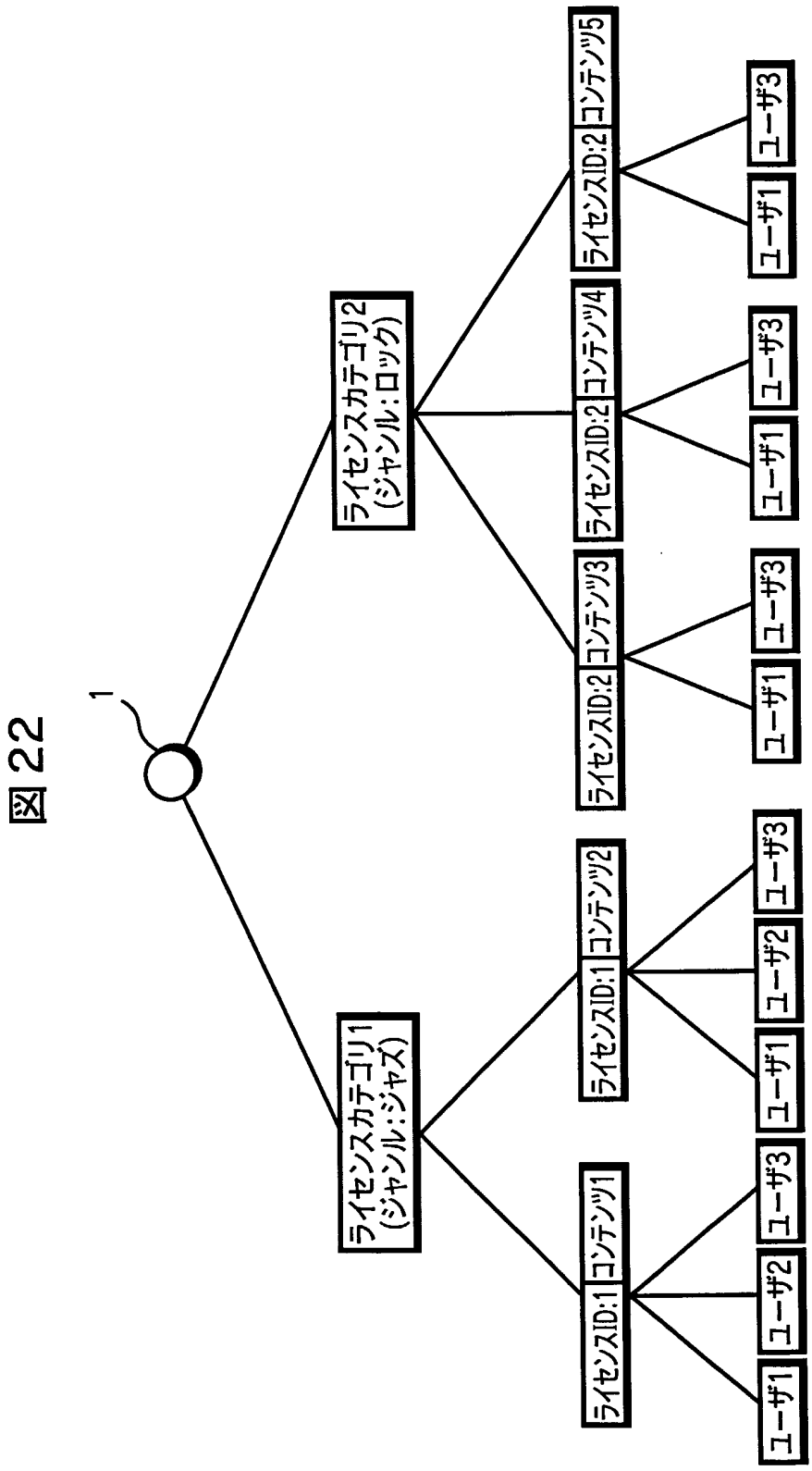

1
ライセンスカテゴリ1
(ジャンル:ジャズ)
ライセンスカテゴリ2
(ジャンル:ロック)
ライセンスID:1
コンテンツ1
ライセンスID:1
コンテンツ2
ライセンスID:2
コンテンツ3
ライセンスID:2
コンテンツ4
ライセンスID:2
コンテンツ5
ユーザ1
ユーザ2
ユーザ3
ユーザ1
ユーザ2
ユーザ3
ユーザ1
ユーザ3
ユーザ1
ユーザ3
ユーザ1
ユーザ3

図23

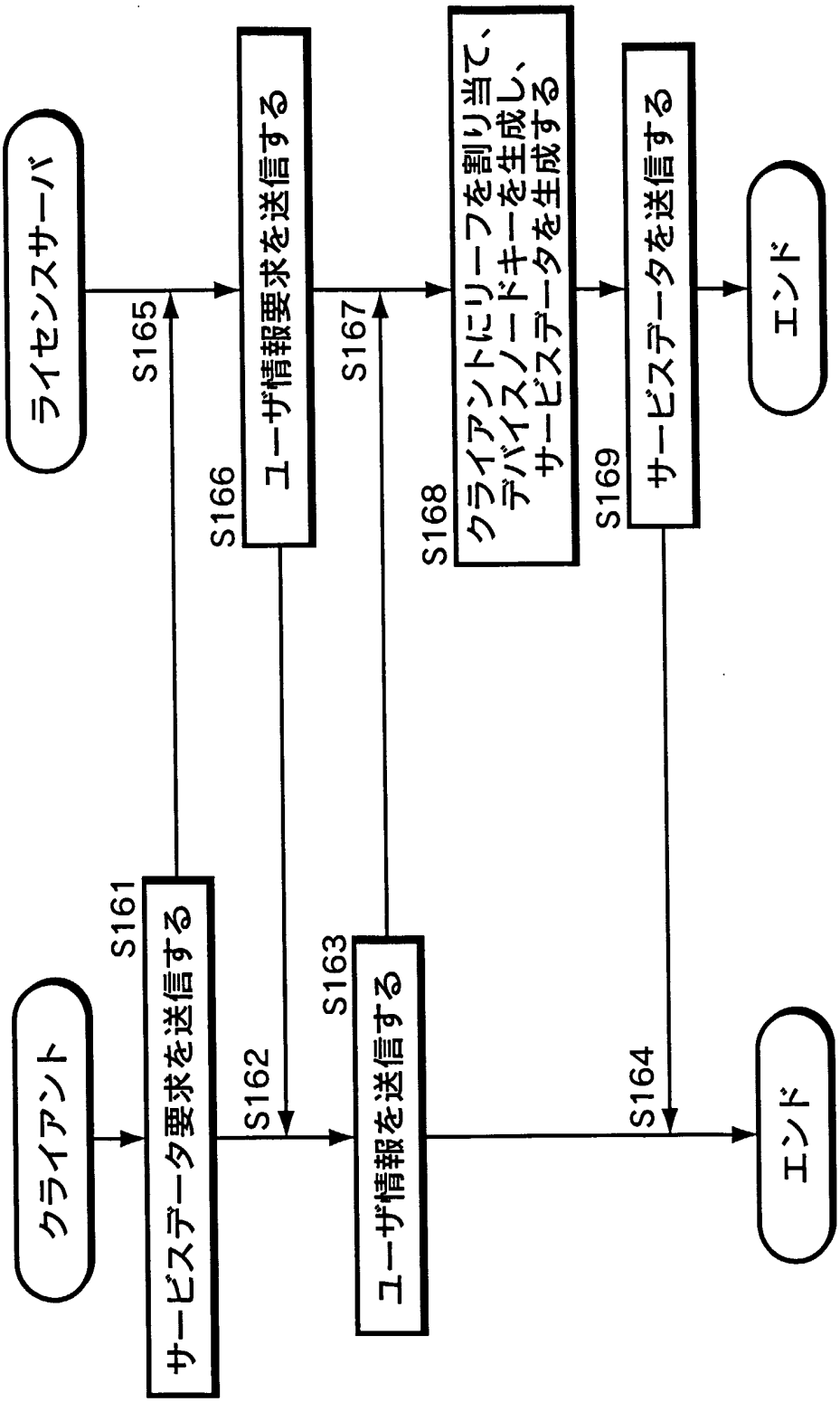

クライアント
ライセンスサーバ
サービスデータ要求を送信する
S161
S162
ユーザ情報要求を送信する
S163
S164
S165
S166
S167
クライアントにリーフを割り当て、デバイスノードキーを生成し、サービスデータを生成する
S168
サービスデータを送信する
S169
エンド
エンド

図24

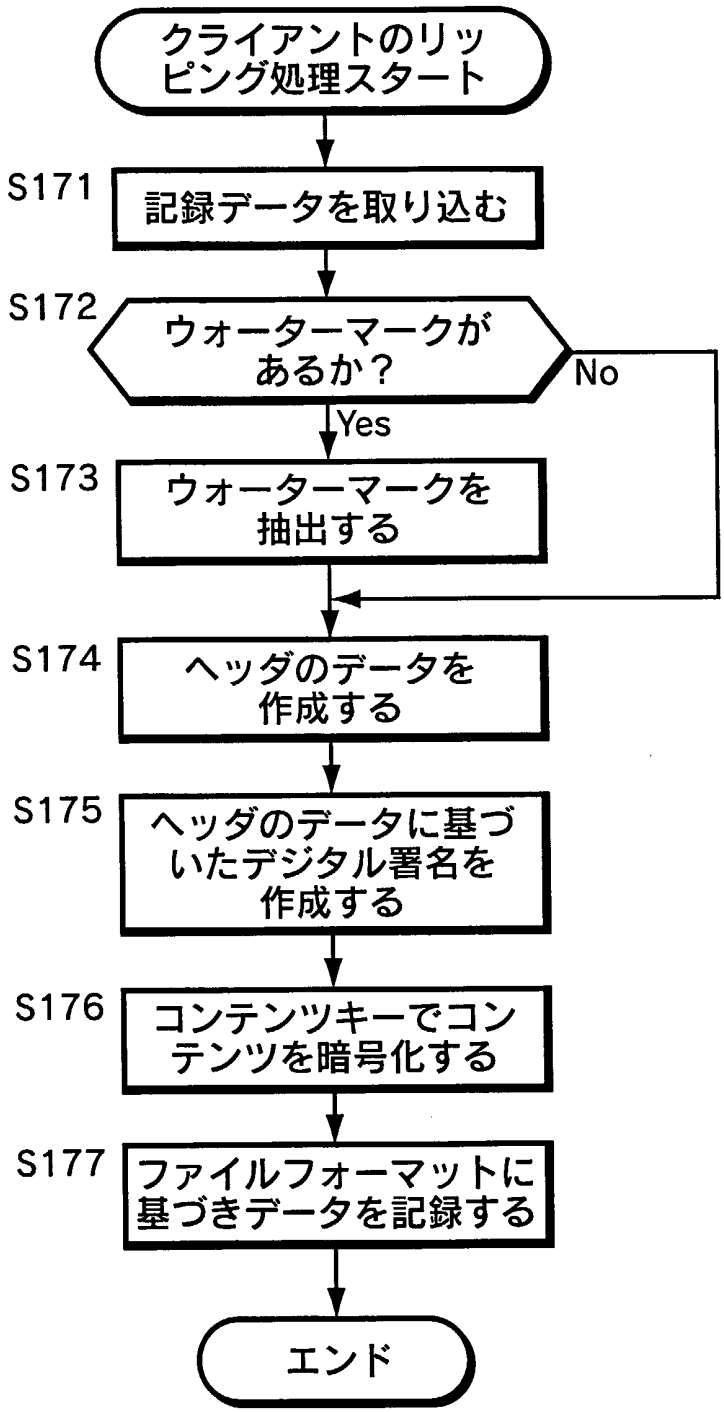

クライアントのリッピング処理スタート
S171
記録データを取り込む
S172
ウォーターマークがあるか？
No
Yes
S173
ウォーターマークを抽出する
S174
ヘッダのデータを作成する
S175
ヘッダのデータに基づいたデジタル署名を作成する
S176
コンテンツキーでコンテンツを暗号化する
S177
ファイルフォーマットに基づきデータを記録する
エンド

