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

特許第3699649号
(P3699649)

(45)発行日　平成17年9月28日(2005.9.28)　(24)登録日　平成17年7月15日(2005.7.15)

(51)Int.Cl.[7]	F I		
G06F 12/14	G06F	12/14	520A
G06F 12/00	G06F	12/00	537A
G06K 19/073	G06K	19/00	P
G06K 19/10	G06K	19/00	R

請求項の数 5　(全 15 頁)

(21)出願番号　特願2000-529650 (P2000-529650)
(86)(22)出願日　平成11年1月20日 (1999.1.20)
(65)公表番号　特表2002-502067 (P2002-502067A)
(43)公表日　平成14年1月22日 (2002.1.22)
(86)国際出願番号　PCT/FR1999/000096
(87)国際公開番号　WO1999/039257
(87)国際公開日　平成11年8月5日 (1999.8.5)
審査請求日　平成13年1月5日 (2001.1.5)
(31)優先権主張番号　98/01008
(32)優先日　平成10年1月29日 (1998.1.29)
(33)優先権主張国　フランス(FR)

(73)特許権者　500342363
ジェムプリュス
ＧＥＭＰＬＵＳ
フランス共和国，エフ－１３８８１　ジェムノ　セデックス，パルク　ダクティヴィテ　ドゥ　ジェムノ，アヴニュ　デュ　ピック　ドゥ　ベルターニュ
(74)代理人　100080447
弁理士　太田　恵一
(72)発明者　クリエ，シャルル
フランス共和国，エフ－１３３６０　ロクベール，ル　カネ，アヴニュ　フレデリック　ミストラル，１９

最終頁に続く

(54)【発明の名称】情報処理アプリケーション・セキュリティ管理システム及び方法

(57)【特許請求の範囲】

【請求項１】

情報処理アプリケーション・セキュリティの管理システムにおける、情報処理アプリケーション・セキュリティの管理方法であって、

該情報処理アプリケーション・セキュリティの管理システムが、以下の（ア）から（オ）の特徴を有し、

（ア）該管理システムが、ｎ階層の根付き木構造として構成され、その階層１が最も高い階層であるようなディレクトリ・ファイルを有する

（イ）該ディレクトリ・ファイルの各階層毎に情報処理アプリケーションが記録される

（ウ）該管理システムが、ｒ個のセキュリティ・レジスタ（Ｒ）を有する

（エ）各セキュリティ・レジスタは、ただ一つのディレクトリに割り当てることが可能であるように適合されている

（オ）セキュリティ・レジスタには、一つのディレクトリのもとで付与された権限を表す情報（権限情報）または秘密情報Ｓ１からＳｐのセットが記録される

さらに、該管理方法が、以下の（ａ）（ｂ）（ｃ）から成る過程を含むことを特徴とする、情報処理アプリケーション・セキュリティの管理方法。

（ａ）管理システムが、後述の規則ＲＧ１，ＲＧ２，ＲＧ３に従って、ディレクトリ（Ｒｅｐ）に対して付与された権限情報（Ｓ１からＳｐ）をセキュリティ・レジスタ（Ｒ）内に記憶し、

（ｂ）管理システムが、根付き木構造の中で、提示された秘密情報を検索し、そして

（ｃ）管理システムが、情報処理アプリケーションの階層で一つまたは複数の権限の有無を検証する。
（ＲＧ１）カレント・ディレクトリのもとで一つの権限が付与されると直ちに、セキュリティ・レジスタをこのディレクトリに割り当てる。ある権限が既にこのディレクトリのもとで付与されている場合には、前記セキュリティ・レジスタの更新を行う。
（ＲＧ２）カレント・ディレクトリが変更される際、変更前にカレント・ディレクトリであったディレクトリをセキュリティ・レジスタに接続するリンクを失わせるが、但し、変更後のカレント・ディレクトリが変更前のカレント・ディレクトリの子にあたる場合は除く。
（ＲＧ３）セキュリティ・レジスタがすべて割り当てられている場合は、最も古く割り当てられたセキュリティ・レジスタを新しいカレント・ディレクトリに割当てる。
【請求項２】
過程（ｂ）が、以下からなる規則を適用することを特徴とする、請求項１に記載の方法。
（ＲＧ４）提示された秘密情報（Ｓ）がカレント・ディレクトリ（Ｎｉ）内、もしくはカレント・ディレクトリよりも上位階層のディレクトリ内に記録されていることを検証する。
【請求項３】
過程（ｂ）が、以下からなる中間過程を含むことを特徴とする、請求項１または２に記載の方法。
（ｂ１）管理システムが、階層（Ｎｉ）のカレント・ディレクトリ内で、ある秘密情報を検索し、アプリケーション内に秘密情報（Ｓ）が存在することを検証し、
（ｂ２）この秘密情報（Ｓ）が存在する場合には、管理システムが、秘密情報の提示が成功していることを検証し、
・提示が成功していれば、管理システムによって、秘密情報（Ｓ）に関連する権限がカレント・アプリケーションの階層（Ｎｉ）で付与され、
・提示が失敗すれば、管理システムによって秘密情報（Ｓ）に関連する権限は付与されずに提示された秘密情報の検索の試みが終了し、
（ｂ３）階層（Ｎｉ）のカレント・アプリケーション内にこの秘密情報（Ｓ）が存在しない場合には、管理システムが、階層Ｎ（ｉ－１）の親のアプリケーション内にこの秘密情報（Ｓ）が存在するかどうかを検索し、
（ｂ４）階層Ｎ（ｉ－１）の親のアプリケーション内にこの秘密情報（Ｓ）が存在する場合には、管理システムが、提示が成功していることを検証し、
・提示が成功していれば、管理システムによって、秘密情報（Ｓ）に関連する権限が階層（Ｎｉ）内のカレント・アプリケーションで付与され、
・提示が失敗すれば、管理システムによって秘密情報（Ｓ）に関連する権限は付与されずに提示された秘密情報の検索の試みが終了し、
（ｂ５）階層Ｎ（ｉ－１）の親のアプリケーション内に秘密情報（Ｓ）が存在しない場合には、管理システムが、階層軸に沿って階層Ｎ（ｉ－２）のアプリケーションの階層で秘密情報（Ｓ）の存在を検索して、提示が成功していることを検証し、
以下同様に、秘密情報（Ｓ）の存在が発見されない限り、最上位の階層の層まで続行し、
（ｂ６）秘密情報（Ｓ）が発見されなかった場合には、提示された秘密情報の検索の試みが終了する。
【請求項４】
過程（ｃ）が、以下からなる規則を適用することを特徴とする、請求項１から３のいずれか一つに記載の方法。
（ＲＧ５）根付き木構造をカレント・アプリケーションからルート・アプリケーションに向かって階層軸に沿って経て、カレント・アプリケーションと秘密情報（Ｓ）を含むアプリケーションとを境界とする、根付き木構造の区間に属するアプリケーションのうちの

