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

特許第5015148号
(P5015148)

(45)発行日　平成24年8月29日(2012.8.29)　(24)登録日　平成24年6月15日(2012.6.15)

(51)Int.Cl.		F I		
G06F 21/00	(2006.01)	G06F	21/00	151
G06F 21/22	(2006.01)	G06F	21/22	112B
H04L 9/32	(2006.01)	H04L	9/00	675B

請求項の数 9　(全 13 頁)

(21)出願番号	特願2008-519391 (P2008-519391)
(86)(22)出願日	平成18年6月22日 (2006.6.22)
(65)公表番号	特表2009-500726 (P2009-500726A)
(43)公表日	平成21年1月8日 (2009.1.8)
(86)国際出願番号	PCT/US2006/024034
(87)国際公開番号	WO2007/005281
(87)国際公開日	平成19年1月11日 (2007.1.11)
審査請求日	平成21年6月18日 (2009.6.18)
(31)優先権主張番号	60/695,944
(32)優先日	平成17年7月1日 (2005.7.1)
(33)優先権主張国	米国(US)
(31)優先権主張番号	11/354,800
(32)優先日	平成18年2月15日 (2006.2.15)
(33)優先権主張国	米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100140109
弁理士　小野　新次郎
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行
(74)代理人　100120112
弁理士　中西　基晴

最終頁に続く

(54)【発明の名称】インタラクティブ・メディア環境におけるアプリケーション・セキュリティ

(57)【特許請求の範囲】

【請求項１】

インタラクティブ・マルチメディア環境においてアプリケーションのセキュリティを保証する、コンピュータにより実行される方法であって、

ａ．前記コンピュータがアプリケーションを受けるステップと、

ｂ．前記コンピュータが、前記アプリケーションが関連するデジタル署名を有するかどうか検出するステップであって、該検出するステップが、前記アプリケーションに関連する目録ファイルを読み取り、前記目録に著作者の署名および証明書が署名されているか否か判定を行うステップを含む、ステップと、

ｃ．前記アプリケーションに関連する有効なデジタル署名が検出されたとき、前記コンピュータが、前記アプリケーションがローカル・ストレージのソースおよびネットワーク・リソースへアクセスするための許可を与える、ステップと、

ｄ．前記アプリケーションに関連する有効なデジタル署名が検出されなかったとき、前記コンピュータが、前記アプリケーションがローカル・ストレージのソースおよびネットワーク・リソースにアクセスするための許可を拒否するステップと、

ｅ．前記コンピュータが、他のアプリケーションを受けるステップと、

ｆ．前記コンピュータが、前記他のアプリケーションが関連するデジタル署名を有するかどうか検出するステップと、

ｇ．前記アプリケーションに関連する有効なデジタル署名あるいは前記他のアプリケーションに関連する有効なデジタル署名のいずれかが検出されなかったとき、前記コンピュ

ータが、双方のアプリケーションがローカル・ストレージのソースおよびネットワーク・リソース双方へアクセスするための許可を拒否するステップと、
ｈ．前記アプリケーションに関連する有効なデジタル署名および前記他のアプリケーションに関連する有効なデジタル署名の双方が検出されたとき、前記コンピュータが、双方のアプリケーションがローカル・ストレージのソースおよびネットワーク・リソース双方へアクセスするための許可を与えるステップと、
を備えた、方法。
【請求項２】
請求項１記載の方法において、前記アプリケーションは、前記目録ファイルと少なくとも１つのリソース・ファイルとを貯蔵するアーカイブを含み、前記目録ファイルは、前記アーカイブにおける最初のファイルである、方法。
【請求項３】
請求項１記載の方法であって、更に、前記アプリケーションに関連した有効なデジタル署名が検出されたとき、前記アプリケーションは、署名済みの起源証明書、コンテンツ廃止リスト、または著作者識別子を含むこと、を備えた方法。
【請求項４】
請求項１記載の方法において、前記ローカル・ストレージのソースは、著作者識別子とキーで結ばれたディレクトリである、方法。
【請求項５】
請求項１記載の方法であって、更に、前記アプリケーションに関連した有効なデジタル署名が検出されなかったときで、光ディスクから前記アプリケーションを受けた場合、前記コンピュータが、前記アプリケーションを前記光ディスクのみから走らせるステップを備えた、方法。
【請求項６】
アプリケーションのためのマルチメディア再生システムであって、
ネットワーク・リソースと、
ローカル・ストレージのソースと、
第１のアプリケーションと第２のアプリケーションとを受けるデバイスと、
前記第１のアプリケーションが関連するデジタル署名を有するかどうか、かつ前記第２のアプリケーションが関連するデジタル署名を有するかどうかを検出するプロセッサであって、前記検出することは、前記アプリケーションに関連する目録ファイルを読み取り、前記目録に著作者の署名および証明書が署名されているか否か判定を行うことを含む、プロセッサと、
を備えており、
前記第１のアプリケーションに関連する有効なデジタル署名および前記第２のアプリケーションに関連する有効なデジタル署名が検出されたとき、前記プロセッサが、前記第１および第２のアプリケーションの双方が前記ローカル・ストレージのソースおよび前記ネットワーク・リソースへアクセスするための許可を与え、
前記第１のアプリケーションに関連する有効なデジタル署名または前記第２のアプリケーションに関連する有効なデジタル署名が検出されなかったとき、前記プロセッサが、前記第１および第２のアプリケーションが前記ローカル・ストレージのソースおよび前記ネットワーク・リソースへアクセスするための許可を拒否する、マルチメディア再生システム。
【請求項７】
請求項６記載のシステムにおいて、前記第１および第２のアプリケーションは各々、前記目録ファイルおよび少なくとも１つのリソース・ファイルを貯蔵するアーカイブを収容し、前記目録ファイルは、前記アーカイブにおける最初のファイルである、システム。
【請求項８】
請求項６記載のシステムにおいて、前記署名ステータスが署名済みである場合、前記第１および第２のアプリケーションは各々、署名済みの起源証明書、コンテンツ廃止リスト

