

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5020301号
(P5020301)

(45) 発行日 平成24年9月5日(2012.9.5)

(24) 登録日 平成24年6月22日(2012.6.22)

| (51) Int.Cl. | | F I | |
|--------------|--------------|------------------|-----------------------|
| G06F | 21/24 | (2006.01) | G O 6 F 21/24 1 6 7 A |
| G06F | 21/22 | (2006.01) | G O 6 F 21/22 1 1 2 L |
| H04L | 9/32 | (2006.01) | H O 4 L 9/00 6 7 5 A |
| G09C | 1/00 | (2006.01) | G O 9 C 1/00 6 4 0 D |

請求項の数 3 (全 32 頁)

| | | | |
|--------------|-------------------------------------|-----------|------------------|
| (21) 出願番号 | 特願2009-250054 (P2009-250054) | (73) 特許権者 | 000005821 |
| (22) 出願日 | 平成21年10月30日(2009.10.30) | | パナソニック株式会社 |
| (62) 分割の表示 | 特願2007-502669 (P2007-502669) の分割 | | 大阪府門真市大字門真1006番地 |
| 原出願日 | 平成18年2月13日(2006.2.13) | (74) 代理人 | 100090446 |
| (65) 公開番号 | 特開2010-33592 (P2010-33592A) | | 弁理士 中島 司朗 |
| (43) 公開日 | 平成22年2月12日(2010.2.12) | (74) 代理人 | 100125597 |
| 審査請求日 | 平成21年11月25日(2009.11.25) | | 弁理士 小林 国人 |
| (31) 優先権主張番号 | 特願2005-36621 (P2005-36621) | (74) 代理人 | 100146798 |
| (32) 優先日 | 平成17年2月14日(2005.2.14) | | 弁理士 川畑 孝二 |
| (33) 優先権主張国 | 日本国(JP) | (74) 代理人 | 100121027 |
| 前置審査 | | | 弁理士 木村 公一 |
| | | (74) 代理人 | 100175411 |
| | | | 弁理士 土田 幸雄 |
| | | (74) 代理人 | 100174861 |
| | | | 弁理士 中島 安洋 |

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】

第1のルート証明書と、第2のルート証明書と、アプリケーションと前記アプリケーションに対応する組織IDと対象となるファイルを利用する権限を前記アプリケーションに提供する提供者組織の提供者組織IDと前記提供者組織IDと同じ組織IDを含むリーフ証明書と、前記リーフ証明書に関連する第3のルート証明書とが記録された記録媒体から前記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、

前記第2のルート証明書は、ルート認証局から配布されたルート証明書を当該記録媒体に割り当てたものであり、

前記記録媒体を装填するドライブ部と、

前記装填された記録媒体から読み出された前記第2のルート証明書と、前記第1のルート証明書とが同じかどうかを判定することにより前記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、前記装填された記録媒体から読み出された前記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、前記記憶手段内に前記第2の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設けるとともに、当該ドメイン領域内に前記アプリケーションに対応する組織IDを用いて特定される領域を設け、前記アプリケーションに対応する組織IDを用いて特定される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、

前記管理手段が正当であると判定した場合、前記実行中のアプリケーションが前記提供者組織IDを含むファイルパスを指定して、前記記憶領域へのアクセスを要求したとき、前記指定された提供者組織IDと同じ組織IDを含むリーフ証明書に関連する第3のルート証明書のハッシュ値と、前記指定された提供者組織IDとを用いたファイルパスに変換し、前記アプリケーションに前記変換したファイルパスにより特定される記憶領域の対象となるファイルを利用する権限を与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設けることができるように構成された記憶手段を備えたコンピュータにおいて、第1のルート証明書と、第2のルート証明書と、アプリケーションと前記アプリケーションに対応する組織IDと対象となるファイルを利用する権限を前記アプリケーションに提供する提供者組織の提供者組織IDと前記提供者組織IDと同じ組織IDを含むリーフ証明書と、前記リーフ証明書に関連する第3のルート証明書とが記録された記録媒体から前記アプリケーションを読み出して、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であって、前記第2のルート証明書は、ルート認証局から配布されたルート証明書を当該記録媒体に割り当てたものであり、

10

前記コンピュータに装填された記録媒体から読み出された前記第2のルート証明書と、前記第1のルート証明書とが同じかどうかを判定することにより前記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、

20

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、前記コンピュータに装填された記録媒体から読み出されたアプリケーションを実行する実行手段と、

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、前記記憶手段内に前記第2の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設けるとともに、当該ドメイン領域内に前記アプリケーションに対応する組織IDを用いて特定される領域を設け、前記アプリケーションに対応する組織IDを用いて特定される領域を前記アプリケーションに割り当てる第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、前記実行中のアプリケーションが前記提供者組織IDを含むファイルパスを指定して、前記記憶領域へのアクセスを要求したとき、前記指定された提供者組織IDと同じ組織IDを含むリーフ証明書に関連する第3のルート証明書のハッシュ値と、前記指定された提供者組織IDとを用いたファイルパスに変換し、前記アプリケーションに前記変換したファイルパスにより特定される記憶領域の対象となるファイルを利用する権限を与える第3のステップ

