(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4837985号
(P4837985)
(45)発行日　平成23年12月14日(2011.12.14)　(24)登録日　平成23年10月7日(2011.10.7)

(51)Int.Cl.　　F I
G06F 21/22 (2006.01)　　G06F 9/06 660J
G06F 21/24 (2006.01)　　G06F 12/14 560C

請求項の数 20　(全 30 頁)

(21)出願番号　特願2005-353934(P2005-353934)
(22)出願日　平成17年12月7日(2005.12.7)
(65)公開番号　特開2006-323814(P2006-323814A)
(43)公開日　平成18年11月30日(2006.11.30)
審査請求日　平成20年11月21日(2008.11.21)
(31)優先権主張番号　11/031,161
(32)優先日　平成17年1月7日(2005.1.7)
(33)優先権主張国　米国(US)

(73)特許権者　500046438
マイクロソフト　コーポレーション
アメリカ合衆国　ワシントン州　９８０５２－６３９９　レッドモンド　ワン　マイクロソフト　ウェイ
(74)代理人　100077481
弁理士　谷　義一
(74)代理人　100088915
弁理士　阿部　和夫
(72)発明者　ジェームズ　アンソニー　シュワルツ　ジュニア
アメリカ合衆国　９８０５２　ワシントン州　レッドモンド　ワン　マイクロソフト　ウェイ　マイクロソフト　コーポレーション内

最終頁に続く

(54)【発明の名称】信頼できる処理モジュールを有するコンピュータを安全にブートするためのシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
記録された値を含み、サブミットされた値を前記記録された値と比較し、前記サブミットされた値が正しい場合に秘密を開示するハードウェア・セキュリティ・モジュール（ＨＳＭ）を有する、コンピュータ上の安全なブート・プロセスのための命令を有するコンピュータ読取可能な記憶メディアであって、
少なくとも１つの値をＨＳＭにサブミットするための命令であって、前記値が正しい場合に前記ＨＳＭが秘密を開示する命令と、
秘密を取り出すための命令と、
前記秘密を取り出した結果アクセス可能になる情報を用いてデータを解読するための命令であって、前記解読のための命令の実行が解読されたデータを生成する命令と、
前記解読されたデータなしで通常のブートを完了できないコンピュータ・ブート・プロセスの少なくとも一部のための命令と、
を備えることを特徴とするコンピュータ読取可能な記憶メディア。
【請求項２】
前記ＨＳＭは、信頼できるプラットフォーム・モジュール（ＴＰＭ）であり、少なくとも１つの値は、少なくとも１つのプラットフォーム構成レジスタ（ＰＣＲ）内に置かれることを特徴とする請求項１に記載のコンピュータ読取可能な記憶メディア。
【請求項３】
前記解読されたデータは、コンピュータ・ブート・プロセスに用いられるソフトウェア

・コンポーネントを含むことを特徴とする請求項１に記載のコンピュータ読取可能な記憶メディア。
【請求項４】
前記解読されたデータは、コンピュータ・プロセスを続けるために、前記コンピュータ・ブート・プロセスに用いられるソフトウェア・コンポーネントが必要とする情報を含むことを特徴とする請求項１に記載のコンピュータ読取可能な記憶メディア。
【請求項５】
前記解読されたデータは、前記コンピュータ読取可能な記憶メディア上に格納されたデータにアクセスするために必要とされる情報を含むことを特徴とする請求項１に記載のコンピュータ読取可能な記憶メディア。
【請求項６】
前記秘密をメモリから除去するための命令をさらに備えることを特徴とする請求項１に記載のコンピュータ読取可能な記憶メディア。
【請求項７】
記録された値を含み、サブミットされた値を前記記録された値と比較し、サブミットされた値が正しい場合に秘密を開示するハードウェア・セキュリティ・モジュール（ＨＳＭ）を有するコンピュータであって、前記コンピュータはまた、
少なくとも１つの値を該ＨＳＭにサブミットする手段であって、前記値が正しい場合に前記ＨＳＭが秘密を開示する手段と、
秘密を取り出すための手段と、
前記秘密を取り出した結果アクセス可能になる情報を用いてデータを解読するための手段であって、前記解読するための手段の動作が解読されたデータを生成する手段と、
前記解読されたデータなしで通常のブートを完了できないコンピュータ・ブート・プロセスの少なくとも一部を含む手段と、
を備えることを特徴とするコンピュータ。
【請求項８】
前記ＨＳＭは、信頼できるプラットフォーム・モジュール（ＴＰＭ）であり、少なくとも１つの値は、プラットフォーム構成レジスタ（ＰＣＲ）内に置かれることを特徴とする請求項７に記載のコンピュータ。
【請求項９】
前記解読されたデータは、前記コンピュータ・ブート・プロセスに用いられるソフトウェア・コンポーネントを含むことを特徴とする請求項７に記載のコンピュータ。
【請求項１０】
前記解読されたデータは、コンピュータ・プロセスを続けるために、前記コンピュータ・ブート・プロセスに用いられるソフトウェア・コンポーネントが必要とする情報を含むことを特徴とする請求項７に記載のコンピュータ。
【請求項１１】
前記解読されたデータは、コンピュータ読取可能な記憶メディア上に格納されたデータにアクセスするために必要とされる情報を含むことを特徴とする請求項７に記載のコンピュータ。
【請求項１２】
前記秘密をメモリから除去するための手段をさらに備えることを特徴とする請求項７に記載のコンピュータ。
【請求項１３】
複数のパーティションと、記録された値を含み、サブミットされた値を前記記録された値と比較し、前記サブミットされた値が正しい場合に秘密を開示するハードウェア・セキュリティ・モジュール（ＨＳＭ）とを有するコンピュータ上に、安全なブート・プロセスのための命令を有するコンピュータ読取可能な記憶メディアであって、前記コンピュータ読取可能な記憶メディアは、
少なくとも１つの値をＨＳＭにサブミットする命令であって、前記値が正しい場合に前

