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

特許第4806396号
(P4806396)

(45)発行日　平成23年11月2日(2011.11.2)　(24)登録日　平成23年8月19日(2011.8.19)

(51)Int.Cl.		F I		
G06F 12/14	(2006.01)	G06F	12/14	510D
G06F 21/24	(2006.01)	G06F	12/14	560C
G06F 21/22	(2006.01)	G06F	12/14	530B
G09C 1/00	(2006.01)	G06F	9/06	660G
H04L 9/32	(2006.01)	G09C	1/00	660D

請求項の数 2　(全 30 頁)　最終頁に続く

(21)出願番号　特願2007-502669 (P2007-502669)
(86)(22)出願日　平成18年2月13日 (2006.2.13)
(86)国際出願番号　PCT/JP2006/302448
(87)国際公開番号　WO2006/085647
(87)国際公開日　平成18年8月17日 (2006.8.17)
審査請求日　平成21年2月4日 (2009.2.4)
(31)優先権主張番号　特願2005-36621 (P2005-36621)
(32)優先日　平成17年2月14日 (2005.2.14)
(33)優先権主張国　日本国(JP)

(73)特許権者　000005821
パナソニック株式会社
大阪府門真市大字門真１００６番地
(74)代理人　100090446
弁理士　中島　司朗
(74)代理人　100072442
弁理士　松村　修治
(74)代理人　100125597
弁理士　小林　国人
(72)発明者　ライクセンリング　ジェルマーノ
大阪府門真市大字門真１００６番地　松下電器産業株式会社内
(72)発明者　金丸　智一
大阪府門真市大字門真１００６番地　松下電器産業株式会社内

最終頁に続く

(54)【発明の名称】アプリケーション実行装置、管理方法、プログラム

(57)【特許請求の範囲】
【請求項１】
ディスクルート証明書と、第１のルート証明書とアプリケーションとが記録されたディスク媒体からアプリケーションを読み出して、実行するアプリケーション実行装置であって、
前記ディスクルート証明書は、ルート認証局から配布された第２のルート証明書を当該ディスク媒体に割り当てたものであり、
前記ディスクルート証明書と、前記第１のルート証明書とが同じかどうかを判断することによりアプリケーションの正当性を判定する管理手段と、
前記管理手段が正当であると判定した場合、アプリケーションを実行する実行手段と、
記憶領域であるドメイン領域を複数設けることができるように構成された記憶手段と、
前記管理手段が正当であると判定した場合、前記記憶手段内のドメイン領域のうち、前記ディスクルート証明書から算出されるハッシュ値に対応するドメイン領域を前記アプリケーションに割り当てる割当手段と
を備えることを特徴とするアプリケーション実行装置。
【請求項２】
記憶領域であるドメイン領域を複数設けることができるように構成された記憶手段を備え、ディスクルート証明書と、第１のルート証明書とアプリケーションとが記録されたディスク媒体からアプリケーションを読み出せるように構成したコンピュータを実行させるアプリケーション実行方法であって、

前記ディスクルート証明書は、ルート認証局から配布された第２のルート証明書を当該ディスク媒体に割り当てたものであり、
前記ディスクルート証明書と、前記第１のルート証明書とが同じかどうかを判断することによりアプリケーションの正当性を判定する第１のステップと、
前記第１のステップにおいて、正当であると判定した場合、アプリケーションを実行する第２のステップと、
前記第１のステップにおいて、正当であると判定した場合、前記記憶手段内のドメイン領域のうち、前記ディスクルート証明書から算出されるハッシュ値に対応するドメイン領域を前記アプリケーションに割り当てる第３のステップを
前記コンピュータに実行させるアプリケーション実行方法。

【発明の詳細な説明】
【技術分野】
【０００１】
アプリケーションに対するリソース割当技術の分野に属する発明である。
【背景技術】
【０００２】
リソース割当技術とは、プラットフォームが有している記憶装置（ローカルストレージ）内の領域を、リソースとしてアプリケーションに割り当てる技術に属する発明である。世界中の様々な組織から供給される様々なアプリケーションを、体系的に管理するため、欧州向けデジタル放送受信装置であるMHP(MultimediaHome Platform)は、以下のような統一様式のファイルパスで、アクセスできるようにローカルストレージのディレクトリ構造を規定している。
ファイルパス:Root/組織ID/appID
ここで組織IDとは、アプリケーションの供給元となる組織を一意に示す識別子であり、appIDとは、アプリケーションを一意に示す識別子である。
【０００３】
以上の領域割当を実現するにあたって、MHPとなる機器は、ルート証明書を用いて、アプリケーションの正当性を判定する。以下、機器(MHP)においてなされていた、アプリケーションの正当性チェックと、アプリケーションに対する領域割り当てについて説明を行う。
アプリケーションを作成した作成者は、アプリケーションを機器に引渡すにあたって、ルート証明書をアプリケーションに付加したうえでアプリケーションを機器に送信する。このルート証明書は、機器固有に割り当てられていたルート証明書と同じものであり、機器は、ルート証明書が付されたアプリケーションを受信して、このアプリケーションに付加されたルート証明書と、機器に割り当てられたルート証明書との同一性を判定する。もし同一性があるなら、そのアプリケーションの供給元の組織に対応する組織用ディレクトリを、当該アプリケーションに割り当てて、そのディレクトリ内のファイルのアクセスを、アプリケーションに行わせる。
【０００４】
アプリケーションのアクセス権限を適切にコントロールする技術は、非特許文献１に記載されている。
【非特許文献１】
「JAVA（登録商標）Security」Scott　Oaks著、O'Reilly発行、　　May　2001、ISBN 0-596-00157-6
【発明の開示】
【発明が解決しようとする課題】
【０００５】
ところで上述した従来技術では、アプリケーション供給元の組織に対応する領域を設けて、その組織に対応するディレクトリの配下の領域を、アクセスを要求したアプリケーションに割り当てている。そのためアプリケーション実行装置が世界のあらゆる地域に配布

され、世界中のあらゆる組織からアプリケーションが供給されるような場合は、世界の様々な組織の組織IDが重複しないような手当が必要になる。何故なら、そうしないと、ある組織に属するアプリケーションが、他の組織のためのディレクトリを自由にアクセスできることになり、アプリケーションが利用するデータの機密性が保てないからである。かかる手当が必要になるので、世界中の各組織に、ユニークな組織IDを割り当てる第３者機関が必要になり、アプリケーション実行装置の標準化に携わるメーカーは、かかる機関の設立・運営に、資本や人材を確保せねばならない。これは、アプリケーション実行装置の標準化に携わるメーカーにとって、多大な負担になる。しかし他の組織が、自組織のためのディレクトリを自由にアクセスするとなると、自組織のアプリケーションのためのデータが、他の組織により、盗用、無断使用され可能性が生じる。装置上において、かかる無法が放置されるとするならば、アプリケーションを作成する作成者は、装置へのアプリケーションの供給を躊躇する事態が多発すると考えられる。これでは、装置で動作するアプリケーションの充実化は望めず、アプリケーションの不足から、アプリケーション実行装置普及の道が閉ざされる恐れがある。

【０００６】

またMHPでは、装置に割り当てられたルート証明書を、アプリケーションの正当性判定に用いている。装置に割り当てられたるルート証明書が、悪意をもった者により暴露された場合、機器に割り当てられたデジタル証明書は、装置の製造元により新たなものにアップデートされる。かかるデジタル証明書のアップデートがあった場合、古いルート証明書が付加されていたアプリケーションは、正当性が正しく判断されず、ローカルストレージをアクセスすることが許されない。無論、MHPで利用されるアプリケーションのように、アプリケーションが、放送時にのみ利用されればよい一過性のものなら、絶えず新しいアプリケーションが送信されるため、そのような扱いで充分であるかもしれない。しかしアプリケーションが、DVD-Videoコンテンツ、BD-ROMコンテンツ等、ディスクに記録された映画作品に関する処理を行う場合、古い映画作品が何度も再生された場合でも、当該映画作品に関する処理を行うアプリケーションは、正しい動作を行うことが求められる。そのためデジタル証明書のアップデートにより、アプリケーションが動作しなくなるというのは、決して望ましくない。つまり従来の技術は、ルート証明書が暴露された場合の、動作保障が不徹底であるという問題がある。

【０００７】

本発明の第１の目的は、世界的なレベルで、重複が生じないような組織IDの管理を必要とせずとも、様々な組織から供給されたアプリケーションにて利用されるファイルの機密性を高めることができるアプリケーション実行装置を提供することである。

本発明の第２の目的は、ルート証明書が暴露された場合でも、高いレベルの動作保障を実現しうるアプリケーション実行装置を提供することである。

【課題を解決するための手段】

【０００８】

上記第１、第２の目的を達成するため、本発明にかかるアプリケーション実行装置は、ディスクルート証明書と、第１のルート証明書とアプリケーションとが記録されたディスク媒体からアプリケーションを読み出して、実行するアプリケーション実行装置であって、

前記ディスクルート証明書は、ルート認証局から配布された第２のルート証明書を当該ディスク媒体に割り当てたものであり、

前記ディスクルート証明書と、前記第１のルート証明書とが同じかどうかを判断することによりアプリケーションの正当性を判定する管理手段と、

前記管理手段が正当であると判定した場合、アプリケーションを実行する実行手段と、

記憶領域であるドメイン領域を複数設けることができるように構成された記憶手段と、

前記管理手段が正当であると判定した場合、前記記憶手段内のドメイン領域のうち、前記ディスクルート証明書から算出されるハッシュ値に対応するドメイン領域を前記アプリケーションに割り当てる割当手段と

を備えることを特徴とする。

【発明の効果】

【０００９】

本発明にかかるアプリケーション実行装置は、上述したような構成を備えるので、ローカルストレージの内部には、複数のドメイン領域が存在し、各ドメイン領域がルート証明書ハッシュ値のそれぞれに割り当てられている。そしてこれらのドメイン領域の配下に、組織毎、アプリケーション毎の領域を作成すれば、世界的なレベルで、組織IDがユニークでなくてもよい。"ドメイン領域"という閉じた世界において、複数の組織を区別できれば足りるので、組織IDを、世界的なレベルで、ユニークな値にする必要はなく、第３者機関による管理は不要になる。組織IDが重ならないような管理を行わずとも、世界中の組織から供給されるアプリケーションを1つのプラットフォームで動作させつつ、アプリケーションによる読み出し／書き込みの対象となるファイルの機密性を高めることができる。

【００１０】

第３者機関による管理を必要とせずとも、アプリケーション提供を作成業者に躊躇させる障壁を取り除き、アプリケーションが利用するファイルの独立性・機密性を高めることができるので、アプリケーション実行装置用アプリケーションの提供を、映画作成者、映画配給者、放送局、出版者、ソフトハウスなど、世界中の多くの組織に、呼び掛けることができる。こうすることで、アプリケーションの豊富化を図ることができるため、装置用アプリケーションを充実させることができ、ディスク媒体再生装置としてのアプリケーション実行装置の普及に一層の弾みを付けることができる。

【００１１】

またルート証明書は、装置本体ではなく、記憶手段内のドメイン領域に割り当てられることになる。あるディスク媒体にて供給されるアプリケーションは、そのディスク媒体のディスクルート証明書に対応付けられていれば、そのディスク媒体がアプリケーション実行装置に装填される限りは、必ず動作することが保障される。もっとも、ディスクルート証明書が暴露される可能性がない訳ではないが、そうした場合、そのディスク媒体を使えないようにするか、また、そのディスク媒体についてのディスクルート証明書のみをアップデートすればよく、他のディスクにて供給されたアプリケーションは、従前通り、ディスクルート証明書を用いればよいので、確実な動作保障を実現することができる。

【００１２】

このように世界的なレベルでの組織IDの管理を必要とせず、また、従前のアプリケーションの動作保障を高いレベルに維持することができるので、本発明にかかるアプリケーション実行装置は、映画作品に関する処理を行うアプリケーションを実行するアプリケーション実行装置の世界的な標準化に大きく寄与することができる。

【００１７】

以下、本発明の実施の形態を、図面を参照しながら説明する。

(第１実施形態)

以降、本発明に係るアプリケーション実行装置の実施形態について説明する。まず始めに、アプリケーション実行装置に対して、アプリケーションを供給する記録媒体について説明する。かかる記録媒体として、本実施形態では、BD-ROMを題材に選ぶ。何故なら、BD-ROMにおけるアプリケーションは、上述したような映画作品に関する処理を行うからである。図１（ａ）は、BD-ROMにおけるファイル・ディレクトリ構成を示す図である。本図の第１段目にBD-ROMを示す。BD-ROMは、他の光ディスク、例えばDVDやCDなどと同様にその内周から外周に向けてらせん状に記録領域を持つ。第2段目は、この記録領域を示している。第2段目に示すように、記録領域は、内周の「リードイン領域」と、外周の「リードアウト領域」の間に論理データを記録できる「論理アドレス空間」を有している。また、リードインの内側にはBCA（Burst Cutting Area）と呼ばれるドライブでしか読み出せない特別な領域がある。この領域はアプリケーションから読み出せないため、例えば著作権保護技術などに利用されることがよくある。

【００１８】

「論理アドレス空間」には、ファイルシステム情報（ボリューム）を先頭に映像データ、クラスファイルとその関連情報の入ったJava（登録商標）アーカイブファイル３０２などのデータが記録されている。ファイルシステムとは、UDF（Universal Disk Format）やISO9660などのことであり、通常のパーソナルコンピュータと同じように記録されている論理データをディレクトリ、ファイル構造を使って読み出しする事が可能になっている。第3段目は、BD-ROMのディレクトリ・ファイル構造を示す。このディレクトリ・ファイル構造は、ルートディレクトリ（ROOT）直下、BDDATAディレクトリが置かれるというものである。ディレクトリBDDATAには、次の２種類のファイルが記録されている。

