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

特許第4397373号
(P4397373)

(45)発行日　平成22年1月13日(2010.1.13)　(24)登録日　平成21年10月30日(2009.10.30)

(51)Int.Cl.		F I		
H04L 12/22	(2006.01)	H04L	12/22	
G06F 13/00	(2006.01)	G06F	13/00	540A
G09C 1/00	(2006.01)	G09C	1/00	640D
H04L 9/32	(2006.01)	H04L	9/00	675A
H04L 12/56	(2006.01)	H04L	12/56	300A

請求項の数 12　(全 18 頁)　最終頁に続く

(21)出願番号　特願2005-501356 (P2005-501356)
(86)(22)出願日　平成15年4月9日 (2003.4.9)
(65)公表番号　特表2006-506025 (P2006-506025A)
(43)公表日　平成18年2月16日 (2006.2.16)
(86)国際出願番号　PCT/KR2003/000713
(87)国際公開番号　WO2004/036449
(87)国際公開日　平成16年4月29日 (2004.4.29)
審査請求日　平成18年4月7日 (2006.4.7)
(31)優先権主張番号　60/418,160
(32)優先日　平成14年10月15日 (2002.10.15)
(33)優先権主張国　米国 (US)
(31)優先権主張番号　60/425,259
(32)優先日　平成14年11月12日 (2002.11.12)
(33)優先権主張国　米国 (US)

(73)特許権者　503447036
サムスン　エレクトロニクス　カンパニー　リミテッド
大韓民国キョンギ－ド，スウォン－シ，ヨントン－ク，マエタン－ドン　４１６
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(74)代理人　100107766
弁理士　伊東　忠重

最終頁に続く

(54)【発明の名称】メタデータの管理方法

(57)【特許請求の範囲】
【請求項１】
メタデータ伝送サーバーでのメタデータ管理方法において、
（ａ）伝送されるメタデータを所定の有意のセグメント単位に分割して複数のフラグメントデータを生成する段階と、
（ｂ）前記生成された複数のフラグメントデータのうち、所定のフラグメントデータを選択する段階と、
（ｃ）前記選択されたフラグメントデータからメタデータ関連情報を生成する段階と、
（ｄ）前記選択されたフラグメントデータと前記生成されたメタデータ関連情報とを、前記メタデータ関連情報を生成するために使用されたフラグメントデータのタイプを表すデータフォーマット情報と共に伝送する段階とを含むことを特徴とする管理方法。
【請求項２】
前記選択されたフラグメントデータ、前記生成されたメタデータ関連情報、及び前記フラグメントデータのフォーマット情報は、一つのメタデータコンテナに挿入されて伝送されることを特徴とする請求項１に記載の管理方法。
【請求項３】
前記データフォーマット情報は、メタデータ関連情報生成のために使用されたフラグメントデータが２進ＸＭＬフォーマットであるか、またはテキストＸＭＬフォーマットであるかを表すことを特徴とする請求項１に記載の方法。
【請求項４】

前記メタデータフラグメントデータは、メタデータの意味あるセグメント単位であることを特徴とする請求項１に記載の方法。
【請求項５】
前記メタデータコンテナには、メタデータの認証レベルを特定する認証レベルフラグが更に挿入されることを特徴とする請求項２に記載の方法。
【請求項６】
前記メタデータ関連情報は、前記選択されたフラグメントデータを一方向関数に入力して得られたメタデータダイジェスト情報であることを特徴とする請求項１に記載の方法。
【請求項７】
前記一方向関数は、ハッシュ関数であることを特徴とする請求項６に記載の方法。
【請求項８】
前記生成されたメタデータ関連情報と第１暗号化キーとを使用してメタデータ認証署名情報を生成する段階を更に含み、前記生成されたメタデータ認証署名情報を、前記選択されたフラグメントデータが挿入されたメタデータコンテナに挿入する段階を更に含むことを特徴とする請求項１に記載の方法。
【請求項９】
前記メタデータ認証署名情報は、前記生成されたメタデータ関連情報と第１暗号化キーとを一方向関数に入力して得られた結果値であることを特徴とする請求項８に記載の方法。
【請求項１０】
前記第１暗号化キーを、第２暗号化キーを使用して暗号化する段階と、前記暗号化された第１暗号化キーを、前記前記選択されたフラグメントデータが挿入されたメタデータコンテナに挿入する段階と、を更に含むことを特徴とする請求項９に記載の方法。
【請求項１１】
前記メタデータコンテナには、複数のフラグメントデータ及び対応する複数のメタデータ関連情報が挿入され、それぞれのフラグメントデータと対応するメタデータ関連情報はポインタ情報により連結されることを特徴とする請求項２に記載の方法。
【請求項１２】
前記メタデータコンテナには、複数のフラグメントデータと対応する複数のメタデータ関連情報及び複数の認証署名情報が挿入され、それぞれのフラグメントデータと対応するメタデータ関連情報及び認証署名情報は、ポインタ情報により連結されることを特徴とする請求項８に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、メタデータ伝送サーバー及びクライアントでのメタデータ管理方法に係り、特に、メタデータが伝送サーバーから伝送された後、クライアント装置により受信されるまでのメタデータのメッセージソース、メッセージ剛健性、及び機密性の認証と関連したメタデータ管理方法に関する。
【背景技術】
【０００２】
マルチメディアシステムで、マルチメディアコンデンツ及びそれと関連したメタデータは、サービス提供者からクライアント装置に提供される。そのようなマルチシステムの例は、データがサーバーからクライアントに伝送されるブロードキャスティングシステムまたはサーバー及びクライアントが相互作用するビデオ・オン・デマンド（ＶＯＤ）のようなサービス方式である。
【０００３】
伝送されるメタデータは、クライアント装置により多様な方式で使用される。メタデータの一つの使用例は、クライアント装置が再生、記録、伝送のような動作を行おうとするマルチメディアコンデンツを選択することである。

【０００４】
一方、最近放送システムにおける、クライアント装置で使用されるメタデータに含まれる情報は次第に豊かになり、その保安の重要性も高くなっている。したがって、受信されたメタデータの場合、メタデータが生成された後に伝送サーバーから伝送されてクライアントにより受信されるまで、ソース認証と剛健性及び機密性が維持されたかについての認証の必要性が更に高くなっているが、そのようなメタデータの認証を効果的に行うためのメタデータ管理方法が存在していなかった。
【発明の開示】
【発明が解決しようとする課題】
【０００５】
本発明は、前記のような必要性を満足するために、伝送されるメタデータの認証を効果的に行わせるためのメタデータ伝送サーバーでのメタデータ管理方法を提供することを目的とする。
【０００６】
また、本発明は、受信されたメタデータの認証を効果的に行わせるクライアントでのメタデータ管理方法を提供することを目的とする。
【課題を解決するための手段】
【０００７】
前記目的は、メタデータ伝送サーバーでのメタデータ管理方法において、伝送されるメタデータを所定の有意のセグメント単位に分割して複数のフラグメントデータを生成する段階と、前記生成された複数のフラグメントデータのうち、所定のフラグメントデータを選択する段階と、前記選択されたフラグメントデータからメタデータ関連情報を生成する段階と、前記選択されたフラグメントデータと前記生成されたメタデータ関連情報とを、前記メタデータ関連情報を生成するために使用されたフラグメントデータのタイプを表すデータフォーマット情報と共に伝送する段階とを含む管理方法により達成される。