記ＨＳＭが秘密を開示することができる命令と、
第１の秘密を取り出すための命令と、
前記第１の秘密を記憶場所から除去するための命令と、
少なくとも１つの第２の値を該ＨＳＭにサブミットするための命令であって、前記第２の値が正しい場合に前記ＨＳＭが前記第１の秘密ではなく第２の秘密を開示することができる命令と、
を備えることを特徴とするコンピュータ読取可能な記憶メディア。
【請求項１４】
前記ＨＳＭは、信頼できるプラットフォーム・モジュール（ＴＰＭ）であり、少なくとも１つの値および前記少なくとも１つの第２の値は、プラットフォーム構成レジスタ（ＰＣＲ）内に置かれることを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【請求項１５】
コンピュータ・ブート・プロセスの少なくとも一部のための命令をさらに備え、前記コンピュータ・ブート・プロセスは、前記第１の秘密なしで通常のブートを完了できないことを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【請求項１６】
コンピュータ読取可能な記憶メディアの少なくとも１つのパーティション上に格納されたほぼ全てのデータにアクセスするために、前記第２の秘密が必要とされることを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【請求項１７】
前記第１の値は、コンピュータ・ブート・プロセスに用いられるソフトウェア・コンポーネントのハッシュ値を含むことを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【請求項１８】
前記第２の値は解読鍵のハッシュ値を含むことを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【請求項１９】
前記第１の秘密および前記第２の秘密のうちの少なくとも１つは、バイナリ・ラージ・オブジェクト（ＢＬＯＢ）であることを特徴とする請求項１３の記載のコンピュータ読取可能な記憶メディア。
【請求項２０】
前記第１の秘密および前記第２の秘密のうちの少なくとも１つは解読鍵であることを特徴とする請求項１３に記載のコンピュータ読取可能な記憶メディア。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、コンピューティングの分野に関する。より具体的には、本発明は、ブート中に用いられるデータの無許可の修正を防止し、ブート中にのみ必要とされるリソースへのブート後のアクセスを防止することによって、コンピュータのセキュリティを強化するためのシステムおよび方法を提供する。
【背景技術】
【０００２】
セキュリティは、コンピュータ・ユーザにとって、広く行きわたった問題になった。ウイルス、ワーム、トロイの木馬、個人情報の盗難、ソフトウェア、およびメディア・コンテンツの違法コピー、およびデータ破壊の脅威を使用する恐喝が横行している。オペレーティング・システムは、こうした攻撃を防ぐために、多数のセキュリティ機能を提供することができる。しかしながら、オペレーティング・システムのセキュリティ機能は、これらがディスエーブルにされた場合に無効になる。こうしたセキュリティ機能をディスエーブルにすることが試みられた場合、オペレーティング・システムのブート中に試みられる

可能性が高い。ブート後、オペレーティング・システムは、該オペレーティング・システム自体および該オペレーティング・システムが管理するデータおよびプロセスを保護するために、多数の適切な機能を所定位置に有することができる。しかしながら、ブート中には、まだ、これらの機能を初期化することができず、バイパスおよび／または不正変更に対して無防備である。

【０００３】

現在のところオペレーティング・システムによって用いられる例示的なセキュリティ機能は、暗号化されたファイル・システム（ＥＦＳ）および信頼できるプラットフォーム・モジュール（ＴＰＭ）である。ＥＦＳ機能は、選択された機密データを暗号化する。オペレーティング・システムは、ユーザがログオンするまでＥＦＳデータにアクセスする必要がない。オペレーティング・システムのブート後、ユーザは、ログオン・プロセスに対してパスワードを提供することができる。このパスワードは、ＥＦＳデータを解読することができる解読鍵へのアクセスを許可するものである。一例として、ＭＩＣＲＯＳＯＦＴ ＷＩＮＤＯＷＳ（登録商標）オペレーティング・システムは、ＳＹＳＫＥＹの可用性に依存して、これらのプロセスの正しい性能を行うことによって、種々のプロセスを保護するために用いられるシステム・キー即ち「ＳＹＳＫＥＹ」を使用する。例えば、オペレーティング・システムによって暗号化された形で格納されるＥＦＳデータを解読するのに必要とされる鍵を、ＳＹＳＫＥＹから導き出すことが可能である。

【０００４】

したがって、制限的な動作を行うのに必要とされる鍵は、ログオン手続きによって保護される。一般に、ユーザは、システムの使用を開始する前に、ユーザ自身を正しく認証しなければならない。鍵の使用は、ユーザが正しく認証する場合にのみイネーブルにされる。しかしながら、鍵へのアクセスを保護するためにログオン手続きを用いることは、オペレーティング・システムが正しいログオン手続きを読み込んだこと、および稼動中の非認識（ｒｏｇｕｅ）コードによって、該鍵の使用が他の方法でイネーブルにされなかったことを仮定する。正しいオペレーティング・システム・ローダーの代わりに、ブート中に非認識オペレーティング・システム・ローダーを使用した場合、この非認識ローダーは、オペレーティング・システムを用いて非認識ログオン・プログラムを読み込ませることがある。この非認識ログオン・プログラムは、適正なパスワードの入力なしに、ＥＦＳ鍵の使用をイネーブルにすることができる。オペレーティング・システムを読み込むことで、セキュリティ侵害の機会がもたらされるので、こうした状況で鍵を保護するには、オペレーティング・システムの読み込みを正しく行うことが確認できる状況のもとで、オペレーティング・システムを読み込むことが必要である。

【０００５】

ＥＦＳは、ブートされたオペレーティング・システムによってサポートされる機能であるので、ブート・プロセス中に漏れる特定のデータの保護に、同じような効果はない。ＥＦＳは、幾つかのシステム・サービスに必要とされる秘密、ネットワーク・サービス（例えば、個人のサーバまたは公開ウェブ・サーバ）によって使用されるデータベース、および企業のドメインに接続するためのＲＡＳ資格証明書のような、ユーザがログオンする前に必要とされるユーザ・データを保護することができない。

【０００６】

信頼できる処理モジュール（ＴＰＭ）は、コンピュータ上で動作するソフトウェアの信頼性を保証する。一般に、このことは、信頼できるデータの測定値をＴＰＭにサブミットし、ＴＰＭに依存して該測定値がどんなものであるべきかを判断することによって達成される。コンピュータ・セキュリティは、多くの場合、ソフトウェア・コンポーネントの動作を予測できることに依存している。一般に、システムのセキュリティは、周知の良好な状態から進行する、動作が理解されている周知のプログラムが、予測可能な方法で動作するという前提から生じる。逆に、一般に、周知のプログラムを交換するかまたは変更することによって、あるいはその動作が理解されていない状態でプログラムを実行することによって、セキュリティの妨害（その設計者が考えた以外の方法でコンピュータ・システム

を動作させることを含むことができる）を実現することができる。したがって、コンピューティング環境にセキュリティを提供する１つの態様は、周知のプログラムが用いられ、該プロセスが周知の良好な状態から進行することを確認するステップを含む。データのハッシュ値のような測定値は、以前にＴＰＭ内に秘匿された（ｓｅａｌｅｄ）値と合致するので、ＴＰＭは、そのデータがどんなものであるべきかを検証することによって、このことを達成する。

