(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4857283号
(P4857283)
(45)発行日　平成24年1月18日(2012.1.18)　(24)登録日　平成23年11月4日(2011.11.4)

(51)Int.Cl.　　F I
G06F 21/24 (2006.01)　　G06F 12/14 520C
H04L 9/32 (2006.01)　　G06F 12/14 530E
　　H04L 9/00 673E

請求項の数 24　(全 40 頁)

(21)出願番号	特願2007-548491 (P2007-548491)
(86)(22)出願日	平成17年12月21日 (2005.12.21)
(65)公表番号	特表2008-524755 (P2008-524755A)
(43)公表日	平成20年7月10日 (2008.7.10)
(86)国際出願番号	PCT/US2005/046689
(87)国際公開番号	WO2006/069274
(87)国際公開日	平成18年6月29日 (2006.6.29)
審査請求日	平成20年12月9日 (2008.12.9)
(31)優先権主張番号	60/638,804
(32)優先日	平成16年12月21日 (2004.12.21)
(33)優先権主張国	米国(US)
(31)優先権主張番号	11/314,052
(32)優先日	平成17年12月20日 (2005.12.20)
(33)優先権主張国	米国(US)

(73)特許権者　507208288
サンディスク　コーポレーション
アメリカ合衆国　９５０３５　カリフォルニア州，ミルピタス，マッカーシー　ブールバード　６０１
(74)代理人　100083895
弁理士　伊藤　茂
(72)発明者　ヨガンド－クーロン，ファブリス
アメリカ合衆国　９４０７０　カリフォルニア州，サン　カルロス，バックランド　アヴェニュー　８５５
(72)発明者　ホルツマン，マイケル
アメリカ合衆国　９５０１４　カリフォルニア州，クパティーノ，バーンハート　プレイス　７６０２

最終頁に続く

(54)【発明の名称】パーティション化による多目的コンテンツ制御

(57)【特許請求の範囲】
【請求項１】
アクセス制御を行うための方法であって、
パーティションを備える不揮発性メモリを有し、アカウントパーミッションを有するアカウントを入れているストレージデバイスによって実行するステップであって、
前記パーティションへのアクセス要求を受信するステップであって、前記要求はセッションＩＤを含み、前記セッションＩＤは前記アカウントに対する認証のアカウントパーミッションを伴い、前記ストレージデバイスは前記アカウントパーミッションを前記エンティティの認証に先立って保存する、ステップと、
前記ストレージデバイス内で前記セッションＩＤと関連付けられた前記アカウントパーミッションを検索するために、前記要求に含まれた前記セッションＩＤを使用するステップと、
前記アカウントパーミッションが、前記パーティションへのアクセス要求を認可するか否かを決定するステップと、及び
前記アカウントパーミッションが、前記パーティションへのアクセス要求を認可する場合、前記パーティションへのアクセス要求を許可するステップと、
を含む方法。
【請求項２】
前記メモリは付加的なパーティションを有し、前記方法は更に、認証された複数のエンティティが前記付加的なパーティション内のデータへアクセスすることを許可するステッ

プを有する請求項１に記載の方法。
【請求項３】
前記パーティションへのアクセスタイプは、前記エンティティを前記アカウントに対して認証するために使用するいずれの認証からも独立している請求項１に記載の方法。
【請求項４】
複数のアカウントが、前記パーティションへのアクセスタイプの一種として認可され得る請求項１に記載の方法。
【請求項５】
前記アカウントは前記ストレージデバイスのメモリコンポーネント内に保存され、認証を行う間に、前記アカウントに関連する情報が前記ストレージデバイスのコントローラによって前記メモリコンポーネントからフェッチされる請求項１に記載の方法。
【請求項６】
前記パーティションは連続したアドレス領域を有する請求項１に記載の方法。
【請求項７】
アクセス制御を行うための方法であって、
パーティションを備える不揮発性メモリを有し、アカウントパーミッションを有するアカウントを入れているストレージデバイスによって実行するステップ、即ち：
前記ストレージシステム内の前記不揮発性メモリにおける前記パーティションに対して読出及び書込アクセスのいずれかを許可するステップであって、前記読出及び書込アクセスのいずれか一方は認証なしに許可されるステップと、
前記アカウントに対するエンティティの認証が行われた場合のみ、前記パーティションへの読出及び書込アクセスのいずれか他方が許可され、前記ストレージデバイスは前記エンティティの認証に先立って前記アカウントパーミッションを保存し、当該認証においてセッションＩＤが前記エンティティに対して提供され、前記アカウントパーミッションと関連付けられるステップであって、
前記パーティションへの読出及び書込アクセスのいずれか他方を実行するための、前記セッションＩＤを有する要求を受信するステップと、
前記ストレージデバイス内で前記セッションＩＤと関連付けられた前記アカウントパーミッションを検索するために、前記要求に含まれた前記セッションＩＤを使用するステップと、及び
前記アカウントパーミッションが、前記パーティションへの読出及び書込アクセスのいずれか他方を認可することを決定するステップと、
を含む方法。
【請求項８】
前記パーティションへのアクセスは少なくとも１つの他のアカウントに基づいて制御される請求項７に記載の方法。
【請求項９】
前記メモリは、少なくとも１つの付加的パーティションを有する請求項７に記載の方法。
【請求項１０】
前記アカウントは、アクセス制御記録を有する請求項７に記載の方法。
【請求項１１】
前記アカウントは前記ストレージデバイスのメモリコンポーネント内に保存され、認証を行う間に、前記アカウントに関連する情報が前記ストレージデバイスのコントローラによって前記メモリコンポーネントからフェッチされる請求項７に記載の方法。
【請求項１２】
前記パーティションは連続したアドレス領域を有する請求項７に記載の方法。
【請求項１３】
パーティションを備え、アカウントパーミッションを有するアカウントを入れている不揮発性メモリと、

前記不揮発性メモリと通信するコントローラと
を備えたストレージデバイスであって、
前記コントローラは
前記パーティションへのアクセス要求を受信するステップであって、前記要求はセッションＩＤを含み、前記セッションＩＤはエンティティの前記アカウントに対する認証のアカウントパーミッションを伴い、前記ストレージデバイスは前記アカウントパーミッションを前記エンティティの認証に先立って保存する、ステップと、
前記ストレージデバイス内で前記セッションＩＤと関連付けられた前記アカウントパーミッションを検索するために、前記要求に含まれた前記セッションＩＤを使用するステップと、
前記アカウントパーミッションが、前記パーティションへのアクセス要求を認可するか否かを決定するステップと、
前記アカウントパーミッションが、前記パーティションへのアクセス要求を認可する場合には、前記パーティションへのアクセス要求を許可するステップと、
を実行するようにされている
ストレージデバイス。
【請求項１４】
前記メモリは追加のパーティションを備え、前記コントローラは更に、認証されたエンティティが前記追加のパーティションの中のデータにアクセスすることを許諾するようにされた、請求項１３に記載のストレージデバイス。
【請求項１５】
前記パーティションへのアクセスの一種は、前記エンティティの前記アカウントに対する認証に用いられたいずれの照明からも独立である、請求項１３に記載のストレージデバイス。
【請求項１６】
複数のアカウントが前記パーティションへの一種類のアクセスを承認され得る、請求項１３に記載のストレージデバイス。
【請求項１７】
認証を行う間に、前記アカウントに関連する情報が前記ストレージデバイスのコントローラによって前記メモリコンポーネントからフェッチされる、請求項１３に記載のストレージデバイス。
【請求項１８】
前記パーティションは連続した範囲のアドレスを有する、請求項１３に記載のストレージデバイス。
【請求項１９】
パーティションを備え、アカウントパーミッションを有するアカウントを入れている不揮発性メモリと、
前記不揮発性メモリと通信するコントローラと
を備えたストレージデバイスであって、
前記コントローラは
前記不揮発性メモリにおける前記パーティションに対して読出及び書込アクセスのいずれかを許可するステップであって、前記読出及び書込アクセスのいずれか一方は認証なしに許可されるステップと、
前記アカウントに対するエンティティの認証が行われた場合のみ、前記パーティションへの読出及び書込アクセスのいずれか他方が許可され、前記ストレージデバイスは前記エンティティの認証に先立って前記アカウントパーミッションを保存し、当該認証においてセッションＩＤが前記エンティティに対して提供され、前記アカウントパーミッションと関連付けられるステップであって、
前記パーティションへの読出及び書込アクセスのいずれか他方を実行するための、前記セッションＩＤを有する要求を受信するステップと、

前記ストレージデバイス内で前記セッションＩＤと関連付けられた前記アカウントパーミッションを検索するために、前記要求に含まれた前記セッションＩＤを使用するステップと、及び

前記アカウントパーミッションが、前記パーティションへの読出及び書込アクセスのいずれか他方を認可することを決定するステップと、

を実行するようにされている

ストレージデバイス。

【請求項２０】

前記パーティションへのアクセスは少なくとも１つの他のアカウントに基づいて制御される請求項１９に記載のストレージデバイス。

【請求項２１】

前記不揮発性メモリは、少なくとも１つの付加的パーティションを有する請求項１９に記載のストレージデバイス。

【請求項２２】

前記アカウントは、アクセス制御記録を有する請求項１９に記載のストレージデバイス。

【請求項２３】

認証を行う間に、前記アカウントに関連する情報が前記コントローラによって前記メモリコンポーネントからフェッチされる請求項１９に記載のストレージデバイス。

【請求項２４】

前記パーティションは連続した範囲のアドレスを有する、請求項１９に記載のストレージデバイス。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、概してメモリシステムに関し、特に多目的なコンテンツ制御の機能を備えたメモリシステムに関する。

【背景技術】

【０００２】

コンピュータデバイス市場は、さらなるデータ交換を行うことにより平均収益を増加させるべく、モバイルストレージデバイスにコンテンツストレージを組み込む方向に発展している。これは、コンピュータデバイスで使用する際に、モバイルストレージメディアのコンテンツを保護する必要があることを意味する。コンテンツは貴重なデータを含むが、これはストレージデバイスの製造者または販売者以外の者が所有するデータであるかもしれない。

【０００３】

米国特許第６４５７１２６号には、暗号機能を備えた一つの種類のストレージデバイスが記載されている。しかし、当該デバイスにより提供される機能はかなり制限（ｑｕｉｅｔ　ｌｉｍｉｔｅｄ）されている。したがって、メモリシステムに、より多目的にコンテンツを制御する機能を提供することが望ましい。

【発明の開示】

【０００４】

モバイルストレージメディア内のコンテンツの保護には、権限のあるユーザまたはアプリケーションのみがメディアに保存されたデータを暗号化するために使用された鍵へのアクセスを有するように、メディア内のデータの暗号化を行うことが可能である。幾つかの従来のシステムでは、データの暗号化および復号に使用された鍵がモバイルストレージメディアの外部のデバイスに保存されている。こうした環境では、コンテンツの所有権を有する企業または個人は、メディア内のコンテンツ利用に関してそれ程管理していない場合がある。メディア内のデータを暗号化するために使用された鍵がメディアの外部に存在するので、当該鍵はコンテンツ所有者による管理を前提としない方法で、あるデバイスから

別のデバイスへ渡される可能性がある。本発明の機能の一つによれば、暗号・復号鍵がメディア自体に保存されており、外部デバイスにとって実質的にアクセス不可能であれば、所有権の所有者は、メディア内のコンテンツへのアクセスを管理し易い。

【０００５】

原則的にメディア外部からの鍵へのアクセスを不可能にすることにより、当該機能は保護されたコンテンツに移植性（ｐｏｒｔａｂｉｌｉｔｙ）を与える。このように、そうした鍵で暗号化された保護コンテンツを含むストレージデバイスは、鍵へのアクセスに関して排他的制御を有しているので、機密保護違反の危険性を有することなく、様々なホストデバイスによるアクセスに使用することが可能となる。適切な証明書をもつホストデバイスのみが、鍵へアクセスできる。

【０００６】

モバイルストレージメディアに保存されたコンテンツの商業価値を高めるために、コンテンツ中の所有権の所有者が、当該コンテンツへのアクセスに関して、様々なエンティティ（ｅｎｔｉｔｙ）に様々なパーミッション（ｐｅｒｍｉｓｓｉｏｎ）を与えることができることが望ましい。したがって、本発明の他の機能は、メディアに保存されたデータへのアクセスに関して、様々なパーミッションを（例えば、様々な権限のあるエンティティに対して）与えるといったアクセスポリシー（ａｃｃｅｓｓ　ｐｏｌｉｃｙ）を保存し得るという認識に基づいている。上記２つの機能の組み合わせを組み込むシステムは特に有益である。ひとつは、コンテンツ所有者または所有権者は、外部デバイスにとって実質的にアクセス不可能な鍵の使用によりコンテンツへのアクセスを管理する能力をもち、同時に、メディアのコンテンツをアクセスするための様々なパーミッションを与える能力をもつ。このように、外部デバイスがアクセスを獲得したとしても、それらのアクセスは、いまだストレージメディアに記録されたコンテンツの所有者または所有権者によって設定された様々なパーミッションに左右され得る。

【０００７】

他の機能は、様々な権限のあるエンティティに異なるパーミッションを与えるという上記ポリシーがフラッシュメモリで実行される場合、コンテンツ保護に特に有用なメディアをもたらすという認識に基づく。

【０００８】

多くのコンピュータホストデバイスは、ファイル形式でデータの読み出しおよび書き込みを実行するが、多くのストレージデバイスはファイルシステムを認識しない。他の機能によれば、ホストデバイスはキーリファレンス（ｋｅｙ　ｒｅｆｅｒｅｎｃｅ）またはＩＤを提供し、これに応じてメモリシステムは、鍵ＩＤ（ｋｅｙ　ＩＤ）に関連する鍵値（ｋｅｙ　ｖａｌｕｅ）を生成し、ここで、鍵値は鍵ＩＤに関連しているファイル中のデータを暗号化処理する際に使用される。ホストは、鍵ＩＤをメモリシステムによって暗号化処理されるファイルと関連付ける。このように、ホストがファイルの制御を維持しつつ、鍵ＩＤは、メモリが暗号プロセス用の鍵値の生成および使用に対して完全で排他的な制御を維持するハンドル（ｈａｎｄｌｅ）として、コンピュータデバイスおよびメモリによって使用される。

【０００９】

スマートカード等のモバイルストレージデバイスの中には、カードコントローラがファイルシステムを管理する。例えば、フラッシュメモリ、磁気ディスクまたは光ディスク等、多くの他の種類のモバイルストレージデバイスでは、デバイスコントローラがファイルシステムを認識しない。その代わりに、デバイスコントローラはファイルシステムを管理するために、ホストデバイス（例えば、パーソナルコンピュータ、デジタルカメラ、ＭＰ３プレーヤ、携帯情報端末、携帯電話）に依存している。本発明の様々な態様は、デバイスコントローラがファイルシステムを認識しないタイプのストレージデバイスに直ちに組み込むことが可能である。これは、本発明の様々な機能が、多種多様の既存のモバイルストレージデバイスにデバイスを再設計することなく、デバイスがファイルシステムを認識して管理するようにデバイスコントローラを作ることが実施し得ることを意味する。

【００１０】
ストレージメディアに保存されたツリー構造は、エンティティがアクセスの獲得後でさえ実行可能な事柄に対して制御することを与えるものである。ツリーを構成する複数ノード（ｎｏｄｅ）の各々は、当該ツリーのノードを通じたエントリ（ｅｎｔｒｙ）を獲得したエンティティによるパーミッションを特定する。ツリーの中には、異なるレベル（ｌｅｖｅｌ）を有しているものもあり、この場合、ツリーのノードにおける一または複数のパーミッションが、同一ツリー内の高位レベル、低位レベル、または同一レベルとなる他のノードにおける一つまたは複数のパーミッションと所定の関係を有する。エンティティに対して、各ノードでこのように特定されたパーミッションに従うことを要求することにより、このアプリケーションのツリーの機能は、ツリーが異なるレベルを有しているか否かに関係なく、どのエンティティが動作可能で、且つ、各エンティティのどの動作が可能であるかについてコンテンツ所有者に制御することを可能とする。
【００１１】
モバイルストレージメディアにより提供可能な商業価値を高めるために、モバイルストレージデバイスが複数のアプリケーションを同時にサポート可能であることが望ましい。２以上のアプリケーションがモバイルストレージデバイスに同時にアクセスする場合、当該アプリケーションが、ここでクロストーク（ｃｒｏｏｓｔａｌｋ）と称す現象で相互干渉しないように、当該アプリケーションのオペレーションを分離可能であることが重要になる。したがって、本発明の他の機能は、好ましくは階層構造になっている２以上のツリーを、メモリへのアクセス制御に提供し得るという認識に基づく。各ツリーは、対応する複数のエンティティセットによるデータへのアクセスを制御するために、異なるレベルのノードを備えており、この場合、各ツリーのノードは、対応する一または複数のエンティティがメモリデータにアクセスするための一または複数のパーミッションを特定する。各ツリーのノードにおける一または複数のパーミッションが、同一ツリー内の高位レベルまたは低位レベルの他のノードにおける一または複数のパーミッションと所定の関係を有する。少なくとも、２つのツリー間でクロストークが存在しないことが好ましい。
【００１２】
以上から、ツリーがコンテンツセキュリティに使用可能な強力な構造であることは明白である。提供される重要な制御の一つは、ツリー生成に対する制御である。このように、本発明に係る他の機能によれば、モバイルストレージデバイスは、対応するエンティティによりメモリ内に保存されたデータへのアクセス制御のために、異なるレベルのノードを備えた少なくとも一つの階層ツリーを生成可能なシステムエージェント（ｓｙｓｔｅｍ ａｇｅｎｔ）が備えられてもよい。ツリーの各ノードは、メモリデータへのアクセスのために、対応する一または複数のエンティティの一つ又は複数のパーミッションを特定する。各ツリーのノードにおける一または複数のパーミッションが、同一ツリー内の高位レベル、低位レベル、または同一レベルのノードでの一または複数のパーミッションと所定の関係を有する。このように、モバイルストレージデバイスは、デバイス購入者が自身の考えるアプリケーションに適用される階層ツリーを自由に生成できるように、既に生成された任意のツリーを使用することなく発行（ｉｓｓｕｅｄ）されてもよい。あるいは、モバイルストレージデバイスは、購入者が面倒なツリーの生成をする必要がないように、既に生成されたツリーを使用して発行されてもよい。両方の状況において、ツリーの所定の機能をさらに変更または修正できないように、デバイス製造後に当該機能を確定可能であることが好ましい。これは、コンテンツ所有者によるデバイス内のコンテンツへのアクセスに対して、さらなる制御を提供する。このように、一実施形態においては、追加のツリーを生成できないように、システムエージェントの無効化が可能であることが好ましい。
【００１３】
モバイルストレージデバイスの中には、コンテンツ保護は、保護領域へのアクセスが事前認証を必要とする別々の領域にメモリを分割することにより得られるものがある。このような態様により、何らかの保護が提供されるのであるが、違法な手段でパスワードを取得したユーザから保護するものではない。このように、本発明の他の態様は、メカニズム