図25

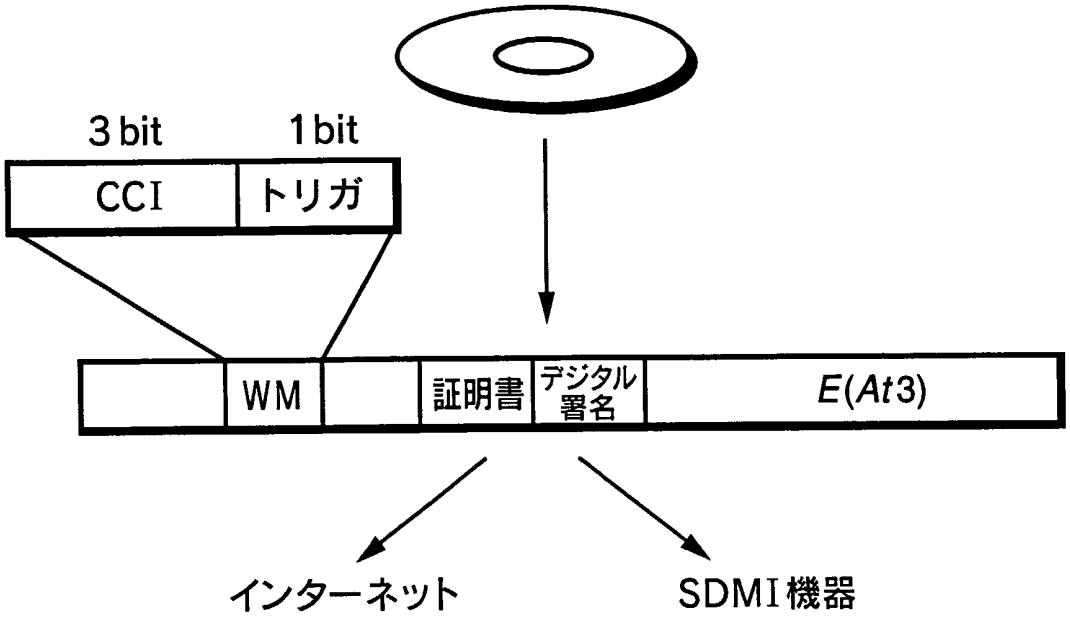

3 bit
1 bit
CCI
トリガ
WM
証明書
デジタル署名
E(At3)
インターネット
SDMI機器

図26

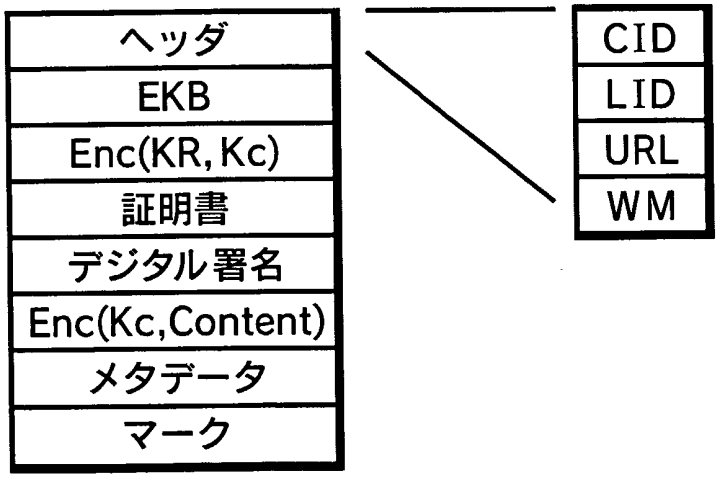

ヘッダ
EKB
Enc(KR, Kc)
証明書
デジタル署名
Enc(Kc,Content)
メタデータ
マーク
CID
LID
URL
WM

図27

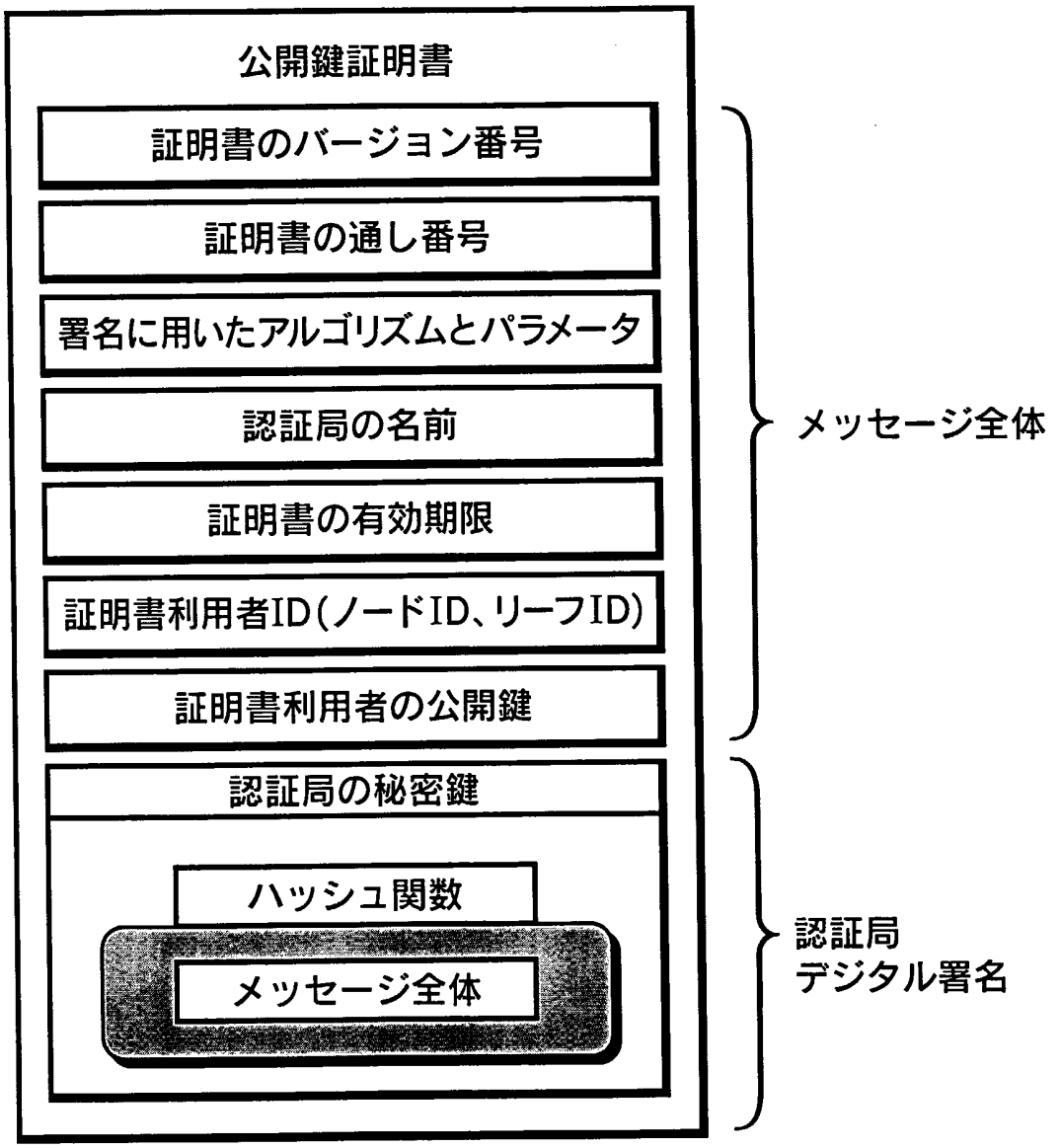

公開鍵証明書
証明書のバージョン番号
証明書の通し番号
署名に用いたアルゴリズムとパラメータ
認証局の名前
証明書の有効期限
証明書利用者ID(ノードID、リーフID)
証明書利用者の公開鍵
認証局の秘密鍵
ハッシュ関数
メッセージ全体
メッセージ全体
認証局
デジタル署名