30

を前記コンピュータに実行させる管理方法。

【請求項3】

記憶領域であるドメイン領域を設けることができるように構成された記憶手段を備えたコンピュータにおいて、第1のルート証明書と、第2のルート証明書と、アプリケーションと前記アプリケーションに対応する組織IDと対象となるファイルを利用する権限を前記アプリケーションに提供する提供者組織の提供者組織IDと前記提供者組織IDと同じ組織IDを含むリーフ証明書と、前記リーフ証明書に関連する第3のルート証明書とが記録された記録媒体から前記アプリケーションを読み出して、前記コンピュータに実行させる管理プログラムであって、前記第2のルート証明書は、ルート認証局から配布されたルート証明書を当該記録媒体に割り当てたものであり、

40

前記コンピュータに装填された記録媒体から読み出された前記第2のルート証明書と、前記第1のルート証明書とが同じかどうかを判定することにより前記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、前記コンピュータに装填された記録媒体から読み出された前記アプリケーションを実行する実行手段と、

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、前記記憶手段内に前記第2の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設けるとともに、当

50

該ドメイン領域内に前記アプリケーションに対応する組織IDを用いて特定される領域を設け、前記アプリケーションに対応する組織IDを用いて特定される領域を前記アプリケーションに割り当てる第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、前記実行中のアプリケーションが前記提供者組織IDを含むファイルパスを指定して、前記記憶領域へのアクセスを要求したとき、前記指定された提供者組織IDと同じ組織IDを含むリーフ証明書に関連する第3のルート証明書のハッシュ値と、前記指定された提供者組織IDとを用いたファイルパスに変換し、前記アプリケーションに前記変換したファイルパスにより特定される記憶領域の対象となるファイルを利用する権限を与える第3のステップ

を前記コンピュータに実行させる管理プログラム。

10

【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(MultimediaHomePlatform)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。

20

【0003】

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。

【0004】

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。

30

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。

40

【0005】

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。

【先行技術文献】

【非特許文献】

【0006】

【非特許文献1】「JAVA(登録商標) Security」Scott Oaks著、O'Reilly発行、May 2001、ISBN0-596-00157-6

【発明の概要】

50

【発明が解決しようとする課題】

【0007】

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すると、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。

10

【0008】

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。

20

30

【0009】

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。

40

【課題を解決するための手段】

【0010】

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、第1のルート証明書と、第2のルート証明書と、アプリケーションと前記アプリケーションに対応する組織IDと対象となるファイルを利用する権限を前記アプリケーションに提供する提供者組織の提供者組織IDと前記提供者組織IDと同じ組織IDを含むリーフ証明書と、前記リーフ証明書に関連する第3のルート証明書とが記録された記録媒体から前記アプリケーションを読み出して、実行するものであり、前記第2のルート証明書は、ルート認証局から配布されたルート証明書を当該記録媒体に割り当てたものであり、前記記録媒体を装填するドライブ部と、前記装填された記録媒体から読み出された前記第2のルート証明

50

書と、前記第1のルート証明書とが同じかどうかを判定することにより前記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、前記装填された記録媒体から読み出された前記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、前記記憶手段内に前記第2の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設けるとともに、当該ドメイン領域内に前記アプリケーションに対応する組織IDを用いて特定される領域を設け、前記アプリケーションに対応する組織IDを用いて特定される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、前記管理手段が正当であると判定した場合、前記実行中のアプリケーションが前記提供者組織IDを含むファイルパスを指定して、前記記憶領域へのアクセスを要求したとき、前記指定された提供者組織IDと同じ組織IDを含むリーフ証明書に関連する第3のルート証明書のハッシュ値と、前記指定された提供者組織IDとを用いたファイルパスに変換し、前記アプリケーションに前記変換したファイルパスにより特定される記憶領域の対象となるファイルを利用する権限を与えることを特徴としている。

10

【発明の効果】

【0011】

本発明にかかるアプリケーション実行装置は、上述したような技術的事項を有しているので、前記管理手段が正当であると判定した場合、前記記憶手段内に前記第2の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設けられるとともに、当該ドメイン領域の配下に、組織ID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。

20

【0012】

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。

30

【0013】

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。

40

【0014】

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。

【図面の簡単な説明】

【0026】

【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) Jav

50

a (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す図である。(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。

【発明を実施するための形態】

【0027】

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(BurstCutting 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

10

20

30

40

50

護技術などに利用されることがよくある。

【 0 0 2 8 】

「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（UniversalDisk 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。

10

【 0 0 2 9 】

(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、CCITTRecommendationX.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。

20

【 0 0 3 0 】

(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](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。