または構造がメモリを複数のパーティションに分割するようにすることができて、少なくともパーティションにおけるデータの中には、鍵を用いて暗号化することが可能なものもあるという認識に基づくものであり、結果として、複数のパーティションの一部へのアクセスに必要な認証に加えて、一または複数の鍵へのアクセスがこうしたパーティションにおける暗号化データの復号に必要とされ得る。

【００１４】

幾つかのアプリケーションでは、ユーザが一つのアプリケーションを使用してメモリシステムにログインし、その後、再度ログインすることなく保護コンテンツにアクセスするために、異なるアプリケーションを使用可能であるという点が、さらに便利であるかもしれない。こうしたイベントでは、この方法でユーザがアクセスを望むコンテンツのすべてが、第１のアカウントに関連付けられ、結果として、複数回数のログインを必要とせず、異なるアプリケーション（例えば、音楽プレーヤ、Ｅメール、移動体通信等）を経由して、当該コンテンツにアクセス可能となるようにしてもよい。そして、異なる認証情報セットは、たとえ異なる複数アカウントが同一ユーザまたはエンティティに設定されている場合であっても、第１のアカウントとは異なるアカウントに存在する保護コンテンツにアクセスするためのログインに使用されてもよい。

【００１５】

上記機能は、コンテンツ所有者に対する制御および／または保護のさらなる汎用性（ｖｅｒｓａｔｉｌｉｔｙ）を提供するためのストレージシステムにおいて、個別に使用し、または如何なるコンビネーションで組み合わせてもよい。

【特許文献１】米国特許第６４５７１２６号

【発明を実施するための最良の形態】

【００１６】

本発明に関する様々な態様を実施し得るメモリシステムの一例が、図１のブロック図により例示されている。図１に図示されているように、メモリシステム１０は、中央演算装置（CPU）１２、バッファマネジメントユニット（ｂｕｆｆｅｒ ｍａｎａｇｅｍｅｎｔ ｕｎｉｔ；BMU）１４、ホストインタフェイスモジュール（ｈｏｓｔ ｉｎｔｅｒｆａｃｅ ｍｏｄｕｌｅ；ＨＩＭ）１６、フラッシュインタフェイスモジュール（ｆｌａｓｈ ｉｎｔｅｒｆａｃｅ ｍｏｄｕｌｅ；ＦＩＭ）１８、フラッシュメモリ２０およびペリフェラルアクセスモジュール（ｐｅｒｉｐｈｅｒａｌ ａｃｃｅｓｓ Ｍｏｄｕｌｅ；ＰＡＭ）２２を備える。メモリシステム１０は、ホストインタフェイスバス２６およびポート２６ａを介してホストデバイス２４と通信する。フラッシュメモリ２０（ＮＡＮＤ型であってもよい）は、ホストデバイス２４にデータストレージを提供する。ＣＰＵ１２用のソフトウェアコードもフラッシュメモリ２０に保存されてもよい。ＦＩＭ１８は、フラッシュインタフェイスバス２８およびポート２８ａを介してフラッシュメモリ２０に接続する。ＨＩＭ１６は、例えば、デジタルカメラ、パーソナルコンピュータ、携帯情報端末（ＰＤＡ）、デジタルメディアプレーヤ、ＭＰ３プレーヤ、携帯電話、その他のデジタルデバイス等のホストシステムへの接続に適している。ペリフェラルアクセスモジュール２２は、例えば、ＣＰＵ１２と通信するためのＦＩＭ，ＨＩＭおよびＢＭＵ等の適切なコントローラモジュールを選択する。一実施形態においては、点線で示すボックス内のシステム１０の全構成要素が、例えばメモリカードやスティック１０’等の単一ユニットに含まれていてもよく、カプセル化されていることが好ましい。

【００１７】

ここで、本発明はフラッシュメモリを参照することにより説明されているが、例えば磁気ディスクや光学式ＣＤ等の他のタイプのメモリだけでなく、あらゆるタイプの書き換え可能な不揮発性メモリシステムに適用してもよい。

【００１８】

バッファマネジメントユニット１４は、ホストダイレクトメモリアクセス（ｈｏｓｔ ｄｉｒｅｃｔ ｍｅｍｏｒｙ ａｃｃｅｓｓ；HDMA）３２、フラッシュダイレクトメモリアクセス（ｆｌａｓｈ ｄｉｒｅｃｔ ｍｅｍｏｒｙ ａｃｃｅｓｓ；FDMA）３

４、アービタ（ａｒｂｉｔｅｒ）３６、バッファランダムアクセスメモリ（ｂｕｆｆｅｒ　ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ；ＢＲＡＭ）３８および暗号化エンジン（ｃｒｙｐｔｏ－ｅｎｇｉｎｅ）４０を備える。アービタ３６は共有バスアービタであり、これにより、一つのマスタまたはイニシエータ（ｉｎｉｔｉａｔｏｒ；ＨＤＭＡ３２、ＦＤＭＡ３４またはＣＰＵ１２がこれに該当し得る）のみが常時アクティブになることが可能である。また、スレーブまたはターゲットはＢＲＡＭ３８である。アービタは、適切なイニシエータ要求をＢＲＡＭ３８にチャネル接続させる役割を担う。ＨＤＭＡ３２およびＦＤＭＡ３４は、ＨＩＭ１６、ＦＩＭ１８およびＢＲＡＭ３８またはＣＰＵランダムアクセスメモリ（ＣＰＵ　ｒａｎｄｏｍ　ａｃｃｅｓｓ　ｍｅｍｏｒｙ；ＣＰＵ　ＲＡＭ）１２ａ間で伝送されたデータに対する役割を担う。ＨＤＭＡ３２のオペレーションおよびＦＤＭＡ３４のオペレーションは従来から実施されているので、ここでは詳細な説明を省略する。ＢＲＡＭ３８は、ホストデバイス２４およびフラッシュメモリ２０間で伝送されたデータを保存するために使用される。ＨＤＭＡ３２およびＦＤＭＡ３４は、ＨＩＭ１６／ＦＩＭ１８およびＢＲＡＭ３８、またはＣＰＵ　ＲＡＭ１２ａ間のデータ伝送、およびセクタ完了の表示に関する役割を担う。

【００１９】

メモリシステム１０は、メモリ２０に保存されたコンテンツの安全性を向上させるために、暗号化および／または復号用に使用される（一または複数の）鍵値を生成し、ここでは、当該鍵値は、ホストデバイス２４等の外部デバイスにとってアクセスが実質的に不可能である。しかし、暗号化および復号は典型的にはファイル単位で実行されるが、これは、ホストデバイスがメモリシステム１０に対してファイル形式でデータの読み出しおよび書き込みを実行するからである。多くの他の種類のストレージデバイスと同様に、メモリデバイス１０はファイルまたはファイルシステムを認識しない。メモリ２０はファイルの論理アドレスが識別されるファイルアロケーションテーブル（ｆｉｌｅ　ａｌｌｏｃａｔｉｏｎ　ｔａｂｌｅ；ＦＡＴ）を保存するが、ＦＡＴは、典型的にはコントローラ１２ではなくホストデバイス２４によってアクセスおよび管理される。したがって、特定ファイルのデータを暗号化するために、コントローラ１２はメモリ２０内のファイルのデータの論理アドレスを送信するために、ホストデバイスに依存しなければならないことになり、そのため、特定ファイルのデータはシステム１０にのみ利用可能な（一または複数の）鍵値を使用するシステム１０によって認識されて、暗号化および／または復号され得る。

【００２０】

ファイル内のデータを暗号化処理する（一または複数の）同一鍵を参照するホストデバイス２４およびメモリシステム１０の両方にハンドルを提供するために、ホストデバイスは、システム１０により生成された各々の鍵値のための参照を提供し、ここでは、当該参照は単なる鍵ＩＤとしてもよい。このように、ホスト２４はシステム１０により暗号化処理される各ファイルを鍵ＩＤと関連付け、システム１０はデータの暗号化処理に使用される各鍵値をホストにより提供された鍵ＩＤと関連付ける。このように、ホストがファイルの暗号化処理を要求する場合、ホストは、鍵ＩＤと、メモリ２０からフェッチされ、またはメモリ２０に保存されるデータの論理アドレスと共に、当該要求をシステム１０に送信する。システム１０は鍵値を生成し、ホスト２４により提供された鍵ＩＤと当該値を関連付けて、暗号化プロセスを実行する。この方法では、メモリシステム１０の動作方法に変更を加える必要はないが、システム１０が（一または複数の）鍵値への排他的アクセスを含む（一または複数の）鍵を使用する暗号化処理を完全に制御することができる。即ち、ＦＡＴの排他的制御により、システム１０はホスト２４のファイル管理を継続して可能にする。その一方で、システム１０は暗号化処理に使用される（一または複数の）鍵値の生成および管理を行うために排他的制御を維持する。ホストデバイス２４は、データの暗号化処理に使用される（一または複数の）鍵値の生成および管理に一切関与しない。

【００２１】

ホスト２４により提供された鍵ＩＤおよびメモリシステムにより生成された鍵値は、以下、実施形態の一つにおいて「コンテンツ暗号化鍵（ｃｏｎｔｅｎｔ　ｅｎｃｒｙｐｔｉ

ｏｎ　ｋｅｙ）」またはＣＥＫとして称される、数量に関する２つの属性を形成する。ホスト２４は各鍵ＩＤを一または複数のファイルと関連付けるが、ホスト２４は、各鍵ＩＤを整理されていないデータまたは何らかの方法で整理されたデータとも関連付けるものの、これは完全なファイルに整理されたデータに限定されない。

【００２２】

ユーザまたはアプリケーションは、システム１０における保護コンテンツまたは保護領域へのアクセスを獲得するために、システム１０に事前登録された証明書を使用して認証を受ける必要がある。証明書（ｃｒｅｄｅｎｔｉａｌ）とは、証明書を有する特定のユーザまたはアプリケーションに許可されるアクセス権限に関係するものである。事前登録プロセスにおいて、システム１０はユーザまたはアプリケーションの身元（ｉｄｅｎｔｉｔｙ）および証明書の記録、およびユーザまたはアプリケーションにより決定され、ホスト２４を通じて提供された当該身元および証明書と関連付けられたアクセス権限（ａｃｃｅｓｓ　ｒｉｇｈｔｓ）を保存する。事前登録の完了後、ユーザまたはアプリケーションがメモリ２０へのデータ書き込みを要求する場合、ホストデバイスを通じて、その身元および証明書、データ暗号化用の鍵ＩＤおよび暗号化データが保存される論理アドレスを提供する必要がある。システム１０は、鍵値を生成し、当該値をホストデバイスにより提供された鍵ＩＤと関連付け、書き込まれるデータの暗号化に使用される鍵値に対応する鍵ＩＤを当該ユーザまたはアプリケーションに関する記録またはテーブルに保存する。その後、システム１０はデータを暗号化し、生成した鍵値だけでなく、暗号化データをホストにより指定されたアドレスに保存する。

【００２３】

ユーザまたはアプリケーションがメモリ２０からの暗号化データの読み出しを要求する場合、その身元および証明書と、要求されたデータを暗号化するために以前使用された鍵に対応する鍵ＩＤと、暗号化データが保存される論理アドレスと、を提供する必要がある。その後、システム１０はホストにより提供されたユーザまたはアプリケーションの身元および証明書と、その記録内に保存された身元および証明書と、を照合する。両者が一致した場合、システム１０は、ユーザまたはアプリケーションにより提供された鍵ＩＤに関連付けられた鍵値をメモリからフェッチし、鍵値を使用して、ホストデバイスにより指定されたアドレスに保存されたデータを復号し、復号されたデータをユーザまたはアプリケーションに送信する。

【００２４】

認証証明書を暗号化処理に使用される鍵の管理と分離することにより、証明書を共有することなくデータへのアクセス権限を共有することが可能となる。このように、異なる証明書を有するユーザグループまたはアプリケーショングループは、同一データにアクセスするための同一鍵へのアクセスを有することが可能であるが、当該グループ以外のユーザはアクセスすることができない。グループ内のすべてのユーザまたはアプリケーションは、同一データへのアクセスを有することが可能であるが、当該ユーザまたはアプリケーションはさらに異なる権利を有することができる。このように、ユーザまたはアプリケーションの中に、読み出しのみのアクセスを有する者があってもよく、それ以外のユーザまたはアプリケーションが書き込みのみのアクセスを有しても良く、その他のユーザまたはアプリケーションが両方のアクセスを有してもよい。システム１０は、ユーザまたはアプリケーションの身元および証明書の記録、ユーザまたはアプリケーションがアクセスする鍵ＩＤ、および各鍵ＩＤに対する関連付けられたアクセス権限を維持するので、システム１０は、鍵ＩＤの追加または削除、および特定のユーザまたはアプリケーション用の当該鍵ＩＤに関連付けられたアクセス権限の変更、あるユーザまたはアプリケーションから他のユーザまたはアプリケーションへのアクセス権限の委譲、複数ユーザまたはアプリケーション用の記録またはテーブルの削除または追加までも実行することが可能となり、これらすべてが適切に認証されたホストデバイスにより制御される。保存された記録は、セキュアチャネル（ｓｅｃｕｒｅ　ｃｈａｎｎｅｌ）が所定の鍵へのアクセスに必要であることを特定してもよい。認証は、パスワードだけでなく、対称アルゴリズム（ｓｙｍｍｅｔｒ

ｉｃ　ａｌｇｏｒｉｔｈｍ）または非対称アルゴリズム（ａｓｙｍｍｅｔｒｉｃ　ａｌｇｏｒｉｔｈｍ）を使用して実行されてもよい。

【００２５】

特に重要なのは、メモリシステム１０における保護されたコンテンツ（ｓｅｃｕｒｅｄ　ｃｏｎｔｅｎｔ）の移植性である。鍵値はメモリシステムにより生成され、実質的に外部システムには利用できないので、メモリシステムまたは当該システムを組み込んでいるストレージデバイスが、ある外部システムから他の外部システムに伝送される場合、そこに保存されたコンテンツの安全性が維持され、外部システムは、メモリシステムに完全制御された方法で認証を受けない限り、当該コンテンツにアクセスすることができない。こうした認証後でさえ、アクセスはメモリシステムにより完全制御され、外部システムはメモリシステムに事前設定された記録に応じて制御される方法でのみアクセス可能である。要求は、そうした記録と適合しない場合には拒否される。

【００２６】

コンテンツ保護をさらに柔軟にするために、以下パーティション（ｐａｒｔｉｔｉｏｎ）と称すメモリの所定の領域が適切に認証されたユーザまたはアプリケーションによってのみアクセス可能であることが想定される。上記の鍵ベースのデータ暗号化の特性を組み合わせた場合、システム１０は優れたデータ保護機能を提供する。図２に示されるように、フラッシュメモリ２０は多くのパーティション、即ち、ユーザ領域（ｕｓｅｒ　ａｒｅａ）またはユーザパーティション（ｕｓｅｒ　ｐａｒｔｉｔｉｏｎ）、およびカスタムパーティション（ｃｕｓｔｏｍ　ｐａｒｔｉｔｉｏｎ）、に分割されたストレージ容量を有してもよい。ユーザ領域またはパーティションＰ０は、認証なしですべてのユーザおよびアプリケーションによるアクセスが可能である。ユーザ領域に保存されたデータのすべてのビット値（ｂｉｔ　ｖａｌｕｅ）は、任意のアプリケーションまたはユーザによって読み出しまたは書き込みが可能であるが、データの読み出しが暗号化される場合、復号権限を有していないユーザまたはアプリケーションは、ユーザ領域に保存されたビット値で表現された情報にアクセスすることはできないことになる。例えば、これはユーザエリアＰ０に保存されたファイル１０２および１０４により説明されている。例えば、すべてのアプリケーションおよびユーザにより読み出しおよび理解が可能なファイル１０６等の暗号化されていないファイルもユーザ領域に保存されている。このように、暗号化されるファイルは、例えばファイル１０２および１０４のように、当該ファイルに関連付けられた錠前と共に象徴的に示されている。

【００２７】

ユーザ領域Ｐ０における暗号化ファイルは、権限のないアプリケーションまたはユーザには理解することができないが、そのアプリケーションまたはユーザが、一部のアプリケーションにとって望ましくない可能性のあるファイルを削除または破壊することが可能であってもよい。また、この目的のために、メモリ２０は事前認証なしでアクセス不可能な、例えばパーティションＰ１およびＰ２等の保護されたカスタムパーティションを備える。以下、本出願に係る複数の実施形態で可能となる認証プロセスについて説明する。

【００２８】

図２に示されるように、様々なユーザまたはアプリケーションがメモリ２０内のファイルにアクセスしてもよい。このように、図２には、ユーザ１乃至２および（デバイスで実行する）アプリケーション１乃至４が示されている。こうしたエンティティは、メモリ２０の保護コンテンツへのアクセスを許可される前に、第一に以下で説明する方法の認証プロセスで認証される。当該プロセスでは、アクセスを要求するエンティティは、ロールベースアクセスコントロール（ｒｏｌｅ　ｂａｓｅｄ　ａｃｃｅｓｓ　ｃｏｎｔｒｏｌ）を行うために、ホスト側で識別される必要がある。このように、アクセスを要求するエンティティは、第一に、例えば「こちらアプリケーション２、ファイル１を読み出せ」等の情報を提供することにより、身元を示す。そして、コントローラ１２は、身元、認証情報および要求をメモリ２０またはコントローラ１２に保存された記録と照合する。すべての要求が満たされた場合、当該エンティティにアクセスが許可される。図２に示すように、ユ

ーザ１はＰ０のファイル１０６に対する読み出しおよび書き込みに関して無制限の権利を有していることに加え、パーティションＰ１のファイル１０１に対する読み出しおよび書き込みを許可されているが、ファイル１０２および１０４に対しては読み出しのみが可能である。一方、ユーザ２はファイル１０１および１０４に対するアクセスを許可されていないが、ファイル１０２に対する読み出しおよび書き込みアクセスを有する。図２で示すように、ユーザ１および２は同一のログインアルゴリズム（ＡＥＳ）を有しているが、アプリケーション１および３は異なるログインアルゴリズム（例えば、ＲＳＡおよび００１００１）を有し、ここで、当該異なるログインアルゴリズムは、ユーザ１および２のログインアルゴリズムとも異なる。