【０００７】

ＥＦＳと同様に、ＴＰＭは、ブートされたコンピュータ上で動作しているアプリケーションの完全性に関して一定の保証を提供するようにうまく使用された。ＴＰＭに対して、多数の付加的な制限も存在する。例えば、標準的な方法でＴＰＭを用いるマシンは、現場で再構成することができない（例えば、会議中にネットワークカードをラップトップに挿入することなど）。ＴＰＭは、初期化されたオペレーティング・システムに、厳しい制限と複雑性を与える。

【０００８】

今日の大部分のＴＰＭは、現在のところ、非特許文献１において入手可能な、「Ｔｒｕｓｔｅｄ　Ｐｌａｔｆｏｒｍ　Ｍｏｄｕｌｅ（ＴＰＭ）仕様バージョン１．２」という名称のＴＲＵＳＴＥＤ　ＣＯＭＰＵＴＩＮＧ　ＧＲＯＵＰ（登録商標）（ＴＣＧ）規格に準拠している。ＴＰＭは、プラットフォームにより実行されるコードの信頼を確立するために、コンピューティング・プラットフォームに組み込むことができるサブシステムである。セキュリティおよび暗号コミュニティが、セキュリティを提供するための機構を評価することを可能にし、顧客の理解および新しいソフトウェア機能の信頼を深めるので、信頼のおけるコードを確立するための機構の標準化は、有益なことである。このことはまた、ＴＣＧ（登録商標）によって考えられ、促進されるような規格の実装および改良におけるイノベーションを促進する。ＴＣＧ（登録商標）仕様に述べられるように、「製造者は、様々な能力およびコストポイントを有するサブシステムをインストールすることにより、市場で競合するであろう」。

【０００９】

オペレーティング・システムによって用いられる上記の例示的なセキュリティ機構は、ブートをセキュリティ保護するための幾つかの技術によって補われる。マシンのグローバル・パスワードを用いて秘密を保護するために、マシンのパスワード認証を用いることができる。しかしながら、このことは、マシンがブートする前にパスワードを入力することを必要とする。マシンの多数のユーザがパスワードを共有しなければならないことが、第１のセキュリティ欠陥である。第２の欠陥は、パスワードの入力のための典型的なユーザ・インターフェースを与えることができないので、使いやすさの問題が生じることである。このことは、タブレット型ＰＣの場合、特に面倒である。マシンのグローバル・パスワードは、紙に書かれ、マシンの前に置かれることが多い。したがって、パスワードは有効であるが、高い頻度で望まれるタイプの高度なユーザ保護を可能にするものではない。

【００１０】

第２に、秘密は、取り外し可能なメディア上に格納することができる。また、この特徴は、セキュリティの観点から理論上有効なものであるが、実際は問題になることが多い。この場合の基本的な問題は、使用に適したシステムの働きを保証するために、取り外し可能なメディアが、ほとんどの場合マシンの内部に残されるという点である。

安全なオペレーティング・システム・ブートの適切な保証がない場合、コンピュータ上のデータを保護するユーザの能力は、該コンピュータ上で動作しているオペレーティング・システムのセキュリティ機能ではなく、こうしたコンピュータが固定されている建物のセキュリティによって制限される。ラップトップの人気と、コンピュータの盗難、特にラップトップの盗難の増加につれて、コンピュータが窃盗犯の手に入ったときに、オペレーティング・システムのセキュリティが危険にさらされないままであることを可能にする解決法が好ましい。

【００１１】

ブート・プロセスをセキュリティ保護するためにＴＰＭを用いるシステムおよび方法は、大部分が未開拓のままである。ブート・プロセスにおいてＴＰＭを使用することに加えて、ブート・プロセスの保守を行い、こうしたコンピュータ上のデータへのアクセスを制御するためのシステムおよび方法が、有用であることが判明した。こうしたシステムおよび方法の説明は、特許文献１、特許文献２、および特許文献３に見出すことができる。また、特許文献４も全体的に本発明に関連している。

【００１２】

【特許文献１】に出願された「Ｓｙｓｔｅｍｓ ａｎｄ Ｍｅｔｈｏｄｓ ｆｏｒ Ｂｏｏｔ Ｒｅｃｏｖｅｒｙ ｉｎ ａ Ｓｅｃｕｒｅ Ｂｏｏｔ Ｐｒｏｃｅｓｓ ｏｎ ａ Ｃｏｍｐｕｔｅｒ ｗｉｔｈ ａ Ｈａｒｄｗａｒｅ Ｓｅｃｕｒｉｔｙ Ｍｏｄｕｌｅ」という名称の米国特許第　　号（代理人整理番号ＭＳＦＴ４６３４／３１１２２６．０１号）

【特許文献２】に出願された「Ｓｙｓｔｅｍｓ ａｎｄ Ｍｅｔｈｏｄｓ ｆｏｒ Ｕｐｄａｔｉｎｇ ａ Ｓｅｃｕｒｅ Ｂｏｏｔ Ｐｒｏｃｅｓｓ ｏｎ ａ Ｃｏｍｐｕｔｅｒ ｗｉｔｈ ａ Ｈａｒｄｗａｒｅ Ｓｅｃｕｒｉｔｙ Ｍｏｄｕｌｅ」という名称の米国特許第　　号（代理人整理番号ＭＳＦＴ４７８４／３１２０８６．０１号）

【特許文献３】に出願された「Ｓｙｓｔｅｍｓ ａｎｄ Ｍｅｔｈｏｄｓ ｆｏｒ Ｃｏｎｔｒｏｌｌｉｎｇ Ａｃｃｅｓｓ ｔｏ Ｄａｔａ ｏｎ ａ Ｃｏｍｐｕｔｅｒ ｗｉｔｈ ａ Ｓｅｃｕｒｅ Ｂｏｏｔ Ｐｒｏｃｅｓｓ」という名称の米国特許出願第＿＿＿号（代理人整理番号ＭＳＦＴ４６３５／３１１２２７．０１号）

【特許文献４】２００４年6月３０日に出願された「Ｓｙｓｔｅｍｓ ａｎｄ ｍｅｔｈｏｄ ｆｏｒ ｐｒｏｔｅｃｔｅｄ ｏｐｅｒａｔｉｎｇ ｓｙｓｔｅｍ ｂｏｏｔ ｕｓｉｎｇ ｓｔａｔｅ ｖａｌｉｄａｔｉｏｎ」という名称の米国特許出願番号第１０／８８２，１３４号

