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

特許第4841563号
(P4841563)

(45)発行日　平成23年12月21日(2011.12.21)　(24)登録日　平成23年10月14日(2011.10.14)

(51)Int.Cl.		F I		
H04L 9/10	(2006.01)	H04L	9/00	621A
G06K 19/10	(2006.01)	G06K	19/00	R
G06K 17/00	(2006.01)	G06K	17/00	T

請求項の数 15　(全 50 頁)

(21)出願番号	特願2007-546014 (P2007-546014)
(86)(22)出願日	平成17年11月30日 (2005.11.30)
(65)公表番号	特表2008-524886 (P2008-524886A)
(43)公表日	平成20年7月10日 (2008.7.10)
(86)国際出願番号	PCT/EP2005/056360
(87)国際公開番号	WO2006/063935
(87)国際公開日	平成18年6月22日 (2006.6.22)
審査請求日	平成20年7月31日 (2008.7.31)
(31)優先権主張番号	11/014,559
(32)優先日	平成16年12月16日 (2004.12.16)
(33)優先権主張国	米国(US)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国１０５０４　ニューヨーク州　アーモンク　ニュー　オーチャード　ロード
(74)代理人　100108501
弁理士　上野　剛史
(74)代理人　100112690
弁理士　太佐　種一
(74)代理人　100091568
弁理士　市位　嘉宏

最終頁に続く

(54)【発明の名称】暗号機能を実行するためのデータ処理システム、方法、およびコンピュータ・プログラム

(57)【特許請求の範囲】

【請求項１】

システム・ユニットと、

前記システム・ユニットに結合された媒体読取装置と、

前記媒体読取装置を制御するためのデバイス・ドライバと、

前記媒体読取装置により読み取り可能な取り外し可能記憶媒体であって、第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを保管する取り外し可能記憶媒体と、

前記システム・ユニットに結合されたハードウェア・セキュリティ・ユニットであって、前記第２の非対称暗号鍵ペアに対応する第２の秘密鍵と前記第１の非対称暗号鍵ペアに対応する第２の公開鍵とを保管し、

前記第１および第２の暗号鍵ペアに基づいて前記取り外し可能記憶媒体と前記ハードウェア・セキュリティ・ユニットとを相互に認証するためのロジックと、

前記取り外し可能記憶媒体および前記ハードウェア・セキュリティ・ユニットが相互に認証された後、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記システム・ユニットが前記ハードウェア・セキュリティ・ユニット上の暗号機能を呼び出せるようにするためのロジックと、

を有するハードウェア・セキュリティ・ユニットとを有し、

前記デバイス・ドライバが、

第３の非対称暗号鍵ペアに対応する第３の秘密鍵と、

第４の非対称暗号鍵ペアに対応する第３の公開鍵と、
を有し、
前記ハードウェア・セキュリティ・ユニット上に保管され、前記第４の非対称暗号鍵ペアに対応する第４の秘密鍵と、前記第３の非対称暗号鍵ペアに対応する第４の公開鍵と、
前記第３および第４の非対称暗号鍵ペアに基づいて前記デバイス・ドライバと前記ハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するためのロジックと、
前記取り外し可能記憶媒体および前記デバイス・ドライバが相互に認証された後、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバが前記ハードウェア・セキュリティ・ユニット上の機能を呼び出せるようにするためのロジックと、
をさらに有する、
データ処理システム。

【請求項２】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記取り外し可能記憶媒体に関するデジタル証明書を生成するためのロジックをさらに有する、請求項１に記載のデータ処理システム。

【請求項３】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバからの要求に応答して前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバからのデータ項目にデジタル署名を行うためのロジックをさらに有する、請求項１、または請求項２に記載のデータ処理システム。

【請求項４】
前記取り外し可能記憶媒体および前記デバイス・ドライバが認証され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままであるときに、アプリケーションを認証するためのロジックをさらに有する、請求項１に記載のデータ処理システム。

【請求項５】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバに関するデジタル証明書を生成するためのロジックをさらに有する、請求項１または請求項４に記載のデータ処理システム。

【請求項６】
暗号機能を実行するための方法であって、
システム・ユニットに結合された媒体読取装置に取り外し可能記憶媒体を連結するステップであって、
前記システム・ユニットがハードウェア・セキュリティ・ユニットと前記媒体読取装置を制御するためのデバイス・ドライバとを含み、
前記取り外し可能記憶媒体が第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを含み、
前記ハードウェア・セキュリティ・ユニットが前記第２の非対称暗号鍵ペアに対応する第２の秘密鍵と前記第１の非対称暗号鍵ペアに対応する第２の公開鍵とを含むステップと、
前記第１および第２の非対称暗号鍵ペアに基づいて前記取り外し可能記憶媒体と前記ハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するステップと、
前記相互認証動作を正常に実行したことに応答して、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記システム・ユニットが前記ハードウェア・セキュリティ・ユニット上の暗号機能を呼び出せるようにするステップとを含み、
前記デバイス・ドライバが第３の非対称暗号鍵ペアに対応する第３の秘密鍵と第４の非対称暗号鍵ペアに対応する第３の公開鍵とを含み、

前記データ処理システムが、前記ハードウェア・セキュリティ・ユニット上に保管され、前記第４の非対称暗号鍵ペアに対応する第４の秘密鍵と、前記第３の非対称暗号鍵ペアに対応する第４の公開鍵とを含み、
前記第３および第４の非対称暗号鍵ペアに基づいて前記デバイス・ドライバと前記ハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するステップと、
前記取り外し可能記憶媒体および前記デバイス・ドライバが相互に認証された後、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバが前記ハードウェア・セキュリティ・ユニット上の機能を呼び出せるようにするステップと、
をさらに含む、
方法。

【請求項７】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記取り外し可能記憶媒体に関するデジタル証明書を生成するステップをさらに含む、請求項６に記載の方法。

【請求項８】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバからの要求に応答して前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバからのデータ項目にデジタル署名を行うステップをさらに含む、請求項６、または請求項７に記載の方法。

【請求項９】
前記取り外し可能記憶媒体および前記デバイス・ドライバが認証され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままであるときに、アプリケーションを認証するステップをさらに含む、請求項６に記載の方法。

【請求項１０】
前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバに関するデジタル証明書を生成するステップをさらに含む、請求項６または請求項９に記載の方法。

【請求項１１】
暗号機能を実行するためにデータ処理システム内で使用するためのコンピュータ可読媒体上のコンピュータ・プログラムであって、
前記コンピュータ可読媒体上に保管され、システム・ユニットに結合された媒体読取装置により取り外し可能記憶媒体を読み取るためのロジックであって、
前記システム・ユニットがハードウェア・セキュリティ・ユニットと前記媒体読取装置を制御するためのデバイス・ドライバとを含み、
前記取り外し可能記憶媒体が第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを含み、
前記ハードウェア・セキュリティ・ユニットが前記第２の非対称暗号鍵ペアに対応する第２の秘密鍵と前記第１の非対称暗号鍵ペアに対応する第２の公開鍵とを含むロジックと、
前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されている間に前記媒体読取装置と前記ハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するためのロジックと、
前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体と前記ハードウェア・セキュリティ・ユニットとの間の前記相互認証動作を正常に実行したことに応答して、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニット上の暗号機能を使用可能にするためのロジックとを含み、
前記デバイス・ドライバが第３の非対称暗号鍵ペアに対応する第３の秘密鍵と第４の非対称暗号鍵ペアに対応する第３の公開鍵とを含み、

前記データ処理システムが、前記ハードウェア・セキュリティ・ユニット上に保管され、前記第４の非対称暗号鍵ペアに対応する第４の秘密鍵と、前記第３の非対称暗号鍵ペアに対応する第４の公開鍵とを含み、

前記コンピュータ可読媒体上に保管され、前記第３および第４の非対称暗号鍵ペアに基づいて前記デバイス・ドライバと前記ハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するためのロジックと、

前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体および前記デバイス・ドライバが相互に認証された後、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバが前記ハードウェア・セキュリティ・ユニット上の機能を呼び出せるようにするためのロジックと、

をさらに含む、

コンピュータ・プログラム。

【請求項１２】

前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記取り外し可能記憶媒体に関するデジタル証明書を生成するためのロジックをさらに含む、請求項１１に記載のコンピュータ・プログラム。

【請求項１３】

前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記デバイス・ドライバからの要求に応答して前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバからのデータ項目にデジタル署名を行うためのロジックをさらに含む、請求項１１、または請求項１２に記載のコンピュータ・プログラム。

【請求項１４】

前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体および前記デバイス・ドライバが認証され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままであるときに、アプリケーションを認証するためのロジックをさらに含む、請求項１１に記載のコンピュータ・プログラム。

【請求項１５】

前記コンピュータ可読媒体上に保管され、前記取り外し可能記憶媒体が前記媒体読取装置に連結されたままである間に前記ハードウェア・セキュリティ・ユニットにより前記デバイス・ドライバに関するデジタル証明書を生成するためのロジックをさらに含む、請求項１４に記載のコンピュータ・プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、改良されたデータ処理システムに関し、特に、暗号方式を使用するデータ記憶保護のための方法および装置に関する。

【背景技術】

【０００２】

本出願は、２００４年１月８日に出願され、「Ｍｅｔｈｏｄ ａｎｄ ｓｙｓｔｅｍ ｆｏｒ ｅｓｔａｂｌｉｓｈｉｎｇ ａ ｔｒｕｓｔ ｆｒａｍｅｗｏｒｋ ｂａｓｅｄ ｏｎ ｓｍａｒｔ ｋｅｙ ｄｅｖｉｃｅｓ」という発明の名称の米国特許出願公報第２００５／０１５４８７５号、ならびに２００４年１月８日に出願され、「Ｍｅｔｈｏｄ ａｎｄ Ｓｙｓｔｅｍ ｆｏｒ Ｐｒｏｔｅｃｔｉｎｇ Ｍａｓｔｅｒ Ｓｅｃｒｅｔｓ Ｕｓｉｎｇ Ｓｍａｒｔ Ｋｅｙ Ｄｅｖｉｃｅｓ」という発明の名称の米国特許出願公報第２００５／０１５４８９８号に関連する。

【０００３】

ほとんどのデータ処理システムは、保護する必要のある機密データを収容している。た

とえば、構成情報のデータ保全性は違法な変更から保護する必要があり、パスワード・ファイルなどのその他の情報は違法な開示から保護する必要がある。所与のデータ処理システムのオペレータは、データ処理システムを保護するために多種多様なタイプのセキュリティ・メカニズムを使用することができる。たとえば、データ処理システム上のオペレーティング・システムは、様々な認証および許可方式など、機密データを保護するための様々なソフトウェア・メカニズムを提供することができ、特定のハードウェア・デバイスおよびソフトウェア・アプリケーションは、ハードウェア・セキュリティ・トークンおよびバイオメトリック・センサ・デバイスなど、機密データを保護するためのハードウェア・メカニズムに依存することができる。機密データを保護するために所与のデータ処理システム内で複数のソフトウェアおよびハードウェア・メカニズムを使用できる場合でも、誰かが暗号化機密データに違法なアクセスをする場合に暗号化機密データを暗号化解除する能力がなければ暗号化機密データのコピーが無用なものになるように機密データを暗号化することもできる。

【０００４】

しかし、データ処理システム内に収容されているすべての情報を最終的に保護するための能力には限界がある。たとえば、パスワード・ファイルをさらに保護しようとして、マスタ・シークレット（mastersecret）と呼ばれる場合が多いパスワードまたは暗号鍵（cryptographickey）などのさらに他の秘密を使用してパスワード・ファイルを暗号化する場合もある。しかし、この新たな秘密も何らかの方法で保護する必要がある。したがって、システム管理者は、他のセキュリティ層を実現しようとすると、その結果、やはり保護する必要のある追加の機密情報が発生するという、ある種のジレンマに陥る可能性がある。ここで本発明を考慮すると、残りの図には、このジレンマを解決する本発明の模範的な諸実施形態が描かれている。

【特許文献１】米国特許出願公報第２００５／０１５４８７５号

【特許文献２】米国特許出願公報第２００５／０１５４８９８号

【非特許文献１】１９９９年３月の「Ｉｎｔｅｒｎｅｔ Ｅｎｇｉｎｅｅｒｉｎｇ Ｔａｓｋ Ｆｏｒｃｅ （ＩＥＴＦ） Ｒｅｑｕｅｓｔ ｆｏｒ Ｃｏｍｍｅｎｔｓ （ＲＦＣ） ２５１１」に掲載されたＭｙｅｒｓ他による「Ｉｎｔｅｒｎｅｔ Ｘ．５０９ Ｃｅｒｔｉｆｉｃａｔｅ Ｒｅｑｕｅｓｔ Ｍｅｓｓａｇｅ Ｆｏｒｍａｔ」

【非特許文献２】１９９９年３月の「ＩＥＴＦ ＲＦＣ ２５１１」に掲載されたＡｄａｍｓ他による「Ｉｎｔｅｒｎｅｔ Ｘ．５０９ Ｐｕｂｌｉｃ Ｋｅｙ Ｉｎｆｒａｓｔｒｕｃｔｕｒｅ Ｃｅｒｔｉｆｉｃａｔｅ Ｍａｎａｇｅｍｅｎｔ Ｐｒｏｔｏｃｏｌｓ」

【発明の開示】

【発明が解決しようとする課題】

【０００５】

したがって、暗号鍵など、秘密情報を安全に保管し管理するためのメカニズムを有することは、有利なことになるであろう。他の秘密情報を保護するために使用されるマスタ・シークレットを安全に保管し管理することは、特に有利なことになるであろう。

【課題を解決するための手段】

【０００６】

本発明の第１の態様は、暗号機能を実行するためにデータ処理システム内で使用するためのコンピュータ可読媒体上のコンピュータ・プログラム（computerprogram product）を提供し、このコンピュータ・プログラムは、コンピュータ可読媒体上に保管され、システム・ユニットに結合された媒体読取装置により取り外し可能記憶媒体を読み取るためのロジックであって、システム・ユニットがハードウェア・セキュリティ・ユニットと媒体読取装置を制御するためのデバイス・ドライバとを含み、取り外し可能記憶媒体が第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを含み、ハードウェア・セキュリティ・ユニットが第２の非対称暗号鍵ペアに対応する第２の秘密鍵と第１の非対称暗号鍵ペアに対応する第２の公開鍵とを含むロジックと

、コンピュータ可読媒体上に保管され、取り外し可能記憶媒体が媒体読取装置に連結されている間に媒体読取装置とハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するためのロジックと、コンピュータ可読媒体上に保管され、取り外し可能記憶媒体とハードウェア・セキュリティ・ユニットとの間の相互認証動作を正常に実行したことに応答して、取り外し可能記憶媒体が媒体読取装置に連結されたままである間にハードウェア・セキュリティ・ユニット上の暗号機能を使用可能にするためのロジックとを含む。

【０００７】

本発明の第２の態様は、暗号機能を実行するための方法を提供し、この方法は、システム・ユニットに結合された媒体読取装置に取り外し可能記憶媒体を連結するステップであって、システム・ユニットがハードウェア・セキュリティ・ユニットと媒体読取装置を制御するためのデバイス・ドライバとを含み、取り外し可能記憶媒体が第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを含み、ハードウェア・セキュリティ・ユニットが第２の非対称暗号鍵ペアに対応する第２の秘密鍵と第１の非対称暗号鍵ペアに対応する第２の公開鍵とを含むステップと、第１および第２の非対称暗号鍵ペアに基づいて取り外し可能記憶媒体とハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するステップと、相互認証動作を正常に実行したことに応答して、取り外し可能記憶媒体が媒体読取装置に連結されたままである間にシステム・ユニットがハードウェア・セキュリティ・ユニット上の暗号機能を呼び出せるようにするステップとを含む。

【０００８】

本発明の他の態様は、暗号機能を実行するためにデータ処理システム内で使用するためのコンピュータ可読媒体上のコンピュータ・プログラムを提供し、このコンピュータ・プログラムは、コンピュータ可読媒体上に保管され、システム・ユニットに結合された媒体読取装置により取り外し可能記憶媒体を読み取るためのロジックであって、システム・ユニットがハードウェア・セキュリティ・ユニットと媒体読取装置を制御するためのデバイス・ドライバとを含み、取り外し可能記憶媒体が第１の非対称暗号鍵ペアに対応する第１の秘密鍵と第２の非対称暗号鍵ペアに対応する第１の公開鍵とを含み、ハードウェア・セキュリティ・ユニットが第２の非対称暗号鍵ペアに対応する第２の秘密鍵と第１の非対称暗号鍵ペアに対応する第２の公開鍵とを含むロジックと、コンピュータ可読媒体上に保管され、取り外し可能記憶媒体が媒体読取装置に連結されている間に媒体読取装置とハードウェア・セキュリティ・ユニットとの間の相互認証動作を実行するためのロジックと、コンピュータ可読媒体上に保管され、取り外し可能記憶媒体とハードウェア・セキュリティ・ユニットとの間の相互認証動作を正常に実行したことに応答して、取り外し可能記憶媒体が媒体読取装置に連結されたままである間にハードウェア・セキュリティ・ユニット上の暗号機能を使用可能にするためのロジックとを含む。

【０００９】

データ処理システムは取り外し可能記憶媒体を受け入れ、その取り外し可能記憶媒体はデータ処理システム内のシステム・ユニットに電気的に連結された状態になり、その後、取り外し可能記憶媒体とハードウェア・セキュリティ・ユニットは相互に認証する。取り外し可能記憶媒体は、第１の非対称暗号鍵ペアの秘密鍵と、ハードウェア・セキュリティ・ユニットに関連する第２の非対称暗号鍵ペアの公開鍵とを保管し、ハードウェア・セキュリティ・ユニットは、第２の非対称暗号鍵ペアの秘密鍵と、取り外し可能記憶媒体に関連する第１の非対称暗号鍵ペアの公開鍵とを保管する。取り外し可能記憶媒体とハードウェア・セキュリティ・ユニットとの間の相互認証動作を正常に実行したことに応答して、取り外し可能記憶媒体がシステム・ユニットに連結されたままである間にシステム・ユニットがハードウェア・セキュリティ・ユニット上の機密暗号機能を呼び出せるようになる。

【００１０】

本発明に特有と思われる新規な特徴は特許請求の範囲に示されている。本発明そのもの、ならびに本発明のその他の目的および利点については、添付図面に併せて読んだときに

以下の詳細な説明を参照することによって最も良く理解されるであろう。
【発明を実施するための最良の形態】
【００１１】
一般に、本発明を含むかまたは本発明に関連する可能性のある装置は多様なデータ処理技術を含む。したがって、本発明についてより詳細に説明する前に、背景として、分散データ処理システム内のハードウェアおよびソフトウェア・コンポーネントの典型的な編成について説明する。
【００１２】
次に図面に関連して説明すると、図１は、それぞれが本発明の一部分を実現可能な複数のデータ処理システムからなる典型的なネットワークを描写している。分散データ処理システム１００はネットワーク１０１を含み、このネットワークは、分散データ処理システム１００内でともに接続されている様々な装置およびコンピュータ間の通信リンクを提供するために使用可能な媒体である。ネットワーク１０１は、ワイヤまたは光ファイバ・ケーブルなどの永久的な接続、あるいは電話または無線通信によって行われる一時的な接続を含むことができる。描写されている例では、サーバ１０２およびサーバ１０３は、記憶装置１０４とともにネットワーク１０１に接続されている。加えて、クライアント１０５～１０７もネットワーク１０１に接続されている。クライアント１０５～１０７およびサーバ１０２～１０３は、メインフレーム、パーソナル・コンピュータ、携帯情報端末（ＰＤＡ：personal digital assistant）などの様々なコンピューティング・デバイスによって表すことができる。分散データ処理システム１００は、図示されていない追加のサーバ、クライアント、ルータ、その他の装置、およびピアツーピア・アーキテクチャを含むこともできる。
【００１３】
描写されている例では、分散データ処理システム１００は、軽量ディレクトリ・アクセス・プロトコル（ＬＤＡＰ：Lightweight Directory Access Protocol）、伝送制御プロトコル／インターネット・プロトコル（ＴＣＰ／ＩＰ：Transport Control Protocol/InternetProtocol）、ハイパテキスト転送プロトコル（ＨＴＴＰ：Hypertext Transport Protocol）、無線アプリケーション・プロトコル（ＷＡＰ：Wireless Application Protocol）など、相互に通信するために様々なプロトコルを使用するネットワークおよびゲートウェイの世界的な集合を表すネットワーク１０１とともにインターネットを含むことができる。当然のことながら、分散データ処理システム１００は、たとえば、イントラネット、ローカル・エリア・ネットワーク（ＬＡＮ：local area network）、または広域ネットワーク（ＷＡＮ：wide area network）など、いくつかの異なるタイプのネットワークも含むことができる。たとえば、サーバ１０２はクライアント１０９とネットワーク１１０を直接サポートし、そのネットワークは無線通信リンクを取り込んでいる。ネットワーク対応電話１１１は無線リンク１１２によりネットワーク１１０に接続し、ＰＤＡ１１３は無線リンク１１４によりネットワーク１１０に接続する。電話１１１およびＰＤＡ１１３は、いわゆるパーソナル・エリア・ネットワーク（ＰＡＮ：personal area network）またはパーソナル・アドホック・ネットワークを作成するためにＢｌｕｅｔｏｏｔｈ（商標）無線技術などの適切な技術を使用して、無線リンク１１５の両端の両者間でデータを直接転送することもできる。同様に、ＰＤＡ１１３は、無線通信リンク１１６を介してＰＤＡ１０７にデータを転送することができる。
【００１４】
本発明は様々なハードウェア・プラットフォーム上で実現することができ、図１は、本発明に関するアーキテクチャ上の制限としてではなく、異機種コンピューティング環境の一例として意図されている。
【００１５】
次に図２に関連して説明すると、同図は、本発明を実現可能な、図１に図示されているものなどのデータ処理システムの典型的なコンピュータ・アーキテクチャを描写している。データ処理システム１２０は、内部システム・バス１２３に接続された１つまたは複数

の中央演算処理装置（ＣＰＵ：central processing unit）１２２を含み、その内部システム・バスは、ランダム・アクセス・メモリ（ＲＡＭ：random access memory）１２４、読み取り専用メモリ１２６、および入出力アダプタ１２８を相互接続し、その入出力アダプタは、プリンタ１３０、ディスク装置１３２、またはオーディオ出力システムなどの図示されていないその他の装置などの様々な入出力装置をサポートする。システム・バス１２３は、通信リンク１３６へのアクセスを可能にする通信アダプタ１３４も接続する。ユーザ・インターフェース・アダプタ１４８は、キーボード１４０およびマウス１４２、またはタッチ・スクリーン、スタイラス、マイクロホンなどの図示されていないその他の装置などの様々なユーザ装置を接続する。ディスプレイ・アダプタ１４４は、システム・バス１２３をディスプレイ装置１４６に接続する。