【００２９】

セキュアストレージアプリケーション（ＳＳＡ）はメモリシステム１０のセキュリティアプリケーションで、本発明の一実施形態を説明するものである。当該アプリケーションは上記機能の多くを実施するために使用可能である。ＳＳＡはメモリ２０またはＣＰＵ１２の不揮発性メモリ（不図示）に保存されたデータベースを有するソフトウェアまたはコンピュータコードとして具現化することができ、ＲＡＭ１２ａに読み込まれ、ＣＰＵ１２により実行される。

ＳＳＡを参照して使用される頭字語（ａｃｒｏｎｙｍｓ）は、以下のテーブルで説明される。

定義、頭字語及び略語（Definitions, Acronyms & Abbreviations）

ＡＣＲ	アクセス制御記録（Access Control Records）
ＡＧＰ	ＡＣＲグループ（ACR Group）
ＣＢＣ	チェーンブロック暗号（Chain Block Cipher）
ＣＥＫ	コンテンツ暗号鍵（Content Encryption Key）
ＥＣＢ	電子コードブック（Electronic Codebook）
ＡＣＡＭ	ＡＣＲ属性管理（ACR Attributes Management）
ＰＣＲ	許可制御記録（Permissions Control Record）
ＳＳＡ	セキュアストレージアプリケーション（Secure Storage Application）
エンティティ	ＳＳＡにログインし、その機能を利用する、実在する個別の存在（ホスト側）を有する任意のもの

【００３０】

《ＳＳＡシステム記述》

データの安全性（ｓｅｃｕｒｉｔｙ）、完全性（ｉｎｔｅｇｒｉｔｙ）およびアクセス制御がＳＳＡの主要な役割である。データは、本来、何らかの大容量記憶装置に単純に保存されるファイルである。ＳＳＡシステムはストレージシステム内に位置し、保存されたホストファイルに対してセキュリティレイヤ（ｓｅｃｕｒｉｔｙ　ｌａｙｅｒ）を追加する。

【００３１】

ＳＳＡの主なタスクは、メモリに保存された（そして保護された）コンテンツに関連付けられた異なる権利を管理することにある。メモリアプリケーションは、複数のユーザおよび複数の保存コンテンツに対応するコンテンツ権を管理する必要がある。ホストアプリケーションは、ホストアプリケーション側から、こうしたアプリケーションが視認可能なドライブおよびパーティションと、ストレージデバイスに保存されたファイルの位置を管理および表現するファイルアロケーションテーブル（ＦＡＴ）と、を確認する。

【００３２】

この場合、ストレージデバイスはパーティションに分割されたＮＡＮＤフラッシュチップ（ＮＡＮＤ　ｆｌａｓｈ　ｃｈｉｐ）を使用するが、ここで他のモバイルストレージデバイスを使用してもよく、当該デバイスは本発明の範囲内に属するものである。これらのパーティションは論理アドレスの連続スレッドであり、ここでは「開始（ｓｔａｒｔ）」アドレスおよび「終了（ｅｎｄ）」アドレスがパーティションの境界を画定する。したが

って、必要に応じて、ソフトウェア（例えば、メモリ２０に保存されたソフトウェア）により隠しパーティション（ｈｉｄｄｅｎ　ｐａｒｔｉｔｉｏｎ）へのアクセスに制限を加えてもよいが、当該ソフトウェアは、こうした制限を境界内のアドレスと関連付ける。パーティションは、ＳＳＡの管理するパーティションの論理アドレス境界により、ＳＳＡが完全に認識可能である。ＳＳＡシステムは、権限のないホストアプリケーションから物理的にデータを保護するためにパーティションを使用する。ホストに対しては、パーティションはデータファイルを保存するための所有権空間を定義するメカニズムである。これらのパーティションは、ストレージデバイスへのアクセスを有する者であればデバイス上のパーティションの存在を確認し、認識することが可能であるという点で公的（ｐｕｂｌｉｃ）であると言える。また、選択されたホストアプリケーションのみがストレージデバイス上でのパーティションへのアクセスを有し、その存在を認識するという点で私的（ｐｒｉｖａｔｅ）または秘匿的（ｈｉｄｄｅｎ）であると言える。

【００３３】

図３は、メモリのパーティション、即ち、Ｐ０、Ｐ１、Ｐ２およびＰ３（４つ未満または４つを超えるパーティションを採用してもよいことは明白である）を説明するメモリの概略図であり、ここで、Ｐ０は認証なしで任意のエンティティがアクセス可能なパブリックパーティション（ｐｕｂｌｉｃ　ｐａｒｔｉｔｉｏｎ）である。

【００３４】

（例えば、Ｐ１、Ｐ２、またはＰ３の）プライベートパーティション（ｐｒｉｖａｔｅ　ｐａｒｔｉｔｉｏｎ）は、その範囲内にファイルへのアクセスを隠している。ホストによるパーティションへのアクセスを妨げることにより、フラッシュデバイス（例えば、フラッシュカード）はパーティション内のデータファイルの保護を実現する。しかし、この種の保護は、パーティション内の論理アドレスに保存されたデータへのアクセスに制限を課すことによって、隠しパーティションに存在するすべてのファイルを対象とする。即ち、当該制限は論理アドレスの範囲に関連付けられている。当該パーティションへのアクセスを有する「ユーザ／ホスト」のすべては、内部のすべてのファイルに対して無制限のアクセスを有する。異なるファイルまたはファイルグループを互いに分離するために、ＳＳＡシステムは鍵およびキーリファレンス、または鍵ＩＤを使用して、ファイル単位またはファイルグループ単位で他のレベルの安全性および完全性を提供する。異なるメモリアドレスでデータを暗号化するために使用される特定の鍵値に関するキーリファレンスまたは鍵ＩＤは、暗号化データを含むコンテナ（ｃｏｎｔａｉｎｅｒ）またはドメイン（ｄｏｍａｉｎ）に類推することができる。こうした理由により、図４においては、キーリファレンスまたは鍵ＩＤ（例えば、「鍵１」および「鍵２」）が鍵ＩＤに関連付けられた鍵値を使用して暗号化されたファイルを取り囲んでいる領域として図示されている。

【００３５】

例えば、図４を参照すると、ファイルＡは鍵ＩＤで囲まれていないので、認証なしですべてのエンティティがアクセス可能である。パブリックパーティションのファイルＢは、すべてのエンティティが読み出しおよび上書き可能であるが、ファイルＢはＩＤ「鍵１」を有する鍵で暗号化されたデータを含む。したがって、ファイルＢに含まれる情報については、鍵へのアクセスを有していない限り、エンティティがアクセスすることはできない。この方法では、鍵値およびキーリファレンス、または鍵ＩＤの使用により、論理保護のみが提供される。これは、上述のパーティションにより提供された種類の保護とは相反する。従って、（パブリックまたはプライベート）パーティションにアクセス可能な任意のホストは、暗号化データを含むパーティション全体のデータの読み出しおよび書き込みが可能である。しかし、データが暗号化されているので、権限のないユーザはデータを破壊することしかできない。権限のないユーザは、露見せずデータを変更することや、データを使用することができないことが好ましい。暗号化鍵および／または復号鍵へのアクセスを制限することにより、この機能は、権限のあるエンティティのみに対してデータの使用を可能にする。また、ファイルＢおよびＣは、Ｐ０の鍵ＩＤ「鍵２」を有する鍵を使用して暗号化されている。

【００３６】
データの機密性（ｃｏｎｆｉｄｅｎｔｉａｌｉｔｙ）および完全性は、コンテンツ暗号化鍵（Ｃｏｎｔｅｎｔ　Ｅｎｃｒｙｐｔｉｏｎ　Ｋｅｙｓ；ＣＥＫ）を使用する対称暗号法を通じて、各ＣＥＫに対して提供可能である。ＳＳＡ実施形態において、ＣＥＫはフラッシュデバイス（例えば、フラッシュカード）により生成され、内部でのみ使用され、外部からは秘密にされる。また、暗号化（ｅｎｃｒｙｐｔｅｄ　ｏｒ　ｃｉｐｈｅｒｅｄ）されたデータは、ハッシュ化（ｈａｓｈｅｄ）されてもよく、データの完全性を確実にするために、暗号がチェーンブロックされてもよい。
【００３７】
必ずしもパーティションのすべてのデータが異なる鍵で暗号化され、異なる鍵ＩＤに関連付けられる訳ではない。パブリックファイルまたはユーザファイルの所定の論理アドレス、またはオペレーティングシステム領域（即ちＦＡＴ）の所定の論理アドレスは、任意の鍵またはキーリファレンスと関連付けられなくてもよく、これにより、パーティション自体にアクセス可能な任意のエンティティによって利用可能となる。
【００３８】
鍵およびパーティションの生成能力に加えて、それらのデータの書き込みおよび読み出し、または鍵の使用に関する能力を必要とするエンティティは、アクセス制御記録（Ａｃｃｅｓｓ　Ｃｏｎｔｒｏｌ　Ｒｅｃｏｒｄ；ＡＣＲ）を通じてＳＳＡシステムにログインすることが必要である。ＳＳＡシステムのＡＣＲの権限は「アクション（Ａｃｔｉｏｎｓ）」と呼ばれる。各ＡＣＲは以下３つのカテゴリのアクション、即ち、パーティションおよび鍵／鍵ＩＤの生成、パーティションおよび鍵へのアクセス、および他のＡＣＲの生成／更新、の３つのカテゴリのアクションを実行する「パーミッション（Ｐｅｒｍｉｓｓｉｏｎｓ）」を有してもよい。
【００３９】
ＡＣＲはＡＣＲグループまたはＡＧＰと呼ばれるグループに整理される。ＡＣＲの認証に一旦成功すれば、ＳＳＡシステムはＡＣＲのアクションの何れかを実行可能な「セッション」（Ｓｅｓｓｉｏｎ）を開始する。
【００４０】
《ユーザパーティション（一または複数）》
ＳＳＡは一または複数のパブリックパーティションを管理し、これをユーザパーティション（一または複数の）と称すこともある。当該パーティションはストレージデバイスに存在し、ストレージデバイスの標準的な読み出しおよび書き込みコマンドによりアクセス可能な一または複数のパーティションである。デバイスにおけるパーティションの存在だけでなく、パーティション（一または複数の）のサイズに関する情報の取得は、ホストシステムから隠蔽不可能であることが好ましい。
【００４１】
ＳＳＡシステムは、標準的な読み出しおよび書き込みコマンド、またはＳＳＡコマンドの何れかにより当該パーティションにアクセス可能である。したがって、パーティションへのアクセスを特定ＡＣＲに制限できないことが好ましい。しかし、ＳＳＡシステムにより、ホストデバイスはユーザパーティションへのアクセス制限を有効にすることが可能である。読み書きアクセスは、個別に有効／無効にすることができる。４つの組み合わせすべて（例えば、書き込みのみ、読み出しのみ（書き込み禁止）、読み書き、アクセス不可）が可能である。
【００４２】
ＳＳＡシステムにより、ＡＣＲは鍵ＩＤをユーザパーティション内のファイルと関連付け、そうした鍵ＩＤと関連付けられた鍵を使用して、個々のファイルを暗号化することが可能である。パーティションへのアクセス権限の設定だけでなく、ユーザパーティション内の暗号化ファイルへのアクセスは、ＳＳＡコマンドセットを使用して実行される（ＳＳＡコマンドの詳細な記述に関する付録Ａを参照。当該付録では、鍵ＩＤを「ドメイン」と称す）。また、上述の機能はファイルに整理されていないデータに適用する。

【００４３】
《ＳＳＡパーティション》
これらは（ホストオペレーティングシステムまたはＯＳから）隠されたパーティションであり、ＳＳＡコマンドを通じてのみアクセス可能である。ＳＳＡシステムによって、ホストデバイスがＡＣＲにログインすることにより確立される（以下に示す）セッションを介すことなくＳＳＡパーティションにアクセスすることができないことが好ましい。同様に、確立されたセッションを通じて当該要求がなされない限り、ＳＳＡはＳＳＡパーティションの存在、サイズおよびアクセス許可に関する情報を提供しないことが好ましい。
【００４４】
パーティションへのアクセス権限はＡＣＲパーミッションに基づいている。ＡＣＲは、ＳＳＡシステムに一旦ログインすると、パーティションを（以下に示す）他のＡＣＲと共有することができる。パーティションの生成時に、ホストは参照名またはＩＤ（例えば、図３および４のＰ０乃至Ｐ３）をパーティションに供給する。当該参照はパーティションに対するさらなる読み出しおよび書き込みコマンドで使用される。
【００４５】
《ストレージデバイスのパーティション化》
デバイスの利用可能なストレージ容量のすべてが、ユーザパーティションおよび現段階で設定されたＳＳＡパーティションに割り当てられることが好ましい。したがって、任意の再パーティションオペレーション（ｒｅｐａｒｔｉｔｉｏｎ　ｏｐｅｒａｔｉｏｎ）は既存のパーティションの再設定を含んでも良い。デバイス容量（全パーティションの合計サイズ）に対する純変化（ｎｅｔ　ｃｈａｎｇｅ）はゼロになる。デバイスメモリ空間におけるパーティションのＩＤは、ホストシステムにより定義される。
【００４６】
ホストシステムは、既存のパーティションの一つを二つのより小さなパーティションに再パーティション化する、または（隣接してもよいし、隣接しなくてもよい）二つの既存のパーティションを一つのパーティションに結合することが可能である。分割パーティションまたは結合パーティションのデータは、ホストの裁量で消去可能であるし、そのままの状態にすることも可能である。
【００４７】
ストレージデバイスの再パーティション化は、（ストレージデバイスの論理アドレス空間におけるデータの消去または移動が原因で）データ損失を引き起し得るので、ＳＳＡシステムにより、再パーティション化の実行に厳重な制限が課される。（以下に示す）ルートＡＧＰに存在するＡＣＲのみが再パーティションコマンドの発行を許可され、それが所有するパーティションを参照することのみ可能である。ＳＳＡシステムは、データが如何にしてパーティション（ＦＡＴまたは他のファイルシステム構造）に整理されるのかを認識しないので、デバイスが再パーティション化される場合は常に、こうした構造を再構築するのはホストの責任となる。
【００４８】
ホストＯＳによって確認されながら、当該パーティションのサイズおよび他の属性は、ユーザパーティションの再パーティション化により変更される。
【００４９】
再パーティション化の実行後、ＳＳＡシステムの任意のＡＣＲが実在しないパーティションを参照しないことを確認するのは、ホストシステムの責任である。こうしたＡＣＲが適切に削除または更新されない場合、こうしたＡＣＲに代わって、実在しないパーティションにアクセスする以後の試みはシステムにより削除および拒否される。削除された鍵および鍵ＩＤに関しても、同様の配慮がなされる。
【００５０】
《鍵、鍵ＩＤおよび論理保護》
ファイルが所定の隠しパーティションに書き込まれる場合、当該ファイルは不特定多数者から隠される。しかし、（敵意があるか否かは別として）一旦あるエンティティが当該

パーティションに関する知識およびアクセスを取得すると、ファイルは利用可能で視認容易なものになる。ファイルをさらに保護するために、ＳＳＡは隠しパーティションでファイルを暗号化することができる。この場合、ファイルを復号するための鍵へのアクセスに関する証明書は、パーティションへのアクセスに関する証明書と異なることが好ましい。ファイルは（ホストにより完全に制御および管理され）ＳＳＡが認識するものではないという事実を考慮すると、ＣＥＫをファイルと関連付けることには問題がある。ファイルをＳＳＡが認知する何らかのもの（例えば、鍵ＩＤ）とリンクすることによりこれを修正する。このように、ＳＳＡにより鍵が生成される場合、ホストはＳＳＡにより生成された鍵を使用して、当該鍵に関する鍵ＩＤを暗号化データと関連付ける。

【００５１】

鍵値および鍵ＩＤは論理的セキュリティ（ｌｏｇｉｃａｌ　ｓｅｃｕｒｉｔｙ）を提供する。所定の鍵ＩＤに関連づけられたすべてのデータは、その位置にかかわらず、ホストアプリケーションにより生成時に一意的に参照名または鍵ＩＤを与えられた同一のコンテンツ暗号鍵（ＣＥＫ）を用いて暗号化される。エンティティが（ＡＣＲを通じた認証により）隠しパーティションへのアクセスを獲得し、当該パーティション内の暗号化ファイルの読み出しまたは書き込みの実行を望む場合、ファイルに関連付けられた鍵ＩＤへのアクセスを有する必要がある。ＳＳＡは、当該鍵ＩＤに対応する鍵へのアクセスを許可する場合、当該鍵ＩＤに関連付けられたＣＥＫの鍵値をロードし、ホストへのデータ送信前にこれを復号する、または、フラッシュメモリ２０へのデータ書き込み前にこれを暗号化する。鍵ＩＤに関連付けられたＣＥＫの鍵値は、ＳＳＡシステムにより一旦ランダムに生成され、維持される。ＳＳＡシステムの外部でＣＥＫの当該鍵値に関する知識またはアクセスを有するものはいない。外部では、ＣＥＫの鍵値ではなく、参照または鍵ＩＤが提供される、または使用されるだけである。鍵値はＳＳＡにより完全に管理され、ＳＳＡのみがアクセス可能である。

【００５２】

ＳＳＡシステムは、以下の暗号モードの内（ユーザが定義した）任意のモードを使用して、鍵ＩＤに関連付けられたデータを保護する（なお、ＣＥＫの鍵値だけでなく、使用される実際の暗号アルゴリズムは、システム制御され、外部に漏れることはない）：

【００５３】

《ブロックモード（Ｂｌｏｃｋ　ｍｏｄｅ）》

データがブロックに分割され、各ブロックは個別に暗号化される。

このモードは一般的に安全性がそれほど高くなく、辞書攻撃を受けやすいと考えられている。しかし、このモードでは、ユーザがデータブロック（ｄａｔａ　ｂｌｏｃｋｓ）の任意のブロックにランダムにアクセス可能である。

【００５４】

《連鎖モード（Ｃｈａｉｎｅｄ　ｍｏｄｅ）》

データがブロックに分割され、暗号化プロセス中にブロックがチェーン化される。

各ブロックは、次のブロックの暗号化プロセスに対するインプットの一つとして使用される。このモードはより安全性が高いと考えられているが、データの書き込みおよび読み出しが常に開始から終了まで順に実行されることを必要とし、ユーザが必ずしも受け入れ可能ではないオーバーヘッド（ｏｖｅｒｈｅａｄ）を生成する。

【００５５】

《ハッシュモード（Ｈａｓｈｅｄ）》

データの完全性の確認に使用可能なデータダイジェスト（ｄａｔａ　ｄｉｇｅｓｔ）の付加生成を伴う連鎖モードである。

【００５６】

《ＡＣＲおよびアクセス制御》

ＳＳＡは複数アプリケーションを処理するように設計されており、各アプリケーションはシステムデータベースにおいて、ノードからなるツリーとして表現される。アプリケーション間の相互排除は、ツリーのブランチ（ｂｒａｎｃｈ）間でクロストークを確実に発