【非特許文献１】ｈｔｔｐｓ：／／ｗｗｗ．ｔｒｕｓｔｅｄｃｏｍｐｕｔｉｎｇｇｒｏｕｐ．ｏｒｇ／ｈｏｍｅ

【発明の開示】

【課題を解決するための手段】

【００１３】

上記に鑑みて、本発明は、ソフトウェア・コンポーネントの完全性メトリックを検証するための信頼できるプラットフォーム・モジュール（ＴＰＭ）を有するコンピュータを、安全にブートするためのシステムおよび方法を提供するものである。本発明と共に用いるＴＰＭは、秘密を、複数のプラットフォーム構成レジスタ（ＰＣＲ）値に秘匿することができる。ＰＣＲ値は、ブート・コンポーネントを測定することによって取得することができる。秘密が秘匿された時点よりブート・コンポーネントが修正されていない場合には、適切なシステム・ブートのために、秘密を取得することができる。ブート・コンポーネントの予想されるハッシュ値をＰＣＲ内に置くことができ、該予想される値が正しい場合に秘密を開示することができる。次に、秘密を用いて、ディスク上の場所から実際のブート・コンポーネントを解読することができる。解読されたブート・コンポーネントのハッシュ値を計算し、その結果を予想される値と比較することができる。別の例は、ブート・プロセスの異なる時点において取得可能な、ＰＣＲ値に秘匿することができる２つの秘密を使用することを伴う。第１の秘密は、第１の複数のＰＣＲ値が読み込まれたときのみアクセス可能となり、第２の秘密は、第１の複数の値の１つまたは複数が新しい値と置き換えられた後にのみアクセス可能となり、これにより、該第２の秘密へのアクセスを許可するために、必然的に第１の秘密への更なるアクセスが無効にされる。本発明の他の利点および特徴が、以下に説明される。

【発明を実施するための最良の形態】

【００１４】

本発明に従ってコンピュータを安全にブートするためのシステムおよび方法が、添付図面を参照してさらに説明される。

【００１５】
本発明の種々の実施形態の完全な理解を提供するために、以下の説明および図面において特定の具体的な詳細が述べられる。しかしながら、本発明の種々の実施形態を不必要に分かりにくくするのを避けるために、以下の開示において、コンピューティングおよびソフトウェア技術と関連することが多い特定の公知の詳細については述べられない。さらに、関連分野の当業者であれば、下記に記載される詳細の１つまたは複数がなくても本発明の他の実施形態を実施できることを理解するであろう。最後に、以下の開示においては、ステップおよびシーケンスを参照して種々の方法が説明されるが、このような説明は、本発明の実施形態を明確に実施するためのものであり、ステップおよび該ステップのシーケンスを本発明の実施に必要なものとみなすべきではない。

【００１６】
以下の詳細な説明は、一般に、上述のような本発明の概略に従っており、必要に応じて、本発明の種々の態様および実施形態の定義をさらに説明し、展開している。この目的のために、この詳細な説明は、まず、本発明に関連したソフトウェアおよび／またはハードウェア技術を実装するのに適した、図１のコンピューティング環境について述べる。現代のコンピューティング技術が、多数の別個の装置にわたって実施できることを強調するために、ネットワーク化されたコンピューティング環境が、基本的なコンピューティング環境の拡張として図２に示される。

【００１７】
次に、ハードウェア・セキュリティ・モジュール（ＨＳＭ）を利用するコンピューティング・プラットフォームの概要が、図３と関連して提供され、どのように測定値をＨＳＭにサブミットすることができるか、これらの測定値が正しい場合、どれを構成して鍵をシステム・リソースに戻すことができるかを説明する。図３に示されるＨＳＭは、当業者により容易に認識されるＨＳＭであるＴＰＭであることに注意されたい。また、ブート時またはその後に関与するソフトウェア・コンポーネントの更なる処理は、ＴＰＭによって保護される秘密の開示を条件として行うことができる。次に、ブート・プロセスにおけるソフトウェア・コンポーネントによるＴＰＭの使用が、図４に示される。図５は、図４のもののようなソフトウェア・コンポーネントによってＴＰＭを使用するための１つの一般的なパターンを示しており、そこでは、次のソフトウェア・コンポーネントの読み込みおよび実行は、次のコンポーネントの実行可能なコードのハッシュ値の確認を条件としている。

【００１８】
図５ａ、図６、図７、および図８においては、ブート・プロセスにおけるプラットフォーム構成レジスタ（ＰＣＲ）と呼ばれるＴＰＭレジスタの使用の態様が、より詳細に説明される。図５ａは、特定の組のブート・コンポーネントが所定位置にない限り、ブート・プロセスが進行できないことを保証するためのシステムおよび方法を示す。図６は、コンピュータをうまくブートすることが、例示的なコンポーネント即ちブート・マネージャをうまく暗号化し、測定することに結びついている、図５ａに示されるシステムおよび方法の一例を実証している。図７および図８は、ブート・プロセスがオペレーティング・システムをうまく起動し、これらのリソース（通常、１つまたは複数のディスク・パーティション上に常駐する）を必要としなくなった後、ブートのために必要とされるシステム・リソースへのアクセスを防止するための機構を示す。

【００１９】
例示的なコンピューティングおよびネットワーク化環境
図１のコンピューティング・システム環境１００は、適切なコンピューティング環境の一例にすぎず、本発明の使用範囲または機能に関して何らかの制限を示唆するように意図されるものではない。コンピューティング環境１００は、例示的なオペレーティング環境１００に示されるコンポーネントのいずれか１つまたは組み合わせに対して何らかの依存関係または要件を有するものとして解釈すべきでもない。

【００２０】

本発明は、多数の他の汎用または特殊用途コンピューティング・システム環境または構成と共に動作可能である。本発明と共に用いるのに適した周知のコンピューティング・システム、環境、および／または構成の例には、これらに限定されるものではないが、パーソナル・コンピュータ、サーバ・コンピュータ、ハンドヘルドまたはラップトップ型装置、マルチプロセッサ・システム、マイクロプロセッサ・ベースのシステム、セットトップボックス、プログラム可能な家庭用電化製品、ネットワークＰＣ、ミニコンピュータ、メインフレーム・コンピュータ、上記のシステムまたは装置のいずれかを含む分散型コンピューティング環境などが含まれる。

【００２１】

本発明は、コンピュータによって実行される、プログラム・モジュールのようなコンピュータ実行可能命令の一般的文脈に即して実装することができる。一般に、プログラム・モジュールは、特定のタスクを実行するかまたは特定の抽象データの型を実装するルーチン、プログラム、オブジェクト、コンポーネント、データ構造などを含む。本発明はまた、通信ネットワークを通してリンクされるリモート処理装置によってタスクが実行される分散型コンピューティング環境においても実施することができる。分散型コンピューティング環境において、プログラム・モジュールは、メモリ記憶装置を含む、ローカルおよびリモート・コンピュータ・ストレージ・メディアの両方に配置することができる。