【００１６】

当業者であれば、図２のハードウェアがシステム実現例次第で様々になる可能性があることが分かるであろう。たとえば、システムは、Ｉｎｔｅｌ（商標）のＰｅｎｔｉｕｍ（登録商標）ベースのプロセッサおよびデジタル信号プロセッサ（ＤＳＰ：digital signal processor）などの１つまたは複数のプロセッサと、１つまたは複数のタイプの揮発性および不揮発性メモリを有することができる。図２に描写されているハードウェアに加えてまたはその代わりに、その他の周辺装置を使用することもできる。描写されている例は、本発明に関するアーキテクチャ上の制限を暗示するためのものではない。

【００１７】

様々なハードウェア・プラットフォーム上に実現できることに加えて、本発明は、様々なソフトウェア環境で実現することができる。典型的なオペレーティング・システムを使用して、それぞれのデータ処理システム内のプログラム実行を制御することができる。たとえば、ある装置はＵｎｉｘ（登録商標）オペレーティング・システムを実行することができ、他の装置は単純なＪａｖａ（登録商標）ランタイム環境を含む。代表的なコンピュータ・プラットフォームはブラウザを含むことができ、そのブラウザは、グラフィック・ファイル、ワード・プロセッシング・ファイル、拡張可能マークアップ言語（ＸＭＬ：Extensible Markup Language）、ハイパテキスト・マークアップ言語（ＨＴＭＬ：Hypertext Markup Language）、ハンドヘルド・デバイス・マークアップ言語（ＨＤＭＬ：Handheld Device Markup Language）、無線用マークアップ言語（ＷＭＬ：Wireless Markup Language）、およびその他の様々なフォーマットおよびタイプのファイルなど、様々なフォーマットのハイパテキスト文書にアクセスするための周知のソフトウェア・アプリケーションである。

【００１８】

図１および図２に関して上述した通り、本発明は様々なハードウェアおよびソフトウェア・プラットフォーム上で実現することができる。しかし、より具体的には、本発明は、ハードウェア・セキュリティ・トークンの使用により秘密情報を保護するためのメカニズムを対象とする。しかし、本発明についてより詳細に説明する前に、本発明の動作効率およびその他の利点を評価するために、デジタル証明書に関する何らかの背景情報を提供する。

【００１９】

デジタル証明書は、通信またはトランザクションに係わる各当事者が公開鍵と秘密鍵という１対の鍵を有する、公開鍵暗号方式をサポートするものである。各当事者の公開鍵は公開され、秘密鍵は秘密に保持される。公開鍵は、特定のエンティティに関連する番号であり、そのエンティティとの信頼できる対話を有する必要があるすべての人に知られることが意図されている。秘密鍵は、特定のエンティティのみに知られる、すなわち、秘密に保持されることが想定されている番号である。典型的な非対称暗号システムでは、１つの秘密鍵は厳密に１つの公開鍵に対応する。

【００２０】

公開鍵暗号方式システム内では、すべての通信が公開鍵のみを必要とし、いかなる秘密鍵も伝送または共用されないので、機密メッセージは、公用情報のみを使用して生成する

ことができ、意図された受信者が単独所有している秘密鍵のみを使用して暗号化解除することができる。さらに、公開鍵暗号方式は、認証すなわちデジタル署名のため、ならびにプライバシすなわち暗号化のために使用することができる。

【００２１】

暗号化とは秘密の暗号化解除鍵なしでは誰も読めない形式にデータを変換することであり、暗号化は、暗号化データを見ることができる人であっても意図されていない人である場合、情報の内容をその人から隠した状態に保持することにより、プライバシを保証する。認証とは、それによりデジタル・メッセージの受信側が送信側のアイデンティティまたはメッセージの保全性あるいはその両方を確信できるプロセスである。

【００２２】

たとえば、送信側がメッセージを暗号化する場合、元のメッセージ内のデータを暗号化メッセージの内容に変換するために、受信側の公開鍵が使用される。送信側は意図された受信者の公開鍵を使用してデータを暗号化し、受信側はその秘密鍵を使用して暗号化メッセージを暗号化解除する。

【００２３】

データを認証する場合、署名者の秘密鍵を使用してデータからデジタル署名を計算することにより、データに署名することができる。データにデジタル署名が行われると、そのデータは、署名者のアイデンティティならびにそのデータが署名者から生じたものであることを証明する署名とともに保管することができる。署名者はその秘密鍵を使用してデータに署名し、受信側は署名者の公開鍵を使用して署名を検証する。

【００２４】

証明書は、個人、コンピュータ・システム、そのシステム上で実行される特定のサーバなど、エンティティのアイデンティティおよび鍵所有権を保証するデジタル文書である。証明書は認証局によって発行される。認証局（ＣＡ：certificate authority）は、他の人またはエンティティに関する証明書に署名するかまたはその証明書を発行することを任されているエンティティ、通常はあるトランザクションに対する信頼のおける第三者機関である。認証局は、通常、公開鍵とその所有者との結合に関するその保証によって、証明書に署名したエンティティを信頼できるようにする、ある種の法律上の責任を有する。多くの商用認証局が存在し、このような認証局は、証明書を発行するときに、あるエンティティのアイデンティティおよび鍵所有権の検証を担当する。

【００２５】

認証局があるエンティティに関する証明書を発行する場合、そのエンティティは、公開鍵と、そのエンティティに関する何らかの情報を提供しなければならない。特別に装備されたＷｅｂブラウザなどのソフトウェア・ツールは、この情報にデジタル署名を行い、それを認証局に送信することができる。認証局は、信頼のおける第三者機関認証局サービスを提供する営利企業である場合もある。その場合、認証局は、証明書を生成し、それを返すことになる。証明書は、通し番号や、その間、証明書が有効である複数の日付などの他の情報を含むことができる。認証局によって提供される価値ある役割の１つは、その認証サービス慣行（ＣＳＰ：Certification Service Practice）に公然と公表されているその検証要件にある程度基づいて、中立かつ信頼できる紹介サービスとして働くことである。

【００２６】

認証局は、その他の識別情報とともに要求側エンティティの公開鍵を組み込み、次に認証局の秘密鍵でデジタル証明書に署名することにより、新しいデジタル証明書を作成する。次に、トランザクションまたは通信中にデジタル証明書を受信した人は誰でも、認証局の公開鍵を使用して、証明書内の署名付き公開鍵を検証することができる。その意図は、認証局の署名がデジタル証明書上の改ざん防止シールとして作用し、それにより、証明書内のデータの保全性を保証することである。

【００２７】

証明書処理のその他の態様も標準化されている。１９９９年３月の「Ｉｎｔｅｒｎｅｔ Ｅｎｇｉｎｅｅｒｉｎｇ Ｔａｓｋ Ｆｏｒｃｅ （ＩＥＴＦ） Ｒｅｑｕｅｓｔ ｆ

ｏｒ Ｃｏｍｍｅｎｔｓ （ＲＦＣ） ２５１１」に掲載されたＭｙｅｒｓ他による「Ｉｎｔｅｒｎｅｔ Ｘ．５０９ Ｃｅｒｔｉｆｉｃａｔｅ Ｒｅｑｕｅｓｔ Ｍｅｓｓａｇｅ Ｆｏｒｍａｔ」では、依存している当事者が認証局からの証明書を要求している場合に使用するよう推奨されているフォーマットが指定されている。１９９９年3月の「ＩＥＴＦ ＲＦＣ ２５１１」に掲載されたＡｄａｍｓ他による「Ｉｎｔｅｒｎｅｔ Ｘ．５０９ Ｐｕｂｌｉｃ Ｋｅｙ Ｉｎｆｒａｓｔｒｕｃｔｕｒｅ Ｃｅｒｔｉｆｉｃａｔｅ Ｍａｎａｇｅｍｅｎｔ Ｐｒｏｔｏｃｏｌｓ」では、証明書を転送するためのプロトコルが指定されている。本発明はデジタル証明書を使用する分散データ処理システムにあり、図３～図４の説明は、デジタル証明書を必要とする典型的な動作に関する背景情報を提供するものである。

【００２８】

次に図３に関連して説明すると、このブロック図は、ある個人がデジタル証明書を入手する際の典型的な方法を描写している。何らかのタイプのクライアント・コンピュータ上で操作しているユーザ２０２は、公開／秘密鍵ペア、たとえば、ユーザ公開鍵２０４およびユーザ秘密鍵２０６を前もって入手または生成している。ユーザ２０２は、ユーザ公開鍵２０４を含む証明書２０８に関する要求を生成し、ＣＡ公開鍵２１２およびＣＡ秘密鍵２１４を所有している認証局２１０にその要求を送信する。認証局２１０は、何らかの方法でユーザ２０２のアイデンティティを検証し、ユーザ公開鍵２１８を含むＸ．５０９デジタル証明書２１６を生成する。この証明書全体はＣＡ秘密鍵２１４で署名され、この証明書は、ユーザの公開鍵と、ユーザに関連する名前と、その他の属性とを含む。ユーザ２０２は新たに生成されたデジタル証明書２１６を受信し、その後、ユーザ２０２は、信頼できるトランザクションまたは信頼できる通信に従事するために必要なデジタル証明書２１６を提示することができる。ユーザ２０２からデジタル証明書２１６を受信したエンティティは、公表され、検証エンティティにとって使用可能であるＣＡ公開鍵２１２を使用することにより、認証局の署名を検証することができる。

【００２９】

次に図４に関連して説明すると、このブロック図は、あるエンティティがデジタル証明書を使用してデータ処理システムに対して認証される際の典型的な方法を描写している。ユーザ３０２はＸ．５０９デジタル証明書３０４を所有しており、その証明書はホスト・システム３０８上のインターネットまたはイントラネット・アプリケーション３０６に伝送され、アプリケーション３０６はデジタル証明書を処理し使用するためにＸ．５０９の機能を有する。ユーザ３０２は、その秘密鍵とともにアプリケーション３０６に送信するデータに署名するかまたはそのデータを暗号化する。

【００３０】

証明書３０４を受信するエンティティは、アプリケーション、システム、サブシステムなどである可能性がある。証明書３０４は、アプリケーション３０６に対してユーザ３０２を識別するサブジェクト名またはサブジェクトＩＤを含み、そのアプリケーションはユーザ３０２のために何らかのタイプのサービスを実行することができる。証明書３０４を使用するエンティティは、ユーザ３０２からの署名付きデータまたは暗号化データについて証明書を使用する前に、証明書の認証性を検証する。

【００３１】

ホスト・システム３０８は、システム３０８内のサービスおよびリソースにアクセスするためにユーザ３０２を許可するために、すなわち、ユーザのアイデンティティとユーザ特権とを調整するために使用されるシステム・レジストリ３１０も含むことができる。たとえば、システム管理者は、特定のセキュリティ・グループに属するようにユーザのアイデンティティを構成している可能性があり、そのユーザは、全体としてそのセキュリティ・グループにとって使用可能になるように構成されたリソースのみにアクセスできるものとして制限される。このシステム内では、許可方式を課すための様々な周知の方法を使用することができる。

【００３２】

デジタル証明書を適正に妥当性検査または検証するために、アプリケーションは、その証明書が取り消されているかどうかをチェックしなければならない。認証局が証明書を発行する場合、認証局は、それによって証明書が識別される固有の通し番号を生成し、この通し番号は、Ｘ．５０９証明書内の「通し番号」フィールド内に保管される。典型的には、取り消されたＸ．５０９証明書は証明書の通し番号を介してＣＲＬ内で識別され、取り消された証明書の通し番号はＣＲＬ内の通し番号のリスト内に現れる。

【００３３】

証明書３０４が依然として有効であるかどうかを判断するために、アプリケーション３０６は、ＣＲＬリポジトリ３１２から証明書取り消しリスト（ＣＲＬ：certificate revocation list）を入手し、ＣＲＬを妥当性検査する。アプリケーション３０６は、証明書３０４内の通し番号を、検索されたＣＲＬ内の通し番号のリストと比較し、一致する通し番号がまったくない場合、アプリケーション３０６は証明書３０４を妥当性検査する。ＣＲＬが一致する通し番号を有する場合、証明書３０４は拒否されるはずであり、アプリケーション３０６は、任意の被制御リソースへのアクセスに関するユーザの要求を拒否するために適切な措置を講ずることができる。

【００３４】

ほとんどのデータ処理システムは、保護する必要のある機密データを収容している。たとえば、構成情報のデータ保全性は違法な変更から保護する必要があり、パスワード・ファイルなどのその他の情報は違法な開示から保護する必要がある。所与のデータ処理システムのオペレータは、データ処理システムを保護するために多種多様なタイプのセキュリティ・メカニズムを使用することができる。たとえば、データ処理システム上のオペレーティング・システムは、様々な認証および許可方式など、機密データを保護するための様々なソフトウェア・メカニズムを提供することができ、特定のハードウェア・デバイスおよびソフトウェア・アプリケーションは、ハードウェア・セキュリティ・トークンおよびバイオメトリック・センサ・デバイスなど、機密データを保護するためのハードウェア・メカニズムに依存することができる。機密データを保護するために所与のデータ処理システム内で複数のソフトウェアおよびハードウェア・メカニズムを使用できる場合でも、誰かが暗号化機密データに違法なアクセスをする場合に暗号化機密データを暗号化解除する能力がなければ暗号化機密データのコピーが無用なものになるように機密データを暗号化することもできる。

【００３５】

しかし、データ処理システム内に収容されているすべての情報を最終的に保護するための能力には限界がある。たとえば、パスワード・ファイルをさらに保護しようとして、マスタ・シークレットと呼ばれる場合が多いパスワードまたは暗号鍵などのさらに他の秘密を使用して、パスワード・ファイルを暗号化する場合もある。しかし、この新たな秘密も何らかの方法で保護する必要がある。したがって、システム管理者は、他のセキュリティ層を実現しようとすると、その結果、やはり保護する必要のある追加の機密情報が発生するという、ある種のジレンマに陥る可能性がある。ここで本発明を考慮すると、残りの図には、このジレンマを解決する本発明の模範的な諸実施形態が描かれている。

【００３６】

次に図５に関連して説明すると、このブロック図は、本発明の一実施形態により、データ処理システム内のハードウェア・セキュリティ・ユニット内で暗号機能を使用可能にするために取り外し可能ハードウェア・デバイスを受け入れるデータ処理システムの一部分を描写している。本発明では、暗号鍵を保持し、暗号化機能を実行する１対の突き合わせスマート・キー・デバイスを使用する。システム・ユニット４０２は、ポータブルまたは取り外し可能デバイスである外部スマート・キー・デバイス（ＥＸＳＫＤ：external smart key device）４０４とのインターフェースを取る。また、システム・ユニット４０２は、マザーボードなどの取り外し可能デバイスを受け入れるホスト・システムの一体部分である突き合わせデバイスである内部スマート・キー・デバイス（ＩＮＳＫＤ：internal smart key device）４０６も含む。内部スマート・キー・デバイスは、好ましくは、ホ

スト・システムから取り外しにくい、パッケージ化された集積回路であり、ハードウェア・セキュリティ・ユニットまたはデバイスとして記述される場合もあれば、命令を実行するための処理装置を有する場合もある。この例では、ＥＸＳＫＤ４０４とＩＮＳＫＤ４０６は対になったデバイスである。取り外し可能デバイスは、システム管理担当者、たとえば、ＩＴ管理者によって物理的に固定され、ＩＴ管理者が、ホスト・マシン上の突き合わせデバイス、すなわち、ＩＮＳＫＤ４０６によってのみ実行可能な特定の暗号機能を使用可能にする必要があるときに、取り外し可能デバイス、すなわち、ＥＸＳＫＤ４０４は、システム・ユニット４０２などのホスト・マシンに挿入される。換言すれば、特定の暗号機能は、外部スマート・キー・デバイスがシステム・ユニットに挿入されたときに使用可能になる。ＩＮＳＫＤ４０６のみが特定の暗号出力を生成するための１つまたは複数の特定の暗号秘密鍵を含んでいるので、ＩＴ管理者が必要とする結果は、ＩＮＳＫＤ４０６のみによって生成することができる。システム・ユニット４０２上のアプリケーション４０８は、ＥＸＳＫＤ４０４およびＩＮＳＫＤ４０６に類似しているソフトウェア・スマート・キー・ユニット（ＳＷＳＫＵ：software smart key unit）４１０を有する。アプリケーション４０８はＳＷＳＫＵ４１０を使用して特定の機能を実行するが、この機能については以下により詳細に説明する。

【００３７】

次に図６に関連して説明すると、このブロック図は、本発明の一実施形態により、内部スマート・キー・デバイスを含み、内部スマート・キー・デバイス内の暗号機能を使用可能にするために外部スマート・キー・デバイスを使用するシステム・ユニットを描写している。図６は、様々なコンポーネント内に保管されている暗号鍵に関する追加の詳細を含むことを除き、図５と同様のものである。

【００３８】

外部スマート・キー・デバイス（ＥＸＳＫＤ）５０２は取り外し可能ハードウェア・デバイスであり、ＥＸＳＫＤ５０２は、好ましくは、システム管理者によって制御され、ハードウェア・セキュリティ・トークンとして動作するポータブル・デバイスである。電気的インターフェース５０４を備えた外部スマート・キー・デバイス５０２は、電気的インターフェース５０８を備えたシステム・ユニット５０６内に挿入可能であり、外部スマート・キー・デバイス５０２およびシステム・ユニット５０６は、それぞれのインターフェースにより電気的に連結し、デジタル情報を表す電気信号を交換する。

【００３９】

外部スマート・キー・デバイス５０２は、外部スマート・キー・デバイス５０２内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン５１０を含む。ＥＸＳＫＤ秘密鍵５１２は、ＥＸＳＫＤ５０２の外部にあるエンティティによって読み取りまたはアクセスを行えないように保管されており、ＥＸＳＫＤ５０２は、ＥＸＳＫＤ秘密鍵５１２のコピーを伝送するかまたはその他の方法で提供するための機能を含んでいない。ＥＸＳＫＤ公開鍵証明書５１４は、非対称暗号鍵ペアとしてＥＸＳＫＤ秘密鍵５１２に対応するＥＸＳＫＤ公開鍵５１６のコピーを含む。また、ＥＸＳＫＤ５０２は、ＩＮＳＫＤ公開鍵証明書５１８のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＩＮＳＫＤ秘密鍵５２６に対応するＩＮＳＫＤ公開鍵５２０のコピーを含む。ＩＮＳＫＤ公開鍵証明書５１８のコピーは、その製造または初期設定プロセスの一部としてＥＸＳＫＤ５０２上に書き込むことができる。

【００４０】

システム・ユニット５０６は、内部スマート・キー・デバイス（ＩＮＳＫＤ）５２２を含む。内部スマート・キー・デバイス５２２は、内部スマート・キー・デバイス５２２内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン５２４を含む。ＩＮＳＫＤ秘密鍵５２６は、ＩＮＳＫＤ５２２の外部にあるエンティティによって読み取りまたはアクセスを行えないように保管されており、ＩＮＳＫＤ５２２は、ＩＮＳＫＤ秘密鍵５２６のコピーを伝送するかまたはその他の方法で提供するための機能を含んでいない。ＩＮＳＫＤ公開鍵証明書５２８は、非対称暗号鍵ペアとしてＩＮＳＫＤ

秘密鍵５２６に対応するＩＮＳＫＤ公開鍵５３０のコピーを含む。また、ＩＮＳＫＤ５２２は、ＥＸＳＫＤ公開鍵証明書５３２のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＥＸＳＫＤ秘密鍵５１２に対応するＥＸＳＫＤ公開鍵５３４のコピーを含む。ＥＸＳＫＤ公開鍵証明書５３２のコピーは、その製造または初期設定プロセスの一部としてＩＮＳＫＤ５２２上に書き込むことができる。

【００４１】

代替諸実施形態では、ＩＮＳＫＤ秘密鍵５２６およびＩＮＳＫＤ公開鍵５３０を他の機能に使用することができる。図６に図示されている好ましい一実施形態では、ＩＮＳＫＤ秘密鍵５２６およびＩＮＳＫＤ公開鍵５３０はＩＮＳＫＤ５２２とＥＸＳＫＤ５０２との間の通信のために予約されており、ＩＮＳＫＤ５２２は１つまたは複数の他の暗号鍵ペアを他の機能に使用する。この例では、ＩＮＳＫＤ５２２と、アプリケーション５４０内のソフトウェア・スマート・キー・ユニット（ＳＷＳＫＵ）５３８との間の通信を保護するために、ＩＮＳＫＤ５２２によってＩＮＳＫＤ＿ＳＷ秘密鍵５３６が使用される。ＩＮＳＫＤ＿ＳＷ公開鍵証明書５４２は、非対称暗号鍵ペアとしてＩＮＳＫＤ＿ＳＷ秘密鍵５３６に対応するＩＮＳＫＤ＿ＳＷ公開鍵５４４のコピーを含む。また、ＩＮＳＫＤ５２２は、ＳＷＳＫＵ公開鍵証明書５４６のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＳＷＳＫＵ秘密鍵５５０に対応するＳＷＳＫＵ公開鍵５４８のコピーを含む。

【００４２】

システム・ユニット５０６は、ＳＷＳＫＵ５３８を含むアプリケーション５４０の実行をサポートし、そのＳＷＳＫＵ自体は、ソフトウェア・スマート・キー・ユニット５３８内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン５５２を含む。ＳＷＳＫＵ５３８は、ＳＷＳＫＵ秘密鍵５５０のコピーを伝送するかまたはその他の方法で提供するための機能を含んでいない。ＳＷＳＫＵ公開鍵証明書５５４は、非対称暗号鍵ペアとしてＳＷＳＫＵ秘密鍵５５０に対応するＳＷＳＫＵ公開鍵５５６のコピーを含む。また、ＳＷＳＫＵ５３８は、ＩＮＳＫＤ＿ＳＷ公開鍵証明書５５８のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＩＮＳＫＤ＿ＳＷ秘密鍵５３６に対応するＩＮＳＫＤ＿ＳＷ公開鍵５６０のコピーを含む。以下にさらにより詳細に説明する通り、ＳＷＳＫＵ５３８にはデジタル署名を行うことができる。図６に図示されている例では、ＳＷＳＫＵ５３８は、ＩＮＳＫＤ＿ＳＷ秘密鍵５３６を使用してＳＷＳＫＵ５３８について計算されたデジタル署名５６２を含み、換言すれば、ＩＮＳＫＤ５２２は、ＩＮＳＫＤ＿ＳＷ秘密鍵５３６を使用してＳＷＳＫＵ５３８にデジタル署名を行う。