生しないようにすることにより達成される。

【００５７】

ＳＳＡシステムへのアクセスを獲得するために、エンティティはシステムのＡＣＲの一つを経由して接続を確立する必要がある。ログイン手順は、ユーザが接続を選択したＡＣＲに組み込まれた定義に応じて、ＳＳＡシステムにより実行される。

【００５８】

ＡＣＲはＳＳＡシステムへの個別のログインポイントである。ＡＣＲはログイン証明書および認証方法を有する。また、ＳＳＡシステム内のログインパーミッションが当該記録に属しており、その中には読み出しおよび書き込み権限がある。これは図５において説明されており、同図では同一ＡＧＰにおいてｎ個のＡＣＲが図示されている。これは、ｎ個のＡＣＲの内、少なくとも幾つかは同一鍵へのアクセスを共有し得ることを意味する。このように、ＡＣＲ＃１およびＡＣＲ＃ｎは、鍵ＩＤ「鍵３」を有する鍵へのアクセスを共有しているが、ここで、ＡＣＲ＃１およびＡＣＲ＃ｎはＡＣＲのＩＤであり、「鍵３」は「鍵３」に関連付けられたデータの暗号化に使用される鍵に対応する鍵ＩＤである。また、同一鍵は複数ファイルまたは複数のデータを暗号化および／または復号するために使用可能である。

【００５９】

ＳＳＡシステムはシステムへの幾つかの種類のログイン形式をサポートしているが、ここで、ユーザが一旦ログインに成功すれば、システムにおけるユーザの権限が変化するように、認証アルゴリズムおよびユーザ証明書が変化してもよい。さらに、図５は異なるログインアルゴリズムおよび証明書を説明している。ＡＣＲ＃１はパスワードログインアルゴリズムおよび証明書としてのパスワードを必要とする。一方、ＡＣＲ＃２はＰＫＩ（公開鍵基盤；ｐｕｂｌｉｃ　ｋｅｙ　ｉｎｆｒａｓｔｒｕｃｔｕｒｅ）ログインアルゴリズムおよび証明書としての公開鍵を必要とする。このように、ログインするために、エンティティは正しいログインアルゴリズムおよび証明書だけでなく、有効なＡＣＲのＩＤを提示する必要がある。

【００６０】

エンティティが一旦ＳＳＡシステムのＡＣＲにログインすると、パーミッション、即ち、ＳＳＡコマンドの使用権は、ＡＣＲに関連付けられたパーミッション制御記録（Ｐｅｒｍｉｓｓｉｏｎｓ　Ｃｏｎｔｒｏｌ　Ｒｅｃｏｒｄ；ＰＣＲ）において定義される。図５において、ＡＣＲ＃１は「鍵３」と関連付けられたデータに対して読み出しのみのパーミッションを認め、ＡＣＲ＃２は示されたＰＣＲに応じて、「鍵５」と関連付けられたデータの読み出しおよび書き込みのパーミッションを認める。

【００６１】

異なるＡＣＲはシステム、例えば、読み出しおよび書き込みに関連付けられた鍵等、において共通の所有権および権限を共有してもよい。これを達成するために、何らかの共通するものを有するＡＣＲは、ＡＧＰにグループ化、即ち、ＡＣＲグループにグループ化される。このように、ＡＣＲ＃１およびＡＣＲ＃ｎは、鍵ＩＤ「鍵３」を有する鍵へのアクセスを共有する。

【００６２】

機密データの安全性を保つセキュア鍵の生成に加えて、ＡＧＰおよびその中のＡＣＲは、階層ツリーに整理される。すなわち、ＡＣＲは、鍵ＩＤ／パーティションに対応する他のＡＣＲエントリの生成も可能であることが好ましい。これらの子ＡＣＲ（ＡＣＲ　ｃｈｉｌｄｒｅｎ）は、親（ｆａｔｈｅｒ）即ちクリエータ（ｃｒｅａｔｏｒ）と同一のパーミッションまたは限られたパーミッションを有しており、親ＡＣＲ自身が生成した鍵へのパーミッションが与えられても良い。言うまでもなく、子ＡＣＲは生成する任意の鍵へのアクセスパーミッションを得る。これは図６において説明されている。このように、ＡＧＰ１２０におけるすべてのＡＣＲは、ＡＣＲ１２２により生成され、当該ＡＣＲの２つはＡＣＲ１２２から「鍵３」と関連付けられたデータへのアクセスパーミッションを引き継いでいる。

【００６３】
《ＡＧＰ》
ＳＳＡシステムへのログインは、ＡＧＰおよびＡＧＰ内のＡＣＲを特定することにより実行される。
【００６４】
各ＡＧＰには固有のＩＤ（参照名）があり、当該ＩＤはＳＳＡデータベースへのエントリ用のインデックスとして使用される。ＡＧＰの生成時に、ＡＧＰ名がＳＳＡシステムに供給される。供給されたＡＧＰ名が当該システムにおいて既に存在する場合、ＳＳＡは生成オペレーションを拒否する。
【００６５】
以下のセクションで記載するように、ＡＧＰはアクセスおよび管理パーミッションの委譲に制限を課すために使用される。図６の２つのツリーが果たす機能の一つは、例えば２つの異なるアプリケーションまたは２つの異なるコンピュータユーザ等、エンティティを完全に分離することによってアクセスを与えることである。当該目的のためには、２つのアクセスプロセスが同時に発生するとしても、実質的に互いに独立する（即ち、実質的にクロストークがない）ことが重要となり得る。これは、各ツリーにおけるＡＣＲおよびＡＧＰのさらなる生成だけでなく、認証およびパーミッションも、他のツリーのそれらと接続されておらず、それらに依存しないことを意味する。したがって、ＳＳＡシステムがメモリ１０で使用される場合、これによりメモリシステム１０が複数のアプリケーションを同時に扱うことができる。また、これにより、２つのアプリケーションが互いに独立する２つの個別のデータ（例えば、複数の写真からなる写真一組および複数の歌曲からなる歌曲一組）にアクセスすることができる。これは図６において説明されている。このように、アプリケーションまたはユーザが図６の上部のツリーにおけるノード（ＡＣＲ）を経由してアクセスする「鍵３」、「鍵Ｘ」および「鍵Ｚ」に関連付けられたデータが複数の写真を含んでもよい。アプリケーションまたはユーザが図６の下部のツリーのノード（ＡＣＲ）を経由してアクセスする「鍵５」および「鍵Ｙ」に関連付けられたデータが複数の歌曲を含んでもよい。ＡＧＰを生成したＡＣＲは、当該ＡＧＰにＡＣＲエントリが存在しない場合にのみ、これを削除するパーミッションを有する。
【００６６】
《エンティティのＳＳＡエントリポイント：アクセス制御記録（ＡＣＲ）》
ＳＳＡシステムにおけるＡＣＲは、エンティティがシステムにログインパーミッションされる方法を記載している。エンティティは、ＳＳＡシステムにログインする場合、実行しようとする認証プロセスに対応するＡＣＲを特定する必要がある。ＡＣＲはパーミッション制御記録（ＰＣＲ）を含み、当該ＰＣＲは、図５で説明されＡＣＲに定義されているように、一旦認証されるとユーザが実施可能な認められたアクションを説明している。ホスト側のエンティティは、ＡＣＲデータフィールドのすべてを供給する。
【００６７】
エンティティのＡＣＲへのログインが成功した場合、エンティティはＡＣＲのパーティション、鍵アクセスパーミッションおよび（以下で説明する）ＡＣＡＭパーミッションのすべてにクエリを行うことが可能となる。
【００６８】
《ＡＣＲのＩＤ》
ＳＳＡシステムのエンティティがログインプロセスを開始した場合、当該エンティティは（ＡＣＲの生成時にホストにより提供されるように）ログイン方法に対応するＡＣＲのＩＤを特定する必要がある。この結果、すべてのログイン要件が満たされた場合、ＳＳＡは正しいアルゴリズムをセットアップし、正しいＰＣＲを選択する。ＡＣＲの生成時に、ＡＣＲのＩＤがＳＳＡに供給される。
【００６９】
《ログイン／認証アルゴリズム》
認証アルゴリズムは、エンティティにより使用されるログイン手順の種類、およびユー

ザの身元の証拠を提供するのに必要となる証明書の種類を特定する。ＳＳＡシステムは幾つかの標準的なログインアルゴリズムをサポートしており、当該アルゴリズムは、手順なし（および証明書なし）およびパスワードベースの手順から、対称暗号化または非対称暗号化の何れかに基づく２方向認証プロトコルに及ぶ。

【００７０】

《証明書》

エンティティの証明書は、ログインアルゴリズムに対応し、ユーザの検証および認証を行うためにＳＳＡにより使用される。証明書の例としては、パスワード認証用のパスワード／ＰＩＮ番号、ＡＥＳ認証用のＡＥＳ鍵等を挙げることができる。証明書（即ち、ＰＩＮ、対称鍵等）の種類／フォーマットは、事前に定義されており、認証モードに基づいている。即ち、それらはＡＣＲの生成時にＳＳＡシステムに供給される。ＳＳＡシステムは、こうした証明書の定義、分配および管理とは関係なく、ＰＫＩベースの認証では例外はあるものの、ＲＳＡ鍵ペアを生成するためにデバイス（例えば、フラッシュカード）を使用可能であり、証明書の生成に対して公開鍵をエクスポート可能である。

【００７１】

《パーミッション制御記録（ＰＣＲ）》

ＰＣＲは、ＳＳＡシステムにログインし、ＡＣＲの認証プロセスの通過に成功した後、エンティティに対して何が許可されるのかを示している。３タイプのパーミッションカテゴリ、即ち、パーティションおよび鍵に関する生成パーミッション、パーティションおよび鍵へのアクセスパーミッション、およびエンティティ・ＡＣＲ属性の管理パーミッションが存在する。

【００７２】

《パーティションへのアクセス》

ＰＣＲに関する当該セクションは、ＡＣＲ段階を首尾良く完了した場合にエンティティがアクセス可能な（ＳＳＡシステムに提供されているようにＩＤを使用する）パーティションのリストを含む。各パーティションのアクセスの種類は、書き込みのみ、または読み出しのみに制限されてもよいし、書き込み／読み出しのフルアクセス権限を指定してもよい。このように、図５のＡＣＲ＃１は、パーティション＃１ではなく、パーティション＃２へのアクセスを有している。ＰＣＲで特定される制約は、ＳＳＡパーティションおよびパブリックパーティションに適用する。

【００７３】

パブリックパーティションは、ＳＳＡシステムをホスティングするデバイス（例えば、フラッシュカード）に対する正規の読み出しおよび書き込みコマンド、または、ＳＳＡコマンドの何れかによりアクセス可能である。（以下で説明する）ルートＡＣＲがパブリックパーティションを制限するパーミッションを用いて生成される場合、当該ＡＣＲはそれを子に伝えることが可能である。ＡＣＲは、正規の読み出しおよび書き込みコマンドがパブリックパーティションにアクセスしないように制限することのみ可能であることが好ましい。ＳＳＡシステムにおけるＡＣＲは、生成に対してのみ制限可能であることが好ましい。ＡＣＲが一旦パブリックパーティションに対する読み出しおよび書き込みに関するパーミッションを有すると、当該パーミッションを取り消しできないことが好ましい。

【００７４】

《鍵ＩＤへのアクセス》

ＰＣＲに関する当該セクションは、ＡＣＲポリシーがエンティティのログインプロセスにより満たされた場合、（ホストによりＳＳＡシステムに供給されるように）エンティティがアクセス可能な鍵ＩＤのリストに関連付けられたデータを含む。指定された鍵ＩＤは、ＰＣＲに登場するパーティションに存在する一または複数のファイルに関連付けられる。鍵ＩＤはデバイス（例えば、フラッシュカード）の論理アドレスに関連付けられていないので、複数のパーティションが特定ＡＣＲと関連付けられる場合、ファイルはパーティションの何れか一つに存在することが可能である。ＰＣＲで指定された複数の鍵ＩＤは、それぞれ異なるセットの複数のアクセス権限を有することができる。鍵ＩＤにより指令さ

れたデータへのアクセスは、書き込みのみ、または読み出しのみに限定することが可能であり、または書き込み／読み出しのフルアクセス権限を指定しても良い。

【００７５】

《ＡＣＲ属性管理（ＡＣＲ　Ａｔｔｒｉｂｕｔｅｓ　Ｍａｎａｇｅｍｅｎｔ；ＡＣＡＭ）》

このセクションは、ＡＣＲのシステム属性が特定の状況で如何に変化し得るかを記載している。

【００７６】

ＳＳＡシステムにおいて許可し得るＡＣＡＭアクションは、以下の通りである：

【００７７】

ＡＧＰおよびＡＣＲの生成／削除／更新

【００７８】

パーティションおよび鍵の生成／削除

【００７９】

鍵およびパーティションへのアクセス権限の委譲

【００８０】

親ＡＣＲはＡＣＡＭパーミッションを編集できないことが望ましい。これは、望ましくはＡＣＲの削除および作り直しを必要とする。また、ＡＣＲにより生成された鍵ＩＤへのアクセスパーミッションを取り除くことはできないことが望ましい。

【００８１】

ＡＧＰおよびＡＣＲの生成／削除／更新

【００８２】

ＡＣＲは、他のＡＣＲおよびＡＧＰを生成する能力を有しても良い。また、ＡＣＲの生成は、クリエータが所有するＡＣＡＭパーミッションの一部またはすべてを生成するＡＣＲに委譲することを意味してもよい。ＡＣＲの生成パーミッションを有することは、以下のアクションに対するパーミッションを有することを意味する：

１．子の証明書の定義および編集、即ち、認証方法は、生成ＡＣＲにより一旦設定されると、編集できないことが好ましい。証明書は既に子に対して定義されている認証アルゴリズムの境界内で変更されてもよい。

２．ＡＣＲの削除

３．子ＡＣＲへの生成パーミッションの委譲（これにより子ＡＣＲは孫（ｇｒｏｕｎｄ　ｃｈｉｌｄｒｅｎ）を有す）

【００８３】

他のＡＣＲの生成パーミッションを有するＡＣＲは、（恐らくＡＣＲをブロック解除するパーミッションを有していないが）当該ＡＣＲが生成するＡＣＲに対してブロック解除するパーミッションを委譲するパーミッションを有する。親ＡＣＲは自身の非ブロッカーへの参照を子ＡＣＲに設置する。

【００８４】

親ＡＣＲは自身の子ＡＣＲを削除するパーミッションを有する唯一のＡＣＲである。ＡＣＲが自身の生成した低位レベルのＡＣＲを削除する場合、自動的に当該低位レベルのＡＣＲにより生み出されたすべてのＡＣＲも同様に削除される。ＡＣＲが削除される場合、それが生成したすべての鍵ＩＤおよびパーティションが削除される。

【００８５】

下記の通り、ＡＣＲが自身の記録を更新可能な２つの例外が存在する。

【００８６】

パスワード／ＰＩＮは、クリエータＡＣＲにより設定されているのだが、それらを含むＡＣＲによってのみ更新可能である。

【００８７】

ルートＡＣＲは自身およびそれが存在するＡＧＰを削除してもよい。

【００８８】

《鍵およびパーティションへのアクセス権限の委譲》
ＡＣＲおよびそれらのＡＧＰは、階層ツリーにまとめられる。当該階層ツリーでは、ルートＡＧＰおよびその中のＡＣＲがツリーの先端部に位置する（例えば、図６のルートＡＧＰ１３０および１３２）。ＳＳＡシステムにはいくつかのＡＧＰツリーが存在することができ、それらは互いに完全に分離している。ＡＧＰ内のＡＣＲは、自身が存在する同一ＡＧＰ内のすべてのＡＣＲ、およびそれらが生成したすべてのＡＣＲに対して、鍵へのアクセスパーミッションを委譲することができる。鍵の生成パーミッションは、鍵を使用するためのアクセスパーミッションを委譲するパーミッションを含むことが好ましい。
鍵へのパーミッションは以下の３つのカテゴリに分割される。
１．アクセス（これは鍵に対するアクセスパーミッション、即ち、読み出し、書き込みを定義する）
２．所有権（鍵を生成したＡＣＲが明らかにその所有者となる）　当該所有権は、（ＡＣＲが同一ＡＧＰまたは子ＡＧＰに存在するという条件で）あるＡＣＲから他のＡＣＲに委譲される。鍵の所有権は、他のＡＣＲにパーミッションを委譲するだけでなく、それを削除するパーミッションを与える。
３．アクセス権限の委譲（このパーミッションにより、ＡＣＲは自身が保有する権利を委譲可能となる）
【００８９】
ＡＣＲは、自身がアクセス権限を有する他のパーティションだけでなく、自身が生成したパーティションへのアクセス権限を委譲することが可能である。
【００９０】
パーミッションの委譲は、指定されたＡＣＲのＰＣＲにパーティションおよび鍵ＩＤ名を追加することにより実行される。鍵へのアクセスパーミッションの委譲は、鍵ＩＤによるものであってもよいし、アクセスパーミッションが委譲するＡＣＲのすべての生成された鍵に対するパーミッションであるという主張によるものであってもよい。
【００９１】
《ＡＣＲのブロックおよびブロック解除》
ＡＣＲは、エンティティのシステムとのＡＣＲ認証プロセスが失敗した場合に、インクリメントするブロッキングカウンタ（ｂｌｏｃｋｉｎｇ　ｃｏｕｎｔｅｒ）を有しても良い。認証の失敗回数が所定の最大値（ＭＡＸ）に達した場合、ＡＣＲはＳＳＡシステムによりブロック（ｂｌｏｃｋ）される。
【００９２】
ブロックされたＡＣＲは、ブロックされたＡＣＲが参照する他のＡＣＲによりブロック解除（ｕｎｂｌｏｃｋ）可能である。ブロック解除を行うＡＣＲを参照することは、当該ＡＣＲのクリエータにより設定される。ブロックを解除するＡＣＲは、ブロックされたＡＣＲのクリエータとして同一ＡＧＰに存在し、「ブロック解除」のパーミッションを有することが好ましい。
【００９３】
当該システムにおいて、ブロックされたＡＣＲをブロック解除可能なＡＣＲは、他には存在しない。ＡＣＲは、ブロック解除ＡＣＲ（ｕｎｂｌｏｃｋｅｒ　ＡＣＲ）を用いずに、ブロッキングカウンタを用いて構成されてもよく、この場合、当該ＡＣＲがブロックされた場合には、それをブロック解除することはできない。
【００９４】
《ルートＡＧＰ―アプリケーションデータベースの生成》
ＳＳＡシステムは、複数のアプリケーションを処理し、各アプリケーションのデータを分離するように設計されている。ＡＧＰシステムのツリー構造は、アプリケーション固有のデータを識別および分離するために使用される主なツールである。ルートＡＧＰは、アプリケーションＳＳＡデータベースツリーの先端部に存在し、幾分異なる行動規範に従う。幾つかのルートＡＧＰはＳＳＡシステムにおいて設定可能である。２つのルートＡＧＰ１３０および１３２が図６に示されている。これより少ない、または多くのＡＧＰを使用