【００１９】

（A）BD.ROOT.CERTIFICATE：ディスクルート証明書３０１

ディスクルート証明書３０１とは、このBD-ROMを作成した作成者が、ルート認証局から配布を受けたルート証明書を、BD-ROMに割り当てたものである。ディスクルート証明書３０１はたとえばX.509の形式で符号されている。X.509の仕様は国際電信電話諮問委員会より発行されており、CCITTRecommendation X.509 (1988), "The Directory - AuthenticationFramework"に記載されている。かかるルート証明書を、可搬型の記録媒体に記録することは、ルート証明書が暴露される可能性が高く、DVDでは、かかる割り当ては、ルート証明書の暴露の危険性があるとして、導入されていなかった。しかしBD-ROMには、DVDよりかなり高度な著作権保護技術が採用されており、かかる著作権保護技術の採用は、"BD-ROMに固有のルート証明書を割り当てる"との考えの導入の追い風になった。BD-ROMにおけるディスクルート証明書３０１の導入には、以上のような背景があることに留意されたい。

（B）XXX.JAR：Java（登録商標）アーカイブファイル３０２

これは、http：//Java（登録商標）.sun.com/j2se/1.4.2/docs/guide/jar/jar.htmlに記載された仕様に準じた、Java（登録商標）アーカイブファイル３０２である。Java（登録商標）アーカイブファイル３０２は、ZIPファイルの形式を、Java（登録商標）に特化したものであり、市販されているZIP展開ソフトウェアにより中身を確認することができる。ここで「XXX」は可変、拡張子「JAR」は固定である。

【００２０】

Java（登録商標）アーカイブファイル３０２は複数のファイルをディレクトリ構造の形で格納している。図１（ｂ）はJava（登録商標）アーカイブファイル３０２の中の構造の一例を示す図である。

この構造は、ルートディレクトリ直下にXXXX.classが存在し、META-INFディレクトリにファイルMANIFEST.MF、ファイルSIG-BD.SF、ファイルSIG-BD.RSA、ファイルbd.XXXX.permが存在するというものである。以下、これらのファイルについて個別に説明してゆく。

(i)XXXX.class:クラスファイル４０１

クラスファイル４０１は、バーチャルマシーン上で実行することができるJava（登録商標）アプリケーションを定義するような構造体を格納したクラスファイル４０１である。

【００２１】

このクラスファイル４０１にて定義されるJava（登録商標）アプリケーションは、Xletインターフェイスを通じて、アプリケーション実行装置のアプリケーションマネージャにより、制御されるJava（登録商標）Xletである。Xletインターフェイスは、"loaded","paused"、"active","destroyed"といった4つの状態をもつ。

(ii)MANIFEST.MF：マニフェストファイル４０２

マニフェストファイル４０２は、デジタル証明書に対応するものであり、Java（登録商標）アーカイブファイル３０２の属性、Java（登録商標）アーカイブファイル３０２内のクラスファイル４０１やデータファイルのハッシュ値が記載されているファイルである。Java（登録商標）アーカイブファイル３０２の属性には、クラスファイル４０１のインスタンスである、Java（登録商標）アプリケーションに付与されるアプリID、Java（登録商標）アーカイブファイル３０２を実行するために最初に実行すべきクラスファイル４０１名がある。上記の二つのJava（登録商標）アーカイブファイル３０２の属性が存在しない場合、Java（登録商標）アーカイブファイル３０２中のクラスファイル４０１のインスタ

ンスであるJava（登録商標）アプリケーションを実行しない。
(iii)SIG-BD.SF：シグネチャファイル４０３、
シグネチャファイル４０３は、マニフェストファイル４０２のハッシュ値が記載されているファイルである。
(iv)SIG-BD.RSA：デジタルシグネチャファイル４０４、
デジタルシグネチャファイル４０４は、「デジタル証明書チェーン」、シグネチャファイル４０３の「署名情報」が記載されているファイルである。
【００２２】
シグネチャファイル４０３に対する「署名情報」は、署名処理をシグネチャファイル４０３に施すことで得られる。これらの署名処理には、デジタルシグネチャファイル４０４内のデジタル証明書チェーンにおける公開鍵に対応する秘密鍵が用いられる。
「デジタル証明書チェーン」とは、一つ目の証明書（ルート証明書）が二つ目の証明書を署名し、また同じようにn番目の証明書がn+１番目の証明書を署名している形をもつ複数の証明書群である。デジタル証明書チェーンの最後の証明書を「リーフ証明書」と呼ぶ。この構成を利用することにより、ルート証明書から順番に次の証明書を保障していくことにより、デジタル証明書チェーンの最後の証明書までを保障することができる。
【００２３】
「ルート証明書」は、BD.ROOT.CERTIFICATEファイルに存在するディスクルート証明書３０１と同じ証明書を格納している。
「リーフ証明書」には、組織IDが記載されている。シグネチャファイル４０３は、PKCS#7という形式により格納されている。PKCS#7は署名および一つ以上のデジタル証明書を格納するためのファイル形式であり、IETF（Internet Engineering Task Force）より発行されたRFC２３１５に記載されている。RFC２３１５はhttp://www.ietf.org/rfc/rfc2315.txtより参照できる。
【００２４】
通常、このデジタル証明書チェーンは、1つであるが、後述する権限の提供がある場合、このデジタル証明書チェーンは2つの作られる。これら2つのデジタル証明書チェーンを、第１デジタル証明書チェーン、第２デジタル証明書チェーンと呼ぶ。第１デジタル証明書チェーンのルート証明書は、権限提供を受ける側の組織のディスクルート証明書を示し、リーフ証明書は、権限提供を受ける側の組織の組織IDを示す。第２デジタル証明書チェーンのルート証明書は、権限を提供する側の組織のディスクルート証明書を示し、リーフ証明書は、権限を提供する側の組織の組織IDを示す。一方、権限の提供がない場合、デジタル証明書チェーンは、1つ(第１デジタル証明書チェーン)のみになる。
【００２５】
マニフェストファイル４０２、シグネチャファイル４０３、デジタルシグネチャファイル４０４の詳細はJava（登録商標）アーカイブファイルの仕様に記載されている。マニフェストファイル４０２、シグネチャファイル４０３、デジタルシグネチャファイル４０４は署名処理および署名検証処理を行うために利用される。最終的にJava（登録商標）アーカイブファイル３０２中のクラスファイル４０１のインスタンスであるJava（登録商標）アプリケーションやパーミッションリクエストファイル４０５をデジタル証明書により署名することが可能になる。以降マニフェストファイル４０２、シグネチャファイル４０３、デジタルシグネチャファイル４０４をまとめて「デジタル証明書による署名」と称する。
【００２６】
(v)bd.XXXX.perm:パーミッションリクエストファイル４０５
パーミッションリクエストファイル４０５は、実行されるJava（登録商標）アプリケーションにどのパーミッションを与えるのかの情報を格納するファイルである。具体的に以下の情報を格納する。
（ア）Credential(デジタル信用証明書)
（イ）アプリ間通信の許可情報

以降、（ア）Credentialについて説明する。"Credential"とは、ある組織に帰属する組織ディレクトリ内のファイルを共有化するための情報である。この共有化は、ある組織に属するアプリケーション用ファイルを利用する権限を、他の組織に属するアプリケーションに提供することでなされる。そのためCredentialは、権限を提供する側の組織を示す提供者組織ID、権限を受領する側の組織の識別を示す受領者組織IDを含む。

【００２７】

図２（ａ）は、Credentialのデータ構造の一例を示す。Credentialには、ルート認証局から提供者組織に配布されたルート証明書のハッシュ値５０１、提供者組織に割り当てられた提供者組織ID５０２、ルート認証局から受領者に配布された受領者ルート証明書のハッシュ値５０３、受領者組織に割り当てられた受領者組織ID５０４、受領者アプリID５０５、提供ファイルリスト５０６からなる。提供ファイルリスト５０６には、一つ以上の提供ファイル名５０７とアクセス方法５０８（読み取り可・書き込み可）の情報が格納されている。Credentialが有効であるためには署名されなければならない。Credentialの署名にはデジタルシグネチャファイル４０４と同様にPKCS#7の方式を利用できる。

【００２８】

図２（ｂ）は、Credentialの具体的な一例を示す図である。本図におけるCredentialは、ファイル「４/５/scores.txt」に対して読み取り許可、ファイル「４/５/etc/settings.txt」に対して読み書き許可をCredentialにより与えていることになる。

続いて（イ）アプリ間通信の許可情報について説明する。一つのJava（登録商標）アーカイブファイル３０２に含まれるJava（登録商標）アプリケーションは通常、他のJava（登録商標）アーカイブファイル３０２に含まれるJava（登録商標）アプリケーションとの通信（アプリ間通信）が許可されない。パーミッションリクエストファイル４０５にアプリ間通信の許可が与えられている場合だけアプリ間通信が可能である。

【００２９】

以上が、パーミッションリクエストファイル４０５についての説明である。続いて、ルート証明書についてのより詳しく説明する。

図３（ａ）は、BD-ROMにおいてルート証明書がどのように割り当てられるかを模式的に示す図である。本図の第１段目は、機器（アプリケーション実行装置）と、この機器に装填されるBD-ROMを示し、第２段目は、これら機器及びBD-ROMを作成する業者（BD-ROMの作成者、機器の製造業者）を示す。第３段目は、ルート証明書を管理するルート認証局を示す。

【００３０】

本図において、BD-ROMの作成者は、ルート認証局からルート証明書の配布を受け(f1)、配布されたルート証明書を、ディスクルート証明書３０１としてBD-ROMに割り当てた上で、このルート証明書をBD.ROOT.CERTIFICATEに格納して、BD-ROMに書き込む(w1)。一方、Java（登録商標）アーカイブファイル３０２を作成するにあたって、このルート証明書と、組織IDを示すリーフ証明書とをSIG-BD.SFに収録して、Java（登録商標）アーカイブファイル３０２に包含させる。

【００３１】

本発明の実施形態の説明ではないが、対比のため、MHPにおけるルート証明書の割り当てについて説明する。

図３（ｂ）は、MHPにおいてルート証明書がどのように割り当てられるかを示す図である。MHPでは、機器を製造する製造業者が、ルート認証局からルート証明書の配布を受け(f2)、当該製造業者は、このルート証明書を機器に割り当てる(w2)。一方放送コンテンツの作成業者は、このルート証明書と、自身の組織IDを示すリーフ証明書とを、アプリケーションを定義するクラスファイルに添付した上で、機器に送り込む。これらの図を比較すると、MHPでは機器に割り当てられていたルート証明書を、BD-ROMに割り当てており、このBD-ROMに割り当てたルート証明書から、証明書チェーンを形成していることがわかる。

【００３２】

BD-ROMではなく、Java（登録商標）アーカイブファイル３０２が、WWWサーバからダウ

ンロードされて、アプリケーション実行装置内の記憶装置に書き込まれる場合も同様である。かかるダウンロードは、BD-ROMに記録されたコンテンツのアップデートを目的とするものだが、かかるダウンロードにおいて、BD.ROOT.CERTIFICATEに収録され、ディスクルート証明書３０１として書き込まれているルート証明書と同一性をもつルート証明書を、SIG-BD.SFに格納してJava（登録商標）アーカイブファイルに包含させる。こうすることで、BD-ROMに記録されたコンテンツのアップデートを目的とするJava（登録商標）アーカイブファイル３０２を、ダウンロードにてアプリケーション実行装置に供給する場合も、BD-ROMに割り当てたディスクルート証明書３０１を用いることにより、Java（登録商標）アーカイブファイル３０２の正当性をアプリケーション実行装置に確認させることができる。

【００３３】

図４は、権限の提供がない場合のSIG-BD.RSA、SIG-BD.SF、BD.ROOT.CERTIFICATE、MANIFEST.MF、bd.XXXX.permの相互関係を示す図である。本図における矢印d1は、これらのファイルの内部構成の情報要素のうち、同一性があるものを示している。権限の提供がない場合、BD.ROOT.CERTIFICATE内のルート証明書（ディスクルート証明書３０１）は、SIG-BD.RSAにおける第１のデジタル証明書チェーン内のルート証明書と同一性を有したものになる。

【００３４】

MANIFEST.MFは、クラスファイルXXXX.classを署名し、SIG-BD.SFは、MANIFEST.MFから算出されたハッシュ値を含み、SIG-BD.RSAは、SIG-BD.SFから算出されたハッシュ値を含んでいるので(矢印h1)、これらの署名が正しいかどうかを確認し、これらの図に示す同一性を判定することで、アプリケーション実行装置は、Java（登録商標）アーカイブファイル３０２が正当なものか、改竄が加えられたものかを判断することができる。また権限の提供がないとの想定なので、本図では、bd.XXXX.permを示していない。