図28

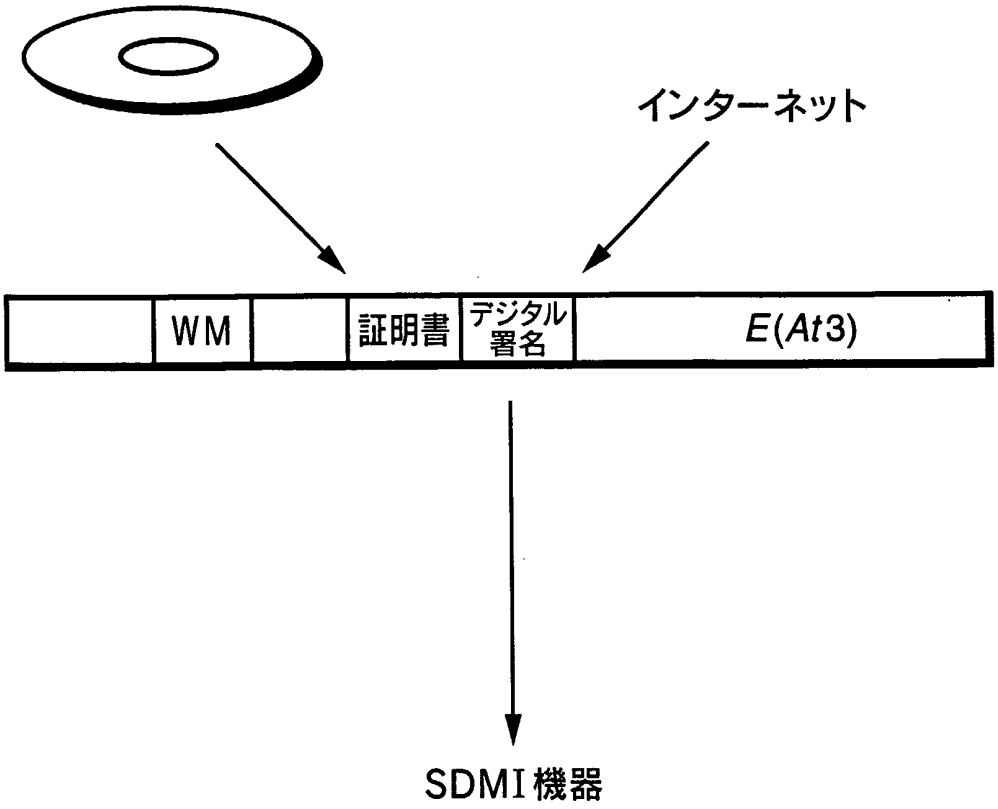


図29

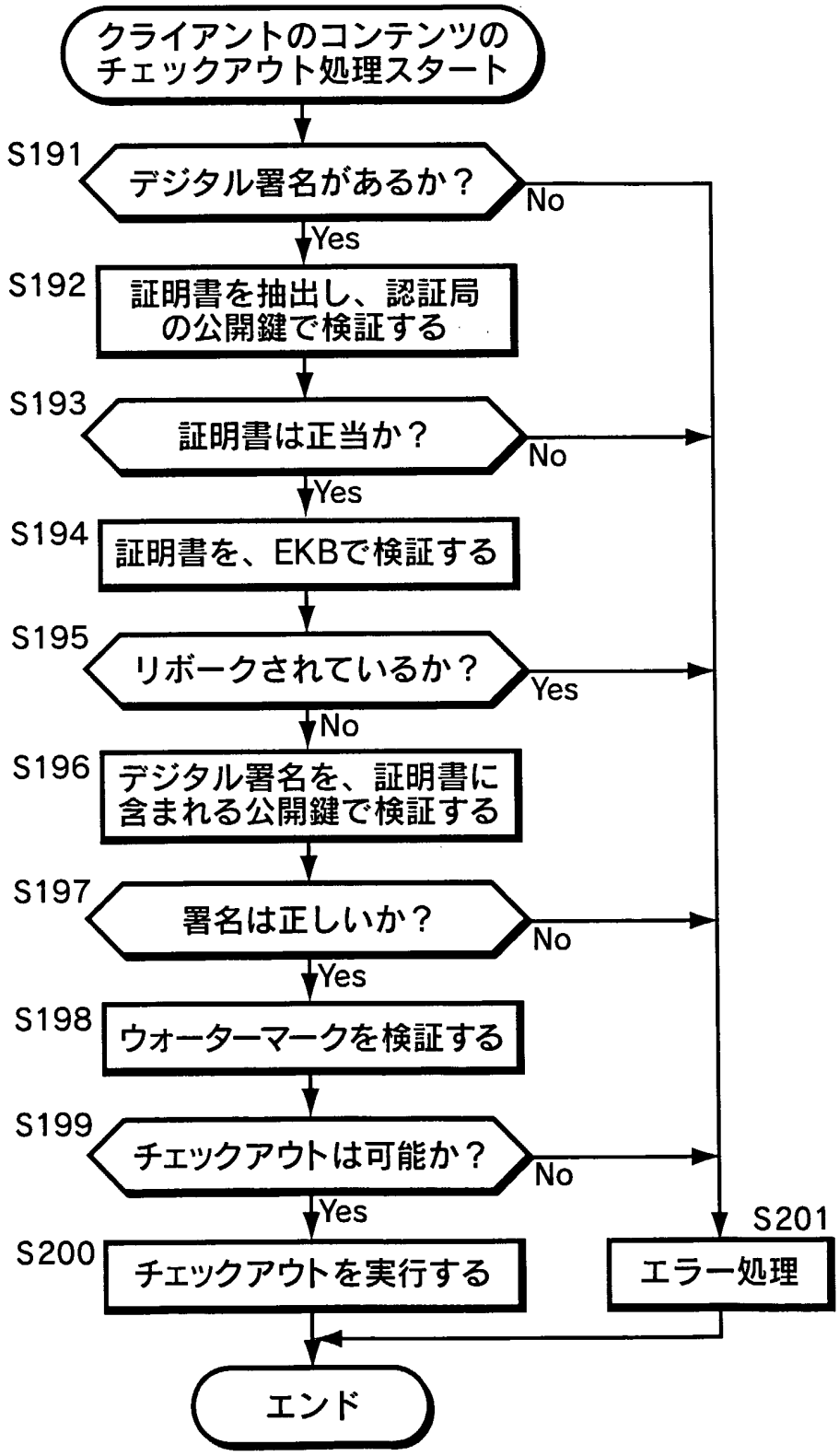

クライアントのコンテンツのチェックアウト処理スタート
S191
デジタル署名があるか？
No
Yes
S192
証明書を抽出し、認証局の公開鍵で検証する
S193
証明書は正当か？
No
Yes
S194
証明書を、EKBで検証する
S195
リボークされているか？
Yes
No
S196
デジタル署名を、証明書に含まれる公開鍵で検証する
S197
署名は正しいか？
No
Yes
S198
ウォーターマークを検証する
S199
チェックアウトは可能か？
No
Yes
S200
チェックアウトを実行する
S201
エラー処理
エンド

図30

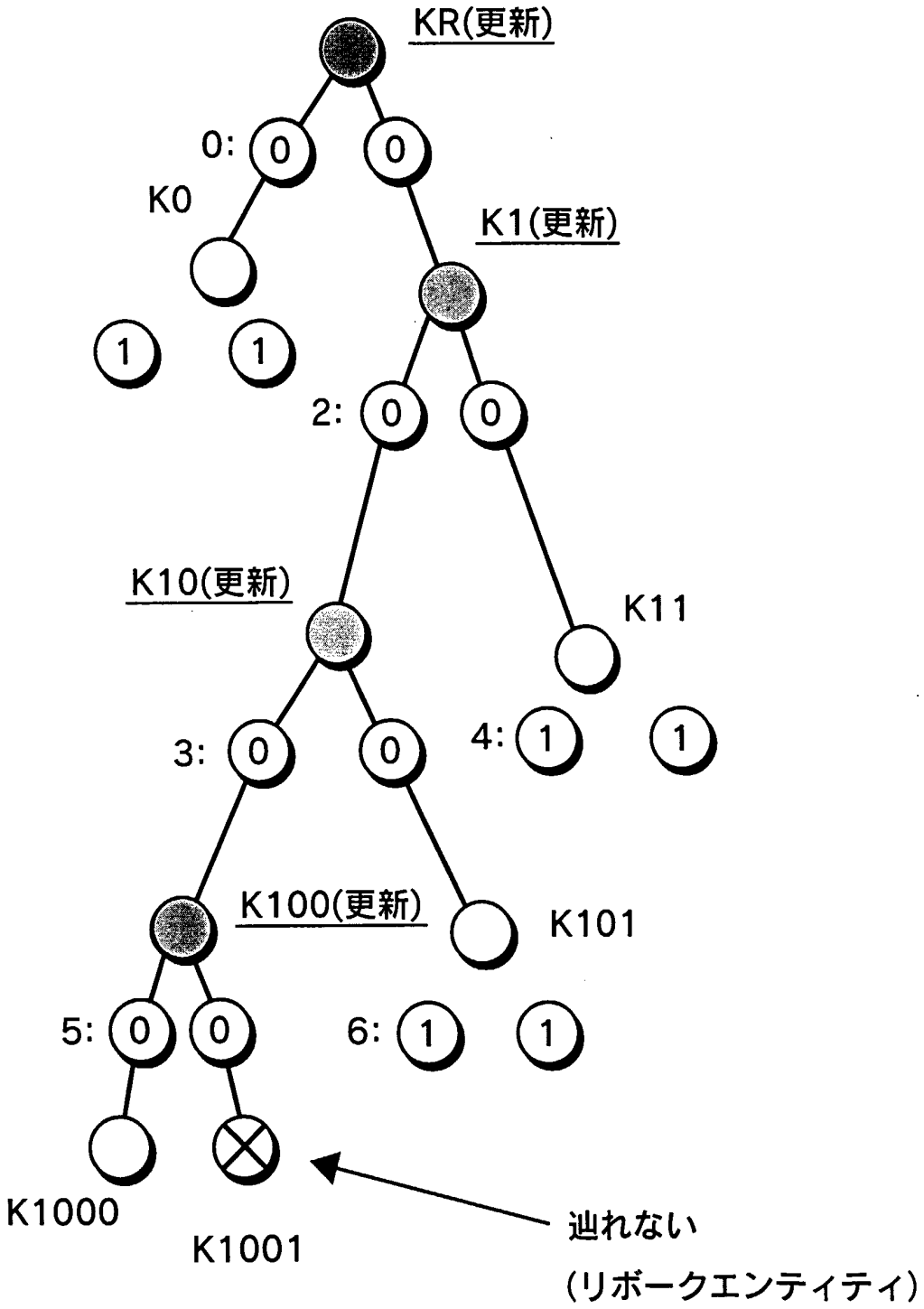
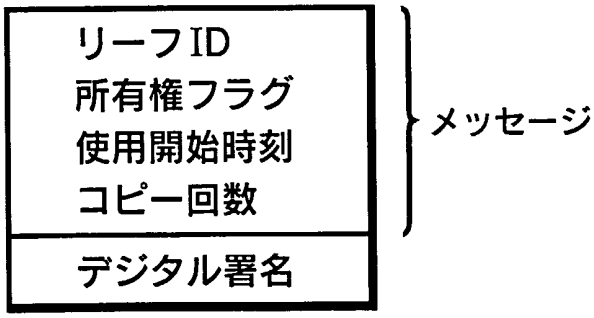

KR(更新)
0:
0
0
K0
K1(更新)
1
1
2:
0
0
K10(更新)
K11
3:
0
0
4:
1
1
K100(更新)
K101
5:
0
0
6:
1
1
K1000
K1001
辿れない
(リボークエンティティ)

図31

データ (暗号化キー)	Enc(K0, K(t)R), Enc(K(t)1, K(t)R) Enc(K(t)10, K(t)1), Enc(K11, K(t)1) Enc(K(t)100, K(t)10), Enc(K101, K(t)10) Enc(K1000, K(t)100)
タグ	0: {0, 0}, 1:{1, 1}, 2:{0, 0}, 3:(0, 0) 4: {1, 1}, 5:{0, 1}, 6:{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

