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

特許第4348190号
(P4348190)

(45)発行日　平成21年10月21日(2009.10.21)　(24)登録日　平成21年7月24日(2009.7.24)

(51)Int.Cl.		F I		
G06F 12/00	(2006.01)	G06F	12/00	546Z
G06F 21/24	(2006.01)	G06F	12/00	547Z
G06K 17/00	(2006.01)	G06F	12/14	560B
G06K 19/07	(2006.01)	G06K	17/00	D
		G06K	19/00	N

請求項の数 27　(全 21 頁)

(21)出願番号	特願2003-550174 (P2003-550174)
(86)(22)出願日	平成14年12月9日 (2002.12.9)
(65)公表番号	特表2005-512205 (P2005-512205A)
(43)公表日	平成17年4月28日 (2005.4.28)
(86)国際出願番号	PCT/GB2002/005565
(87)国際公開番号	WO2003/049056
(87)国際公開日	平成15年6月12日 (2003.6.12)
審査請求日	平成17年12月9日 (2005.12.9)
(31)優先権主張番号	0129360.4
(32)優先日	平成13年12月7日 (2001.12.7)
(33)優先権主張国	英国 (GB)
(31)優先権主張番号	0225036.3
(32)優先日	平成14年10月28日 (2002.10.28)
(33)優先権主張国	英国 (GB)

(73)特許権者　504218369
エセブス・リミテッド
イギリス国　ロンドン　イーシー３エヌ　１エイエイチ，アルドゲート・ハイ・ストリート　９，ザ・マトリックス
(74)代理人　100089705
弁理士　社本　一夫
(74)代理人　100076691
弁理士　増井　忠弐
(74)代理人　100075270
弁理士　小林　泰
(74)代理人　100080137
弁理士　千葉　昭男
(74)代理人　100096013
弁理士　富田　博行

最終頁に続く

(54)【発明の名称】スマートカード・システム

(57)【特許請求の範囲】

【請求項１】

ファイルシステムと、オンデバイスの前記ファイルシステムが少なくとも１つのオフデバイスのファイルおよび／またはアプリケーションとインターフェースデバイスによりインターフェースをとることを可能にするオペレーティングソフトウェアと、を担持するプログラマブルデバイスであって、

前記ファイルシステムの構造及び内容、或いは、前記ファイルシステムにアクセスするために使用されるコマンド或いはそれに関連するセキュリティ条件を変更するために、スクリプトを実行する手段を備え、

前記オンデバイスのファイルシステムの構造及び内容、前記ファイルシステムにアクセスするために使用されるコマンド、及び、それに関連するセキュリティ条件が、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた少なくとも１つのファイルによって定義され、

前記スクリプトを実行する手段が、前記オンデバイスのファイルシステムの構造及び内容、或いは、前記ファイルシステムにアクセスするために使用されるコマンド或いはそれに関連するセキュリティ条件を変更するために、前記少なくとも１つのファイルから導出されて前記インターフェースデバイスにおいて前記プログラマブルデバイスにロードされるスクリプトを実行するように動作する、プログラマブルデバイス。

【請求項２】

請求項１に記載のプログラマブルデバイスであって、自己記述メッセージングのための

前記Ｗｅｂ（インターネット）標準の言語が、拡張マークアップ言語（「ＸＭＬ」）であり、前記少なくとも１つのファイルがＸＭＬドキュメントである、デバイス。

【請求項３】

請求項１又は請求項２に記載のプログラマブルデバイスであって、前記少なくとも１つのファイルが、圧縮フォーマットで格納される、デバイス。

【請求項４】

請求項１～請求項３の何れか１項に記載のデバイスであって、インターフェースデバイスからロードされる１又はそれ以上のＡＰＤＵ（アプリケーション・プロトコル・データ・ユニット）を実行し、前記ファイルシステム、構造及び内容、或いは、前記ファイルシステムにアクセスするために使用されるコマンド或いはそれに関連するセキュリティ条件を変更することが可能なスクリプトエンジンを備えるデバイス。

【請求項５】

請求項４に記載のデバイスであって、前記スクリプトエンジンによって実行される前に、前記インターフェースデバイスからロードされたＡＰＤＵを暗号化された形式で復号する手段を備えるデバイス。

【請求項６】

請求項１～請求項５の何れか１項に記載のデバイスであって、前記インターフェースデバイスからロードされるスクリプトを前記スクリプトエンジンが実行する前に、デジタル署名を検証する手段を備えるデバイス。

【請求項７】

請求項１～請求項６の何れか１項に記載のデバイスであって、前記インターフェースデバイスからロードされるスクリプトを前記スクリプトエンジンが実行する前に、リテイルＭＡＣ（メッセージ認証コード）を検証する手段を備えるデバイス。

【請求項８】

カード読み取り装置において請求項１に記載のプログラマブルデバイスとインターフェースをとるためのカード読み取りインターフェースデバイスであって、

自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた少なくとも１つのファイルから導出されるスクリプトを前記プログラマブルデバイスにロードすることによって前記オンデバイスのファイルシステムをアップグレードし、前記オンデバイスのファイルシステムの構造及び内容、或いは、前記ファイルシステムにアクセスするために使用されるコマンド或いはそれに関連するセキュリティ条件の定義を変更する手段を備えるインターフェースデバイス。

【請求項９】

請求項８に記載のインターフェースデバイスであって、

カード読み取り装置においてプログラマブルデバイスによって実行される前記オンデバイスのファイルシステムのバージョンを判定し、前記バージョンと該インターフェースデバイスに保持されるバージョンとの比較に依存してＡＰＤＵのスクリプトを生成する手段と、

前記オンデバイスのファイルシステムをアップグレードするために、前記プログラマブルデバイスに前記スクリプトをロードする手段と、

を備えるインターフェースデバイス。

【請求項１０】

請求項８に記載のインターフェースデバイスであって、

ソフトウェア配信システムからのＡＰＤＵのスクリプトを受け取る手段と、

前記オンデバイスのファイルシステムをアップグレードするために、前記プログラマブルデバイスに前記スクリプトをロードする手段と、

を備えるインターフェースデバイス。

【請求項１１】

請求項１０に記載のインターフェースデバイスであって、

前記ソフトウェア配信システムから受け取ったＡＰＤＵのスクリプトを暗号化された形

式で復号する手段を備えるインターフェースデバイス。
【請求項１２】
請求項８～請求項１１の何れか１項に記載のインターフェースデバイスであって、
前記プログラマブルデバイスにロードする前に、前記ＡＰＤＵのスクリプトを暗号化する手段を備えるインターフェースデバイス。
【請求項１３】
請求項８～請求項１２の何れか１項に記載のインターフェースデバイスであって、
前記プログラマブルデバイスにロードする前に、前記ＡＰＤＵのスクリプトにデジタル署名を加える手段を備えるインターフェースデバイス。
【請求項１４】
請求項８～請求項１３の何れか１項に記載のインターフェースデバイスであって、
前記プログラマブルデバイスにロードする前に、前記ＡＰＤＵのスクリプトにＭＡＣを加える手段を備えるインターフェースデバイス。
【請求項１５】
請求項１に記載のプログラマブルデバイスと請求項８に記載のインターフェースデバイスとを備えるシステムであって、
自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされたファイルを生成又は変更し、前記オンデバイスのファイルシステムの構造及び内容、並びに、前記ファイルシステムにアクセスするために使用されるコマンド及びそれに関連するセキュリティ条件を定義するためのソフトウェアを実行する手段と、
前記ソフトウェアを実行する手段と前記インターフェースデバイスとの間の安全なソフトウェア配信手段であって、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルの安全な配信を提供するソフトウェア配信手段と、
を備えるシステム。
【請求項１６】
請求項１５に記載のシステムであって、前記ソフトウェアを実行する手段は、
配信の前に、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルを暗号化する手段を備える、システム。
【請求項１７】
請求項１５又は請求項１６に記載のシステムであって、
配信の前に、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルに、デジタル署名を加える手段を備えるシステム。
【請求項１８】
請求項１５～請求項１７の何れか１項に記載のシステムであって、
配信の前に、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルに、ＭＡＣを加える手段を備えるシステム。
【請求項１９】
請求項１５～請求項１８の何れか１項に記載のシステムであって、
前記ソフトウェアを実行する手段は、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされたファイルからＡＰＤＵのスクリプトを生成し、前記スクリプトを前記ソフトウェア配信手段によって前記インターフェースデバイスに配信するように動作し、
前記インターフェースデバイスは、前記オンデバイスのファイルシステムをアップグレードするために、前記スクリプトを前記プログラマブルデバイスにロードする手段を備える、システム。
【請求項２０】