、または著作者の識別子を含む、システム。
【請求項９】
請求項８記載のシステムにおいて、前記ローカル・ストレージのソースは、著作者識別子とキーで結ばれたディレクトリである、システム。
【発明の詳細な説明】
【技術分野】
【０００１】
（関連出願の申告）
本願は、２００５年７月１日に出願した仮特許出願第６０／６９５，９４４号の優先権を主張する。その内容は、ここで言及したことにより、本願にも含まれるものとする。
【背景技術】
【０００２】
マルチメディア再生システムの中には、オーディオ／ビデオ再生中にインタラクティブ・グラフィックを制限して提供する場合がある。インタラクティブ再生システムの能力が高い程、違法行為につけ込まれる状況も多くなる。ウイルス、スパイウェア、およびその他の悪意のソフトウェアに対抗して再生システムのセキュリティを維持することは肝要である。悪意のソフトウェアのために、インタラクティブ再生システムが誤動作を起こしたり、秘密のユーザ情報を集めて送信してしまう虞れがある。加えて、インタラクティブ再生システムをネットワークに接続することもある。ソフトウェアまたはユーザ情報は、再生システムから、ネットワークに取り付けられている他の精算システムに伝搬する可能性がある。その結果、インタラクティブ再生システムがしかるべきセキュリティ装備を含むことは、欠くことができなくなっている。
【発明の開示】
【発明が解決しようとする課題】
【０００３】
インタラクティブ・マルチメディアの分野において未署名アプリケーション(unsigned application)の特権を制御するセキュリティ・システムを提供する。
【課題を解決するための手段】
【０００４】
インタラクティブ・マルチメディアとは、アプリケーションが、通例、ユーザ入力イベントに応答して、同期を取りリアル・タイムで精度高くフレーム毎に、グラフィックス、オーディオ、およびビデオを含むマルチメディア・オブジェクトを管理する環境のことである。アプリケーションは高品位ＤＶＤ（ディジタル・バーサタイル・ディスク）メディアに関係するので、ここでは「ｉＨＤ」アプリケーションと呼ぶ。しかしながら、開示するセキュリティ・システムは、更に一般的に、その他のインタラクティブ・マルチメディア環境にも適用可能である。
【０００５】
具体的には、本システムは、コンテンツ・セキュリティではなく、アプリケーション・セキュリティに適用され、署名システムを含む、アプリケーション・セキュリティに合わせてフレームワークを確立し、更にセキュリティをサポートするファイル・フォーマットを規定する。インタラクティブ・マルチメディア・アプリケーションは、インタラクティブ再生システム（単体のハードウェア・デバイスとして実施され、あるいは、例えば、パーソナル・コンピュータ上で走るソフトウェア・アプリケーションとして実施される）上で走り、署名済みでも未署名でもよい。
【０００６】
アプリケーションに署名されていれば、実際上無制限の適用が許可される。未署名のアプリケーションでは、これがアクセスできる範囲が大幅に制限される。更に、署名済みのアプリケーションおよび未署名のアプリケーション双方が走っている場合、未署名のアプリケーションのセキュリティ・レベルおよびアクセス特権のみが双方に与えられる。未署名のアプリケーションに備えることによって、豊富な双方向参加機構を用いて家庭で著作