少なくとも一つにおいて第一の秘密情報（Ｓ）が正しく提示されている場合には、管理システムが、秘密情報（Ｓ）を知っていることを必要とする機能を許可し、しかも、その場合にのみ許可する。

【請求項５】

過程（ｃ）が、以下からなる過程を含むことを特徴とする、請求項１から４のいずれか一つに記載の方法。

（ｃ１）管理システムが、セキュリティ・レジスタが階層Ｎｉのカレント・アプリケーションに関連づけられていることを検証し、

（ｃ２）階層Ｎｉのカレント・アプリケーションに関連づけられているセキュリティ・レジスタがあり、かつ、該セキュリティ・レジスタが必要な権限情報を含んでいる場合には、管理システムが、機能を許可して検証を終了し、

（ｃ３）どのセキュリティ・レジスタもカレント・アプリケーションに関連づけられていない場合、または関連づけられたレジスタが必要な権限情報を含んでいない場合には、管理システムが、階層Ｎｉのカレント・アプリケーション内で参照記号Ｓの秘密情報の存在を検索し、

（ｃ４）秘密情報がカレント・アプリケーション内に存在する場合には、管理システムが機能を拒絶して検証を終了し、

（ｃ５）階層Ｎｉのカレント・アプリケーション内に参照記号Ｓの秘密情報が存在しない場合には、管理システムが、カレント・アプリケーションの階層Ｎ（ｉ－１）の親のアプリケーションにセキュリティ・レジスタが関連付けられていることを検証し、

（ｃ６）親のアプリケーションに関連づけられたセキュリティ・レジスタが、機能を使うのに必要な権限情報を含んでいる場合には、管理システムが、機能を許可して検証を終了し、

（ｃ７）どのセキュリティ・レジスタも親のアプリケーションに関連づけられていない場合、または関連づけられたセキュリティ・レジスタが必要な権限情報を含んでいない場合には、管理システムが、カレント・アプリケーションの階層Ｎ（ｉ－１）の親のアプリケーション内で参照記号Ｓの秘密情報の存在を検索し、

（ｃ８）参照記号Ｓの秘密情報が、階層Ｎ（ｉ－１）の親のアプリケーション内に存在する場合には、管理システムが機能を拒絶して検証を終了し、

（ｃ９）階層Ｎ（ｉ－１）の「親」のアプリケーション内に参照記号Ｓの秘密情報が存在しない場合には、管理システムが、カレント・アプリケーションからルート・アプリケーションに向かって階層軸に沿って階層Ｎ（ｉ－２）の「親の親」のアプリケーションにセキュリティ・レジスタが関連付けられていることを検証し、

以下同様に、参照記号Ｓの秘密情報の存在が発見されない限り、続行し、

（ｃ１０）秘密情報が発見されなかった場合は、管理システムが機能を拒絶して検証を終了する。

【発明の詳細な説明】

【０００１】

本発明は情報処理システムに関するものであり、そのようなシステムの中でも特に、これらの情報処理システムによって使用される可能性のある様々なアプリケーションへのアクセス条件を管理する為のシステムと方法に関するものである。本発明は、保健衛生、銀行、輸送、移動電話等、用途は問わず、チップ・カードのマイクロプロセッサによって使用されることを優先的に目的としているが、それに限られるというわけではない。

【０００２】

周知のセキュリティ管理方法の主な欠点は、次の通りである。

－第一の欠点は、一つのアプリケーションを選択するのに階層構造を示さなければならず、つまり、「親の親」のアプリケーションから始めて、次に「親」のアプリケーション、そして「子」のアプリケーションというように選択肢を辿って行かねばならないということであり、ハード・ディスクのディレクトリの中でファイルを選択する為にそれと同様の選択の道筋を辿らなければならないことである。しかも、セキュリティの問題について

は何も準備されていない。
従って、選択の階層とセキュリティの階層の間に関係が存在しない。
【０００３】
－第二の欠点は、セキュリティの階層の数またはアプリケーションの数が制限されることである。事実、各アプリケーションに一つのセキュリティ・レジスタが専用となっていて、該セキュリティ・レジスタは、パスワード等の秘密情報を知っていることによりこのアプリケーションによって付与された権限情報を記憶する。ｎ個の階層を更に付け加える為、つまり、一連のマルチ・アプリケーションを使用する為には、例えば、一つのセキュリティ・レジスタを各アプリケーションに関連づけなければならず、それは結局のところ、それらのセキュリティ・レジスタが保存されている高速メモリの相当な部分を使ってしまうことになる。この高速メモリの容量は限られているので、そこに多くのセキュリティ・レジスタを保存しておくのは望ましくない。従って、幾つかのシステムでは、階層の層の数またはアプリケーションの数が三つ、つまり、三個のセキュリティ・レジスタに制限されている。
－第三の欠点は、アプリケーションの単純な「解放」を妨げること、つまり、「子」のアプリケーションをその「親」のアプリケーションから独立させることができないことである。事実、新しいアプリケーションを造り上げる際には、「親」のアプリケーションの権限情報及び秘密情報を使うことが不可欠であり、「子」のアプリケーションに固有の秘密情報を造り上げるまで、該権限情報及び秘密情報だけが使用可能となる。
【０００４】
本発明の目的は、以上に述べられたような欠点のない情報処理アプリケーション・セキュリティ管理方法を実施することであり、従って、次のようなことを可能とする。
－階層の層の数またはアプリケーションの数が制限されず、
－セキュリティの観点から、「親」のアプリケーションを通ることなく、「子」のアプリケーションを「親」のアプリケーションから独立させる。
【０００５】
従って本発明は情報処理アプリケーション・セキュリティ管理システムに関するものであり、該システムは次の特徴を有する。
－ｎ階層の根付き木構造として構成されたディレクトリ・ファイル内に情報処理アプリケーションが記録され、階層１のディレクトリが最も高い階層にあり、
－ｒ個のセキュリティ・レジスタは、それぞれただ一つのディレクトリに割り当てることが可能であり、各セキュリティ・レジスタは一つのディレクトリのもとで付与された権限情報または秘密情報Ｓ１からＳｐのセットを含んでいる。
【０００６】
本発明はまた、上記の管理システムにおける情報処理アプリケーション・セキュリティ管理の方法に関するものであって、該方法は、以下から成る過程を含むことを特徴とする。
（ａ）所定の規則に従って、ディレクトリのもとで付与された権限情報をセキュリティ・レジスタ内に記憶し、
（ｂ）根付き木構造の中で、提示された秘密情報を検索し、そして
（ｃ）アクセス条件を満たすために、情報処理アプリケーションの階層で一つ（または複数）の権限に相当する識別内容を検証する。
【０００７】
本発明のその他の特徴と長所は特徴的な実施例に従って説明を読むことによって明らかとなるが、前記説明は添付された次の書類との関連でなされている。
－図１はディレクトリの根付き木構造の一例であり、
－図２．１から２．１４は、一つのディレクトリに対する一つのセキュリティ・レジスタの割当てまたは離反の三つの規則の適用例を示しており、
－図３．１ａから３．６ａ、及び３．１ｂから３．６ｂは秘密情報の提示規則の適用例を示しており、