請求項１５～請求項１９の何れか１項に記載のシステムであって、
前記ソフトウェアを実行する手段は、プログラマブルデバイスの全部のメモリイメージを備えるファイル又は「仮想カード」ファイルを生成するように動作し、
前記インターフェースデバイスは、前記仮想カードファイルを前記プログラマブルデバイスにロードするように動作する、システム。
【請求項２１】
請求項１に記載のプログラマブルデバイス上のオンデバイスのファイルシステムをアップグレードする方法であって、
カード読み取り装置においてプログラマブルデバイスとインターフェースをとるためのカード読み取りインターフェースデバイスを提供するステップと、
前記インターフェースデバイスにおいて、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた少なくとも１つのファイルから導出されるスクリプトを前記プログラマブルデバイスにロードするステップと、
スクリプトを実行するための前記プログラマブルデバイス上の手段によって、前記少なくともファイルから導出された前記スクリプトを実行して、前記オンデバイスのファイルシステムの構造及び内容、或いは、前記ファイルシステムにアクセスするために使用されるコマンド或いはそれに関連するセキュリティ条件の定義を変更するステップと、
を含む方法。
【請求項２２】
請求項２１に記載の方法であって、ＡＰＤＵのスクリプトは、ソフトウェア配信システムから前記インターフェースデバイスにおいて受け取られ、前記オンデバイスのファイルをアップグレードするために前記プログラマブルデバイスにロードされる、方法。
【請求項２３】
請求項２１に記載の方法であって、
前記インターフェースデバイスにおいて、プログラマブルデバイスによって実行される前記オンデバイスのファイルシステムのバージョンが判定され、
ＡＰＤＵのスクリプトは、前記バージョンと前記インターフェースデバイスに保持されるバージョンとの比較に依存して生成され、前記オンデバイスのファイルシステムをアップグレードするために前記プログラマブルデバイスにロードされる、方法。
【請求項２４】
請求項２２に記載の方法であって、
前記ソフトウェアを実行する手段を使用して、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされたファイルを生成又は変更し、ファイルシステムの構造及び内容、並びに、前記ファイルシステムにアクセスするために使用されるコマンド及びそれに関連するセキュリティ条件を定義するステップと、
自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルを、前記安全なソフトウェア配信手段によって前記インターフェースデバイスに配信するステップとを含む、方法。
【請求項２５】
請求項２４に記載の方法であって、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルは、配信の前に暗号化される、方法。
【請求項２６】
請求項２４又は請求項２５に記載の方法であって、デジタル署名は、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファイル又は該ファイルから導出される任意のスクリプトもしくはファイルに加えられる、方法。
【請求項２７】
請求項２４～請求項２６の何れか１項に記載の方法であって、ＭＡＣは、自己記述メッセージングのためのＷｅｂ（インターネット）標準の言語でフォーマットされた前記ファ

イル又は該ファイルから導出される任意のスクリプトもしくはファイルに加えられる、方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、いわゆる「スマートカード」などのプログラマブルデバイスを使用するシステムに関連し、そのシステムには、そのようなデバイスを財務トランザクションのために利用するシステムが含まれる。詳細には、本発明は、少なくとも１つのオンデバイスのファイルおよび／またはアプリケーションを担持するプログラマブルデバイスと、少なくとも１つのオフデバイスのファイルおよび／またはアプリケーションを備えたインターフェースデバイスであって、プログラマブルデバイスまたは各プログラマブルデバイスとインターフェースをとるためのインターフェースデバイスとを含むシステムにおいて使用するためのプログラマブルデバイス、インターフェースデバイス、およびオペレーティングソフトウェアを担持するマシン可読媒体に関連し、そのオペレーティングソフトウェアは、使用中、オンデバイスのファイルおよび／またはアプリケーションとオフデバイスのファイルおよび／またはアプリケーションが互いにインターフェースをとることを可能にする。

【背景技術】

【０００２】

今日のスマートカードソリューションのほとんどは、メーカ独自のものであり、特定のビジネス問題を解決するように構築されている。システムは、いくつかのＩＳＯ標準に基づいているが、一般に、それらのシステムは、すべてメーカ独自の技術に基づくスマートカード自体、カードアプリケーション、および端末アプリケーションから、構築されている。これは、顧客を納入業者に縛る効果を有する。

【０００３】

また、そのようなシステムは、開発する費用がかさみ、柔軟性を欠き、変更するのが困難であり、展開し、管理するのが困難でもある。

以上の問題が、より広い市場におけるスマートカード技術の到来を遅らせている。

【０００４】

しかし、これは、大手の発行銀行によって「ＥＭＶ」（ユーロペイ（Ｅｕｒｏｐａｙ）、マスターカード（Ｍａｓｔｅｒｃａｒｄ）、およびビザ（Ｖｉｓａ）の支払アプリケーション規格）の標榜の下に発行されるチップ対応のクレジットカードおよびデビットカードの幅広い採用とともに変わらなければならない可能性がある。銀行業とは別に、大量輸送、政府機関および地方自治体機関、教育関係省庁、医療機関、および軍隊を含め、その他のセクタも、スマートカード技術を使用する可能性に注目している。以上のセクタのすべてが、スマートカードアプリケーションを使用することによって最もよく満たすことができる、安全なＩＤから、カード所持者認証、愛顧感謝ボーナススキーム、ならびに健康記録などの個人情報を格納することまでを含む要件を有する。そのようなシステムは、様々な一連のスマートカードアプリケーションを利用してその要件を実行し、管理することができる。

【０００５】

今日、スマートカード対応になることが可能な非常に多くの他のシステムが存在するが、そのようなシステムの統合は高価であり、多くのメーカ独自のソリューションが関与するため、顧客を特定のカード納入業者に縛ることにもなりがちである。

【０００６】

今日、次の２つのカテゴリのスマートカードソリューション、すなわち、特別仕様で作られ、単一の特定のタイプのチップ上で実行されるメーカ独自の単独アプリケーションと、常駐アプリケーションから基礎にあるチップの実装を隠すオペレーティングシステムを利用するマルチアプリケーション・オペレーティングシステムとが存在する。

【０００７】

メーカ独自の単独アプリケーション・システムが利用可能な場合、別のタイプのチップにアプリケーションを移植することは、時間がかかり、高価なプロセスになる可能性があるが、システムにおいて使用されるカード自体は安価である。

【０００８】

他方、マルチアプリケーション・スマートカード・システムは、新しいアプリケーションとしての構成するのが非常に容易であり、あるいはアプリケーションの新しいバージョンを、ライブの動作中にスマートカードにロードすることができる。顧客は、特定のカード納入業者に縛られず、そのバージョンのオペレーティングシステムをサポートする任意のチップを選択することができる。ただし、マルチアプリケーション・オペレーティングシステムは、不正なコードまたはデータがカードにロードされること、またはカード上のアプリケーションの許可のない削除を防止するように強いセキュリティで保護されていなければならない。これらのセキュリティ機能、ならびに関連するキー管理の実装により、マルチアプリケーション・システムで使用されるカードは高価になり、管理するのが難しくなり、カードにアプリケーションをロードするための能力、ならびにカードからアプリケーションを削除するための能力を提供するのに複雑なインフラストラクチャを要する傾向がある。

【０００９】

さらに、マルチアプリケーション・システムの耐用年数の間に、カード上でも、カード外でも、システムに変更を行わなければならない可能性が高く、これもやはり、高価な、時間のかかる作業になる可能性がある。

【発明の開示】

【課題を解決するための手段】

【００１０】

本発明によれば、自己記述メッセージ用のＷｅｂ（インターネット）標準言語でフォーマットされた少なくとも１つのファイルを特徴とするオペレーティングソフトウェアを担持するプログラマブルデバイス、インターフェースデバイス、またはマシン可読媒体が提供され、前記ファイルは、オンデバイスのファイルおよび／またはアプリケーションとオフデバイスのファイルおよび／またはアプリケーションがインターフェースをとることを可能にするファイルシステムと、ファイルシステムにアクセスするためのファイルコマンドとの少なくともいくつかを含む。