したディスクをカスタム化することを可能にするが、ネットワーク、例えば、インターネットや、再生システム内に格納されている機密情報へのアクセスを、許可を付与された者に制限する。

【０００７】

署名済みのアプリケーションには、特殊なファイル・フォーマットを規定し、ファイル全体を解析する必要なく、署名ステータスの判定を可能にする。

【発明を実施するための最良の形態】

【０００８】

インタラクティブ・マルチメディア・アプリケーションとは、アプリケーションがユーザ・イベントに応答するアプリケーションである。一例を上げると、ユーザがアクセスするアプリケーション内部に実装されたメニューがあり、ユーザが入力を提出すると、アプリケーションが状態を変化させる。このような場合、双方向参加は、メニュー・グラフィクスによって行われ、メニュー・グラフィクスがレンダリングされている間、その下で、例えば、ｚ＝０レイヤ上で、リアル・タイムでフレーム同期に基づいて、ビデオを再生する。双方向参加により、例えば、ビデオ・ストリームの表示方法の変化をもたらすことができる。

【０００９】

例えば、下地のビデオは高品位ムービーとすることができる。グラフィック・オーバーレイは、ムービーのディレクタによる批評の一部とすることができ、例えば、場面自体の上に被せた種々のカメラ位置の模式図を示す。ユーザは、リモコンを用いて、これらのカメラ位置のいずれによって撮影された視界(view)にも切り換えることができる。

【００１０】

先に記したように、インタラクティブ再生システムの能力が高い程、違法行為につけ込まれる状況も増える。悪意のソフトウェアのために、インタラクティブ再生システムが誤動作を起こしたり、秘密のユーザ情報を集めて送信してしまう虞れがある。

【００１１】

本システムでは、再生システムにおいて用いるインタラクティブ・アプリケーションには、署名しても署名しなくてもよい。署名済みのアプリケーションとは、信頼できる起源の権威(trusted root authority)から起源証明書(root certificate)を継承し、安全と見なされるアプリケーションである。

【００１２】

署名済みのアプリケーションには、高レベルのアクセス特権が与えられる。この殆ど無制限の特権により、例えば、ネットワーキング、ファイルＩ／Ｏ、セキュリティ、および診断ＡＰＩへのアクセスが許され、永続的ストレージにアクセスして、アプリケーションの呼び出し(invocation)を何回行っても変化しないデータを格納および検索することができる。

【００１３】

一方、未署名のアプリケーションには、低レベルのアクセス特権が与えられる。これらは、高アクセス(high access)によって得られる種類の機能へのアクセスを拒絶される。これらは、マークアップ言語、ならびに、例えば、以下のECMAScriptにおけるＡＰＩ例からのある種のオブジェクト、ＸＭＬ（Ｉ／Ｏ機能なし）、グローバル化、グラフィック・エレメントに関連する描画機能、およびユーザ入力動作の利用に制限される場合がある。

【００１４】

このレベルの機能は、いずれのネットワーキング、セキュリティ、またはファイルＩ／Ｏへのアクセスをも禁止する。永続的ローカル・ストレージから前述の名称空間(namespace)またはロード・リソース(load resource)の外側の関数をコールしようとしても、いずれもが、例外となり、アプリケーションが終了する。

【００１５】

一実施形態では、メディア・ディスク、例えば、ＨＤ－ＤＶＤ上にアプリケーション集合があり、これらを用いてインタラクティブ・グラフィクスおよびビデオ・アプリケーシ

ョンを走らせる。図１を参照すると、再生システムがメディア・ディスクを受ける（ステップ１２）。再生システムは、汎用のコンピュータ・システムまたは更に特殊化したメディア・センタ・システムでもよく、メディア上のアプリケーションの署名ステータスを判定する（ステップ１４）。全てのアプリケーションの署名ステータスが署名済みであると判定されると（ステップ１６）、アプリケーションの全てに高アクセス特権が与えられる（ステップ１８）。いずれか１つのアプリケーションでも署名ステータスが未署名であると判定された場合、全てのアプリケーションには低アクセス特権が与えられる（ステップ２２）。即ち、未署名のアプリケーションが走っている場合、同時に走っている全てのアプリケーションが署名済みであれ未署名であれ、これらを未署名アプリケーション許可レベルに制限することができる。これによって、未署名のアプリケーションが、同時の署名済みアプリケーションの特権を利用することを防止する。