【００２２】

図１を参照すると、本発明を実施するための例示的なシステムが、コンピュータ１２１の形態の汎用コンピューティング装置を含む。コンピュータ１２１のコンポーネントは、これらに限られるものではないが、処理ユニット１０１、システム・メモリ１０３、および、システム・メモリを含む種々のシステム・コンポーネントを処理ユニット１０１に結合させるシステム・バス１０２を含むことができる。システム・バス１０２は、メモリ・バスまたはメモリ・コントローラ、周辺バス、および種々のバス・アーキテクチャのいずれかを用いるローカル・バスを含むバス構造体のうちの幾つかのタイプのいずれかとすることができる。限定ではなく一例として、こうしたアーキテクチャは、業界標準アーキテクチャ（ＩＳＡ）バス、マイクロ・チャネル・アーキテクチャ（ＭＣＡ）バス、Ｅｎｈａｎｃｅｄ　ＩＳＡ（ＥＩＳＡ）バス、Ｖｉｄｅｏ　Ｅｌｅｃｔｒｏｎｉｃｓ　Ｓｔａｎｄａｒｄ　Ａｓｓｏｃｉａｔｉｏｎ（ＶＥＳＡ）ローカル・バス、およびメザニン・バスとしても知られているＰｅｒｉｐｈｅｒａｌ　Ｃｏｍｐｏｎｅｎｔ　Ｉｎｔｅｒｃｏｎｎｅｃｔ（ＰＳＩ）バスを含む。

【００２３】

ＨＳＭは図１に示されていないが、こうした装置は、本発明を実施するコンピュータの一部とすることができる。以下に図３を参照して説明されるように、図３は、コンピュータのコンポーネントと一体化されたＨＳＭ（図３の実施形態におけるＴＰＭ）を示す。代表的な環境においては、一定範囲のセキュリティ機能を提供するために、マザーボードに溶接されたハードウェアか、あるいはチップセットまたは図１のもののようなコンピュータの他のハードウェア・コンポーネントと一体化されたハードウェア・チップとすることができる。しかしながら、本明細書の目的上、ＨＳＭは、ハードウェアまたはソフトウェア内に実装することができ、本発明の動作のために必要な信頼できる機能、即ちＨＳＭにサブミットされる測定値の比較および検証、並びに暗号化されたメモリ・リソースへのアクセスのための鍵の開示できる機能ユニットとして広く定義されることを理解すべきである。ＴＰＭは、業界標準ＴＰＭのためのＴＣＧ（登録商標）仕様に説明されるような、一定範囲の他の機能を提供することもできる。

【００２４】

コンピュータ１２１は、一般に、種々のコンピュータ可読メディアを含む。コンピュータ可読メディアは、コンピュータ１２１によってアクセスすることができる何らかの使用可能なメディアとすることができ、揮発性メディアおよび不揮発性メディアの両方、並びに取り外し可能なメディアおよび取り外し不能なメディアの両方を含む。限定ではなく一例として、コンピュータ可読メディアは、コンピュータ・ストレージ・メディアおよび通

信メディアを含むことができる。コンピュータ・ストレージ・メディアは、コンピュータ可読命令、データ構造、プログラム・モジュール、または他のデータのような情報の格納のための何らかの方法または技術に実装される揮発性メディアおよび不揮発性メディア、並びに取り外し可能メディアおよび取り外し不能メディアの両方を含む。コンピュータ・ストレージ・メディアは、これらに限られるものではないが、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、フラッシュ・メモリ、または他のメモリ技術、ＣＤ－ＲＯＭ、デジタル多用途ディスク（ＤＶＤ）または他の光学ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶装置、あるいは所望の情報を格納するために用いることができ、コンピュータ１２１によってアクセスできる他の何らかのメディアを含む。通信メディアは、一般に、コンピュータ可読命令、データ構造、プログラム・モジュール、あるいは搬送波のような変調されたデータ信号または他の搬送機構における他のデータを具体化し、何らかの情報伝達メディアを含む。「変調されたデータ信号」という用語は、信号内の情報をエンコードするような方法で設定または変更された特性の１つまたは複数を有する信号を意味する。限定ではなく一例として、通信メディアは、有線ネットワークまたは直接有線接続のような有線メディア、音響、ＲＦ、赤外線、または他の無線メディアのような無線メディアを含む。上記のいずれかの組み合わせをも、コンピュータ可読メディアの範囲内に含ませるべきである。

【００２５】

システム・メモリ１０３は、読取り専用メモリ（ＲＯＭ）１０４およびランダム・アクセス・メモリ（ＲＡＭ）１０６のような揮発性および／または不揮発性メモリの形態のコンピュータ・ストレージ・メディアを含む。起動中などにコンピュータ１２１内の要素間の情報転送を助ける基本ルーチンを含む基本入出力システム１０５（ＢＩＯＳ）は、一般に、ＲＯＭ１０４内に格納されている。ＲＡＭ１０６は、一般に、処理ユニット１０１に直ちにアクセス可能な、および／または処理ユニット１０１によって現在動作しているデータおよび／またはプログラム・モジュールを含む。限定ではなく一例として、図１は、オペレーティング・システム１０７、アプリケーション・プログラム１０８、他のプログラム・モジュール１０９、およびプログラム・データ１１０を示す。

【００２６】

コンピュータ１２１はまた、他の取り外し可能／固定の、揮発性／不揮発性のコンピュータ・ストレージ・メディアを含むこともできる。ほんの一例として、図１は、固定の不揮発性磁気メディアとの間で読み取りまたは書き込みを行うハードディスク・ドライブ１１２と、取り外し可能な不揮発性磁気ディスク１１９との間で読み取りまたは書き込みを行う磁気ディスク・ドライブ１１８と、ＣＤ　ＲＯＭまたは他の光メディアのような取り外し可能な不揮発性光ディスク２５３との間で読み取りまたは書き込みを行う光ディスク・ドライブ１２０とを示す。例示的なオペレーティング環境に用いることができる他の取り外し可能／固定の、揮発性／不揮発性コンピュータ・ストレージ・メディアは、これらに限られるものではないが、磁気テープ・カセット、フラッシュメモリ・カード、デジタル多用途ディスク、デジタル・ビデオテープ、ソリッドステートＲＡＭ、ソリッドステートＲＯＭなどを含む。ハードディスク・ドライブ１１２は、一般に、インターフェース１１１のような固定のメモリ・インターフェースを通してシステム・バス１０２に接続され、磁気ディスク・ドライブ１１８および光ディスク・ドライブ１２０は、一般に、インターフェース１１７のような取り外し可能なメモリ・インターフェースを通してシステム・バス１０２に接続される。