【 0 0 3 1 】

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。

30

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。

【 0 0 3 2 】

(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。

40

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。

【 0 0 3 3 】

(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

50

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。

【0034】

(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。

【0035】

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。

【0036】

「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。

【0037】

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。

【0038】

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。

【0039】

(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。

10

【0040】

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。

【0041】

20

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。

【0042】

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。

30

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。

【0043】

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。

40

【0044】

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

50

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。

【 0 0 4 5 】

BD-ROMではなく、Java（登録商標）アーカイブファイル 3 0 2 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 3 0 1 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 3 0 2 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 3 0 1 を用いることにより、Java（登録商標）アーカイブファイル 3 0 2 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。

10

【 0 0 4 6 】

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 3 0 1 ）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。

20

【 0 0 4 7 】

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 3 0 2 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。

【 0 0 4 8 】

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 3 0 1 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）。

30

40

【 0 0 4 9 】

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）。

【 0 0 5 0 】

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

50

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。

【0051】

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。

10

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)PersonalBasis Profile(PBP 1.0)と、Globally Executable MHPspecification(GEM1.0.2)forpackage 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。

【0052】

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。

20

（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。

【0053】

（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(MultimedaiHomePlatform)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。

30

【0054】

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。

40

【0055】

（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、

50

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いければ、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、パーチャルマシーン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、パーチャルマシーン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。

【0056】

（ハードディスク4）

ハードディスク4は、Java（登録商標）10 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。

【0057】

（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。

【0058】

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

10

20

30

40

50

)。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）。

【0059】

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）。

【0060】

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）。

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）。

【0061】

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。

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）。

【0062】

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.XXX X.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の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）。

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。

【0063】

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）。

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）。

【0064】

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）。

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）。

【0065】

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

10

20

30

40

50

(SY07)。一致しない場合、Credentialが不正であり、検証が失敗する(SY02)。

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)。

【0066】

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。

10

【0067】

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。

【0068】

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、パーチャルマシーン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。パーチャルマシーン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。

20

【0069】

Java(登録商標)アプリケーションはパーチャルマシーン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。

【0070】

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。

30

【0071】

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を組み合わせて、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。

40

【0072】

Java(登録商標)アプリケーションが図11のとおり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

50

(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。

【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(SD01)。bd.XXXX.permがあるかどうかはパーチャルマシン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)。

10

【0074】

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)。

【0075】

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)。

20

【0076】

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。

30

【0077】

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)。

40

【0078】

「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。

50

【 0 0 7 9 】

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 1 5 と図 1 6 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 (SH 0 1)。bd.XXXX.permがない場合、「ファイル読み出し」を拒否する (SH 0 2)。

【 0 0 8 0 】

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 3 0 1 のハッシュ値を取得する (SH 0 3)。

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 3 0 1 のハッシュ値に対応するローカルストレージルート 1 3 0 2 を取得する (SH 0 4)。

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)。

【 0 0 8 1 】

ここで、変換前後において、フルパスは以下ようになる。

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 (SH 0 6)。ファイル名のフルパスは「ローカルストレージルート 1 3 0 2 + "/" + 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 1 3 0 2 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 (SH 0 2)。

【 0 0 8 2 】

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 (SH 0 7)。そして、指定組織IDが提供者組織ID 5 0 2 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 (SH 1 0)。Credentialを取得できなかった場合、アクセスを拒否する (SH 0 2)。

【 0 0 8 3 】

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 5 0 7 に存在し、提供アクセス方法 5 0 8 にて読み出しが許可されているかどうかを確認する (SH 1 1)。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 5 0 1 を取得する (SH 1 2)。

【 0 0 8 4 】

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 (SH 0 8)。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 (SH 0 9)。一致していない場合、読み出しを拒否する (SH 0 2)。

【 0 0 8 5 】

10

20

30

40

50

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 5 0 1 が既に登録されているかどうかを確認する (SH 1 3)。登録されている場合、提供者ルート証明書のハッシュ値 5 0 1 に対応するローカルストレージルート 1 3 0 2 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 (SH 1 4)。

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 5 0 1 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 (SH 1 5)。そのエントリーには提供者ルート証明書のハッシュ値 5 0 1 と、テーブル内で一意となるローカルストレージルート 1 3 0 2 が記載される。

【 0 0 8 6 】

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 1 3 0 2 を提供者ローカルストレージルートとして確定する (SH 1 6)。

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 1 3 0 2 を提供者ローカルストレージルートに置き換える (SH 1 7)。

【 0 0 8 7 】

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 (SH 1 8)。

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の説明する。

「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 1 7 と図 1 8 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。

【 0 0 8 8 】

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 (SI 0 1)。bd.XXXX.perm がない場合、「ファイル書き込み」を拒否する (SI 0 2)。

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 3 0 1 のハッシュ値を取得する (SI 0 3)。