【００４３】

次に図７に関連して説明すると、この流れ図は、ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を描写している。このプロセスはブロック６０２から始まり、そこで、外部スマート・キー・デバイスが内部スマート・キー・デバイスを含むシステム・ユニットに電気的に連結される。たとえば、ＩＴ管理者は、外部スマート・キー・デバイスを受け入れるためのスロットを含む受け入れユニット（receiving unit）内に外部スマート・キー・デバイスを挿入することができる。次に、内部スマート・キー・デバイスおよび外部スマート・キー・デバイスは、ブロック６０４中に相互認証手順を実行し、その後、ブロック６０６中に内部スマート・キー・デバイスが暗号機能を実行できるようになり、プロセスが終了する。相互認証手順にエラーがあれば、その結果、内部スマート・キー・デバイスの使用不可が継続するものと想定することができる。あまり制限的ではない一実施形態では、ホスト・システム上で実行される任意のアプリケーションによって内部スマート・キー・デバイスの暗号機能を呼び出すことができる。より制限的な一実施形態では、図８に図示されている通り、ソフトウェア・スマート・キー・ユニットを含むアプリケーションのみによって内部スマート・キー・デバイスの暗号機能を呼び出すことができる。

【００４４】

次に図８に関連して説明すると、この流れ図は、本発明の一実施形態により、特定のソフトウェア・スマート・キー・ユニットによって使用するためのホスト・システムの内部

スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスを描写している。このプロセスはブロック７０２から始まり、そこで、たとえば、アプリケーション・プログラミング・インターフェース（ＡＰＩ：application programming interface）によりソフトウェア・スマート・キー・ユニットを含むアプリケーションまたはアプレットが内部スマート・キー・デバイスの暗号機能を呼び出す。次に、内部スマート・キー・デバイスおよびソフトウェア・スマート・キー・ユニットは、ブロック７０４中に相互認証手順を実行し、その後、ブロック７０６中に内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットに関する暗号機能を実行できるようになり、プロセスが終了する。ホスト・システム上の複数のソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスとの相互認証手順を完了したものと想定すると、複数のソフトウェア・スマート・キー・ユニットに代わって、内部スマート・キー・デバイスが同時に暗号機能を実行できるようにすることができる。

【００４５】

外部スマート・キー・デバイスが内部スマート・キー・デバイスを含むシステム・ユニットに連結されたままである間に、内部スマート・キー・デバイスは、認証局として動作する、すなわち、新しい公開証明書を生成するための機能を提供できるようになる。一実施形態では、新しいソフトウェア・パッケージをインストールするときに、内部スマート・キー・デバイスを含むシステム・ユニットに外部スマート・キー・デバイスを連結しなければならない。ソフトウェア・インストール中に新しいソフトウェア・パッケージに対して新しい公開証明書を発行することができ、新たに発行されたデジタル証明書内の公開鍵に対応する秘密鍵をソフトウェア・パッケージ内に組み込むことができ、内部スマート・キー・デバイスによってソフトウェア・パッケージに署名させることにより、秘密鍵を保護することができる。さらに、Ｊａｖａ（登録商標）環境では、悪意のあるユーザが秘密鍵を改ざんするのを防止するために、秘密鍵が組み込まれているＪＡＲファイルおよびＪａｖａ（登録商標）パッケージをさらにシールすることができる。

【００４６】

次に図９に関連して説明すると、この流れ図は、本発明の一実施形態により、ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用不可にするためのプロセスを描写している。このプロセスはブロック８０２から始まり、そこで、たとえば、外部スマート・キー・デバイスが挿入され、内部スマート・キー・デバイスが使用可能になった後の何らかのその後の時点で、外部スマート・キー・デバイスが内部スマート・キー・デバイスを含むシステム・ユニットから電気的に分離される。システム・ユニットが外部スマート・キー・デバイスの分離を検出すると、ブロック８０４中に内部スマート・キー・デバイスがさらに暗号機能を実行できない状態になり、プロセスが終了する。

【００４７】

図９に図示されているプロセスは、図７または図８に図示されているプロセスのいずれかに対する補足的プロセスとして機能する。しかし、本発明の実現例次第で、完全に使用不可にならないようないくつかの機能を内部スマート・キー・デバイスが依然として実行できることに留意されたい。内部スマート・キー・デバイス内の暗号機能は、ソフトウェアまたはハードウェアにより使用可能または使用不可にすることができるものと想定することができる。たとえば、ハードウェア・モードでは、外部スマート・キー・デバイスが受け入れられているかどうかを表す使用可能化状態に基づいて設定またはクリアしなければならない特定のフリップフロップまたはその他のメカニズムにより、内部スマート・キー・デバイス内の特定の回路の動作が動作可能状態に入ることを防止できる可能性があり、ソフトウェア・モードでは、暗号機能の実行を論理的に制御する特殊な使用可能化フラグを設定しクリアすることにより、特定の暗号機能の動作を保護することができる。

【００４８】

次に図１０～図１１に関連して説明すると、この１対の流れ図は、図７のブロック６０４に図示されている相互認証手順に関する詳細を描写している。図１０は内部スマート・キー・デバイスが外部スマート・キー・デバイスを認証するためのプロセスを描写し、図

１１は外部スマート・キー・デバイスが内部スマート・キー・デバイスを認証するためのプロセスを描写している。図１０に図示されているプロセスは図１１に図示されているプロセスより前に実行することができ、逆の場合も同様であり、本発明が実現される方法次第で、これらのプロセスは独立したものになる場合もあれば、たとえば、試行中の動作を示す適切な信号または状況フラグにより、同時に実行される場合もある。

【００４９】

次に図１０を参照すると、このプロセスはブロック９０２から始まり、そこで、内部スマート・キー・デバイスが外部スマート・キー・デバイスの公開鍵を使用して、メッセージ、たとえば、ランダム・テキスト・ストリングを暗号化する。ブロック９０４中に内部スマート・キー・デバイスは、ホスト・システムの適切なインターフェースにより、暗号化メッセージを外部スマート・キー・デバイスに転送し、次にその外部スマート・キー・デバイスがブロック９０６中にその秘密鍵により暗号化メッセージを暗号化解除する。次に外部スマート・キー・デバイスは、ブロック９０８中に内部スマート・キー・デバイスの公開鍵により暗号化解除メッセージを暗号化し、ブロック９１０中に暗号化メッセージを内部スマート・キー・デバイスに渡す。次に内部スマート・キー・デバイスは、ブロック９１２中にその秘密鍵により暗号化メッセージを暗号化解除し、ブロック９１４中に受信メッセージをその元のメッセージと比較する。２つのメッセージが一致する場合、ブロック９１６中に内部スマート・キー・デバイスは、たとえば、適切な信号によるかまたは論理フラグ変数を設定することにより、外部スマート・キー・デバイスが本物であると内部スマート・キー・デバイスが判断したことを示し、それによりプロセスを終了する。

【００５０】

次に図１１を参照すると、このプロセスはブロック９２２から始まり、そこで、外部スマート・キー・デバイスが内部スマート・キー・デバイスの公開鍵を使用して、メッセージ、たとえば、ランダム・テキスト・ストリングを暗号化する。ブロック９２４中に外部スマート・キー・デバイスは暗号化メッセージを内部スマート・キー・デバイスに転送し、次にその内部スマート・キー・デバイスがブロック９２６中にその秘密鍵により暗号化メッセージを暗号化解除する。次に内部スマート・キー・デバイスは、ブロック９２８中に外部スマート・キー・デバイスの公開鍵により暗号化解除メッセージを暗号化し、ブロック９３０中に暗号化メッセージを外部スマート・キー・デバイスに渡す。次に外部スマート・キー・デバイスは、ブロック９３２中にその秘密鍵により暗号化メッセージを暗号化解除し、ブロック９３４中に受信メッセージをその元のメッセージと比較する。２つのメッセージが一致する場合、ブロック９３６中に外部スマート・キー・デバイスは、たとえば、適切な信号によるかまたは論理フラグ変数を設定することにより、内部スマート・キー・デバイスが本物であると外部スマート・キー・デバイスが判断したことを示し、それによりプロセスを終了する。

【００５１】

次に図１２～図１３に関連して説明すると、この１対の流れ図は、図８のブロック７０４に図示されている相互認証手順に関する詳細を描写している。図１２はソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスを認証するためのプロセスを描写し、図１３は内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットを認証するためのプロセスを描写している。図１２に図示されているプロセスは図１３に図示されているプロセスより前に実行することができ、逆の場合も同様であり、本発明が実現される方法次第で、これらのプロセスは独立したものになる場合もあれば、たとえば、試行中の動作を示す適切なメッセージまたは状況フラグにより、同時に実行される場合もある。

【００５２】

次に図１２を参照すると、このプロセスはブロック１００２から始まり、そこで、ソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスの公開鍵を使用して、メッセージ、たとえば、ランダム・テキスト・ストリングを暗号化する。ブロック１００４中にソフトウェア・スマート・キー・ユニットは暗号化メッセージを内部スマート

・キー・デバイスに転送し、次にその内部スマート・キー・デバイスがブロック１００６中にその秘密鍵により暗号化メッセージを暗号化解除する。次に内部スマート・キー・デバイスは、ブロック１００８中にソフトウェア・スマート・キー・ユニットの公開鍵により暗号化解除メッセージを暗号化し、ブロック１０１０中に暗号化メッセージをソフトウェア・スマート・キー・ユニットに渡す。次にソフトウェア・スマート・キー・ユニットは、ブロック１０１２中にその秘密鍵により暗号化メッセージを暗号化解除し、ブロック１０１４中に受信メッセージをその元のメッセージと比較する。２つのメッセージが一致する場合、ブロック１０１６中にソフトウェア・スマート・キー・ユニットは、たとえば、適切なメッセージによるかまたは論理フラグ変数を設定することにより、内部スマート・キー・デバイスが本物であるとソフトウェア・スマート・キー・ユニットが判断したことを示し、それによりプロセスを終了する。

【００５３】

図１２とは対照的に、図１３は、２つのエンティティ間で渡されるメッセージとして、ランダム・テキスト・ストリングの代わりにセッション・キーを使用する場合を例示している。セッション・キーは、２つのエンティティ間の相互認証プロセスが正常に完了した場合に２つのエンティティ間のセッション中に後続メッセージ・トラフィックを保護するために使用され、ソフトウェア・エンティティの実行の終了またはハードウェア・エンティティの電源遮断などの特定のイベントによって、セッションが時間設定される場合もあれば、セッションが終了する場合もある。セッション・キーは、暗号化の前に他の情報を含むより大きいメッセージ内に入れることができ、その後、暗号化メッセージが２つのエンティティ間で渡される。代替一実施形態では、ランダム・テキスト・ストリングを認証手順に使用することができ、その後、２つのエンティティがセッション・キーを交換することができる。以下にさらにより詳細に説明する通り、情報を交換するために使用されるアクションの数を削減するために、認証プロセス中に２つのエンティティ間で追加の情報を安全に渡すことができる。

【００５４】

次に図１３を参照すると、このプロセスはブロック１０２２から始まり、そこで、内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットの公開鍵を使用して、セッション・キーを暗号化する。ブロック１０２４中に内部スマート・キー・デバイスは暗号化セッション・キーをソフトウェア・スマート・キー・ユニットに転送し、次にそのソフトウェア・スマート・キー・ユニットがブロック１０２６中にその秘密鍵により暗号化セッション・キーを暗号化解除する。次にソフトウェア・スマート・キー・ユニットは、ブロック１０２８中に内部スマート・キー・デバイスの公開鍵により暗号化解除セッション・キーを暗号化し、ブロック１０３０中に暗号化セッション・キーを内部スマート・キー・デバイスに渡す。次に内部スマート・キー・デバイスは、ブロック１０３２中にその秘密鍵により暗号化セッション・キーを暗号化解除し、ブロック１０３４中に受信セッション・キーをその元のセッション・キーと比較する。２つのバージョンのセッション・キーが一致する場合、ブロック１０３６中に内部スマート・キー・デバイスは、たとえば、適切なメッセージによるかまたは論理フラグ変数を設定することにより、ソフトウェア・スマート・キー・ユニットが本物であると内部スマート・キー・デバイスが判断したことを示し、それによりプロセスを終了する。

【００５５】

図８に図示されているプロセスに併せて追加のセキュリティ・アクションを実行することができる。たとえば、ブロック７０２では、アプリケーションまたはアプレットは、内部スマート・キー・デバイス内に組み込まれた機能の使用を要求している。図１３に図示されているプロセスを開始する前の何らかの時点で、内部スマート・キー・デバイスは、要求側アプリケーションまたはアプレット内のソフトウェア・スマート・キー・ユニットが安全なコードを含むかどうかを検証する追加のアクションを実行することができる。図６に関して前述した通り、ＳＷＳＫＵ５３８にはデジタル署名を行うことができ、ＳＷＳＫＵ５３８は、ＩＮＳＫＤ＿ＳＷ秘密鍵５３６を使用してＳＷＳＫＵ５３８について計算

されたデジタル署名５６２を含む。このため、内部スマート・キー・デバイスは、ソフトウェア・スマート・キー・ユニットに関連するデジタル署名を検証することにより、要求側アプリケーションまたはアプレット内のソフトウェア・スマート・キー・ユニットが安全なコードを含むかどうかを検証することができる。

【００５６】

Ｊａｖａ（登録商標）環境では、ソフトウェア・スマート・キー・ユニットを署名付きＪＡＲファイルとして実現することができ、一実施形態では、内部スマート・キー・デバイスを使用して、署名付きＪＡＲファイルのデジタル署名を検証する。異なる環境では、シールされたＪＡＲファイルからパッケージ内のすべてのコードをロードしなければならないことをクラス・ローダが強制するように、ＪＡＲファイルおよびＪａｖａ（登録商標）パッケージをさらにシールすることができる。ＪＡＲファイルおよびＪａｖａ（登録商標）パッケージをシールする行為は、クラス・パス内にコードを注入することにより悪意のあるユーザによって機能が変更されるのを防止することができる。その上、クラス・ローダそのものには、クラス・ローダの保全性を検証できるように署名しシールすることができる。

【００５７】

より汎用的な計算環境では、内部スマート・キー・デバイスはソフトウェア・スマート・キー・ユニットにデジタル署名を行い、あとでデジタル署名を妥当性検査することができるが、ソフトウェア・スマート・キー・ユニットが署名され妥当性検査されることを保証するプロセスは、内部スマート・キー・デバイスからの援助によってデータ処理システム内の適切なオペレーティング・システム・モジュールにより、たとえば、実行のためにソフトウェア・モジュールをロードするプログラム・ローダにより、制御することができる。ソフトウェア・モジュールが実行できるようにする前に、プログラム・ローダは追加のセキュリティ・プロセスを実行できるであろう。その上、プログラム・ローダそのものには、プログラム・ローダの保全性を検証できるように署名しシールすることができる。

【００５８】

前述のプロセスはソフトウェア・スマート・キー・ユニットの保全性を確保するためのメカニズムを提供しているが、ソフトウェア・スマート・キー・ユニットを含むコードを検査することにより、その暗号鍵を表示しコピーすることができるので、データ処理システム内のソフトウェア・スマート・キー・ユニットの動作は依然としていくらか脆弱なものと見なされる可能性があり、暗号鍵はソフトウェア・スマート・キー・ユニット内に明文で保管されているものと想定することができる。

【００５９】

このため、ソフトウェア・スマート・キー・ユニット、特にその秘密鍵を保護するために、図８に図示されているプロセスに併せて、さらに他のセキュリティ・アクションを実行することができる。前の何らかの時点で、ソフトウェア・スマート・キー・ユニットを暗号化し、それにより、ソフトウェア・スマート・キー・ユニット内の任意の機密情報、特にその秘密鍵を隠すことができる。異なる一実施形態では、ソフトウェア・スマート・キー・ユニットを含むソフトウェア・モジュールを暗号化することができるであろう。たとえば、ソフトウェア・モジュールをデータ処理システム上にインストールするときに、データ処理システム上の内部スマート・キー・デバイスは、そのソフトウェア・モジュールを含むアプリケーション・プログラム用のインストール手順の一部として、そのソフトウェア・モジュールを暗号化することができるであろう。

【００６０】

この追加のアクションが実行されるシステムでは、ソフトウェア・スマート・キー・ユニットまたはソフトウェア・スマート・キー・ユニットを含むソフトウェア・モジュールあるいはその両方は、実行可能になる前に暗号化解除を必要とするであろう。デジタル署名を使用するソフトウェア・スマート・キー・ユニットの保全性の保護に関して上述されたものと同様のある時点で、たとえば、図１３に図示されているプロセスを開始する前の何らかの時点で、内部スマート・キー・デバイスは、ソフトウェア・スマート・キー・ユ

ニットまたはソフトウェア・スマート・キー・ユニットを含むソフトウェア・モジュールあるいはその両方を暗号化解除する追加のアクションを実行するであろう。この場合も、上述したものと同様に、内部スマート・キー・デバイスからの援助によって、データ処理システム内の適切なオペレーティング・システム・モジュールにより暗号化解除プロセスを制御することができる。内部スマート・キー・デバイスに併せて使用するためにインストール時にソフトウェア・モジュールを変更するプロセスならびにこのようなソフトウェア・モジュールを安全に実行するプロセスに関する詳細については以下に示す。

【００６１】

次に図１４に関連して説明すると、この流れ図は、外部スマート・キー・デバイスの存在に基づいて動作が使用可能または使用不可になるソフトウェア・スマート・キー・ユニットによって要求された動作を実行するための内部スマート・キー・デバイス内のプロセスを描写している。このプロセスはブロック１１０２から始まり、そこで、内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットから要求メッセージを受信し、要求メッセージは、ソフトウェア・スマート・キー・ユニットによって要求されている動作のタイプを示すメッセージタイプ変数を含む。次にブロック１１０４中に、ソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスによって認証されているかどうかに関する判断が行われ、この判断は、たとえば、図１３に関して上述した通り、前の認証手順中に内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットに渡したセッション・キーを使用して受信メッセージの内容を正常に暗号化解除することにより実行することができる。ソフトウェア・スマート・キー・ユニットが認証されていない場合、ブロック１１０６中に内部スマート・キー・デバイスが適切なエラー応答を生成し、ブロック１１０８中に要求側ソフトウェア・スマート・キー・ユニットに応答メッセージを返し、それにより、プロセスを終了する。

【００６２】

ソフトウェア・スマート・キー・ユニットが認証されている場合、ブロック１１１０中に外部スマート・キー・デバイスが依然としてシステム・ユニットに電気的に連結されているかどうかを内部スマート・キー・デバイスが判断する。たとえば、この判断は単に、システム・ユニットと外部スマート・キー・デバイスとの間の電気接続が切断された場合にクリアされたと思われる特殊レジスタのチェックを必要とするだけである場合もある。外部スマート・キー・デバイスがシステム・ユニットに電気的に連結されていない場合、内部スマート・キー・デバイスは、ブロック１１０６でエラー応答を生成し、ブロック１１０８で応答メッセージをソフトウェア・スマート・キー・ユニットに返し、それにより、プロセスを終了する。

【００６３】

ソフトウェア・スマート・キー・ユニットが認証されており、外部スマート・キー・デバイスが依然としてシステム・ユニットに電気的に連結されている場合、内部スマート・キー・デバイスは、可能であれば、ソフトウェア・スマート・キー・ユニットのために要求された機能を実行する。ブロック１１１２およびブロック１１１４は、内部スマート・キー・デバイスによって提供される可能性のある機能の例を描写しており、これらの例の列挙は、本発明の他の実現例でその他の機能が使用可能ではない可能性があることを暗示しているわけではない。好ましい一実施形態では、内部スマート・キー・デバイスは、相互認証後に外部スマート・キー・デバイスが内部スマート・キー・デバイスに電気的に連結されたままである場合のみ、認証局として動作しながら新しいデジタル証明書を発行する機能と、内部スマート・キー・デバイスの秘密鍵を使用してソフトウェア・モジュールに署名する機能であって、その秘密鍵が使用可能な公開鍵証明書に対応する機能を実行する。本発明は内部スマート・キー・デバイスの秘密鍵を検索するためのいかなるインターフェースも許可せず、このため、その秘密鍵を使用して署名動作を実行することは内部スマート・キー・デバイスのみによって実行可能であることに留意されたい。

【００６４】

ソフトウェア・スマート・キー・ユニットが、要求メッセージ内に組み込まれたデータ

項目に対するデジタル署名を要求している場合、ブロック１１１２中に内部スマート・キー・デバイスは、適切な秘密鍵を使用してデータ項目についてデジタル署名を計算し、デジタル署名を（好ましくは、それが返すデータ項目のコピーとともに）応答メッセージ内に挿入する。ソフトウェア・スマート・キー・ユニットがデジタル証明書を要求している場合、ブロック１１１４中に内部スマート・キー・デバイスは、適切な秘密鍵を使用してデジタル証明書を生成し、デジタル証明書を応答メッセージ内に挿入し、デジタル証明書は、要求メッセージ内でソフトウェア・スマート・キー・ユニットによって提供された様々な識別情報を含む可能性がある。適切なセッション・キーにより任意の機密データを暗号化することを含むものと思われる適切な応答メッセージが生成された後、ブロック１１０８で応答メッセージがソフトウェア・スマート・キー・ユニットに返され、プロセスが終了する。

【００６５】

もう一度、ブロック１１１２を参照すると、任意のタイプのデジタル・データ項目に署名することができる。もう一度、図５を参照すると、アプリケーション４０８は、本発明の機能を取り込むことができる多種多様なタイプのアプリケーションを表している。一実施形態では、このアプリケーションは、Ｊａｖａ（登録商標）ＪＡＲファイル、アプリケーション・サーバによって直接生成されたファイル、またはホスト・システム上の他のアプリケーションに代わって生成されたファイルに署名するアプリケーション・サーバにすることができる。特定のケースでは、新たに生成されたＪＡＲファイルはそれ自体が、ホスト・システムの内部スマート・キー・デバイス内の機能を呼び出すことができるソフトウェア・スマート・キー・ユニットを含むことができる。