【００１１】

好ましくは、自己記述メッセージのためのＷｅｂ標準言語は、拡張マークアップ言語（「ＸＭＬ」）である。

さらなる好ましい実施形態では、自己記述メッセージ用のＷｅｂ（インターネット）標準言語でフォーマットされた前記ファイルまたは各ファイルが、圧縮フォーマットでデバイス上に格納される。これにより、端末装置またはインターフェースデバイスが、カードの中のファイルシステムのフォーマットを「検出する」ことができるようになり、より容易な管理が可能になる。

【００１２】

一実施形態では、本発明のシステムは、ＩＳＯ７８１６に準拠するスマートカードアプリケーションと対話するコンポーネントベースのアーキテクチャ・フレームワークである。このアーキテクチャにより、新しいアプリケーションおよび既存のアプリケーションが、スマートカード上に格納された情報と対話することが、その情報がどのようにどこをソースとしているかの知識を全く有さずに、できるようになる。システムは、実行時に、１組のセキュリティポリシーおよびセキュリティ条件を使用して、カード上に格納されたファイルおよびオブジェクトに対するアクセス権を決定し、相応してシステムの挙動を変化させる。例えば、ファイルに対する読み取りコマンドが、ユーザにＰＩＮ認証を受けさせることを要する場合、製品は、製品がデータを読み取る前に、ユーザがＰＩＮを入力し、ＰＩＮが検証されることを確実にする。スマートカードは、アプリケーションと同じセキュリティポリシーおよびセキュリティ条件を実施し、したがって、カード上のオブジェク

トに対する適切な許可されたアクセスを確実にする最終的な責任を負う。

【００１３】

ファイルシステム、ファイル構造、およびファイル内容、ファイルシステムにアクセスするためのコマンド、ならびにファイルシステム内のファイルに関連するセキュリティ条件はすべて、自己記述メッセージ用のＷｅｂ標準である拡張マークアップ言語（「ＸＭＬ」）でフォーマットされたファイルを使用して一意的に記述することができる。

【００１４】

このシステムは、スマートカード常駐アプリケーション、および対応するカードを受け入れる端末装置を以下のステップによって迅速かつ安価に市場に出すことができるという利点を有する。すなわち、

既製のＸＭＬコンフィギュレータツールを使用してファイルシステムおよびセキュリティ環境条件を構成するステップ、

カード上でファイルシステムおよびセキュリティを作成するステップ（適切なアプリケーションが既にロードされているものと想定して）、および

ＸＭＬファイルを構築情報ドキュメントとして使用して１組のあらかじめ構成された機能から将来のカードアプリケーションを構築するステップである。

【００１５】

ＸＭＬスタイルシート、およびその他の標準の技術を使用してＸＭＬドキュメントから、特定のクライアントユーザインターフェースアプリケーションおよび汎用のクライアントユーザインターフェースアプリケーションを、迅速に開発することができる。また、再構成可能なコンポーネントおよび既存のアプリケーションが、システムのＸＭＬインターフェースを使用してスマートカード情報にアクセスし、許可された場合、そのデータを変更して、カードに再び書き込むことができる。

【００１６】

本発明によるシステムの実施形態を、例として、図面を参照して以下に詳細に説明する。

【発明を実施するための最良の形態】

【００１７】

以上に概説したとおり、本発明のシステムは、汎用コンポーネントを使用してオフカードのアプリケーションにオンカードのアプリケーションに対するアクセスを提供することを目的とする。この汎用コンポーネントは、本発明によれば、構成データを格納するように、ＸＭＬファイル、またはその他のＷｅｂ標準マークアップ言語の使用を介して非常に容易に構成される。このファイルは、以下を記述する。すなわち、

カードアプリケーションに送ることができる「ＡＰＤＵ」（アプリケーション・プロトコル・データ・ユニット）コマンドの内容および構造、

カード上の安全なファイルシステム上で読み取ることができ、更新することができるデータの構造および内容、および

カード上のオブジェクトに対するコマンドに関連するセキュリティ条件およびセキュリティ・アクション。

【００１８】

システムが始動した際、システムのスマートカード端末読み取り装置が構成される。オフカードのアプリケーションが、コントローラコンポーネントに登録されて、一部のカードアプリケーションに関心があることを示す。カードがカード読み取り装置に挿入された際、コントローラコンポーネントがカードに問い合わせて、カード上で実行されているアプリケーションを特定する。コントローラは、次に、見つかったオンカードのアプリケーションに関心があるすべてのオフカードのアプリケーションに通知を行う。すると、これらのオフカードのアプリケーションコンポーネントは、カードが取り外されるまで、オンカードのアプリケーションと対話することができる。

【００１９】

コントローラコンポーネントは、オフカードのアプリケーションによるカード上のアプ

リケーションに対するアクセスを整理して、カードアプリケーションとオフカードアプリケーションが対話している間の同期問題を防止する。

【００２０】

図１は、本発明のシステムのコンポーネント・アーキテクチャを示している。システムは、様々なオンカードのアプリケーション１１を担持するスマートカード１０を含む。システムは、読み取り装置抽象レイヤ２０、カードデータ表現３０、および一般ビジネスオブジェクト４０を含むカード読み取りインフラストラクチャ１９をさらに提供する。

【００２１】

図１に示すとおり、オフカードアプリケーションは、読み取り装置抽象インターフェース２２および汎用カードアプリケーションハンドラ２４を含む読み取り装置抽象レイヤ２０を含む。

【００２２】

読み取り抽象インターフェース２２は、システムのスマートカード端末装置の中に存在するカード読み取り装置（図示せず）を介してオンカードのアプリケーションとインターフェースをとる。インターフェース２２は、システムにおいて使用されるカード読み取り装置およびカード読み取り装置ドライバをカプセル化して、この装置およびドライバを、呼び出し側コンポーネントから隠す。好ましい実施形態では、読み取り装置抽象インターフェースは、任意のＰＣ／ＳＣ対応またはオープン・カード・フレームワーク（ＯＣＦ）対応の読み取り装置ドライバを使用してスマートカードアプリケーションに対するアクセスを提供するオープンソースのＪａｖａ（登録商標）ベースのスマートカードフレームワークであるＯＣＦのまわりの独自ラッパである。このインターフェースは、ＯＣＦを代替の、例えば、小型デバイスプラットフォーム上のメーカ独自の読み取り装置アプリケーション・プログラミング・インターフェース（「ＡＰＩ」）とインターフェースをとることができるフレームワークで置き換えることにより、ＯＣＦへの依拠を回避する形で設計することもできる。

【００２３】

読み取り装置抽象レイヤ２０の第２のコンポーネントである汎用カードアプリケーションコマンドハンドラ２４は、カードアプリケーションごとに動作し、ＡＰＤＵコマンドを策定し、そのコマンドをカードアプリケーションに読み取り装置抽象インターフェース２２を介して送ることを主に担う。適切な汎用カードアプリケーションコマンドハンドラは、例えば、ＩＳＯ規格７８１６パート４によって定義される。

【００２４】

汎用カードアプリケーションコマンドハンドラ２４は、オンカードのアプリケーションと通信することを望むクライアントコンポーネントにｄｏＣｏｍｍａｎｄなどのメソッドを提供する。クライアントコンポーネントは、実行されるべきコマンドのタイプ、コマンドが実行されるべき対象のオブジェクト、ならびにコマンドが実行されるべき対象のファイルのＩＤなどのあらゆる他の関連のある情報を示す。例えば、クライアントコンポーネントは、ｓｅｌｅｃｔＦｉｌｅというコマンドが、ｆｉｌｅ１という名前の付いた基本ファイルに対して実行されるべきことを示すことができる。

【００２５】

次に、コマンドハンドラ２４は、クライアントコンポーネントからの情報と、オンカードアプリケーションによってサポートされるすべての有効なコマンドの構造およびフォーマットを含むあらかじめ構成されたＸＭＬファイルとを使用して、適切なＡＰＤＵコマンドを構成してカード１０に送る。コマンドがＸＭＬファイルの中で見つからない場合、そのコマンドは、カード上でサポートされる有効なコマンドではない。通常、ＡＰＤＵの中のデータブロックは、最大で２５６バイトのデータしか含むことができない。それより大きいデータブロックの場合、コマンドハンドラ２４は、すべてのデータがカード１０に送られるまで、最大で２５６バイトをそれぞれが含むいくつかのＡＰＤＵコマンドを構成し、送らなければならない。