【００１０】
本発明に係るメタデータ認証方法は、メタデータがサーバーからクライアントに伝送される間に変更如何を確認し、該当メタデータがどのサービス提供者またはメタデータコンデンツ提供者から伝送されるかを効果的に認証するための伝送サーバー及びクライアント装置でのメタデータ管理方法に関する。
【発明を実施するための最良の形態】
【００１１】
受信者がメタデータを受信する場合、受信されたメタデータの認証が行われねばならない。ここには、伝送レベルでの認証とソースレベルでの認証とに大別されうる。
【００１２】
伝送レベルでの認証は、伝送レベルでのメッセージソース、メッセージ剛健性、及びメッセージ機密性についての認証を行う。
【００１３】
伝送レベルでのメッセージソースの認証は、メッセージ、すなわち、メタデータコンデンツを生成したソースについての認証ではなく、メッセージを伝送するソースを認証することである。例えば、図１に示されたように、メタデータコンデンツ提供者１２０が存在し、ＳＫ　Ｔｅｌｅｃｏｍ　Ｃｏｒｐ．のようなサービス提供者１４０が別途に存在する場合、クライアント１６０が受信するメタデータＡについての伝送レベルでのメッセージソース認証は、サービス提供者１４０がメタデータＡをクライアント１６０に伝送したということについての認証を意味する。
【００１４】
また、伝送レベルでのメッセージ剛健性についての認証は、サービス提供者１４０からクライアント１６０に伝送されたメタデータＡが伝送中にいかなる変更もなかったという

ことを認証することである。

【００１５】

また、伝送レベルでのメッセージ機密性についての認証は、サービス提供者１４０からクライアント１６０に伝送されたメタデータＡが伝送中に公開されないということを認証することである。そのような伝送レベルでの認証は、ＴＣＰ／ＩＰプロトコルでのＳＳＬ／ＴＬＳアルゴリズム、ＩＥＥＥ１３９４プロトコルでのＤＴＣＰアルゴリズム、及びＤＶＩプロトコルでのＨＤＣＰアルゴリズムにより行われる。

【００１６】

ソースレベルでの認証は、ソースレベルでのメッセージソース、メッセージ剛健性、及びメッセージ機密性についての認証を意味する。

【００１７】

ソースレベルでのメッセージソース認証は、メッセージ、すなわち、メタデータコンデンツを生成したソースを認証することである。例えば、図１に示されたように、メタデータコンデンツ提供者１２０が存在し、ＳＫ ｔｅｌｅｃｏｍ ｃｏｒｐ．のようなサービス提供者１４０が別途に存在する場合、クライアント１６０が受信するメタデータＡについてのソースレベルでのメッセージソース認証は、メタデータコンデンツ提供者１２０が該当メタデータＡをクライアント１６０に伝送したということについての認証を意味する。

【００１８】

また、ソースレベルでのメッセージ剛健性についての認証は、メタデータコンデンツ提供者１２０からクライアント１６０に伝送されたメタデータＡが伝送中にいかなる変更もなかったということを認証することである。

【００１９】

また、ソースレベルでのメッセージ機密性についての認証は、メタデータコンデンツ提供者１２０からクライアント１６０に伝送されたメタデータＢが伝送中に公開されないということを認証することである。

【００２０】

したがって、ソースレベルでのメタデータ認証が行われる場合には、伝送レベルでのメタデータ認証が別途に行われる必要がない。

【００２１】

図２（ａ）、（ｂ）、及び（ｃ）は、それぞれのレベルにおける物理階層でのメタデータの伝達方式を説明するための図である。

【００２２】

図２（ａ）は、伝送レベルの認証時に使用される伝送パケットを示している。伝送レベルでの認証は、図２（ａ）に示されたそれぞれの伝送パケットに対して行われ、それらの伝送パケットは２進ＸＭＬタイプである。

【００２３】

図２（ｃ）は、ソースレベル認証時に使用されるメタデータを示す。図２（ｃ）のメタデータはテキストＸＭＬタイプである。

【００２４】

図２（ｂ）は、本発明の一実施例を説明するためのメタデータコンテナレベルの認証時に使用されるメタデータコンテナを示す。それらのメタデータコンテナには、メタデータの意味あるセグメント単位で挿入される。それらのメタデータコンテナの例は、図３及び図４に示されている。

【００２５】

図３は、単方向チャンネルでのメタデータコンテナレベルの認証に使用されるメタデータコンテナのフォーマットを示す図である。図３に示されたように、メタデータコンテナはヘッダー、フラグメントデータ及びメタデータ認証情報などを含み、コンテナヘッダーには、メタデータコンテナレベルでの認証のための制御情報が含まれている。

【００２６】
制御情報には、第１制御情報Ｆ＿１、第２制御情報Ｆ＿２、第３制御情報Ｆ＿３、第４制御情報Ｆ＿４、及び第５制御情報Ｆ＿５があり、それらの制御情報は、一つの信号またはフラグで形成される。
【００２７】
第１制御情報Ｆ＿１は、該当コンテナが運搬するメタデータのフラグメントデータにメタデータコンテナレベルの認証のための方法が適用されたか否かを表す認証フラグである。メタデータコンテナ認証のための方法としては、メディアアクセス制御（ｍｅｄｉａ ａｕｔｈｅｎｔｉｃａｔｉｏｎ ｃｏｄｅ：以下、ＭＡＣ）または電子署名アルゴリズム（ｄｉｇｉｔａｌ ｓｉｇｎａｔｕｒｅ ａｌｇｏｒｉｔｈｍ：以下、ＤＳＡ）などがある。
【００２８】
第２制御情報Ｆ＿２は、メタデータコンテナレベルでの認証情報を生成するために使用される特定アルゴリズムを表すための情報である。第２制御情報Ｆ＿２は、一つの２進コード集合を使用して表現でき、そのような２進コードと特定アルゴリズムとの関係はあらかじめ定義されており、サービスを提供するサーバー及びメタデータコンテナを受信するクライアントにあらかじめ知られている。
【００２９】
第３制御情報Ｆ＿３は、Ｆ＿２により特定されるアルゴリズムが、メタデータコンテナに共に含まれたフラグメントデータにどのような方式で具体的に使用されるかを表すためのデータフォーマット情報である。それは、メタデータコンテナレベル認証アルゴリズムが適用されたフラグメントデータは、テキスト形態から変換された２進ＸＭＬ形態でもあり、または元来のテキストＸＭＬでもあるためである。
【００３０】
第３制御情報Ｆ＿３が必要である理由は、本発明に係るメタデータの認証情報の生成は、メタデータをハッシュ関数に入力して得た出力値、すなわち、ハッシュ値を利用して行われるため、テキストＸＭＬデータの認証情報は、対応する２進ＸＭＬデータの認証署名情報と関連性がないためである。言い換えれば、クライアント装置で受信されたメタデータコンテナに含まれたメタデータとハッシュ値とから認証署名の有効如何を判断するには、ハッシュ値を計算するために使用されたメタデータのフォーマットを知っておらねばならないためである。
【００３１】
第４制御情報Ｆ＿４は、メタデータ認証と関連した暗号化キーの情報を意味する。暗号化キーの情報は、メタデータと共にメタデータコンテナに挿入されて伝送サーバーからクライアント装置に伝送されるか、または選択的に別途の保安チャンネルを通じて伝送サーバーからクライアント装置に伝送される。
【００３２】
第５制御情報Ｆ＿５は、適用された認証レベルを表示する認証レベルフラグであって、適用された認証レベルが伝送レベルであるか、またはソースレベルであるかを表示する。例えば、Ｆ＿５が‘０’に設定された場合には、認証レベルが伝送レベルであることを表示し、Ｆ＿５が‘１’に設定された場合には、認証レベルがソースレベルであることを表示する。そのような認証レベルフラグを使用することによって、クライアント装置のアプリケーションは、伝送されたメタデータの認証レベルが伝送レベルであるか、またはソースレベルであるかによってメタデータの信頼程度を判断して伝送されたメタデータの使用如何を決定できる。
【００３３】
また、メタデータコンテナはフラグメントデータの保存領域を含み、前記フラグメントデータ保存領域には、少なくとも一つ以上のフラグメントデータが挿入される。本実施例に係るコンテナには、メタデータの各意味のセグメント単位、例えば、一つのプログラムについてのプログラム情報のようなフラグメントデータが挿入される。しかし、選択的に