図32



マーク

図33

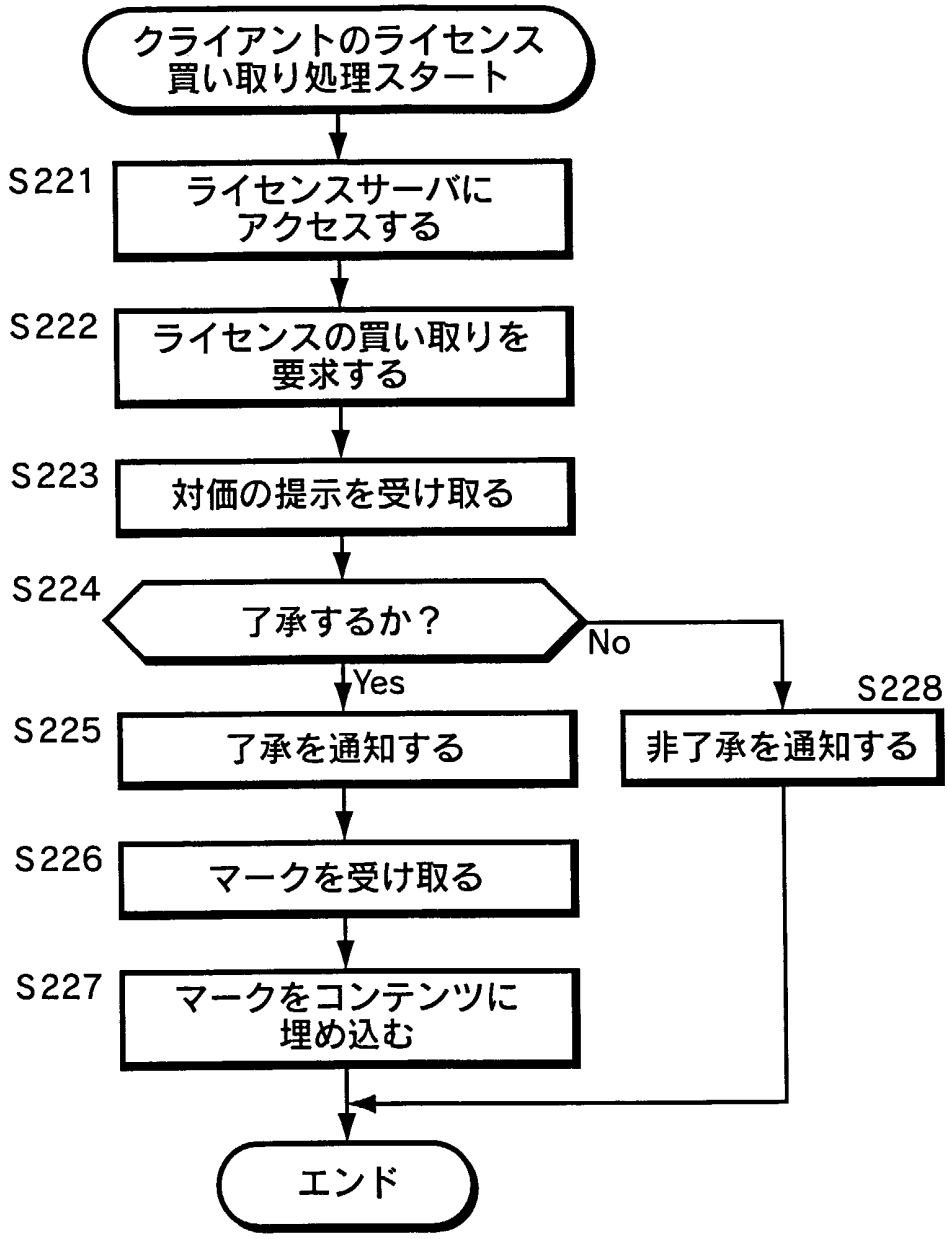

クライアントのライセンス
買い取り処理スタート
S221
ライセンスサーバに
アクセスする
S222
ライセンスの買い取りを
要求する
S223
対価の提示を受け取る
S224
了承するか？
No
Yes
S228
非了承を通知する
S225
了承を通知する
S226
マークを受け取る
S227
マークをコンテンツに
埋め込む
エンド

図34

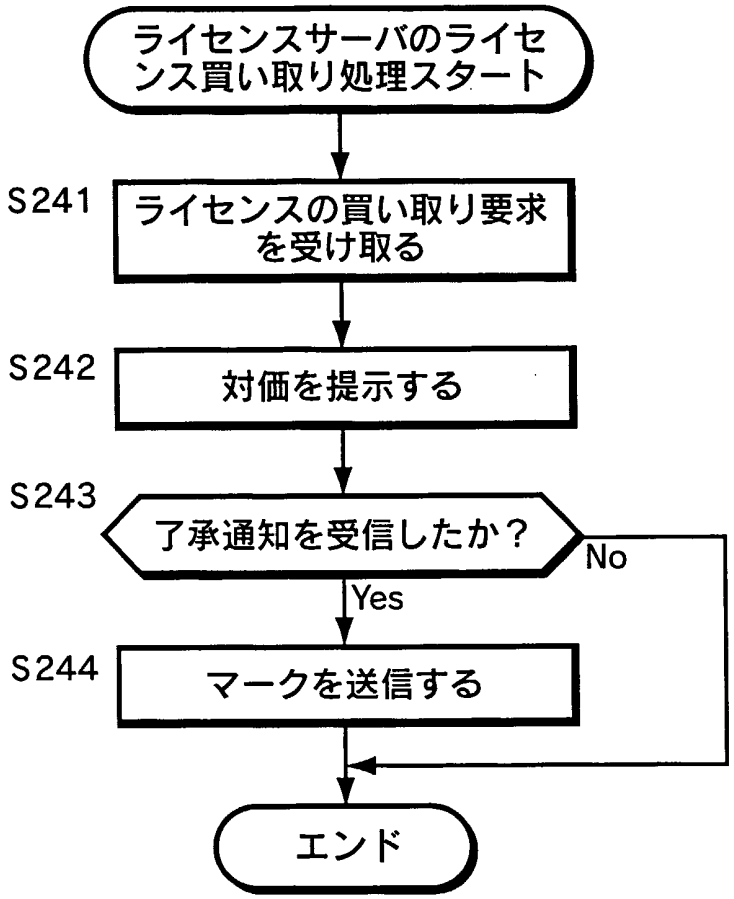

ライセンスサーバのライセンス買い取り処理スタート
S241
ライセンスの買い取り要求を受け取る
S242
対価を提示する
S243
了承通知を受信したか？
No
Yes
S244
マークを送信する
エンド

図35

Mark = {LeafID, Own, Sigs(LeafID, Own)}

図36

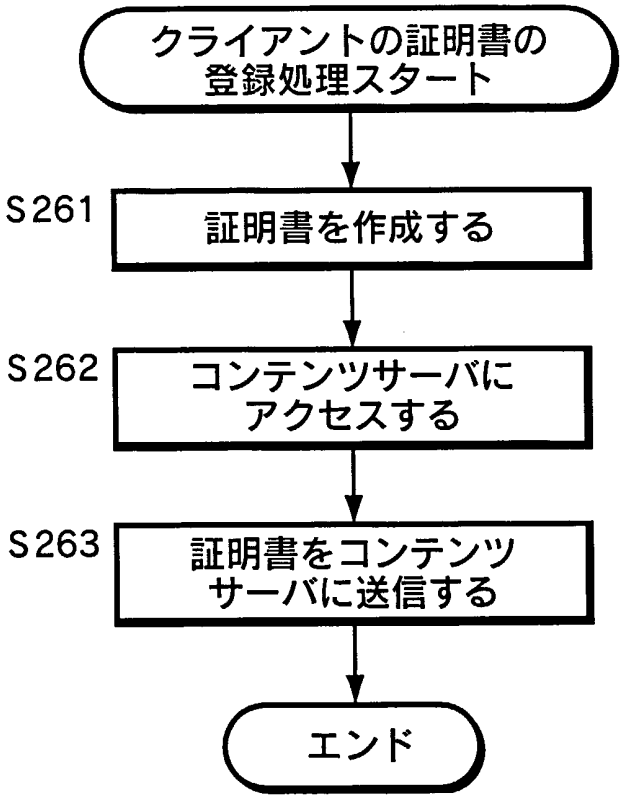


図37

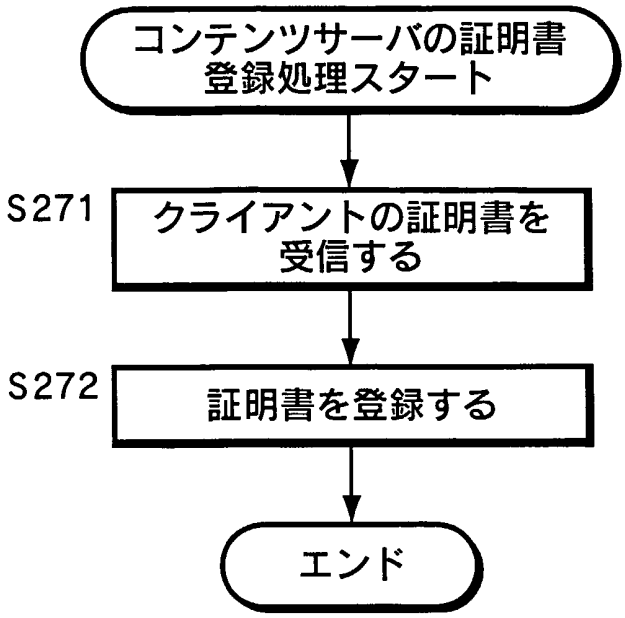

コンテンツサーバの証明書
登録処理スタート
S271
クライアントの証明書を
受信する
S272
証明書を登録する
エンド

図38

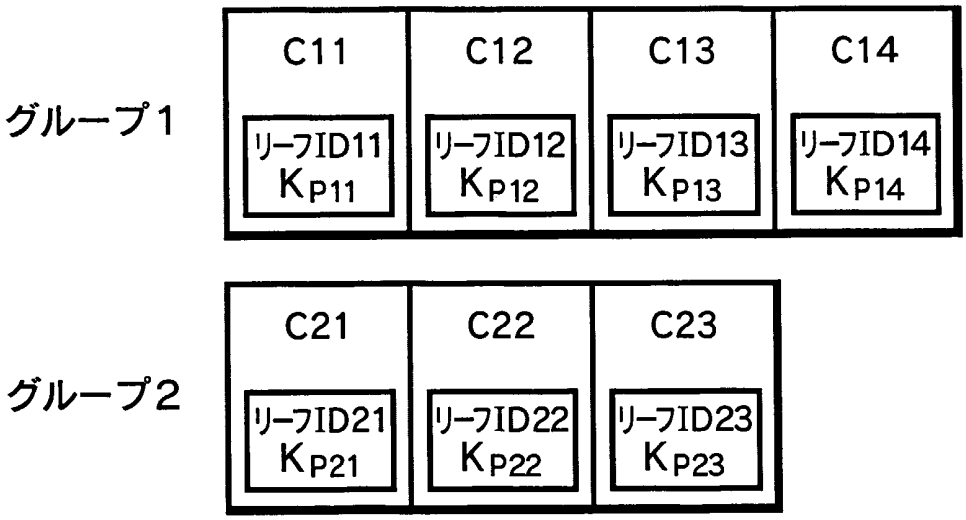

グループ1
C11
C12
C13
C14
リーフID11
KP11
リーフID12
KP12
リーフID13
KP13
リーフID14
KP14
グループ2
C21
C22
C23
リーフID21
KP21
リーフID22
KP22
リーフID23
KP23