しても良く、又、このことが本発明の範囲内にあることは明白である。

【００９５】

新たなアプリケーションに対するデバイス（例えば、フラッシュカード）の登録、および／または、デバイスに対する新たなアプリケーションの証明書の発行は、新たなＡＧＰ／ＡＣＲツリーをデバイスに追加するプロセスを通じて実行される。

【００９６】

ＳＳＡシステムは、（ルートＡＧＰの全てのＡＣＲおよびそれらのパーミッションだけでなく）ルートＡＧＰの生成に関する３つの異なるモードをサポートしている：

１．オープンモード（Ｏｐｅｎ）：如何なる種類の認証も必要としない任意のユーザまたはエンティティ、または（以下に説明する）システムＡＣＲを通じて認証されたユーザ／エンティティは、新たなルートＡＧＰを生成することができる。オープンモードは、すべてのデータ伝送がオープンチャネル（即ち、発行機関（ｉｓｓｕａｎｃｅ ａｇｅｎｃｙ）の安全な環境の中）で実行される間に、何もセキュリティ手段を使用することなく、または、システムＡＣＲ認証（即ち、無線（Ｏｖｅｒ Ｔｈｅ Ａｉｒ；ＯＴＡ）および事後発行手順（ｐｏｓｔ ｉｓｓｕａｎｃｅ ｐｒｏｃｅｄｕｒｅ））を通じて確立されたセキュアチャネルを通じて、ルートＡＧＰの生成を可能にする。

システムＡＣＲが設定されず（これは随意的な機能である）、ルートＡＧＰ生成モードがオープンに設定されている場合、オープンチャネルオプション（ｏｐｅｎ ｃｈａｎｎｅｌ ｏｐｔｉｏｎ）のみが利用可能である。

２．制御モード（Ｃｏｎｔｒｏｌｌｅｄ）：システムＡＣＲを通じて認証されたエンティティのみが新たなルートＡＧＰを生成することができる。ＳＳＡシステムは、システムＡＣＲが設定されていない場合、このモードに設定することができない。

３．ロックモード（Ｌｏｃｋｅｄ）：ルートＡＧＰの生成が不可能であり、増設ルートＡＧＰをシステムに追加することができない。

２つのＳＳＡコマンドがこの機能を制御する（これらのコマンドは認証無しで任意のユーザ／エンティティに利用可能である）：

１．方法設定コマンド（Ｍｅｔｈｏｄ ｃｏｎｆｉｇｕｒａｔｉｏｎ ｃｏｍｍａｎｄ）－３つのルートＡＧＰ生成モードの何れか一つを使用するようにＳＳＡシステムを設定するために使用される。以下のモード変更のみが許可される：オープンモード -> 制御モード（Ｏｐｅｎ －＞ Ｃｏｎｔｒｏｌｌｅｄ）、制御モード－＞ロックモード（Ｃｏｎｔｒｏｌｌｅｄ－＞Ｌｏｃｋｅｄ）（即ち、ＳＳＡシステムが制御モードに現在設定されている場合、ロックモードへの変更のみ可能である）。

２．方法設定ロックコマンド（Ｍｅｔｈｏｄ ｃｏｎｆｉｇｕｒａｔｉｏｎ ｌｏｃｋ ｃｏｍｍａｎｄ）－方法設定コマンドを無効にし、現在選択されているモードを永久にロックするために使用される。

【００９７】

ルートＡＧＰが生成される時、（ルートＡＧＰの生成に適用されたのと同一のアクセス制限を使用して）ＡＣＲの生成および設定を可能にするのは、特別の初期化モードにおいてである。ルートＡＧＰ設定プロセスの最後に、エンティティが明確にそれをオペレーションモードに切り替えると、既存のＡＣＲはそれ以後の更新が不可能となり、追加のＡＣＲを生成することが不可能となる。

【００９８】

ルートＡＧＰが一旦標準モードになると、ルートＡＧＰは、ルートＡＧＰの削除パーミッションを与えられたルートＡＧＰ内のＡＣＲの１つを通じてシステムにログインすることによってのみ、削除可能となる。特別の初期化モードに加えて、これはルートＡＧＰのもう一つの例外なのである；即ち、ツリー内の次のレベルのＡＧＰとは対照的に、自身が属するＡＧＰの削除パーミッションを有するＡＣＲを含むことが可能であるのは、当該ＡＧＰのみであることが好ましい。

【００９９】

ルートＡＣＲと標準的なＡＣＲとの３つ目の、且つ、最後の相違点は、ルートＡＣＲが

、パーティションの生成および削除パーミッションを有することができるシステム内における唯一のＡＣＲであるという点である。

【０１００】

《ＳＳＡシステムＡＣＲ》

システムＡＣＲは以下の２つのＳＳＡオペレーションに使用することができる：

１．悪意のある環境における保護チャネルの保護下でのＡＣＲ／ＡＧＰツリーの生成。

２．ＳＳＡシステムをホスティングするデバイスの識別および認証。

【０１０１】

ＳＳＡにシステムＡＣＲが１つだけ存在することが好ましく、一旦定義されたら、変更不可能であることが好ましい。システムＡＣＲの生成時にはシステム認証を必要としない；即ち、ＳＳＡコマンドのみが必要なのである。システムＡＣＲ生成機能を無効にすることが可能である（ルートＡＧＰ生成機能についても同様）。一つのシステムＡＣＲのみが許可されることが好ましいので、システムＡＣＲの生成後には、システムＡＣＲ生成コマンドは効果を持たない。

【０１０２】

生成プロセスにおいては、システムＡＣＲは使用することができない。完了した時には、システムＡＣＲが生成され、準備完了であることを示す特別のコマンドを発行する必要がある。これより後の時点で、システムＡＣＲの更新または差し替えは不可能であることが好ましい。

【０１０３】

システムＡＣＲはＳＳＡにおいてルートＡＣＲ／ＡＧＰを生成する。システムＡＣＲは、ホストがそれに満足し、ブロックする時までは、ルートレベルを追加／変更するパーミッションを有する。ルートＡＧＰをブロックすることは、本質的には、システムＡＣＲとの接続を遮断し、それを改ざん防止状態にすることである。この時点で、ルートＡＧＰおよびその中のＡＣＲを変更／編集することができるものはない。これはＳＳＡコマンドを通じて実行される。ルートＡＧＰの生成を無効にすることは、永久的な効果があり、覆すことはできない。システムＡＣＲを含む上記の機能は、図７において説明されている。システムＡＣＲは３つの異なるルートＡＧＰを生成するために使用される。これらが生成された後の一定の時間にシステムＡＣＲからルートＡＧＰをブロックするために、ＳＳＡコマンドがホストから送信され、これにより、図７においてシステムＡＣＲをルートＡＧＰに接続する破線で示しているように、ルートＡＧＰ生成機能が無効となる。これにより、３つのルートＡＧＰが改ざん防止の状態になる。３つのルートＡＧＰは、ルートＡＧＰがブロックされる前でもされた後でも、３つの別々のツリーを形成する子ＡＧＰを生成することに使用することができる。

【０１０４】

上記機能は、コンテンツを備えたセキュア製品の設定において、コンテンツ所有者に高い柔軟性をもたらす。セキュア製品は、「発行される（Ｉｓｓｕｅｄ）」必要がある。発行とは、デバイスによるホストの識別およびホストによるデバイスの識別が可能な識別鍵を設置するプロセスである。デバイス（例えば、フラッシュカード）を識別することにより、ホストは、識別鍵を備えた秘密を信用できるか否かを決定することができる。一方、ホストを識別することにより、デバイスは、ホストが許可される場合にのみ、セキュリティポリシーを実行することができる（特定のホストコマンドを許可し、実行する）。

【０１０５】

複数のアプリケーションを扱うように設計された製品は、幾つかの識別鍵を有する。当該製品は「事前発行」（ｐｒｅ－ｉｓｓｕａｒａｎｃｅ）、即ち、鍵を発送前の製造工程で保存すること、または、「事後発行」（ｐｏｓｔ－ｉｓｓｕｒａｎｃｅ）、即ち、新しい鍵を発送後に追加することが可能である。事後発行に関しては、メモリデバイス（例えば、メモリカード）は、何らかのマスタ（ｍａｓｔｅｒ）、またはデバイスへのアプリケーションの追加を許可されたエンティティの識別に使用されているデバイスレベル鍵を含む必要がある。

【０１０６】
上記機能により、製品は事後発行の有効化／無効化を設定することができる。さらに、事後発行設定が、発送後に安全に実施することができる。デバイスは、上記のマスタまたはデバイスレベル鍵の他には、鍵を備えていない小売製品として購入され、その後に、新たな所有者により、さらに事後発行アプリケーションの有効化または無効化の何れかが設定されてもよい。
【０１０７】
このように、システムＡＣＲ機能により、上記目的を達成するための以下の機能が与えられる。
・システムＡＣＲを備えていないメモリデバイスは、無制限かつ規制のないアプリケーションの追加を許可する。
・システムＡＣＲを備えていないメモリデバイスは、システムＡＣＲの生成を無効にするように設定可能である。これは、（新たなルートＡＧＰの生成機能が同様に無効にならない限り）新たなアプリケーションの追加を制御する方法がないことを意味する。
・システムＡＣＲを備えたメモリデバイスは、セキュアチャネルを経由する制御されたアプリケーションの追加のみを、システムＡＣＲ証明書を使用する認証手続きを通じて確立することができる。
・システムＡＣＲを備えたメモリデバイスは、アプリケーションの追加前または追加後に、アプリケーション追加機能を無効にするように設定してもよい。
【０１０８】
《鍵ＩＤリスト》
鍵ＩＤは、特定のＡＣＲ要求毎に生成される。しかし、メモリシステム１０においては、鍵ＩＤはＳＳＡシステムのみにより使用される。鍵ＩＤの生成時に、以下のデータは生成するＡＣＲにより提供される、または生成するＡＣＲに対して提供される。
１．鍵ＩＤ。ＩＤはホストを通じてエンティティにより提供され、鍵と、さらなる読み出しアクセスまたは書き込みアクセスにおいて、鍵を使用して暗号化または復号されるデータを参照するために使用される。
２．鍵暗号化およびデータ完全性モード（上記のブロックモード、チェーンモードおよびハッシュモードと下記の説明の通り）
【０１０９】
ホストにより提供された属性に加えて、以下のデータはＳＳＡシステムによって維持される。
１．鍵ＩＤ所有者。所有者であるＡＣＲのＩＤ。鍵ＩＤの生成時に、クリエータＡＣＲがその所有者となる。しかし、鍵ＩＤの所有権は、他のＡＣＲに譲渡されてもよい。鍵ＩＤの所有者のみが、鍵ＩＤの所有権の譲渡（ｔｒａｎｓｆｅｒ）および委譲（ｄｅｌｅｇａｔｅ）を許可されることが好ましい。関連する鍵へのアクセスパーミッションの委譲、およびこれらの権利の無効化は、鍵ＩＤ所有者または委譲パーミッションを与えられた任意の他のＡＣＲの何れかにより実行可能である。これらのオペレーションの何れか一つを実行するための試みがなされる時は常に、要求するＡＣＲに権限が与えられている場合に限って、ＳＳＡシステムが当該試みを許可する。
２．ＣＥＫ。これは、鍵ＩＤと関連付けられ、鍵ＩＤにより示されたコンテンツを暗号化するために使用されるＣＥＫである。ＣＥＫは、ＳＳＡシステムによって生成された１２８ビットのＡＥＳ乱数鍵であってもよい。
３．ＭＡＣおよびＩＶ値（ＩＶ　ｖａｌｕｅｓ）。チェーンブロック暗号（ＣＢＣ）暗号アルゴリズムで使用される動的情報（メッセージ認証コードおよび初期化ベクトル）。
【０１１０】
また、ＳＳＡの様々な機能が図８Ａ乃至１６のフローチャートを参照して説明されており、当該図面において、ステップの左側に記載された「Ｈ」はオペレーションがホストにより実行されることを意味し、「Ｃ」はオペレーションがカードにより実行されることを意味する。システムＡＣＲを生成するために、ホストはメモリデバイス１０のＳＳＡに対

して、システムＡＣＲを生成するコマンドを発行する（矩形２０２）。デバイス１０は、システムＡＣＲが既に存在するか否かをチェックすることにより対応する（矩形２０４、菱形２０６）。システムＡＣＲが既に存在する場合、デバイス１０は失敗ステータスを返し、ストップする（楕円２０８）。システムＡＣＲが存在しない場合、メモリ１０はシステムＡＣＲの生成が許可されるか否かを確認するためにチェックを行い（菱形２１０）、許可されない場合、失敗ステータスを返す（矩形２１２）。このように、例えば、システムＡＣＲを必要としないように、必要なセキュリティ機能が予め定められている場合、デバイス発行者がシステムＡＣＲの生成を許可しないこともある。ＡＣＲの生成が許可される場合、デバイス１０はＯＫステータスを返し、ホストからのシステムＡＣＲ証明書を待つ（矩形２１４）。ホストは、ＳＳＡのステータスをチェックし、デバイス１０がシステムＡＣＲの生成が許可されることを示しているか否かをチェックする（矩形２１６および菱形２１８）。生成が許可されない場合、またはシステムＡＣＲが既に存在する場合、ホストはストップする（楕円２２０）。デバイス１０がシステムＡＣＲの生成を許可している場合、ホストはそのログイン証明書を定義するためのＳＳＡコマンドを発行し、これをデバイス１０に送信する（矩形２２２）。デバイス１０は受信した証明書を使用してシステムＡＣＲ記録を更新し、ＯＫのステータスを返す（矩形２２４）。このステータス信号に対応して、ホストはシステムＡＣＲの準備完了を示すＳＳＡコマンドを発行する（矩形２２６）。デバイス１０は、システムＡＣＲの更新または差し換えができないように、システムＡＣＲをロックすることにより対応する（矩形２２８）。これは、システムＡＣＲの機能およびホストに対してデバイス１０を識別するための身元を確定する。

【０１１１】

新たなツリー（新たなルートＡＧＰ（Ｎｅｗ　Ｒｏｏｔ　ＡＧＰｓ）およびＡＣＲ）の生成手順は、これらの機能がデバイスに設定される方法により決定される。図９はこの手順を説明している。ホスト２４およびメモリシステム１０の両方がこれに従う。新たなルートＡＧＰの追加が完全に無効となる場合、新たなルートＡＧＰを追加することはできない（菱形２４６）。新たなＡＧＰの追加が有効であるが、システムＡＣＲを必要とする場合、ホストはルートＡＧＰ生成コマンドの発行前に（矩形２５４）、システムＡＣＲを通じて認証し、セキュアチャネルを確立する（菱形２５０、矩形２５２）。システムＡＣＲを必要としない場合（菱形２４８）、ホスト２４は認証なしでルートＡＧＰ生成コマンド（Ｒｏｏｔ　ＡＧＰ　ｃｏｍｍａｎｄ）を発行することが可能であり、矩形２５４に進む。システムＡＣＲが存在する場合には、システムＡＣＲを必要としない場合であっても（フローチャートには図示せず）、ホストはこれを使用してもよい。デバイス（例えば、フラッシュカード）は、上記機能が無効の場合には、新たなルートＡＧＰを生成する如何なる試みも拒否し、システムＡＣＲを必要とする場合には、認証なしで新たなルートＡＧＰを生成する試みを拒否する（菱形２４６および２５０）。矩形２５４で新たに生成されたＡＧＰおよびＡＣＲは、ＡＧＰ内のＡＣＲの更新、さもなければ変更ができないように、ここでオペレーションモードに切り替えられる。これにより、ＡＣＲをさらに追加できないようになる（矩形２５６）。そして、さらにルートＡＧＰを生成できないように、システムが選択可能にロックされる（矩形２５８）。矩形２５８は、このステップが選択可能なステップであることを示すために、慣例として破線で図示されている。本出願の図面のフローチャートにおいて破線で示す矩形のすべては選択的なステップである。これにより、コンテンツ所有者は、合法的なコンテンツを有する本物のメモリデバイスを模倣し得る他の違法目的のデバイス１０のユーザをブロックすることが可能となる。

【０１１２】

図１０に示されているように、（上記の通り、ルートＡＧＰ内のＡＣＲ以外の）ＡＣＲを生成するために、ＡＣＲを生成する権利を有する任意のＡＣＲから開始してもよい（矩形２７０）。エンティティは、エントリポイントＡＣＲアイデンティティ（ｅｎｔｒｙ　ｐｏｉｎｔ　ＡＣＲ　ｉｄｅｎｔｉｔｙ）と、それが生成したいあらゆる必要な属性を備えたＡＣＲと、を提供することにより、ホスト２４を通じてツリーへのエントリを試みるかもしれない（矩形２７２）。ＳＳＡは、ＡＣＲの身元との照合をチェックし、当該身元

を有するＡＣＲがＡＣＲの生成パーミッションを有するか否かをチェックする（菱形２７４）。当該要求に権限があると証明された場合、デバイス１０のＳＳＡはＡＣＲを生成する（矩形２７６）。

【０１１３】

図１１は、図１０の方法を使用するセキュリティアプリケーションにおいて有用なツリーを説明する２つのＡＧＰを示している。このように、マーケティングＡＧＰにおいて身元ｍ１を備えたＡＣＲは、ＡＣＲの生成パーミッションを有する。また、ＡＣＲ　ｍ１は、鍵ＩＤ「マーケティング情報（Ｍａｒｋｅｔｉｎｇ　Ｉｎｆｏｒｍａｔｉｏｎ）」に関連付けられたデータ、および、鍵ＩＤ「価格リスト（Ｐｒｉｃｅ　Ｌｉｓｔ）」に関連付けられたデータを読み出しおよび書き込みするための鍵の使用パーミッションを有する。図１０の方法を使用して、ＡＣＲ　ｍ１は２つのＡＣＲを備えたセールスＡＧＰを生成するが、ここで、２つのＡＣＲとは、ｓ１およびｓ２であり、鍵ＩＤ「マーケティング情報」に関連付けられたデータへのアクセスに必要な鍵ではなく、鍵ＩＤ「価格リスト」に関連付けられた価格データにアクセスするための鍵の読み出しパーミッションのみを備えている。この方法で、ＡＣＲｓ　ｓ１およびｓ２を備えたエンティティは、価格データの読み出しのみが可能であり、同データを変更することはできず、マーケティングデータへのアクセスを有していない。一方、ＡＣＲ　ｍ２は、ＡＣＲの生成パーミッションを有していないが、鍵ＩＤ「価格リスト」および鍵ＩＤ「マーケティング情報」に関連付けられたデータにアクセスするための鍵の読みだしパーミッションのみを有している。