任意の単位のメタデータを運搬する場合にも適用されうる。また、関連されたメタデータは、一連のコンテナによりサービス提供者からクライアント装置に伝送される。また、一つのメタデータコンテナは、一つのメタデータフラグメントまたは複数のフラグメントを含む。例えば、一つのメタデータのフラグメントデータは、ＸＭＬツリー構造で表現された全体メタデータのうち、一つの副ツリーである。

【００３４】

また、メタデータコンテナレベルの認証情報には、メタデータダイジェスト情報と認証署名情報とがある。

【００３５】

メタデータダイジェスト情報は、フラグメントデータ保存領域に保存されたフラグメントデータのうち一つを、第２制御情報Ｆ＿２により特定されたハッシュ関数のような一方向関数に挿入して得られた結果値を意味する。それぞれのメタデータダイジェスト情報は、ポインタを使用してそれぞれの対応フラグメントデータと関連している。例えば、第１メタデータダイジェスト情報は、ポインタにより第１フラグメントデータと関連している。本実施例では、メタデータダイジェスト情報を生成するためにハッシュ関数を使用したが、選択的に、一方向関数の特性を有する所定の関数を使用してメタデータダイジェスト情報を求めることも可能である。

【００３６】

認証署名情報は、ダイジェスト情報と暗号化キーＫとが結合した値を、第２制御情報Ｆ＿２により特定されるハッシュ関数のような一方向関数に挿入して得られた結果値を意味する。メタデータダイジェスト情報と同様に、それぞれのメタデータ認証署名情報は、ポインタを使用してそれぞれの対応フラグメントデータと関連している。例えば、第１認証署名情報は、ポインタにより第１フラグメントデータと関連している。本実施例では、認証署名情報を生成するためにハッシュ関数を使用したが、選択的に、一方向関数の特性を有する所定の関数を使用して認証署名情報を求めることも可能である。

【００３７】

図４は、双方向チャンネルでのメタデータコンテナレベルの認証に使用されるＳＯＡＰエンベロープのフォーマットを示す図である。図４に示されたように、認証関連情報はＳＯＡＰヘッダーに含まれており、メタデータフラグメントデータは本文に含まれている。

【００３８】

認証関連情報のうち、‘Ａｌｇｏｒｉｔｈｍ　ＩＤ’情報、‘ＳｉｇｎａｔｕｒｅＶａｌｕｅＢａｓｅＴｙｐｅ’情報、及び‘ＫｅｙＩｎｆｏ’情報は、図３の第２制御情報Ｆ＿２、第３制御情報Ｆ＿３、及び第４制御情報Ｆ＿４に対応する。‘Ｄｉｇｅｓｔ’情報及び‘ＳｉｇｎａｔｕｒｅＶａｌｕｅ’情報は、図３のメタデータダイジェスト情報及びメタデータ認証署名情報にそれぞれ対応する。‘Ａｕｔｈｅｎｔｉｃａｔｉｏｎａｌ　Ｌｅｖｅｌ’情報は、メタデータの認証レベルを特定するための情報であって、図３の第５制御情報Ｆ＿５認証レベルフラグに対応する。

【００３９】

図３及び図４に示されたように、メタデータを意味単位のセグメンテーション単位に分割したフラグメントデータをメタデータコンテナに挿入することにより、効率的な暗号化管理及びメタデータ管理が可能となる。

【００４０】

例えば、それぞれのフラグメントデータ単位でインデクシング情報を付加することで、図５に示されたように、キャッシュ５２０に入力されたメタデータのうち、インデックスリスト保存部５２２に保存されたインデックスリストに基づいて選別されたメタデータのみを保存装置５４０に保存することが可能である。また、図４に示されたように、メタデータフラグメントデータがプログラム情報、セグメンテーション情報などの意味ある単位に分割されるため、それらそれぞれのフラグメントデータを選択的に暗号化することも可能となる。

【００４１】

図６は、図３及び図４に示されたメタデータコンテナを使用したメタデータコンテナレベル認証方法において、図１のメタデータコンデンツ提供サーバー１２０またはサービス提供サーバー１４０で行われる手続きを説明するフローチャートである。

【００４２】

段階６１０では、伝送されるメタデータを所定のセグメント単位に分割して複数のフラグメントデータを生成する。本実施例で生成されるフラグメントデータは、メタデータで、例えば、一つのプログラムについてのプログラム情報のような意味があるセグメント単位である。

【００４３】

段階６２０では、生成されたフラグメントデータのうち、所定のフラグメントデータを選択する。

【００４４】

段階６３０では、選択されたフラグメントデータをハッシュ関数、例えば、ＳＨＡ－１のような保安ハッシュアルゴリズムに挿入して得られた結果値であるメタデータダイジェスト情報を生成する。本実施例では、メッセージダイジェスト情報を生成するためにハッシュ関数を使用した。しかし、選択的に、ハッシュ関数のような一方向関数の特性を有する他の関数を使用することも可能である。

【００４５】

段階６４０では、選択されたフラグメントデータと生成されたメタデータダイジェスト情報及び選択されたフラグメントデータのフォーマットが２進ＸＭＬであるか、またはテキストＸＭＬであるかを表すデータフォーマット情報を含むメタデータコンテナを生成した後、それをクライアントに伝送する。

【００４６】

選択されたフラグメントデータタイプを表示する理由は、メタデータのフラグメントデータが同じである場合にも、段階６２０でのメタデータダイジェスト情報生成時に使用されるフラグメントデータのタイプによってメタデータダイジェスト情報が変わるためである。

【００４７】

段階６４０で生成されるメタデータコンテナの例は、図３及び図４に示されている。また、選択的に、段階６４０では、生成されたメタデータコンテナの認証フラグを設定して、該当メタデータコンテナが運搬するメタデータのフラグメントデータにメタデータコンテナレベルの認証方法が適用されたことを表す。