－図４．１から４．６は必要な権限の付与を検証する規則の適用例を示している。

【０００８】

これから本発明をチップ・カードへ、さらに厳密には、チップ・カード内で使用されるマイクロプロセッサへのその適用に則して記述していく。しかしながら、本発明は、システムによって提供される特定のサービスまたは機能が特定のユーザーまたはオペレーターにのみアクセス可能であることが必要とされるか、または、単にそれが望ましいような情報処理システム全てに適用が可能である。

【０００９】

例えば銀行カードや移動電話のカードのようなチップ・カードの場合、ユーザーが使えるサービスや機能は、申し込まれた使用契約のタイプによっては許可を受けることが可能であり、それらのような許可（または権限）は、そのサービスやその機能の実施に必要なファイルにアクセスできるような秘密情報を知っているということを証明することによって、付与される。

【００１０】

以下の説明においては、下記の規定が採り入れられる。

－ファイルとはアクセス条件によって保護可能なデータ・セットである。

－ディレクトリＲｅｐは、根付き木構造（図１）として構成されたファイル・セットおよび／またはディレクトリ・セットであり、通常は一つのディレクトリは一つのアプリケーション専用となっている。

－一つのファイルまたは一つのディレクトリＲｅｐへのアクセス条件は、ファイルまたはディレクトリにおいて何らかの機能を実行可能とする為の、秘密コードまたは外的認証の提示のような、満たすべき判定条件を規定している。

－ファイルおよびディレクトリは幾つもの階層のある根付き木構造として構成されており、該階層の最上位の階層（階層１）のディレクトリは「ルート・ディレクトリ」または根付き木構造の根と呼ばれている。一つの階層は、複数のディレクトリが階層上で同位である特徴となっている。ディレクトリを使うことで一枚のチップ・カードのデータを構造化することが可能になる。図１では、ディレクトリＲｅｐ１，Ｒｅｐ２，Ｒｅｐ３１，Ｒｅｐ３２，Ｒｅｐ４１，Ｒｅｐ４２，Ｒｅｐ５１そしてＲｅｐ５２のみが提示されており、それぞれが一つまたは複数のファイルを含むことが可能である。ディレクトリＲｅｐ１はｎ＝５のディレクトリ階層を有する根付き木構造の根であり、ディレクトリ４１および４２は階層ｉ＝４に属している。

－一つのセキュリティ・レジスタＲは一つのディレクトリのもとで付与された権限情報全体を含んでおり、一つの権限情報は名前、番号識別子などの参照によって識別されるある秘密情報を知っていることを証明するものである。ある秘密情報を知っていることを証明するにはいくつもの方法があり、例えば端末とチップ・カードの間で秘密情報の値を交換したり、またはこの秘密情報を利用して計算されたデータを交換してもよい。その作業、即ちある秘密情報を知っていることを証明する作業は秘密情報の提示と言われる。

【００１１】

一般的には、一枚のチップ・カード上にセキュリティを設定することは、チップ・カードのサービスや機能の使用を、一つまたは複数の秘密情報を知っているという証明に従属させることが可能であるということである。従って、カードのある機能を使用可能とする為には、次の条件が必要となる。

－チップ・カードが、セキュリティ・レジスタ内に一つまたは複数の秘密情報を知っているということを証明する情報を前もって記憶し、

－チップ・カードの所持者または端末が、機能を保護する一つ（または複数）の秘密情報を知っているということを証明し、

－機能を使用するに際して、一つ（または複数）の秘密情報がよく知られたものであることを、カードが検証する。

【００１２】

本発明の方法は、以下のことから成る過程からなる。

（ａ）一つのディレクトリに対する一つのセキュリティ・レジスタの割当てまたは離反の規則に従って、一つまたは複数の秘密情報をチップ・カードの所有者または端末が知っているという証明、即ち付与された権限情報をチップ・カードに記憶し、
（ｂ）根付き木構造の中で、提示された一つまたは複数の秘密情報を検索し、
（ｃ）アクセス条件を満たすための一つまたは複数の秘密情報を知っているかどうかを検証する。

【００１３】

チップ・カードに秘密情報を知っているということを記憶させる（過程（ａ））為に必要なのは、秘密情報を正確に提示することであり、それは結局のところ、例えば端末やカードの所持者などの外部の者が前記秘密情報を知っているという証明を示すことになるのであって、この知っているということによってカードで提供される機能やサービスを使う権限が付与されることになる。その権限情報こそが一つのアプリケーションごとに一つのレジスタという割合でセキュリティ・レジスタ内に記憶されるものである。

【００１４】

一つのセキュリティ・レジスタＲはｐ個の数字または位置を含んでおり、各位置は付与された一つの権限に相当する一つの秘密情報を知っているということに割り当てられている。ｐ＝８の位置を持つレジスタは八つの付与された権限に相当するＳ１からＳ８の八つの秘密情報を知っているということを記録することが可能である。

【００１５】

セキュリティ・レジスタＲの数ｒは任意であり、後述する例ではｒ＝３となっている。セキュリティ・レジスタは先行技術において付与されているような一つの階層またはディレクトリ専用となっておらず、一つのディレクトリと一つのセキュリティ・レジスタとの間のリンクは動的なものであり、つまり、このリンクは本発明による方法の規則に従って繋がったり切れたりできるものなのである。