図39

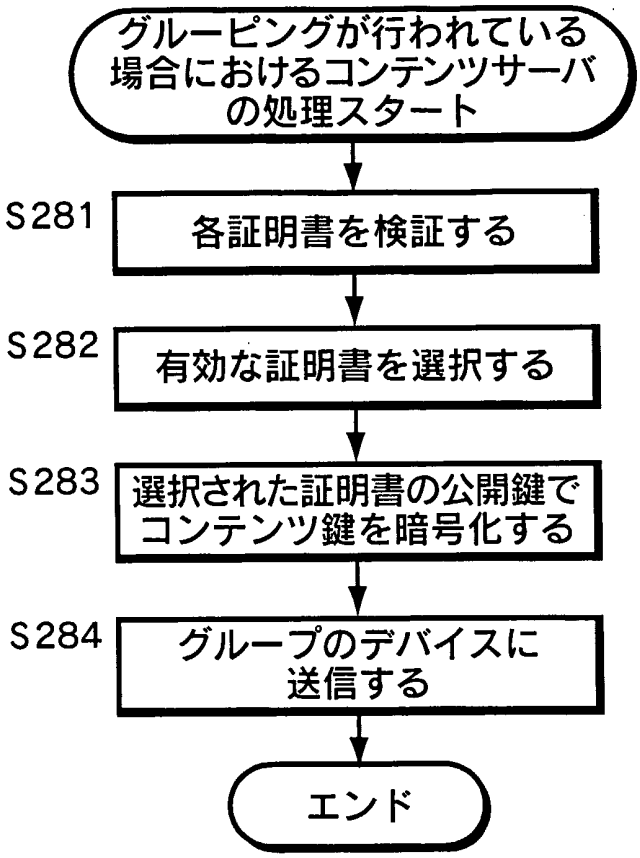

グルーピングが行われている場合におけるコンテンツサーバの処理スタート
S281
各証明書を検証する
S282
有効な証明書を選択する
S283
選択された証明書の公開鍵でコンテンツ鍵を暗号化する
S284
グループのデバイスに送信する
エンド

図40

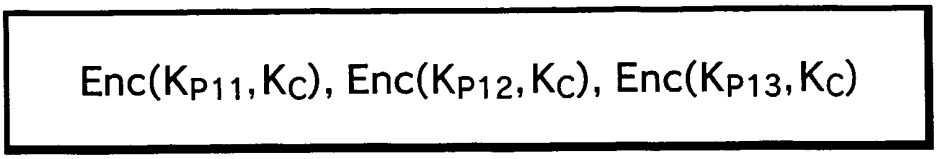

Enc(KP11, KC), Enc(KP12, KC), Enc(KP13, KC)

図41

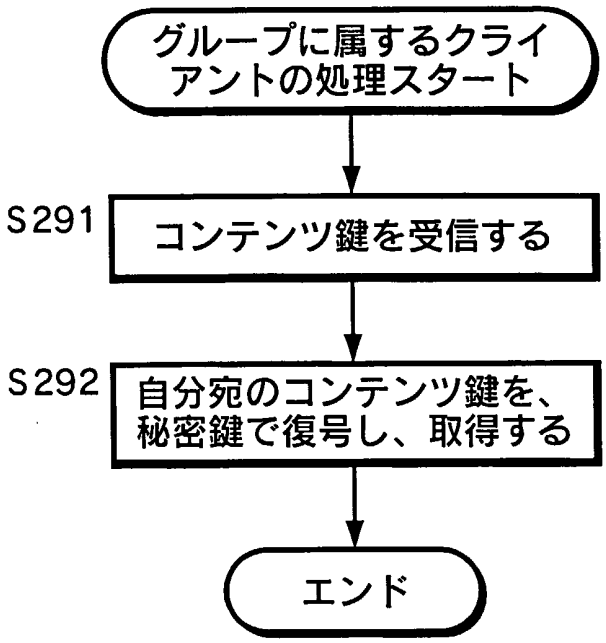

グループに属するクライアントの処理スタート
S291
コンテンツ鍵を受信する
S292
自分宛のコンテンツ鍵を、秘密鍵で復号し、取得する
エンド

図42

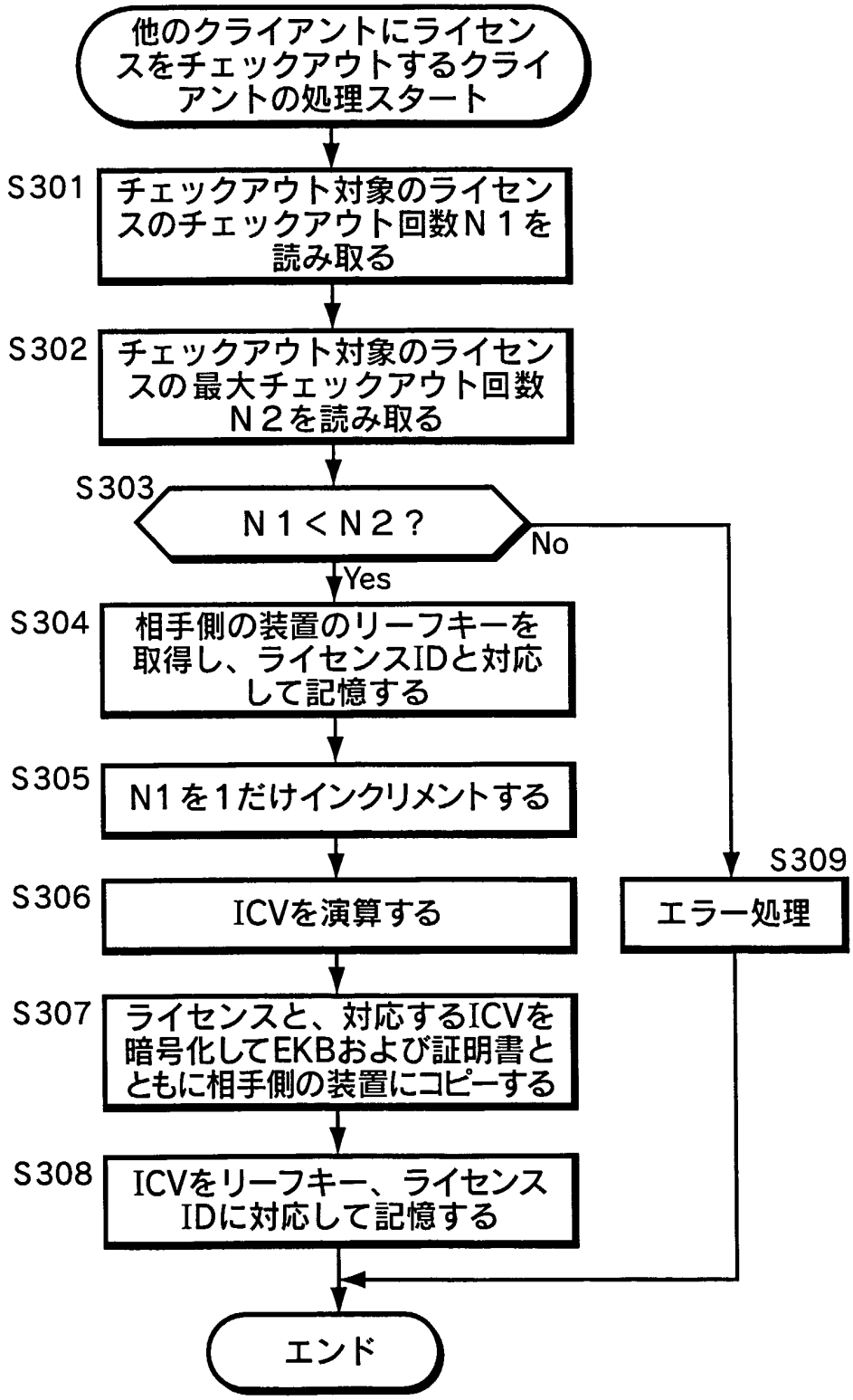

他のクライアントにライセンスをチェックアウトするクライアントの処理スタート
S301
チェックアウト対象のライセンスのチェックアウト回数Ｎ１を読み取る
S302
チェックアウト対象のライセンスの最大チェックアウト回数Ｎ２を読み取る
S303
Ｎ１＜Ｎ２？
No
Yes
S304
相手側の装置のリーフキーを取得し、ライセンスIDと対応して記憶する
S305
N1を1だけインクリメントする
S306
ICVを演算する
S307
ライセンスと、対応するICVを暗号化してEKBおよび証明書とともに相手側の装置にコピーする
S308
ICVをリーフキー、ライセンスIDに対応して記憶する
S309
エラー処理
エンド

図43

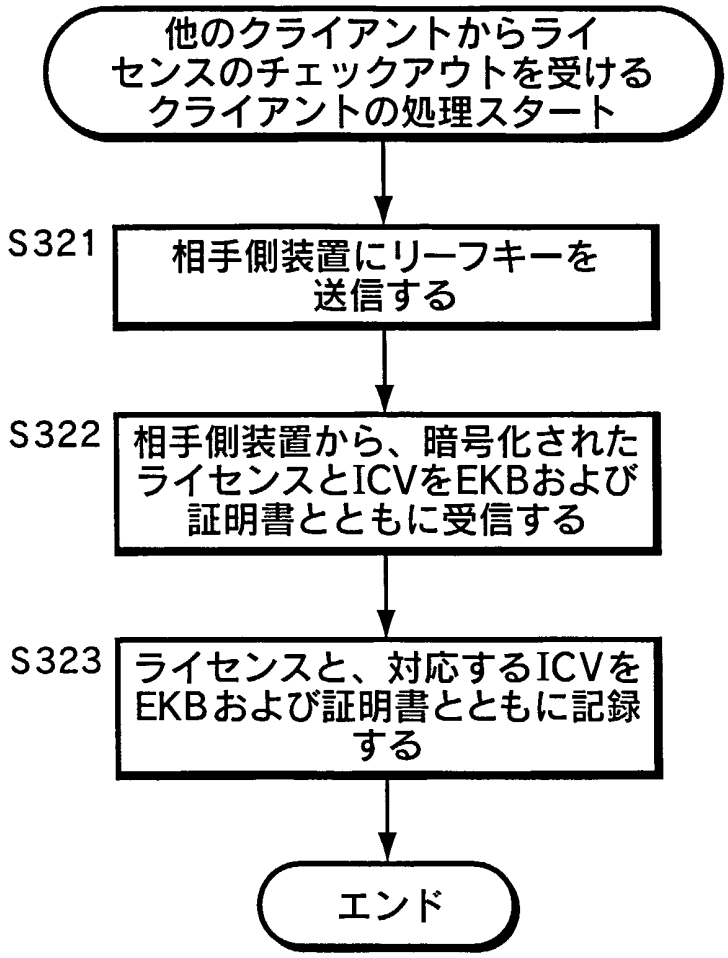

他のクライアントからライセンスのチェックアウトを受けるクライアントの処理スタート
S321
相手側装置にリーフキーを送信する
S322
相手側装置から、暗号化されたライセンスとICVをEKBおよび証明書とともに受信する
S323
ライセンスと、対応するICVをEKBおよび証明書とともに記録する
エンド