【００４８】

選択的に、メタデータコンテナに、メタデータダイジェスト情報生成のために使用されたアルゴリズム情報を挿入する。例えば、段階６３０でメタデータダイジェスト情報を生成するためにハッシュ関数を使用した場合、ハッシュ関数が認証情報生成アルゴリズムを使用したことを表すアルゴリズム情報を挿入する。しかし、アルゴリズム情報が伝送サーバーとクライアントとの間に知られている場合には、アルゴリズム情報はメタデータコンテナに挿入されない。

【００４９】

また、フラグメントデータタイプの情報と共にメタデータ認証レベルを特定するフラグを挿入することも可能である。メタデータ認証レベルを特定するためのフラグは、メタデータコンテナを使用したメタデータ認証が伝送レベルで行われるか、またはソースレベルで行われるかを特定する。

【００５０】

また、生成されるメタデータコンテナに挿入されるフラグメントデータが複数である場合、メタデータコンテナには、それらのそれぞれについて計算されたメタデータダイジェストが含まれ、それらのそれぞれのフラグメントデータとメタデータダイジェストとの関連関係を表すポインタ情報が共に含まれる。

【００５１】

また、生成されるメタデータコンテナに挿入されるフラグメントデータが複数である場合、メタデータコンテナには、それらのそれぞれのフラグメントデータについてのインデクシング情報が共に含まれる。

【００５２】

図７は、図３及び図４に示されたメタデータコンテナを使用したメタデータコンテナレベル認証方法において、図１のメタデータクライアントサーバー１６０で行われる低続きを説明するフローチャートである。

【００５３】

段階７１０では、メタデータコンデンツ提供サーバー１２０またはサービス提供サーバー１４０から伝送されたメタデータコンテナを受信する。

【００５４】

段階７２０では、受信されたメタデータコンテナのヘッダーに含まれた第１制御情報Ｆ＿１、すなわち、認証フラグを読み取る。

【００５５】

段階７３０では、段階７３０での認証フラグを読み取った結果、該当メタデータコンテナに挿入されたフラグメントデータについて認証方法が適用されたと判断された場合には段階７４０に進行し、認証方法が適用されていないと判断される場合には段階７４２に進行する。

【００５６】

段階７４０では、第２制御情報Ｆ＿２、すなわち、認証情報生成のために使用されたアルゴリズムを認識して、メタデータコンテナに挿入されているメタデータダイジェスト情報を生成するために適用されたアルゴリズムを読み取る。本実施例で使用された認証情報生成アルゴリズムはハッシュ関数である。一方、認証情報生成アルゴリズムが伝送サーバーとクライアントとの間にあらかじめ特定された場合には、前記アルゴリズムの読み取り段階を省略する。

【００５７】

また、第３制御情報Ｆ＿３、すなわち、メタデータフォーマット情報を認識して、メタデータコンテナに挿入されたメタデータダイジェストの計算のために使用されたフラグメントデータのフォーマットを認識する。

【００５８】

段階７４２では、メタデータコンテナレベルの認証手続きを終了する。

【００５９】

段階７５０では、メタデータで、所定のフラグメントデータとそれに対応するメタデータ関連情報とを読み取る。

【００６０】

段階７６０では、段階７４０で読み取られたフラグメントデータと、データフォーマット情報とに基づいて、段階７４０で認識されたアルゴリズム、例えば、ハッシュ関数を使用してメタデータダイジェスト情報を生成する。

【００６１】

段階７７０では、段階７６０で生成されたメタデータダイジェスト情報と、段階７５０での所定のフラグメントデータに対応するメタデータダイジェスト情報とを比較して、伝送されたメタデータの認証の有効如何を決定する。

【００６２】

また、選択的に、受信されたメタデータコンテナには、メタデータ認証レベルフラグが更に含まれており、クライアント装置のアプリケーションは、設定されたメタデータ認証レベルを読み取ることにより、メタデータの認証が伝送レベルに設定されたか、またはソースレベルに設定されたかが分かり、したがって、伝送されたメタデータの信頼程度を判断して、伝送されたメタデータの使用如何を決定することが可能となる。

【００６３】

図８は、図３及び図４に示されたメタデータコンテナを使用したメタデータコンテナレ

ベルの認証方法において、図１のメタデータコンデンツ提供サーバー１２０またはサービス提供サーバー１４０で行われる手続きを説明するフローチャートである。

【００６４】

段階８１０では、伝送されるメタデータを所定のセグメント単位に分割して複数のフラグメントデータを生成する。本実施例で生成されるフラグメント単位は、メタデータで、例えば、一つのプログラムについてのプログラム情報のような意味を有するセグメント単位である。

【００６５】

段階８２０では、生成されたフラグメントデータのうち、所定のフラグメントデータを選択する。

【００６６】

段階８３０では、選択されたフラグメントデータをハッシュ関数に挿入して得られた結果値であるメタデータダイジェスト情報を生成する。本実施例では、メッセージダイジェスト情報を生成するためにハッシュ関数を使用した。しかし、選択的に、ハッシュ関数のような一方向関数の特性を有する他の関数を使用することも可能である。

【００６７】

段階８４０では、段階８３０で生成されたメタデータダイジェスト情報と暗号化キーＫとをハッシュ関数に入力して、メタデータ認証署名を生成する。使用された暗号化キーＫは、サービス提供者に特有のものである。本実施例では、メタデータ認証情報を生成するためにハッシュ関数を使用した。しかし、選択的に、ハッシュ関数のような一方向関数の特性を有する他の関数を使用することも可能である。メタデータ認証署名を生成するために使用された暗号化キーは、更に他の暗号化キーＬを使用して暗号化される。ここで、暗号化キーＬを使用して暗号化された暗号化キーＫ値をＥ（Ｋ）という。計算された暗号化キーＥ（Ｋ）は、メタデータコンテナに挿入されて運搬されるか、または別途のチャンネルを通じてクライアント装置に伝送される。また、暗号化キーＬは、別途の保安チャンネルを使用してクライアント装置に伝送される。

【００６８】

段階８５０では、選択されたフラグメントデータと対応するメタデータダイジェスト情報、メタデータ認証署名、及び選択されたフラグメントデータのフォーマット情報を含むメタデータコンテナを生成した後、それをクライアントに伝送する。

【００６９】

段階８５０で生成されるメタデータコンテナの例は、図３及び図４に示されている。また、選択的に、段階８５０では、生成されたメタデータコンテナの認証フラグを設定して、該当メタデータコンテナが運搬するメタデータのフラグメントデータにメタデータコンテナレベルの認証方法が適用されたことを表す。

【００７０】

選択的に、生成されたメタデータコンテナにメタデータダイジェスト情報生成のために使用されたアルゴリズム情報を挿入する。

また、選択されたフラグメントデータのフォーマット情報は、メタデータダイジェスト情報及び認証情報を生成するために使用されたフラグメントデータのフォーマットが、例えば、２進ＸＭＬであるか、またはテキストＸＭＬであるかを表す。

【００７１】

また、生成されるメタデータコンテナに挿入されるフラグメントデータが複数である場合、メタデータコンテナには、それらのそれぞれについて計算されたメタデータダイジェスト情報及び認証署名情報が含まれ、それらのそれぞれのフラグメントデータとメタデータダイジェスト情報及びメタデータ認証署名情報との関連関係を表すポインタ情報が共に含まれる。