【００１６】

一つの権限情報を一つのセキュリティ・レジスタに記憶させる為には、まず以下の三つの規則ＲＧ１からＲＧ３に従って、一つのディレクトリに対して一つのセキュリティ・レジスタを割り当てまたは離反させなければならない。

【００１７】

規則ＲＧ１

カレント・ディレクトリのもとで、例えば一つの秘密コードや認証などによって、一つの権限が付与されると直ちに、レジスタをそのディレクトリに割り当てる。ある権限が既にこのディレクトリのもとで付与されている場合には、そのディレクトリに専用のレジスタの更新を行う。

【００１８】

規則ＲＧ２

カレント・ディレクトリが変更される際、変更前にカレント・ディレクトリであったディレクトリをセキュリティ・レジスタに接続するリンクを失わせるが、但し、変更後のカレント・ディレクトリが変更前のカレント・ディレクトリの「子」にあたる場合は除く。

【００１９】

規則ＲＧ３

ｒ個のセキュリティ・レジスタが満杯の場合、即ち、記載された例ではｒ＝３個のセキュリティ・レジスタが全て使用されている場合は、最も古く割り当てられた、即ち根付き木構造の最上位の階層のレジスタを、規則ＲＧ１に従って変更後のカレント・ディレクトリに割り当てる。

【００２０】

注目すべきは、規則ＲＧ２の適用によって二つのセキュリティ・レジスタを同一の階層に割り当てることは不可能になり、その結果、一つのディレクトリに一つのセキュリティ・レジスタを割り当てることは、当該セキュリティ・レジスタに割り当てられた階層の

、ｉが１からｎの変数である、層Ｎｉによって実現可能になることである。

【００２１】

図２．１から２．１４は規則ＲＧ１、ＲＧ２及びＲＧ３の適用例を示す。これら及びその他の図において、黒丸はディレクトリを示し、灰色丸は選択されたディレクトリを示し、白丸はある権限を行使して選択されたディレクトリを示す。

【００２２】

図２．１はディレクトリの選択が行われていない状態を示し、一方図２．２と図２．３はそれぞれディレクトリＲｅｐ１とＲｅｐ２を選択した状態を示す。

【００２３】

規則ＲＧ１の適用は図２．４、２．６、２．８、２．１０、２．１２及び２．１４に示されている。図２．４は階層Ｎ２のディレクトリＲｅｐ２のもとでの一つの秘密情報を提示している様子を示している。図２．６は階層Ｎ３のディレクトリＲｅｐ３１のもとでの一つの秘密情報を提示している様子を示している。図２．８は階層Ｎ４のディレクトリＲｅｐ４１のもとでの一つの秘密情報を提示している様子を示している。図２．１０は階層Ｎ５のディレクトリＲｅｐ５１のもとでの一つの秘密情報を提示している様子を示している。図２．１２はディレクトリＲｅｐ４１のもとでの一つの秘密情報を提示している様子を示している。図２．１４はディレクトリＲｅｐ４２のもとでの一つの秘密情報を提示している様子を示している。

【００２４】

規則ＲＧ２の適用が図２．５、２．７及び２．９に示されているが、それは、そのディレクトリの「子」としての新しいディレクトリを選択する際、一つのセキュリティ・レジスタとそのディレクトリとの間のリンクの維持に関するものである。

【００２５】

図２．５、２．７及び２．９はそれぞれディレクトリＲｅｐ３１，Ｒｅｐ４１またはＲｅｐ５１の選択を示している。

【００２６】

規則ＲＧ２の適用が図２．１１及び図２．１３に示されているが、それは一つのセキュリティ・レジスタとそのディレクトリとの間のリンクの切断に関するものである。従って、図２．１１がディレクトリＲｅｐ４１の選択を示す一方で、図２．１３はディレクトリＲｅｐ４２の選択を示している。

【００２７】

規則ＲＧ３の適用が図２．１０に示されているが、そこでは最も古く割り当てられたレジスタＲ１が新しく選択されたディレクトリＲｅｐ５１に割り当てられている。

【００２８】

秘密情報を知っていることに結びついている権限情報を記憶することから成る過程（ａ）が実現されると、チップ・カードの所持者または端末によって提示された秘密情報を根付き木構造内で検索することから成る過程（ｂ）を実行することができる。

【００２９】

一つのアプリケーションの階層で提示された秘密情報は、その同じアプリケーションの階層で使用権が付与される。従って、階層の層Ｎｉの一つのアプリケーション内で一つの秘密情報の提示が成功すると、たとえ提示されたその秘密情報が物理的にはより上位の階層の層にある場合でも、規則ＲＧ１に従って、その階層の層専用のセキュリティ・レジスタが更新される。

【００３０】

一つの秘密情報を提示する規則は以下の通りである。

【００３１】

規則ＲＧ４

参照記号Ｓの一つの秘密情報を提示するという作業は、カレント・アプリケーションからルート・ディレクトリに向かって階層軸に沿って経ることで見つけられたその参照記号Ｓの第一の秘密情報の値を、チップ・カードの所持者または端末が知っていることを検