【００２６】

汎用カードアプリケーションコマンドハンドラ２４が、操作の成功、あるいはそうでないことを示すＡＰＤＵ応答を、読み取り装置抽象インターフェース２２から返されて受け取る。ＡＰＤＵ応答は、カードからのデータを含むデータブロックまたはデータブロック群を含むことが可能である。これらが、呼び出し側コンポーネントに戻される。

【００２７】

カードデータ表現レイヤ３０は、カードアプリケーションファイルシステムの内容をカプセル化し、明確に定義されたインターフェースを介してそのデータに対するアクセスを提供するカードアプリケーションファイル内容マネージャ３２を含む。ファイルシステムの内容は、ＸＭＬドキュメントで表現され、ＸＭＬドキュメントの中に格納される。

【００２８】

クライアントコンポーネントが、内容マネージャ３２からのデータの要求を行い、要求のデータを含むＸＭＬノードを戻される。データに対する更新は、内容マネージャ３２を介しても行われる。すなわち、クライアントコンポーネントが、更新済みのデータを含むＸＭＬノードを送り、このノードが、汎用カードアプリケーションコマンドハンドラ２４によってオンカードのアプリケーションに送られるコマンドに変換される。内容マネージャ３２は、ＸＭＬエンコーダ／デコーダ３６を使用して、オンカードのアプリケーションから内部フォーマットで受け取られたデータを解析してＸＭＬにし、その逆も同様に行う。ＸＭＬエンコーダ／デコーダは、カードから受け取られたデータを取り込み、そのデータを適切なＸＭＬノードの中に、あるいは、代替として、ＸＭＬノードを取り込み、カードアプリケーションによって要求される形態で適切なオブジェクトを構成する。例えば、カードアプリケーションは、データをＴＬＶ－ＢＥＲ（タグ、長さ、値－基本的なエンコード規則）フォーマットで格納することが可能であるが、内容マネージャ３２は、そのようなデータをさらなる処理のためにＸＭＬノードにエンコードしてＸＭＬノードにする。

【００２９】

また、内容マネージャ３２は、カードセキュリティ－アクセス権マネージャと協力して、ファイルシステム内のファイルに対するコマンドのアクセス権を決定する。これにより、データを受け取るクライアントに、データが読み取り専用であるか、更新専用であるか、またはクライアントが読み取りアクセスおよび書き込みアクセスを有するかどうかが通知される。

【００３０】

端末装置には、カードフォーマットごとに１つのＸＭＬドキュメントが関連付けられている。このドキュメントは、ファイル名で識別され、そのカードのオンカードのアプリケーションに関するすべてのファイル関連データを含む。このドキュメントは、すべての専用ファイルおよび基本ファイル、それらのファイルの中に含まれるデータ、関連するデータ範囲（つまり、最小値および最大値）、ならびにそれらのファイルに対するコマンドに関するセキュリティ条件を明らかにする。また、このドキュメントは、ファイルの中のデータに埋め込まれたビジネス規則も含み、タスクエンジンが、解釈を行い、適切なアクションを行うことができるようにしていることが可能である。

【００３１】

このＸＭＬドキュメントは、圧縮フォーマットでファイルシステム上およびカード上に格納することができる。ＸＭＬドキュメントは、カードに必要なカードアプリケーション機能をポピュレートするオンカードのプロセスにおいても使用される。

【００３２】

カード読み取り装置インフラストラクチャの一般ビジネスオブジェクトレイヤ４０は、カードセキュリティと、ＸＭＬファイルの中に表現されたセキュリティポリシーおよびセキュリティ条件を使用して、カード１０上に格納されたオブジェクトに対するオフカードのアプリケーションのアクセス権を決定するアクセス権マネージャ（「ＳＡＲＭ」）４２とを含む。

【００３３】

ＳＡＲＭは、以下の手段を提供する。すなわち、

オブジェクトに対する特定の操作が許されるかどうかを判定する手段、
オブジェクトに関するセキュリティ条件を特定し、オブジェクトに関連するポリシーおよび条件に基づいて要求される認証を行い、例えば、ＳＡＲＭが、適切な場合、グローバルＰＩＮを使用して認証を行う手段、
デジタル署名を行い、署名を検査し、暗号化し、復号化する手段
システムに挿入されたカードの妥当性を検査する、つまり、カードがブロックされているか、失効しているか等を検査する手段、および／または
カードアプリケーションの妥当性を検査する、つまり、カードアプリケーションがブロックされているか、失効しているか等を検査する手段である。

【００３４】

ＳＡＲＭ設計は、ＩＳＯ７８１６パート４、セキュリティ環境（「ＳＥ」）の実施に基づいている。ＳＥは、オブジェクトに関するセキュリティポリシーおよびセキュリティ条件の実際の実施を制御し、異なるカードアプリケーションにわたって共有されることが可能である。ＳＥは、アプリケーション特有であり、特定のビジネス環境に関する安全な機能を制御していることも可能である。

【００３５】

カードおよびカードアプリケーションの状態は、カードの使用、ならびにファイルシステム内のオブジェクトに対して許されるアクセスタイプに直接に影響を及ぼす。例えば、カードがブロックされており、したがって、ブロック解除コマンドだけが許されることが可能であり、あるいはカードがＰＩＮ検証されて、ファイルシステム内の一部のファイルに読み取り／書き込みアクセスを許し、他のファイルに読み取り専用アクセスを許し、残りのファイルにアクセスを全く許さないことが可能である。

【００３６】

許される状態、および状態の変化をもたらすアクションは、図２を形成する状態図の中で定義される。図に示す認証状態は、カードが動作する状態にある場合にだけ該当する。カードのライフサイクルの他の段階では、他のセキュリティ状態が存在することが可能である。

【００３７】

カードオブジェクトに対するアクセス権は、カード上、およびカードＸＭＬドキュメントの中で定義される。説明する実施形態において、サポートされることが可能な２つの例示的なスキームは、以下のとおりである。すなわち、
ＩＳＯ７８１６－４（２００２）、セキュリティ環境ベース、または
ユーザの諸カテゴリがオブジェクトに対する異なるアクセス権を有することを可能にするマスクベースのアクセス権である。

【００３８】

現在の状態および現在のアクセス権を提供するアクセス権状態オブジェクトが作成される。
図３は、２つのカードのトランザクションに関する制御の典型的な認証の流れを、両方のカードがユーザ認証されてからでないとトランザクションを行うことができない場合において示している。

【００３９】

図３の認証スキームの場合、カードアプリケーションオブジェクトは、要求された操作が許されることを示すアクセスモード（ＡＭ）バイトを有する。
ＡＭバイトの中で、各ビット７～１は、ゼロに設定されている場合にセキュリティ条件バイトの欠如を、あるいは１に設定されている場合にそのようなバイトの存在を同じ順序７～１で示す。ビット８が１に設定されている場合、その他のビット７～４の一部は、他のコマンド、例えば、アプリケーション特有のコマンドのために使用することができる。２００２年４月に公表されたＩＳＯ規格案７８１６パート４が、専用ファイル（ＤＦ）、基本ファイル（ＥＦ）、データオブジェクト、データテーブル、およびデータビューに関するＡＭバイトを定義している。