【００７２】

図９は、図３及び図４に示されたメタデータコンテナを使用したメタデータコンテナレベル認証方法において、図１のメタデータクライアントサーバー１６０で行われる手続き

を説明するフローチャートである。

【００７３】

段階９１０では、メタデータコンデンツ提供サーバー１２０またはサービス提供サーバー１４０から伝送されたメタデータコンテナを受信する。

【００７４】

段階９２０では、受信されたメタデータコンテナのヘッダーに含まれた第１制御情報Ｆ＿１、すなわち、認証フラグを読み取る。

【００７５】

段階９３０では、段階９２０で認証フラグを読み取った結果、該当メタデータコンテナに挿入されたフラグメントデータについて認証方法が適用されたと判断された場合には段階９４０に進行し、認証方法が適用されていないと判断される場合には段階９４２に進行する。

【００７６】

段階９４０では、第２制御情報Ｆ＿２、すなわち、認証情報生成のために使用されたアルゴリズムを認識して、メタデータコンテナに挿入されているメタデータダイジェスト情報を生成するために適用されたアルゴリズムを読み取る。本実施例で使用された認証情報生成アルゴリズムはハッシュ関数である。認証情報を生成するためのアルゴリズムが伝送サーバーとクライアントとの間に予め特定されている場合には、前記適用アルゴリズムの読み取り段階を省略する。また、第３制御情報Ｆ＿３、すなわち、フラグメントデータのフォーマット情報を認識して、メタデータコンテナに挿入されたメタデータダイジェスト情報の計算のために使用されたフラグメントデータのフォーマットを認識する。

【００７７】

段階９４２では、メタデータコンテナレベルの認証手続きを終了する。

【００７８】

段階９５０では、メタデータで、所定のフラグメントデータとそれに対応するメタデータ関連情報、メタデータ認証署名、及びデータフォーマット情報を読み取る。

【００７９】

段階９６０では、読み取られたフラグメントデータとデータフォーマット情報とに基づいて、段階９４０で読み取られたアルゴリズム、例えば、ハッシュ関数を使用してメタデータダイジェスト情報を生成する。

【００８０】

段階９７０では、クライアント装置に保存された暗号化キーＬを使用して暗号化されたキーＫを復号化する。暗号化キーＬは、メタデータ伝送サーバーから別途の保安チャンネルを通じて伝送されたものである。

【００８１】

段階９８０では、段階９６０で生成されたメタデータダイジェスト情報と復号化されたキーＫとからメタデータ認証署名Ｓを生成する。

【００８２】

段階９９０では、段階９８０で生成されたメタデータ認証署名情報と段階９５０で読み取られた認証署名情報とを比較して、伝送されたメタデータ認証署名の有効如何を決定する。

【００８３】

また、選択的に、受信されたメタデータコンテナには、メタデータ認証レベルを表示する認証レベルフラグが更に含まれており、クライアント装置のアプリケーションはメタデータ認証レベルを読み取って、認証レベルのタイプによってメタデータ使用如何を決定する。

【００８４】

また、別途の利用可能な剛健性の確認方法が存在する。そのような例のうち一つは、公開キーを使用した暗号作成法である。その場合、サービス提供者は、秘密キー及び公開キーからなる一対のキー、すなわち、Ｋ＿ｓ、Ｋ＿ｐを保有しており、Ｋ＿ｓを使用してメ

ッセージに署名する。ここで、Ｋ＿ｓは秘密キー、Ｋ＿ｐは公開キーを意味する。
【００８５】
クライアント装置は、信頼すべきソースを通じて公開キーを獲得することが可能である。したがって、クライアントが署名を有するメタデータコンテナを受信する場合、クライアント装置は、受信されたメタデータコンテナが伝送されるサービス提供者を確認し、確認されたサービス提供者に対応する公開キーＫ＿ｐを獲得する。クライアントは、受信された署名が有効であるか否かを確認するために公開キーを使用する。
【００８６】
以下では、メタデータの保安を維持するためのメタデータ認証要件及び方法について更に具体的に説明する。
【００８７】
メタデータについての適切な保安を維持するには、メタデータアクセス及び利用についての認証と、メタデータの剛健性及びメタデータの機密性の維持と、メタデータの部分集合及び２進フォーマット及びテキストフォーマットに対する効率的な保護が必要である。
【００８８】
すなわち、アプリケーションによるメタデータまたはメタデータの一部についてのアクセス認証は、適切な認証規則を従わねばならない。そのような認証手続きは、アプリケーション単位またはアプリケーション及びメタデータ単位で行われる。
【００８９】
また、メタデータの全体またはメタデータの一部のアクセスを通しる使用例には、視聴、変形、及び複写などがある。視聴は、アクセスを得ることとほぼ同じである最も簡単な使用例である。メタデータの変形またはローカル複写を制御するには、メタデータファイル管理システムを必要とする。また、遠隔アプリケーションでメタデータを複写すること、例えば、クライアントがメタデータをサービス提供者に伝送することは、メタデータ要請についての認証と、保安認証チャンネルを通じて要請されたデータ及びソース認証情報とを伝送することを必要とする。
【００９０】
また、メタデータについての保安を維持するには、メタデータの機密性の維持が必要である。メタデータの引導及び保存中に、メタデータのうち、一部は高い価値またはプライバシーと関連したデータを含んでいるなどの多様な理由で暗号化される必要がある。そのために、伝送レベル、すなわち、伝送中の機密性は、メタデータの伝送ユニットまたはコンテナを暗号化することにより維持されうる。また、ソースレベルでの暗号化は、伝送及び保存レベルでの機密性と関連した問題を何れも解決する。
【００９１】
以下では、条件的なアクセスシステムと関連した単方向環境及び双方向チャンネル（ＴＬＳ）でのメタデータ保安について説明する。
【００９２】
条件的なアクセスシステムと関連した単方向環境、すなわち、ブロードキャスト環境の例は、地上波放送（ＡＴＳＣ、ＤＶＢ）、衛星放送（Ｄｉｒｅｃｔ　ＴＶ）、ケーブルＴＶ、及びＩＰ－マルチキャストなどがある。それらは、トランザクションのような情報の交換のために使用される別途の復帰チャンネルを除いては、単方向チャンネルである。下記の機能は、そのような環境で支援される。
【００９３】
ハードウェア装置を備える加入された受信器と送信器との間の認証は自動的に行われる。また、受信器及び送信器は、メインブロードキャストチャンネルとは別途のチャンネルを利用してコモンシークレット共有する。ここで、コモンシークレットは、受信器と送信器のみが共有している別途のコードを意味する。パケットペイロードは、暗号化されて伝送された後、前記コモンシークレットを使用して解読するか、または前記コモンシークレットを使用して解読されたキーを使用して解読する。
【００９４】

双方向チャンネルの環境下では、ハンドシェイクプロトコルを使用し、サーバーまたはクライアントは、第３の証明書認証機関により発行された証明書を交換できるように認証される。コモンシークレットは、クライアントとサーバーとの間に共有され、その後、セッションキーが生成される。パケットペイロードは、セッションキーを使用して暗号化された後に伝送され、その後、同じセッションキーを使用して復号化される。ソース認証は、ＤＳＡまたはＭＡＣなどのようなアルゴリズムを通じて行われ得る。

【００９５】

また、双方向チャンネルの環境下では、クライアント／サーバーの認証は、信頼できる第３の機関による証明書の認証及び交換を通じて行われ、受信されたデータの認証及び伝送間の機密維持は、パケットペイロードの暗号化及びメッセージ認証により行われる。