【 0 0 8 9 】

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 3 0 1 のハッシュ値に対応するローカルストレージルート 1 3 0 2 を取得する (SI 0 4)。

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)。

【 0 0 9 0 】

ここで、変換前後において、フルパスは以下ようになる。

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 (SI 0 6)。ファイル名のフルパスは「ローカルストレージルート 1 3 0 2 + "/" + 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 1 3 0 2 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 (SI 0 2)。

【 0 0 9 1 】

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 (SI 0 7)。そして、指定組織IDが提供者組織ID 5 0 2 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 (SI 1 0)。存在しない場合、書き込みを拒否する (SI 0 2)。

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 5 0 7 として存在し、提供アクセス方法 5 0 8 にて書き込みが許可されているかどうかを確認する (SI 1 1)。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 (SI 0 8)。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 (SI 0 9)。一致しない場合、書き込みを拒否する (SI 0 2)。

10

【 0 0 9 2 】

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 5 0 1 を取得する (SI 1 2)。

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 5 0 1 が既に登録されているかどうかを確認する (SI 1 3)。登録されている場合、提供者ルート証明書のハッシュ値 5 0 1 に対応するローカルストレージルート 1 3 0 2 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 (SI 1 4)。

20

【 0 0 9 3 】

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 5 0 1 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 (SI 1 5)。その行には提供者ルート証明書のハッシュ値 5 0 1 と、テーブル内で一意となるローカルストレージルート 1 3 0 2 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 1 3 0 2 を提供者ローカルストレージルートとして確定する (SI 1 6)。

【 0 0 9 4 】

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 1 3 0 2 を提供者ローカルストレージルートに置き換える (SI 1 7)。

30

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 (SI 1 8)。

【 0 0 9 5 】

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、第三者機関による管理は不要になる。

40

【 0 0 9 6 】

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。

【 0 0 9 7 】

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

50

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。

【0098】

(「Java(登録商標)アーカイブファイル302の選択)

10

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。

【0099】

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。

(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。

20

【0100】

(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。

(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。

30

【0101】

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。

【0102】

(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定めるようにしても良い。

40

(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。

【0103】

(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

50

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。

【0104】

(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。

【0105】

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としているが、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よい。

【0106】

(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よい。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。

【0107】

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で作られた攻撃用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よう。

【0108】

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)

10

20

30

40

50

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。

【0109】

（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。

10

【0110】

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。

20

【0111】

（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。

30

【0112】

（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り出して、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。

40

【0113】

・「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Credential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。

50

【 0 1 1 4 】

・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、装置組み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。

【 0 1 1 5 】

(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。

【 0 1 1 6 】

・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。

【 0 1 1 7 】

・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。

【 0 1 1 8 】

・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Local Storage上のコンテンツ(off-discコンテンツ)とをバインドする機構(BindingScheme)を提供する。

【 0 1 1 9 】

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exists()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exists()メソッドの用例を以下に示す。

【 0 1 2 0 】

例：

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

10

20

30

40

50

(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。

【0121】

- ・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BD MV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。

10

【0122】

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。

【0123】

(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。

20

【0124】

(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Virtual Package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。

30

【0125】

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。Virtual Package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Virtual Package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。

40

【0126】

50

(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、プロセッサのネイティブコード、JAVA(登録商標)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。

【0127】

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。まず初めに、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。

【0128】

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。

【0129】

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クを起動する。リンク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。

【0130】

・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。

(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。

【0131】

(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。

【0132】

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。

【0133】

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。

【0134】

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注目するとシステム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。

【0135】

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。

【0136】

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。

【0137】

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(System in 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

10

20

30

40

50

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。

【0138】

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 (FieldProgrammableGate 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