【００１６】

別の実施形態では、同様の方法を、再生システムにロードされたアプリケーションに直接適用することができる。図２を参照すると、アプリケーションを再生システムにロードすることができる（ステップ２４）。次いで、アプリケーションの署名ステータスを検出する（ステップ２６）。署名ステータスが署名済みであると判定されると（ステップ２８）、アプリケーションを高特権アクセス・レベルで走らせることができる（ステップ３２）。しかしながら、署名ステータスが未署名であると判定されると（ステップ２８）、低特権アクセス・レベルでアプリケーションを走らせる（ステップ３４）。この場合、アプリケーションをメディア、例えば、ディスクから直接走らせる（ステップ３６）。これによって、セキュリティを強化する。何故なら、未署名のアプリケーションは全て、走ることも、再生システムのローカル永続的ストレージからリソースをロードすることも妨げられるからである。追加のアプリケーションをロードする際には（ステップ３８）、これらの署名ステータスを検査してもしなくてもよい。一般に、これらには低アクセス・レベルが与えられる（ステップ３４）。アプリケーションが署名され、高アクセス特権が与えられ、その後にロードされたアプリケーションが未署名である場合、高アクセス・アプリケーションは低アクセス・レベルに格下げとなる。

【００１７】

図３を参照すると、署名済みアプリケーションに対して、再生システムは、メディアを再生システムに導入したとき（ステップ４２）に、検出した著作者識別子集合の採用を含むことができる（ステップ４４）。即ち、各メディアまたはアプリケーションには著作者識別子を関連させることができる。著作者識別子は、コンテンツ著作者を一意に識別し、永続的ストレージにおけるアプリケーション、即ち、呼び出しの度に変化しないことが望ましいデータを格納および検索するために永続的ストレージにアクセスすることができるアプリケーションのセキュリティには特に重要である。

【００１８】

次いで、著作者識別子を、当該著作者識別子と関連のあるディレクトリの作成と関連付ける（ステップ４６）。そのメディアからのアプリケーションは、永続的ストレージにおいてその著作者識別子に対応するディレクトリにしかアクセスすることができない。ファイル・システムは、アプリケーションがそれを見ると、そのディレクトリを起源とする。アプリケーションはサブディレクトリを管理することができるが、そのルート・ディレクトリを超えることや、他の著作者のデータを見ることはできない。

【００１９】

著作者識別子は、ディスク上の全てのアプリケーションを含むディスク、あるいはそのディスク上にある、数枚のディスクに跨って分散されている、またはそれ以外では例えばインターネット・ダウンロードによって再生システムにロードした、１つのアプリケーションのいずれにでも関連付けることができる。更に、所与のメディアに１つの著作者識別子を関連付けることができるが、所与の著作者識別子が多数のメディア上で発見されることもある。他の実施形態では、アプリケーションに署名したキーによって示される識別子を用いることができる。異なるアプリケーションが異なる人によって１つのメディア上で

署名することができると仮定すると、この実施形態では、ストレージを更に大きく分離することになる。最後の署名の代わりに証明書の連鎖を用いると、更に一層そのようになる。

アプリケーションの構造

これより、署名済みアプリケーションの構造について説明する。図４を参照すると、署名済みアプリケーション５０は、目録ファイル５２と少なくとも１つのリソース・ファイル５４とを含むことができる。目録ファイル５２には著作者の署名および証明書が署名されており、それが参照するリソース全てを認証する。

【００２０】

アプリケーションは、その目録ファイル５２および全てのリソース・ファイル５４～５８を纏めて１つの未圧縮アーカイブ４８にすることもできる。アーカイブ４８のファイル・フォーマットは、暗号化をサポートする必要もない。アーカイブ４８は、本質的に、コンテナであり、一般に独立して署名する必要はない。目録ファイル５２は、インタラクティブ・アプリケーションのリソース・ファイル５４～５８の各々を参照することができる。アーカイブ４８のアーキテクチャは、アーカイブ４８を効率的に流動させるように指定することができる。例えば、署名済みの目録ファイル５２をアーカイブ４８における最初のファイルとすれば、アーカイブ全体を読み取ることなく、署名の検証が可能となる。アーカイブ・フォーマットの後続バージョンは、以前のバージョンと下位互換性を有するとよい。

【００２１】

アーカイブ４８におけるデータの認証は、例えば、ＲＦＣ３２７５が定義するＸＬＭ－署名の使用によって行うことができる。

目録ファイルのフォーマット

一例では、署名済み目録ファイル５２のフォーマットは、ＲＦＣ３２７５によって定義されたＸＭＬ－署名シンタックスおよび処理に対するＷ３Ｃ推奨の部分集合を利用することができる。このように、以下の要素の部分集合を含めてサポートすることができる。