【００４０】
アクセスモードバイトに関連しているのが、オブジェクトがユーザ検証および外部認証を要求することを示すセキュリティ条件（ＳＣ）バイトである。
ＳＣバイトは、どのようなセキュリティ条件が満たされなければならないか、ならびに、どのようにその条件が満たされるべきかを明示するセキュリティ環境（ＳＥ）の参照を定義する。アクセスモードバイトおよびセキュリティ条件バイトは、カード上のファイルベースのオブジェクトを記述するのと同じＸＭＬドキュメントの中にエンコードされる。
【００４１】
セキュリティ環境（ＳＥ）は、ＩＳＯ７８１６－４（２００２）において定義され、完全に仕様が定められたセキュリティ機構のセットをグループ化するのに使用される。ＳＥは、実行されるべき暗号アルゴリズム、動作モード、使用されるべきキー、ならびに初期ブロック値などの要求されるさらなるデータを参照するのに使用することができる。ＳＥは、ＳＣバイトによって要求され、ＳＣバイトの中で定義されるセキュリティ機能をサポートしなければならない。ＳＥがサポートすることが要求される可能性がある機能のリストは、以下のとおりである。すなわち、
外部アーキテクチャ検査（正しいキーが外部から認証が行われていることを検証する）、
ＰＩＮ検査（正しいＰＩＮまたはユーザ認証（バイオメトリック）データがサブミットされていることを検証する）、
復号化（コマンドまたはコマンドデータを復号化する）、
暗号化（応答データを暗号化する）、
署名検証（受け取られたＭＡＣを検証する）および
署名生成（応答データについてＭＡＣを生成する）である。
【００４２】
ＳＥは、以上の機能のすべてをサポートする必要はなく、ＳＥが一部を成すアプリケーションによって使用される機能だけをサポートすればよい。各アプリケーションは、それ自体のＳＥを提供することも、カードプラットフォームによって提供されるグローバルＳＥを使用することもできる。ＳＥは、以下により詳細に説明するとおり、構成可能であることが可能である。
【００４３】
オフカードのアプリケーションが、特定のオブジェクトに対する操作を扱うことを要求する。ＳＡＲＭが、ＡＭおよびＳＣを使用してＳＥのインスタンスを生成する。ＳＥは、グローバルＰＩＮ認証が要求されると判定し、ＣａｒｄＨｏｌｄｅｒＶｅｒｉｆｉｅｒ上でｄｏＵｓｅｒＡｕｔｈｅｎｔｉｃａｔｉｏｎメソッドを呼び出すことが可能である。このクラスは、ユーザ認証を実行する。ＳＥは、次に、外部認証が要求されると判定し、第２のカードが挿入されるのを待つか、または別のメソッドを使用して別のホストと直接に認証を実行することができる。ＳＥは、これをユーザとのダイアログにおいて要求してもよい。
【００４４】
要求される場合、適切なアプリケーションを有する第２のカードが挿入される。第２のカード上のオブジェクトは、関連するＡＭバイトおよびＳＣバイトを有する。ＡＭバイトおよびＳＣバイトにより、カード所持者検証が行われることが要求されることが可能であり、その場合、別のＳＥがこのプロセスを制御する。
【００４５】
第２のカードが認証されると、外部認証が行われることが可能である。ＳＥは、内部認証コマンドを使用して第２のカードによって署名された第１のカードからのチャレンジ生成を発行する。この発行の結果が、図４に示すとおり、外部認証コマンドにおいて第１のカードに送られる。
【００４６】
相互認証スキームでは、各カードが互いを認証する。相互認証は、共有キースキームを

使用して対称的であること、または証明書スキームを使用して非対称的であることが可能である。相互認証は、例えば、コンフィギュレータ製品を使用してシステム内で構成可能である。

【００４７】

ＳＡＲＭは、スマートカードに送られるデータ、およびスマートカードから受け取られるデータの機密性および完全性を判定するのにもＡＭバイトおよびＳＣバイトを使用する。ＳＣバイトの中でセキュアメッセージングビットを設定して、データに関して暗号化および／またはメッセージ認証証明書（ＭＡＣ）が要求されるかどうかを示すことができる。関連するＳＥが、使用されるべきキーを示す。

【００４８】

コンフィギュレータを使用して、カードアプリケーションおよびオフカードアプリケーションに関するセキュリティポリシーおよびセキュリティ条件を構成することができる。代替として、既製のＸＭＬツールを使用してＸＭＬドキュメントを編集することができる。ただし、後者の方法を使用すると、より誤りが生じやすい。

【００４９】

特定ビジネスオブジェクトレイヤ内のコンポーネントは、アプリケーション特有のクラスを含む。クラスは、アプリケーションのビジネス特有の論理を実行し、ＳＡＲＭおよびカードアプリケーションファイル内容マネージャのサービスを利用してカード上のデータを取り出し、更新する。クラスは、カードによって提供されるセキュリティサービスも使用する。クラスは、以上のコンポーネントと直接にインターフェースを取ることにより、または、より普通には、一般ビジネスオブジェクトを使用してこれら他のサービスにアクセスすることによってこれを行うことができる。

【００５０】

このレイヤにおけるアプリケーションは、クラスがカードアプリケーションの適切なバージョンと対話していることを確実にすることを担う。

カードコントローラは、カードに対するアクセスを制御して、複数のオフカードのアプリケーションがカード上のアプリケーションと対話することを望む場合に同期問題が生じないことを確実にする。

【００５１】

カードアプリケーションにアクセスすることを望むクライアントはまず、このコンポーネントから適切なリソースを要求しなければならない。このコンポーネントは、以下のとおりリソースを割り振る。すなわち、

他のいずれのクライアントも、現在、リソースを使用していない場合、リソースは、要求を行っているクライアントに割り振られる（これは、適切なオブジェクトに対する参照を送り返す形で行われる）；

それ以外の場合、要求は拒否され、クライアントは、後に再び試みなければならない（将来、クライアントは、リソースが割り振られるのを待つか、またはコールバックを要求する）；

クライアントが、トランザクションを完了した時点で、リソースを解放し、その他のクライアントがリソースに対するアクセスを得ることができるようにする。

【００５２】

カードコントローラは、読み取り装置抽象インターフェースとインターフェースをとり、いつカードが挿入されたか、またはいつカードが取り外されたかを特定し、以下のとおり適切なアクションを行う。すなわち、

カード挿入の場合、このコンポーネントは、カード上のアプリケーションを特定し、登録済みのオフカードのアプリケーションに通知を行い、

カード取り外しの場合、コンポーネントは、すべての登録済みのクライアントに通知を行う。次のアクションを行うのはクライアント次第である（カードは、トランザクションの最中に取り外されている可能性がある）。

【００５３】

ＸＭＬを使用してオフカードのファイルシステムおよびコマンド構造、ならびにオンカードのファイルシステムおよびコマンド構造を構成することは、多くの利点を有する。第１に、オフカードのシステムが、オンカードのシステムと完全に同期した状態に保たれ、この２つの統合がより迅速で、より容易になる。

【００５４】

Ｗ３Ｃ組織によって提案される業界インターネット標準であるＸＭＬは、多数の既製のツールおよびパーサを生み出した。それらのツールおよびパーサは、ＸＭＬファイルを作成し、構成するのに容易に使用することができる。ＸＭＬは、システム間統合に関する事実上のインターネット標準になってきている。ＸＭＬを使用することにより、本発明のシステムは、ＸＭＬをインターフェースとして使用するシステムに容易に組み込むことができる。これにより、既存のシステムが、迅速に、安価な形でスマートカード対応になることが可能になる。

【００５５】

システムは、非常に構成しやすく、単にＸＭＬファイルを変更することによってファイル、システム属性、およびセキュリティポリシーを変更することを可能にする。再設計またはコーディングを全く行う必要がない。

【００５６】

システムは、ビジネスアナリストおよびセキュリティアナリストが、ビジネスデータ、ならびにセキュリティを支配する規則を構成してシステムに組み込むことができるようにする。それらの規則は、オンカードのアプリケーションを構成するのに構築時に使用され、実行時におけるオンカードのアプリケーションの挙動とオフカードのアプリケーションの挙動との両方に関して使用される。

【００５７】

本発明のシステムによって提供されるソリューションの重要な要素は、アプリケーション管理の分野に属する。オフカードの環境に緊密に関連する極めて構成しやすいスマートカードアプリケーション環境を提供することにより、現場で、つまり、カード所持者が自身のカードをカードアプリケーションと対話することができる端末装置に提出した際に、ファイルシステムをアップグレードする可能性が許される。

【００５８】

そのようなアプリケーション管理の要件は、ユーザが以下を行うことができることである。すなわち、

新たなファイルシステムを作成するためにも、既存のファイルシステムを変更するためにもファイルシステムおよびファイル内容を容易に構成することができ、

新たなセキュリティ条件を作成するためにも、既存のセキュリティ条件を変更するためにもファイル上のセキュリティ条件を容易に構成することができ、

直接に、またはスクリプト処理（以下を参照）を使用してカード上のファイルシステムを作成することができ、

現場で、ファイルシステム構成を変更して、それにより、カード上のファイルシステムを変更できるようにすることができることである。

【００５９】

そのようなシステムの１つの前提条件は、異なるバージョンのファイルシステムを認識し、識別することができることである。前述したコンフィギュレータを使用してスマートカード上のファイルシステムを構成し、ＸＭＬシードドキュメントがポピュレートされるようにすることができる。カード発行者／スキーム所有者が、ＸＭＬファイルのベースラインを定める、そのファイルに所与のバージョン番号を与えることが許される。コンフィギュレータを使用してＸＭＬファイルを変更して、ファイルを追加する、削除する、または変更することができる。これにより、新しいバージョンのＸＭＬファイルが生成されることになる。