10

【産業上の利用可能性】

【0139】

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。

20

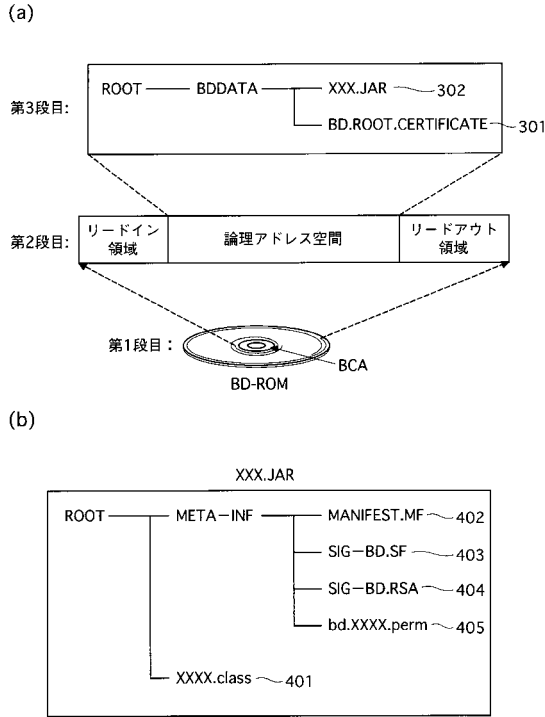
【符号の説明】

【0140】

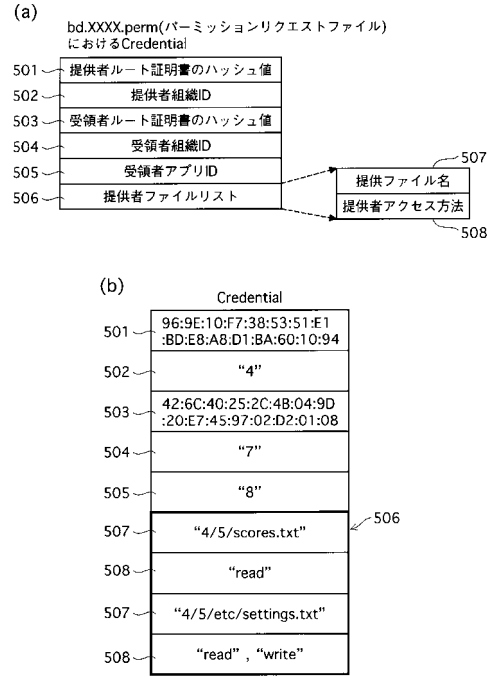
- 1 BDドライブ
- 2 アプリケーションマネージャ
- 3 バーチャルマシーン
- 4 ハードディスク
- 5 セキュリティマネージャ
- 301 ディスクルート証明書
- 302 Java (登録商標) アーカイブファイル
- 401 クラスファイル
- 402 マニフェストファイル
- 403 シグネチャファイル
- 404 デジタルシグネチャファイル
- 405 パーミッションリクエストファイル
- 501 提供者ルート証明書のハッシュ値
- 502 提供者組織ID
- 503 受領者ルート証明書のハッシュ値
- 504 受領者組織ID
- 505 受領者アプリID
- 506 ファイルリスト