【００２２】

ds:Signature
ds:SignatureValue
ds:SignatureType
ds:Reference
ds:Reference/ds:DigestValue

他の要素は、システムによって決定することができる。目録に含まれるリソース項目毎のダイジェスト値(digest value)はds:Reference 要素としてリストに纏めることができる。

証明書および署名

一例として、要求される署名タイプが、例えば、Ｘ．５０９とすることができる。ds:SignatureMethodによって定義される署名方法は、ＲＳＡ－ＳＨＡ１とすることができる。カノニカル化方法(canonicalization method)は、Exclusive XML canonicalization 1.0とするように指定することができる。ダイジェスト方法は、署名方法、ＲＳＡ－ＳＨＡ１と同一とすることができる。キー情報は、システムによって、アプリケーションを走らせているメディアまたはローカル・ストレージ・エリアの個体情報(identity)から推論することができる。

証明書廃止リスト

悪化したアプリケーションの廃止および交換のためのメカニズムを設けるために、各インタラクティブ・ビデオおよびグラフィクス・アプリケーションの著作者は、廃止したアプリケーションのダイジェスト値を纏めてリストとしたコンテンツ廃止リスト（「ＣＲＬ」）を含むことができる。このＣＲＬは、別個のファイルに含めることができる。このファイルは、廃止した署名ダイジェストを纏めたリスト、およびディスクを著作したコンテンツ作成者の署名を収容する。廃止した各アプリケーションの元の著作者の署名が、ＣＲ

Ｌファイルにおける署名のそれと一致したと仮定すると、リストにあるアプリケーション・ダイジェストは、ローカル・ストレージのコンテンツ提供業者の制限された区域に格納されており、もはや走ることは許されない。ＣＲＬがアプリケーションに含まれている場合、Revocation.xmlのような認識可能な名称を与えることができる。

【００２３】

アプリケーション著作者は、廃止したアプリケーションを新バージョンと交換することを望むこともあり得る。これは、数種類の異なる方法で遂行することができる。インターネット接続プレーヤ上で走っているタイトル(title)に、これらのホーム・サーバをチェックさせて、新たにダウンロードするアプリケーションを指定する更新プレーリストまたはインタラクティブ・ビデオおよびグラフィクス群を求めることができる。あるいは、アプリケーションを廃止するメディアは、それ自体で代わりを供給することもできる。

【００２４】

以下の表は、アーカイブ・ファイルに可能な１つのフォーマットを、フィールドについて記述するコメントと共に記載する。尚、多数のその他のフォーマットも使用可能であることを記しておく。この表では、タイプを表すために略語を用いている。Uinは、ｎビットの符号無し整数を表す。例えば、Ui8は、８ビットの符号無し整数であり、Ui32は、３２ビットの符号内整数である。角括弧を用いることによってタイプのアレイを示し、アレイの長さをこれらの括弧の間に示す。長さが直前のフィールドに左右される場合、そのフィールドの名称、またはそのフィールド内に示されるそれよりも短い名称を用いて、そのフィールドの値に言及することができる。１６進数は、0xdd表記を用いて、示す。全ての可変長ストリング、したがって、リソース名は、UTF-8を用い、パスカル・ストリング表記（８ビット長とそれに続くバイト）を用いてエンコードすることができる。

【００２５】

【表１】

	フィールド		タイプ	コメント
アーカイブ・ヘッダ	マジック		Ui8[5]	５バイトの値は 0x69、0x48、0x44、0x61、0x72 でなければならない。
	バージョン		Ui8	フォーマットのバージョン。値は 0x01 でなければならない。
リソース・カタログ	リソース・エントリ#１	リソース・エントリ長	Ui16	このリソース・エントリの長さ
		リソース・オフセット	Ui32	リソース・データ・ブロックにおけるリソースのバイト・オフセット
		リソース長	Ui32	バイト単位でのリソースの長さ
		リソース・チェックサム	Ui32	リソース・バイトのCRC－32チェックサム
		リソース・タイプ		
		リソース名の長さ	Ui8	RNL
		リソース名	Ui8[RNL]	リソースのファイル・システム名