【００６６】

次に図１５に関連して説明すると、この流れ図は、外部スマート・キー・デバイスの存在によって動作が使用可能になる必要がないソフトウェア・スマート・キー・ユニットによって要求された動作を実行するための内部スマート・キー・デバイス内のプロセスを描写している。このプロセスはブロック１１２２から始まり、そこで、内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットから要求メッセージを受信し、要求メッセージは、ソフトウェア・スマート・キー・ユニットによって要求されている動作のタイプを示すメッセージタイプ変数を含む。次にブロック１１２４中に、ソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスによって認証されているかどうかに関する判断が行われ、この判断は、たとえば、図１３に関して上述した通り、前の認証手順中に内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットに渡したセッション・キーを使用して受信メッセージの内容を正常に暗号化解除することにより実行することができる。ソフトウェア・スマート・キー・ユニットが認証されていない場合、ブロック１１２６中に内部スマート・キー・デバイスが適切なエラー応答を生成し、ブロック１１２８中に要求側ソフトウェア・スマート・キー・ユニットに応答メッセージを返し、それにより、プロセスを終了する。

【００６７】

ソフトウェア・スマート・キー・ユニットが認証されている場合、内部スマート・キー・デバイスは、可能であれば、ソフトウェア・スマート・キー・ユニットのために要求された機能を実行する。ブロック１１３０およびブロック１１３２は、内部スマート・キー・デバイスによって提供される可能性のある機能の例を描写しており、これらの例の列挙は、本発明の他の実現例でその他の機能が使用可能ではない可能性があることを暗示しているわけではない。好ましい一実施形態では、外部スマート・キー・デバイスの存在なしで、必要なキーが与えられた暗号化および暗号化解除の機能、証明書が与えられたデジタル署名を妥当性検査する機能、ソフトウェア・スマート・キー・ユニットを相互に認証する機能、ならびに保管された機密情報が相互に認証されたソフトウェア・スマート・キー・ユニットによって読み取り／書き込みアクセスできるようにする機能が、内部スマート・キー・デバイスによって実行されるであろう。

【００６８】

ソフトウェア・スマート・キー・ユニットが、要求メッセージ内に組み込まれたマスタ・シークレットの登録を要求している場合、ブロック１１３０中に内部スマート・キー・デバイスは、ソフトウェア・スマート・キー・ユニットに関する何らかの識別情報に関連してマスタ・シークレットを保管し、応答メッセージを生成する。ソフトウェア・スマート・キー・ユニットが、前に登録されたマスタ・シークレットの検索を要求している場合、ブロック１１３２中に内部スマート・キー・デバイスは、ソフトウェア・スマート・キー・ユニットのアイデンティティに基づいてマスタ・シークレットを検索し、応答メッセージを生成する。適切なセッション・キーにより任意の機密データを暗号化することを含むものと思われる適切な応答メッセージが生成された後、ブロック１１２８で応答メッセージがソフトウェア・スマート・キー・ユニットに返され、プロセスが終了する。

【００６９】

このように、デジタル証明書の発行など、特に機密の動作が内部スマート・キー・デバイスによって実行される必要がある場合に、外部スマート・キー・デバイスを内部スマート・キー・デバイスに電気的に連結された状態に保持することが必要であるだけである。図１５に関して上述した通り、ソフトウェア・スマート・キー・ユニットは、外部スマート・キー・デバイスの存在を要求せずに、内部スマート・キー・デバイスと相互に認証した後、暗号鍵などの機密情報を内部スマート・キー・デバイスに保管することができ、機密情報は同じソフトウェア・スマート・キー・ユニットのみによって検索することができる。

【００７０】

ソフトウェア・スマート・キー・ユニットは、外部スマート・キー・デバイスから独立した方法で内部スマート・キー・デバイスと相互に認証することができるので、この手法は有利なものである。たとえば、この手法は、無人モード、すなわち、外部スマート・キー・デバイスを挿入するための人間が存在しないモードでのソフトウェア・プログラムの始動を可能にし、プログラムは、前に署名されシールされたソフトウェア・スマート・キー・ユニットを使用して、内部スマート・キー・デバイスから任意の機密情報を検索することができる。ソフトウェア・プログラムは、内部スマート・キー・デバイスからマスタ・シークレットを検索し、パスワードおよびその他の暗号化構成情報を暗号化解除して、人間の介入なしで安全に始動プロセスを完了することができる。

【００７１】

次に図１６に関連して説明すると、このブロック図は、マスタ・シークレットを保護するための本発明の一実施形態を例示している。上記の通り、データ処理システム上に保管されている秘密情報はマスタ・シークレットにより暗号化することができ、これはマスタ・シークレットを保護する必要性を伴うものである。従来技術のシステムでは、マスタ・シークレットの保護は、典型的には、マスタ・シークレットが使用されているホスト・システムの外部にあるメカニズムにより保護される。典型的な従来技術のシステムとは対照的に、本発明の一実施形態を使用すると、マスタ・シークレットが使用されることになるホスト・システム上でマスタ・シークレットを保護することができる。

【００７２】

図１６は図５と同様のものであり、システム・ユニット１２０２は外部スマート・キー・デバイス１２０４とのインターフェースを取り、システム・ユニット１２０２は内部スマート・キー・デバイス１２０６も含む。また、システム・ユニット１２０２は、ソフトウェア・スマート・キー・ユニット１２０８～１２１２もサポートする。しかし、図５とは対照的に、図１６の内部スマート・キー・デバイス１２０６は、パスワード、暗号化キー、または何らかのその他の形式である可能性のあるマスタ・シークレットを保護するためにマスタ・シークレット・レジストリ１２１４を含むように強化されている。図１５のブロック１１３０および１１３２に関して簡単に上述した通り、ソフトウェア・スマート・キー・ユニット１２０８～１２１２は、安全な要求／応答メカニズムにより内部スマート・キー・デバイス１２０６にマスタ・シークレットを保管することができる。内部スマート・キー・デバイス１２０６は、要求側ソフトウェア・スマート・キー・ユニットに関

する識別情報に関連してソフトウェア・スマート・キー・ユニット１２０８～１２１２からマスタ・シークレットを保管する。たとえば、マスタ・シークレット・レジストリ１２１４はマスタ・シークレット１２１８に関連するＳＷＳＫＵ　ＩＤ１２１６を含み、ＳＷＳＫＵ　ＩＤ１２１６について実行される可能性のあるルックアップ動作はそれをマスタ・シークレット１２１８に関連付けることになるである。代わって、マスタ・シークレット・レジストリ１２１４はソフトウェア・スマート・キー・ユニットあたり２つ以上のマスタ・シークレットをサポートすることができ、それぞれの要求された動作により、適宜、マスタ・シークレットのグループを登録または検索することができる。図１５は登録動作および検索動作を例示しているだけであるが、マスタ・シークレットの管理に関連する可能性のあるその他の動作、たとえば、削除動作または上書き動作もサポートすることができる。

【００７３】

図１３の説明で上記の通り、情報を交換するために使用されるアクションの数を削減するために、認証プロセス中に内部スマート・キー・デバイスとソフトウェア・スマート・キー・ユニットとの間で追加の情報を安全に渡すことができる。そのために、認証プロセス中にソフトウェア・スマート・キー・ユニットに関するマスタ・シークレットを渡すことができる。本物のソフトウェア・スマート・キー・ユニットはそのソフトウェア・スマート・キー・ユニットの秘密鍵のコピーを備えているべき唯一のエンティティであるので、認証プロセス中に内部スマート・キー・デバイスによって提供されるソフトウェア・スマート・キー・ユニットのマスタ・シークレットを暗号化解除できるのは、そのソフトウェア・スマート・キー・ユニットのみでなければならない。

【００７４】

次に図１７～図１９に関連して説明すると、これらのブロック図は、複数の外部スマート・キー・デバイスと複数の内部スマート・キー・デバイスとの種々の関係を例示している。これまでの図面の説明は、外部スマート・キー・デバイスと内部スマート・キー・デバイスとの間に固有の１対１の関係が存在することを暗示しているように思われる可能性がある。図１７を参照すると、複数の外部スマート・キー・デバイス１３０４～１３０８のいずれかを使用することにより、単独の内部スマート・キー・デバイス１３０２を使用可能にすることができる。たとえば、小規模グループのＩＴ管理者のそれぞれは、内部スマート・キー・デバイス１３０２を含む特定のサーバ・マシン内に挿入することができる取り外し可能スマート・キー・デバイスを有することができる。図１８を参照すると、単独の外部スマート・キー・デバイス１４０２は、複数の内部スマート・キー・デバイス１４０４～１４０８のいずれかを使用可能にすることができる。たとえば、ＩＴ管理者は、そのそれぞれが内部スマート・キー・デバイス１４０４～１４０８のうちの１つだけを含む複数のサーバ・マシンで単一の取り外し可能スマート・キー・デバイスを使用することができる。図１９を参照すると、複数の外部スマート・キー・デバイス１５０２～１５０６は、複数の内部スマート・キー・デバイス１５１２～１５１６のうちのいずれかを使用可能にすることができる。たとえば、小規模グループのＩＴ管理者のそれぞれは、そのそれぞれが内部スマート・キー・デバイス１５１２～１５１６のうちの１つだけを含む多種多様なサーバ・マシン内に挿入することができる取り外し可能スマート・キー・デバイスを有することができる。所与のスマート・キー・デバイス上で多対１の関係または１対多の関係をサポートするために、所与のスマート・キー・デバイスは、追加の対応する内部スマート・キー・デバイスまたは外部スマート・キー・デバイスあるいはその両方に関する追加の公開鍵証明書の保管または構成のみを必要とする。

【００７５】

本発明の追加の諸実施形態について論ずる前に、本発明の追加の諸実施形態の動作効率およびその他の利点を評価するために、デジタル証明書に基づく信頼関係に関する何らかの背景情報を提供する。

【００７６】

次に図２０～図２２に関連して説明すると、各ブロック図は、典型的な１組の信頼でき

る関係を描写している。次に図２０を参照すると、認証局１６０２は、サーバ１６０４および１６０６にデジタル証明書を発行している。上記の通り、認証局は、おそらく人間のユーザである他のエンティティに代わって、しかし、アプリケーションまたはデータ処理システムなどのプログラマチック・エンティティまたはハードウェア・エンティティに代わって、デジタル証明書を発行する、信頼できるエンティティである。したがって、サーバ１６０４および１６０６は、図３または図４に図示されているユーザ２０２または３０２などのユーザによって表されている可能性があり、代わって、サーバ１６０４および１６０６は、図５に図示されているアプリケーション４０８などの何らかの他のタイプのプログラマチック・エンティティである可能性がある。認証局１６０２は、サーバ１６０４および１６０６にデジタル証明書を発行している。サーバ１６０４および１６０６は、本発明によって記載されている通り、その後、認証局１６０２との相互認証を実行することにより、認証局１６０２との信頼関係１６０８および１６１０を確立することができる。何らかの時点で、サーバ１６０４は、サーバ１６０６によって提供されるサービスを要求しながら、対応する秘密鍵の所有証明、たとえば、その秘密鍵を使用して署名されているデータ項目とともに、そのデジタル証明書をサーバ１６０６に提示することができる。サーバ１６０６は認証局１６０２を信頼しているので、サーバ１６０６は、サーバ１６０４から受信したデジタル証明書が認証局１６０２によって署名されたことを検証することにより、サーバ１６０４を認証することができる。逆の状況もまた真実であり、サーバ１６０４はサーバ１６０６を認証できるであろう。このように、サーバ１６０４およびサーバ１６０６は両者間の信頼関係１６１２を確立することができる。

【００７７】

図２１を参照すると、サーバ１６１４は、サーバ１６０６との信頼関係１６１６を確立している。この例では、信頼関係１６１６の根拠はまったく示されておらず、サーバ１６０４はサーバ１６１４との信頼関係１６１６を受け入れていない。

【００７８】

図２２を参照すると、同様の参照番号は図２０に図示されているものと同様の要素を指し示しているが、図２２は図２０に図示されているものに対する追加の要素を図示している。認証局１６２０はサーバ１６０６および１６２２にデジタル証明書を発行している。認証局１６２０がサーバ１６０６および１６２２にデジタル証明書を発行している場合、認証局は、サーバ１６０６および１６２２との信頼関係１６２４および１６２６をそれぞれ確立していると言われる。何らかの時点で、サーバ１６２２は、サーバ１６０６によって提供されるサービスを要求しながら、サーバ１６０６にそのデジタル証明書を提示することができる。サーバ１６２２は認証局１６２０を信頼しているので、サーバ１６０６は、サーバ１６２２から受信したデジタル証明書が認証局１６２０によって署名されたことを検証することにより、サーバ１６２２を認証することができる。逆の状況もまた真実であり、サーバ１６２２はサーバ１６０６を認証できるであろう。このように、サーバ１６２２およびサーバ１６０６は両者間の信頼関係１６２８を確立することができる。

【００７９】

信頼関係は推移的である可能性がある。図２１に関して上記した通り、サーバ１６０６はサーバ１６１４との信頼関係１６１６を確立している。しかし、おそらく、サーバ１６０６は信頼関係１６１６の根拠に関する十分な情報を提供できなかったので、サーバ１６０４は信頼関係１６１６を承認しなかった。しかし、図２２では、サーバ１６０６は、サーバ１６０６が信頼関係を確立しているサーバ間のその信頼できる関係に関する十分な情報を提供することができる。この例では、サーバ１６０６はサーバ１６０４に信頼関係１６２８に関する情報を提供する。サーバ１６０４とサーバ１６０６との間の信頼関係１６１２ならびにサーバ１６０６とサーバ１６２２との間の信頼関係１６２８を考慮すると、サーバ１６０４およびサーバ１６２２は、サーバ１６０４とサーバ１６２２との間の推移的な信頼関係１６３０を確立することができる。サーバは、前述した証明書管理プロトコルにより証明書を転送することができる。

【００８０】

このように、サーバは、サーバと認証局との間で複雑な階層型信頼関係を形成することができる。各認証局は木構造のルートと見なすことができ、特に木構造内の他のエンティティが２次認証局としても動作するときに、ある認証局がルート局（root authority）と呼ばれる場合もある。複数の認証局を使用すると、たとえば、図２２に図示されている通り、複数の木構造がオーバラップすることができる。次に本発明に戻ると、残りの図には、上述した内部および外部スマート・キー・デバイスの利点を使用して信頼モデルを構築するように本発明が実現される本発明の諸実施形態の例が描かれている。

【００８１】

次に図２３に関連して説明すると、このブロック図は、本発明の一実施形態により、内部スマート・キー・デバイスによって提供される信頼に基づく信頼関係から構築される信頼モデルの一例を描写している。本発明の内部スマート・キー・デバイスは、認証局として動作する際に高レベルの信頼性を提供する。他の図に関して上述した通り、内部スマート・キー・デバイスは、情報を保護するためのメカニズムを提供する。図１４および図１５に関して上述した通り、内部スマート・キー・デバイスによって提供可能な機能の１つはデジタル証明書の発行である。内部スマート・キー・デバイスは、たとえば、マザーボード上の特殊チップなど、データ処理システム内のシステム・ユニットの一部として実現されることになるので、内部スマート・キー・デバイスは、物理的に保護し、それにより、悪意のあるユーザが不適切なスキームを実現するのを困難にしなければならない。加えて、内部スマート・キー・デバイスによるデジタル証明書の発行が外部スマート・キー・デバイスの使用によりシステム管理者によって制御できることによって、内部スマート・キー・デバイスの信頼性が強化される。このため、内部スマート・キー・デバイスがデジタル証明書を発行する能力により、内部スマート・キー・デバイスは信頼モデルの基礎として動作することができる。

【００８２】

このように、種々のタイプのエンティティ、たとえば、種々の種類のハードウェアおよびソフトウェア・コンピューティング・リソースは、それらと、ハードウェアベースの認証局として動作する内部スマート・キー・デバイスとの間で複雑な階層型信頼関係を形成することができる。この信頼モデルでは、信頼は、データ処理システム上の内部スマート・キー・デバイスによって提供される認証局機能に根付いている。信頼関係階層は、図２３のように、内部スマート・キー・デバイスが逆ピラミッドの頂点にある逆ピラミッドによって表すことができ、コンピューティング・リソースがその逆ピラミッドを形成する。分散データ処理環境では、図２３に図示されている通り、信頼関係はオーバラップする逆ピラミッドの集合として見ることができ、それぞれのピラミッドは各マシン上の内部スマート・キー・デバイスに基づくものである。

【００８３】

図２３では、信頼モデルの一例が２つの内部スマート・キー・デバイス１７０２および１７０４を示しており、その内部スマート・キー・デバイスはそれぞれ、各内部スマート・キー・デバイスが認証局として動作できるようにするための機能を含む認証局モジュール１７０６および１７０８を含む。内部スマート・キー・デバイス１７０４は２次ソフトウェア認証局モジュール１７１０に証明書を発行しており、そのモジュールは、内部スマート・キー・デバイス１７０４が存在する同じシステム・ユニット上で実行されるソフトウェア・アプリケーションである。２次ソフトウェア認証局モジュール１７１０など、データ処理システム内の階層的に上位のソフトウェア認証局モジュールは、データ処理システム上の内部スマート・キー・デバイス、すなわち、内部スマート・キー・デバイス１７０４の認証局機能によって提供されるルート信頼など、階層的に下位のソフトウェア認証局から権限を得る。たとえば、内部スマート・キー・デバイス１７０４は２次ソフトウェア認証局モジュール１７１０のデジタル証明書に署名することができ、そのモジュールはそれが発行するデジタル証明書に署名するために対応する秘密鍵を使用する。このように、２次ソフトウェア認証局モジュール１７１０は、内部スマート・キー・デバイス１７０４に対する従属認証局として動作し、これは、内部スマート・キー・デバイス１７０４が

根付いている証明書チェーン内に反映されるであろう。他の例では、内部スマート・キー・デバイス１７０４は従属ソフトウェア認証局モジュールに署名することができ、そのモジュール自体は他の従属ソフトウェア認証局モジュールに署名することができる。

【００８４】

内部スマート・キー・デバイス１７０２はエンティティ１７１２～１７１８にデジタル証明書を発行しており、２次ソフトウェア認証局１７１０はエンティティ１７２２～１７２８にデジタル証明書を発行しており、それにより、証明書発行者（certificate issuer）と証明書被発行者（certificate issuee）との間の信頼関係を確立し、エンティティ１７１２～１７１８およびエンティティ１７２２～１７２８はアプリケーションまたは何らかのその他のタイプのプログラマチック・エンティティにすることができる。加えて、２次ソフトウェア認証局１７１０はエンティティ１７１６にデジタル証明書を発行しており、それにより、この２つのエンティティ間の信頼関係を確立する。

【００８５】

図２３は、コンピューティング・リソースのすべてが相互に認証するための証明書処理機能を含む可能性のある信頼モデルを表しており、これらのコンピューティング・リソースは証明書処理機能を含むように構成する必要がある。たとえば、図２３の種々のエンティティがソフトウェア・アプリケーションを表している場合、これらのソフトウェア・アプリケーションは、固有の公開鍵証明書が提供されており、対応する固有の秘密鍵を持つモジュールを含む必要がある。

【００８６】

たとえば、他のリソースとの認証動作を実行するための能力を必要とするように独立して動作するはずの各コンピューティング・リソースは、たとえば、アプリケーション５４０がＳＷＳＫＵ５３８を含む図６に図示されているように、組み込みソフトウェア・スマート・キー・ユニットを有する可能性がある。アプリケーション５４０は、ＳＷＳＫＵ秘密鍵５５０を含むＳＷＳＫＵ５３８を含み、ＳＷＳＫＵ公開鍵証明書５５４は、非対称暗号鍵ペアとしてＳＷＳＫＵ秘密鍵５５０に対応するＳＷＳＫＵ公開鍵５５６のコピーを含む。また、ＳＷＳＫＵ５３８は、ＩＮＳＫＤ＿ＳＷ公開鍵証明書５５８のコピーも含む。このため、アプリケーション５４０は、ＩＮＳＫＤ５２２に根付いている信頼階層の一部である。ＳＷＳＫＵ５３８内に組み込まれている情報およびＳＷＳＫＵ５３８の機能上の能力を使用して、アプリケーション５４０は、同じくＩＮＳＫＤ５２２を信頼する任意の他のコンピューティング・リソースを認証することができる。したがって、コンピューティング・リソースのすべてが本発明により相互に認証するための証明書処理機能を含む可能性のある信頼モデルを実現するために、システム管理者は、各コンピューティング・リソースがデータ処理デバイスである場合にそのコンピューティング・リソースが内部スマート・キー・デバイスを有するか、または各コンピューティング・リソースがプログラマチック・エンティティである場合にそのコンピューティング・リソースがソフトウェア・スマート・キー・ユニットを有することを保証する必要がある。

【００８７】

しかし、図６に図示されている例では、ＳＷＳＫＵ５３８は、何らかの方法でアプリケーション５４０に組み込まれるようになっていた。後述するように、様々なプロセスを使用して、プログラマチック・リソースのそれぞれに必要な機能を組み込むことができる。

【００８８】

次に図２４に関連して説明すると、このブロック図は、本発明の一実施形態により、オペレーティング・システム内の各プログラマチック・エンティティが内部スマート・キー・デバイスに基づいて信頼階層内に信頼関係を確立するための機能を含む、オペレーティング・システム・ファイルを生成するためのデータ処理システムを描写している。図２４は図５と同様のものであり、システム・ユニット１８０２は外部スマート・キー・デバイス１８０４とのインターフェースを取り、システム・ユニット１８０２は内部スマート・キー・デバイス１８０６も含む。

【００８９】

この例では、オペレーティング・システム・インストール・アプリケーション１８０８は、システム・ユニット１８０２を含むマシン上でのオペレーティング・システム・ファイルのインストールを担当する。インストール手順中に、オペレーティング・システム・インストール・アプリケーション１８０８は、以下により詳細に説明するように、磁気テープまたはＣＤ－ＲＯＭなどの配布媒体からオペレーティング・システム・ファイル１８１２を読み取り、完全に動作可能なモジュール１８１４を生成する。

【００９０】

図２４はオペレーティング・システム・ファイルに関してアクションが実行される一例を描写しているが、代替一実施形態は任意のタイプのアプリケーション・ファイルに適用可能であることに留意されたい。たとえば、オペレーティング・システム・インストール・アプリケーション１８０８は、任意の所与のソフトウェア・アプリケーションに関するインストール・アプリケーションとして記述されるように汎用化することができ、その所与のソフトウェア・アプリケーションは、オペレーティング・システム・ファイル１８１２と同様のものである汎用アプリケーション・ファイルによって表すことができる。インストール・プロセスが完了した後、インストール・アプリケーションは、署名付きオペレーティング・システム・ファイル１８１４と同様のものである証明書所持ソフトウェア・スマート・キー・ユニットを備えたアプリケーション・ファイルを生成している。