【００３５】

図５は、権限の提供がない場合のSIG-BD.RSA、SIG-BD.SF、BD.ROOT.CERTIFICATE、MANIFEST.MF、bd.XXXX.permの相互関係を示す図である。本図における矢印d1～d6は、これらのファイルの内部構成の情報要素のうち、同一性があるものを示している。権限の提供権限の提供があった場合も、BD.ROOT.CERTIFICATE内のルート証明書（ディスクルート証明書）は、SIG-BD.RSAにおける第１のデジタル証明書チェーン内のルート証明書と同一性を有したものになる(矢印d1)。一方、権限の提供があると、BD.ROOT.CERTIFICATE内のディスクルート証明書３０１は、受領者のものになるので、bd.XXXX.permにおけるCredentialの、受領者ルート証明書が、BD.ROOT.CERTIFICATE内のルート証明書と同一性をもつ(矢印d2)。また、Credentialにおける受領者組織IDは、第１のデジタル証明書チェーンにおけるリーフの組織IDと同一性をもつ(矢印d3)。

【００３６】

bd.XXXX.permにおけるCredentialの、提供者組織のルート証明書は、SIG-BD.RSA内の第２のデジタル証明書チェーンにおけるルート証明書と同一性をもつ(矢印d4)。また、Credentialにおける提供者組織IDは、SIG-BD.RSAの第２のデジタル証明書チェーンにおけるリーフの組織IDと同一性をもつ(矢印d5)。Credentialにおける受領者アプリIDは、bd.XXXX.permにおいてCredential以外の部分に存在するアプリIDと同一性をもつ(矢印d6)。

【００３７】

MANIFEST.MFは、クラスファイルXXXX.classから算出したハッシュ値を含み、SIG-BD.SFは、MANIFEST.Mから算出したハッシュ値を含み、SIG-BD.RSAは、SIG-BD.SFから取得したハッシュ値を含むので(矢印h1)、これらの署名が正しいかどうかを確認し、これらの図に示す同一性を判定することで、アプリケーション実行装置は、Java（登録商標）アーカイブファイル３０２が正当なものか、改竄が加えられたものかを判断することができる。あらかじめ断っておくが、本実施形態では、ルート証明書の同一性を、それぞれのルート証明書について算出されたハッシュ値を比較し、これらのハッシュ値が一致しているかどうかで判断するものとする。またハッシュ値の算出は一度行えばよく、算出したものをメモ

リ等に記憶して、利用することが一般的に行われる。ルート証明書からハッシュ値を算出することや、メモリに格納したハッシュ値を取出すことを、ハッシュ値の「取得」と呼ぶ。

【００３８】

以上が、BD-ROMについての説明である。続いて、本発明にかかるアプリケーション実行装置の内部構成について説明する。

本実施形態にかかるアプリケーション実行装置は、CPU、ROM、RAM、ハードディスクドライブ、BD-ROMドライブ、AVデコーダ、入出力機器等を具備したコンピュータシステムに、Java（登録商標）2Micro_Edition(J2ME)Personal Basis Profile(PBP 1.0)と、Globally Executable MHPspecification(GEM1.0.2)for package media targetsとをフル実装することでJava（登録商標）プラットフォームを構成し、以下に示す機能的な構成要素をこのJava（登録商標）プラットフォームに設けることで工業的に生産することができる。

【００３９】

図６は、本実施の形態におけるアプリケーション実行装置の機能構成を示すブロック図である。アプリケーション実行装置はBDドライブ１、アプリケーションマネージャ２、バーチャルマシーン３、ハードディスク４、セキュリティマネージャ５で構成される。

（BDドライブ１）

BDドライブ１は、BD-ROMのローディング/イジェクトを行い、BD-ROM内のデータのアクセスを実行する。BD-ROMをBDドライブ１にローディング/イジェクトした場合、その旨を、BDドライブ１がアプリケーションマネージャ２を通知する。

【００４０】

（アプリケーションマネージャ２）

アプリケーションマネージャ２は、バーチャルマシーン３内のヒープ領域内で動作するシステムアプリケーションであり、アプリケーションシグナリングを実行する。"アプリケーションシグナリング"とは、GEM1.0.2が規定するMHP(MultimedeiHome Platform)において、"サービス"を生存区間としてアプリケーションの起動、実行を行う制御をいう。本実施形態におけるアプリケーションマネージャ２は、この"サービス"の代わりに、BD-ROMにおける"タイトル"を生存区間にして、アプリケーションの起動、実行の制御を実現する。ここで、"タイトル"とは、BD-ROMに記録されている映像・音声データの再生単位であり、アプリケーション管理テーブル(AMT)が一意に割り当てられている。

【００４１】

アプリケーションを起動させるにあたって、アプリケーションマネージャ２は、起動しようとするアプリケーションが正当なものかどうかを判定する。この判定手順は以下の通りである。BD-ROMがローディングされれば、/BDDATA/BD.ROOT.CERTIFICATEというファイルの存在を確認する。ファイルが存在する場合、アプリケーションマネージャ２がこのディスクルート証明書３０１をBD-ROMから読み取り、メモリ上に保持する。その後、Java（登録商標）アーカイブファイル３０２を読みだし、このJava（登録商標）アーカイブファイル３０２に存在する署名を検証する。この検証が正しいのなら、アプリケーションマネージャ２はBD-ROM上に存在するJava（登録商標）アーカイブファイル３０２中のクラスファイル４０１を、バーチャルマシーン３内に読みだし、このクラスファイル４０１のインスタンスをヒープ領域に生成することで、Java（登録商標）アプリケーションを起動する。

【００４２】

（バーチャルマシーン３）

バーチャルマシーン３は、クラスファイルをBD-ROMから読み出すユーザクラスローダ、クラスファイルに対応するインスタンスをJava（登録商標）アプリケーションとして格納するヒープメモリ、スレッド、Java（登録商標）スタックから構成されるJava（登録商標）アプリケーションの実行主体である。ここでスレッドは、Java（登録商標）アプリケーションにおけるメソッドを実行する論理的な実行主体であり、ローカル変数や、オペランドスタックに格納された引数をオペランドにして演算を行い、演算結果を、ローカル変数

又はオペランドスタックに格納する。スレッドによるメソッド実行は、メソッドをなすバイトコードを、CPUのネイティブコードに変換した上、CPUに発行することでなされる。このネイティブコード変換については、本願の主眼から外れるため、説明を省く。Java（登録商標）アーカイブファイル３０２内にパーミッションリクエストファイル４０５が存在する場合、マニフェストファイル４０２の中に、Java（登録商標）アプリケーションの正しいハッシュ値が入っていなければ、そのJava（登録商標）アプリケーションを実行してはならない。かかるハッシュ値の判定のため、バーチャルマシーン３は、実行するJava（登録商標）アプリケーションはどのJava（登録商標）アーカイブファイル３０２に格納されていたのかを示す情報をメモリ上に保持する。このパーミッションリクエストファイル４０５を参照することで、バーチャルマシーン３はアプリケーションマネージャ２が保持するアプリ間通信の許可を確認し、Java（登録商標）アプリケーションに対してアプリ間通信の機能を提供する。

【００４３】

（ハードディスク４）

ハードディスク４は、Java（登録商標） IO Packageからのメソッドを用いることにより、アクセスすることができるローカルストレージである。このローカルストレージは、複数のドメイン領域を有する。ここでドメイン領域とは、各ディスクルート証明書３０１に対応するディレクトリ(図中のR1,R2)のことであり、これらのディレクトリの配下に、組織毎のディレクトリ(図中のorg1,org2,org3)を格納するというものである。組織のアプリケーション毎のディレクトリ(図中のorg1/app1,org1/app2,org1/app3・・・・)は、MHPのものと同じである。つまりローカルストレージでは、MHPに規定された各組織のアプリケーション毎のディレクトリ(図中のorg1/app1,org1/app2,org1/app3・・・・)を、ルート証明書に対応したディレクトリ(図中のR1,R2)の配下に配置した構成になっている。こうすることで、MHPの格納方式と、互換を維持することができる。ここでローカルストレージのディレクトリ構成をアクセスするためのファイルパスのうち、ルート証明書に対応した部分まで(図中のRoot/R1,Root/R2)を、"ローカルストレージルート"という。

【００４４】

（セキュリティマネージャ５）

セキュリティマネージャ５は、ルート証明書から算出されたハッシュ値と、ローカルストレージルートとの組みを複数示したハッシュ管理テーブルを保持しており、アプリケーションから、ファイルの読み出し／書き込みが要求されれば、要求元のアプリケーションに対応するルート証明書について、ハッシュ値を算出して、そうして算出したハッシュ値に対応したローカルストレージルートをハッシュ管理テーブルから選ぶ。そうして選んだローカルストレージルートを、ファイルパスに組み入れる。また、Credentialに基づきファイルパスの組織IDに対応するディレクトリを置き換える。こうすることで、アプリケーションのファイルパスの記述は、MHPに規定されたものと互換を保つことができる。

【００４５】

以降、アプリケーションマネージャ２、セキュリティマネージャ５の具体的なソフトウェアによる実装について説明する。アプリケーションマネージャ２は、図７に示すようなプログラムを作成して、CPUに実行させることで、アプリケーション実行装置に実装することができる。

図７は、アプリケーションマネージャ２による、Java（登録商標）アーカイブファイル３０２内のクラスファイル４０１に基づくアプリケーションの起動手順を示すフローチャートである。アプリケーションマネージャ２はJava（登録商標）アーカイブファイル３０２の中にSIG-BD.SF、SIG-BD.RSA、bd.XXXX.permが存在するかどうかを確認する（SA０１）。一つも存在しない場合、Java（登録商標）アプリケーションが改ざんされている可能性があり、Java（登録商標）アプリケーションの実行を行わない（SA０４）。

【００４６】

上記のファイル３つとも存在する場合、MANIFEST.MF、SIG-BD.SF、SIG-BD.RSAを利用しbd.XXXX.permおよびJava（登録商標）アプリケーションを署名検証する（SA０３）。署名

検証が成功しなかった場合、有効なbd.XXXX.permがないことを記憶し、デフォルトのパーミッションを有するクラスファイル４０１をバーチャルマシーン３に実行させる（SA０２）。

署名検証が成功した場合、第１のデジタル証明書チェーンに存在するルート証明書と、BD.ROOT.CERTIFICATE内のディスクルート証明書３０１との同一性を判定する（SA０５）。それぞれのルート証明書が異なる場合、Java（登録商標）アーカイブファイル３０２が不正であると判断し、Java（登録商標）アプリケーションの実行を行わない（SA０４）。

【００４７】

それぞれのルート証明書に同一性がある場合、第１のデジタル証明書チェーンにおけるリーフ証明書内に組織IDが存在しているか否かを確認する（SA０６）。組織IDが存在しない場合、Java（登録商標）アーカイブファイル３０２が不正であると判断し、Java（登録商標）アプリケーションの実行を行わない（SA０４）。

組織IDが存在する場合、bd.XXXX.permの中にCredentialが存在するかどうかを確認する（SA０７）。存在しない場合、ステップSA１０に続く。

【００４８】

Credentialが存在する場合、Credentialを検証する（SA０８）。検証処理の詳細は後述する。Credentialが複数ある場合、このステップは各Credentialに対して行われる。

Credentialの検証に一つでも失敗した場合(SA０９でNo)、Java（登録商標）アーカイブファイル３０２が不正であると判断し、Java（登録商標）アプリケーションの実行を行わない（SA０４）。

Credentialの検証がすべて成功した場合、またはCredentialが存在しない場合、bd.XXXX.permと組織IDとMANIFEST.MF内に存在するアプリIDを記憶し、MANIFEST.MFに記載されている最初に実行すべきクラスファイル４０１をバーチャルマシーンに読み込ませて、これのインスタンスとなるJava（登録商標）アプリケーションを実行させる（SA１０）。

【００４９】

図８を参照して、アプリケーションマネージャ２がCredentialを検証するフローチャートを説明する。図８は、アプリケーションマネージャ２による、Credentialの署名検証の手順を示すフローチャートである。

先ず始めに、受領者ルート証明書のハッシュ値５０３が、BD.ROOT.CERTIFICATEにおけるディスクルート証明書３０１と一致するかどうかを確認する（SY０１）。一致しない場合、検証が失敗する（SY０２）。

【００５０】

受領者組織ID５０４と、第１のデジタル証明書チェーンにおけるリーフ証明書に記載された組織IDと一致するかどうかを確認する（SY０３）。一致しない場合、検証が失敗する（SY０２）。

受領者アプリID５０５が、bd.XXXX.perm内のCredential以外の箇所に記載されたアプリIDと一致するかどうかを確認する（SY０４）。一致しない場合、検証が失敗する（SY０２）。Credentialにおける提供ファイルの名前の先頭が提供者組織ID５０２と一致するかどうかを確認する（SY０５）。一致しない場合、検証が失敗する（SY０２）。

【００５１】

SIG-BD.RSAにおける署名情報の正当性を確認する（SY０６）。署名情報が正当でない場合、Credentialが改ざんされている可能性があり、検証が失敗する（SY０２）。

署名情報が正当である場合、第２のデジタル証明書チェーンにおけるルート証明書のハッシュ値と、Credentialにおける提供者ルート証明書のハッシュ値との同一性を確認する（SY０７）。一致しない場合、Credentialが不正であり、検証が失敗する（SY０２）。

【００５２】

第２のデジタル証明書チェーンにおけるリーフ証明書の組織IDが、Credentialの提供者組織ID５０２と一致するかどうかを確認する（SY０８）。一致しない場合、Credentialが不正であり、検証が失敗する（SY０２）。

すべての確認が成功した場合、検証が成功する（SY０９）。

図９は本発明に係わるアプリケーション実行装置においてアプリケーションマネージャ２が保持する管理情報の一例である。ディスクルート証明書３０１と、実行されているJava（登録商標）アーカイブファイル３０２の「Jarファイル名」と、「組織ID」と、「アプリID」と、「アプリ間通信」と、「Credential」とをテーブルの形で管理する。