【００６０】

アプリケーション管理プロセスは、「端末装置主導型アップグレード」（「プロセス１

」）および「スクリプト処理」（「プロセス２」）の２つの処理を包含する。これらの処理を図５に示している。

【００６１】

図５で「プロセス１」として示す端末装置主導型アップグレードプロセスでは、顧客は、ファイルシステムがポピュレートされたカード５０を所有し、カード５０が対話することが意図されている端末装置５２は、そのファイルシステムの構造を記述する関連するＸＭＬファイルを含む。カード発行者／スキーム所有者は、バックオフィスにおいてカード上のファイル構造を変更することを所望する場合、コンフィギュレータツール５４を使用して所望の変更を行う。ファイルシステムに対して行われた変更を含む新しいバージョンのＸＭＬファイルが生成される。この新しいＸＭＬファイルは、構成バンドルの中でカード発行者／スキーム所有者のデータベース５６に格納され、構成バンドルは、ＸＭＬファイルのその新しいバージョンとすべての以前のバージョンをともに含む。

【００６２】

カード発行者／スキーム所有者は、次に、ソフトウェア配信システムを使用して、ＸＭＬファイルをバックオフィスからすべての端末装置５２に配信する。端末装置５２において、ＸＭＬファイルの新しいバージョンが古いバージョンとともに格納される。

【００６３】

カードが挿入された際、カード５０上に保持されるファイルシステムのバージョンを示すバージョン番号がカードから読み取られる。カードが、古いバージョンを保持している場合、端末装置５２におけるＸＭＬファイルのその古いバージョンが、カード５０上に保持されるデータを使用してポピュレートされる。次に、カードアップグレードモジュールが実行されて、ファイルの古いバージョンを新しいバージョンに変換し、古いＸＭＬファイルの中のカードファイルからのデータが、新しいＸＭＬドキュメントに移される。次に、新しいＸＭＬドキュメントが、カードに再び書き込まれ、ファイルシステムのバージョンが新しいバージョンにアップグレードされる。

【００６４】

カードファイルシステムが新しいバージョンにアップグレードされると、端末装置５２は、新しいＸＭＬドキュメントを使用してカードと対話する。

前述した端末装置主導型アップグレードの代わりに、アプリケーション管理は、スクリプト処理（図５の「プロセス２」）を使用して実施することもできる。

【００６５】

この場合も、カード発行者／スキーム所有者は、バックオフィスにおいてコンフィギュレータツールを使用して、要求される変更を行い、新しいＸＭＬファイルをバックオフィスデータベースにおいて構成バンドルの中に格納する。

【００６６】

次に、コンフィギュレータツールを使用して、以前のＸＭＬバージョンから新しいバージョンへの変化をカバーするＡＰＤＵリストが生成される。カード発行者／スキーム所有者は、元のファイルシステムを試験カード上にロードし、ファイルの中のＡＰＤＵリストをキャプチャすることによってこれを行う。次に、カード発行者／スキーム所有者は、第２のＸＭＬファイルを試験カードに適用し、ファイルの中の送られたＡＰＤＵをキャプチャすることによって試験カード上のファイルシステムを更新する。コンフィギュレータは、２つのＡＰＤＵリストの違いを特定し、新しいＡＰＤＵを含む第３のＡＰＤＵリストを生成する。リストは、スキームに関するセキュリティアーキテクチャの中で定義されたキーおよびアルゴリズムを使用して暗号化される。

【００６７】

スクリプトおよび新しいＸＭＬファイルは、端末装置５２がオンラインになった時点で、またはソフトウェア配信プロセス中に各端末装置５２に配信される。

端末装置において、顧客のカード５０が挿入された際、カード上のファイルシステムのバージョンが、ＡＰＤＵリストを生成したバージョンと比較される。カードバージョンの方が古い場合、カードアップグレードモジュールが、スクリプト処理を実行してカード５

０をアップグレードする。アップグレードが完了すると、端末装置５２は、その新しいＸＭＬファイルを使用して、対応するバージョン番号を有するカード５２上のファイルシステムと対話する。

【００６８】

したがって、要約すると、端末装置５２において、カード統合モジュールは、構成された際にファイルに対するＡＰＤＵリストを生成し、カード５０上のファイルシステムのバージョンを端末装置５２上のバージョンと照合し、カード上のバージョンの方が端末装置におけるバージョンより古い場合、アップグレードが行われるべきであると判定する。

【００６９】

端末装置のカードアップグレードモジュールが、１つのポピュレートされたＸＭＬファイルを新しいバージョンに変換し、カードをＸＭＬファイルの新しいバージョンからポピュレートし、ＡＰＤＵリストが提供されている場合、スクリプト処理を実行する。

【００７０】

カード発行者／スキーム所有者のバックオフィスにおいて使用されるコンフィギュレータツールは、ファイルシステムを変更することを可能にし、ゼロからポピュレートするため、または違いに関してポピュレートするために試験カードを使用してＡＰＤＵリストを生成することを可能にし、新しいスキームの試験などの特殊なケースのために１つのＡＰＤＵリストを生成するようにＡＰＤＵリスト間の違いを扱う。

【００７１】

システムは、前述したセキュリティ環境が、ファイル参照を使用することも可能にする。キーとＰＩＮ（個人識別番号）をＳＥに関連するメモリの中にハードコードする代わりに、キー、ならびにＰＩＮなどのその他のセキュリティオブジェクトをファイルシステム内のファイルの中に格納することもできる。

【００７２】

それらのファイルは、端末装置によって要求されたセキュリティ操作の制御を扱うカード上のＳＥ内から参照することができる。同様に、使用されるアルゴリズムタイプなどの任意のセキュリティ制御パラメータも、ファイルの中に格納し、ＳＥから参照することができる。操作が行われるオブジェクトのヘッダ内のセキュリティバイト（ＡＤＦ、ＤＦ、およびＥＦ）は、ＰＩＮやキーなどの要求されるセキュリティオブジェクトに対する参照を示す。

【００７３】

図６は、例として、更新のためにセキュリティ属性が、キー参照Ａを使用して外部認証に設定され、ＰＩＮ参照Ｂを使用してＰＩＮに設定されている基本ファイルＡを更新するためにＳＥがセキュリティを扱うのに要求されるファイルを示している。

【００７４】

オフカードのアプリケーションは、基本ファイルＡを更新することを望む場合、まず、チャレンジ獲得を発行し、次に、外部認証を発行しなければならない。外部認証コマンドは、ＡＰＤＵコマンドのＰ２の中のキーＡを参照する。オフカードアプリケーションが、次に、ＰＩＮ参照Ｂ（ＡＰＤＵコマンドのＰ２の中の）および候補ＰＩＮを使用してＰＩＮ検証を発行することにより、ＰＩＮを使用してカード所持者検証を行う。オンカードのＳＥが、キーおよびＰＩＮに関する参照を使用して、実際の値を含む適切なセキュリティファイルにアクセスする。このファイルは、アクセスが許可される条件を規定している関連するセキュリティバイトも有する。

【００７５】

十分なセキュリティを備えた管理者が、カード上のオブジェクトに対する各操作に関するセキュリティを構成することができる。これにより、オブジェクトがカード上で作成される際に、オブジェクトのセキュリティヘッダの中でセキュリティバイトが設定される。管理者は、ファイルに対する操作に関してＰＩＮやキーなどのセキュリティオブジェクト参照を選択することができ、そのセキュリティオブジェクトを含むセキュリティファイルに関するセキュリティを構成することができる。