【００９１】

図２４は、適切にインストールされたオペレーティング・システム・モジュールのみをシステム・ユニット１８０２上で実行できるようにすべてのオペレーティング・システム・ファイルが保護されるシステムの一例を描写しており、前述の代替実施形態は、システム内のすべてのソフトウェアの実行を制限できるであろう。インストールされた各アプリケーションに適切なインストール・プロセスを使用すると、それぞれのアプリケーション・モジュールを保護することができる。このように、システム・ユニット１８０２は、外部スマート・キー・デバイスの存在によって制御されるプロセスによりシステム上にインストールされたソフトウェア・モジュールのみにソフトウェア実行を制限することができる。本発明のＪａｖａ（登録商標）ベースの実現例では、すべてのＪａｖａ（登録商標）アプリケーションが、インストール・プロセス中にアプリケーション内に置かれるソフトウェア・スマート・キー・ユニットを含む必要がある可能性があり、前述の通り、シールされたＪＡＲファイルからパッケージ内のすべてのコードをロードしなければならないことをクラス・ローダが強制するように、すべてのＪＡＲファイルおよびＪａｖａ（登録商標）パッケージをシールすることができる。

【００９２】

次に図２５に関連して説明すると、この流れ図は、本発明の一実施形態により、オペレーティング・システム・モジュールが相互に認証動作を実行できるようなソフトウェア・スマート・キー・ユニットを含むオペレーティング・システム・モジュールを生成するためのプロセスを描写している。このプロセスはブロック１９０２から始まり、そこで、まだ処理されていない少なくとも１つの追加のオペレーティング・システム・モジュールが存在するかどうかをオペレーティング・システム・インストール・アプリケーションがチェックする。存在しない場合、プロセスが終了する。存在する場合、ブロック１９０４中に、オペレーティング・システム・インストール・アプリケーションは配布媒体からオペレーティング・システム・モジュールを読み取る。たとえば、もう一度、図２４を参照すると、配布媒体上のオペレーティング・システム・モジュールは完全なものではなく、オペレーティング・システム・モジュールは、さらに処理しないとインストールすることができない。オペレーティング・システム・モジュール１８１２は、オペレーティング・システム・ファイルの配布バージョンの形でスタブ・ルーチンまたは空のモジュールを取り込み、これらのオペレーティング・システム・ファイルがインストールされ、追加の変更なしに実行された場合、オペレーティング・システム・サービスは、認証動作を実行できなくなり、それにより、オペレーティング・システムが動作不能になるであろう。

【００９３】

このため、オペレーティング・システム・インストール・アプリケーションが、磁気テープまたはＣＤ－ＲＯＭなどの配布媒体からオペレーティング・システム・モジュール１８１２を読み取った後、ブロック１９０６中に、オペレーティング・システム・インストール・アプリケーションは、現在処理中のオペレーティング・システム・モジュールからスタブ・ルーチンまたは空のモジュールを削除する。ブロック１９０８中に、オペレーティング・システム・インストール・アプリケーションは、非対称暗号鍵ペアを生成し、次にブロック１９１０中に、現在処理中のオペレーティング・システム・モジュールに代わって新たに生成された鍵ペアに基づいてデジタル証明書を発行するよう、ローカル・システム・ユニット上の内部スマート・キー・デバイスに要求する。このように、オペレーティング・システム・インストール・アプリケーションのＳＷＳＫＵは、それに代わってデジタル証明書が要求され発行されているエンティティを偽装するが、代わって、オペレーティング・システム・インストール・アプリケーション内のソフトウェア認証局機能はデジタル証明書を発行することができ、それにより、それに代わってデジタル証明書が要求され発行されているエンティティの証明書チェーンの一部になるために内部スマート・キー・デバイスの公開鍵証明書とともにソフトウェア認証局の公開鍵証明書を必要とする。オペレーティング・システム・インストール動作は、外部スマート・キー・デバイスを所有するシステム管理者によって制御されるものと想定することができ、オペレーティング・システム・インストール手順中に外部スマート・キー・デバイスをシステム・ユニットに連結することにより、システム管理者は内部スマート・キー・デバイスがデジタル証明書を発行できるようにし、それにより、インストール手順が悪意のあるユーザによって何らかの方法でスプーフされるのを防止する。また、各オペレーティング・システム・モジュールが、固有のＩＤをデジタル証明書に取り込めるようにオペレーティング・システム・モジュールのすべてを包含するネームスペース内に固有のＩＤを有するものと想定することもできる。

【００９４】

次にブロック１９１２中にオペレーティング・システム・インストール・アプリケーションはソフトウェア・スマート・キー・ユニットのインスタンスを生成する。新たに生成されたＳＷＳＫＵは、新しいＳＷＳＫＵに代わってオペレーティング・システム・インストール・アプリケーションによって生成された固有の秘密鍵を取り込む。また、この新しいＳＷＳＫＵは、ローカルＩＮＳＫＤによって発行された秘密鍵に対応する公開鍵証明書も取り込み、加えて、新しいＳＷＳＫＵに関するデジタル証明書チェーンの一部を形成する任意の他の公開鍵証明書も含むことができる。証明書チェーンは、信頼階層による信頼パス（trust path）を表している。公開鍵証明書は一般に自由に与えられ、自由に入手可能であるが、証明書チェーンの構築はコンピュータ使用上、高価なものになる可能性があり、したがって、新しいＳＷＳＫＵがその証明書チェーンを表すために必要とする可能性のある任意のデジタル証明書を含めると、実行されたときに、新しいＳＷＳＫＵが認証動作中にその証明書チェーンを迅速に提示することができ、それにより、認証動作がより効率的なものになる。

【００９５】

次にブロック１９１４中にオペレーティング・システム・インストール・アプリケーションは、現在処理中のオペレーティング・システム・モジュール内に、すなわち、除去されたスタブおよび空のモジュールの代わりに新しいＳＷＳＫＵを組み込むことにより、図２４のモジュール１８１４のうちの１つなどの完全に動作可能なモジュールを生成する。次にプロセスは、任意の未処理オペレーティング・システム・モジュールが存在するかどうかをチェックするためにブロック１９０２にループバックし、存在しない場合、プロセスが終了する。オペレーティング・システム・モジュールが処理されるにつれて、新たに生成されたＳＷＳＫＵモジュールは、必要に応じて、変更されたオペレーティング・システム・モジュール内に取り込まれる。配備されたオペレーティング・システム・モジュールまたは新たに組み込まれたＳＷＳＫＵモジュールあるいはその両方には、その認証性を示すために、ＳＷＳＫＵ１８１０によりデジタル署名を行うこともできる。

【００９６】
このように、すべてのオペレーティング・システム・ファイルは、信頼関係を実現するための組み込み機能とともに認証動作を実行できるようになる。オペレーティング・システム・インストール手順中に、ＩＮＳＫＤ１８０６がデジタル証明書を発行するための認証局として動作するか、代わって、オペレーティング・システム・インストール・アプリケーション１８０８がモジュール１８１４のためのデジタル証明書を発行するための認証局として動作し、それぞれの証明書チェーンでは、モジュール１８１４内の各モジュールは、それ専用の秘密鍵および対応する公開鍵証明書と、ＩＮＳＫＤ１８０６の公開鍵証明書と、それが認証局として動作していたために必要であれば、オペレーティング・システム・インストール・アプリケーション１８０８の公開鍵証明書とを有する。したがって、各モジュールは、ＩＮＳＫＤ１８０６に基づく信頼階層を表明する証明書チェーンを有する。ランタイム環境では、モジュール１８１４内の第１のモジュールがモジュール１８１４内の第２のモジュールに対して認証しようと試みる場合、第１のモジュールは適切な所有証明、たとえば、対応する秘密鍵を使用して署名されたデジタル署名とともに、その証明書チェーンを第２のモジュールに提示することになり、第２のモジュールは第１のモジュールの証明書チェーンが基づいているＩＮＳＫＤ１８０６を信頼しているので、第２のモジュールは第１のモジュールを認証し信頼することになる。モジュール１８１４内の各モジュールはＩＮＳＫＤ１８０６を信頼しており、ＩＮＳＫＤ１８０６に関連する証明書チェーンを提示することができるので、各モジュールは残りの同様のモジュールを信頼することができ、それにより、図２３に関して説明した信頼モデルを実現する。

【００９７】
次に図２６に関連して説明すると、このブロック図は、本発明の一実施形態により、各プログラマチック・エンティティが内部スマート・キー・デバイスに基づいて信頼階層内に信頼関係を確立するための機能を含む、プロジェクト・コードを生成するためのデータ処理システムを描写している。図２６は図５と同様のものであり、システム・ユニット２００２は外部スマート・キー・デバイス２００４とのインターフェースを取り、システム・ユニット２００２は内部スマート・キー・デバイス２００６も含む。

【００９８】
この例では、ソフトウェア構成管理（ＳＣＭ：software configuration management）アプリケーション２００８は、ソフトウェア・アプリケーションが作成される特定のプロジェクトに関するすべてのコード・モジュールおよびその他のタイプのファイルの管理を担当する。プロジェクト・ファイルがソフトウェア・エンジニアによって作成されると、プロジェクト・ファイルはＳＣＭシステム内にチェックインされ、そのＳＣＭシステムは、矛盾報告（discrepancy report）およびプロジェクト・タイムライン（project timeline）によりソース・コードのバージョンを追跡することができる。認証上の考慮事項の完全実現を顧慮せずにプロジェクト・モジュールの予備バージョンをテストし統合することができるように、エンジニアはプロジェクト・モジュールにスタブ・ルーチンまたは空のモジュールを取り込む。

【００９９】
しかし、顧客に配布可能であるかまたはその他の方法で実稼働環境内に配備可能な、いわゆる実稼働レベルのアプリケーションを生成する必要性が発生すると、ＳＣＭシステムは、スタブおよびからのモジュールを除去し、ソフトウェア・モジュールそのものである組み込みソフトウェア・スマート・キー・ユニットでそれらを置き換える。このため、最終コンパイルおよびリンク動作が行われる何らかの時点で、ＳＣＭアプリケーション２００８内のＳＷＳＫＵ２０１０は、新たに生成された鍵ペアおよび対応するデジタル証明書を含むＳＷＳＫＵモジュールとともに、非対称鍵ペアを生成する。プロジェクト・モジュール２０１２が処理されると、新たに生成されたＳＷＳＫＵモジュールは、必要に応じて、プロジェクト・モジュール２０１４内にリンクされる。実稼働レベル・プロジェクト・モジュール２０１４または新たに組み込まれたＳＷＳＫＵモジュールあるいはその両方には、それぞれの認証性を示すために、ＳＷＳＫＵ２０１０によってデジタル署名を行うこ

ともできる。

【０１００】

このように、認証動作を完了する能力を必要とするプロジェクト・アプリケーション内の各コンピューティング・リソースには、認証動作を実行できるソフトウェア・スマート・キー・ユニットを提供することができる。しかし、図２６内に例示されているシナリオは、図２４内に例示されているシナリオとは著しく異なっている。図２４では、オペレーティング・システム・モジュール１８１４は、システム・ユニット１８０２上のオペレーティング・システム・インストール・アプリケーション１８０８によって変更される。好ましい一実施形態では、変更されたオペレーティング・システム・モジュール１８０８内のＳＷＳＫＵに対して発行されたデジタル証明書は、システム・ユニット１８０２上のＩＮＳＫＤ１８０６によって署名されている。

【０１０１】

このため、変更されたオペレーティング・システム・モジュールがランタイム環境で実行される場合、変更されたオペレーティング・システム・モジュールに関するデジタル証明書を発行した認証局はランタイム環境の一部である。これは、図２６に提示されているシナリオのケースではない。変更されたプロジェクト・モジュールがランタイム環境で実行される場合、変更されたプロジェクト・モジュールのＳＷＳＫＵ内に組み込まれたデジタル証明書は、プロジェクト・アプリケーションの実稼働バージョンが作成されたシステム・ユニットの内部スマート・キー・デバイスによって署名されている。換言すれば、変更されたプロジェクト・モジュールのＳＷＳＫＵに対してデジタル証明書を発行した認証局はランタイム環境の一部ではない。変更されたプロジェクト・モジュールが他の変更されたプロジェクト・モジュールとの認証動作を完了しようと試みる場合、変更されたプロジェクト・モジュールのそれぞれはプロジェクト・アプリケーションの実稼働バージョンが作成されたシステム・ユニットの内部スマート・キー・デバイスを信頼しているので、認証動作を完了することができる。しかし、変更されたプロジェクト・モジュールがオペレーティング・システム・モジュール、たとえば、オペレーティング・システム・モジュール１８１４のうちの１つとの認証動作を完了しようと試みる場合、オペレーティング・システム・モジュールは、オペレーティング・システム・モジュールのデジタル証明書に関する認証局として動作した内部スマート・キー・デバイスを信頼していないので、認証動作は失敗に終わる。したがって、ランタイム環境で信頼関係を拡張するためのメカニズムが必要である。

【０１０２】

次に図２７に関連して説明すると、この流れ図は、本発明の一実施形態により、内部スマート・キー・デバイスに関する証明書チェーンを拡張するためのプロセスを描写している。上記の通り、ランタイム環境内で実行される何らかのモジュールは、ランタイム環境内に存在する内部スマート・キー・デバイスに基づく信頼関係を確立するための機能を有することができ、内部スマート・キー・デバイスはこれらのモジュールに関する認証局として動作しているので、これらのモジュールは、内部スマート・キー・デバイスが信頼階層のルートに位置するために容易に検証可能なデジタル証明書チェーンを提示することができる。本発明の内部スマート・キー・デバイスをサポートするランタイム環境内にアプリケーションがインストールされる場合、アプリケーション・モジュールは、アプリケーション・モジュール間の信頼関係を確立するための機能を有する可能性があるが、ルート認証局が異なるのでランタイム環境内の他のモジュールとの信頼関係を確立する能力は欠けている可能性があり、残りのモジュールはアプリケーション・モジュールによって提示されるデジタル証明書を信頼する能力を備えていない。

【０１０３】

図２７に関して以下に説明するプロセスは、アプリケーション・モジュールが信頼すべきものとしてそれらを確立できるようにするためのメカニズムを提供する。このプロセスは好ましくは、アプリケーション・モジュールが内部スマート・キー・デバイスを含むランタイム環境内でインストールされるときに実行されるが、ランタイム環境は、アプリケ

ーション・モジュールがランタイム環境内で実行される前にいつでも変更することができる。しかし、この例では、アプリケーション・モジュールを変更する必要はない。したがって、以下に記載するプロセスは、オペレーティング・システム・モジュールの変更が必要であった図２５に関して説明したプロセスとは異なっている。

【０１０４】

このプロセスはブロック２１０２から始まり、そこで、内部スマート・キー・デバイスがインストール・アプリケーション内のソフトウェア・スマート・キー・ユニットから、または何らかの他の形式の管理ユーティリティ・アプリケーションから要求メッセージを受信し、その要求メッセージはフォーリン（foreign）内部スマート・キー・デバイスの、すなわち、ローカル・ランタイム環境の外側のルート・デジタル証明書を表明するための要求を示す。たとえば、管理ユーティリティ・アプリケーションは、すでにインストールされたかまたはローカル・ランタイム環境内でインストール中のアプリケーション・モジュールの実稼働バージョンに付随する構成ファイルにアクセスできる。これらの構成ファイルは、たとえば、図２６に関して記載されたものと同様に、アプリケーション・モジュール内に組み込まれたソフトウェア・スマート・キー・ユニットに関するデジタル証明書を生成するためにフォーリン内部スマート・キー・デバイスによって使用されたデジタル証明書のコピーを含む。換言すれば、構成ファイルには、インストール中のアプリケーションを生産したベンダのランタイム環境のフォーリン内部スマート・キー・デバイスによって使用された公開鍵証明書のコピーが付随する可能性がある。フォーリン内部スマート・キー・デバイスのデジタル証明書を表明するための要求は、現行ランタイム環境の内部スマート・キー・デバイスが共通の信頼できるエンティティについてチェックする能力なしで行われ、各内部スマート・キー・デバイスはそれ専用の信頼階層内のルートの信頼できるエンティティとして動作するので、現行ランタイム環境の内部スマート・キー・デバイスとフォーリン内部スマート・キー・デバイスのための信頼の基礎を置くことができる共通の信頼できるエンティティが他にまったく存在しない。このため、デジタル証明書を表明するプロセスは、タスクを完了するための信頼性を提供する安全な手順でなければならない。

【０１０５】

フォーリン内部スマート・キー・デバイスのデジタル証明書を表明するための動作の信頼性を保証するために、ブロック２１０４中に、要求側アプリケーションのソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスによって認証されているかどうかに関する判断が行われ、この判断は、たとえば、図１３に関して上述した通り、前の認証手順中に内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットに渡したセッション・キーを使用して、受信メッセージの内容を正常に暗号化解除することによって実行することができる。ソフトウェア・スマート・キー・ユニットが認証されていない場合、内部スマート・キー・デバイスは、ブロック２１０６中に適切なエラー応答を生成し、ブロック２１０８中に応答メッセージを要求側ソフトウェア・スマート・キー・ユニットに返し、それにより、プロセスを終了する。

【０１０６】

ソフトウェア・スマート・キー・ユニットが認証されている場合、ブロック２１１０中に、内部スマート・キー・デバイスは、外部スマート・キー・デバイスが依然としてシステム・ユニットに電気的に連結されているかどうかを判断する。このように、手順全体は、手順を実行する特権を有するシステム管理者の制御下にあると判断される。外部スマート・キー・デバイスがシステム・ユニットに電気的に連結されていない場合、内部スマート・キー・デバイスは、ブロック２１０６中にエラー応答を生成し、ブロック２１０８中に応答メッセージをソフトウェア・スマート・キー・ユニットに返し、それにより、プロセスを終了する。

【０１０７】

ソフトウェア・スマート・キー・ユニットが認証されており、外部スマート・キー・デバイスが依然としてシステム・ユニットに電気的に連結されている場合、内部スマート・

キー・デバイスはソフトウェア・スマート・キー・ユニットのために要求された機能を実行する。ブロック２１１２中に内部スマート・キー・デバイスは、信頼できるルート証明書のテーブルまたはリストにフォーリン内部スマート・キー・デバイスの表明されたルート証明書を追加し、そのテーブルまたはリストはおそらく前に表明された複数の証明書を含む。ブロック２１１４中に適切な応答メッセージが作成された後、ブロック２１０８で応答メッセージがソフトウェア・スマート・キー・ユニットに返され、プロセスが終了する。

【０１０８】

次に図２８に関連して説明すると、このブロック図は、本発明の一実施形態により、フォーリン内部スマート・キー・デバイスに関する複数のルート証明書を含む証明書チェーンを維持する単一のローカル内部スマート・キー・デバイスによって提供される信頼に基づく信頼関係から構築される信頼モデルの一例を描写している。図６およびその他の図に関して説明した通り、内部スマート・キー・デバイスは少なくとも１つの秘密鍵とそれに対応する公開鍵証明書とを所有し、同様に、図２８は、デジタル証明書２２０４を含む内部スマート・キー・デバイス２２０２を図示している。図２７に関して説明した通り、システム管理者が特定のランタイム環境の信頼階層に対して追加のルート証明書を表明することが必要である場合があり、図２８は、デジタル証明書２２０６および２２０８が前に表明されており、その時点でその信頼できる証明書チェーンの一部として内部スマート・キー・デバイス２２０２内に保管されていることを図示している。

【０１０９】

上記の通り、本発明の内部スマート・キー・デバイスをサポートするランタイム環境内にアプリケーション・モジュールがインストールされる場合、アプリケーション・モジュールは、アプリケーション・モジュール間の信頼関係を確立するための機能が提供されている可能性があるが、ルート認証局が異なるのでランタイム環境内の他のモジュールとの信頼関係を確立する能力は欠けている可能性がある。アプリケーション・モジュールは、異なる信頼階層内に存在する残りのモジュールとともに１つの信頼階層内に存在するものと見なすことができる。

【０１１０】

この問題を克服するため、図２７に関して記載したプロセスは、単一ランタイム環境内に複数の信頼階層を導入するためのメカニズムを例示している。この解決策については、図２８に関してさらに例示する。デジタル証明書２２０６および２２０８を受け入れることにより、内部スマート・キー・デバイス２２０２は、受け入れられたデジタル証明書に関連するフォーリン内部スマート・キー・デバイスとの信頼関係２２１０および２２１２を暗黙のうちに形成する。このように、内部スマート・キー・デバイス２２０２は、ルート証明書２２０４、２２０６、および２２０８により、信頼階層２２１４、２２１６、および２２１８をそれぞれサポートする。ルート証明書２２０６および２２０８によって表されるフォーリン内部スマート・キー・デバイスによって署名されたアプリケーション・モジュールのデジタル証明書を妥当性検査するためにルート証明書２２０６および２２０８が使用可能である場合、ランタイム環境内の他のモジュールは、信頼階層同士に橋を架ける信頼関係２２２０および２２２２を形成することができる。

【０１１１】

次に図２９に関連して説明すると、この流れ図は、ローカル内部スマート・キー・デバイスによって維持される現行ルート証明書チェーンを入手するためのプロセスを描写している。図２７は、ルート証明書をローカル・スマート・キー・デバイス内に保管することにより、システム管理者が特定のランタイム環境の信頼階層内にルート証明書を表明するためのプロセスを描写しており、図２９は、ローカル内部スマート・キー・デバイスから現行ルート証明書チェーンを入手するためのプロセスを例示している。このプロセスはブロック２３０２から始まり、そこで、内部スマート・キー・デバイスがソフトウェア・スマート・キー・ユニットから要求メッセージを受信し、それにより、ローカル内部スマート・キー・デバイスによって維持されている現行ルート証明書チェーンを要求する。次に

ローカル内部スマート・キー・デバイスは、ブロック２３０４中に、現行ルート証明書チェーンを含む応答メッセージを要求側ソフトウェア・スマート・キー・ユニットに返し、プロセスが終了する。ローカル内部スマート・キー・デバイスは、要求側ソフトウェア・スマート・キー・ユニットが前にローカル内部スマート・キー・デバイスに対して認証したことを要求する可能性がある。システム管理者が外部スマート・キー・デバイスを使用して動作を使用可能にする場合にのみ実行される内部スマート・キー・デバイス内の動作を例示している図１４～図１５または図２７とは対照的に、図２９に例示されているプロセスは外部スマート・キー・デバイスによる使用可能化を必要としない。