図44

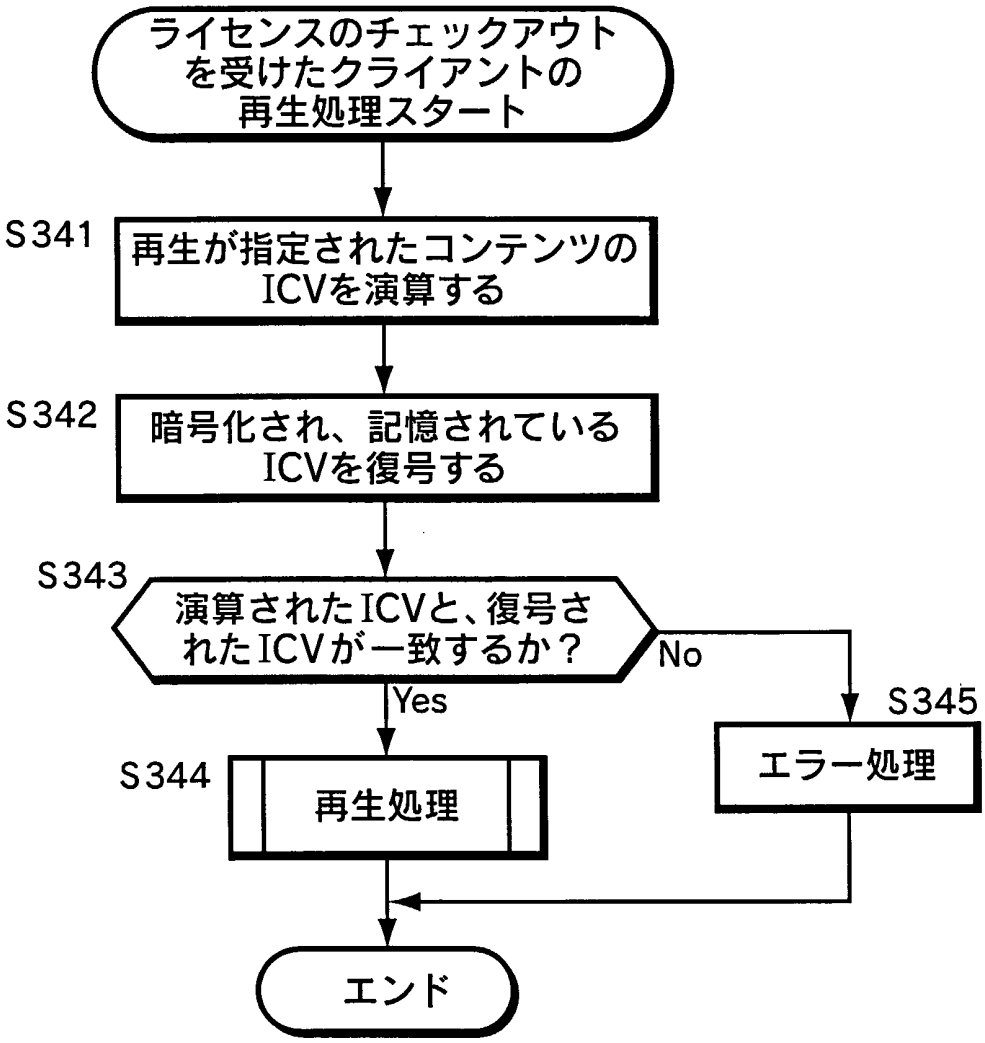

ライセンスのチェックアウトを受けたクライアントの再生処理スタート
S341
再生が指定されたコンテンツのICVを演算する
S342
暗号化され、記憶されているICVを復号する
S343
演算されたICVと、復号されたICVが一致するか？
No
Yes
S345
エラー処理
S344
再生処理
エンド

図45

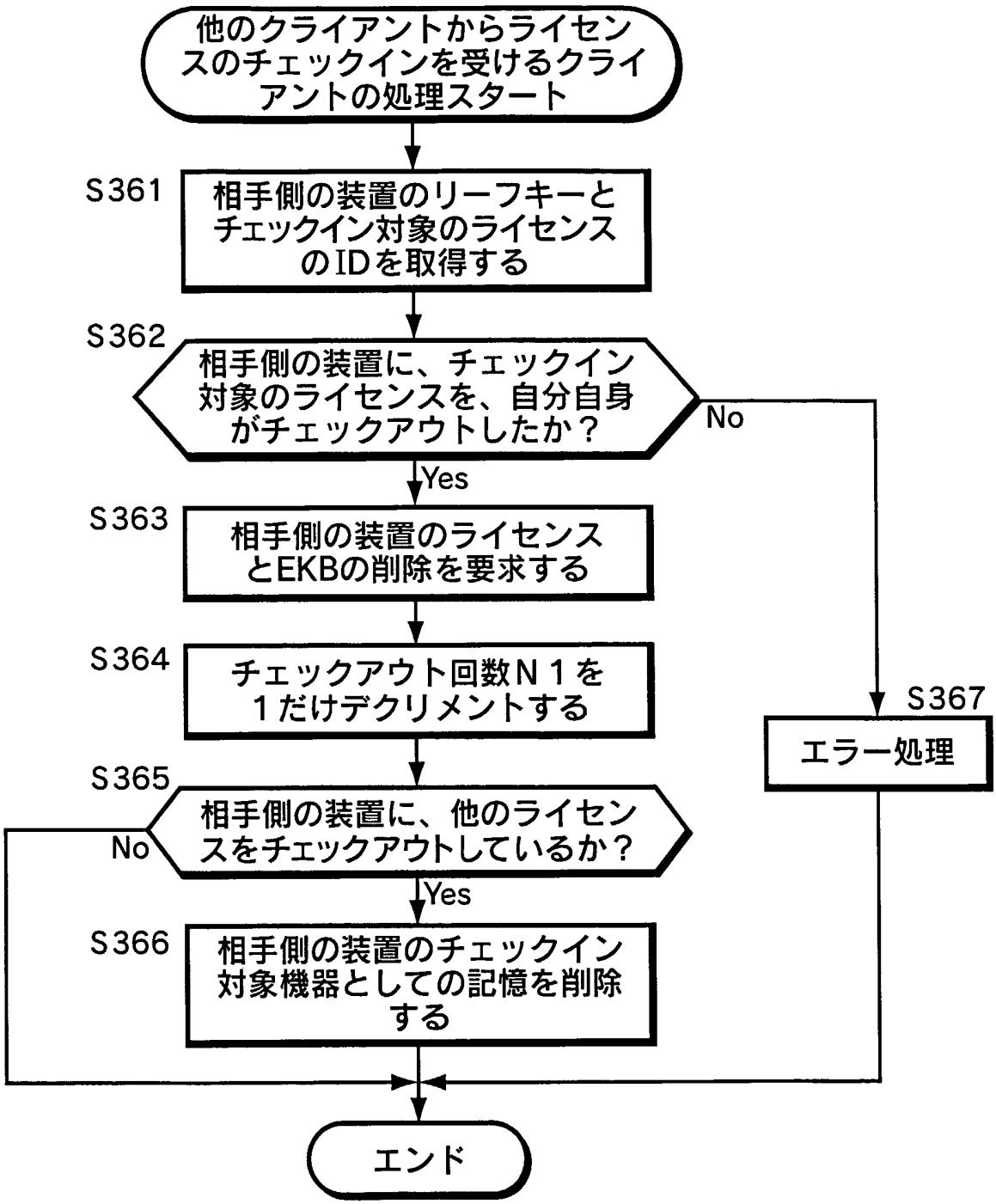

他のクライアントからライセンスのチェックインを受けるクライアントの処理スタート
S361
相手側の装置のリーフキーとチェックイン対象のライセンスのIDを取得する
S362
相手側の装置に、チェックイン対象のライセンスを、自分自身がチェックアウトしたか？
No
Yes
S363
相手側の装置のライセンスとEKBの削除を要求する
S364
チェックアウト回数Ｎ１を１だけデクリメントする
S365
相手側の装置に、他のライセンスをチェックアウトしているか？
No
Yes
S366
相手側の装置のチェックイン対象機器としての記憶を削除する
S367
エラー処理
エンド

図46

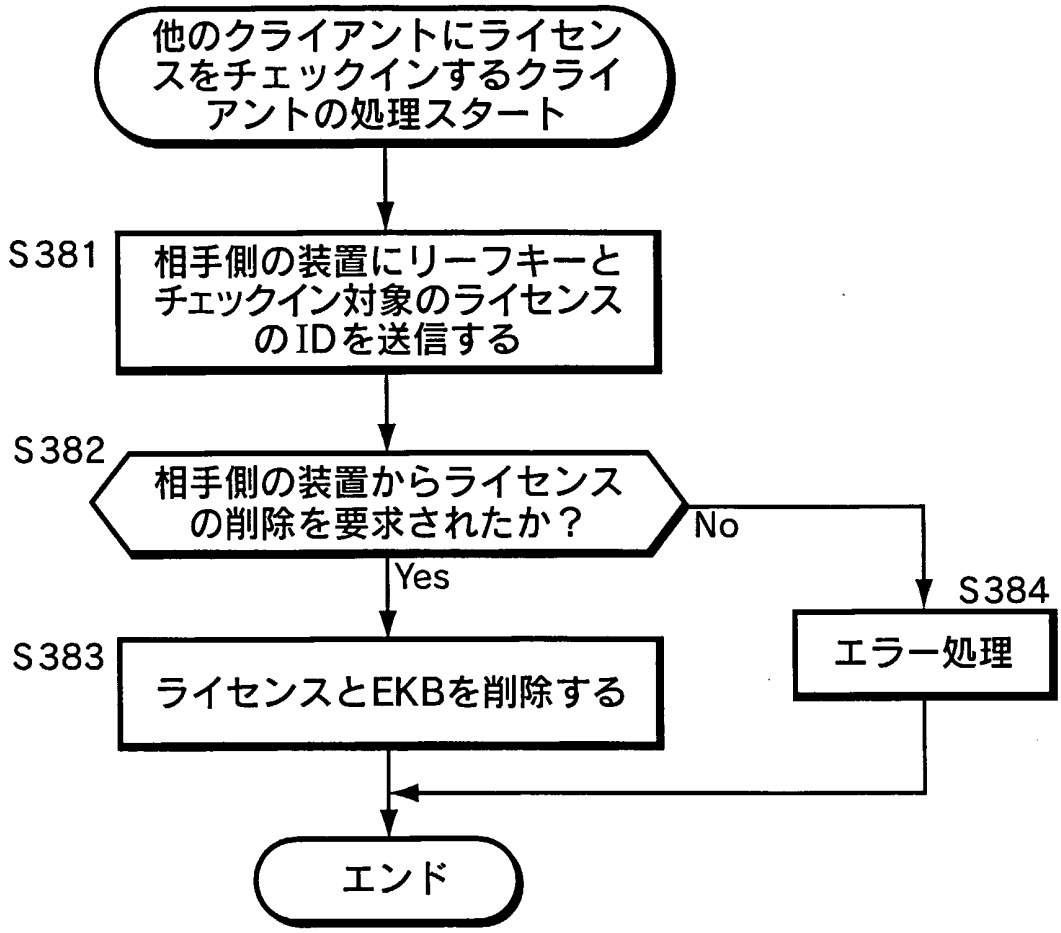

他のクライアントにライセンスをチェックインするクライアントの処理スタート
S381
相手側の装置にリーフキーとチェックイン対象のライセンスのIDを送信する
S382
相手側の装置からライセンスの削除を要求されたか？
No
Yes
S384
エラー処理
S383
ライセンスとEKBを削除する
エンド