30

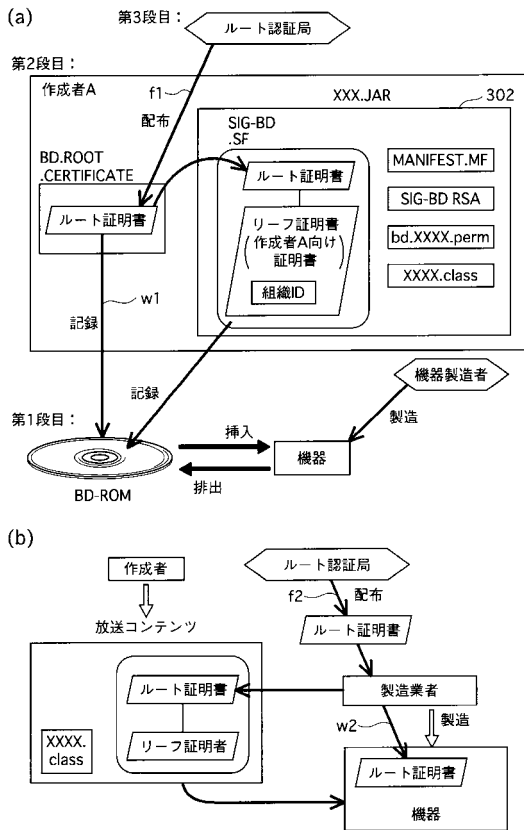
【図1】



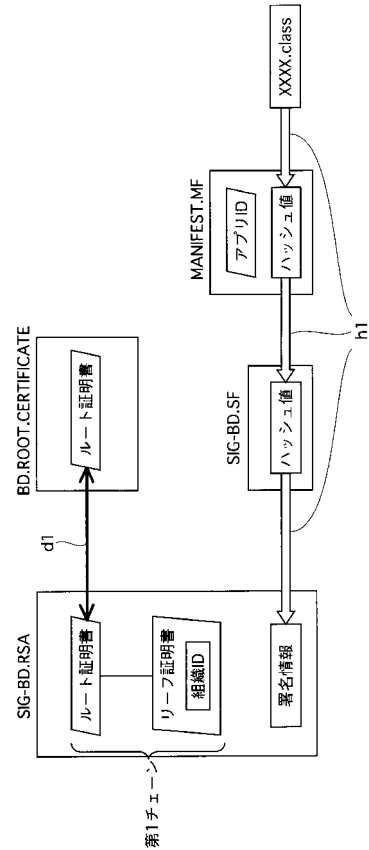
【図2】



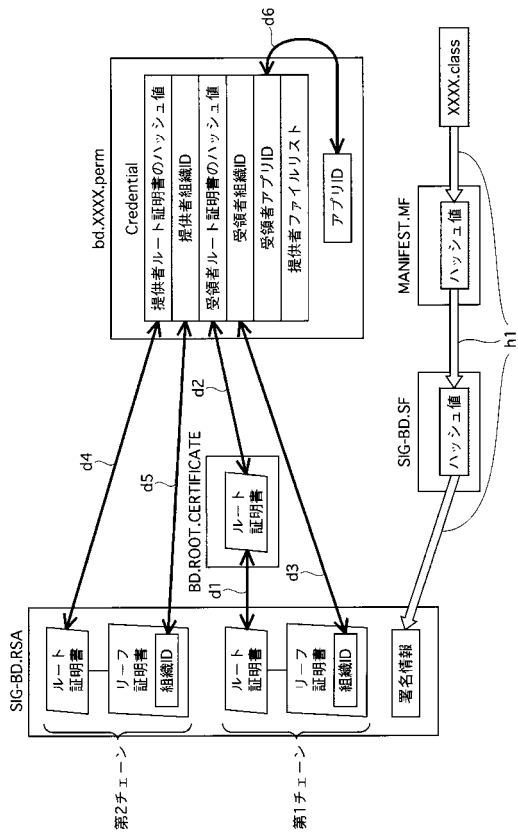
【図3】



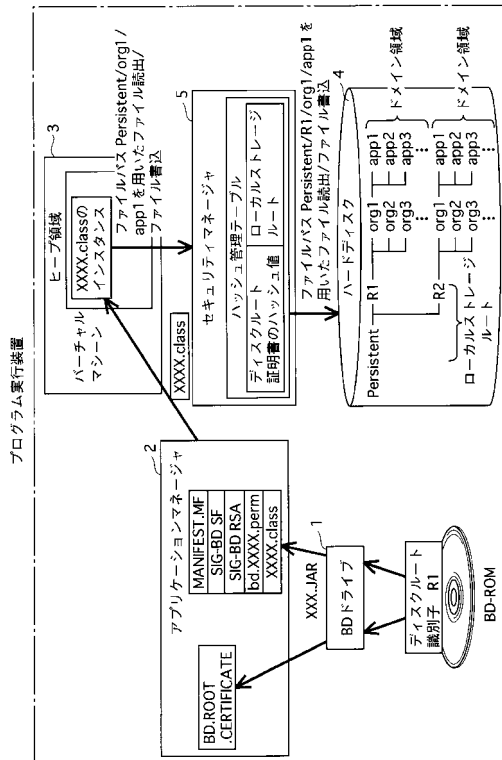
【図4】



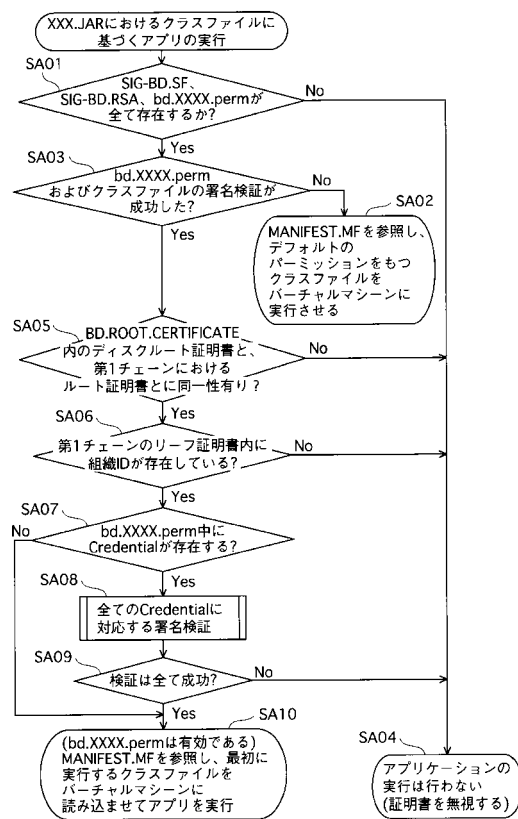
【図5】



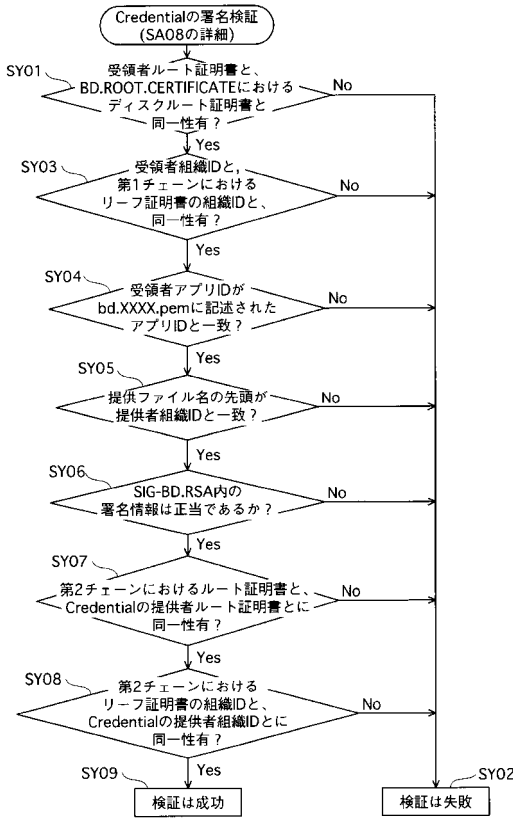
【図6】



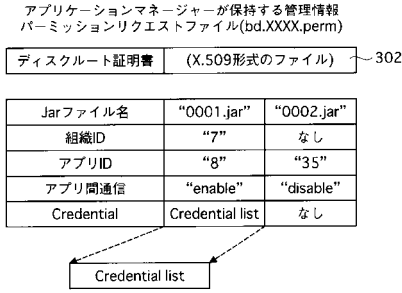
【図7】



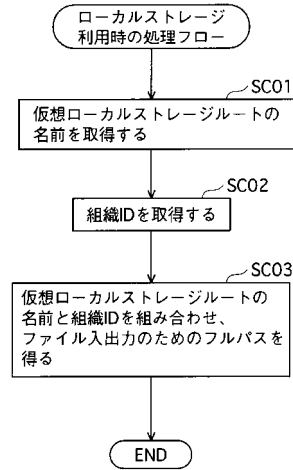