【００５３】

図１０は本発明に係わるアプリケーション実行装置においてアプリケーションマネージャ２が保持する管理情報の中のCredentialテーブルの一行の一例である。Credentialテーブルには「提供者ルート証明書のハッシュ値」５０１、「提供者組織ID」５０２、「提供者ファイルリスト」５０６からなる。この「提供者ファイルリスト」５０６には、提供ファイル名５０７、提供者アクセス方法５０８が記載されている。

【００５４】

Java（登録商標）アーカイブファイル３０２にパーミッションが与えられる場合、バーチャルマシーン３がJava（登録商標）アプリケーションの実行前に以下の処理を行う。バーチャルマシーン３はMANIFEST.MFにあるJava（登録商標）アプリケーションのハッシュ値と、Java（登録商標）アプリケーションの実際のハッシュが一致するかどうかを確認し、一致しない場合、Java（登録商標）アプリケーションを実行しない。

【００５５】

Java（登録商標）アプリケーションはバーチャルマシーン３により実行される。図１１を参照して、典型的なJava（登録商標）アプリケーションがローカルストレージを利用したい場合に行う処理手順を説明する。

図１１は、Java（登録商標）アプリケーションがハードディスク４を利用するにあたっての処理手順を示すフローチャートである。

【００５６】

Java（登録商標）アプリケーションは、仮想ローカルストレージルートを取得する(SC01)。ここで仮想ローカルストレージルートとは、MHPにおける形式で、アクセス先となるファイルを指定するファイルパスであり、/Root、又は、/Storage/Rootという名前で表現される。この仮想ローカルストレージルートは、MHPにおけるファイルパスの形式との互換を意図したに過ぎず、ハードディスク４におけるファイルアクセス時において、当該ローカルストレージ名は、ルート証明書ごとに決まるローカルストレージルートに変換される。

【００５７】

続いてJava（登録商標）アプリケーションは組織IDを取得する（SC０２）。組織IDはたとえば「４」の数字である。

Java（登録商標）アプリケーションはローカルストレージルートの名前と組織IDを組み合わせ、読み書きしたいファイルの名前を指定しファイルへの入出力を行う（SC０３）。たとえばローカルストレージルートの名前が「/persistent/0003」、組織IDが「７」、読み書きしたいファイルの相対パスが「８/scores.txt」の場合、「/persistent/0003/7/8/scores.txt」のフルパスよりファイルを指定できる。

【００５８】

Java（登録商標）アプリケーションが図１１のとおりの処理を行うためにセキュリティマネージャ５はJava（登録商標）アプリケーションに対して以下の関数呼び出しを提供する。

（ア）ローカルストレージルートの取得
（イ）組織IDの取得
（ウ）ファイルからの読み出し
（エ）ファイルへの書き込み

図１２を参照して、Java（登録商標）アプリケーションによる「ローカルストレージルートの取得」関数呼び出しの処理のフローチャートを説明する。

【００５９】

まずセキュリティマネージャ５は、呼び出し元のJava（登録商標）アプリケーションに

bd.XXXX.permがあるかどうかを確認する（SD０１）。bd.XXXX.permがあるかどうかはバーチャルマシーン３からJava（登録商標）アプリケーションをインスタンスとするクラスファイル４０１を格納したJava（登録商標）アーカイブファイル３０２を取得し、Java（登録商標）アーカイブファイル３０２の内容を参照することでなされる。bd.XXXX.permがない場合、「ローカルストレージルートの取得」を拒否する（SD０２）。

【００６０】

bd.XXXX.permがある場合、アプリケーションマネージャ２よりディスクルート証明書３０１のハッシュ値を取得する（SD０３）。セキュリティマネージャ５はハッシュ管理テーブル（後述）を確認し、ディスクルート証明書３０１のハッシュ値が既に登録されているかどうかを確認する（SD０４）。登録されている場合、ディスクルート証明書３０１のハッシュ値に対応するローカルストレージルートをJava（登録商標）アプリケーションに返す（SD０５）。

【００６１】

ハッシュ管理テーブルにディスクルート証明書３０１のハッシュ値が登録されていない場合、セキュリティマネージャ５はハッシュ管理テーブルに新たなエントリーを追加する（SD０６）。そのエントリーにはディスクルート証明書３０１のハッシュ値と、テーブル内で一意となるローカルストレージルートが記載される。ハッシュ管理テーブルに新しく登録したエントリーのローカルストレージルートをJava（登録商標）アプリケーションに返す（SD０７）。

【００６２】

図１３を参照して、セキュリティマネージャ５が保持するハッシュ管理テーブルの一例を示す。ハッシュ管理テーブルにはルート証明書のハッシュ値１３０１とローカルストレージルート１３０２が存在する。本図におけるローカルストレージルートの「/0001」,「0003」は、図６のディレクトリ名/R1,/R2をそれぞれ意味するものである。これらのローカルストレージルート１３０２は、ルート証明書のハッシュ値１３０１と一対一に対応している。そのため、同じディスクルート証明書３０１を持つBDディスクにあるJava（登録商標）アーカイブファイル３０２中のクラスファイル４０１のインスタンスであるJava（登録商標）アプリケーションに対しては同じローカルストレージルート１３０２を返し、異なるディスクルート証明書３０１を持つBD-ROMにあるJava（登録商標）アーカイブファイル３０２中のクラスファイル４０１のインスタンスであるJava（登録商標）アプリケーションに対しては異なるローカルストレージルート１３０２を返すことになる。

【００６３】

図１４を参照して、「組織IDの取得」関数呼び出しのフローチャートを説明する。

まずセキュリティマネージャ５は、呼び出し元のJava（登録商標）アプリケーションにbd.XXXX.permがあるかどうかを確認する（SF０１）。bd.XXXX.permがない場合、組織IDを確定できないため「組織IDの取得」を拒否する（SF０２）。 bd.XXXX.permがある場合、アプリケーションマネージャ２よりJava（登録商標）アプリケーションに対応する組織IDを取得しJava（登録商標）アプリケーションに返す（SF０３）。

【００６４】

「ファイル読み出し」関数呼び出しは読み出ししたいファイル名のフルパスをパラメータとしてJava（登録商標）アプリケーションからセキュリティマネージャ５へ伝える。このフルパスは、組織ID/appID/という形式のものであり、アプリケーションは、MHPにおけるローカルストレージ内のファイルをアクセスするのと同じ手順で、アプリケーション実行装置内のファイルをアクセスしようとする。

【００６５】

ファイルアクセスが成功する場合、セキュリティマネージャ５がデータをJava（登録商標）アプリケーションに返す。図１５と図１６を参照して、「ファイル読み出し」関数呼び出しのフローチャートを説明する。

まずセキュリティマネージャ５は、呼び出し元のJava（登録商標）アプリケーションにbd.XXXX.permがあるかどうかを確認する（SH０１）。bd.XXXX.permがない場合、「ファイ

ル読み出し」を拒否する（SH０２）。

【００６６】

bd.XXXX.permがある場合、アプリケーションマネージャ２よりディスクルート証明書３０１のハッシュ値を取得する（SH０３）。

セキュリティマネージャはハッシュ管理テーブルを確認し、ディスクルート証明書３０１のハッシュ値に対応するローカルストレージルート１３０２を取得する（SH０４）。

続いて、ファイル名を指定するフルパスの先頭のルート名が仮想ローカルストレージルートの名前と一致するかどうかを判定する(SH05)。これは、装置に依存した形式(ここではMHPと互換をもつ形式)のファイルパスでアクセス先ファイルを指定しているかどうかを、チェックするものであり、一致する場合はルート名を、ディスクルート証明書に対応した形式である、ローカルストレージルートに変換する(SH13)。一致しない場合はアクセスを拒否する(SH02)。

【００６７】

ここで、変換前後において、フルパスは以下のようになる。

アプリ指定のフルパス：

/仮想ローカルストレージルート/指定組織ID/指定パス

SH05による変換後のパス：

/ローカルストレージルート/指定組織ID/指定パス

その後、セキュリティマネージャ５はファイル名のフルパスを分解する（SH０６）。ファイル名のフルパスは「ローカルストレージルート１３０２+"/"+指定組織ID+"/"+指定パス」の形式を利用するため、ファイル名のフルパスはローカルストレージルート１３０２と指定組織IDと指定パスに分解することができる。分解できない場合、読み出しを拒否する（SH０２）。

【００６８】

セキュリティマネージャ５はアプリケーションマネージャ２より呼び出し元のJava（登録商標）アプリケーションの組織IDを取得する（SH０７）。そして、指定組織IDが提供者組織ID５０２と一致するCredentialをアプリケーションマネージャ２から取得できるかどうかを試みる（SH１０）。Credentialを取得できなかった場合、アクセスを拒否する（SH０２）。

Credentialを取得し得た場合、Java（登録商標）アプリケーションからの指定パスがCredentialの提供ファイル名５０７に存在し、提供アクセス方法５０８にて読み出しが許可されているかどうかを確認する（SH１１）。許可されている場合、セキュリティマネージャ５はCredential内にある提供者ルート証明書のハッシュ値５０１を取得する（SH１２）。

【００６９】

許可されていない場合、セキュリティマネージャ５は指定組織IDとJava（登録商標）アプリケーションの組織IDが一致するかどうかを確認する（SH０８）。一致した場合、Java（登録商標）アプリケーションが指定したフルパスに基づき、ファイルをハードディスクより読み出しアプリに返す（SH０９）。一致してない場合、読み出しを拒否する（SH０２）。

セキュリティマネージャ５はハッシュ管理テーブルを確認し、提供者ルート証明書のハッシュ値５０１が既に登録されているかどうかを確認する（SH１３）。登録されている場合、提供者ルート証明書のハッシュ値５０１に対応するローカルストレージルート１３０２を取得し、提供者ローカルストレージルートとして確定する（SH１４）。

【００７０】

ハッシュ管理テーブルに提供者ルート証明書のハッシュ値５０１が登録されていない場合、セキュリティマネージャ５はハッシュ管理テーブルに新たなエントリーを追加する（SH１５）。そのエントリーには提供者ルート証明書のハッシュ値５０１と、テーブル内で一意となるローカルストレージルート１３０２が記載される。

ハッシュ管理テーブルに新しく登録した行のローカルストレージルート１３０２を提供

者ローカルストレージルートとして確定する（SH１６）。
【００７１】
提供者ローカルストレージルートが確定された後、セキュリティマネージャ５はファイル名のフルパスの中のローカルストレージルート１３０２を提供者ローカルストレージルートに置き換える（SH１７）。
セキュリティマネージャ５は置き換えた後のファイル名のフルパスのファイルをハードディスク４より読み出して、Java（登録商標）アプリケーションに返す（SH１８）。
【００７２】
以上が、ファイル読み出しについての説明である。続いて、ファイル書き込みについて説明する。
「ファイル書き込み」関数呼び出し時は、書き込みしたいファイル名のフルパスと書き込みたいデータをパラメータとしてJava（登録商標）アプリケーションからセキュリティマネージャ５へ伝える。成功する場合、セキュリティマネージャ５がデータをファイルに書き込む。図１７と図１８を参照して、「ファイル書き込み」関数呼び出しのフローチャートを説明する。
【００７３】
まずセキュリティマネージャ５は、呼び出し元のJava（登録商標）アプリケーションにbd.XXXX.permがあるかどうかを確認する（SI０１）。bd.XXXX.permがない場合、「ファイル書き込み」を拒否する（SI０２）。
bd.XXXX.permがある場合、アプリケーションマネージャ２よりディスクルート証明書３０１のハッシュ値を取得する（SI０３）。
【００７４】
セキュリティマネージャはハッシュ管理テーブルを確認し、ディスクルート証明書３０１のハッシュ値に対応するローカルストレージルート１３０２を取得する（SI０４）。
そして、ファイル名を指定するフルパスの先頭のルート名が仮想ローカルストレージルートの名前と一致するかどうかを判定する(SI05)。一致する場合はルート名をローカルストレージルートに変換する(SI13)。一致しない場合はアクセスを拒否する(SI02)。
ここで、変換前後において、フルパスは以下のようになる。
アプリ指定のフルパス：
/仮想ローカルストレージルート/指定組織ID/指定パス
SH05による変換後のパス：
/ローカルストレージルート/指定組織ID/指定パス
その後、セキュリティマネージャ５はファイル名のフルパスを分解する（SI０６）。ファイル名のフルパスは「ローカルストレージルート１３０２+"/"+指定組織ID+"/"+指定パス」の形式を利用するため、ファイル名のフルパスはローカルストレージルート１３０２と指定組織IDと指定パスに分解することができる。分解できない場合、書き込みを拒否する（SI０２）。
【００７５】
セキュリティマネージャ５は呼び出し元のJava（登録商標）アプリケーションの組織IDを取得する（SI０７）。そして、指定組織IDが提供者組織ID５０２と一致するCredentialをアプリケーションマネージャ２より取得するかどうかを試みる（SI１０）。存在しない場合、書き込みを拒否する（SI０２）。
取得できた場合、Java（登録商標）アプリケーションからの指定パスがCredentialの提供ファイル名５０７として存在し、提供アクセス方法５０８にて書き込みが許可されているかどうかを確認する（SI１１）。許可されていない場合、セキュリティマネージャ５は指定組織IDとJava（登録商標）アプリケーションの組織IDが一致するかどうかを確認する（SI０８）。一致した場合、Java（登録商標）アプリケーションが指定したフルパスに基づき、ファイルにデータを書き込む（SI０９）。一致しない場合、書き込みを拒否する（SI０２）。
【００７６】