証するという作業になる。
【００３２】
　階層の層Ｎｉに位置したカレント・アプリケーションの階層での参照記号Ｓの秘密情報の提示は以下のことから成る中間過程によって実現される。
（ｂ１）カレント・ディレクトリの中で、つまり、階層Ｎｉで、セキュリティ管理システムを利用して参照記号Ｓの一つの秘密情報を検索し、アプリケーション内にこの秘密情報が存在することを検証し、
（ｂ２）この秘密情報が存在する場合には、例えば秘密コードの為の値、鍵の為の暗号等の、秘密情報の提示が成功していることを検証し、
　提示が成功していれば、参照記号Ｓの秘密情報に関連する権限が階層Ｎｉのカレント・アプリケーションの階層で付与され、
　提示が失敗すれば、参照記号Ｓの秘密情報に関連する権限は付与されずに提示の試みが終了し、
（ｂ３）階層Ｎｉのカレント・アプリケーション内に参照記号Ｓの秘密情報が存在しない場合には、カレント・アプリケーションの階層Ｎ（ｉ－１）の親のアプリケーション内に同じ参照記号の秘密情報が存在するかどうかを検索し、
（ｂ４）階層Ｎ（ｉ－１）の親のアプリケーションの階層に秘密情報が存在する場合には、提示が成功していることを検証し、
　提示が成功していれば、参照記号Ｓの秘密情報に関連する権限が階層Ｎｉのカレント・アプリケーションの階層で付与され、
　提示が失敗すれば、その参照記号Ｓの秘密情報に関連する権限は付与されずに提示の試みが終了し、
（ｂ５）階層Ｎ（ｉ－１）の親のアプリケーション内にその参照記号Ｓの秘密情報が存在しない場合には、階層Ｎ（ｉ－２）で参照記号Ｓの秘密情報の存在を検索して、以下同様に、階層軸に沿って、参照記号Ｓの秘密情報の存在が発見されない限り続行し、
（ｂ６）参照記号Ｓの秘密情報が発見されなかった場合には、提示の試みが終了する。
【００３３】
　規則ＲＧ４の幾つかの適用例が図３．１ａから３．６ａ及び３．１ｂから３．６ｂに示されている。図３．１ａと３．１ｂ、３．２ａと３．２ｂ、３．３ａと３．３ｂは権限が付与された例に対応しているが、一方、図３．４ａと３．４ｂ、３．５ａと３．５ｂ、３．６ａと３．６ｂは権限が付与されない例に対応している。
【００３４】
　図３．１ａでは秘密情報Ｓ３がディレクトリＲｅｐ４１のもとで局所的に存在し、ディレクトリＲｅｐ４１にはいかなるレジスタも割り当てられていない。図３．１ｂでは、秘密情報Ｓ３を知っていることが証明され、階層Ｎ４のディレクトリＲｅｐ４１にレジスタＲ３が割り当てられており、権限が付与されている。
【００３５】
　図３．２ａでは秘密情報Ｓ３がディレクトリＲｅｐ４１のもとで局所的に存在し、ディレクトリＲｅｐ４１にはレジスタＲ３が既に割り当てられている。従って、秘密情報Ｓ３を知っていることが証明されているならば、ディレクトリＲｅｐ４１に割り当てられたセキュリティ・レジスタＲ３は更新され（Ｓ３）、その結果、権限が付与されている（図３．２ｂ）。
【００３６】
　図３．３ａでは秘密情報Ｓ２はディレクトリＲｅｐ４１のもとで局所的に存在しておらず、ディレクトリＲｅｐ４１にはレジスタＲ３が既に割り当てられており、秘密情報Ｓ２がディレクトリＲｅｐ２、Ｒｅｐ１、Ｒｅｐ４２及びＲｅｐ５１のもとで同時に存在している。従って、秘密情報Ｓ２を知っていることが証明されているならば、ディレクトリＲｅｐ４１に割り当てられたセキュリティ・レジスタは更新され、その結果、権限が付与されている（図３．３ｂ）。
【００３７】

図３．４ａでは秘密情報Ｓ２はディレクトリＲｅｐ４１のもとで局所的に存在しておらず、ディレクトリＲｅｐ４１にはレジスタＲ３が既に割り当てられており、秘密情報Ｓ２がディレクトリＲｅｐ２、Ｒｅｐ１、Ｒｅｐ４２及びＲｅｐ５１のもとで同時に存在している。従って、秘密情報Ｓ２を知っていることが証明されなければ、その結果、ディレクトリＲｅｐ４１に割り当てられたセキュリティ・レジスタＲ３は更新されず、権限が付与されない（図３．４ｂ）。

【００３８】

図３．５ａでは秘密情報Ｓ２はディレクトリＲｅｐ４１のもとで局所的に存在しておらず、ディレクトリＲｅｐ４１にはレジスタＲ３が既に割り当てられており、秘密情報Ｓ２がディレクトリＲｅｐ２、Ｒｅｐ１、Ｒｅｐ４２及びＲｅｐ５１のもとで同時に存在している。従って、秘密情報Ｓ２を知っていることが証明されなければ、その結果、ディレクトリＲｅｐ４１に割り当てられたセキュリティ・レジスタＲ３は更新されず、権限が付与されない（図３．５ｂ）。

【００３９】

図３．６ａでは秘密情報Ｓ２はディレクトリＲｅｐ４１のもとで局所的に存在しておらず、ディレクトリＲｅｐ４１にはレジスタＲ３が既に割り当てられており、秘密情報Ｓ２がディレクトリＲｅｐ２、Ｒｅｐ１、Ｒｅｐ４２及びＲｅｐ５１のもとで同時に存在している。従って、秘密情報Ｓ２を知っていることが証明されなければ、その結果、ディレクトリＲｅｐ４１に割り当てられたセキュリティ・レジスタＲ３は更新されず、権限が付与されない（図３．６ｂ）。

【００４０】

過程（ｃ）は、アクセス条件を満たすための一つまたは複数の秘密情報を知っているということ、即ち、チップ・カードのある機能及びあるサービスの使用を保護する秘密情報が外界によく知られていること、即ち、必要とされる権限が確かに付与されているということを検証することからなる。

【００４１】

その為に、本発明では第五の規則ＲＧ５を実施し、それは以下のように記述される。

【００４２】

規則ＲＧ５

根付き木構造をカレント・アプリケーションからルート・アプリケーションに向かって階層軸に沿って経て、出会った第一の秘密情報Ｓが、カレント・アプリケーションと秘密情報Ｓを含むアプリケーションとを境界とする根付き木構造の区間に属するアプリケーションのうちの少なくとも一つによって知られている場合、即ち、正確に提示された場合には秘密情報Ｓを知っているということを必要とする機能を許可し、しかも、その場合にのみ許可するが、但し、秘密情報Ｓがカレント・アプリケーションの中に存在する場合にはこれらのアプリケーションは混同されることがあり得る。

【００４３】

過程（ｃ）を実行する為には、管理システムが以下の過程を実行しなければならない。

（ｃ１）セキュリティ・レジスタが階層Ｎｉのカレント・アプリケーションに関連づけられていることを検証し、

（ｃ２）階層Ｎｉのカレント・アプリケーションに関連づけられているセキュリティ・レジスタがあり、かつ、該セキュリティ・レジスタが必要な権限情報を含んでいる場合には、機能を許可して検証を終了し、

（ｃ３）どのセキュリティ・レジスタもカレント・アプリケーションに関連づけられていない場合、または関連づけられたレジスタが必要な権限情報を含んでいない場合には、階層Ｎｉのカレント・アプリケーション内で参照記号Ｓの秘密情報の存在を検索し、