【００２６】

【表２】

	フィールド		タイプ	コメント
	リソース・エントリ#n	リソース・エントリ長	Ui16	このリソース・エントリの長さ
		リソース・オフセット	Ui32	リソース・データ・ブロックにおけるリソースのバイト・オフセット
		リソース長	Ui32	バイト単位でのリソースの長さ
		リソース・チェックサム	Ui32	リソース・バイトのＣＲＣ－３２チェックサム
		リソース・タイプ		
		リソース名の長さ	Ui8	RNL
		リソース名	Ui8[RNL]	リソースのファイル・システム名
		連続ブロックにおける全てのリソース・データ		
リソース・データ・ブロック				

以上に記したアプリケーション・リソースにある種の規則を適用してもよい。例えば、リソース名は、ファイル・システムの名称または論理ＵＲＩとしなければならない。アーカイブ・ファイルを抽出するディレクトリは、その抽出中におけるファイル・システムのルートとであると見なすことができる。このように、全ての名称をそのディレクトリに関して作り、絶対パスが関係あるものと同じように振る舞うようにする。ある名称からそのディレクトリの外部の位置が得られた場合、名称およびエントリを無効と見なすことができる。

【００２７】

以下の章では、前述のアーカイブ・ファイル例の種々のフィールドおよびセクションに関する更に詳細な情報を示す。

【００２８】

アーカイブ・ヘッダ

マジック・フィールド

これは、アーカイブを一意に識別するために用いられる「マジック数」である。これは、ストリング"iHDar"、即ち、UTF-8、ASCII等の５文字値のシーケンスとして符号化したiHDアーカイブ、即ち、0x69、0x48、0x44、0x61、0x72から成るとすることができる。

【００２９】

バージョン・フィールド

バージョン・フィールドは、アーカイブリーダが異なるバージョンのアーカイブ・フォーマットを読みとれるようにする。バージョン・フィールドを見ることにより、ファイルの異なるセクションにおいて何が予期されるかを知ることができ、したがってあるバージョンのファイル・フォーマットにはなかった情報を読み出すことができる。このフィールドの値は、例えば、0x01とすることができる。今後のバージョンは、0x02から0xffまでの値を有することができる。

【００３０】

リソース・カタログおよびリソース・エントリ

リソース・カタログは、多数のリソース・エントリを含む角エントリは、同じフォーマットに従う。

【００３１】

リソース・エントリ長

これは、リソース・エントリ自体の長さであり、バイト単位で表す。この値は、バージ

ョンがわからないフォーマットを読んでいるリーダによって用いられる。バージョン２のフォーマットを用いて書かれたアーカイブが、バージョン１に合わせて作られたリーダによって見られていると仮定すると、リーダは、それに分かるフィールドを読み、次いで、現エントリの長さが分かるので、次のリソース・エントリまで飛ばすことができる。

【００３２】

リソース・オフセット

これは、リソース・データ・ブロックにおけるリソースのバイト・オフセットを示す。最初のリソースのオフセットは0x0000である。

【００３３】

リソース長

これは、リソースの長さであり、バイト単位である。

【００３４】

リソースＡおよびＢがアーカイブ・ファイルにおいて連続している場合、Ｂに対するリソース・オフセットは、Ａのリソース・オフセットとリソース長との和に等しい。

【００３５】

リソース・チェックサム

これは、ISO3309が定義する、リソースのＣＲＣ－３２チェックサムを表す。尚、このチェックサムは、信頼性のない媒体を通じて輸送されたリソースの保全性の単純な検証に限って用いてもよいことを記しておく。ＣＲＣ－３２チェックサムは、キーが付けられておらず、耐変転(collision-proof)でもないので、認証の目的には用いてはならない。リソースを認証する必要がある場合、前述の署名メカニズムを用いるとよい。

【００３６】

リソース・タイプ

これは、リソースのMIMEタイプである。

【００３７】

リソース名の長さ

これは、リソース名の長さであり、バイト単位である。リソース名はこのフィールドの直後にある。

【００３８】

リソース名

これは、リソース名自体である。

【００３９】

リソース・データ・ブロック

リソース・データ・ブロックは、リソースに対する全てのバイトを、リソース・カタログに出てくる順に収容する。一般に、２つのリソース間には明示的な分離はない。これは、そのオフセットおよび長さが周知の量であるためである。

【００４０】

本システムの説明は、全体的に、コンピュータが実行するプログラム・モジュールのような、コンピュータ実行可能命令に関して行う。一般に、プログラム・モジュールは、ルーチン、オブジェクト、コンポーネント、データ構造等を含み、特定のタスクを実行するか、または特定の抽象的データ・タイプを実現する。本システムおよび方法は、分散型計算環境においても採用することができ、その場合、通信ネットワークを通じてリンクされているリモート処理デバイスによってタスクを実行する。分散型計算環境では、プログラム・モジュールは、メモリ記憶デバイスを含む、ローカルおよびリモート・コンピュータ記憶デバイス双方に位置することもできる。