書き込みが許可されていることが確認できた場合、セキュリティマネージャ５はCredential内にある提供者ルート証明書のハッシュ値５０１を取得する（SI１２）。

セキュリティマネージャ５はハッシュ管理テーブルを確認し、提供者ルート証明書のハッシュ値５０１が既に登録されているかどうかを確認する（SI１３）。登録されている場合、提供者ルート証明書のハッシュ値５０１に対応するローカルストレージルート１３０２を取得し、提供者ローカルストレージルートとして確定する（SI１４）。

【００７７】

ハッシュ管理テーブルに提供者ルート証明書のハッシュ値５０１が登録されていない場合、セキュリティマネージャ５はハッシュ管理テーブルに新たな行を追加する（SI１５）。その行には提供者ルート証明書のハッシュ値５０１と、テーブル内で一意となるローカルストレージルート１３０２が記載される。

ハッシュ管理テーブルに新しく登録した行のローカルストレージルート１３０２を提供者ローカルストレージルートとして確定する（SI１６）。

【００７８】

提供者ローカルストレージルートが確定された後、セキュリティマネージャ５はファイル名のフルパスの中のローカルストレージルート１３０２を提供者ローカルストレージルートに置き換える（SI１７）。

セキュリティマネージャ５は置き換えた後のファイル名のフルパスのファイルを書き込む（SI１８）。

【００７９】

以上のように本実施形態によれば、ローカルストレージ４の内部にはルート証明書ハッシュ値のそれぞれに割り当てられたディレクトリが存在するので、これらのディレクトリの配下に、組織毎、アプリケーション毎の領域を作成すれば、世界的なレベルで、組織IDがユニークでなくてもよい。"ドメイン領域"という閉じた世界において、複数の組織を区別できれば足りるので、組織IDを、世界的なレベルで、ユニークな値にする必要はなく、第３者機関による管理は不要になる。

【００８０】

またルート証明書は、装置本体ではなく、記憶手段内のドメイン領域に割り当てられるので、BD-ROMがアプリケーション実行装置に装填される限りは、必ず動作することが保障される。もっとも、ディスクルート証明書が暴露される可能性がない訳ではないが、そうした場合、そのBD-ROMを使えないようにするか、また、そのBD-ROMについてのディスクルート証明書のみをアップデートすればよく、他のBD-ROMにて供給されたアプリケーションは、従前通り、ディスクルート証明書を用いればよいので、確実な動作保障を実現することができる。

【００８１】

このように世界的なレベルでの組織IDの管理を必要とせず、また、従前のアプリケーションの動作保障を高いレベルに維持することができるので、本発明にかかるアプリケーション実行装置は、映画作品に関する処理を行うアプリケーションを実行するアプリケーション実行装置の世界的な標準化に大きく寄与することができる。

(備考)

以上、本願の出願時点において、出願人が知り得る最良の実施形態について説明したが、 以下に示す技術的トピックについては、更なる改良や変更実施を加えることができる。各実施形態に示した通り実施するか、これらの改良・変更を施すか否かは、何れも任意的であり、実施する者の主観によることは留意されたい。

【００８２】

（「Java（登録商標）アーカイブファイル３０２の選択）

BD-ROMに映像などのデータが共存する場合、映像再生におけるイベント（チャプター２の再生開始等）により指定されるJava（登録商標）アーカイブファイル３０２や、ユーザの画面上の選択に応じたJava（登録商標）アーカイブファイル３０２を選択することも考えられる。

デジタル証明書チェーンが一つの証明書から構成される場合もある。その場合、ルート証明書もリーフ証明書も同じ証明書である。

【００８３】

（パーミッションの種類）

もっと豊富な機能を持つアプリケーション実行装置の場合、bd.XXXX.permに他の種類のパーミッションが含まれていてもよい。bd.XXXX.permの中に複数のデジタル信用証明書３１２が含まれても良い。

（組織ID）

実施の形態によっては組織IDがBD-ROM上の異なるファイルに記載されることが考えられる。そのとき、「組織IDの取得」を拒否する必要はなく、別の方法により確定された組織IDを返すことにしてもよい。

【００８４】

（ファイルの読み出し／書き込み）

図１５、図１６、図１７、図１８ではファイルの読み出しを中心に説明しているが、同様にJava（登録商標）アプリケーションからディレクトリにアクセスすることも可能である。ディレクトリにアクセスする場合、ファイル名のフルパスに指定パスがない場合も考えられる。

【００８５】

また図１５と図１６ではファイル全体を読み取る関数呼び出しとして説明しているが、同様にファイルを部分的に取得したりして典型的なファイルアクセスをすることも可能である。

図１１に示す制御フローは一例であり、Java（登録商標）アプリケーションのつくりによって大きく異なることもある。

【００８６】

（他の方式との併用）

本方式は他のファイルアクセスパーミッション方式（たとえばUNIX（登録商標）によく利用されるユーザ・グループ・ワールドアクセスモデル）と併用することが可能である。たとえば、本方式と第２の方式を併用し以下の優先度を決めるようにしても良い。

（ア）ステップSH０６においてフルパスに分解できなかった場合、あるいはステップSH１０においてデジタル信用証明書が存在しない場合、第２の方式のアクセス制御を利用する。

【００８７】

（イ）ステップSH０９において第２の方式のアクセス制御を利用する。

（ウ）その他に関しては本方式を優先する。

（ハッシュ値）

本実施の形態におけるハッシュ値とはSHA-１やMD５などのセキュアハッシュ関数を利用した結果の値である。セキュアハッシュ関数は、同じハッシュ値を持つ異なるデータを見つけるのは実質不可能であるという特徴を持っている。

【００８８】

（ルート証明書のハッシュ値）

本実施の形態におけるルート証明書のハッシュ値とは、必ずしもルート証明書全体のデータから算出する必要はなく、少なくともルート証明書の中に含まれる公開鍵のデータのみから算出することにしてもよい。MANIFEST.MF、SIG-BD.SF、SIG-BD.RSAの中に格納されるハッシュ値の計算に利用されるセキュアハッシュ関数はディスク作成者が明示的に選択できる。

【００８９】

本実施の形態ではbd.XXXX.permの中にあるデジタル信用証明書には提供者ルート証明書のハッシュ値５０１および受領者ルート証明書のハッシュ値５０３の計算に利用されるセキュアハッシュ関数が固定されていることを前提にしているが、bd.XXXX.permの中にあるデジタル信用証明書の中に、提供者ルート証明書のハッシュ値５０１および受領者ルート

証明書のハッシュ値５０３の計算に利用されるセキュアハッシュ関数を明示するようにしても良い。

【００９０】

（ルート証明書の比較）

ステップSA０５におけるルート証明書の比較は、ルート証明書が同じとあるかどうかの比較、ルート証明書の中に含まれる公開鍵が同じであるかどうかの比較を行うようにしても良い。また別の方式としては、デジタルシグネチャファイルの中にある一つ目の証明書（ルート証明書）を無視し、ルート証明書に続く二つ目の証明書がディスクルート証明書３０１により署名されているかどうかを確認するようにしてもよい。どの方式を使ってもディスクルート証明書３０１がデジタルシグネチャファイルの中の二つ目の証明書を保障していることになるためセキュリティ観点での効果が同じである。

【００９１】

ステップSA０５による比較はアプリケーション間通信を悪用した攻撃を防ぐのが主な目的である。アプリケーション間通信を悪用した攻撃は以下のとおりに作られた攻撃用BD-ROMを利用し試みることが考えられる。

１.攻撃対象のディスク作成者により作成された正当なBD-ROMから、デジタル証明書により署名されている攻撃対象のJava（登録商標）アーカイブファイル３０２を読み取る

２.攻撃するためのJava（登録商標）アーカイブファイル３０２を作成し、デジタル証明書により署名を行う

３.攻撃対象のJava（登録商標）アーカイブファイル３０２と攻撃するためのJava（登録商標）アーカイブファイル３０２を攻撃用BD-ROMに記録

攻撃するためのJava（登録商標）アーカイブファイル３０２と攻撃対象のJava（登録商標）アーカイブファイル３０２はともにデジタル証明書により署名されているが、どちらも異なるルート証明書を利用する。アプリケーション実行装置において二つのJava（登録商標）アーカイブファイル３０２にあるJava（登録商標）アプリケーションに対してアプリ間通信のパーミッションが与えられれば、攻撃するためのJava（登録商標）アーカイブファイル３０２が攻撃対象のJava（登録商標）アーカイブファイル３０２に対して不正なアプリ間通信を行うことが可能になり、攻撃対象のJava（登録商標）アーカイブファイル３０２は自身で利用する記憶領域に対して攻撃対象のディスク作成者により予期しない動作をしうる。

【００９２】

上記の攻撃を防止するためにステップSA０５においてルート証明書の比較が必要である。尚、ステップSA０５の変わりに異なるルート証明書を利用するJava（登録商標）アプリケーション同士のアプリ間通信を防止するようにしてもよい。そのとき、一つのBD-ROMに複数のディスクルート証明書３０１を持つようにしてもよい。

(ローカルストレージ名の名前取得関数)

実施形態では、ローカルストレージ名の名前取得関数のステップSC01において、MHPと互換をもつ形式のファイルパスである、仮想ローカルストレージ名を一旦アプリケーションに返してから、SH05,SI05において、アプリケーション実行装置におけるローカルストレージ名に変換したが、ローカルストレージ名の名前取得関数のステップSC01において、直接、ハッシュ管理テーブルに記述されたローカルストレージ名を直接、Java（登録商標）アプリケーションに返し、仮想ローカルストレージ名からローカルストレージ名への変換を省いてもよい。

【００９３】

（タイトル）

BD-ROM再生装置として、アプリケーション実行装置を製造する場合、BD-ROMの装填やユーザ操作、装置の状態に応じてタイトルを選択する"モジュールマネージャ"をアプリケーション実行装置に設けるのが望ましい。BD-ROM再生装置内のデコーダは、この"モジュールマネージャ"によるタイトル選択に応じて、プレイリスト情報に基づくAVClipの再生を行う。

【００９４】
アプリケーションマネージャ２は、"モジュールマネージャ"がタイトルの選択を行った際、前のタイトルに対応するAMTと、カレントタイトルに対応するAMTとを用いてシグナリングを実行する。このシグナリングは、前のタイトルに対応するAMTには記載されているが、カレントタイトルに対応するAMTには記載されていないアプリケーションの動作を終了させ、前のタイトルに対応するAMTには記載されておらず、カレントタイトルに対応するAMTには記載されているアプリケーションの動作を開始させるという制御を行う。そして、このアプリケーションシグナリングがなされる度に、上述したようなディスクルート証明書を用いた検証を行うのが望ましい。
【００９５】
（BD-BOX）
長編の映画作品やグループ物の映画作品を複数のBD-ROMに記録して、いわゆるBD-BOXを構成する際、これらのBD-ROMには、同一のディスクルート証明書３０１を割り当てるのが望ましい。このように、複数BD-ROMに同一のディスクルート証明書３０１を割り当てた場合、アプリケーションによっては、ディスクの交換前後で、動作するものもでてくる。このように、ディスクの交換前後で動作するアプリケーションを"ディスクアンバウンダリィアプリケーション"という。ここでアプリケーションマネージャ２は、BD-ROMの交換がなされた場合において、新たに装填されたBD-ROMのディスクルート証明書３０１を読みだし、そのディスクアンバウンダリィアプリケーションを定義するJava（登録商標）アーカイブファイル３０２内のディスクルート証明書３０１との同一性を確認するという処理を行うのが望ましい。そして、同一性があれば、ディスクアンバウンダリィアプリケーションの動作を継続させ、もし同一性がなければ、ディスクアンバウンダリィアプリケーションを強制的に終了させる。このようにすることで、BD-ROMの交換の前後において、正当なアプリケーションのみを、動作させることができる。
【００９６】
（Credential）
・Credentialは、複数のXML文書のうち、特定のタグにて囲まれた部分を取出して、これらを結合することで構成するのが望ましい。
・Credentialのデータを提供者組織のリーフ証明書（の公開鍵）により署名した値を、Credentialの署名情報として、bd.XXXXX.permの中に記載してもよい。
・「権限の提供」を行う場合において、bd.XXXXX.permは複数あってもよいが、複数ある場合、bd.XXXXX.permにおいて、SIG-BD.SFのどのリーフ証明書がどのCredentalとの照合に使われるかの情報を記述するのが望ましい。
・Credentialのやり方として、bd.XXXXX.perm中に提供者ファイル名を書いておき、Credentialの実体を、他のファイルに書いてある値から算出してもよい。
・そして、これらを全部組み合わせて、提供者ファイル、リーフ証明書を特定する情報、署名情報がbd.XXXXX.permに得られるようにするのが望ましい。
（ローカルストレージ）
本実施形態においてローカルストレージは、装置組込み型のハードディスクであるとしたが、セキュアな記録媒体であれば、過搬型のものを採用してもよい。例えば、SDメモリカードを採用してもよい。
【００９７】
（実装すべきパッケージ）
アプリケーション実行装置の実施にあたっては、以下のBD-J Extensionをアプリケーション実行装置に実装するのが望ましい。BD-JExtensionは、GEM[1.0.2]を越えた機能を、Java（登録商標）プラットフォームに与えるために特化された、様々なパッケージを含んでいる。BD-JExtensionにて供給されるパッケージには、以下のものがある。
・org.bluray.media
このパッケージは、Java（登録商標） Media FrameWorkに追加すべき、特殊機能を提供する。アングル、音声、字幕の選択についての制御が、このパッケージに追加される。