【０１１４】

このように、ｍ１がｓ１およびｓ２に対して価格データの読み出し権を委譲するという上述の方法で、アクセス権限が委譲されてもよい。大規模なマーケティングおよびセールスグループが関与している状況では、特にこの方法は有用である。営業担当者が存在するが、その人数が１名または数名の場合には、図１０の方法を使用する必要はない。その代わりに、図１２に図示すように、ＡＣＲが同一ＡＧＰ内の低位レベルまたは同一レベルのＡＣＲにアクセス権限を委譲してもよい。第一に、上述の方法でホストを通じてツリー上のＡＣＲを特定することにより、エンティティはそうしたＡＧＰに対するツリーに入る（矩形２８０）。次に、ホストはＡＣＲおよび委譲する権利を特定する。ＳＳＡは、当該ＡＣＲに対する一または複数のツリーをチェックし、ＡＣＲが特定された他のＡＣＲに権利を委譲するパーミッションを有するか否かをチェックする（菱形２８２）。パーミッションを有する場合、権利は委譲され（矩形２８４）、パーミッションを有さない場合には、権利委譲をストップする。この結果が図１３で説明されている。この場合、ＡＣＲ　ｍ１はＡＣＲ　ｓ１に対して読み出しパーミッションを委譲するパーミッションを有しており、ｓ１はパーミッションの委譲後に価格データにアクセスする鍵を使用することが可能となる。これは、ｍ１が価格データにアクセスする権利と同一またはより高位レベルの権利を有し、このように委譲するパーミッションを有する場合に実行されてもよい。一実施形態に於いて、ｍ１はパーミッションの委譲後に当該アクセス権限を保有する。アクセス権限は、好ましくは、例えば限られた時間、限られたアクセス回数等、（その後、むしろ永遠に）制限された状況下で委譲されてもよい。

【０１１５】

図１４には、鍵および鍵ＩＤの生成プロセスが示されている。エンティティがＡＣＲを通じて認証を行う（矩形３０２）。エンティティがホストにより特定されたＩＤを備えた鍵の生成を要求する（矩形３０４）。ＳＳＡは、特定されたＡＣＲが鍵の生成パーミッションを有するか否かをチェックし確認する（菱形３０６）。例えば、鍵が特定のパーティションにおけるデータにアクセスするために使用される場合、ＳＳＡはＡＣＲが当該パーティションにアクセス可能か否かをチェックし確認する。ＡＣＲが権限を有する場合、メモリデバイス１０はホストにより供給された鍵ＩＤに関連付けられた鍵値を生成し（矩形３０８）、鍵ＩＤをＡＣＲに保存し、鍵値をそのメモリ（コントローラ関連メモリまたはメモリ２０）に保存する。そして、エンティティにより供給された情報に従って、権利およびパーミッションを割り当て（矩形３１０）、当該割り当てられた権利およびパーミッ

ションを用いて、当該ＡＣＲのＰＣＲを修正する（矩形３１２）。このように、鍵のクリエータは、例えば、読み出しおよび書き込みパーミッション、委譲権および同一ＡＧＰの他のＡＣＲまたは低位レベルのＡＣＲとの共有権、鍵の所有権の譲渡権など、あらゆる利用可能な権利を有する。

【０１１６】

図１５に図示すように、ＡＣＲはＳＳＡシステムにおいて他のＡＣＲのパーミッション（またはその存在すべて）を変更することが可能である。エンティティは、既に述べたように、ＡＣＲを通じてツリーに入ることが可能である。即ち、一例ではエンティティが認証されると、ＡＣＲを特定する（矩形３３０、３３２）。エンティティは、ターゲットＡＣＲの削除またはターゲットＡＣＲにおけるパーミッションを要求する（矩形３３４）。特定されたＡＣＲまたはこの時点でアクティブなＡＣＲが当該要求に関する権利を有する場合（菱形３３６）、ターゲットＡＣＲが削除されるか、ターゲットＡＣＲのＰＣＲが当該パーミッションを削除するために変更される（矩形３３８）。これが権限を与えられない場合、システムはストップする。

【０１１７】

上記のプロセス後、ターゲットは、当該プロセス以前にアクセス可能であったデータにアクセスすることができなくなる。図１６に示されているように、エンティティはターゲットＡＣＲでツリーに入ろうとするかもしれないが（矩形３５０）、以前存在していたＡＣＲのＩＤがＳＳＡにもはや存在していないので、認証プロセスに失敗していることが判明する。その結果、アクセス権限は拒否される（菱形３５２）。ＡＣＲのＩＤが削除されていないことを想定した場合、エンティティはＡＣＲを特定し（矩形３５４）、特定のパーティションにおける鍵ＩＤおよび／またはデータを特定する（矩形３５６）。そして、ＳＳＡはそうしたＡＣＲのＰＣＲにしたがって、鍵ＩＤまたはパーティションアクセス要求が許可されたか否かを確認するためにチェックする（菱形３５８）。パーミッションが削除されている、または期限切れとなっている場合、当該要求は再び拒否される。それ以外の場合、当該要求は認められる（矩形３６０）。

【０１１８】

上記のプロセスは、ＡＣＲおよびＡＣＲのＰＣＲが他のＡＣＲによって変更されたかどうか、または、初めからそのように設定されていたかどうかに関係なく、保護データへのアクセスがデバイス（例えば、フラッシュカード）によってどのように管理されるかを説明している。

【０１１９】

《セッション》

ＳＳＡシステムは、同時にログインした複数ユーザを処理するように設計されている。この機能は、ＳＳＡにより受信された各コマンドが特定のエンティティに関連付けられており、このエンティティの認証に使用されたＡＣＲが要求されたアクションに対するパーミッションを有する場合にのみ実行されることを必要とする。

【０１２０】

複数のエンティティはセッションコンセプト（ｓｅｓｓｉｏｎ　ｃｏｎｃｅｐｔ）を通じてサポートされる。セッションは認証プロセス中に確立され、ＳＳＡシステムによりセッションＩＤが割り当てられる。セッションＩＤは、システムへのログインに使用されるＡＣＲと内部的に関連付けられており、さらなるＳＳＡコマンドすべてにおいて使用されるエンティティに対してエクスポートされる。

【０１２１】

ＳＳＡシステムは以下の２種類のセッションをサポートする。すなわち、オープンセッション（ｏｐｅｎ　ｓｅｓｓｉｏｎ）とセキュアセッション（ｓｅｃｕｒｅ　ｓｅｓｓｉｏｎ）である。特定の認証プロセスに関連付けられたセッションタイプは、ＡＣＲで定義される。ＳＳＡシステムは、認証自体を実行する方法と類似の方法でセッションの確立を実行する。ＡＣＲはエンティティのパーミッションを定義するので、このメカニズムにより、システム設計者はセキュアトンネル（ｓｅｃｕｒｅ　ｔｕｎｎｅｌｉｎｇ）を、特定

の鍵ＩＤへのアクセスと、特定のＡＣＲ管理オペレーションの呼び出し（ｉｎｖｏｋｉｎｇ）（即ち、新しいＡＣＲの生成および証明書の設定）と、の何れかと関連付けることが可能になる。

【０１２２】

《オープンセッション》

オープンセッションとは、バス暗号化を使用せずに、セッションＩＤを使用して識別されるセッションであり、すべてのコマンドおよびデータが問題なく通過する。このオペレーションモードは、エンティティが脅威モデルの一部ではなく、バスの盗聴もしていない、複数ユーザまたは複数エンティティの環境で使用されることが好ましい。

【０１２３】

データの伝送保護も、ホスト側アプリケーション間の効果的なファイアウォールの有効化も行わないが、オープンセッションモードにより、ＳＳＡシステムは、現状で認証されたＡＣＲに許可された情報のみへのアクセスを許可することができる。

【０１２４】

また、オープンセッションは、パーティションまたは鍵の保護が必要となる場合に使用することが可能である。しかし、妥当な認証プロセス後に、アクセスはホストにおけるすべてのエンティティに認められる。認証されたＡＣＲのパーミッションを獲得するために様々なホストアプリケーションが唯一共有する必要があるのは、セッションＩＤである。これは図１７Ａにおいて説明されている。ライン部４００より上に記載されているステップは、ホスト２４が実行するステップである。エンティティは、ＡＣＲ１に対して認証された後（矩形４０２）、メモリデバイス１０において鍵ＩＤＸに関連付けられたファイルへのアクセスを要求する（矩形４０４、４０６、４０８）。ＡＣＲ１のＰＣＲが当該アクセスを許可する場合、デバイス１０は当該要求を認める（菱形４１０）。それ以外の場合、システムは矩形４０２に戻る。　認証完了後、メモリシステム１０は、割り当てられたセッションＩＤ（ＡＣＲ証明書ではない）によってのみコマンドを発行するエンティティを識別する。オープンセッションにおいて、一旦ＡＣＲ１がそのＰＣＲにおける鍵ＩＤに関連付けられたデータへのアクセスを獲得すると、任意の他のアプリケーションまたはユーザが、ホスト２４における異なるアプリケーション間で共有される正しいセッションＩＤを特定することにより、同一データにアクセスすることが可能である。この機能がアプリケーションで有益であるのは、一度だけでログイン可能であり、且つ、異なるアプリケーションに対してログインが実行されるアカウントに関連付けられたすべてのデータにアクセス可能であるので、ユーザの利便性がより高いという点である。このように、携帯電話のユーザは、ログインを複数回数行うことなく、保存されたＥメールにアクセスし、メモリ２０に保存された音楽を聴くことが可能であってもよい。一方、ＡＣＲ１に含まれていないデータにはアクセス不可能である。このように、同一の携帯電話ユーザは、別のアカウントＡＣＲ２を通じてアクセス可能な、例えばゲームおよび写真等の有益なコンテンツを有するかもしれない。ユーザが自身の最初のアカウントＡＣＲ１を通じて利用可能なデータに他者がアクセスすることを気にしなくてよいとしても、ＡＣＲ２を通じて利用可能なコンテンツは、ユーザが自身の電話を借りる他者がアクセスすることを望まないデータなのである。オープンセッションにおいてＡＣＲ１へのアクセスを許可しつつ、データへのアクセスを２つの別々のアカウントに分割することにより、価値のあるデータの保護を利用可能にするだけでなく、利便性も提供する。

【０１２５】

ホストアプリケーション間のセッションＩＤを共有するプロセスをさらに容易にするために、ＡＣＲがオープンセッションを要求する際に、ＡＣＲはセッションに「０（ゼロ）」ＩＤを割り当てることを具体的に要求することが可能である。このように、アプリケーションを事前に定義されたセッションＩＤを使用するように設計することが可能である。明白な理由から、唯一の制約は、セッション０を要求する一つのＡＣＲだけを特定の時間に認証可能であるということである。セッション０を要求する他のＡＣＲを認証する試みは拒否される。

【０１２６】
《セキュアセッション（ｓｅｃｕｒｅ　ｓｅｓｓｉｏｎ）》
図１７Ｂに示すように、セキュリティレイヤ（ｌａｙｅｒ　ｏｆ　ｓｅｃｕｒｉｔｙ）を追加するために、セッションＩＤを使用してもよく、この場合、メモリ１０はアクティブなセッションのセッションＩＤを保存する。図１７Ｂにおいて、例えば、鍵ＩＤ　Ｘに関連付けられたファイルへのアクセスを可能にするために、エンティティは、ファイルへのアクセスを許可される前に、例えばセッションＩＤ「Ａ」等のセッションＩＤを提供する必要もある（矩形４０４、４０６、４１２および４１４）。この方法では、要求するエンティティが正しいセッションＩＤを知らなければ、エンティティはメモリ１０にアクセスすることができない。セッション終了後にセッションＩＤは削除され、セッションＩＤがセッション毎に異なるので、エンティティはセッション番号を供給可能な状況にある場合に限りアクセスを獲得することができる。
【０１２７】
ＳＳＡシステムは、セッション番号を使用する以外に、実際にコマンドが適切に認証されたエンティティから送信されていることを確かめる方法がない。攻撃者が悪意のあるコマンドを送信するためにオープンチャネルの使用を試みる脅威が存在する場合でのアプリケーションの使用に際して、ホストアプリケーションはセキュアセッション（セキュアチャネル）を使用する。
【０１２８】
セキュアチャネルを使用する場合、コマンド全体だけでなく、セッションＩＤもセキュアチャネル暗号（セッション）鍵で暗号化され、セキュリティレベルはホスト側で実行されているセキュリティレベルと同程度に高い。
【０１２９】
《セッションの終了》
下記のシナリオの内、何れか一つのシナリオでセッションは終了し、ＡＣＲはログオフされる。
１．エンティティが明確なセッション終了コマンドを発行する。
２．通信時のタイムアウト。特定のエンティティが、ＡＣＲパラメータの一つとして定義された期間（ｔｉｍｅ　ｐｅｒｉｏｄ）において、コマンドを発行しなかった。
３．デバイス（例えば、フラッシュカード）の再起動および／または電源の入れ直し後、すべてのオープンセッションが終了する。
【０１３０】
《データ完全性サービス（ｄａｔａ　ｉｎｔｅｇｒｉｔｙ　ｓｅｒｖｉｃｅｓ）》
ＳＳＡシステムは、（ＡＣＲ，ＰＣＲ等のすべてを含む）ＳＳＡデータベースの完全性を検証する。さらに、データ完全性サービスが鍵ＩＤメカニズムを通じてエンティティデータに提供される。
【０１３１】
鍵ＩＤに暗号アルゴリズムとしてハッシュアルゴリズムが組み込まれている場合、ハッシュ値（ｈａｓｈ　ｖａｌｕｅ）がＣＥＫおよびＩＶとともにＣＥＫ記録に保存される。ハッシュ値は書き込みオペレーション中に算出され、保存される。ハッシュ値は読み出しオペレーション中に再び算出され、以前の書き込みオペレーション中に保存されたハッシュ値と比較される。エンティティが鍵ＩＤにアクセスする度に、追加のデータが古いデータおよび更新された（読み出しまたは書き込みに対する）適切なハッシュ値に（暗号方法的に）連結される。
【０１３２】
ホストのみが、鍵ＩＤに関連付けられた、または鍵ＩＤにより示されたデータファイルを区別しているので、ホストは以下に示す方法で、データ完全性機能の幾つかの面を明確に管理する：
１．鍵ＩＤに関連付けられた、または鍵ＩＤにより示されたデータファイルが最初から最後まで書き込まれ、読み出される。ＳＳＡシステムがＣＢＣ暗号法を使用し、データ全

体のハッシュ化メッセージダイジェスト（ｈａｓｈｅｄ　ｍｅｓｓａｇｅ　ｄｉｇｅｓｔ）を生成するので、ファイルの一部にアクセスする任意の試みは失敗する。

２．中間ハッシュ値がＳＳＡシステムにより維持されるので、連続ストリームにおいてデータを処理する必要はない（データストリームは他の鍵ＩＤのデータストリームと交互配置する（ｉｎｔｅｒｌｅａｖｅｄ）ことができ、複数セッションに亘って分割してもよい）。しかし、データストリームが再起動される場合、エンティティは明確にＳＳＡシステムに対してハッシュ値をリセットするように指示する必要がある。

３．読み出しオペレーションの完了時に、ホストは、ＳＳＡシステムに対して、読み出しハッシュを書き込みオペレーション中に算出されたハッシュ値と比較することにより読み出しハッシュを確認するように、明確に要求しなければならない。

４．さらに、ＳＳＡシステムは「ダミー読み出し（ｄｕｍｍｙ　ｒｅａｄ）」オペレーションを提供する。この機能は、暗号化エンジンを通じてデータをストリームするが、ホストに対してデータを送信しない。この機能は、実際にデータがデバイス（例えば、フラッシュカード）から読み出される前に、データ完全性を検証するために使用可能である。

【０１３３】

《乱数の生成》

ＳＳＡシステムは、外部のエンティティが内部の乱数ジェネレータ（ｒｕｎｄｏｍ　ｎｕｍｂｅｒ　ｇｅｎｅｒａｔｏｒ）を利用し、乱数がＳＳＡシステムの外部で使用されるように要求することを可能にする。このサービスは任意のホストに利用可能であり、認証を必要としない。

【０１３４】

《ＲＳＡ鍵ペアの生成》

ＳＳＡシステムは、外部のユーザが、内部のＲＳＡ鍵ペア生成機能を利用し、ＲＳＡ鍵ペアがＳＳＡシステムの外部で使用されるように要求することを可能にする。このサービスは任意のホストに利用可能であり、認証を必要としない。

【０１３５】

《代替実施形態》

図１８で説明するように、階層アプローチを使用する代わりに、データベースアプローチを使用して同様の結果を達成することが可能である。

【０１３６】

図１８に示すように、エンティティに対する証明書のリスト、認証方法、失敗した試行の最大回数、およびブロック解除に必要な証明書の最小数が、コントローラ１２またはメモリ２０に保存されたデータベースに入力されてもよく、これは、そうした証明書の要求をメモリ１０のコントローラ１２により実行されたデータベースにおけるポリシー（鍵およびパーティションに対する読み出し、書き込みアクセス、およびセキュアチャネル要求）に関連付けるものである。また、鍵およびパーティションへのアクセスに関する制約（ｃｏｎｓｔｒａｉｎｔｓ）および制限（ｌｉｍｉｔａｔｉｏｎ）がデータベースに保存されている。このように、エンティティ（例えば、システム管理者）の中には、ホワイトリストに掲載されものがいるかもしれないが、これは、これらのエンティティが常にすべての鍵およびパーティションにアクセス可能であることを意味する。他のエンティティはブラックリストに掲載されるかもしれないが、他のエンティティが任意の情報にアクセスを試みる場合にはブロックされる。制限はグローバル（ｇｌｏｂａｌ）であってもよいし、鍵および／またはパーティション固有（ｋｅｙ　ａｎｄ／ｏｒ　ｐａｒｔｉｔｉｏｎ　ｓｐｅｃｉｆｉｃ）であってもよい。これは、所定のエンティティのみが常に特定の鍵およびパーティションにアクセス可能であり、所定のエンティティは常にこれが不可能であることを意味する。コンテンツが存在するパーティションまたはコンテンツを暗号化または復号するために使用される鍵に関係なく、制約もコンテンツ自体に課すことができる。このように、所定のデータ（例えば、歌曲）は、どのエンティティがアクセスしたかに関係なく、所定のデータにアクセスする最初の５つのホストデバイスによってのみ、当該所定のデータにアクセス可能であるという属性、または、他のデータ（例えば、映画）が限ら