【００２７】

上記に説明され、図１に示されるドライブおよびこれに関連したコンピュータ・ストレージ・メディアは、コンピュータ可読命令、データ構造、プログラム・モジュール、およびコンピュータ１２１のための他のデータを格納する。図１においては、例えば、ハードディスク・ドライブ１１２は、オペレーティング・システム１１３、アプリケーション・プログラム１１４、他のプログラム・モジュール１１５、およびプログラム・データ１１６を格納するように示されている。これらのコンポーネントは、オペレーティング・シス

テム１０７、アプリケーション・プログラム１０８、他のプログラム・モジュール１０９、およびプログラム・データ１１０と同じものにしても、異なるものにしてもよいことに注意されたい。オペレーティング・システム１１３、アプリケーション・プログラム１１４、他のプログラム・モジュール１１５、およびプログラム・データ１１６には、少なくともこれらが異なるコピーであることを示すために、ここでは異なる番号が与えられている。ユーザは、キーボード１２８、および一般にマウス、トラックボール、またはタッチパッドと呼ばれるポインティング・デバイス１２７のような入力装置を通して、コマンドおよび情報をコンピュータ１２１に入力することができる。他の入力装置（図示せず）は、マイク、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナなどを含むことができる。これらおよび他の入力装置は、多くの場合、システム・バスに結合されたユーザ入力インターフェース１２６を通して処理ユニット１０１に接続されるが、パラレルポート、ゲームポート、またはユニバーサル・シリアル・バス（ＵＳＢ）のような他のインターフェースおよびバス構造体によって接続することもできる。モニタ１３９または他のタイプの表示装置もまた、ビデオ・インターフェース２３２のようなインターフェースを介してシステム・バス１０２に接続される。モニタに加えて、コンピュータは、出力周辺インターフェース１２３を通して接続することができる、スピーカ１３８およびプリンタ１３７のような他の周辺出力装置を含むこともできる。

【００２８】

コンピュータ１２１は、リモート・コンピュータ１３１のような１つまたは複数のリモート・コンピュータへの論理接続を用いて、ネットワーク化環境において動作することができる。リモート・コンピュータ１３１は、パーソナル・コンピュータ、サーバ、ルータ、ネットワークＰＣ、ピア装置、または他の共通のネットワーク・ノードとすることができ、一般的には、コンピュータ１２１に関連して上述された要素の多くまたは全てを含むが、図１には、メモリ記憶装置１３２だけが示された。図１に示される論理接続は、ローカル・エリア・ネットワーク（ＬＡＮ）１３５および広域エリア・ネットワーク（ＷＡＮ）１３０を含むが、他のネットワークを含むこともできる。こうしたネットワーク環境は、事務所、企業全体にわたるコンピュータ・ネットワーク、イントラネット、およびインターネットにおいて当たり前のことである。

【００２９】

ＬＡＮネットワーク環境に用いられるとき、コンピュータ１２１は、ネットワーク・インターフェースまたはアダプタ１３４を通してＬＡＮ１３５に接続される。ＷＡＮネットワーク環境に用いられるとき、コンピュータ１２１は、一般に、モデム１２９、またはインターネットのようなＷＡＮ１３０にわたって通信を確立するための他の手段を含む。内蔵式にも、外付け式にもできるモデム１２９は、ユーザ入力インターフェース１２６または他の適切な機構を介して、システム・バス１０２に接続することができる。ネットワーク化環境においては、コンピュータ１２１に関連して示されたプログラム・モジュールまたはその一部を、リモートのメモリ記憶装置内に格納することができる。限定ではなく一例として、図１は、リモート・アプリケーション・プログラム１３３をメモリ装置１３２上に常駐するものとして示す。示されるネットワーク接続は例示的なものであり、コンピュータ間に通信リンクを確立する他の手段を用い得ることも理解されるであろう。

【００３０】

ここに説明される種々の技術は、ハードウェアまたはソフトウェア、または必要に応じてその両方の組み合わせと共に実装できることを理解すべきである。このように、本発明の方法および装置、並びにその特定の態様または部分は、フロッピィ・ディスケット、ＣＤ－ＲＯＭ、ハードドライブ、または他のいずれか機械可読ストレージ・メディアのような、有形メディアに具体化されたプログラム・コード（即ち命令）の形態を取ることができ、そこで、該プログラム・コードがコンピュータのようなマシンに読み込まれ、該マシンによって実行されたときに、該マシンが本発明を実施するための装置となる。プログラム可能なコンピュータ上でプログラム・コードを実行する場合には、コンピューティング装置は、一般に、プロセッサ、該プロセッサによって読取り可能なストレージ・メディア

（揮発性および不揮発性のメモリおよび／または記憶素子を含む）、少なくとも１つの入力装置、並びに少なくとも１つの出力装置を含む。１つまたは複数のプログラムが、例えばＡＰＩ、再使用可能な制御装置などを用いて、本発明に関連して説明されたプロセスを実行または利用することができる。こうしたプログラムは、コンピュータ・システムと通信するために、高レベルの手続き型言語またはオブジェクト指向言語で実装することが好ましい。しかしながら、所望であれば、プログラムをアセンブリ言語または機械語で実装することもできる。いずれの場合も、言語は、コンパイルされた言語または解釈された言語とすることができ、ハードウェア実装と組み合わせることもできる。

【００３１】

例示的な実施形態は、１つまたは複数の独立型コンピュータ・システムに即して本発明の利用に言及するが、本発明はそのように限定されるのではなく、寧ろネットワーク・コンピューティング環境または分散型コンピューティング環境のような何らかのコンピューティング環境と共に実施することができる。さらに、本発明は、複数の処理チップまたは装置内に、または複数の処理チップまたは装置にわたって実装することができ、同様に複数の装置にわたって保存を行うこともできる。こうした装置は、パーソナル・コンピュータ、ネットワーク・サーバ、手持ち式装置、スーパーコンピュータ、または自動車および航空機のような他のシステムに内蔵されたコンピュータを含むことができる。

【００３２】

図２において、例示的なネットワーク化コンピューティング環境が提供される。当業者であれば、ネットワークは、あらゆるコンピュータまたは他のクライアントまたはサーバ装置を接続することができ、あるいは分散型コンピューティング環境において接続できることを理解することができる。この点で、如何なる数の処理、メモリ、または記憶ユニット、および如何なる数のアプリケーションおよび同時に生じるプロセスをも有する、いずれのコンピュータ・システムまたは環境も、提供されるシステムおよび方法と共に用いるのに適していると考えられる。

【００３３】

分散型コンピューティングは、コンピューティング装置とシステムとの間の交換によって、コンピュータ・リソースおよびサービスを共有する。これらのリソースおよびサービスには、情報の交換、ファイルのキャッシュ保管およびディスク保管が含まれる。分散型コンピューティングは、ネットワーク接続性を利用し、クライアントが全体的な力を利用して、企業全体に利益をもたらすことを可能にする。この点で、種々の装置は、ここで説明されるプロセスに関係させ得るアプリケーション、オブジェクト、またはリソースを有することができる。