【００９６】

そのように、メタデータ伝送と関連した保安を満足するには、送信器と受信器において相互間の認証が行われ、データ認証及びデータを暗号化して伝送することが可能であるように、コモンシークレットが安全に共有されねばならない。

【００９７】

以下では、伝送レベル及びソースレベルでのメタデータについての保護方式について記述する。

【００９８】

伝送中のメタデータ保護と関連して、送信器及び受信器の認証は伝送レベルで行われ、メタデータ認証及び機密維持は放送システムレベルで行われる。

【００９９】

例えば、単方向チャンネルの場合には、それぞれのＳＯＡＰ応答（ヘッダー＋本文）が保護単位として使用されうる。単方向である場合のそのような方法の例は、図１０に示されている。また、双方向チャンネルの場合には、データ署名情報は、ＳＯＡＰ応答を使用して伝送されうる。その場合、図１１に示されたように、署名情報はＳＯＡＰヘッダーに含まれており、データ情報は本文に含まれている。本文のデータ部分は暗号化されうる。

【０１００】

また、ソースレベルでのメタデータ保護と関連して、下記では放送装置内でのメタデータの剛健性及び機密維持及びメタデータアクセス及び使用制御について記述する。

【０１０１】

放送装置でのメタデータの剛健性及び機密維持は、認証署名をメタデータに関連付け、それを暗号化することにより可能である。また、メタデータのあらゆる部分が暗号化されるか、または剛健性が維持される必要がないということを考慮して、ポインタを使用して暗号化または認証手続きが行われたメタデータの特定部分を表す必要がある。そのような動作は、権利管理保護（ｒｉｇｈｔ　ｍａｎａｇｅｍｅｎｔ　ｐｒｏｔｅｃｔｉｏｎ：ＲＭＰ）システムによりそのようなポインタが維持されるソースレベルで行われ得る。ソースレベル署名を使用することにより、メタデータソースは実質的に認証されうる。もちろん、メタデータは、そのような情報、すなわち、ソースの認証署名をあらかじめ含んでおらねばならない。

【０１０２】

また、メタデータのアクセス及び使用制御のためには、標準アクセス及び使用権利技術及びそれについての強制が必要である。それと関連した標準技術は、ＸＭＬスキーマの一つの形態を有するか、または意味上で明確な意味を有する一つの集合のデータ要素の形態を有しうる。そのような技術の構成に関連しうる既存の道具は、ＸｒＭＬ、ＸＡＣＭＬ、及びＳＡＭＬなどがある。ライセンス技術及び利用規則はメタデータから分離されうる。

【０１０３】

また、選択的に、メタデータの使用が記述されうる部分的なメタデータの数が潜在的に多いということを考慮して、アクセス／使用制御を更に単純化するために、一旦、一つのアプリケーションのアクセスが認証される場合、該当アプリケーションの動作は、デフォルトとして使用規則を従うと見なすことも可能である。

【０１０４】
また、そのような問題と関連したものは、アクセス／使用と関連したＲＭＰシステムでのアプリケーションプログラムインターフェース（ＡＰＩ）である。それは、アクセス／使用制御情報がＴＶＡ　ＲＭＰシステムによって管理される場合に必要である。例えば、ＡＰＩはアクセスを要請及び許可し、変形、複写、及び外部に送る動作を行う。
【０１０５】
前記で説明したように、構造的なレベルで認証が行われ得る数種の認証がある。
【０１０６】
第一は、伝送レベルで行われるものであり、第二は、単方向チャンネルでのコンテナまたはＳＯＡＰメッセージのような双方向チャンネルで行われるものであり、第三は、ソースレベルで行われるものである。
【０１０７】
ソースレベルでの認証は、認証が行われるメタデータの具体的な部分についてポインタを使用して認証情報を提供する。ＳＯＡＰメッセージレベル認証の場合、認証情報は、ＳＯＡＰメッセージの本文に含まれたメタデータの一部または全部についてのポインタと共にヘッダーに含まれる。
【０１０８】
また、伝送中に単に剛健性保証のみが要求される場合、伝送レベルでの認証のみで十分である。一方、伝送独立性が必要である場合、コンテナレベルまたはＳＯＡＰメッセージレベル認証がそのような要求条件を満たし得る。コンテナまたはＳＯＡＰメッセージの本文に含まれるメタデータのサイズは、伝送パケットのサイズよりはるかに大きいため、そのような伝送レベルの認証は、システムにあたえる負荷を減らす。もちろん、保安チャンネルは、そのような場合に維持される必要がない。
【０１０９】
メタデータソースの認証のためには、コンテナ及びＳＯＡＰレベル認証が必要である。ソース認証を可能にするコンテナのシンタックスは、図１１に示された通りである。
【０１１０】
ソースと最終受信地との間の多くの中間ノードにおいても、ソース認証が行われるようにするには、各中間ノードでソース認証が保存されねばならない。
【０１１１】
更に具体的に、それぞれの中間ノードで受信されたメタデータは、以前のノードから受信された認証情報を使用して認証され、新たな認証情報が生成されて次のノードに伝達されるか、または以前のノードから伝達されたメタデータ及び全ての認証情報が次のノードに伝達される。
【０１１２】
したがって、幾つかの中間ノードを含むソースレベル認証を使用したメタデータの伝達の場合、一つのノードで以前のノードの認証情報を使用してソースレベルでの認証以後、新たな認証情報が生成されるか否かを表すフラグまたは信号が認証情報に挿入されうる。そのフラグを使用することにより、受信器は、ソース認証情報の有無によって該当メタデータを受信するか否かを決定する。
【０１１３】
本発明は、前記の実施例に限定されず、本発明の思想の範囲内で当業者による変形が可能である。
【０１１４】
本発明は、またコンピュータ可読記録媒体にコンピュータ可読コードとして具現することが可能である。コンピュータ可読記録媒体は、コンピュータシステムによって読み取られうるデータが保存されるあらゆる種類の記録装置を含む。コンピュータ可読記録媒体の例としては、ＲＯＭ、ＲＡＭ、ＣＤ－ＲＯＭ、磁気テープ、ハードディスク、フロッピー（登録商標）ディスク、フラッシュメモリ、光データ保存装置などがあり、また、キャリアウェーブ（例えば、インターネットによる伝送）の形態で具現されるものも含む。また

、コンピュータ可読記録媒体は、ネットワークに連結されたコンピュータシステムに分散されて、分散方式でコンピュータ可読コードとして保存されて実行されうる。

【産業上の利用可能性】

【０１１５】

本発明に係るメタデータ管理方法によれば、メタデータコンテナレベルでメタデータの認証を行わせることで、チャンネル環境に関係なく伝送レベルでの認証が可能であり、また、コンテナに認証のために計算されるメタデータのフォーマットを表す情報を挿入して、伝送レベル及びソースレベルでの認証が何れも選択的に可能であるようにして、伝送レベルでのパケットに比べて、メタデータコンテナレベルのパケットのサイズが比較的に大きいため、伝送されるパケットの数を減らしてシステム複雑度を減少させることが可能であるという効果がある。