れた回数だけ読み出し可能であるという属性を有してもよい。
《認証》
パスワード保護
・　パスワード保護は、保護領域にアクセスするためにパスワードを提示する必要があることを意味する。パスワード保護が２つ以上のパスワードになることが不可能でなければ、パスワードは、例えば読み出しアクセス、または読み出し／書き込みアクセス等の異なる権利と関係付けることができる。
・　パスワード保護とは、デバイス（例えば、フラッシュカード）がホストにより提供されたパスワードを検証することができることを意味する。即ち、デバイスもデバイスが管理するセキュアメモリ領域に保存されたパスワードを有する。
問題および限界
・　パスワードは、リプライ攻撃（ｒｅｐｌｙ　ａｔｔａｃｋ）を受けやすい。パスワードは、毎回提示された後に変化しないので、全く同じようにパスワードを再送することができる。これは、保護されるデータに価値があり、コミュニケーションバスが容易にアクセス可能である場合、パスワードをそのまま使用してはならないということを意味している。
・　パスワードは保存されたデータへのアクセスを保護することはできるが、（鍵ではなく）データの保護に使用すべきではない。
・　ハッカーがシステム全体を破壊しないように、パスワードに関連付けられたセキュリティレベルを高めるためには、マスタ鍵を使用してパスワードを多様化することができる。パスワード送信のために、セキュアコミュニケーションチャネルに基づくセッション鍵を使用可能である。
【０１３７】
図１９は、パスワードを使用する認証を説明するフローチャートである。エンティティは、アカウントＩＤおよびパスワードをシステム１０（例えば、フラッシュメモリカード）に送信する。システムは、パスワードがメモリ内のパスワードと一致するか否かを確認するためにチェックする。一致する場合、認証ステータスが返される。そうでなければ、そのアカウントに対してエラーカウンタがインクリメントされ、エンティティはアカウントＩＤおよびパスワードを再入力するよう求められる。カウンタがオーバーフロー（ｏｖｅｒｆｌｏｗ）した場合、システムはアクセス拒否のステータスを返す。
【０１３８】
《チャレンジレスポンス（ｃｈａｌｌｅｎｇｅ　ｒｅｓｐｏｎｓｅ）》
図２０は、チャレンジ／レスポンス型の方法を使用する認証を説明するフローチャートである。エンティティは、アカウントＩＤを送信し、システム１０からのチャレンジ（ｃｈａｌｌｅｎｇｅ）を要求する。システム１０は乱数を生成し、ホストに提示する。ホストは、当該乱数からレスポンス（ｒｅｓｐｏｎｓｅ）を算出し、これをシステム１０に送信する。システム１０は当該レスポンスと保存された値を比較する。残りのステップは、図１９のアクセスを認めるか否かを決定するステップと同様である。
【０１３９】
図２１は、他のチャレンジ／レスポンス型の方法を使用する認証を説明するフローチャートである。図２１に示す認証は以下の点で図２０の認証と異なる。即ち、図２１の認証は、ホストがシステム１０により認証されることを要求するだけでなく、システム１０がチャレンジ／レスポンスにより認証されることも要求するのであり、当該チャレンジ／レスポンスでは、システム１０はホストからのチャレンジを要求し、ホストによりチェックされるレスポンスを返す。
【０１４０】
図２２は、他のチャレンジ／レスポンス型の方法を使用する認証を説明するフローチャートである。この場合、システム１０の認証のみが必要であり、当該認証ではホストはシステム１０にチャレンジを送信する。システム１０は、システム１０に関する記録との照合のためにホストによりチェックされるレスポンスを算出する。

【０１４１】
《対称鍵》
対称鍵アルゴリズムとは、ホスト側およびデバイス側で「同一の」鍵(ＳＡＭＥ　Ｋｅｙ)が暗号化および復号に使用されることを意味する。即ち、これは鍵がコミュニケーション前に事前に合意されていることを意味する。また、ホスト側およびデバイス側のそれぞれが、互いのリバースアルゴリズム（ｒｅｖｅｒｓｅ　ａｌｇｏｒｉｔｈｍ）を実行すべきである。即ち、一方で暗号化アルゴリズムを実行し、他方で復号アルゴリズムを実行するということである。ホスト側およびデバイス側の双方がコミュニケーションのために両方のアルゴリズムを実行する必要はない。
《認証》
対称鍵認証とは、デバイス（例えば、フラッシュカード）およびホストが同一の鍵を共有し、同一の暗号アルゴリズム（ダイレクトアルゴリズムおよびリバースアルゴリズム、例えば、ＤＥＳおよびＤＥＳ－１）を有することを意味する。
対称鍵認証とは、（リプライ攻撃から保護する）チャレンジレスポンスを意味する。保護されたデバイスは、他のデバイスに対してチャレンジを生成し、両者がレスポンスを算出する。認証デバイスはレスポンスを返信し、保護されたデバイスは、当該レスポンスをチェックし、これにより認証を確認する。そして、認証に関連付けられた権利を認めることができる。
可能な認証の種類は下記の通りである：
外部認証：デバイス（例えば、フラッシュカード）が外部を認証する。即ち、デバイスが所定のホストまたはアプリケーションの証明書を確認する。
相互認証：チャレンジがデバイス側およびホスト側で生成される。
内部認証：ホストアプリケーションがデバイス（例えば、フラッシュカード）を認証する。即ち、ホストが、そのアプリケーションに対してデバイスが本物であるか否かをチェックする。
システム全体のセキュリティレベルを高めるためには（即ち、破壊者がすべてを破壊しない）、
・通常、対称鍵がマスタ鍵を使用する多様化（ｄｉｖｅｒｓｉｆｉｃａｔｉｏｎ）と併用される。
・相互認証が、チャレンジが本物のチャレンジであることを確実にするために、デバイス側およびホスト側からのチャレンジを使用する。
暗号化：対称鍵暗号化も暗号化に使用されているが、これは、対称鍵暗号化が非常に効果的なアルゴリズムだからである。すなわち、対称鍵暗号化は暗号化処理に強力なＣＰＵを必要としない。
【０１４２】
コミュニケーションチャネルの保護に使用される場合：
両デバイスは、チャネルの保護に使用されるセッション鍵を理解しなければならない（即ち、すべての発信データを暗号化し、すべての着信データを復号する）。通常、当該セッション鍵は、事前に共有された秘密対称鍵またはＰＫＩを用いて確立される。
両デバイスは、同一の暗号アルゴリズムを理解し、実行しなければならない。
【０１４３】
《署名》
また、対称鍵をデータの署名に使用することが可能であり、この場合、署名は暗号化の部分的な結果である。結果を部分的に維持することにより、鍵値を暴露することなく、必要な回数だけ署名することを許可する。
【０１４４】
《問題および限界》
対称アルゴリズムは非常に効果的で安全であるが、事前に共有された秘密に基づいている。問題は、この秘密を動的な方法で安全に共有すること、および、恐らく秘密を（例えば、セッション鍵のように）ランダムにすることである。つまり、共有された秘密を長期

間安全に守ることは困難であり、複数の人々と共有することは殆ど不可能だということである。

【０１４５】

このオペレーションを促進するために、公開鍵アルゴリズムが発明されたが、これは、当該アルゴリズムにより、秘密を共有することなく秘密の交換が可能となるからである。

【０１４６】

《公開鍵暗号化》

非対称鍵アルゴリズムは、一般に公開鍵暗号と称す。これは非常に複雑であり、通常、ＣＰＵによる集中的な数学的処理により実行される。当該アルゴリズムは、対称鍵アルゴリズムに関連する鍵配送（ｋｅｙ　ｄｉｓｔｒｉｂｕｔｉｏｎ）の問題を解決するために発明された。また、これはデータの完全性を確実にするために使用される署名機能を提供する。

【０１４７】

非対称鍵アルゴリズムは、それぞれがプライベート鍵および公開鍵と称される、プライベート要素および公開要素を有する鍵を使用する。プライベート鍵と公開鍵の両方は、数学的に関連付けられている。公開鍵は共有可能であり、一方、プライベート鍵は秘匿性を維持しなければならない。これらの鍵に関しては、ラップ（ｗｒａｐ）およびラップ解除（ｕｎｗｒａｐ）、または署名および証明を提供するために、（一つがプライベート鍵用であり、一つが公開鍵用の）２つの数学的関数を使用する。

【０１４８】

《鍵交換および鍵配送》

鍵交換は、ＰＫアルゴリズムを使用すれば非常に簡単になる。デバイスは公開鍵を他のデバイスに送信する。他のデバイスは、この公開鍵を用いて自身の秘密鍵をラップし、前者のデバイスに暗号化されたデータを返す。前者のデバイスは、データのラップ解除に自身のプライベート鍵を使用し、両者に周知であって且つデータ交換に使用可能な秘密鍵を取り出す。対称鍵はこのように容易に交換可能であることから、通常、乱数鍵が使用される。

【０１４９】

《署名》

通常、公開鍵アルゴリズムは、その性質上、小量のデータの署名のみに使用される。データの完全性を確実にするために、その後、公開鍵アルゴリズムはメッセージの一方向フットプリント（ｏｎｅ－ｗａｙ　ｆｏｏｔ　ｐｒｉｎｔ）を提供するハッシュ関数と併用される。

【０１５０】

プライベート鍵はデータの署名に使用される。（自由に利用可能な）公開鍵は署名の検証を許可する。

【０１５１】

《認証》

通常、認証は署名を必要とする。即ち、チャレンジが署名され、検証のために返される。

【０１５２】

鍵の公開部分が検証に使用される。誰でも鍵ペアを生成できるので、公開鍵の所有者が正しい鍵を使用している正しい人物であることを証明するために、当該所有者を認証する必要がある。認証局（ｃｅｒｔｉｆｉｃａｔｉｏｎ　ａｕｔｈｏｒｉｔｙ）は、証明書を供給し、署名された証明書に公開鍵を組み込む。当該証明書は認証局自身により署名される。そして、署名の検証に公開鍵を使用するということは、その鍵を組み込む証明書を発行した認証局が信用できるものであり、証明書がハッキングされてない、即ち、認証局によりハッシュ署名された証明書が正しいことを検証可能であることを意味する。また、これはユーザが認証局の公開鍵証明書を有しており、これを信用していることを意味する。

【０１５３】

ＰＫ認証の提供に関する最も一般的な方法は、認証局またはルート証明書（ｒｏｏｔ　ｃｅｒｔｉｆｉｃａｔｅ）を信用し、且つ、所定の認証局に認証されたすべての鍵ペアを間接的に信用することである。そして認証は、チャレンジに署名し、チャレンジレスポンスおよび証明書を提供することにより、各々が有するプライベート鍵が証明書と一致することを証明する問題となる。そして、証明書は、ハッキングされておらず、信用できる認証局により署名されていることを確かめるためにチェックされる。そして、チャレンジレスポンスが検証される。証明書が信用できるものであり、チャレンジレスポンスが正しい場合、認証は成功する。

【０１５４】

デバイス（例えば、フラッシュカード）における認証は、デバイスが信用できるルート証明書を使用してロードされており、デバイスがハッシュ署名された証明書だけでなくチャレンジレスポンスを検証可能であることを意味する。

【０１５５】

《ファイルの暗号化》

ＰＫアルゴリズムは過度にＣＰＵに依存しているので、大量のデータの暗号化には使用されない。しかし、通常、ＰＫアルゴリズムはコンテンツの暗号化のために生成されたランダム暗号化／復号鍵を保護するために使用される。例えば、ＳＭＩＭＥ（セキュアＥメール；ｓｅｃｕｒｅ　ｅｍａｉｌ）は、すべての受信者の公開鍵を用いて暗号化される鍵を生成する。

【０１５６】

《問題および限界》

何でも鍵ペアを生成することが可能であるので、鍵ペアの出所を確実にするために認証を受けなければならない。鍵の交換中に、秘密鍵が正しいデバイスに提供されているかを確かめたいと思うかもしれない。即ち、提供された公開鍵の出所はチェックを受ける必要がある。そして、証明書管理が鍵の妥当性に関する情報と、鍵が無効な状態であるか否かに関する情報を提供することができることから、証明書管理がセキュリティの一部となる。

【０１５７】

本発明は、様々な実施形態を参照することにより上述の通り記載されているが、当然のことながら、変更および修正は本発明の範囲から逸脱することなく可能であり、本発明は添付の請求項およびそれらの等価物によってのみ定義されるべきである。ここで参照されたすべての参考文献は、参照することにより本発明に組み込まれる。

【図面の簡単な説明】

【０１５８】

【図１】本発明の説明に好適なホストデバイスと通信するメモリシステムのブロック図である。

【図２】本発明の一実施形態を説明する概略図であって、メモリの異なるパーティションと異なるパーティションに保存された暗号化されていないファイルおよび暗号化ファイルに関するもので、ここでは、あるパーティションおよび暗号化ファイルへのアクセスがアクセスポリシーおよび認証手順により制御されるものである。

【図３】メモリ内の異なるパーティションを例示するメモリの概略図である。

【図４】本発明の一実施形態を説明するための、図３に図示されたメモリ内の異なるパーティションに関するファイルロケーションテーブルの概略図であって、ここではパーティション内のいくつかのファイルが暗号化されているものである。

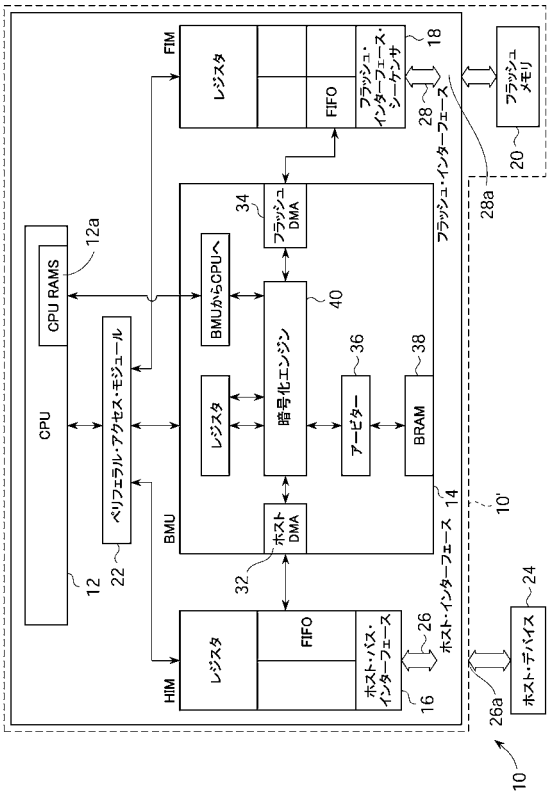
【図５】本発明の一実施形態を説明するための、アクセス制御された記録グループおよび関連するキーリファレンスにおけるアクセス制御記録に関する概略図である。

【図６】本発明の一実施形態を説明するためアクセス制御された記録グループおよびアクセス制御された記録によって形成されたツリー構造の概略図である。

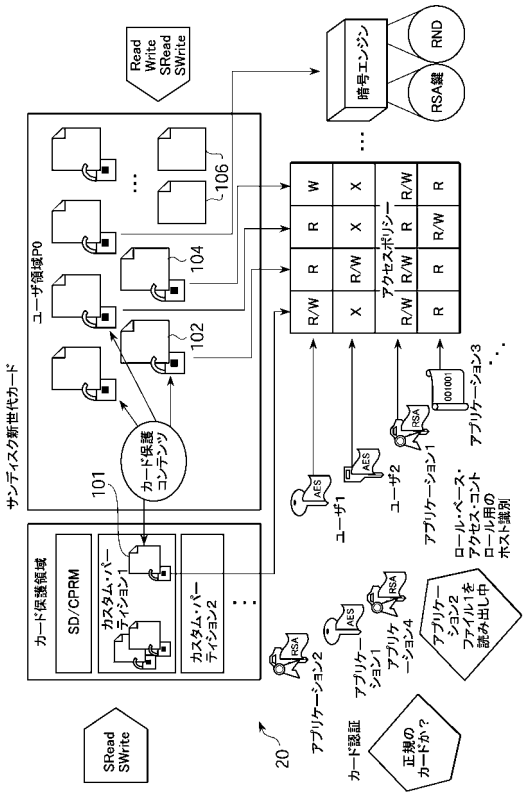
【図７】ツリーの形成プロセスを説明するためアクセス制御された記録グループの３階層ツリーを例示したツリーの概略図である。

【図８Ａ】ホストデバイスおよびメモリデバイス（例えば、システムアクセス制御記録を生成および使用するためのメモリカード等）が実行するプロセスを例示したフローチャートである。
【図８Ｂ】ホストデバイスおよびメモリデバイス（例えば、システムアクセス制御記録を生成および使用するためのメモリカード等）が実行するプロセスを例示したフローチャートである。
【図９】本発明を説明するためアクセス制御された記録グループを生成するためにシステムアクセス制御記録を使用するプロセスを例示したフローチャートである。
【図１０】アクセス制御記録の生成プロセスを例示したフローチャートである。
【図１１】階層ツリーの特定アプリケーションの説明に好適な２つのアクセス制御記録グループの概略図である。
【図１２】特定の権利の委譲プロセスを例示したフローチャートである。
【図１３】図１２の委譲プロセスを説明するためアクセス制御された記録グループおよびアクセス制御記録の概略図である。
【図１４】暗号化および／または復号目的の鍵生成プロセスを例示したフローチャートである。
【図１５】アクセスされ制御された記録に応じて、アクセス権限および／またはデータアクセスパーミッションを取り除くプロセスを例示したフローチャートである。
【図１６】アクセス権限および／またはアクセスパーミッションが削除された、または期限切れになった場合のアクセス要求プロセスを例示したフローチャートである。
【図１７Ａ】本発明の他の実施形態を説明するための、暗号鍵へのアクセスを許可するための認証およびポリシーに対するルール構造の構成を例示した概略図である。
【図１７Ｂ】本発明の他の実施形態を説明するための、暗号鍵へのアクセスを許可するための認証およびポリシーに対するルール構造の構成を例示した概略図である。
【図１８】幾つかのセッションがオープン状態での認証セッションおよびアクセスセッションを例示したフローチャートである。
【図１９】図１９は、他の認証プロセスを例示したフローチャートである。
【図２０】図２０は、他の認証プロセスを例示したフローチャートである。
【図２１】図２１は、他の認証プロセスを例示したフローチャートである。
【図２２】図２２は、他の認証プロセスを例示したフローチャートである。　本出願においては、説明を簡略化するために、同一部分については同一符号が付与されている。