【００３４】

図２は、例示的なネットワーク化または分散型コンピューティング環境の概略図を提供する。この環境は、コンピューティング装置２７１、２７２、２７６および２７７、並びにオブジェクト２７３、２７４および２７５、並びにデータベース２７８を含む。これらのエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８の各々は、プログラム、方法、データ・ストア、プログラム可能な論理などを含むかまたはこれらを利用することができる。エンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８は、ＰＤＡ、音声／ビデオ装置、ＭＰ３プレイヤー、パーソナル・コンピュータなどのような同じまたは異なる装置の部分にわたることができる。各々のエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８は、通信ネットワーク２７０を介して、別のエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８と通信することができる。この点で、いずれのエンティティも、データベース２７８または他の記憶素子の保守および更新を担うことができる。

【００３５】

このネットワーク２７０は、これ自体が、サービスを図２のシステムに提供する他のコンピューティング・エンティティを含むことができ、これ自体が、多数の相互接続された

ネットワークを表すことができる。本発明の態様によると、各々のエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８は、他のエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８の１つまたは複数のサービスを要求するために、ＡＰＩ、または他のオブジェクト、ソフトウェア、ファームウェア、および／またはハードウェアを利用できる別個の関数型プログラム・モジュールを含むことができる。

【００３６】

２７５のようなオブジェクトは、別のコンピューティング装置２７６上でホストすることができることも理解できるであろう。このように、示される物理的環境は、接続された装置をコンピュータとして示しているが、こうした例証は単に例示的なものであり、物理的環境は、代替的に、ＰＤＡ、テレビ、ＭＰ３プレイヤー等のような種々のデジタル装置、およびインターフェース、のようなソフトウェア・オブジェクト、ＣＯＭオブジェクト等を含むものとして示すまたは説明することができる。

【００３７】

分散型コンピューティング環境をサポートする種々のシステム、コンポーネント、およびネットワーク構成が存在する。例えば、コンピューティング・システムは、有線または無線のシステム、ローカル・ネットワークまたは広域分散型ネットワークによって、互いに接続することができる。現在、多くのネットワークは、広域分散型コンピューティングにインフラストラクチャを提供し、多くの異なるネットワークを含むインターネットに結合されている。インターネットに結合されていてもいなくても、提供されたシステムおよび方法と共にこうした何らかのインフラストラクチャを用いることができる。

【００３８】

ネットワーク・インフラストラクチャは、クライアント／サーバ、ピア・ツー・ピア、またはハイブリッド型アーキテクチャのようなネットワーク・トポロジーのホストをイネーブルにする。「クライアント」とは、これが関係していない別のクラスまたはグループのサービスを使用するクラスまたはグループのメンバーである。コンピューティングにおいては、クライアントとは、別のプログラムによって提供されるサービスを要求するプロセス、即ち大ざっぱに言えば１組の命令またはタスクである。クライアント・プロセスは、他のプログラムまたはサービス自体についてのいずれの作業詳細も「知る」必要なく、要求されたサービスを利用する。クライアント／サーバ・アーキテクチャ、特にネットワーク化されたシステムにおいては、クライアントは、通常、例えばサーバのような別のコンピュータによって提供される共有ネットワーク・リソースにアクセスするコンピュータである。図２の例においては、状況によって、いずれのエンティティ２７１、２７２、２７３、２７４、２７５、２７６、２７７、および２７８も、クライアント、サーバ、またはその両方と見なすことができる。

【００３９】

サーバは、一概には言えないが、通常、インターネットのようなリモート・ネットワークまたはローカル・ネットワークにわたってアクセス可能なリモート・コンピュータ・システムである。クライアント・プロセスは、第１のコンピュータ・システムにおいてアクティブになることができ、サーバ・プロセスは、第２のコンピュータ・システムにおいてアクティブになることができ、通信メディア上で互いに通信し、これにより分散機能が与えられ、多数のクライアントがサーバの情報収集能力を利用することが可能になる。いずれのソフトウェア・オブジェクトをも、多数のコンピューティング装置またはオブジェクトにわたって分散させることができる。

【００４０】

クライアントおよびサーバは、プロトコル層によって与えられた機能を利用して、互いに通信する。例えば、ＨｙｐｅｒＴｅｘｔ　Ｔｒａｎｓｆｅｒ　Ｐｒｏｔｏｃｏｌ（ＨＴＴＰ）は、Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（WWW）即ち「ウェブ」と共に用いられる一般的なプロトコルである。一般に、インターネット・プロトコル（ＩＰ）アドレスのようなコンピュータ・ネットワーク・アドレス、またはＵｎｉｖｅｒｓａｌ　Ｒｅｓｏｕｒｓ

ｅ　Ｌｏｃａｔｏｒ（ＵＲＬ）のような他の参照を用いて、サーバ・コンピュータまたはクライアント・コンピュータを互いに識別することができる。ネットワーク・アドレスは、ＵＲＬアドレスと呼ぶことができる。通信メディアを通して通信を提供することができ、例えば、大容量の通信のために、ＴＣＰ／ＩＰ接続を介してクライアントおよびサーバを互いに結合させることができる。

【００４１】

図１の一般的なフレームワークに従って構築できる多様なコンピューティング環境、および図２のもののようなネットワーク環境のコンピューティングにおいて生じ得る更なる多様性に鑑みると、ここに提供されたシステムおよび方法を、多少なりとも特定のコンピューティング・アーキテクチャに制限されるものとして解釈することはできない。代わりに、本発明は、いずれの単一の実施形態に限定すべきでもなく、寧ろ、本発明の範囲は、添付の特許請求の範囲に従った広がりおよび範囲において解釈すべきである。

【００４２】

例示的なＴＰＭのセキュリティ保護されたブート・シーケンス

本発明の実施形態は、安全なブート・プロセスにおいてＴＰＭを使用する。ＴＰＭは、図３のコンピュータ・アーキテクチャとの関連で示される。本発明の実施形態に用いることが考えられるＴＰＭは、ＴＣＧ（登録商標）１．２準拠にすることができるが、ＰＣＲ内に配置されたもののような測定値を検証し、該測定値が正しい場合に秘密を開示するように、ＨＳＭを用いることができる。