【図面の簡単な説明】

【０１１６】

【図１】メタデータ認証レベルを説明するためのブロック図である。

【図２】物理階層でのメタデータの伝達方式を説明するための図である。

【図３】単方向チャンネルでのメタデータコンテナレベル認証に使用されるメタデータコンテナのフォーマットを示す図である。

【図４】双方向チャンネルでのメタデータコンテナレベル認証に使用されるＳＯＡＰメッセージを示す図である。

【図５】メタデータのインデクシング情報を利用したメタデータの分類方法を説明するための図である。

【図６】本発明の一実施例に係るメタデータ伝送サーバーでのメタデータ管理方法を説明するためのフローチャートである。

【図７】本発明の一実施例に係るメタデータクライアントでのメタデータ管理方法を説明するためのフローチャートである。

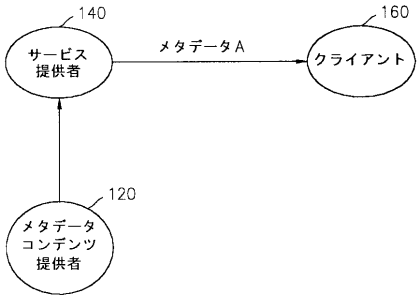
【図８】本発明の一実施例に係るメタデータ伝送サーバーでのメタデータ管理方法を説明するためのフローチャートである。

【図９】本発明の一実施例に係るメタデータクライアントでのメタデータ管理方法を説明するためのフローチャートである。

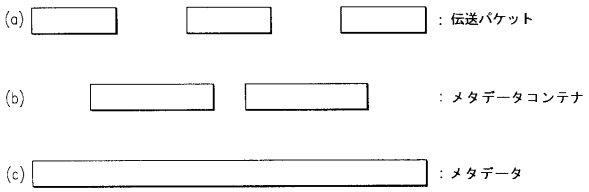
【図１０】単方向チャンネルの場合のデータコンテナフォーマットを示す図である。

【図１１】双方向チャンネルの場合のＳＯＡＰメッセージを示す図である。

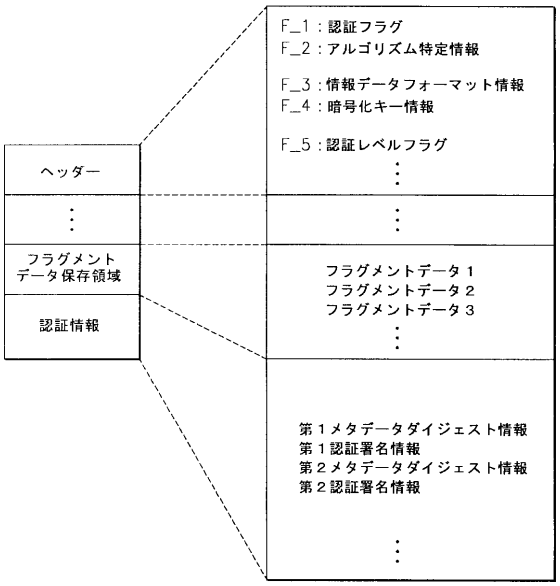
【図１】



【図２】



【図３】

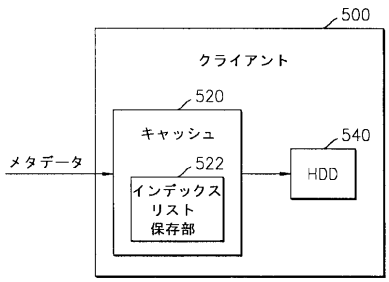


【図４】

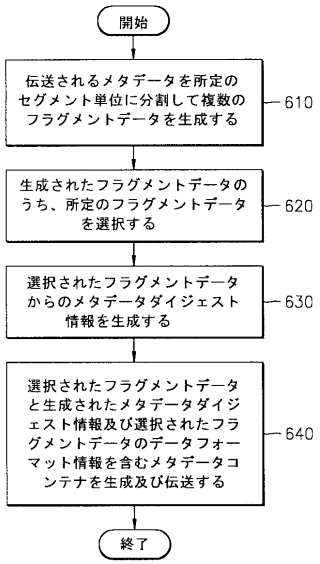
FIG. 4

```
<SOAP:Envelope ...>
  <SOAP:Header>
      <Signature fragrefID = 1>
        <Algorithm ID=1>
        <Digest> ... <\Digest>
        <SignatureValue>  ...  </SignatureValue>
        <KeyInfo>  ... </KeyInfo>
        <SignatureValueBaseType>Text</SignatureValueBaseType>
        <AuthenticatioinLevel>Transport</AuthenticationLevel>
      </Signature>
  </SOAP:Header>
<SOAP:Body>
  <TVAmetadataFragment id=1>
      <ProgramInformation>...</ProgramInformation>
  </TVAmetadataFragment>
  <TVAmetadataFragment id=2>
      <SegmentInformation>...</SegmentInformation>
  </TVAmetadataFragment>
</SOAP:Body>
</SOAP:Envelope>
```