【００７６】
前述したとおり、本発明によって使用されるＳＥは、それ自体、構成可能であることが可能である。例えば、構成可能なＳＥにより、以下のセキュリティ属性の調整ができるようになることが可能である。すなわち、
－特定のファイルにアクセスするためにいずれのキー（キーＩＤ）が使用されるかを定義する、
－特定のキーに関連するキー値およびその他の属性を定義する、
－認証、安全なメッセージング等のためにいずれのアルゴリズムが使用されるかを定義する、
－認証アルゴリズムを調整する（例えば、リテイルメッセージ認証コード（ＭＡＣ）に対する完全な暗号ブロック連鎖（ＣＢＣ）ＭＡＣ）、
－カード所有者認証のためにどのようなカード所有者認証（ＣＨＶ）データ（ファイルＩＤおよびデータの値）が使用されるかを定義する、
－ファイルが認証のための「グローバル」値を使用すべきか、またはローカル値を使用すべきかを規定することができる、
－既定で、セキュリティ機構は、作成された直後に有効になってはならず、別のファイル（ＥＦまたはＤＦ）が選択された後、すなわち、作成されたファイルがもはや現在、選択されたファイルではなくなった後に初めて有効になる。
【００７７】
前述した基本的なシステムは、以下のとおり、２つの領域で拡張することができる。
第１の拡張は、安全なメモリカードのような入門レベルのセキュリティの低いカードから基本的な記憶機能を有するあらゆるスマートカードまであらゆるタイプのカードに関する超特急のアプリケーション作成－管理体制としてのシステムの使用に関する。言い換えれば、オンカードの実行時のアプリケーション環境が選択肢となることが可能である。
【００７８】
拡張は、それぞれの現実のカードタイプつき１つの「仮想カード」データモデルが構成される、前述したコンフィギュレータツールに選択肢として導入される追加の段階から成る。これは、複数の目標マイクロコントローラを有する高レベル言語コンパイラと概念が同様である。
【００７９】
拡張されたコンフィギュレータツールにより、ユーザは、適切なＸＭＬドキュメントバージョンを生成して、あらゆるカード上でアプリケーションを作成し、変更することができるようになる。本明細書で説明する端末装置常駐の態様（「ミドルウェア」）は、ＸＭＬドキュメントバージョンを使用してカード上のデータ（どのように保持されているかに関わらず）をＸＭＬドキュメントの中で記述される構造にマップする。端末装置ミドルウェアは、ＸＭＬドキュメントを使用して、データに対する操作に関連するセキュリティを決定し、その要件が満たされてからでないと操作を実行できない、例えば、ＰＩＮが要求されてからでないと「読み取り」操作を実行できないことを確実にする。ミドルウェアは、アプリケーション特有のセキュリティ環境も使用して、カードのセキュリティを扱い、存在する場合、オンカードの機能を利用する。
【００８０】
ミドルウェアは、カード特有の汎用コマンドハンドラも含み、このハンドラは、適切なＡＰＤＵコマンドを生成してカードと通信する。
前述したソフトウェアの基本的な拡張されていないバージョンが使用されたときカード上で起こるのは、実行時コードのセットアップまたは構成は、カードベースのコードにより、端末装置から着信するコマンドが解釈されてアプリケーションがリアルタイムで実行されることを意味するが、カード内部では、実際、カードソフトウェアが、各コマンドに関連するデータをカードのそれ自体のメモリにマップすることである。
【００８１】
拡張は、その入力をミドルウェアの出力として変換する別の「変換」プロセスを追加し

て、何らかの他のカード上にロードされるべき実際のメモリイメージを構成する。このイメージは、仮想カードを表すが、基本的なストレージを有するあらゆるカードにロードすることができる。イメージは、ＴＬＶデータを効率的な（例えば、）線形バイナリファイルに再フォーマットすることによって構築される。固有の暗号化を伴う圧縮を使用することができる。このプロセスは、「フラッシュメモリスティック」などの全くスマートカードの形態でないセキュリティで保護されていない取り外し可能な記憶デバイスに適用することができ、したがって、マップすることができる。データは、記憶デバイスから再び読み取られる際、ＸＭＬによって記述された構造をポピュレートするように再フォーマットされる。

【００８２】

データイメージは、妨げるものなしに、つまり、暗号化されずに格納された場合、解釈され／リバースエンジニアリングを受けてサービスがハッキングされる可能性がある。しかし、ミドルウェアの追加の変換プロセスが、現実のカードそれ自体のセキュリティ能力の知識を有する場合、その能力を「仮想カード」にトランスペアレントに（すなわち、アプリケーションの可視のセキュリティ条件が内部でエミュレートされる一括セキュリティ機能として端末装置によって使用されて）、またはトランスペアレントにではなく、現実のセキュリティカード機能に対する仮想カードセキュリティ機能の「１対１」マッピングとしてアクティブにすることができる。これは、異なるカード機能セットを伴って個々の事例で異なり、そのことが、ミドルウェアの「仮想カードコンパイラ」のために選択できる１組の「目標の」現実のカードプラットフォームが存在する理由である。

【００８３】

いくつかのオプションをセキュリティに利用することができる。例えば、カードが、セキュリティ機能を全く有さない最も単純なメモリタイプのものである場合、すべてのメモリイメージが、カードに格納される／カードから取り出される前に、端末装置によってオンザフライで暗号化／復号化される、またはＭＡＣされることが可能であり、そのためのキーをアプリケーション特有の「マスタキー」およびカードの固有の通し番号から導出することができる。

【００８４】

最低レベルのセキュリティのため、マスタキーは、端末装置の実行時アプリケーションの中に格納することができるが、より高いセキュリティのためには、マスタキーは、端末装置内部に存在し、カード特有のキーをオンザフライで計算する安全なアクセスモジュール（ＳＡＭ）の中に格納することができる。このようにすると、端末装置のメモリの中で実行されているカード特有のキーが見られた場合でも、その１つのカードだけがハッキングされ、スキームはハッキングされない。セキュリティを次第に高めるのに使用することができるいくつかの戦略が存在するが、セキュリティおよびパフォーマンスの点で「最高条件」まで使用される特定のカードの制約に基づいてのみ使用可能であり、「最高条件」とは、カードが、完全な対応する構成可能なアプリケーションを実行する元のモデルであることに留意されたい。

【００８５】

この手法の重要な利点は、同じアプリケーションをコンフィギュレータツールを使用して超特急で構築し、管理することができるが、その後、カードが少なくとも単純なデータ記憶機能を有するという条件付きで、任意のカードタイプにコンパイルすることができることである。これを行う能力は、実際の目標のカード自体の限界により、容量、パフォーマンス、およびセキュリティが制限されるだけである。

【００８６】

以上に概要を述べた基本的なシステムのさらなる実施形態は、カードまたは端末装置の回収の必要なしに、そのカード上の現在のデータを乱すことなしに、現場で既に動作しているカード上および端末装置上の既存のアプリケーションの変更および／または拡張、および新しいアプリケーションの導入に関する。

【００８７】

現在のスマートカードマルチアプリケーション・オペレーティングシステム（ＭＡＯＳ）（Ｊａｖａ（登録商標）ＣａｒｄおよびＭＵＬＴＯＳによって代表される）は、６０年代のマルチユーザのメインフレームコンピュータのために最初に発明されたままに仮想マシンモデルを使用する。このＭＡＯＳを有するスマートカードの特定の利点は、ＭＡＯＳが、アプリケーションを分離する「ファイアウォール」を有すると主張できることである。ただし、その結果、そのアプリケーションは、それぞれがそれ自体のコードおよびデータを有するプログラムでなければならず、したがって、データを共用することは、個々のすべてのアプリケーションに、その他のアプリケーションの知識が書き込まれ、個々のアプリケーション内部に通信プロトコルが組み込まれることを要する。これは、いくつかのアプリケーションによって共通のデータが必要とされる場合、これは極めて非効率であるが、そのための標準の手法は存在しない。

【００８８】

ＭＡＯＳでは、カードオペレーティングシステムおよび各アプリケーションがポリシーを実施するのに依拠する「役割保持者およびキー保持者」をアプリケーション管理およびアプリケーション権限が追跡することができない。しかし、アプリケーションが変更される必要がある場合、アプリケーションは、ＭＡＯＳにより、コードとデータの分離のない「原子」エンティティとして扱われる。つまり、アプリケーションのすべてが削除され、新しいバージョンで置き換えられる。したがって、すべてのカードデータが失われる。これは、カード／アプリケーション管理システムが、必要に応じてデータを保存するためにどのようにか便宜を図らなければならないということで、カード／アプリケーション管理システムに負担を課し、したがって、各カード／アプリケーション管理システムが、非常に「アプリケーションデータ特有」になる。