・org.bluray.ti
このパッケージは、GEM[1.0.2]における"サービス"を"タイトル"にマップして動作するためのAPIや、BD-ROMからタイトル情報を問い合わせる機構や新たなタイトルを選択する機構を含む。
・org.bluray.application
このパッケージは、アプリケーションの生存区間を管理するためのAPIを含む。また、アプリケーションを実行させるにあたってのシグナリングに必要な情報を問い合わせるAPIを含む。
・org.bluray.ui
このパッケージは、BD-ROMに特化されたキーイベントのための定数を定義し、映像再生との同期を実現するようなクラスを含む。
・org.bluray.vfs
このパッケージは、データの所在に拘らず、データをシームレスに再生するため、BD-ROMに記録されたコンテンツ(on-discコンテンツ)と、BD-ROMに記録されていないLocalStorage上のコンテンツ(off-discコンテンツ)とをバインドする機構(Binding Scheme)を提供する。

【００９８】

Binding Schemeとは、BD-ROM上のコンテンツ(AVClip、字幕、BD-Jアプリケーション)と、Local Storage上の関連コンテンツとを関連付けるものである。このBindingSchemeは、コンテンツの所在に拘らず、シームレス再生を実現する。

（ローカルストレージのアクセス）

例えば、ファイルパス「／Persistent／1／1／streams／に、所望のファイルが存在するか確認するは、Java（登録商標）.ioのexits()メソッドを用いることで行われる。所望のファイルが0.m2tsである場合の、Java（登録商標）.ioのexits()メソッドの用例を以下に示す。

例：

```
new Java（登録商標）.io.File("／Persistent／1／1／streams／0.m2ts").exists();
```

（パーミッションリクエストファイル）

パーミッションリクエストファイルは、以下の機能を許可するか否かを定めてもよい。

【００９９】

・ネットワーク接続の利用
・BD-ROMのアクセス
・BD-ROMにおける他のタイトルの選択
・他のプラットフォームの実行制御

（映像・音声）

図１（ａ）に示したBD-ROMのディレクトリ構造において、ROOTディレクトリの配下にBDMVディレクトリを設け、このディレクトリに、MPEG2-TS形式のAVストリームであるAVClipや、これの再生経路を規定するプレイリスト情報を記録してもよい。そしてプレイリスト情報を通じた再生を行うようJava（登録商標）アプリケーションを記述してもよい。

【０１００】

プレイリスト情報が00001.mplsというファイルに格納された場合、Java（登録商標）アプリケーションは、JMFライブラリに基づき、JMFプレーヤインスタンスを生成する。JMFA"BD://00001.mpls";は、PLを再生するプレーヤインスタンスの生成をバーチャルマシンに命じるメソッドである。A.playは、JMFプレーヤインスタンスに再生を命じるメソッドである。

（BD-ROMコンテンツ）

BD-ROMに記録されるアプリケーションは、映画作品を構成するものであるとしたが、ローカルストレージにインストールして利用されるアプリケーションではなく、BD-ROMに記録された状態で利用されるアプリケーションであるなら、これ以外のものを構成するものであってもよい。例えば、ゲームソフトを構成するアプリケーションであってもよい。ま

た、本実施形態ではディスク媒体として、BD-ROMを題材に選んだが、可搬体であり、著作権保護がなされた記録媒体であるなら、他の記録媒体を採用してもよい。

【０１０１】

（Virtual Package）

Virtual Packageを生成させるような処理をセキュリティマネージャ５に行わせてもよい。Virtual Packageとは、BD-ROM等のリードオンリー型の記録媒体に記録されているデジタルストリームと、ハードディスク等のリライタブル型の記録媒体に記録されているデジタルストリームとを動的に組み合わせて、仮想的なパッケージを構築することにより、リードオンリー型記録媒体の内容拡張を図る技術である。ここBD-ROMに記録されているデジタルストリームが、映画作品の本編を構成するものであり、ハードディスクに記録されているデジタルストリームが、映画作品の続編を構成するものである場合、上述したVirtualPackageを構築することにより、BD-ROM上の本編と、ハードディスク上の続編とを、1つの長編の映画作品として取り扱い、再生に供することができる。

【０１０２】

これは、セキュリティマネージャ５がVirtual Package情報を生成することでなされる。Virtual Package情報とは、BD-ROMにおけるボリューム管理情報を拡張した情報である。ここでボリューム管理情報は、ある記録媒体上に存在するディレクトリ－ファイル構造を規定する情報であり、ディレクトリについてのディレクトリ管理情報、ファイルについてのファイル管理情報とからなる。VirtualPackage情報とは、BD-ROMのディレクトリ－ファイル構造を示すボリューム管理情報に、新たなファイル管理情報を追加することにより、BD-ROMにおけるディレクトリ－ファイル構造の拡張を図ったものである。かかるVirtualPackage情報の生成により、アプリケーションは、BD-ROMをアクセスするのと同じ感覚で、ローカルストレージにおけるディスクルート証明書毎のドメイン領域の、組織毎の領域をアクセスすることができる。

【０１０３】

(制御手順の実現)

各実施形態においてフローチャートを引用して説明した制御手順や、機能的な構成要素による制御手順は、ハードウェア資源を用いて具体的に実現されていることから、自然法則を利用した技術的思想の創作といえ、"プログラムの発明"としての成立要件を満たす。

・本発明に係るプログラムの生産形態

本発明に係るプログラムは、コンピュータが実行することができる実行形式のプログラム(オブジェクトプログラム)であり、実施形態に示したフローチャートの各ステップや、機能的構成要素の個々の手順を、コンピュータに実行させるような1つ以上のプログラムコードから構成される。ここでプログラムコードは、プロセッサのネィティブコード、JAVAバイトコードというように、様々な種類がある。またプログラムコードによる各ステップの実現には、様々な態様がある。外部関数を利用して、各ステップを実現することができる場合、この外部関数をコールするコール文が、プログラムコードになる。また、1つのステップを実現するようなプログラムコードが、別々のオブジェクトプログラムに帰属することもある。命令種が制限されているRISCプロセッサでは、算術演算命令や論理演算命令、分岐命令等を組合せることで、フローチャートの各ステップが実現されることもある。

【０１０４】

本発明にかかるプログラムは、以下のようにして作ることができる。先ず初めに、ソフトウェア開発者は、プログラミング言語を用いて、各フローチャートや、機能的な構成要素を実現するようなソースプログラムを記述する。この記述にあたって、ソフトウェア開発者は、プログラミング言語の構文に従い、クラス構造体や変数、配列変数、外部関数のコールを用いて、各フローチャートや、機能的な構成要素を具現するソースプログラムを記述する。

【０１０５】

記述されたソースプログラムは、ファイルとしてコンパイラに与えられる。コンパイラ

は、これらのソースプログラムを翻訳してオブジェクトプログラムを生成する。
コンパイラによる翻訳は、構文解析、最適化、資源割付、コード生成といった過程からなる。構文解析では、ソースプログラムの字句解析、構文解析および意味解析を行い、ソースプログラムを中間プログラムに変換する。最適化では、中間プログラムに対して、基本ブロック化、制御フロー解析、データフロー解析という作業を行う。資源割付では、ターゲットとなるプロセッサの命令セットへの適合を図るため、中間プログラム中の変数をターゲットとなるプロセッサのプロセッサが有しているレジスタまたはメモリに割り付ける。コード生成では、中間プログラム内の各中間命令を、プログラムコードに変換し、オブジェクトプログラムを得る。

【０１０６】

オブジェクトプログラムが生成されるとプログラマはこれらに対してリンカを起動する。リンカはこれらのオブジェクトプログラムや、関連するライブラリプログラムをメモリ空間に割り当て、これらを１つに結合して、ロードモジュールを生成する。こうして生成されるロードモジュールは、コンピュータによる読み出しを前提にしたものであり、各フローチャートに示した処理手順や機能的な構成要素の処理手順を、コンピュータに実行させるものである。以上の処理を経て、本発明に係るプログラムを作ることができる。

・本発明に係るプログラムの使用形態

本発明に係るプログラムは、以下のようにして使用することができる。

【０１０７】

(i)組込プログラムとしての使用

本発明に係るプログラムを組込プログラムとして使用する場合、プログラムにあたるロードモジュールを、基本入出力プログラム(BIOS)や、様々なミドルウェア(オペレーションシステム)と共に、命令ROMに書き込む。こうした命令ROMを、制御部に組み込み、CPUに実行させることにより、本発明に係るプログラムを、アプリケーション実行装置の制御プログラムとして使用することができる。

【０１０８】

(ii)アプリケーションとしての使用

アプリケーション実行装置が、ハードディスク内蔵モデルである場合は、基本入出力プログラム(BIOS)が命令ROMに組み込まれており、様々なミドルウェア(オペレーションシステム)が、ハードディスクにプレインストールされている。また、ハードディスクから、システムを起動するためのブートROMが、アプリケーション実行装置に設けられている。

【０１０９】

この場合、ロードモジュールのみを、過搬型の記録媒体やネットワークを通じて、アプリケーション実行装置に供給し、1つのアプリケーションとしてハードディスクにインストールする。そうすると、アプリケーション実行装置は、ブートROMによるブートストラップを行い、オペレーションシステムを起動した上で、1つのアプリケーションとして、当該アプリケーションをCPUに実行させ、本発明に係るプログラムを使用する。

【０１１０】

ハードディスクモデルのアプリケーション実行装置では、本発明のプログラムを1つのアプリケーションとして使用しうるので、本発明に係るプログラムを単体で譲渡したり、貸与したり、ネットワークを通じて供給することができる。

(アプリケーションマネージャ２～セキュリティマネージャ４)

アプリケーションマネージャ２～セキュリティマネージャ４は、一個のシステムLSIとして実現することができる。

【０１１１】

システムLSIとは、高密度基板上にベアチップを実装し、パッケージングしたものをいう。複数個のベアチップを高密度基板上に実装し、パッケージングすることにより、あたかも1つのLSIのような外形構造を複数個のベアチップに持たせたものも、システムLSIに含まれる(このようなシステムLSIは、マルチチップモジュールと呼ばれる。)。

ここでパッケージの種別に着目するとシステムLSIには、QFP(クッド フラッド アレイ)

、PGA(ピン グリッド アレイ)という種別がある。QFPは、パッケージの四側面にピンが取り付けられたシステムLSIである。PGAは、底面全体に、多くのピンが取り付けられたシステムLSIである。

【０１１２】

これらのピンは、他の回路とのインターフェイスとしての役割を担っている。システムLSIにおけるピンには、こうしたインターフェイスの役割が存在するので、システムLSIにおけるこれらのピンに、他の回路を接続することにより、システムLSIは、アプリケーション実行装置の中核としての役割を果たす。

システムLSIにパッケージングされるベアチップは、"フロントエンド部"、"バックエンド部"、"デジタル処理部"からなる。"フロントエンド部"は、アナログ信号を、デジタル化する部分であり、"バックエンド部"はデジタル処理の結果、得られたデータを、アナログ化して出力する部分である。

【０１１３】

各実施形態において内部構成図として示した各構成要素は、このデジタル処理部内に実装される。

先に"組込プログラムとしての使用"で述べたように、命令ROMには、プログラムにあたるロードモジュールや、基本入出力プログラム(BIOS)、様々なミドルウェア(オペレーションシステム)が書き込まれる。本実施形態において、特に創作したのは、このプログラムにあたるロードモジュールの部分なので、プログラムにあたるロードモジュールを格納した命令ROMを、ベアチップとしてパッケージングすることにより、本発明に係るシステムLSIは生産することができる。

【０１１４】

具体的な実装については、SoC実装やSiP実装を用いることが望ましい。SoC(System on chip)実装とは、1チップ上に複数の回路を焼き付ける技術である。SiP(Systemin Package)実装とは、複数チップを樹脂等で1パッケージにする技術である。以上の過程を経て、本発明に係るシステムLSIは、各実施形態に示したアプリケーション実行装置の内部構成図を基に作ることができる。

【０１１５】

尚、上述のようにして生成される集積回路は、集積度の違いにより、IC、LSI、スーパーLSI、ウルトラLSIと呼称されることもある。

さらに、各アプリケーション実行装置の構成要素の一部又は全てを1つのチップとして構成してもよい。集積回路化は、上述したSoC実装,SiP実装に限るものではなく、専用回路又は汎用プロセスで実現してもよい。LSI製造後に、プログラムすることが可能なFPGA（FieldProgrammable Gate Array）や、LSI内部の回路セルの接続や設定を再構成可能なシリコンフィギュラブル・プロセッサを利用することが考えられる。更には、半導体技術の進歩又は派生する技術によりLSIに置き換わる集積回路化の技術が登場すれば、当然、その技術を用いて機能ブロックの集積回路化を行っても良い。例えば、バイオ技術の適応などが可能性としてありうる。

【産業上の利用可能性】

【０１１６】

本発明に係るアプリケーション実行装置は、上記実施形態に内部構成が開示されており、この内部構成に基づき量産することが明らかなので、資質において工業上利用することができる。このことから本発明に係るアプリケーション実行装置は、産業上の利用可能性を有する。