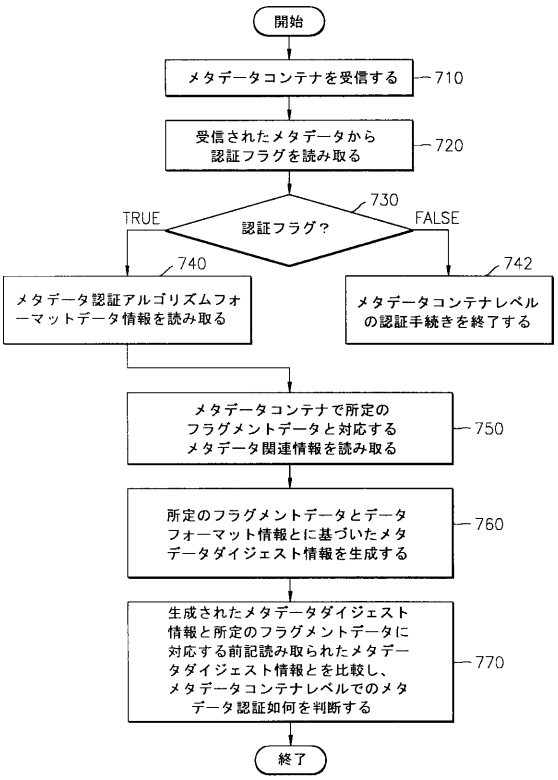
【図５】



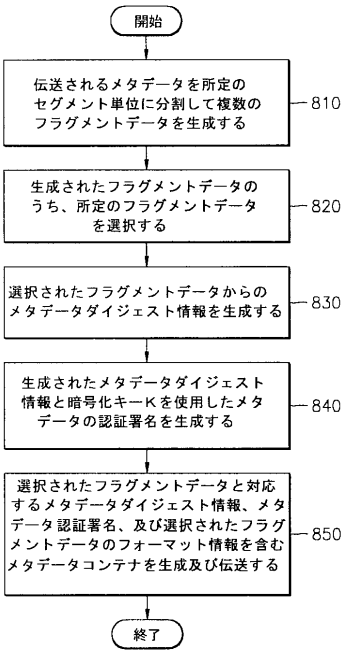
【図６】



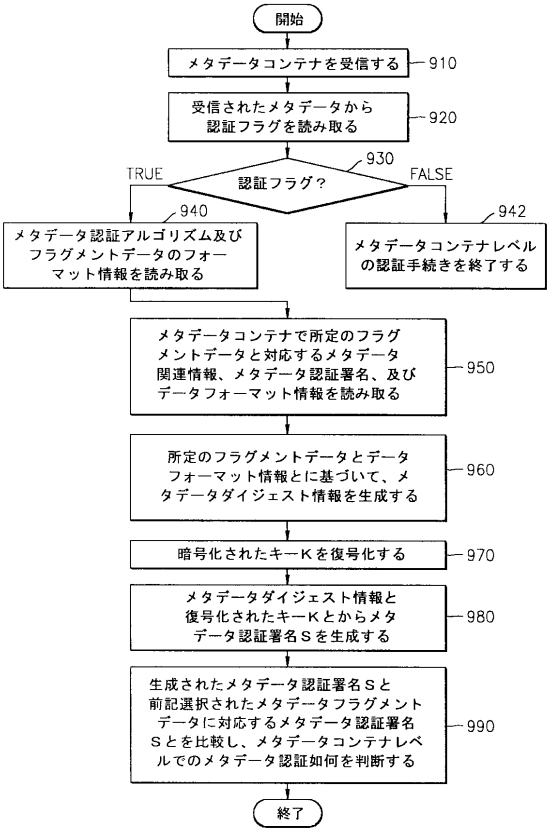
【図７】


開始
メタデータコンテナを受信する
710
受信されたメタデータから認証フラグを読み取る
720
730
TRUE
認証フラグ？
FALSE
740
メタデータ認証アルゴリズムフォーマットデータ情報を読み取る
742
メタデータコンテナレベルの認証手続きを終了する
メタデータコンテナで所定のフラグメントデータと対応するメタデータ関連情報を読み取る
750
所定のフラグメントデータとデータフォーマット情報とに基づいたメタデータダイジェスト情報を生成する
760
生成されたメタデータダイジェスト情報と所定のフラグメントデータに対応する前記読み取られたメタデータダイジェスト情報とを比較し、メタデータコンテナレベルでのメタデータ認証如何を判断する
770
終了

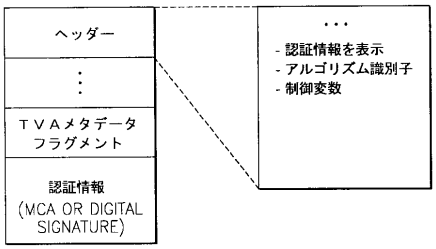
【図８】


開始
伝送されるメタデータを所定のセグメント単位に分割して複数のフラグメントデータを生成する
810
生成されたフラグメントデータのうち、所定のフラグメントデータを選択する
820
選択されたフラグメントデータからのメタデータダイジェスト情報を生成する
830
生成されたメタデータダイジェスト情報と暗号化キーＫを使用したメタデータの認証署名を生成する
840
選択されたフラグメントデータと対応するメタデータダイジェスト情報、メタデータ認証署名、及び選択されたフラグメントデータのフォーマット情報を含むメタデータコンテナを生成及び伝送する
850
終了

【図９】


開始
メタデータコンテナを受信する
910
受信されたメタデータから認証フラグを読み取る
920
930
TRUE
認証フラグ？
FALSE
940
メタデータ認証アルゴリズム及びフラグメントデータのフォーマット情報を読み取る
942
メタデータコンテナレベルの認証手続きを終了する
メタデータコンテナで所定のフラグメントデータと対応するメタデータ関連情報、メタデータ認証署名、及びデータフォーマット情報を読み取る
950
所定のフラグメントデータとデータフォーマット情報とに基づいて、メタデータダイジェスト情報を生成する
960
暗号化されたキーＫを復号化する
970
メタデータダイジェスト情報と復号化されたキーＫとからメタデータ認証署名Ｓを生成する
980
生成されたメタデータ認証署名Ｓと前記選択されたメタデータフラグメントデータに対応するメタデータ認証署名Ｓとを比較し、メタデータコンテナレベルでのメタデータ認証如何を判断する
990
終了

【図１０】


ヘッダー
TVAメタデータフラグメント
認証情報
(MCA OR DIGITAL SIGNATURE)
- 認証情報を表示
- アルゴリズム識別子
- 制御変数

【図１１】
FIG. 11


<?xml version="1.0" encoding="utf-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP:Header>
<wssec:credentials xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SigningCertificate">
<ds:X509Data>
<ds:X509Certificate>MIIH1zCCBr+gAwIBA...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</wssec:credentials>
<wssec:integrity xmlns:wssec="http://schemas.xmlsoap.org/ws/2001/10/security">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/xml-exc-c14n#"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI=""/>
<ds:Transforms>
<ds:Transform Algorithm="http://schemas.xmlsoap.org/2001/10/security
#RoutingSignatureTransform"/>
<ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>j8iwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>aiYECAxNqK2PiwQsRweWaJXup5zJa...</ds:SignatureValue>
<ds:KeyInfo>
<wssec:licenseLocation>
#SigningCertificate
</wssec:licenseLocation>
</ds:KeyInfo>
</ds:Signature>
</wssec:integrity>
</SOAP:Header>
<SOAP:Body>
<TVAmetadataFragment>
<ProgramInformation>.....</ProgramInformation>
<enc:EncryptedData>
<enc:EncryptionMethod Algorithm="xxx_algorithm"/>
<ds:KeyInfo>.....</ds:KeyInfo>
<enc:CipherData>
<enc:CipherValue>9asyBTw2+HcSHtHg...</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedData>
<enc:EncryptedKey>
<enc:EncryptionMethod Algorithm="yyy_algorithm"/>
<ds:KeyInfo>
<ds:KeyName>Public/Private Key for TVA metadata</ds:KeyName>
</ds:KeyInfo>
<enc:CipherData>
<enc:CipherValue>CCBPowCwYDVR0PBA...</enc:CipherValue>
</enc:CipherData>
</enc:EncryptedKey>
</TVAmetadataFragment>
</SOAP:Body>
</SOAP:Envelope>

フロントページの続き

(51)Int.Cl. F I
H04N 7/173 (2006.01) H04N 7/173 610Z

(31)優先権主張番号 10-2003-0013002
(32)優先日 平成15年3月3日(2003.3.3)
(33)優先権主張国 韓国(KR)

(72)発明者 チェー, ヤン－リム
大韓民国 ４６３－０６０ キョンギード ソンナム－シ ブンダン－グ イメードン １２４ ハンシン・アパート ２１０－１５０９

審査官 齋藤 浩兵

(56)参考文献 特開２００１－２７４７８８（ＪＰ，Ａ）
特開２０００－２２４２５７（ＪＰ，Ａ）
特開平１１－３０６０６８（ＪＰ，Ａ）
国際公開第０１／０５２１７８（WO，Ａ１）
特開２００１－２４３１１９（ＪＰ，Ａ）
特開平１１－２２５１６８（ＪＰ，Ａ）
重吉宏樹 他, 放送関連情報提供に向けた番組メタデータ構成手法の提案, 電子情報通信学会２００１年総合大会講演論文集 基礎・境界, ２００１年 3月 7日, p.332, A-16-9

(58)調査した分野(Int.Cl., ＤＢ名)
H04L 12/22
G06F 13/00
G09C 1/00
H04L 9/32
H04L 12/56
H04N 7/173