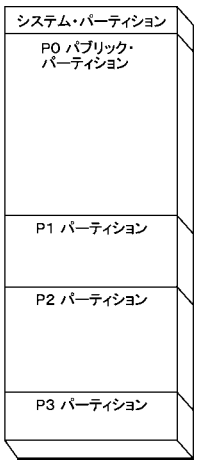
【図１】


FIM
レジスタ
FIFO
フラッシュ・インターフェース・シーケンサ
18
28
フラッシュ・インターフェース
フラッシュメモリ
20
28a
34
フラッシュDMA
BMUからCPUへ
40
暗号化エンジン
レジスタ
36
アービター
38
BRAM
ホストDMA
BMU
14
10'
32
CPU RAMS
12a
CPU
ペリフェラル・アクセス・モジュール
12
22
HIM
レジスタ
FIFO
ホスト・バス・インターフェース
26
ホスト・インターフェース
16
26a
ホスト・デバイス
24
10

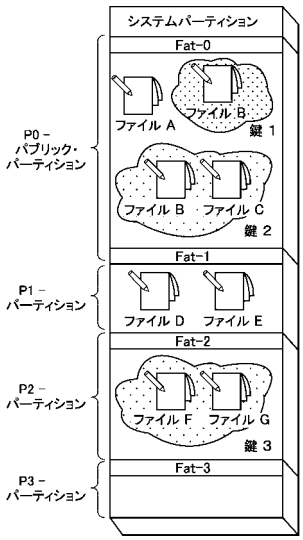
【図２】


Read
Write
SRead
SWrite
暗号エンジン
RND
RSA鍵
ユーザ領域P0
106
104
102
サンディスク新世代カード
101
カード保護コンテンツ
アクセスポリシー
R/W
R
X
カード保護領域
SD/CPRM
カスタム・パーティション1
カスタム・パーティション2
SRead
SWrite
20
ユーザ1
ユーザ2
アプリケーション1
アプリケーション3
ロール・ベース・アクセス・コントロール用のホスト識別
アプリケーション2
アプリケーション4
アプリケーション1
カード認証
正規のカードか？
アプリケーション2ファイル1を読み出し中

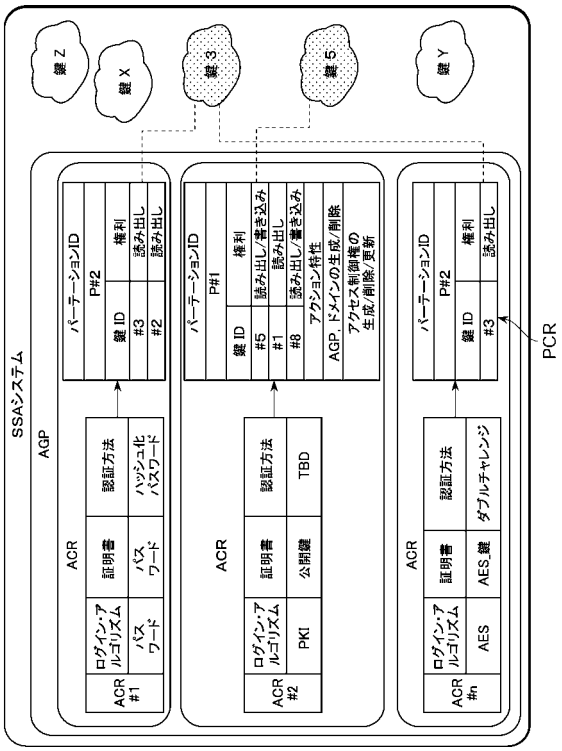
【図３】


システム・パーティション
P0 パブリック・パーティション
P1 パーティション
P2 パーティション
P3 パーティション

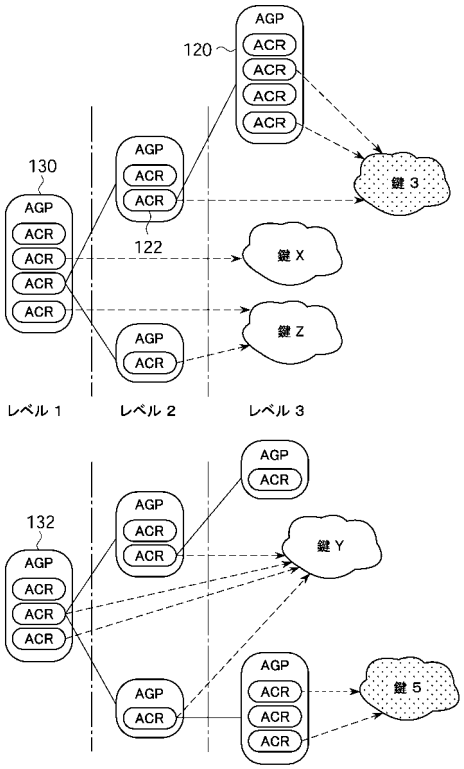
【図４】


システムパーティション
Fat-0
ファイル A
ファイル B
鍵 1
ファイル B
ファイル C
鍵 2
Fat-1
ファイル D
ファイル E
Fat-2
ファイル F
ファイル G
鍵 3
Fat-3
P0 – パブリック・パーティション
P1 – パーティション
P2 – パーティション
P3 – パーティション

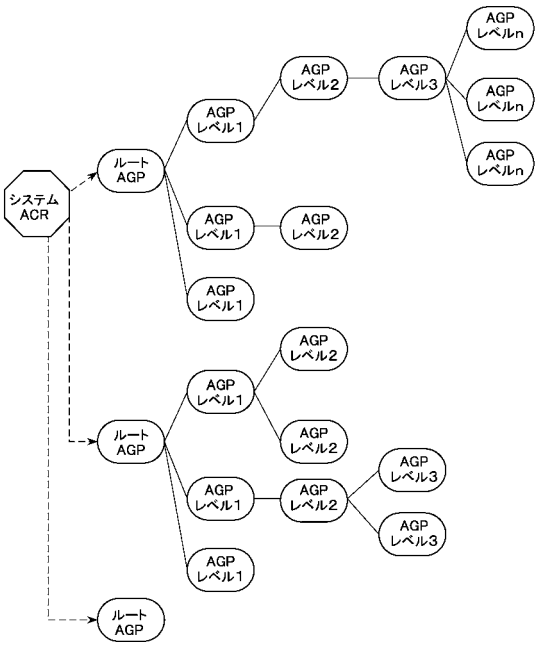
【図５】


鍵 Z
鍵 X
鍵 3
鍵 5
鍵 Y
SSAシステム
AGP
ACR
ACR #1
ACR #2
ACR #n
ログイン・アルゴリズム
パスワード
証明書
認証方法
ハッシュ化パスワード
TBD
公開鍵
PKI
AES
AES_鍵
ダブルチャレンジ
バーテーションID
P#1
P#2
鍵 ID
権利
読み出し
読み出し/書き込み
アクション特性
AGP, ドメインの生成/削除
アクセス制御表の生成/削除/更新
PCR

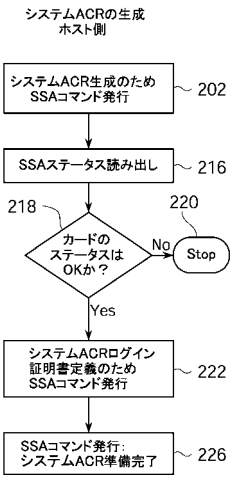
【図６】


AGP
ACR
120
130
122
132
鍵 3
鍵 X
鍵 Z
鍵 Y
鍵 5
レベル 1
レベル 2
レベル 3

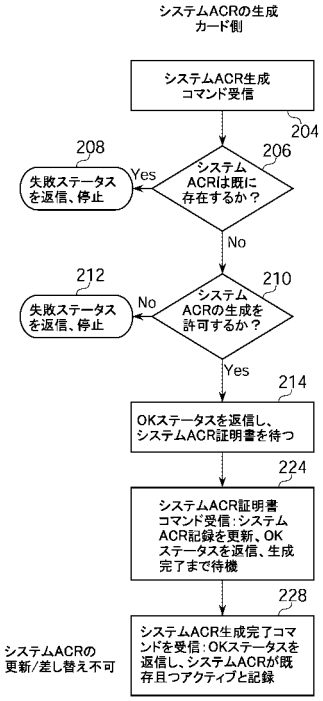
【図７】


システム ACR
ルート AGP
AGP レベル1
AGP レベル2
AGP レベル3
AGP レベルn

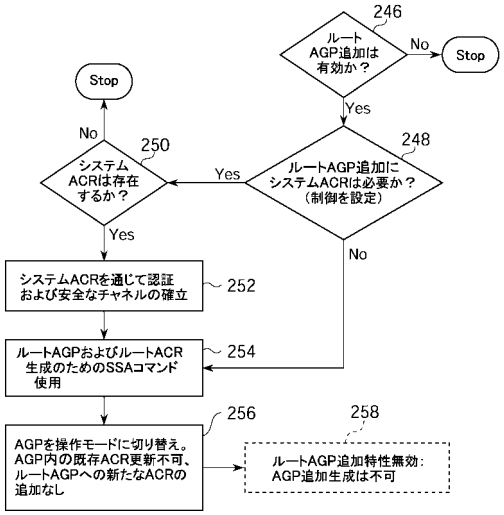
【図８Ａ】


システムACRの生成
ホスト側
システムACR生成のため SSAコマンド発行
202
SSAステータス読み出し
216
218
カードのステータスはOKか？
No
Stop
220
Yes
システムACRログイン 証明書定義のため SSAコマンド発行
222
SSAコマンド発行: システムACR準備完了
226

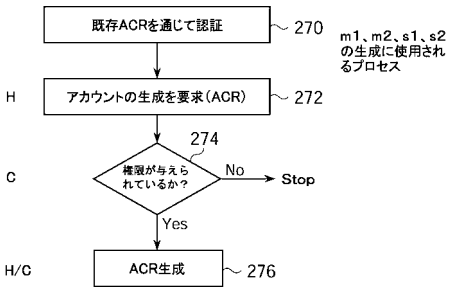
【図８Ｂ】



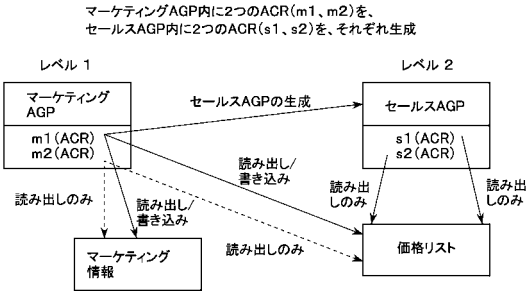
【図９】



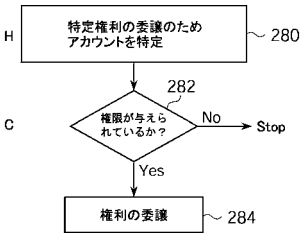
【図１０】



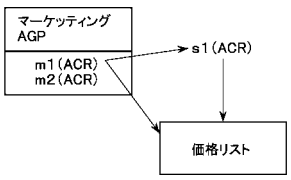
【図１１】



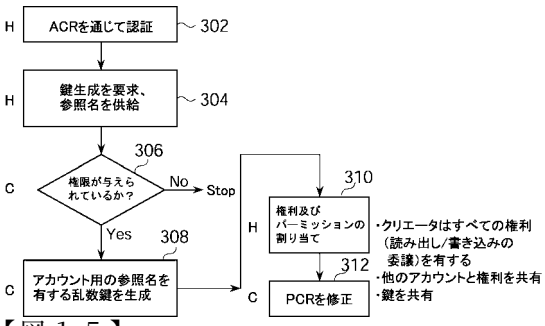
【図１２】



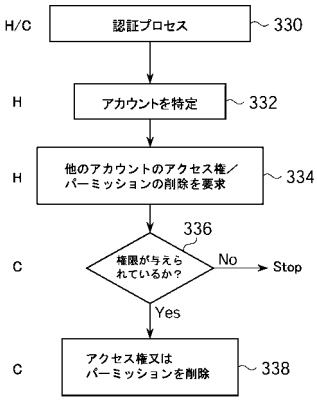
【図１３】



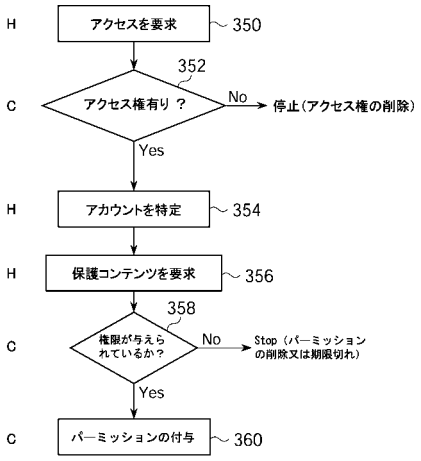
【図１４】



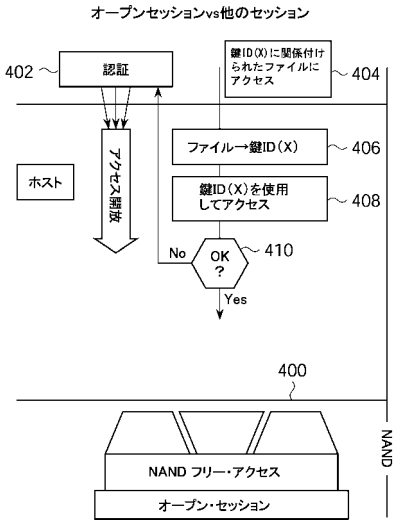
【図１５】



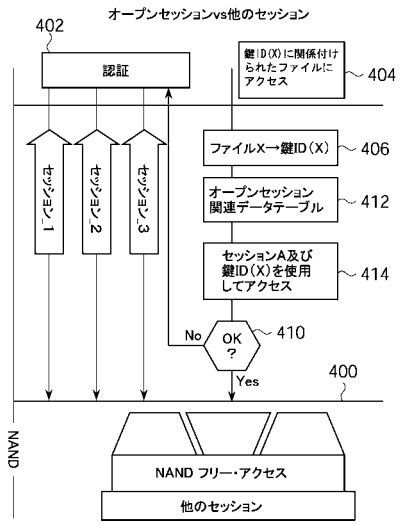
【図１６】



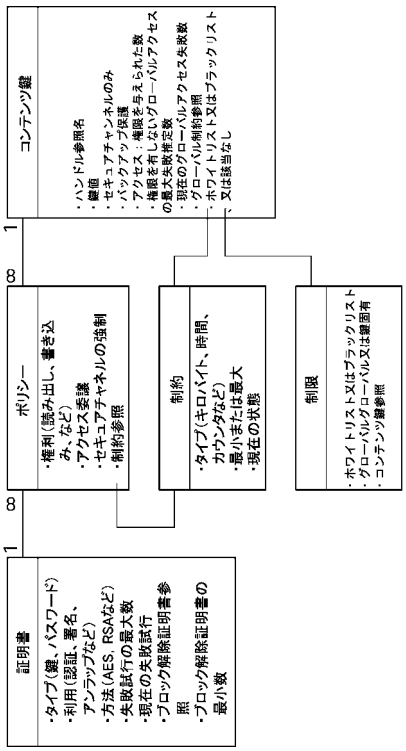
【図１７Ａ】



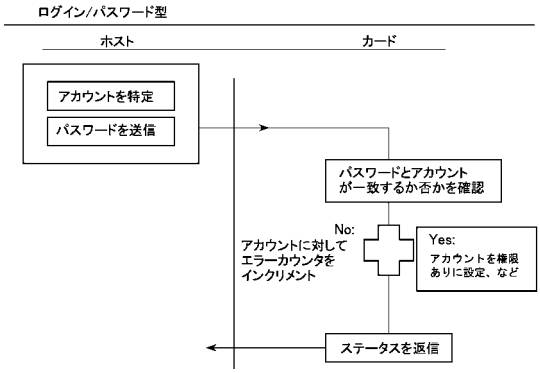
【図１７Ｂ】



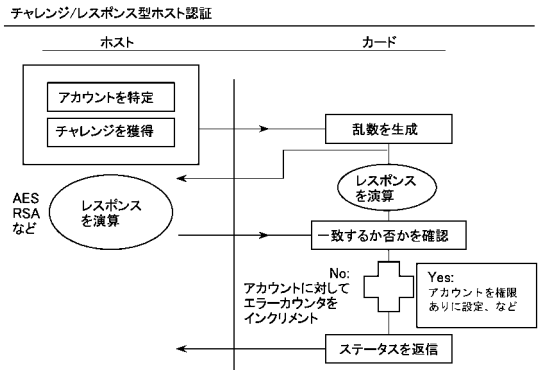
【図１８】



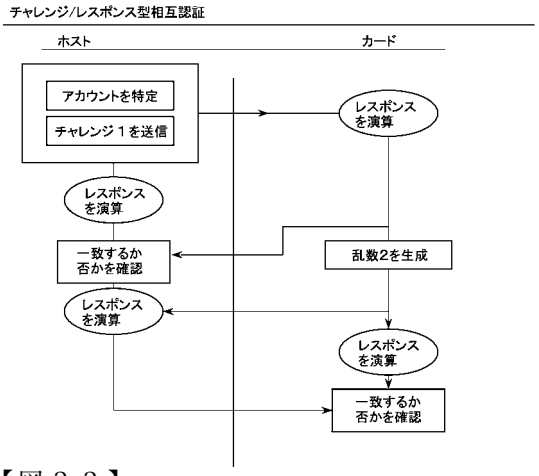
【図１９】



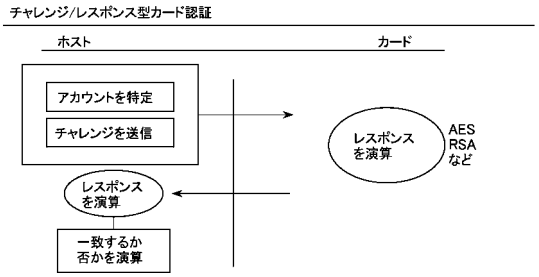
【図２０】



【図２１】



【図２２】



フロントページの続き

(31)優先権主張番号 11/314,053
(32)優先日 平成17年12月20日(2005.12.20)
(33)優先権主張国 米国(US)

(72)発明者 カワミ，バーマン
アメリカ合衆国 ９５１３８ カリフォルニア州，サンノゼ，キラーニー サークル ５８９９
(72)発明者 バーズィライ，ロン
イスラエル国 ２５１４７ クファー－ヴラディム，メロン ストリート ６７

審査官 岸野 徹

(56)参考文献 特開２００４－２９５３５２（ＪＰ，Ａ）
特開２００１－１６６９９６（ＪＰ，Ａ）
特開２００３－２９６２７７（ＪＰ，Ａ）
特開平１１－１６１５５２（ＪＰ，Ａ）
特開２００４－３１０５５７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F 21/24
H04L 9/32