【００４１】

本方法およびシステムを実行する命令は、種々のコンピュータ読み取り可能媒体上に格納することができる。コンピュータ読み取り可能媒体は、コンピュータがアクセス可能な入手可能な媒体であればいずれでも可能であり、揮発性および不揮発性の双方、リムーバブル、および非リムーバブル媒体を含む。限定ではない一例をあげると、コンピュータ読

み取り可能媒体は、コンピュータ記憶媒体および通信媒体から成ると考えられる。コンピュータ記憶媒体は、コンピュータ読み取り可能命令、データ構造、プログラム・モジュール、またはその他のデータというような情報の格納のために、あらゆる方法または技術で実施される、揮発性および不揮発性の双方、リムーバブル、および非リムーバブル媒体を含む。コンピュータ記憶媒体は、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリまたはその他のメモリ技術、ＣＤ－ＲＯＭ、ディジタル・ビデオ・ディスク（ＤＶＤ）またはその他の光ディスク・ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージまたはその他の磁気記憶デバイス、あるいは所望の情報を格納するために用いることができしかもコンピュータがアクセス可能なその他のいずれの媒体も含むが、これらに限定されるのではない。通信媒体は、通例、コンピュータ読み取り可能命令、データ構造、プログラム・モジュール、またはその他のデータを、搬送波またはその他の移送機構のような変調データ信号において具体化し、あらゆる情報配信媒体を含む。「変調データ信号」という用語は、その特性集合の１つ以上が、情報を信号内にエンコードするようなやり方で、変化している信号を意味する。限定ではなく、一例として、通信媒体は、有線ネットワークまたは直接有線接続というような有線媒体、ならびに音響、ＲＦ、赤外線、およびその他のワイヤレス媒体というようなワイヤレス媒体を含む。前述のいずれの組み合わせも、コンピュータ読み取り可能媒体の範囲に当然含まれるものとする。

【００４２】

コンピュータを含む計算システム環境の一例に関して説明したが、本システムは、本発明は、その他の多数の汎用または特殊目的計算システム環境または構成でも動作可能である。計算システム環境は、本発明の使用や機能の範囲に関していずれの限定をも示唆することを意図していない。更に、計算システム環境は、動作環境例に示す構成要素のいずれの１つまたは組み合わせに関しても、いかなる依存性や必須要件をも有するという解釈は行わないこととする。本発明と共に用いるのに適していると思われる周知の計算システム、環境、および／または構成の例には、パーソナル・コンピュータ、サーバ・コンピュータ、ハンド・ヘルドまたはラップトップ・デバイス、マルチプロセッサ・システム、マイクロプロセッサに基づくシステム、セット・トップ・ボックス、プログラマブル消費者用電子機器、移動体電話、ネットワークＰＣ、ミニコンピュータ、メインフレーム・コンピュータ、前述のシステムまたはデバイスのいずれをも含む分散型計算機環境等が含まれる。

【００４３】

ここに記載したシステムおよび方法は、ソフトウェアまたはハードウェア、あるいは一部は当技術分野では周知である技法を用いて双方で実施することができる。

【００４４】

ここに図示し説明した方法の実行順序は、特に指定がない限り、必須ではない。即ち、方法の要素は、特に指定がない限り、いずれの順序で実行してもよく、本方法は、ここに開示した要素よりも多いまたは少ない要素を含んでもよい。

【００４５】

本発明またはその実施形態の要素を導入する際において、冠詞「a」、「an」、「the」、「said」は、１つ以上の要素があることを意味することを意図している。「comprising」、「including」、および「having」は、内包的であることを意図しており、掲示する要素以外にも追加の要素があり得ることを意味する。

【００４６】

前述の構造、生産物、および方法には、本発明の範囲から逸脱することなく種々の変更を行うことができるので、以上の説明に内包される事項全ては、限定の意味ではなく、例示と解釈することを意図している。

【００４７】

以上構造的特徴および／または方法論的ステップに対して特定の言葉で主題について説明したが、添付した特許請求に定めた主題は、必ずしも前述の特定の特徴や行為(act)に限定されるのではないことは言うまでもない。逆に、前述の特定の特徴や行為は、特許請