【図8】



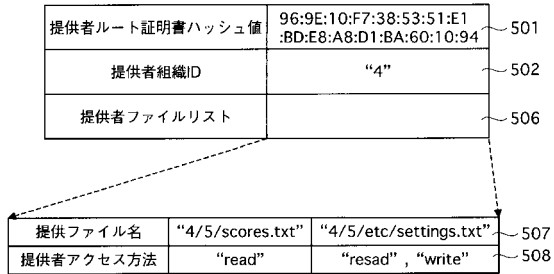
【図9】



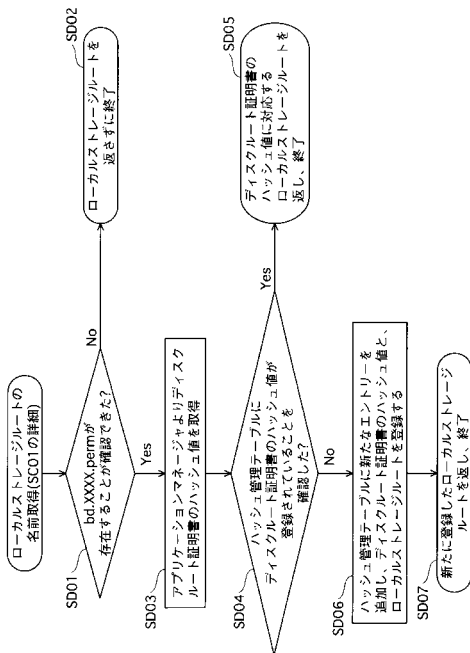
【図11】



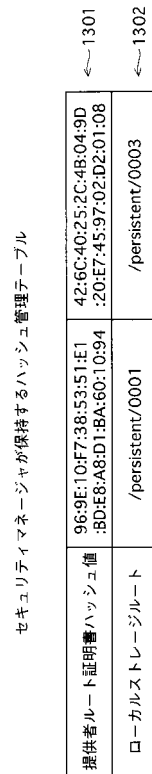
【図10】



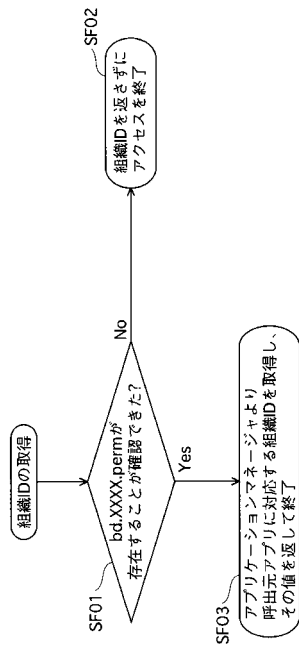
【図12】



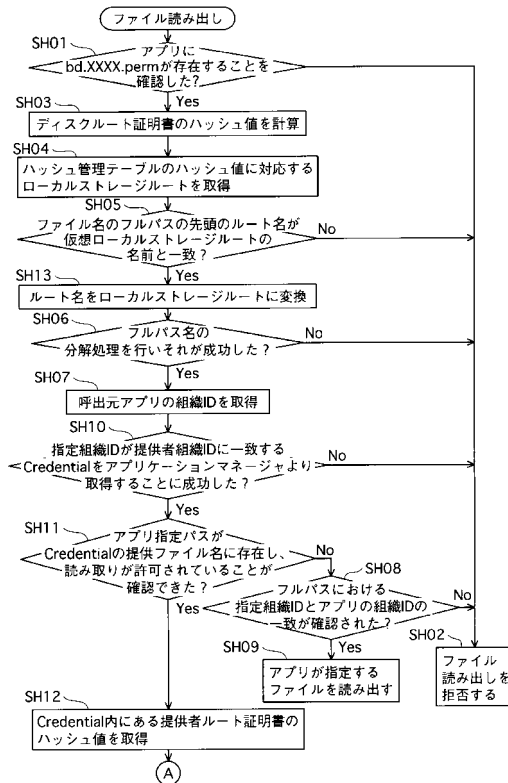
【図13】



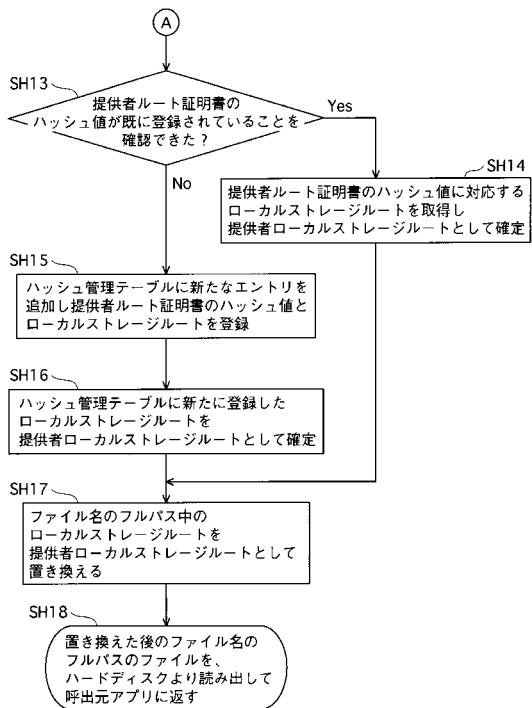
【図14】



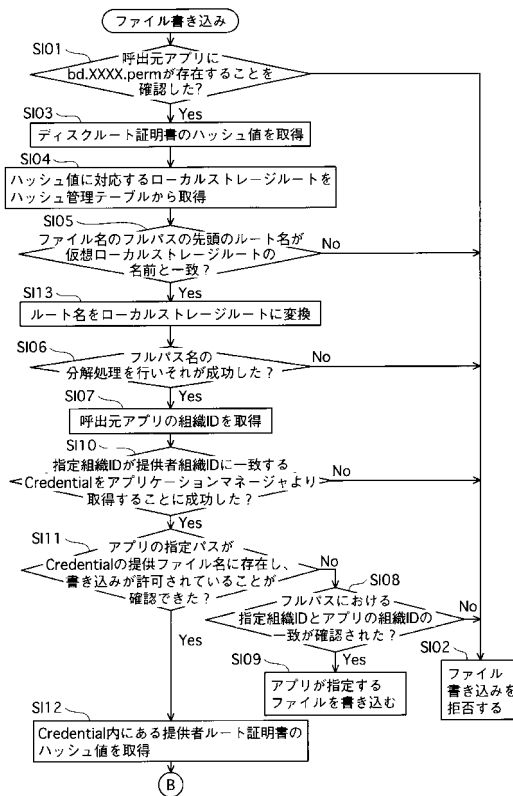
【図15】