図47

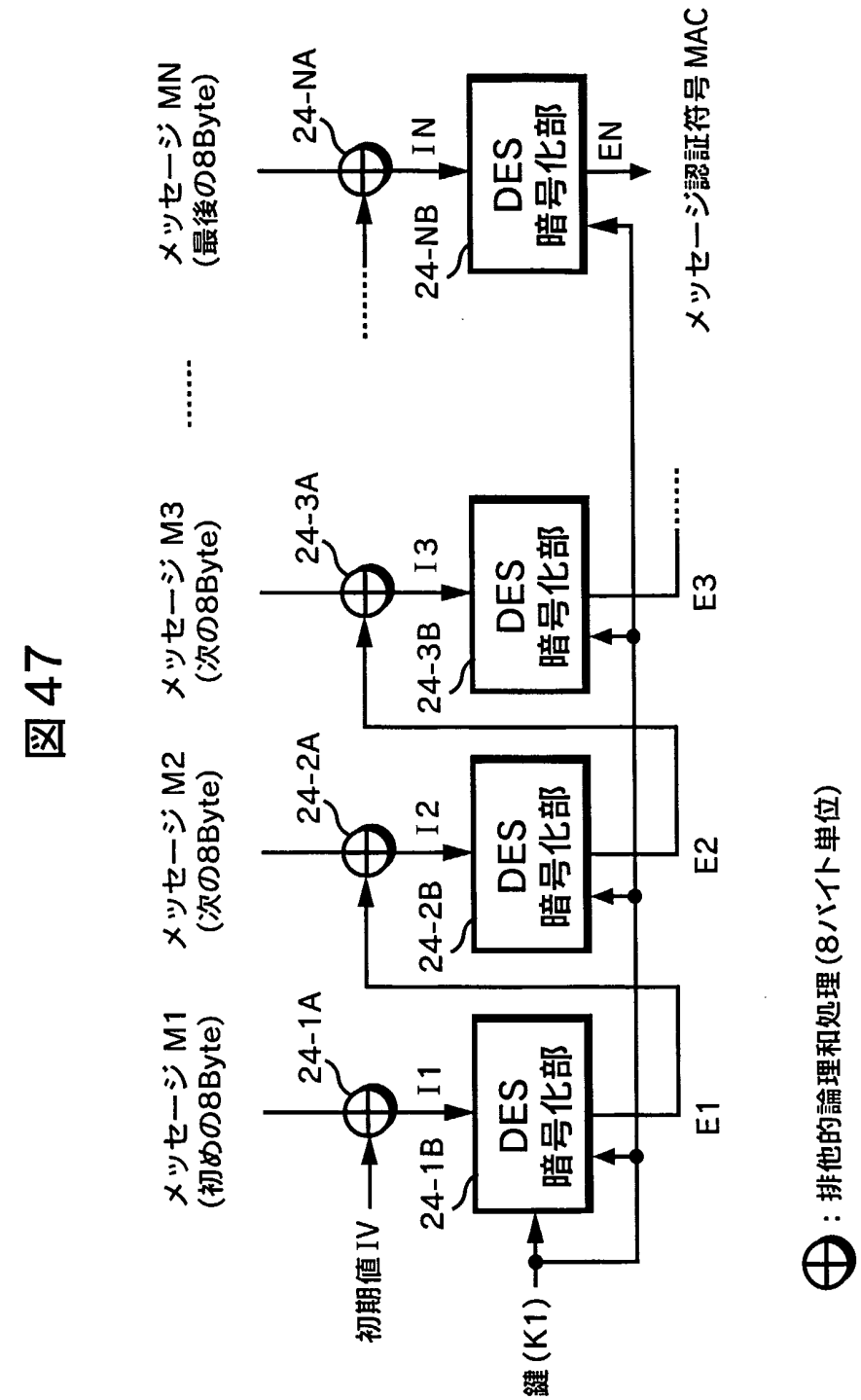

メッセージ M1
(初めの8Byte)
メッセージ M2
(次の8Byte)
メッセージ M3
(次の8Byte)
.....
メッセージ MN
(最後の8Byte)
24-1A
24-2A
24-3A
24-NA
I1
I2
I3
IN
24-1B
24-2B
24-3B
24-NB
DES
暗号化部
EN
E1
E2
E3
初期値IV
鍵(K1)
メッセージ認証符号 MAC
⊕：排他的論理和処理(8バイト単位)

図48

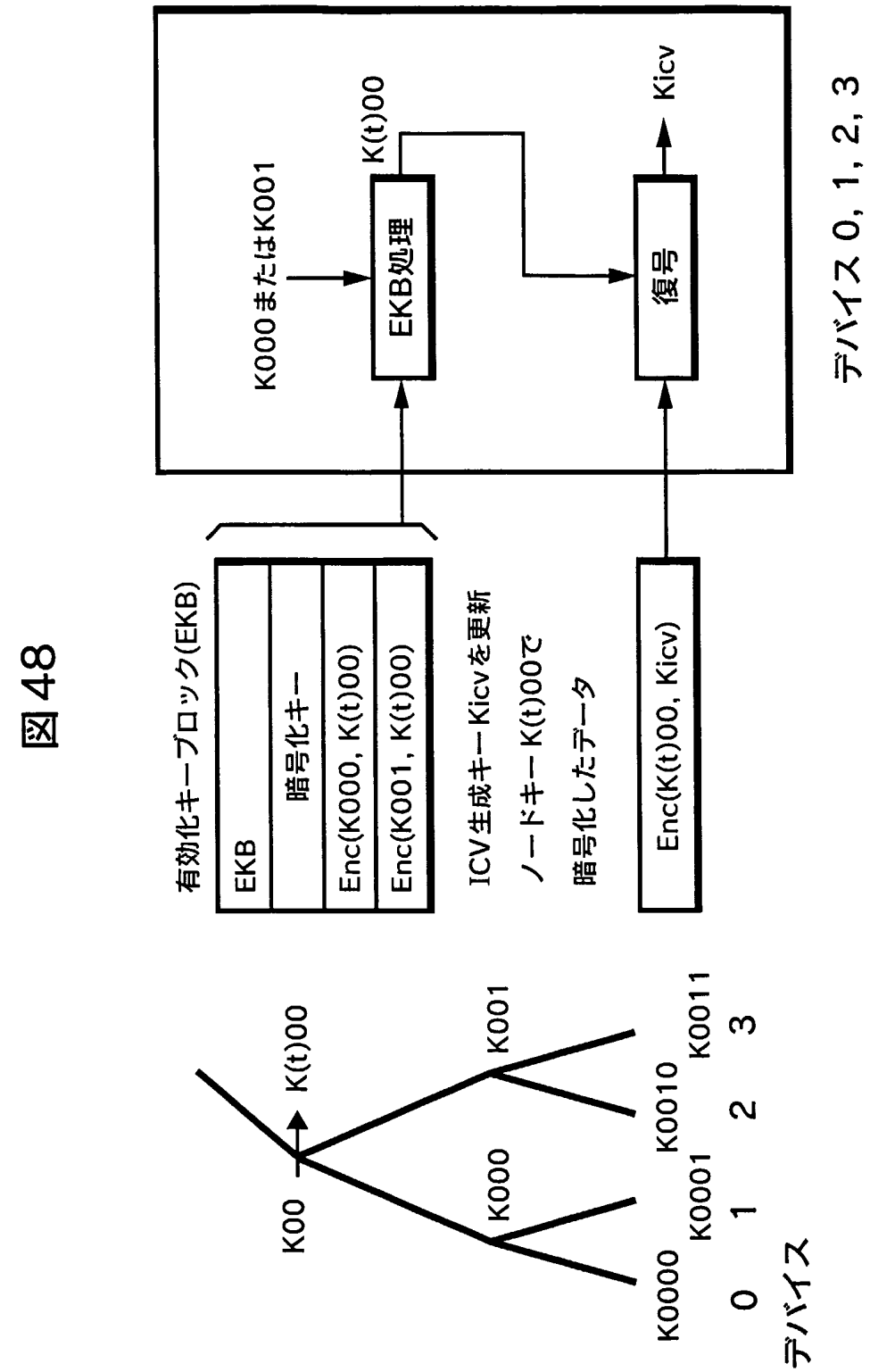
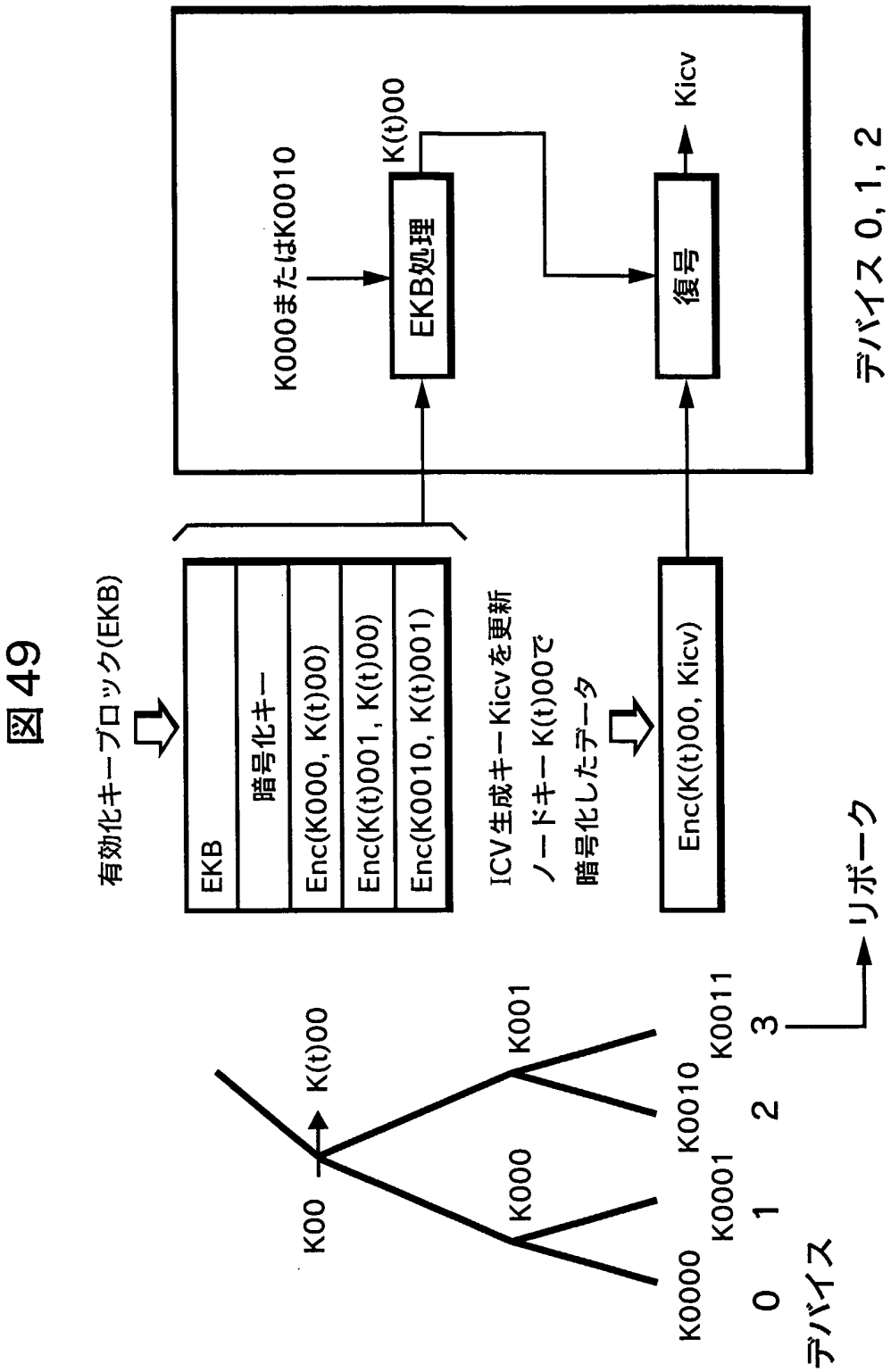