【０１１２】

次に図３０に関連して説明すると、この流れ図は、フォーリン内部スマート・キー・デバイスからのデジタル証明書が信頼すべきものであるかどうかを判断するためのプロセスを描写している。何らかの時点で、あるモジュールは、ランタイム環境内の他のモジュールによって制御されるコンピューティング・リソースへのアクセスを要求する。この２つのモジュールが前に相互認証動作を完了していないと想定すると、２つのモジュールは、たとえば、図１０～図１１に関して説明した相互認証動作と同様の相互認証動作を完了しようと試みる。この例では、所望のコンピューティング・リソースを制御しているモジュールがローカル内部スマート・キー・デバイスに基づくローカル信頼階層内に含まれ、要求側モジュールがフォーリン内部スマート・キー・デバイスに基づく信頼階層内に含まれるものと想定することができるが、フォーリン内部スマート・キー・デバイスに関するルート証明書は前にローカル・スマート・キー・デバイス内に表明されている。

【０１１３】

このプロセスはブロック２４０２から始まり、そこで、制御モジュールと要求側モジュールが認証動作を開始している。次に制御モジュールはブロック２４０４中に、要求側モジュールのデジタル証明書を入手するが、要求側モジュールから直接入手する可能性が最も高く、デジタル証明書からの公開鍵を使用して、要求側モジュールが公開鍵に対応する秘密鍵を所有しているかどうかを判断するが、これらのアクションは図３０には図示されていない。

【０１１４】

要求側モジュールのデジタル証明書上のデジタル署名の認証性を判断するために、制御モジュールはフォーリン内部スマート・キー・デバイスのデジタル証明書の信頼すべきコピーを要求し、それにより、デジタル署名を生成するために使用された秘密鍵に対応する公開鍵のコピーを提供する。要求側モジュールは要求側モジュールのデジタル証明書を発行したフォーリン内部スマート・キー・デバイスに関するデジタル証明書のコピーを所有し、それにより、要求側モジュールがフォーリン内部スマート・キー・デバイスのデジタル証明書のコピーを制御モジュールに提供できるようにしなければならないが、制御モジュールは、フォーリン内部スマート・キー・デバイスのデジタル証明書のコピーを入手するために独立した信頼すべき方法を必要とする。フォーリン内部スマート・キー・デバイスのデジタル証明書のコピーを入手しようとして、制御モジュールはブロック２４０６中に、ローカル内部スマート・キー・デバイスによって現在維持されているルート証明書チェーンを入手する。

【０１１５】

次に制御モジュールはブロック２４０８中に、フォーリン内部スマート・キー・デバイスに関するルート証明書が検索されたルート証明書チェーン内にあることを検証する。前述の通り、図３０に図示されている例では、フォーリン内部スマート・キー・デバイスに関するルート証明書が前にローカル・スマート・キー・デバイス内に表明されているものと想定することができる。このため、ブロック２４０６の結果、フォーリン内部スマート・キー・デバイスのデジタル証明書のコピーを含むルート証明書チェーンが返される。

【０１１６】

次に制御モジュールはブロック２４１０で、要求側モジュールのデジタル証明書上のデジタル署名を検証することにより、要求側モジュールのデジタル証明書の認証性を検証し

、プロセスが終了する。デジタル署名が検証されたものと想定して、制御モジュールは認証動作を続行することができる。

【０１１７】

図３１および図３２に関して以下に本発明の他の実施形態を示すが、この実現例の例は、前に記載されている本発明の様々な態様に依存するものである。上述の通り、内部スマート・キー・デバイスなどのデータ処理システム内のハードウェア・セキュリティ・ユニットは認証局として機能することができる。図２３に関して記載した通り、内部スマート・キー・デバイスの認証局機能は、データ処理システム内のコンピューティング・リソースが信頼関係階層内のエンティティである信頼モデルのルートとして見ることができる。信頼関係階層は、図２３のように、内部スマート・キー・デバイスが逆ピラミッドの頂点にある逆ピラミッドによって表すことができ、コンピューティング・リソースがその逆ピラミッドを形成する。図２４～図２６に関して記載した通り、ハードウェア・セキュリティ・ユニットの認証局機能は、ソフトウェア暗号モジュール、すなわち、ソフトウェア・セキュリティ・ユニットまたはソフトウェア・スマート・キー・ユニットに署名するために、ならびに、ソフトウェア暗号モジュールに対してデジタル証明書を発行するために使用することができる。簡単に前述した通り、コードの改ざんを防止するために、ソフトウェア暗号モジュールのソフトウェア・パッケージをシールすることができる。

【０１１８】

次に図３１に関連して説明すると、このデータフロー・ダイアグラムは、本発明の一実現例により、ソフトウェア・モジュールの保全性を確保するために使用可能なハードウェア支援信頼モデルを実現するデータ処理システム内のエンティティを例示している。図３１について説明する前に、Ｊａｖａ（登録商標）ランタイム環境内の具体的な例について説明する。コードの改ざんを防止するために、何らかの形のソフトウェア暗号ユニットを含むＪａｖａ（登録商標）アプリケーションのクラス・ファイルをシールした後、クラス・ローダによってプログラムの保全性が強制される。クラス・ローダを信頼できることを保証するために、クラス・ローダは署名され、シールも行われる必要がある。クラス・ローダの保全性を保証するために、クラス・ローダをロードするローダ、すなわち、オペレーティング・システム・プログラム・ローダは、何らかの方法で署名されシールされる必要がある。オペレーティング・システム・プログラム・ローダの保全性を保証するために、オペレーティング・システム・プログラム・ローダをロードするローダ、すなわち、データ処理システムのＲＯＭ内のブート・ローダは、署名されシールされる必要がある。

【０１１９】

より汎用的な非Ｊａｖａ（登録商標）環境に関して説明すると、コードの改ざんを防止するためにソフトウェア暗号モジュールのソフトウェア・パッケージがシールされた後、オペレーティング・システム・プログラム・ローダによってプログラムの保全性が強制される。オペレーティング・システム・プログラム・ローダを信頼できることを保証するために、オペレーティング・システム・プログラム・ローダは署名され、シールも行われる必要がある。オペレーティング・システム・プログラム・ローダの保全性を保証するために、オペレーティング・システム・プログラム・ローダをロードするローダ、すなわち、システムＲＯＭ内のブート・ローダは、署名され、シールも行われる必要がある。これらの要件および動作は図３１に反映されている。

【０１２０】

ブートＲＯＭ２５０２は内部スマート・キー・デバイス２５０４の秘密鍵によって署名されており、これは、製造プロセス中、フラッシュ・メモリ更新を使用してブートＲＯＭが構成されるサイト固有のインストール手順中、または何らかの他の方法で、行われる可能性がある。その後、ブートＲＯＭ２５０２は内部スマート・キー・デバイス２５０４との相互認証手順を実行することができ、それにより、ブートＲＯＭ２５０２と内部スマート・キー・デバイス２５０４との間の信頼関係を作成する。

【０１２１】

また、オペレーティング・システム・プログラム・ローダ２５０６も内部スマート・キ

ー・デバイス２５０４の秘密鍵によって署名されており、これは、図２４および図２５に関して記載されているプロセスにより行われる可能性がある。ブートＲＯＭ２５０２は、内部スマート・キー・デバイス２５０４からの援助によってオペレーティング・システム・プログラム・ローダ２５０６のシールされたプログラム・モジュール（複数も可）上の署名を妥当性検査することにより、オペレーティング・システム・プログラム・ローダ２５０６の保全性を保証することができ、この内部スマート・キー・デバイス２５０４は、相互認証手順の完了によりブートＲＯＭ２５０２との信頼関係をすでに確立しているのでブートＲＯＭ２５０２を支援する。その後、オペレーティング・システム・プログラム・ローダ２５０６は、内部スマート・キー・デバイス２５０４との相互認証手順を実行することができ、それにより、オペレーティング・システム・プログラム・ローダ２５０６と内部スマート・キー・デバイス２５０４との間の信頼関係を作成する。

【０１２２】

アプリケーション・モジュール２５０８は、内部スマート・キー・デバイス２５０４の秘密鍵によるかまたはルート認証局として動作する内部スマート・キー・デバイス２５０４とともに認証局として動作するオペレーティング・システム内のソフトウェア暗号ユニットによって署名されており、これは、図２６に関して記載されているプロセスにより行われる可能性がある。オペレーティング・システム・プログラム・ローダ２５０６は、内部スマート・キー・デバイス２５０４からの援助によってシールされたアプリケーション・プログラム・モジュール上の署名を妥当性検査することにより、アプリケーション・モジュール２５０８の保全性を保証することができ、この内部スマート・キー・デバイス２５０４は、相互認証手順の完了によりオペレーティング・システム・プログラム・ローダ２５０６との信頼関係をすでに確立しているのでオペレーティング・システム・プログラム・ローダ２５０６を支援する。その後、アプリケーション・モジュール２５０８は、必要に応じて信頼関係を確立するために、内部スマート・キー・デバイス２５０４、オペレーティング・システム・モジュール２５１０、またはその他のアプリケーション・モジュール２５１２との相互認証手順を実行することができる。

【０１２３】

次に図３２に関連して説明すると、この流れ図は、本発明の一実現例により、ソフトウェア・モジュールの保全性を確保するためのプロセスを例示している。このプロセスはブロック２６０２から始まり、そこで、データ処理システムの始動中に、データ処理システム内のハードウェア回路がデータ処理システム内の内部スマート・キー・ユニットの援助によりブートＲＯＭ上のデジタル署名を妥当性検査する。ブートＲＯＭ上のデジタル署名が正常に妥当性検査されたと想定して、ブロック２６０４中にデータ処理システム上の始動ハードウェアがデータ処理システムのブートＲＯＭを起動し、それにより、内部スマート・キー・デバイスがそれを妥当性検査するまでブートＲＯＭが多くのタイプの動作を実行するのを防止するか、または代替実現例では、内部スマート・キー・デバイスがそれを妥当性検査するまでブートＲＯＭが任意の動作を実行するのを防止する。

【０１２４】

その後の何らかの時点で、おそらく依然としてデータ処理システムの始動手順中に、ブートＲＯＭはブロック２６０６中に、データ処理システムをさらに初期設定するために必要な署名／シールされたオペレーティング・システム・モジュール（複数も可）上のデジタル署名（複数も可）を検証する。ブートＲＯＭがオペレーティング・システム・モジュール（複数も可）上のデジタル署名（複数も可）を妥当性検査できると想定して、ブートＲＯＭは、ブロック２６０８中にオペレーティング・システム・モジュール（複数も可）をロードし、ブロック２６１０中に実行制御をオペレーティング・システム・モジュール（複数も可）に渡す。

【０１２５】

その後の何らかの時点で、ブロック２６１２中に、オペレーティング・システム内のプログラム・ローダは、たとえば、データ処理システムのユーザによる要求に応答して、データ処理システム上で呼び出されている署名／シールされたアプリケーション・モジュー

ル（複数も可）上のデジタル署名を検証する。プログラム・ローダがアプリケーション・モジュール（複数も可）上のデジタル署名（複数も可）を妥当性検査できると想定して、プログラム・ローダは、ブロック２６１４中にアプリケーション・モジュール（複数も可）をロードし、ブロック２６１６中に実行制御をアプリケーション・モジュール（複数も可）に渡し、それにより、プロセスを終了する。このように、本発明はデータ処理システム上で実行されるすべてのソフトウェア・モジュールの保全性を確保するために使用することができ、データ処理システム上で実行されるすべてのソフトウェアには、内部スマート・キー・デバイスによるか、または内部スマート・キー・デバイスによって信頼されるソフトウェア認証局モジュールによる署名が行われなければならない。この信頼関係は、ソフトウェア認証局モジュールと内部スマート・キー・デバイスとの間の相互認証により、さらに内部スマート・キー・デバイス内の信頼できる証明書のリスト内にソフトウェア認証局モジュールの証明書を追加するための構成プロセスにより確立される。図３１に関して部分的に記載され、これまでの図に関してより詳細に記載されている通り、それぞれのエンティティ内に前に組み込まれているデジタル証明書を使用する相互認証手順によりソフトウェア実行中に適切な信頼関係が確立される。

【０１２６】

次に図３３に関連して説明すると、このブロック図は、本発明の一実施形態により、データ処理システム内のハードウェア・セキュリティ・ユニット内で暗号機能を使用可能にするために取り外し可能記憶媒体を受け入れるデータ処理システムの一部分を描写している。本発明は、スマート・キー・デバイスとともに、コンパクト・ディスク（ＣＤ：compact disk）、デジタル多目的またはビデオ・ディスク（ＤＶＤ：digital versatile or video disk）、あるいは磁気テープなどであるが、これらに限定されない取り外し可能記憶媒体を使用し、これらはいずれも暗号鍵を保持するものである。この暗号鍵は暗号化機能を使用可能にするために使用される。システム・ユニット２７０２は、システム・ユニット２７０２内に取り付けられ、ＣＤ２７１２を読み取ることができる媒体読取装置、この例ではＣＤ装置２７０４とのインターフェースを取る。上記で説明した通り、現在使用可能であり、まだ開発予定である様々なタイプのＣＤ、ＤＶＤ、および磁気テープ装置のいずれかなどであるが、これらに限定されないその他のタイプの媒体読取装置および対応する媒体を使用することができる。ＣＤ装置２７０４は多くの可能な構成の１つにおいてシステム・ユニット２７０２に結合することができ、これは当業者にとって明白であるはずである。たとえば、ＣＤ読取装置は、システム・ユニット２７０２の外部または内部に取り付けることができる。

【０１２７】

システム・ユニット２７０２は、ホスト・システム２７０２の一体部分である、すなわち、マザーボード（図示せず）上など、システム２７０２内に取り付けられている内部スマート・キー・デバイス（ＩＮＳＫＤ）２７０６を含む。内部スマート・キー・デバイス２７０６は、好ましくは、ホスト・システムから取り外しにくい、パッケージ化された集積回路である。これは、ハードウェア・セキュリティ・ユニットまたはデバイスとして記述される場合もあれば、命令を実行するための処理装置を有する場合もある。この例では、ＣＤ２７１２がＩＮＳＫＤ２７０６の機能を使用可能にするように、ＣＤ２７１２とＩＮＳＫＤ２７０６が対になっている。ＣＤ２７１２は、システム管理担当者、たとえば、ＩＴ管理者によって物理的に固定される。ＩＴ管理者が、ホスト・マシン上の突き合わせデバイス、すなわち、ＩＮＳＫＤ２７０６によってのみ実行可能な特定の暗号機能を使用可能にする必要があるときに、ＣＤ２７１２は、システム・ユニット２７０２などのホスト・マシンに結合されたＣＤ装置２７０４に挿入される。換言すれば、特定の暗号機能は、ＣＤ２７１２がＣＤ装置２７０４に挿入されたときに使用可能になる。ＩＮＳＫＤ２７０６のみが特定の暗号出力を生成するための１つまたは複数の特定の暗号秘密鍵を含んでいるので、ＩＴ管理者が必要とする結果は、ＩＮＳＫＤ２７０６のみによって生成することができる。

【０１２８】

システム・ユニット２７０２上のアプリケーション２７０８は、ＣＤ２７１２およびＩＮＳＫＤ２７０６に類似しているソフトウェア・スマート・キー・ユニット（ＳＷＳＫＵ）２７１０を有する。アプリケーション２７０８はＳＷＳＫＵ２７１０を使用して特定の機能を実行するが、この機能については以下により詳細に説明する。また、システム・ユニット２７０２は、ＣＤ装置２７０４の動作を制御するロジックであるＣＤデバイス・ドライバ２７１４も含む。ＣＤデバイス・ドライバ２７１４の標準的な非暗号機能は当業者が精通しているはずである。ＩＮＳＫＤ２７０６、アプリケーション２７０８、ＳＷＳＫＵ２７１０、およびＣＤデバイス・ドライバ２７１４については図３５に併せて以下により詳細に説明する。

【０１２９】

図３４に関連して説明すると、このブロック図は、内部スマート・キー・デバイス内の暗号機能を使用可能にする取り外し可能記憶媒体を描写している。この例の取り外し可能記憶媒体はＣＤ２７１２（図３３）である。ＣＤ２７１２は、暗号エンジン２８１０、ＣＤ秘密鍵２８１２、内部スマート・キー・デバイス（ＩＮＳＫＤ）公開鍵２８１４、それ自体がＣＤ公開鍵２８１８を含むＣＤ公開鍵証明書２８１６、およびデジタル署名２８２０のための記憶域を含む。暗号エンジン２８１０、ＣＤ秘密鍵２８１２、ＩＮＳＫＤ公開鍵２８１４、ＣＤ公開鍵証明書２８１６、ＣＤ公開鍵２８１８、およびデジタル署名２８２０については図３５に併せて以下により詳細に説明する。ＩＮＳＫＤ公開鍵２８１４は、図３５に併せて後述するＩＮＳＫＤ公開鍵２９３０のコピーである。ＣＤ２７１２などのＣＤは複製することができるので、システムのセキュリティはＣＤ２７１２の安全な保管次第であり、すなわち、ＣＤ２７１２が複製されるのを防止するためにＣＤ２７１２は安全な場所に保管しなければならない。

【０１３０】

一実施形態では、ＣＤ装置２７０４（図３３）は、ＣＤ２７１２から読み取り、ＣＤ２７１２に書き込む。その場合、ＣＤ２７１２上に含まれる情報は、システム・ユニット２７０２およびＣＤ装置２７０４を使用して、システム管理者がＣＤ２７１２に書き込むことができる。代替例では、ＣＤ装置２７０４はＣＤ２７１２を読み取ることしかできない。この場合、ＣＤ２７１２は、ＣＤ２７１２を使用してＩＮＳＫＤ２７０６（図３３）の暗号機能を使用可能にする前に、別個の装置（図示せず）を使用してシステム管理者によって１回、構成される。

【０１３１】

次に図３５に関連して説明すると、このブロック図は、図３３のシステム・ユニット２７０２をより詳細に描写している。図３３に併せて上記で説明した通り、システム・ユニット２７０２は、本発明の一実施形態により、ＩＮＳＫＤ２７０６内の暗号機能を使用可能にするために媒体読取装置またはＣＤ装置２７０４（図３３）とともにＣＤ２７１２（図３３）を使用するＩＮＳＫＤ２７０６（図３３）を含む。図３５は、図３５が様々なコンポーネントおよびそのコンポーネント内に保管される暗号鍵に関する追加の詳細を含むことを除き、図３３と同様のものである。

【０１３２】

図３３に併せて上記で説明した通り、デバイス・ドライバによって制御された媒体読取装置は、ＣＤ、ＤＶＤ、磁気テープ媒体、または任意のその他の外部記憶媒体などの外部記憶媒体を読み取り、おそらくそれに書き込む。この例では、ＣＤ装置２７０４は、ＣＤデバイス・ドライバ２７１４（図３３）の制御下で、ＣＤ２７１２から読み取り、そこに書き込む。ＣＤ２７１２は、システム管理者によって制御され、ハードウェア・セキュリティ・トークンとして動作する。ＣＤ２７１２はＣＤ読取装置２７０４に挿入可能であり、ＣＤ読取装置２７０４はＣＤ２７１２がシステム・ユニット２７０２に結合できるようにする。ＣＤ２７１２およびＣＤ読取装置２７０４は、デジタル情報を表す電気信号を交換するために、ＣＤデバイス・ドライバ２７１４を介してシステム・ユニット２７０２と連結し、通信する。ＣＤ２７１２の論理図については、図３４に併せて上述されている。

【０１３３】

ＩＮＳＫＤ２７０６は、ＩＮＳＫＤ２７０６内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン２９２４を含む。ＩＮＳＫＤ秘密鍵２９２６は、ＩＮＳＫＤ２７０６の外部にあるエンティティによって読み取りまたはアクセスを行えないように保管されている。ＩＮＳＫＤ２７０６は、ＩＮＳＫＤ秘密鍵２９２６のコピーを伝送するかまたはその他の方法で提供するための機能を含んでいない。ＩＮＳＫＤ公開鍵証明書２９２８は、非対称暗号鍵ペアとしてＩＮＳＫＤ秘密鍵２９２６に対応するＩＮＳＫＤ公開鍵２９３０のコピーを含む。また、ＩＮＳＫＤ２７０６は、ＣＤ公開鍵証明書２９３２のコピーも含み、その証明書自体は、ＣＤ公開鍵２９３３のコピーを含む。ＣＤ公開鍵２９３３はＣＤ公開鍵２８１８（図３４）のコピーであり、いずれも非対称暗号鍵ペアとしてＣＤ秘密鍵２８１２（図３４）に対応する。ＣＤ公開鍵証明書２９３２は、その製造または初期設定プロセスの一部としてＩＮＳＫＤ２７０６に書き込むことができる。

【０１３４】

また、ＩＮＳＫＤ２７０６は、ＣＤＤ公開鍵証明書２９３４のコピーも含み、その証明書自体は、ＣＤＤ公開鍵２９３５のコピーを含む。ＣＤＤ公開鍵２９３５はＣＤＤ公開鍵２９８８のコピーであり、これは非対称暗号鍵ペアとしてＣＤＤ秘密鍵２９９６に対応する。ＣＤＤ公開鍵証明書２９３４は、その製造または初期設定プロセスの一部としてＩＮＳＫＤ２７０６に書き込むことができる。

【０１３５】

代替諸実施形態では、ＩＮＳＫＤ秘密鍵２９２６およびＩＮＳＫＤ公開鍵２９３０を複数の機能に使用することができる。図３５に図示されている一実施形態では、ＩＮＳＫＤ秘密鍵２９２６およびＩＮＳＫＤ公開鍵２９３０はＩＮＳＫＤ２７０６とＣＤデバイス・ドライバ２７１４との間ならびにＩＮＳＫＤ２７０６とＣＤ２７１２との間の通信のために予約されており、ＩＮＳＫＤ２７０６は１つまたは複数の他の暗号鍵ペアを他の機能に使用する。この例では、ＩＮＳＫＤ２７０６と、アプリケーション２７０８（図３３）内のソフトウェア・スマート・キー・ユニット（ＳＷＳＫＵ）２７１０（図３３）との間の通信を保護するために、ＩＮＳＫＤ２７０６によってＩＮＳＫＤ＿ＳＷ秘密鍵２９３６が使用される。ＩＮＳＫＤ＿ＳＷ公開鍵証明書２９４２は、非対称暗号鍵ペアとしてＩＮＳＫＤ＿ＳＷ秘密鍵２９３６に対応するＩＮＳＫＤ＿ＳＷ公開鍵２９４４のコピーを含む。この例では単一のＳＷＳＫＵ２７１０を示しているが、複数のＳＷＳＫＵも可能である。その場合、各ＳＷＳＫＵは、ＩＮＳＫＤ＿ＳＷ秘密鍵２９３６に対応する、それ専用の秘密鍵を有することになるであろう。また、ＩＮＳＫＤ２７０６は、ＳＷＳＫＵ公開鍵証明書２９４６のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＳＷＳＫＵ２７１０内のＳＷＳＫＵ秘密鍵２９５０に対応するＳＷＳＫＵ公開鍵２９４８のコピーを含む。