【図面の簡単な説明】

【０１１７】

【図１】（ａ）BD-ROMにおけるファイル・ディレクトリ構成を示す図である。　（ｂ）Java（登録商標）アーカイブファイル３０２の中の構造の一例を示す図である。

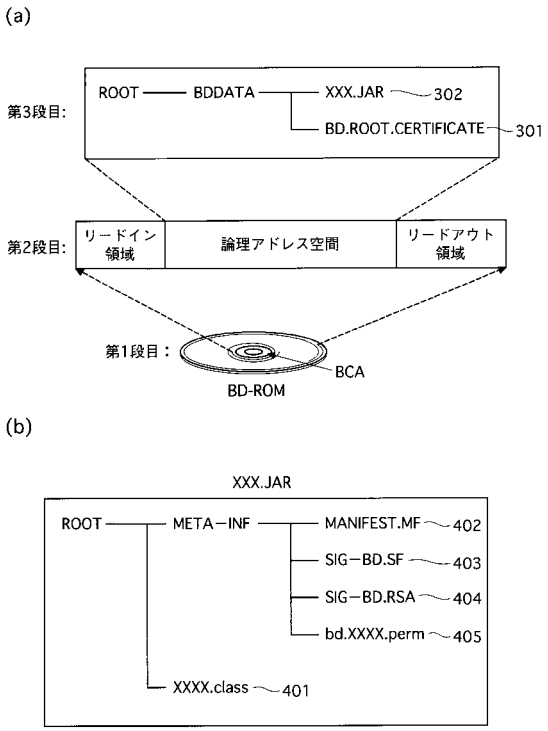
【図２】（ａ）Credentialのデータ構造の一例を示図である。

【０１１８】

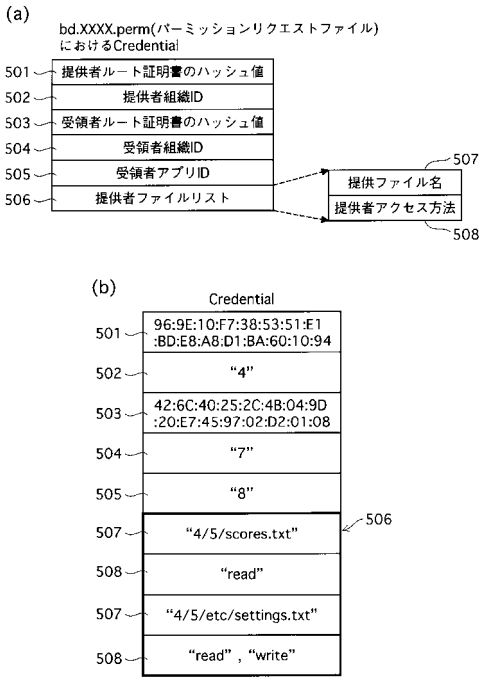
（ｂ）Credentialの具体的な一例を示す図である。
【図３】（ａ）BD-ROMにおいてルート証明書がどのように割り当てられるかを模式的に示す図である。　（ｂ）MHPにおいてルート証明書がどのように割り当てられるかを模式的に示す図である。
【図４】権限の提供がない場合のSIG-BD.RSA、SIG-BD.SF、BD.ROOT.CERTIFICATE、MANIFEST.MF、bd.XXXX.permの相互関係を示す図である。
【図５】権限の提供がある場合のSIG-BD.RSA、SIG-BD.SF、BD.ROOT.CERTIFICATE、MANIFEST.MF、bd.XXXX.permの相互関係を示す図である。
【図６】本実施の形態におけるアプリケーション実行装置の機能構成を示すブロック図である。
【図７】アプリケーションマネージャ２による、Java（登録商標）アーカイブファイル３０２内のクラスファイルに基づくアプリケーションの起動手順を示すフローチャートである。
【図８】アプリケーションマネージャ２による、Credentialの署名検証の手順を示すフローチャートである。
【図９】アプリケーションマネージャ２が保持する管理情報の一例を示す図である。
【図１０】アプリケーションマネージャ２が保持する管理情報の一例を示す図である。
【図１１】Java（登録商標）アプリケーションがハードディスク４を利用するにあたっての処理手順を示すフローチャートである。
【図１２】セキュリティマネージャ５によるローカルストレージ名の取得手順の詳細を示すフローチャートである。
【図１３】セキュリティマネージャ５が保持するハッシュ管理テーブルの一例を示す図である。
【図１４】セキュリティマネージャ５による組織IDの取得関数の詳細を示すフローチャートである。
【図１５】セキュリティマネージャ５によるファイル読出関数の詳細を示すフローチャートである。
【図１６】セキュリティマネージャ５によるファイル読出関数の詳細を示すフローチャートである。
【図１７】セキュリティマネージャ５によるファイル書込関数の詳細を示すフローチャートである。
【図１８】セキュリティマネージャ５によるファイル書込関数の詳細を示すフローチャートである。
【符号の説明】
【０１１９】
１　BDドライブ
２　アプリケーションマネージャ
３　バーチャルマシーン
４　ハードディスク
５　セキュリティマネージャ
３０１　ディスクルート証明書
３０２　Java（登録商標）アーカイブファイル
４０１　クラスファイル
４０２　マニフェストファイル
４０３　シグネチャファイル
４０４　デジタルシグネチャファイル
４０５　パーミッションリクエストファイル
５０１　提供者ルート証明書のハッシュ値
５０２　提供者組織ID
５０３　受領者ルート証明書のハッシュ値

５０４　受領者組織ID
５０５　受領者アプリID
５０６　ファイルリスト

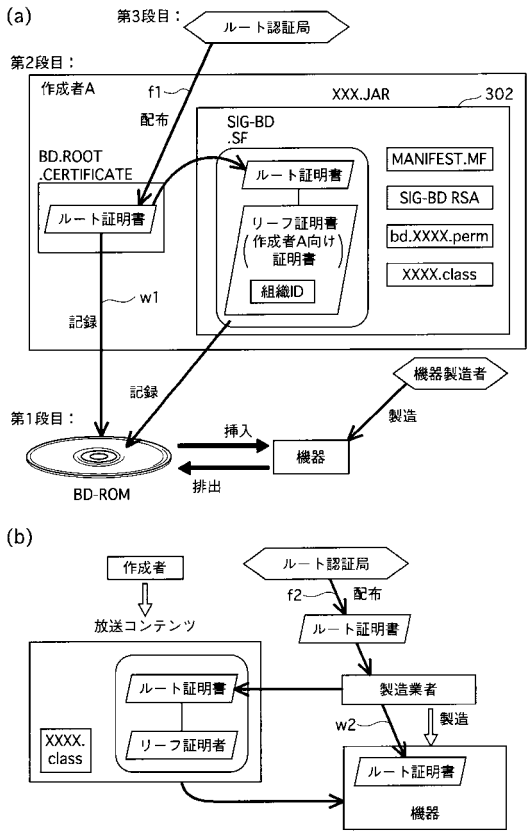
【図１】



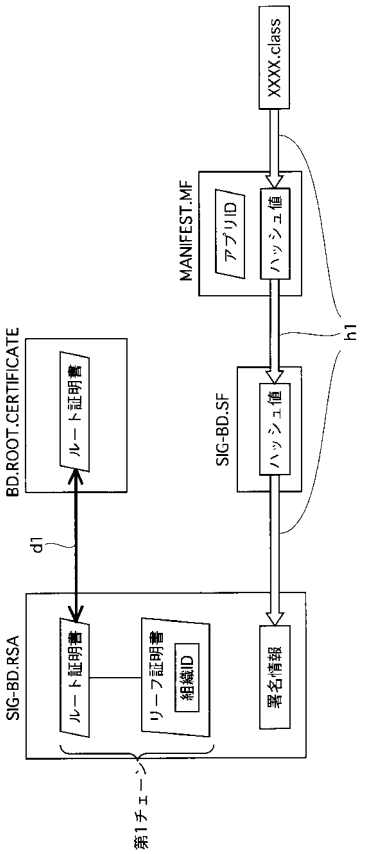
【図２】



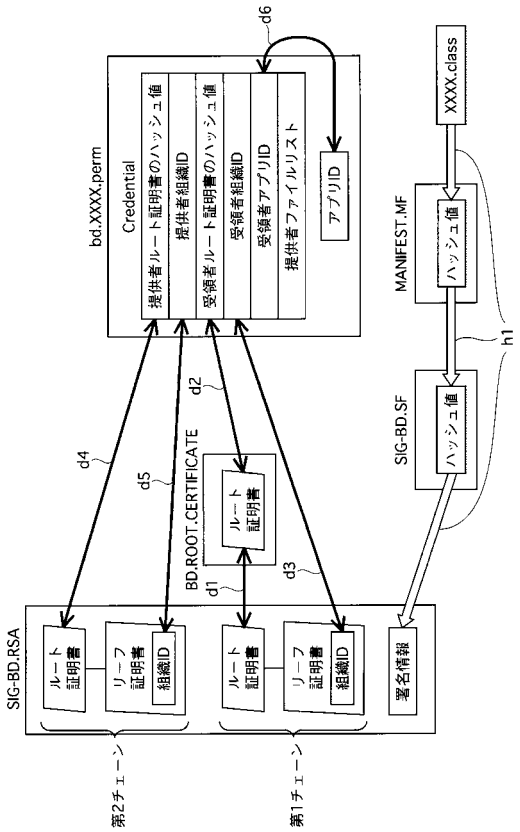
【図３】


(a)
第3段目：
ルート認証局
第2段目：
作成者A
f1
配布
XXX.JAR
302
SIG-BD
.SF
BD.ROOT
.CERTIFICATE
ルート証明書
ルート証明書
リーフ証明書
(作成者A向け)
証明書
組織ID
MANIFEST.MF
SIG-BD RSA
bd.XXXX.perm
XXXX.class
w1
記録
記録
機器製造者
製造
第1段目：
挿入
機器
排出
BD-ROM
(b)
作成者
放送コンテンツ
ルート認証局
f2
配布
ルート証明書
製造業者
ルート証明書
リーフ証明者
XXXX.
class
w2
製造
ルート証明書
機器

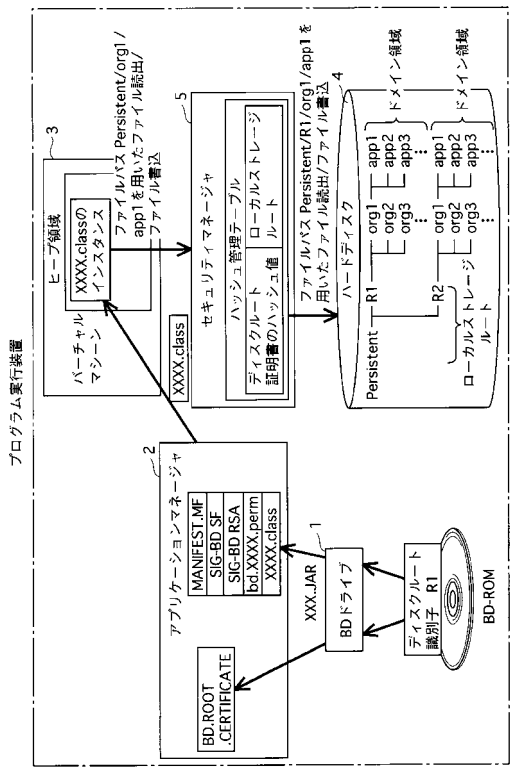
【図４】


XXXX.class
MANIFEST.MF
アプリID
ハッシュ値
h1
BD.ROOT.CERTIFICATE
ルート証明書
SIG-BD.SF
ハッシュ値
d1
SIG-BD.RSA
ルート証明書
リーフ証明書
組織ID
署名情報
第1チェーン

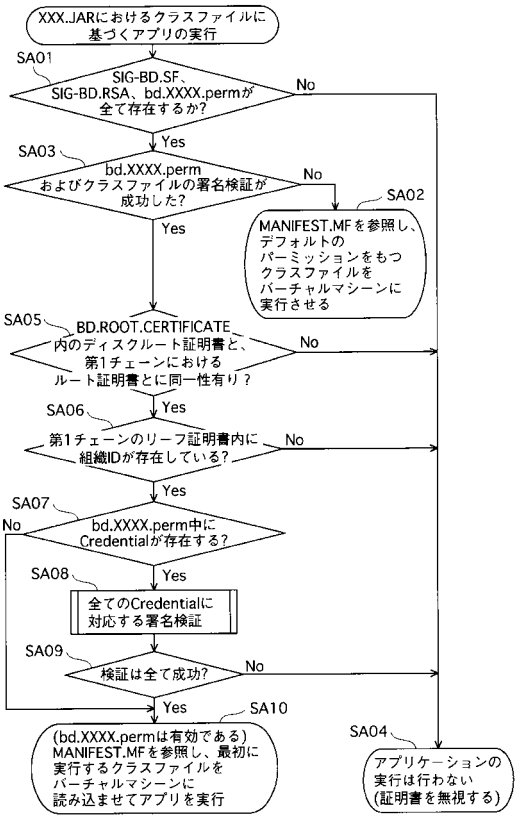
【図５】


d6
bd.XXXX.perm
Credential
提供者ルート証明書のハッシュ値
提供者組織ID
受領者ルート証明書のハッシュ値
受領者組織ID
受領者アプリID
提供者ファイルリスト
アプリID
XXXX.class
MANIFEST.MF
ハッシュ値
h1
SIG-BD.SF
ハッシュ値
d2
d4
d5
BD.ROOT.CERTIFICATE
ルート証明書
d3
d1
SIG-BD.RSA
ルート証明書
リーフ証明書
組織ID
ルート証明書
リーフ証明書
組織ID
署名情報
第2チェーン
第1チェーン