デバイス
0
1
2
3
K00
K000
K001
K0000
K0001
K0010
K0011
K(t)00
有効化キーブロック(EKB)
EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K001, K(t)00)
ICV生成キー Kicvを更新
ノードキー K(t)00で
暗号化したデータ
Enc(K(t)00, Kicv)
K000またはK001
EKB処理
K(t)00
復号
Kicv
デバイス 0, 1, 2, 3

図49


有効化キーブロック(EKB)
EKB
暗号化キー
Enc(K000, K(t)00)
Enc(K(t)001, K(t)00)
Enc(K0010, K(t)001)
K000またはK0010
EKB処理
K(t)00
Kicv
復号
デバイス 0, 1, 2
ICV生成キーKicvを更新
ノードキーK(t)00で
暗号化したデータ
Enc(K(t)00, Kicv)
リボーク
K00
K(t)00
K000
K001
K0000
K0001
K0010
K0011
0
1
2
3
デバイス

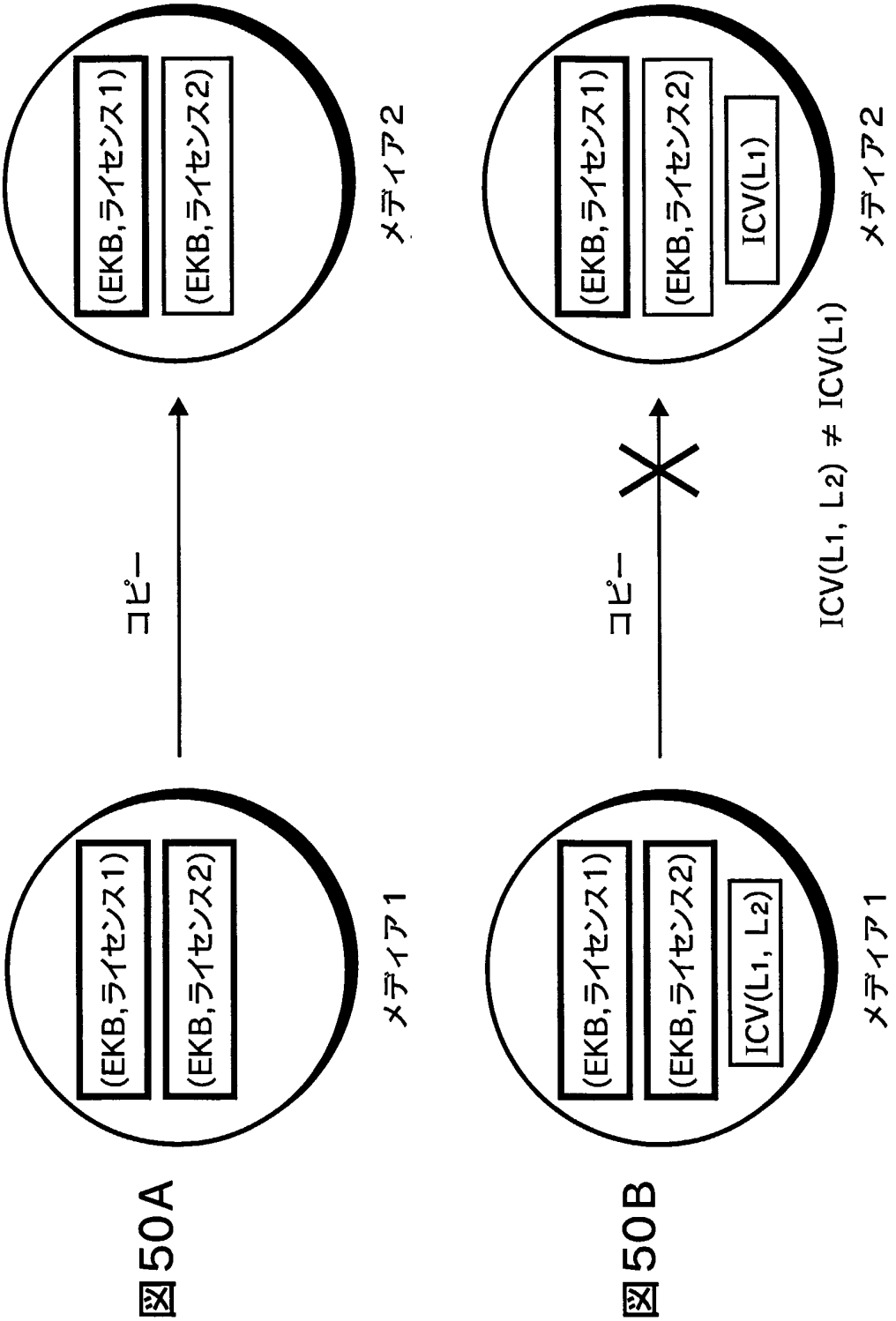
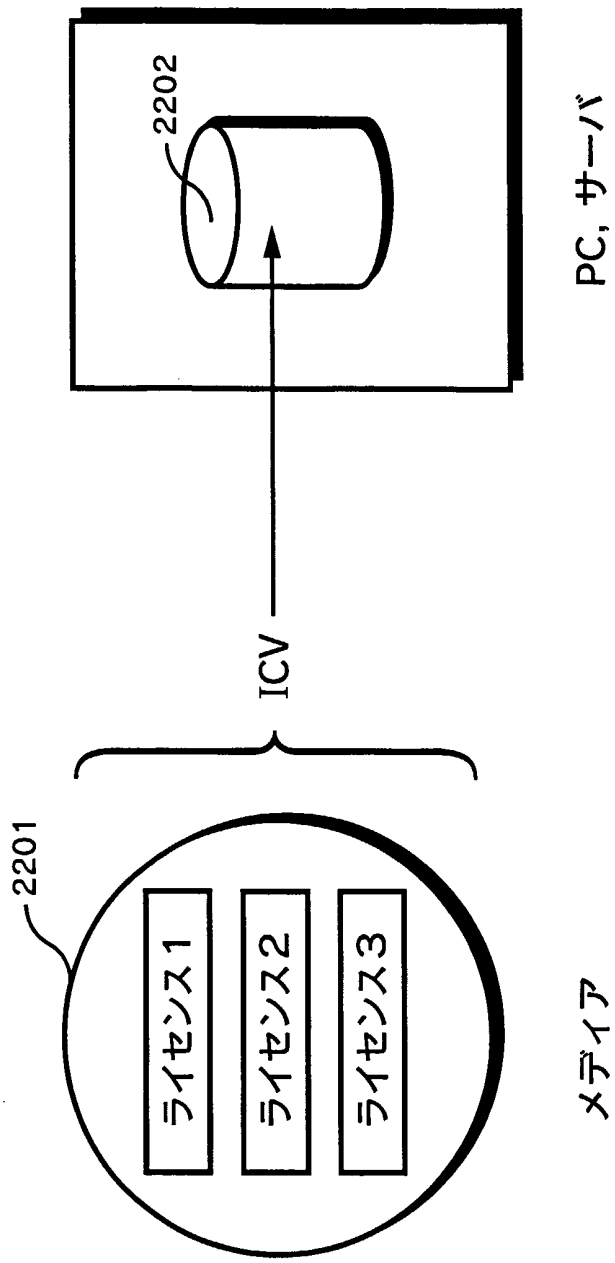

(EKB,ライセンス1)
(EKB,ライセンス2)
メディア1
コピー
(EKB,ライセンス1)
(EKB,ライセンス2)
メディア2
図50A
(EKB,ライセンス1)
(EKB,ライセンス2)
ICV(L1, L2)
メディア1
コピー
ICV(L1, L2) ≠ ICV(L1)
(EKB,ライセンス1)
(EKB,ライセンス2)
ICV(L1)
メディア2
図50B

図51



INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP02/02958

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] H04L9/08, G06F17/60

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] H04L9/08, G06F17/60

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2002
Kokai Jitsuyo Shinan Koho 1971-2002 Jitsuyo Shinan Toroku Koho 1996-2002

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JICST FILE(JOIS), WPI, INSPEC(DIALOG)
cantent, key, tree, revoking, revocation

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 11-187013 A (IBM Japan, Ltd.), 09 July, 1999 (09.07.99), Par. No. [0017] & CN 1224962 A	1-3,6-8 4,5
X Y	US 6049878 A (Sun Microsystems, Inc.), 11 April, 2000 (11.04.00), Column 7, lines 23 to 39 (Family: none)	1-3,6-8 4,5
X Y	The VersaKey Framework: Versatile Group Key Management, IEEE Journal on Selected Areas in Communications, Vol.17, No.9, pages 1614 to 1631, 1999 September, especially, B. Centralized, Tree-Based Key Management	1-3,6-8 4,5

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 July, 2002 (03.07.02)	23 July, 2002 (23.07.02)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP02/02958

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2000-244483 A (Nippon Telegraph And Telephone Corp.), 08 September, 2000 (08.09.00), Par. Nos. [0010] to [0012] (Family: none)	4,5

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０２／０２９５８

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Ｉｎｔ．Ｃｌ⁷　Ｈ０４Ｌ９／０８，Ｇ０６Ｆ１７／６０

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Ｉｎｔ．Ｃｌ⁷　Ｈ０４Ｌ９／０８，Ｇ０６Ｆ１７／６０

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００２年
日本国登録実用新案公報	１９９４－２００２年
日本国実用新案登録公報	１９９６－２００２年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

ＪＩＣＳＴファイル（ＪＯＩＳ），ＷＰＩ，ＩＮＳＰＥＣ（ＤＩＡＬＯＧ）
content, key, tree, revoking, revocation

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X	JP 11-187013 A (日本アイ・ビー・エム株式会社) 1999.07.09, 第17段落 & CN 1224962 A	1-3,6-8
Y		4,5
X	US 6049878 A (Sun Microsystems, Inc.) 2000.04.11, 第7欄第23-39行 (ファミリなし)	1-3,6-8
Y		4,5
X	The VersaKey Framework:Versatile Group Key Management, IEEE Journal on Selected Areas in Communications,	1-3,6-8
Y	Vol.17, No.9 ,p.1614-1631, 1999.09 especially B. Centralized, Tree-Based Key Management	4,5

☒ Ｃ欄の続きにも文献が列挙されている。　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー
「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの
「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「Ｏ」口頭による開示、使用、展示等に言及する文献
「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献
「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「＆」同一パテントファミリー文献

国際調査を完了した日　０３．０７．０２	国際調査報告の発送日　23.07.02
国際調査機関の名称及びあて先 日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	特許庁審査官（権限のある職員） 中里　裕正　5M　９３６４ 電話番号　０３－３５８１－１１０１　内線　３５９７

様式ＰＣＴ／ＩＳＡ／２１０（第２ページ）（１９９８年７月）

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ０２／０２９５８

C（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Y	JP 2000-244483 A（日本電信電話株式会社） 2000.09.08, 第10-12段落　(ファミリなし)	4,5

様式ＰＣＴ／ＩＳＡ／２１０（第２ページの続き）（１９９８年７月）