（ｃ４）秘密情報がカレント・アプリケーション内に存在する場合には、機能を拒絶して検証を終了し、

（ｃ５）階層Ｎｉのカレント・アプリケーション内に参照記号Ｓの秘密情報が存在しな

い場合には、カレント・アプリケーションの階層Ｎ（ｉ－１）の親のアプリケーションにセキュリティ・レジスタが関連付けられていることを検証し、

（ｃ６）親のアプリケーションに関連づけられたセキュリティ・レジスタが、機能を使うのに必要な権限情報を含んでいる場合には、機能を許可して検証を終了し、

（ｃ７）どのセキュリティ・レジスタも親のアプリケーションに関連づけられていない場合、または関連づけられたセキュリティ・レジスタが必要な権限情報を含んでいない場合には、カレント・アプリケーションの階層Ｎ（ｉ－１）の親のアプリケーション内で参照記号Ｓの秘密情報の存在を検索し、

（ｃ８）参照記号Ｓの秘密情報が、階層Ｎ（ｉ－１）の親のアプリケーション内に存在する場合には、機能を拒絶して検証を終了し、

（ｃ９）階層Ｎ（ｉ－１）の親のアプリケーション内に参照記号Ｓの秘密情報が存在しない場合には、カレント・アプリケーションからルート・アプリケーションに向かって階層軸に沿って階層Ｎ（ｉ－２）のアプリケーション（親の親）にセキュリティ・レジスタが関連付けられていることを検証し、以下同様に、参照記号Ｓの秘密情報の存在が発見されない限り、続行し、

（ｃ１０）秘密情報が発見されなかった場合は、機能を拒絶して検証を終了する。

【００４４】

図４．１と図４．２では機能が許可された二つの例を示しており、一方、図４．３、４．４、４．５と４．６では機能が拒絶された四つの例を示している。

【００４５】

図４．１では、機能が受け入れられているが、それは秘密情報Ｓ３が局所的に存在しており、それがディレクトリＲｅｐ４１のもとで知られているからである。

【００４６】

図４．２では、機能が受け入れられているが、それは秘密情報Ｓ１が局所的に存在していないけれども、それがディレクトリＲｅｐ２のもとで知られているからである。

【００４７】

図４．３では、機能が拒絶されているが、それは秘密情報Ｓ３がディレクトリＲｅｐ４１のもとで局所的に存在しており、そのディレクトリのもとではいかなる権限も付与されなかったからである。

【００４８】

図４．４では、機能が拒絶されているが、それは秘密情報Ｓ３がディレクトリＲｅｐ４１のもとで局所的に存在しており、ディレクトリＲｅｐ４１にセキュリティ・レジスタＲ３が割り当てられてはいるが、秘密情報Ｓ３を知っていることが証明されていないからである。

【００４９】

図４．５では、機能が拒絶されているが、それはディレクトリＲｅｐ４１のもとでは局所的に存在せず、ディレクトリＲｅｐ３１の中にも存在しない秘密情報Ｓ２がディレクトリＲｅｐ２のもとでは存在しており、ディレクトリＲｅｐ２にはいかなるセキュリティ・レジスタも割り当てられてはいないからである。注目すべきは、秘密情報Ｓ２がディレクトリＲｅｐ１のもとで知られているにもかかわらず、機能が拒絶されていることである。

【００５０】

図４．６では、機能が拒絶されているが、それは、秘密情報Ｓ１がディレクトリＲｅｐ５１及びＲｅｐ３２のもとで存在しているのにもかかわらず、ディレクトリＲｅｐ４１の階層軸をディレクトリＲｅｐ１に向かって経ても秘密情報Ｓ１が見つからないからである。

【図面の簡単な説明】

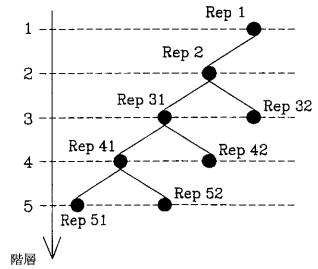
【図１】　図１はディレクトリの根付き木構造の一例である。

【図２】　図２．１から２．１４は、一つのディレクトリに対する一つのセキュリティ・レジスタの割当てまたは離反の三つの規則の適用例を示している。

【図３】　図３．１ａから３．６ａ、及び３．１ｂから３．６ｂは秘密情報の提示規則の適用例を示している。
【図４】　図４．１から４．６は必要な権限の付与を検証する規則の適用例を示している。