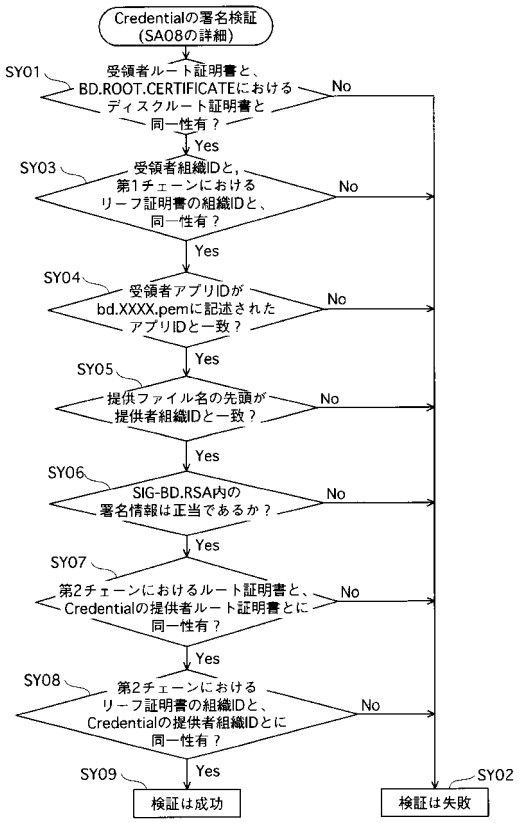
【図６】


プログラム実行装置
3
ヒープ領域
XXXX.classの
インスタンス
ファイルパス Persistent/org1/
app1を用いたファイル読出/
ファイル書込
バーチャル
マシーン
XXXX.class
5
セキュリティマネージャ
ハッシュ管理テーブル
ディスクルート
証明書のハッシュ値
ローカルストレージ
ルート
ファイルパス Persistent/R1/org1/app1を
用いたファイル読出/ファイル書込
4
ハードディスク
Persistent
R1
org1
org2
org3
app1
app2
app3
ドメイン領域
R2
ローカルストレージ
ルート
2
アプリケーションマネージャ
MANIFEST.MF
SIG-BD.SF
SIG-BD.RSA
bd.XXXX.perm
XXXX.class
BD.ROOT
.CERTIFICATE
XXX.JAR
1
BDドライブ
ディスクルート
識別子 R1
BD-ROM

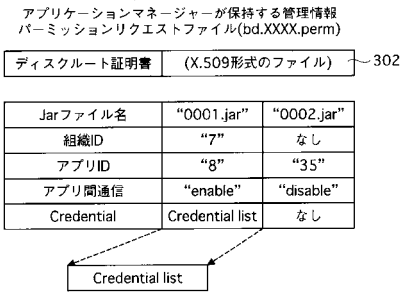
【図７】



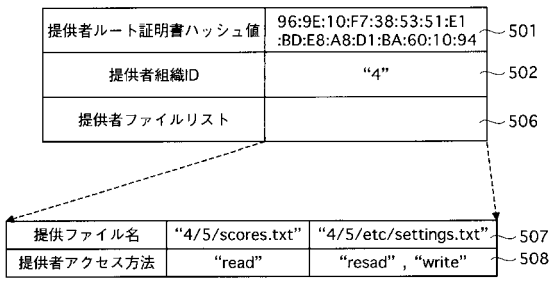
【図８】



【図９】

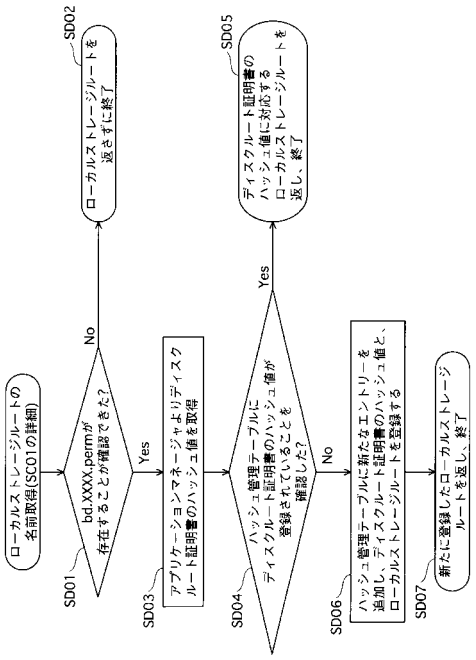


【図１０】



【図１１】

【図１２】

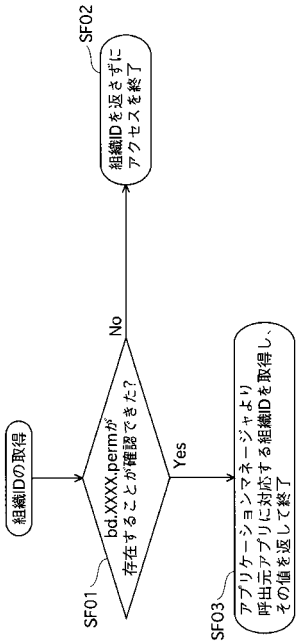


【図１３】

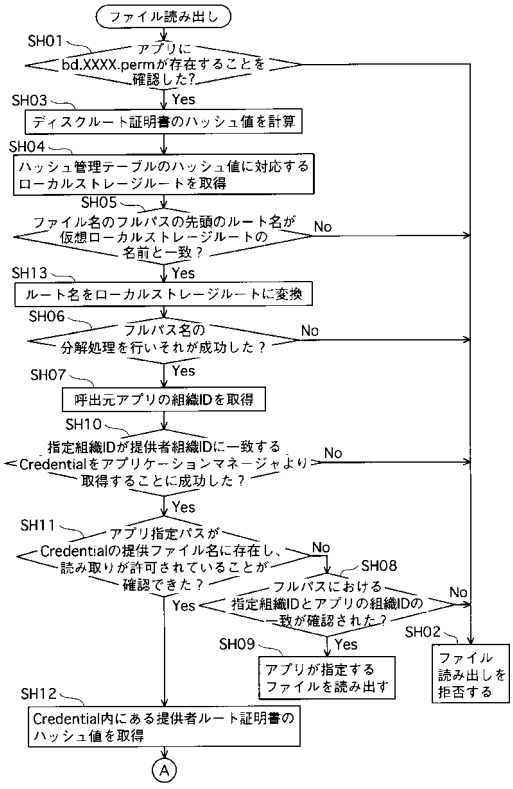
セキュリティマネージャが保持するハッシュ管理テーブル

	1301	1302
提供者ルート証明書ハッシュ値	96:9E:10:F7:38:53:51:E1:BD:E8:A8:D1:BA:60:10:94	42:6C:40:25:2C:4B:04:9D:20:E7:45:97:02:D2:01:08
ローカルストレージルート	/persistent/0001	/persistent/0003

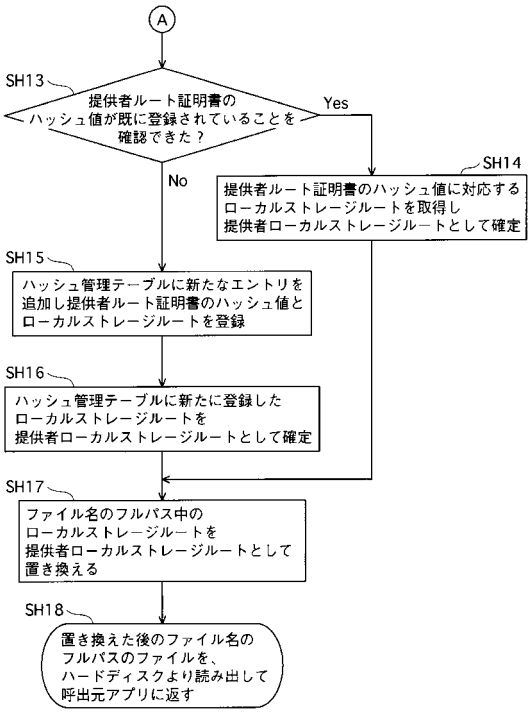
【図１４】



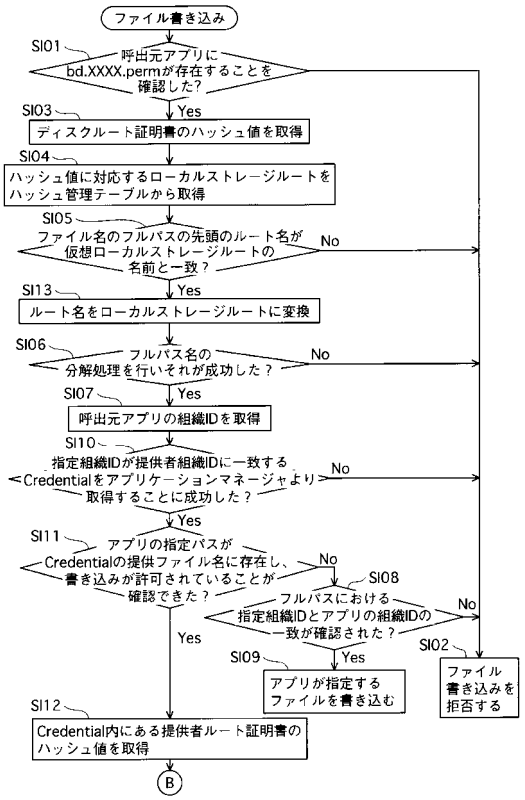
【図１５】



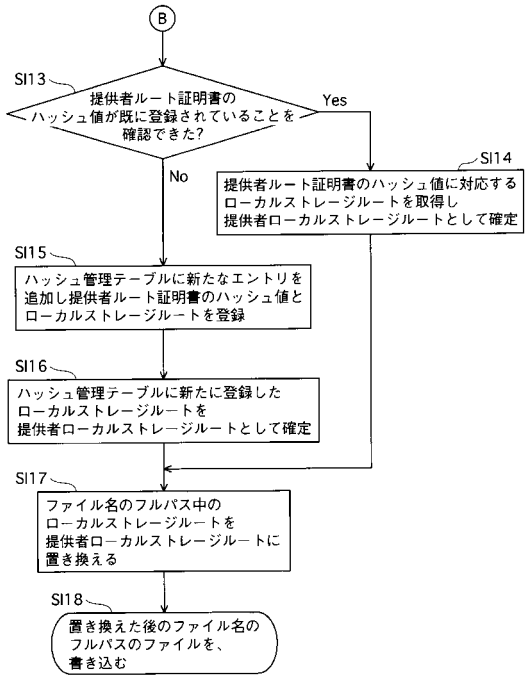
【図１６】


A
SH13
提供者ルート証明書の
ハッシュ値が既に登録されていることを
確認できた？
Yes
No
SH14
提供者ルート証明書のハッシュ値に対応する
ローカルストレージルートを取得し
提供者ローカルストレージルートとして確定
SH15
ハッシュ管理テーブルに新たなエントリを
追加し提供者ルート証明書のハッシュ値と
ローカルストレージルートを登録
SH16
ハッシュ管理テーブルに新たに登録した
ローカルストレージルートを
提供者ローカルストレージルートとして確定
SH17
ファイル名のフルパス中の
ローカルストレージルートを
提供者ローカルストレージルートとして
置き換える
SH18
置き換えた後のファイル名の
フルパスのファイルを、
ハードディスクより読み出して
呼出元アプリに返す

【図１７】


ファイル書き込み
SI01
呼出元アプリに
bd.XXXX.permが存在することを
確認した?
Yes
SI03
ディスクルート証明書のハッシュ値を取得
SI04
ハッシュ値に対応するローカルストレージルートを
ハッシュ管理テーブルから取得
SI05
ファイル名のフルパスの先頭のルート名が
仮想ローカルストレージルートの
名前と一致？
No
Yes
SI13
ルート名をローカルストレージルートに変換
SI06
フルパス名の
分解処理を行いそれが成功した？
No
Yes
SI07
呼出元アプリの組織IDを取得
SI10
指定組織IDが提供者組織IDに一致する
Credentialをアプリケーションマネージャより
取得することに成功した？
No
Yes
SI11
アプリの指定パスが
Credentialの提供ファイル名に存在し、
書き込みが許可されていることが
確認できた？
No
SI08
フルパスにおける
指定組織IDとアプリの組織IDの
一致が確認された？
No
Yes
SI09
Yes
アプリが指定する
ファイルを書き込む
SI02
ファイル
書き込みを
拒否する
SI12
Credential内にある提供者ルート証明書の
ハッシュ値を取得
B

【図１８】


B
SI13
提供者ルート証明書の
ハッシュ値が既に登録されていることを
確認できた?
Yes
No
SI14
提供者ルート証明書のハッシュ値に対応する
ローカルストレージルートを取得し
提供者ローカルストレージルートとして確定
SI15
ハッシュ管理テーブルに新たなエントリを
追加し提供者ルート証明書のハッシュ値と
ローカルストレージルートを登録
SI16
ハッシュ管理テーブルに新たに登録した
ローカルストレージルートを
提供者ローカルストレージルートとして確定
SI17
ファイル名のフルパス中の
ローカルストレージルートを
提供者ローカルストレージルートに
置き換える
SI18
置き換えた後のファイル名の
フルパスのファイルを、
書き込む

フロントページの続き

(51)Int.Cl. FI
H04L 9/00 675B

審査官 後藤 彰

(56)参考文献 特開平１０－８３３１０（ＪＰ，Ａ）
特開２００４－３０２９７３（ＪＰ，Ａ）
特開２００５－５２４９１０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/14
G06F 21/22
G06F 21/24
G09C 1/00
H04L 9/32