求の範囲を実施する形態例として開示したものである。

【図面の簡単な説明】

【００４８】

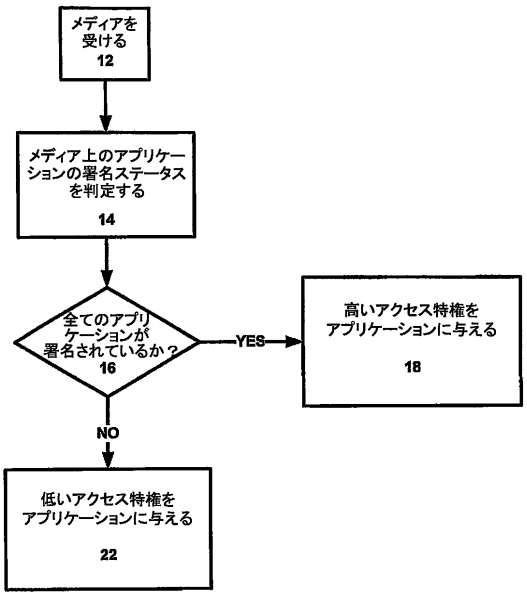
【図１】図１は、アプリケーションの署名ステータスをディスクから検出する、特権をアプリケーションに割り当てる方法を示すフローチャートである。

【図２】図２は、再生システムに装填するときにアプリケーションの署名ステータスを検出する、特権をアプリケーションに割り当てる方法を示すフローチャートである。

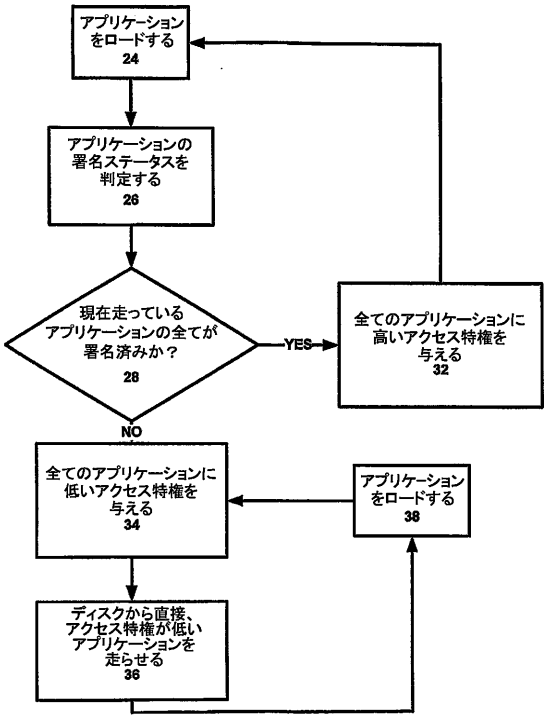
【図３】図３は、著作者識別子－キー・ディレクトリ(keyed directory)の作成を示すフローチャートである。

【図４】図４は、アプリケーション・ファイルの模式図である。

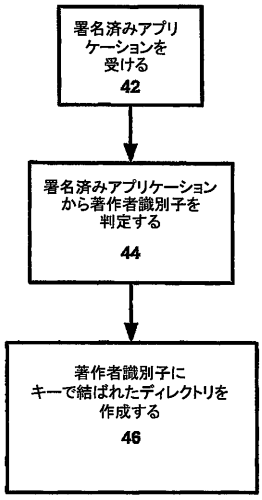
【図１】



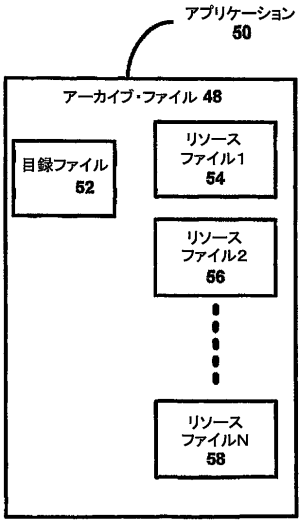
【図２】



【図３】


署名済みアプリケーションを受ける
42
署名済みアプリケーションから著作者識別子を判定する
44
著作者識別子にキーで結ばれたディレクトリを作成する
46

【図４】


アプリケーション
50
アーカイブ・ファイル 48
目録ファイル
52
リソースファイル1
54
リソースファイル2
56
リソースファイルN
58

フロントページの続き

(72)発明者　ヒューズ，ジュニア，ロバート・ケイ
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ
(72)発明者　アロウエ，イブス
アメリカ合衆国ワシントン州９８０５２－６３９９，レッドモンド，ワン・マイクロソフト・ウェイ

審査官　市川　武宜

(56)参考文献　特開２００５－１４９３９４（ＪＰ，Ａ）
特開平１０－２５４７８３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　21/00
G06F　21/22
H04L　 9/32