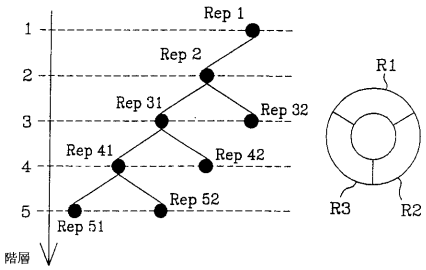
【図１】

FIG.1



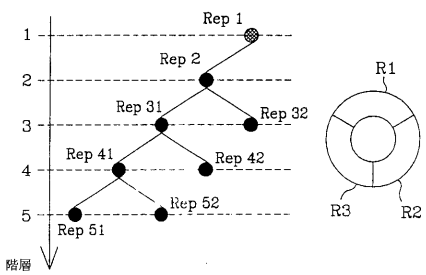
【図2.1】

FIG.2.1



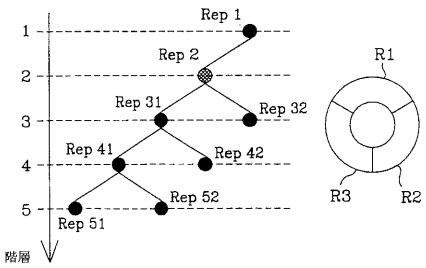
【図2.2】

FIG.2.2



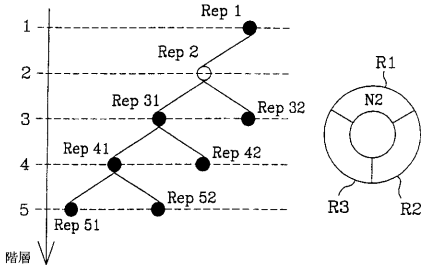
【図2.3】

FIG.2.3



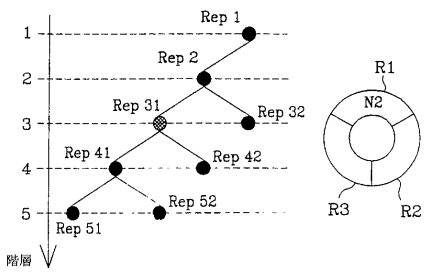
【図2.4】

FIG.2.4



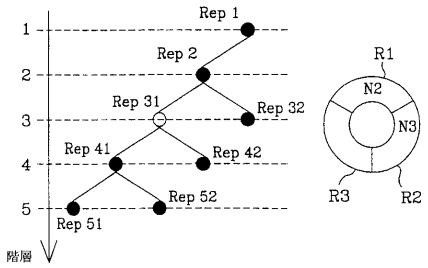
【図2.5】

FIG.2.5



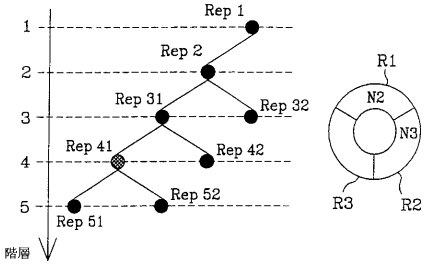
【図2.6】

FIG.2.6



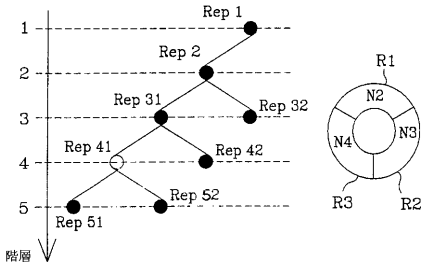
【図2.7】

FIG.2.7



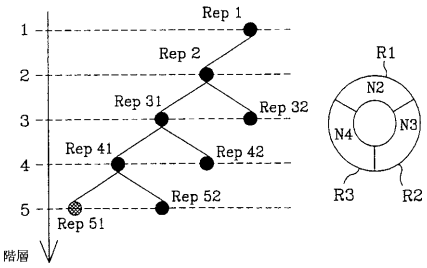
【図2.8】

FIG.2.8



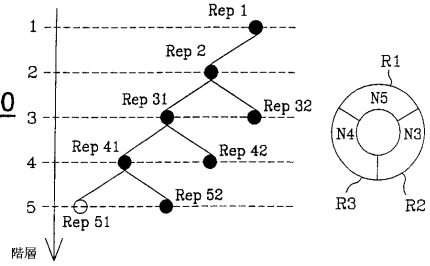
【図2.9】

FIG.2.9



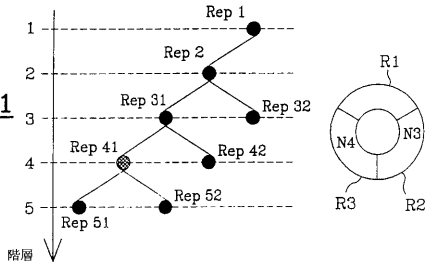
【図2.10】

FIG.2.10



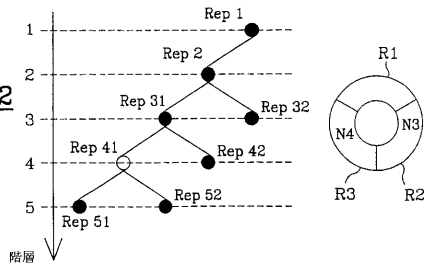
【図2.11】

FIG.2.11



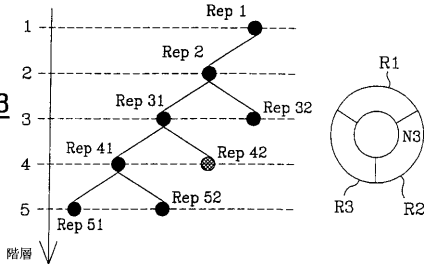
【図2.12】

FIG.2.12



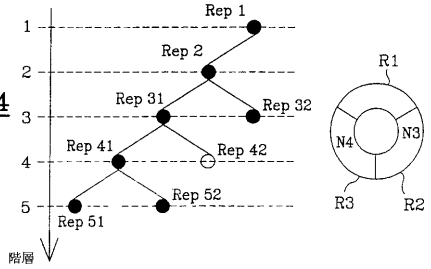
【図2.13】

FIG.2.13



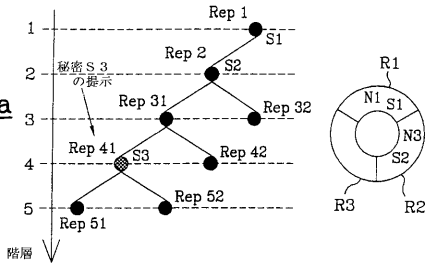
【図2.14】

FIG.2.14



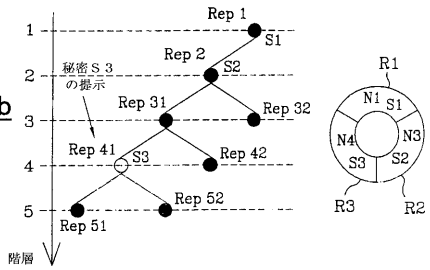
【図3.1a】

FIG.3.1a



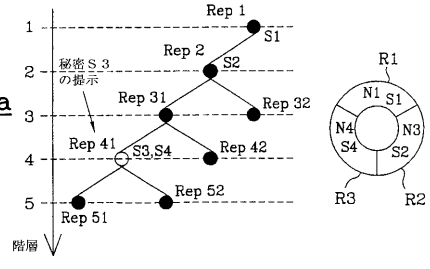
【図3.1b】

FIG.3.1b



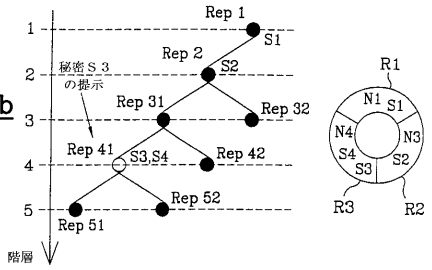
【図3.2a】

FIG.3.2a



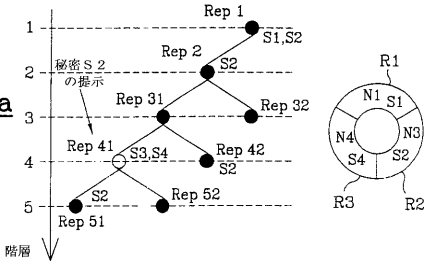
【図3.2b】