【０１３６】

システム・ユニット２７０２は、ＳＷＳＫＵ２７１０を含むアプリケーション２７０８の実行をサポートし、そのＳＷＳＫＵ自体は、ソフトウェア・スマート・キー・ユニット２７１０内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン２９５２を含む。ＳＷＳＫＵ公開鍵証明書２９５４は、非対称暗号鍵ペアとしてＳＷＳＫＵ秘密鍵２９５０に対応するＳＷＳＫＵ公開鍵２９５６のコピーを含む。また、ＳＷＳＫＵ２７１０は、ＩＮＳＫＤ＿ＳＷ公開鍵証明書２９５８のコピーも含み、その証明書自体は、非対称暗号鍵ペアとしてＩＮＳＫＤ＿ＳＷ秘密鍵２９３６に対応するＩＮＳＫＤ＿ＳＷ公開鍵２９４４のコピーを含む。以下により詳細に説明する通り、ＳＷＳＫＵ２７１０にはデジタル署名を行うことができる。図３５に図示されている例では、ＳＷＳＫＵ２７１０は、ＩＮＳＫＤ＿ＳＷ秘密鍵２９３６を使用してＳＷＳＫＵ２７１０について計算されたデジタル署名２９６２を含む。換言すれば、ＩＮＳＫＤ２７０６は、ＩＮＳＫＤ＿ＳＷ秘密鍵２９３６を使用してＳＷＳＫＵ２７１０にデジタル署名を行う。前述の通り、典型的には、アプリケーションの各インスタンスごとに異なるＩＮＳＫＤ＿ＳＷ秘密鍵２９３６が存在することになり、種々のアプリケーションが種々のＩＮＳＫＤ＿ＳＷ

を有することになるであろう。

【０１３７】

ＣＤデバイス・ドライバ２７１４（図３３）はＣＤドライバ・スマート・キー・ユニット（ＣＤＤＳＫＵ：CD driver smart key unit）２９８２を含む。ＣＤＤＳＫＵ２９８２は、ＣＤ２７１２およびＣＤＤＳＫＵ２９８２内に保管されている様々なデータ項目を使用して暗号機能を実行するための暗号エンジン２９８４を含む。この代替例では、暗号エンジン２９２４を使用するか、あるいはＣＤ２７１２上に保管されている暗号エンジン２８１０（図３４）をダウンロードして使用することができる。ＣＤ２７１２、ＣＤデバイス・ドライバ２７１４、およびそこに保管されている鍵は読み取ることができるが、鍵は、本明細書の複数の例で上述した通り、ＳＷＳＫＵ２７１０を保護するために使用されるプロセスと同様に、ＩＮＳＫＤ２７０６の署名および検証プロセスにより保護される。ＣＤＤＳＫＵ公開鍵証明書２９８６は、非対称暗号鍵ペアとしてＣＤＤ秘密鍵２９９６に対応するＣＤＤ公開鍵２９８８を使用する。

【０１３８】

ＣＤＤＳＫＵ２９８２は、ＩＮＳＫＤ秘密鍵２９２６を使用してＣＤＤＳＫＵ２９８２について計算されたデジタル署名２９９０を含む。換言すれば、ＩＮＳＫＤ２７０６は、ＩＮＳＫＤ秘密鍵２９２６およびＣＤＤ秘密鍵２９９６を使用してＣＤＤＳＫＵ２９８２にデジタル署名を行う。このプロセスはＣＤデバイス・ドライバ２７１４を認証するものである。

【０１３９】

次に図３６に関連して説明すると、この流れ図は、ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を描写している。このプロセスはブロック３００２から始まり、そこで、ＣＤまたはその他の取り外し可能記憶媒体が互換性のある媒体読取装置に連結される。ブロック３００４中に、媒体読取装置は内部スマート・キー・デバイスを含むシステム・ユニットに電気的に連結され、ＣＤまたはその他の取り外し可能記憶媒体は媒体読取装置に連結される。たとえば、ＩＴ管理者は、ＣＤ装置２７０４（図３３および図３５）内にＣＤ２７１２（図３３および図３４）を挿入し、それにより、ＣＤ２７０４をシステム・ユニット２７０２（図３３および図３５）およびＩＮＳＫＤ２７０６（図３３および図３５）などの互換性のある装置に結合することができる。ブロック３００６中に、ＩＮＳＫＤ２７０６およびＣＤ２７１２ならびにＩＮＳＫＤ２７０６およびＣＤデバイス・ドライバ２７１４（図３３および図３５）は相互認証手順を実行する。この相互認証手順は、ＩＮＳＫＤ秘密鍵２９２６に基づいてＩＮＳＫＤ２７０６がデジタル署名２９９０（図３５）を妥当性検査することと、同じくＩＮＳＫＤ秘密鍵２９２６に基づいてＩＮＳＫＤ２７０６がデジタル署名２８２０（図３４）を妥当性検査することを含む。この代替例では、ＩＮＳＫＤ２７０６は、ＣＤ２７１２およびＣＤデバイス・ドライバ２７１４の認証のために異なる秘密鍵を使用できるであろう。この妥当性検査は、ＣＤ２７１２とＣＤデバイス・ドライバ２７１４の両方の認証性を保証するものである。次にブロック３００８中に、ＩＮＳＫＤ２７０６が暗号化を実行できるようになり、プロセスが終了する。

【０１４０】

相互認証手順にエラーがあれば、その結果、ＩＮＳＫＤ２７０６が相互認証プロセスに失敗した装置またはソフトウェアにデジタル署名を行うのを防止するものと想定することができる。換言すれば、ＣＤ２７１２がなければ、ＩＮＳＫＤ２７０６は新しいソフトウェアに署名することができず、したがって、システム２７０２上のソフトウェアの変更またはインストールを防止する。すでにインストールされているソフトウェアは正常に実行することができ、ＩＮＳＫＤ２７０６はデジタル署名妥当性検査、暗号化解除、および暗号化のサービスを提供することができる。あまり制限的ではない一実施形態では、ホスト・システム上で実行される任意のアプリケーションによってＩＮＳＫＤ２７０６の暗号機能を呼び出すことができる。より制限的な一実施形態では、ＳＷＳＫＵ２７１０（図３３および図３５）などのソフトウェア・スマート・キー・ユニットを含むアプリケーション

のみによってＩＮＳＫＤ２７０６の暗号機能を呼び出すことができる。

【０１４１】

次に図３７に関連して説明すると、この流れ図は、本発明の一実施形態により、特定のＣＤスマート・キー・ユニットによって使用するためのホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスを描写している。このプロセスはブロック３１０２から始まり、そこで、ＣＤＤＳＫＵなどのＣＤドライバ・スマート・キー・ユニットを含むＣＤデバイス・ドライバ２７１４（図３３および図３５）が、たとえば、アプリケーション・プログラミング・インターフェース（ＡＰＩ）により、ＩＮＳＫＤ２７０６（図３３および図３５）の認証手順を呼び出す。ＩＮＳＫＤ２７０６または代替システム・ソフトウェア（図示せず）は、デジタル署名２９９０（図３５）が確かにＩＮＳＫＤ秘密鍵２９２６（図３５）によって署名されていることを検証するために、ＩＮＳＫＤ公開鍵２９３０（図３５）を使用する。この妥当性検査プロセスは、ＣＤＤＳＫＵ２９８２の相互認証鍵が他のアプリケーションまたは装置によって強奪されていないことを保証するものである。

【０１４２】

ブロック３１０４中にＩＮＳＫＤ２７０６はＣＤＤＳＫＵ２９８２に関する暗号機能を実行できるようになる。ブロック３１０６中にＣＤＤＳＫＵ２９８２はＩＮＳＫＤ２７０６の暗号機能を呼び出し、プロセスが終了する。ＣＤＤＳＫＵ２９８２に加えて、ホスト・システム上の複数のソフトウェア・スマート・キー・ユニットが内部スマート・キー・デバイスとの相互認証手順を完了したものと想定すると、ＩＮＳＫＤ２７０６は同時に、ＣＤＤＳＫＵ２９８２および複数のソフトウェア・スマート・キー・ユニットに代わって暗号機能を実行できるようになる可能性がある。

【０１４３】

ＣＤ２７１２がＣＤ装置２７０４（図３３および図３５）と、したがってＩＮＳＫＤ２７０６を含むシステム・ユニット２７０２とに連結されたままである間に、ＩＮＳＫＤ２７０６は、認証局として動作するための、すなわち、新しい公開証明書を生成するための機能を提供できるようになる。一実施形態では、ＣＤ２７１２は、新しいソフトウェア・パッケージをインストールするときにＣＤ装置２７０４およびシステム・ユニット２７０２に連結されているはずである。ＩＮＳＫＤ２７０６は、新しいソフトウェアに相互認証証明書を発行し、鍵とともにソフトウェアに署名しシールする。新しいソフトウェアに署名した後、ＣＤ２７１２を取り外すことができ、署名されたソフトウェアは、ＩＮＳＫＤ２７０６の暗号機能を始動し使用し続けることができる。

【０１４４】

ソフトウェア・インストール中に新しいソフトウェア・パッケージに新しい公開証明書を発行することができ、新たに発行されたデジタル証明書内の公開鍵に対応する秘密鍵はソフトウェア・パッケージ内に組み込むことができ、内部スマート・キー・デバイスによってソフトウェア・パッケージに署名させることにより、秘密鍵を保護することができる。さらに、Ｊａｖａ（登録商標）環境では、悪意のあるユーザが秘密鍵を改ざんするのを防止するために、秘密鍵が組み込まれているＪＡＲファイルおよびＪａｖａ（登録商標）パッケージをさらにシールすることができる。

【０１４５】

次に図３８に関連して説明すると、この流れ図は、本発明の一実施形態により、ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用不可にするためのプロセスを描写している。このプロセスはブロック３２０２から始まり、そこで、ＣＤ２７１２（図３３および図３４）がＣＤ装置２７０４から取り外され、したがって、ＩＮＳＫＤ２７０６（図３３および図３５）を含むシステム・ユニット２７０２（図３３および図３５）からＣＤ２７１２を切り離す。ブロック３２０４中にシステム・ユニット２７０２がＣＤ２７１２の分離を検出すると、ＩＮＳＫＤ２７０６は、さらにデジタル署名を実行するか、鍵を発行するか、または証明を実行することができない状態になり、プロセスが終了する。署名の妥当性検査、暗号化、および暗号化解除サービスなどの他の暗号機能は、ＩＮ

ＳＫＤ２７０６により実行し続けられることに留意されたい。

【０１４６】

図３８に図示されているプロセスは、図３６または図３７に図示されているプロセスのいずれかに対する補足的なプロセスとして機能する。しかし、ＩＮＳＫＤ２７０６は、この代替例では、本発明の実現例次第で、完全に使用不可にならないようないくつかの機能を実行し続ける可能性があることに留意されたい。また、図１０、図１１、図１２、図１３、図１４、図１５、図２５～図３２に併せて上述した内部スマート・キー・デバイスおよび外部スマート・キー・デバイスに関連するプロセスは、図３３～図３８に併せて上述した取り外し可能記憶媒体セキュリティ・システムの暗号機能にも適用可能であることにも留意されたい。さらに、ＣＤデバイス・ドライバ２７１４はシステム・ユニット２７０２上で実行されるソフトウェアであるので、ソフトウェアに関して本明細書に記載されているセキュリティ機能はＣＤデバイス・ドライバ２７１４に等しく適用可能である。簡単にするために、ＣＤＤＳＫＵ２９８２およびＣＤ２７１２に関して図１０、図１１、図１２、図１３、図１４、図１５、図２５～図３２に対応する図は複製されていない。

【０１４７】

内部スマート・キー・デバイス内の暗号機能は、ソフトウェアまたはハードウェアにより使用可能または使用不可にすることができるものと想定することができる。たとえば、ハードウェア・モードでは、外部スマート・キー・デバイスが受け入れられているかどうかを表す使用可能化状態に基づいて設定またはクリアしなければならない特定のフリップフロップまたはその他のメカニズムにより、内部スマート・キー・デバイス内の特定の回路の動作が動作可能状態に入ることを防止できる可能性があり、ソフトウェア・モードでは、暗号機能の実行を論理的に制御する特殊な使用可能化フラグを設定しクリアすることにより、特定の暗号機能の動作を保護することができる。

【０１４８】

本発明の利点は、上記で示されている詳細な説明を考慮して明らかになるはずである。本発明は、システム管理者がハードウェア・セキュリティ・トークンによりそれを物理的に可能にするときにのみ使用可能になるように、ホスト・システム内の暗号機能を保護するためのメカニズムを提供する。加えて、ハードウェア・セキュリティ・ユニットはデータ処理システム内に統合され、ハードウェア・セキュリティ・ユニットはハードウェア認証局として動作する。ハードウェア・セキュリティ・ユニットは、分散データ処理システム内の信頼階層または信頼フレームワークをサポートするものと見なすことができる。ハードウェア・セキュリティ・ユニットは、ハードウェア・セキュリティ・ユニットを含むマシン上にインストールされたソフトウェアに署名することができる。マシン上で実行される署名付きソフトウェアを使用するサーバ・プロセスは、ハードウェア・セキュリティ・ユニットとの相互信頼関係ならびにハードウェア・セキュリティ・ユニットに対するそれぞれの共通する信頼に基づく他のサーバ・プロセス間の相互信頼関係を確立することができる。

【０１４９】

本発明は完全に機能するデータ処理システムに関連して説明されているが、当業者であれば、分散を実行するために実際に使用される特定のタイプの信号伝送媒体にかかわらず、コンピュータ可読媒体内の命令の形および様々な他の形で本発明のプロセスを分散可能であることが分かることは、留意すべき重要なことである。コンピュータ可読媒体の例としては、ＥＰＲＯＭ、ＲＯＭ、テープ、紙、フレキシブル・ディスク、ハード・ディスク・ドライブ、ＲＡＭ、およびＣＤ－ＲＯＭなどの媒体と、デジタルおよびアナログ通信リンクなどの伝送タイプの媒体とを含む。

【０１５０】

方法は、一般に、所望の結果に至る首尾一貫した一連のアクションであると考えられている。このようなアクションは、物理量の物理的操作を必要とする。通常、これらの量は、保管、転送、結合、比較、およびその他の操作が可能な電気信号または磁気信号の形を取るが、必ずしもそうであるわけではない。時には、主に一般的使用法の理由により、こ

れらの信号をビット、値、パラメータ、項目、要素、オブジェクト、記号、文字、項、数などと呼ぶことは便利なことである。しかし、これらの用語および同様の用語はいずれも、適切な物理量に関連付けられるものであり、これらの量に適用される便利なラベルに過ぎないことに留意されたい。

【０１５１】

本発明の説明は、例示のために提示されているが、網羅するためのものでも、開示された諸実施形態に限定されるものでもない。当業者には多くの変更例および変形例が明らかになるであろう。諸実施形態は、他の企図された用途に適している可能性のある様々な変更により様々な諸実施形態を実現するために、本発明の原理およびその実用的な適用例を説明し、他の当業者が本発明を理解できるようにするために選択されている。

【０１５２】

疑いを回避するために、この説明および特許請求の範囲で使用する「有する（comprising）」という用語は、「のみで構成される（consisting only of）」という意味として解釈すべきではない。

【図面の簡単な説明】

【０１５３】

【図１】それぞれが本発明を実現可能な複数のデータ処理システムからなる典型的なネットワークを描写する図である。

【図２】本発明を実現可能なデータ処理システム内で使用できる典型的なコンピュータ・アーキテクチャを描写する図である。

【図３】ある個人がデジタル証明書を入手する際の典型的な方法を示すブロック図である。

【図４】あるエンティティがデジタル証明書を使用してデータ処理システムに対して認証される際の典型的な方法を示すブロック図である。

【図５】データ処理システム内のハードウェア・セキュリティ・ユニット内で暗号機能を使用可能にするために取り外し可能ハードウェア・デバイスを受け入れるデータ処理システムの一部分を示すブロック図である。

【図６】内部スマート・キー・デバイスを含み、内部スマート・キー・デバイス内の暗号機能を使用可能にするために外部スマート・キー・デバイスを使用するシステム・ユニットを示すブロック図である。

【図７】ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を示す流れ図である。

【図８】特定のソフトウェア・スマート・キー・ユニットによって使用するためのホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を示す流れ図である。

【図９】ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用不可にするためのプロセスを示す流れ図である。

【図１０】図７のブロック６０４に図示されている相互認証手順に関する詳細を示す１対の流れ図の一方である。

【図１１】図７のブロック６０４に図示されている相互認証手順に関する詳細を示す１対の流れ図のもう一方である。

【図１２】図８のブロック７０４に図示されている相互認証手順に関する詳細を示す１対の流れ図の一方である。

【図１３】図８のブロック７０４に図示されている相互認証手順に関する詳細を示す１対の流れ図のもう一方である。

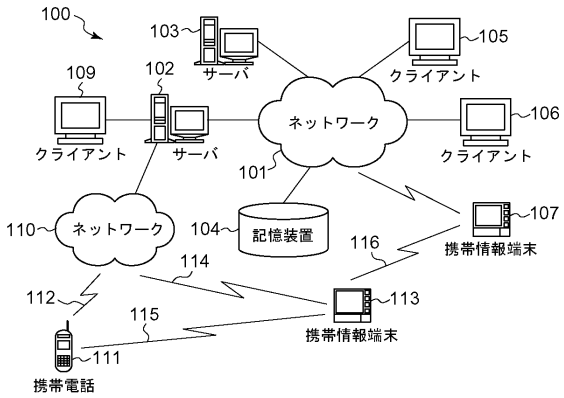
【図１４】外部スマート・キー・デバイスの存在に基づいて動作が使用可能または使用不可になるソフトウェア・スマート・キー・ユニットによって要求された動作を実行するための内部スマート・キー・デバイス内のプロセスを示す流れ図である。

【図１５】外部スマート・キー・デバイスの存在によって動作が使用可能になる必要がないソフトウェア・スマート・キー・ユニットによって要求された動作を実行するための内

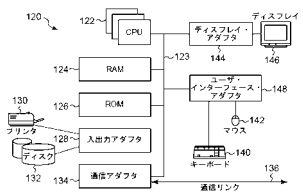
部スマート・キー・デバイス内のプロセスを示す流れ図である。
【図１６】マスタ・シークレットを保護するための本発明の一実施形態を示すブロック図である。
【図１７】複数の外部スマート・キー・デバイスと複数の内部スマート・キー・デバイスとの種々の関係を示すブロック図である。
【図１８】複数の外部スマート・キー・デバイスと複数の内部スマート・キー・デバイスとの種々の関係を示すブロック図である。
【図１９】複数の外部スマート・キー・デバイスと複数の内部スマート・キー・デバイスとの種々の関係を示すブロック図である。
【図２０】典型的な１組の信頼できる関係を示すブロック図である。
【図２１】典型的な１組の信頼できる関係を示すブロック図である。
【図２２】典型的な１組の信頼できる関係を示すブロック図である。
【図２３】内部スマート・キー・デバイスによって提供される信頼に基づく信頼関係から構築される信頼モデルの一例を示すブロック図である。
【図２４】オペレーティング・システム内の各プログラマチック・エンティティが内部スマート・キー・デバイスに基づいて信頼階層内に信頼関係を確立するための機能を含む、オペレーティング・システム・ファイルを生成するためのデータ処理システムを示すブロック図である。
【図２５】オペレーティング・システム・モジュールが相互に認証動作を実行できるようなソフトウェア・スマート・キー・ユニットを含むオペレーティング・システム・モジュールを生成するためのプロセスを示す流れ図である。
【図２６】各プログラマチック・エンティティが内部スマート・キー・デバイスに基づいて信頼階層内に信頼関係を確立するための機能を含む、プロジェクト・コードを生成するためのデータ処理システムを示すブロック図である。
【図２７】内部スマート・キー・デバイスに関する証明書チェーンを拡張するためのプロセスを示す流れ図である。
【図２８】フォーリン内部スマート・キー・デバイスに関する複数のルート証明書を含む証明書チェーンを維持する単一のローカル内部スマート・キー・デバイスによって提供される信頼に基づく信頼関係から構築される信頼モデルの一例を示すブロック図である。
【図２９】ローカル内部スマート・キー・デバイスによって維持される現行ルート証明書チェーンを入手するためのプロセスを示す流れ図である。
【図３０】フォーリン内部スマート・キー・デバイスからのデジタル証明書が信頼すべきものであるかどうかを判断するためのプロセスを示す流れ図である。
【図３１】ソフトウェア・モジュールの保全性を確保するために使用可能なハードウェア支援信頼モデル内のエンティティを示すデータフロー・ダイアグラムである。
【図３２】ソフトウェア・モジュールの保全性を確保するためのプロセスを示す流れ図である。
【図３３】データ処理システム内のハードウェア・セキュリティ・ユニット内で暗号機能を使用可能にするために取り外し可能記憶媒体を受け入れるデータ処理システムの一部分を示すブロック図である。
【図３４】内部スマート・キー・デバイス内の暗号機能を使用可能にする取り外し可能記憶媒体のブロック図である。
【図３５】内部スマート・キー・デバイス内の暗号機能を使用可能にするための図３４の取り外し可能記憶媒体とともに、内部スマート・キー・デバイスと取り外し可能記憶媒体スマート・キー・デバイスとを含むシステム・ユニットを示すブロック図である。
【図３６】ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を示す流れ図である。
【図３７】特定の取り外し可能媒体によって使用するためのホスト・システムの内部スマート・キー・デバイスの暗号機能を使用可能にするためのプロセスの概要を示す流れ図である。