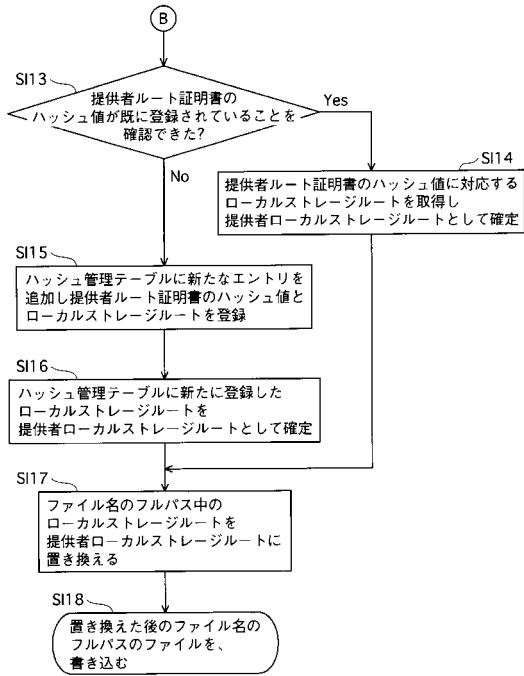
【図16】



【図17】



【図18】



フロントページの続き

(74)代理人 100148194

弁理士 小林 義周

(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 パナソニック株式会社内

(72)発明者 金丸 智一

大阪府門真市大字門真1006番地 パナソニック株式会社内

審査官 和田 財太

(56)参考文献 特開平10-083310(JP, A)

国際公開第2003/096238(WO, A1)

米国特許第06584495(US, B1)

(58)調査した分野(Int.Cl., DB名)

G06F 21/22 - 21/24

G09C 1/00

H04L 9/32