FIG.3.2b



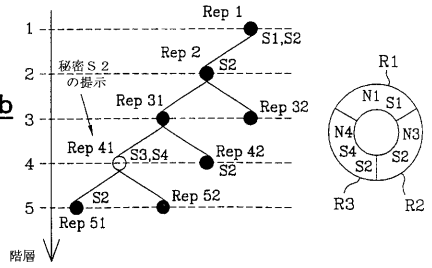
【図3.3a】

FIG.3.3a



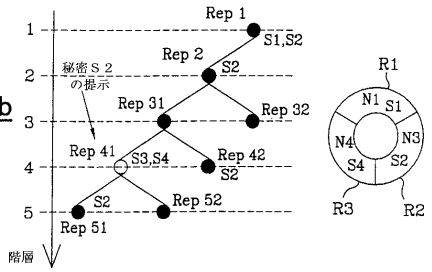
【図3.3b】

FIG.3.3b



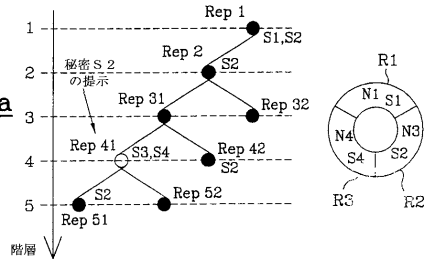
【図3.5b】

FIG.3.5b



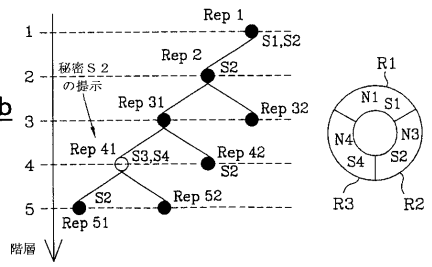
【図3.6a】

FIG.3.6a



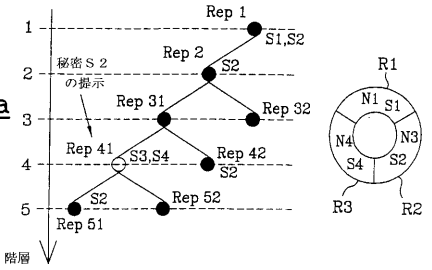
【図3.6b】

FIG.3.6b



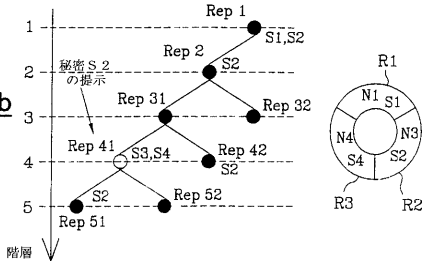
【図3.4a】

FIG.3.4a



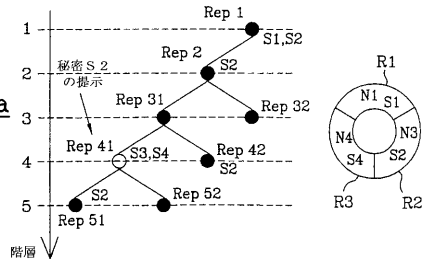
【図3.4b】

FIG.3.4b



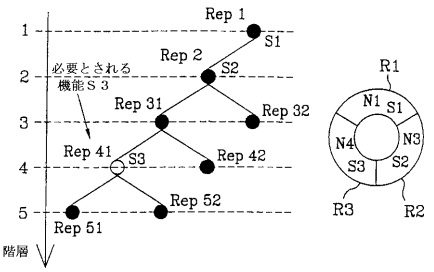
【図3.5a】

FIG.3.5a



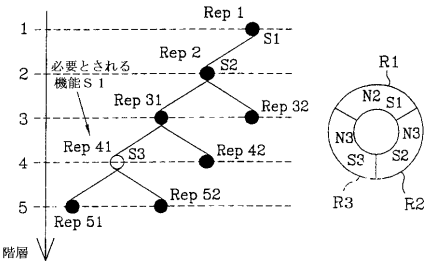
【図4.1】

FIG.4.1



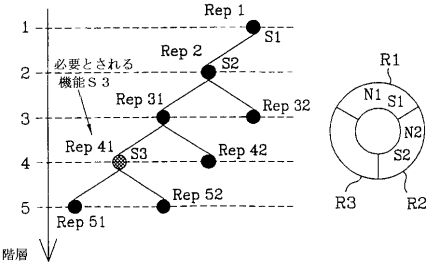
【図4.2】

FIG.4.2



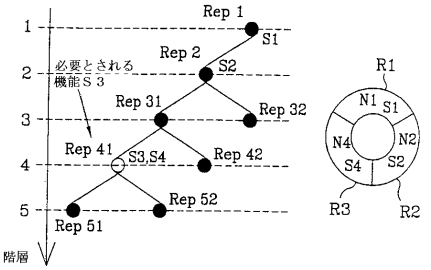
【図4.3】

FIG.4.3



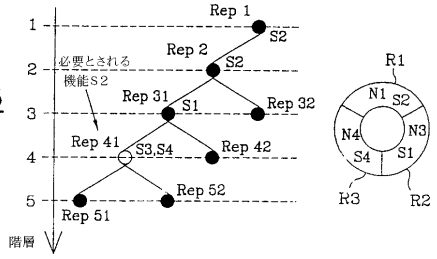
【図4.4】

FIG.4.4



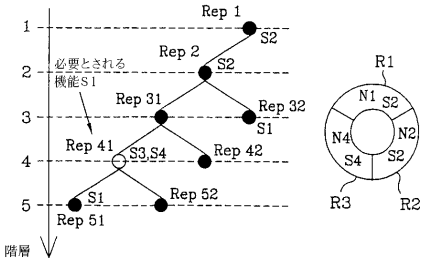
【図4.5】

FIG.4.5



【図4.6】

FIG.4.6



フロントページの続き

(72)発明者　ブラン，フィリップ
フランス共和国，エフ－１３６００　ラ　シオタ，ロティスモン　デ　セヴリエ，アレ　デュ　リバ，１４

審査官　高橋　克

(56)参考文献　特開平０６－０６０２３５（ＪＰ，Ａ）
特開平０２－０６７６５１（ＪＰ，Ａ）
特開平０６－１０３２３９（ＪＰ，Ａ）
特開平０６－２７４３９７（ＪＰ，Ａ）
特開平０７－１０４８８２（ＪＰ，Ａ）
特開平０９－０７３４１６（ＪＰ，Ａ）
特開平０９－２９３０２３（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06F 12/14
G06F 12/00
G06K 19/073
G06K 19/10