【図３８】ホスト・システムの内部スマート・キー・デバイスの暗号機能を使用不可にするためのプロセスを示す流れ図である。

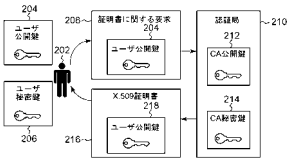
【図１】



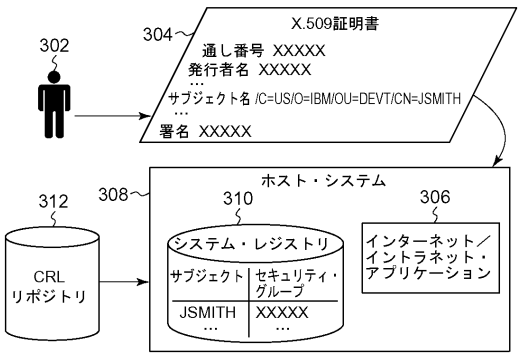
【図２】



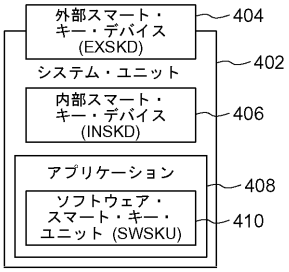
【図３】



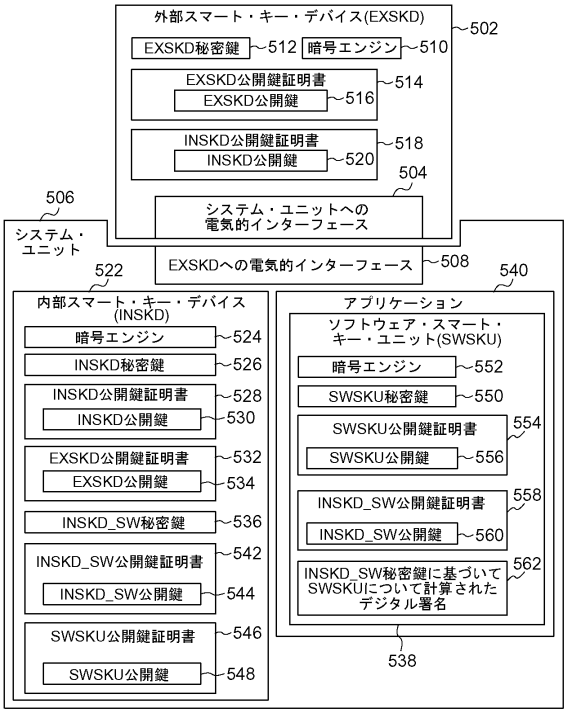
【図４】



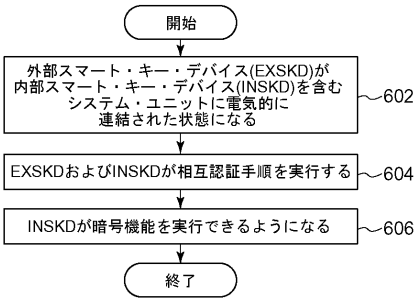
【図５】


外部スマート・
キー・デバイス
(EXSKD)
404
システム・ユニット
402
内部スマート・
キー・デバイス
(INSKD)
406
アプリケーション
408
ソフトウェア・
スマート・キー・
ユニット (SWSKU)
410

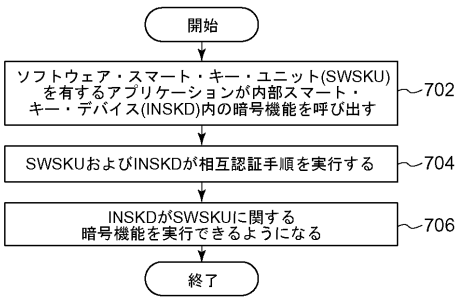
【図６】


外部スマート・キー・デバイス(EXSKD)
502
EXSKD秘密鍵
512
暗号エンジン
510
EXSKD公開鍵証明書
514
EXSKD公開鍵
516
INSKD公開鍵証明書
518
INSKD公開鍵
520
504
システム・ユニットへの
電気的インターフェース
506
システム・
ユニット
EXSKDへの電気的インターフェース
508
522
540
内部スマート・キー・デバイス
(INSKD)
暗号エンジン
524
INSKD秘密鍵
526
INSKD公開鍵証明書
528
INSKD公開鍵
530
EXSKD公開鍵証明書
532
EXSKD公開鍵
534
INSKD_SW秘密鍵
536
INSKD_SW公開鍵証明書
542
INSKD_SW公開鍵
544
SWSKU公開鍵証明書
546
SWSKU公開鍵
548
アプリケーション
ソフトウェア・スマート・
キー・ユニット(SWSKU)
暗号エンジン
552
SWSKU秘密鍵
550
SWSKU公開鍵証明書
554
SWSKU公開鍵
556
INSKD_SW公開鍵証明書
558
INSKD_SW公開鍵
560
INSKD_SW秘密鍵に基づいて
SWSKUについて計算された
デジタル署名
562
538

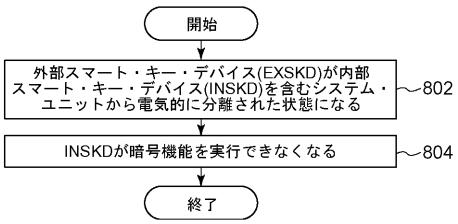
【図７】


開始
外部スマート・キー・デバイス(EXSKD)が
内部スマート・キー・デバイス(INSKD)を含む
システム・ユニットに電気的に
連結された状態になる
602
EXSKDおよびINSKDが相互認証手順を実行する
604
INSKDが暗号機能を実行できるようになる
606
終了

【図８】


開始
ソフトウェア・スマート・キー・ユニット(SWSKU)
を有するアプリケーションが内部スマート・
キー・デバイス(INSKD)内の暗号機能を呼び出す
702
SWSKUおよびINSKDが相互認証手順を実行する
704
INSKDがSWSKUに関する
暗号機能を実行できるようになる
706
終了

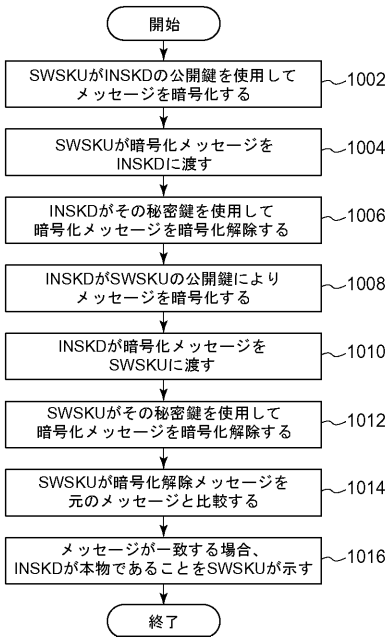
【図９】


開始
外部スマート・キー・デバイス(EXSKD)が内部
スマート・キー・デバイス(INSKD)を含むシステム・
ユニットから電気的に分離された状態になる
802
INSKDが暗号機能を実行できなくなる
804
終了

【図１０】

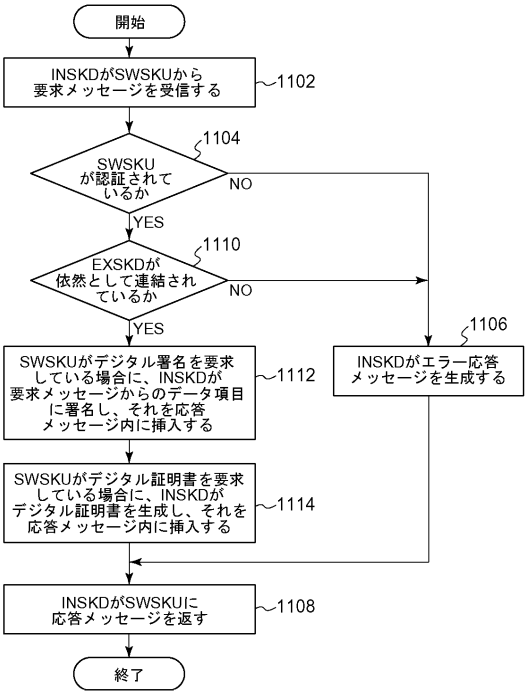
【図１１】

【図１２】

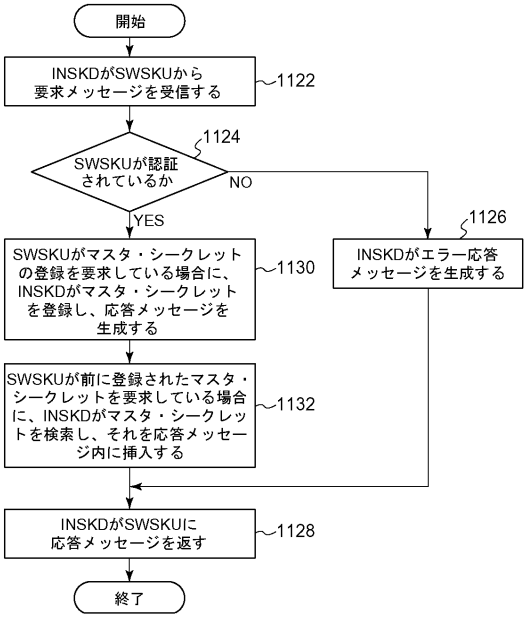


【図１３】

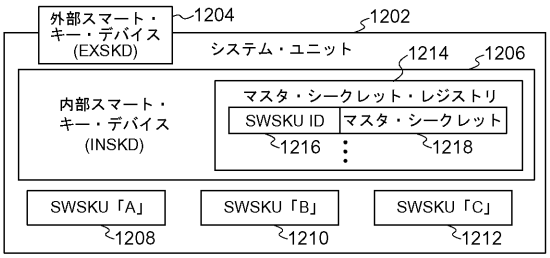
【図１４】



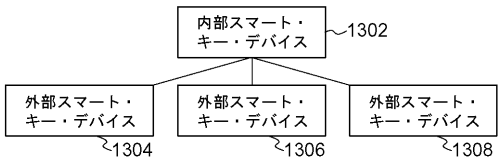
【図１５】



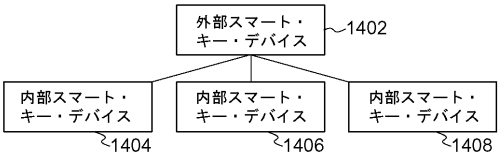
【図１６】



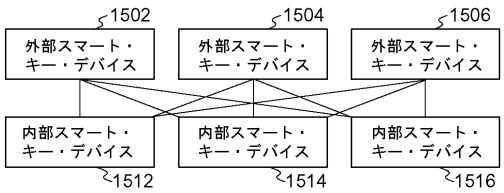
【図１７】



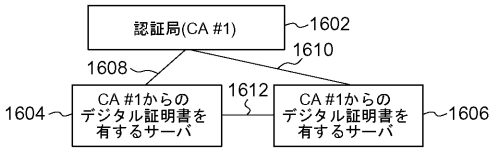
【図１８】



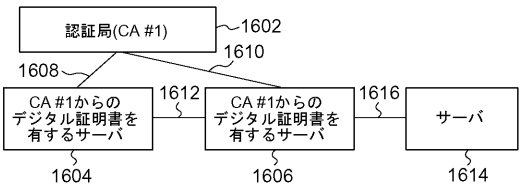
【図１９】



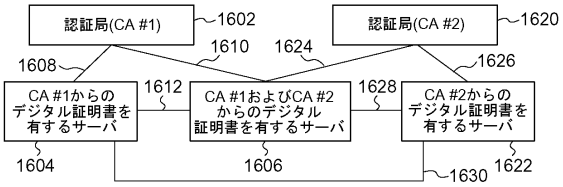
【図２０】



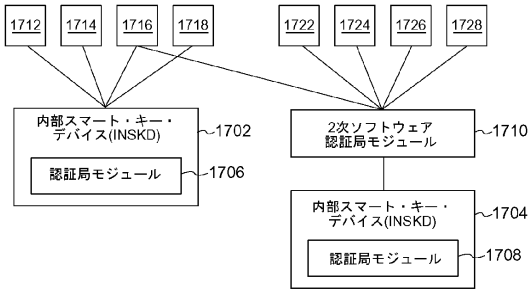
【図２１】



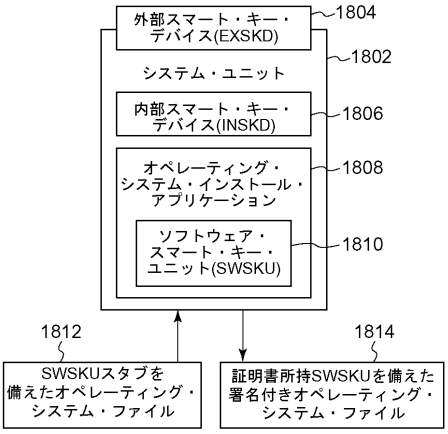
【図２２】



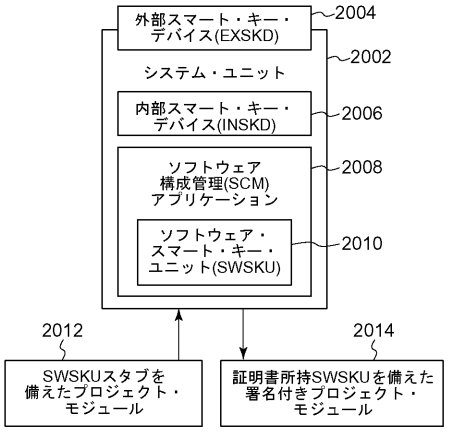
【図２３】



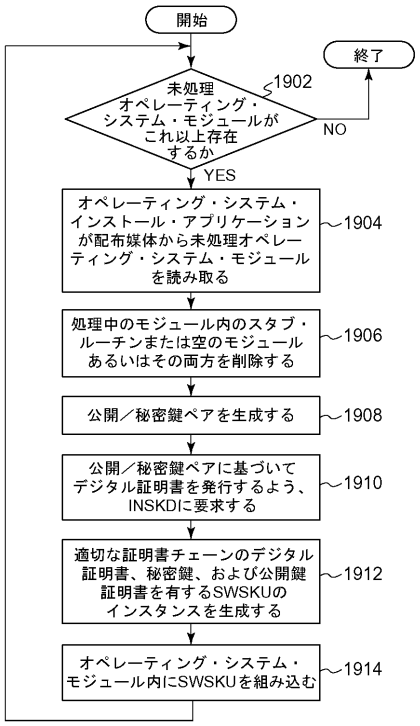
【図２４】



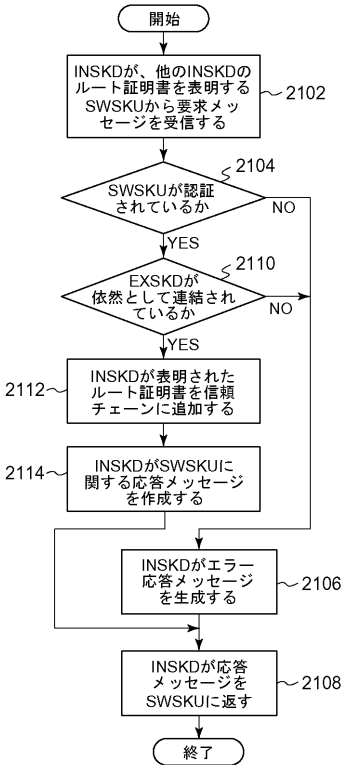
【図２５】



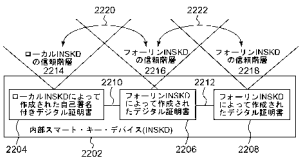
【図２６】



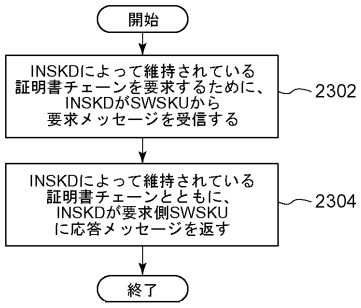
【図２７】



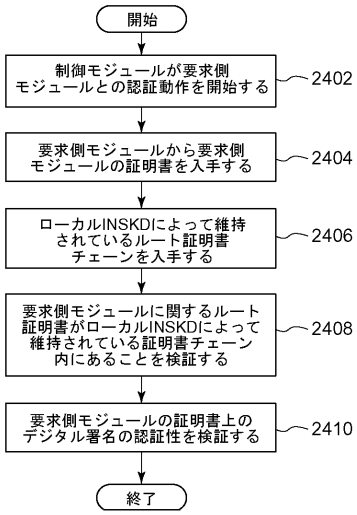
【図２８】



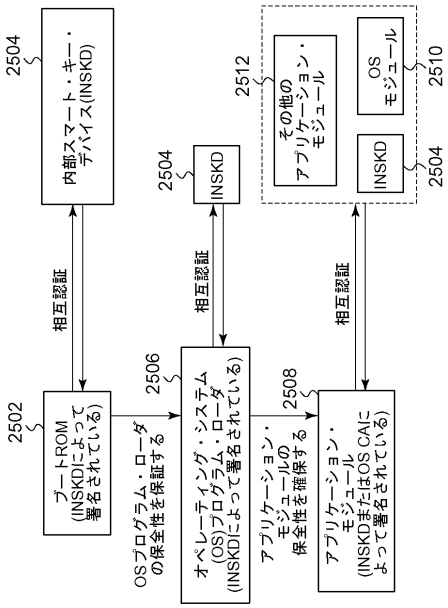
【図２９】



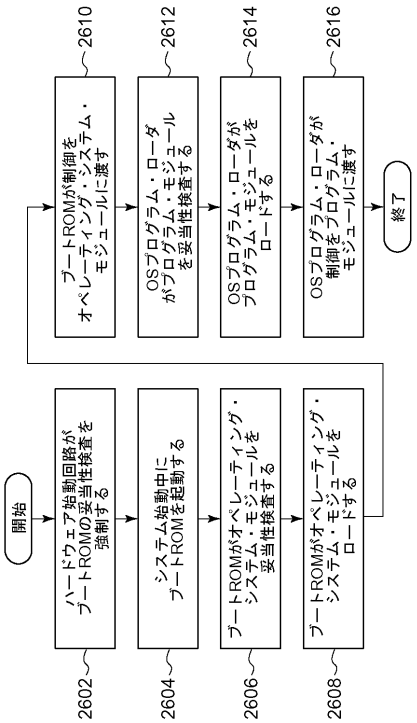
【図３０】



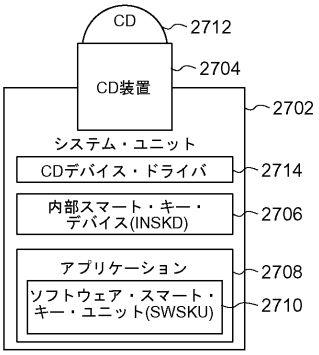
【図３１】



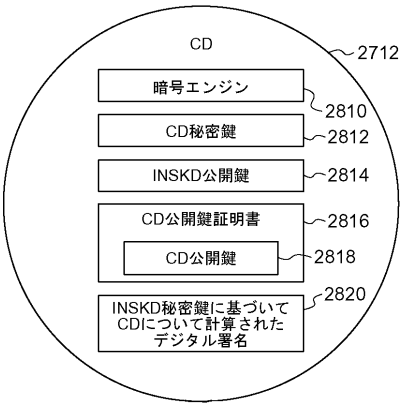
【図３２】



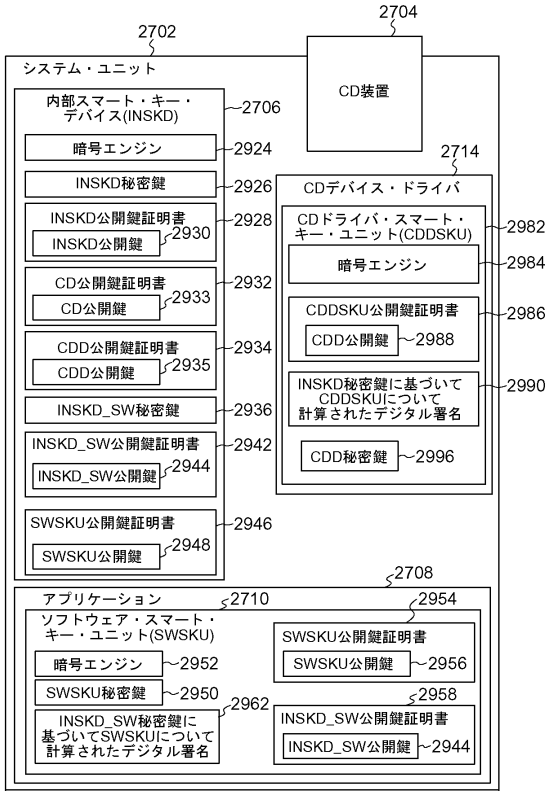
【図３３】



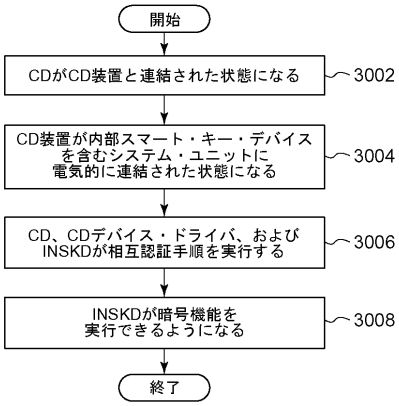
【図３４】



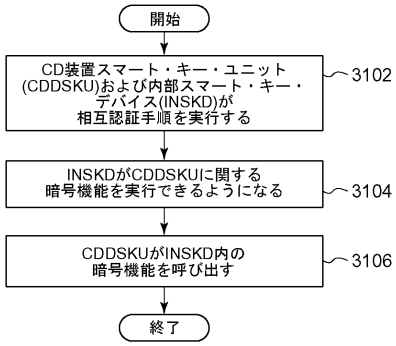
【図３５】



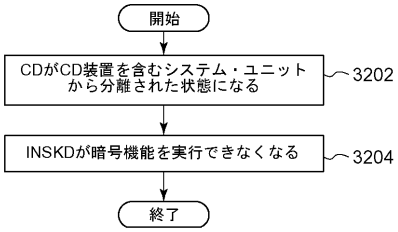
【図３６】


開始
CDがCD装置と連結された状態になる
3002
CD装置が内部スマート・キー・デバイス
を含むシステム・ユニットに
電気的に連結された状態になる
3004
CD、CDデバイス・ドライバ、および
INSKDが相互認証手順を実行する
3006
INSKDが暗号機能を
実行できるようになる
3008
終了

【図３７】


開始
CD装置スマート・キー・ユニット
(CDDSKU)および内部スマート・キー・
デバイス(INSKD)が
相互認証手順を実行する
3102
INSKDがCDDSKUに関する
暗号機能を実行できるようになる
3104
CDDSKUがINSKD内の
暗号機能を呼び出す
3106
終了

【図３８】


開始
CDがCD装置を含むシステム・ユニット
から分離された状態になる
3202
INSKDが暗号機能を実行できなくなる
3204
終了

フロントページの続き

(74)代理人 100086243
弁理士 坂口 博
(72)発明者 ベイド、スティーヴン
アメリカ合衆国７８６２８ テキサス州ジョージタウン バック・ベンド２０４
(72)発明者 チョア、チン－ユン
アメリカ合衆国７８７５０ テキサス州オースティン ラスティック・ロック・レーン１１５１８

審査官 新田 亮

(56)参考文献 特表２００２－５３６７５７（ＪＰ，Ａ）
特開２００４－３２０５９３（ＪＰ，Ａ）
国際公開第２００３／７３６８８（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L 9/10
G06K 17/00
G06K 19/10