【００８９】

本発明には、特定のセキュアファイルシステム管理コマンドを使用して構成された「オンカードの」アプリケーション環境の使用が既に関与しているので、オンカードのモデルとオフカードのモデルの両方を表す特定のバージョンを構築し、次に、それらのバージョンの正確な違いを計算し、既に構成済みのデータの挙動またはカードの挙動を全く削除する必要なしに、現場で端末装置およびカードをアップグレードすることが可能である。

【００９０】

違いが計算された後、新しいアプリケーション挙動、ならびに必要とされる可能性があるセキュリティ要素を導入する新しいＡＰＤＵのスクリプトが作成される。そのセキュリティ要素は、キー、およびその他のセンシティブなデータを含む可能性があるので、妨げるものなしにカードベースに伝送することはできない。アップグレードを要するカードが、このスキームにおいて端末装置に導入された場合はいつでも、安全なメッセージングチャネルを中央サーバにセットアップすることができるが、これは、端末装置が、中央サーバに対して常時、「オンライン」であることを要するという欠点を有する。カードおよび端末装置のアップグレードを実行するより洗練された費用効率の高いやり方は、以下のとおりである。

【００９１】

差分ファイルが、カード用に１つ、端末装置用に１つ、前述したとおり作成される。ＸＭＬバージョンが、アップグレードされたカードから期待される新しいデータモデルおよび新しい挙動で端末装置アプリケーション環境をアップグレードする。（すべてのセキュリティ要素はカードベースであるため、このファイルは、おそらく、端末装置部分が破損している場合、「サービス拒否」攻撃を防止するため以外では、保護される必要がなく、標準の完全性検査を使用することができる。）

ただし、カード用のファイルは、ＡＰＤＵの安全なスクリプトから成り、このスクリプトが、カードにロードされる。この安全なスクリプトは、「フレーム」と呼ばれ、暗号化とＭＡＣの両方によって保護される。フレームがロードされた際、カードは、ＭＡＣを介してフレームの完全性を検査し、合格である場合、カードは、ペイロードを復号化してＡＰＤＵのスクリプトを明らかにする。カードは、安全な端末装置をエミュレートする「ス

クリプトエンジン」を有し、したがって、カード自体の上でコマンドを実行する。このようにして、特定のコマンドが、既存のアプリケーションファイルを正確に変更することができ、新しいデータおよびセキュリティ規則を導入することもできる。

【００９２】

フレームは、最初、業界標準のハードウェアセキュリティモジュール（ＨＳＭ）を使用して構築することができ、ＨＳＭでは、すべての重要な材料が安全な形で格納される。フレームを構築するのに使用されるソフトウェアは、カード固有であることが可能な、カードのアプリケーションや、ロードされる、またはロードされることを要する各キーのＩＤなどの各カードのプロファイルを含むデータベースを参照する。ＨＳＭは、次に、フレームコンストラクタレイヤによって配置されたプレースホルダの中に実際の重要な材料を挿入した後、その材料を暗号化し、署名することによってフレームをポピュレートし、端末装置に配信する準備ができて、端末装置が、次に、カードアプリケーション環境の構成のバージョン番号に基づいてカードをアップグレードする。

【図面の簡単な説明】

【００９３】

【図１】本発明のシステムのコンポーネント・アーキテクチャの概要を示す図である。

【図２】カードおよびカードアプリケーションの動作状態を示す概略図である。

【図３】あるエンティティの別のエンティティによる認証のための制御の典型的な流れを示す図である。

【図４】上記の外部認証の流れを示す図である。

【図５】本発明のシステムにおけるアプリケーション管理プロセスを示す図である。

【図６】本発明のシステム内のファイルを更新するためのセキュリティを扱う際に使用されるセキュリティファイルを示す概略図である。

【図１】

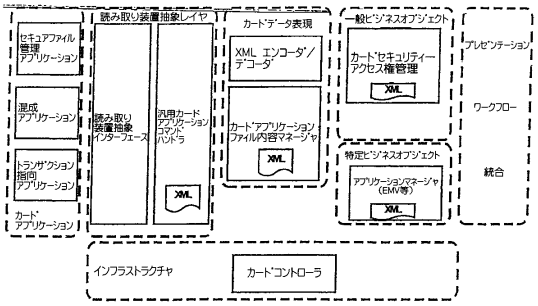

図1 －コンポーネントアーキテクチャ

【図２】

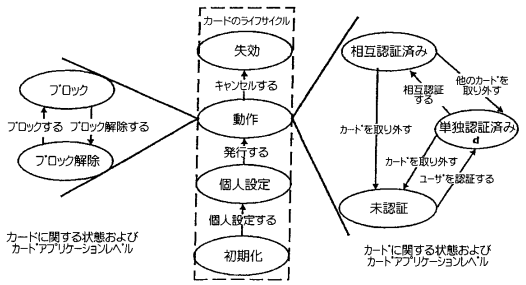

図2 －カード／カード・アプリケーション動作状態

【図３】

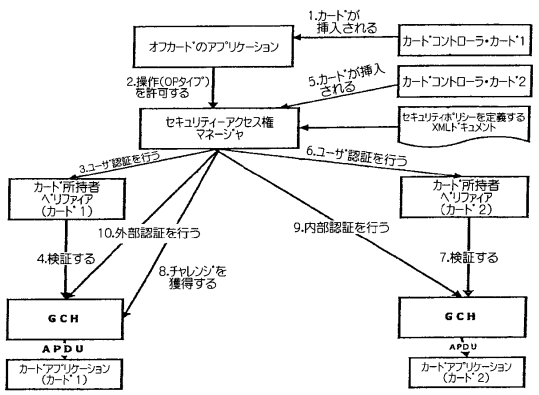

図3 －制御の典型的な認証フロー

【図４】

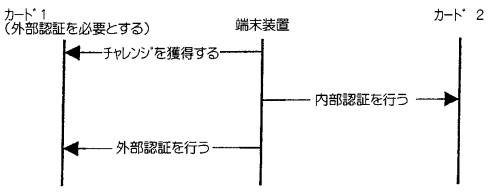

図4 －フローの外部認証

【図５】

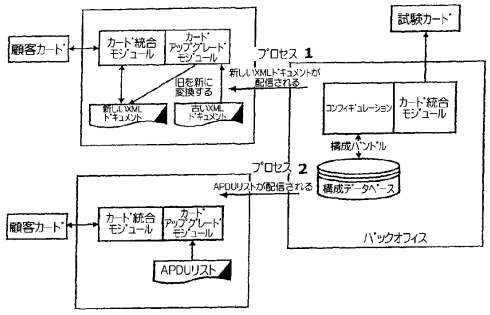

図５－アプリケーション管理プロセス

【図６】

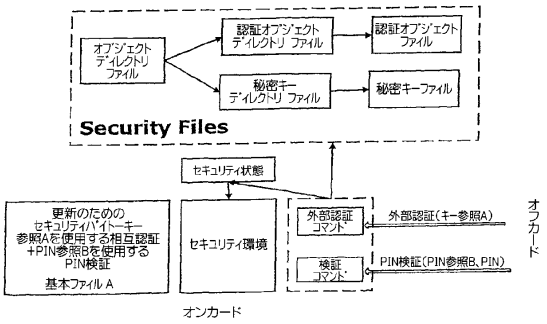

図６－ファイルを更新するためのセキュリティの扱い

フロントページの続き

(74)代理人 100120558
弁理士 住吉 勝彦
(72)発明者 ブレズリン，アンソニー
イギリス国スコットランド ジー７５・８エフダブリュー，イースト・キルブリッジ，ストラスナーリン・アベニュー ２１
(72)発明者 ピータース，マイケル
イギリス国スコットランド ジー６６・４イーイー，グラスゴー，レンジー，ボグヘッド・ロード ３６
(72)発明者 ホックフィールド，バリー・シン
イギリス国スコットランド ジー４６・６アールビー，ギフノック，ダルサーフ・クレセント ２１

審査官 田川 泰宏

(56)参考文献 欧州特許出願公開第０１０８５３９５（ＥＰ，Ａ１）
樋浦 裕二，ＩＣカードの最新技術と利用動向 第３回，ＢＵＳＩＮＥＳＳ ＣＯＭＭＵＮＩＣＡＴＩＯＮ，日本，株式会社ビジネスコミュニケーション社，２００１年 ６月 １日，第38巻，第6号

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 12/00
G06F 21/24
G06K 17/00
G06K 19/07