【００４３】

この点で、図３は、図１のもののようなコンピュータの極めて一般化された図において、メモリ３０５へのアクセスを有するＣＰＵ３００を示す。ＣＰＵ３００は、幾つかのセキュリティ機能をＴＰＭ３０１に依存することができる。一般に、ＣＰＵ３００は、まず、ブート・プロセス内に含まれるデータの測定を行い、これらの測定値は、秘匿されたＰＣＲ値３０４に示されるように、ＴＰＭ３０１に安全に格納することができる。種々の実施形態において、本明細書の図面に示される種々のＰＣＲ値３０４および３０３は、実際には、ＴＣＧ（登録商標）１．２仕様に定義されるような代数式によって拡張された１つまたは複数の単一の格納先に格納できることに注意されたい。

【００４４】

秘密３０２は、ＴＰＭ３０１内の特定のＰＣＲ値３０４に秘匿することができる。ＴＰＭ３０１から秘密３０２を取り出すために、正しいＰＣＲ値をＰＣＲ３０３に入力しなければならない。これらの正しい値は、ＴＰＭ３０１内に秘匿されたＰＣＲ値３０４を得るために測定されたものと同じデータを測定することによって得ることができる。多数の秘密３０２を種々のＰＣＲ３０４内に秘匿することができる。例えば、第１の秘密Ａを取り出すために、正しい値がＰＣＲ［１］、ＰＣＲ［２］、およびＰＣＲ［３］内に格納されていることが必要である。第２の秘密Ｂを得るために、ＰＣＲ［４］内に第４の正しい値が必要である。

【００４５】

ＴＰＭ３０１内に秘匿された測定値の値と合致しない測定値がＰＣＲ３０３内に置かれた場合には、ＴＰＭ３０１に秘密３０２の開示が要求されたとき、その開示に失敗する。正しい測定値がＰＣＲ３０３内に置かれた場合には、秘密３０２の開示が要求されたとき、ＴＰＭ３０１を信頼し、秘密３０２を開示することができる。したがって、この用途についての「正しい」測定値または正しい値は、秘密３０２が秘匿される測定値であり、これによりＴＰＭ３０１による秘密３０２の開示が可能になる。幾つかの実施形態においては、正しい測定値は、悪質なコードの測定値とすることができることに注意されたい。これは、例えば、ＴＰＭ３０１内に秘匿された最初の測定値３０４が壊れていた場合である。

【００４６】

特定の測定値に秘匿される秘密は、如何なるデータとすることもできる。一般に、秘密３０２は、解読鍵および／またはバイナリ・ラージ・オブジェクト（ＢＬＯＢ）の形態を

取る。一般に、鍵は、データを解読するのに用いることができる情報を提供する。秘匿されたＢＬＯＢは、鍵、および有用な他のデータを含むことができる。この点で、当業者には理解されるように、鍵をＢＬＯＢと置き換えること、およびその逆に置き換えることによって、ここで説明される種々の技術の等価物を構築することができる。したがって、ＣＰＵ３００が、正しい測定値を３０３のＰＣＲにサブミットする場合、鍵のような対応する秘密３０２が要求されたとき、ＴＰＭ３０１は、該秘密３０２を開示することができる。次に、３０２からの鍵を用いて、ＣＰＵ３００によってアクセス可能なメモリ３０５の一部を解読することができる。本発明の実施形態においては、図３に示されるように、ＴＰＭ３０１は、３つの秘密Ａ、Ｂ、およびＣへのアクセスを許可するように構成することができる。必要とされる種々のＰＣＲ値に秘密３０２を秘匿することができ、よって、特定の測定が行われた後にのみアクセス可能になる。これらの３つの鍵、即ち３つの秘密は、ここでは、第１のものはブート・アクセス専用秘密、第２のものはボリューム制限秘密（ｖｏｌｕｍｅ－ｂｏｕｎｄ　ｓｅｃｒｅｔ）、第３のものはパスワード秘密と呼ばれる。

【００４７】

ＴＰＭに関係した活動は、ログ３０７内に格納することができる。幾つかの実施形態において、ログ３０７は、コンピュータのＢＩＯＳによって維持することができる。いずれかの他のプロセスが、ログ３０７の維持を担当することもできる。したがって、ソフトウェア・コンポーネント３０８または他のデータ３０９のようなデータが測定されてＰＣＲ３０３内に入れられる場合には、ログ３０７において測定されたデータを識別することができる。秘密開示要求がなされた場合には、ログ３０７において要求イベントを識別することができる。これらは、ＴＰＭに関係した活動をログ３０７内に格納する２つの例にすぎず、ログ３０７は、他の広範囲のイベントおよび活動の記録を含むことができる。

【００４８】

一般に、ＴＰＭ３０１は、Ｓｔａｔｉｃ　Ｒｏｏｔ　ｏｆ　Ｔｒｕｓｔ　Ｍｅａｓｕｒｅｍｅｎｔ（ＳＲＴＭ）と共に動作し、信頼できる測定を行い、それらをＴＰＭ３０１にサブミットする。しかしながら、ＤＲＴＭ　Ｎｅｘｕｓの使用などを通して安全な測定を行うための他のプロセスも利用可能である。本発明の実施形態は、こうした方法でＳＲＴＭのような信頼できる測定プロセスを用いることができ、この点で、ＳＲＴＭは、初期ディスク・ベースのブート・コードを測定するために、ここで説明される（プロセスおよびＲＴＭとも呼ばれる）種々のソフトウェア・コンポーネントによって用いられるＢＩＯＳ標準ＳＲＴＭとすることができる。システムはまた、オペレーティング・システムをブートする初期段階を測定できるように、ＳＲＴＭを拡張し、オペレーティング・システムをブートする初期段階に含まれる他のコードおよび重要なデータを測定することもできる。ＰＣＲ３０３が、いずれから得られた値をも含み得ることに注意されたい。この値は、ソフトウェア・コンポーネント３０８または他のデータ３０９のようなデータの測定値とすることができる。本発明は、データの測定値、またはＰＣＲ３０３内に置かれた他の値の何らかの排他的な組み合わせに限定されるものではない。

【００４９】

ＴＰＭのセキュリティ保護されたブート・プロセスにおいて、図３に表示される構成を用いて、図４に示される例示的なソフトウェア・コンポーネントを測定し、その測定値をＰＣＲ３０３内に格納することができる。本発明の実施形態によって測定されるように選択することができる、図４に示されるブート・コンポーネント、具体的にはディスク・ベースのコード・コンポーネントは、滅多に変更しないことが知られており、簡単に攻撃を受けやすい。したがって、ここで説明されるような条件付きの保守および更新プロセスによる以外は、特定のブート・コンポーネントが変更されないままにすることにより、データ・セキュリティを著しく向上させるのに支払う価格が比較的少なく済む。

【００５０】

図４を参照すると、コンピュータに例示的なブート・プロセスを提供するための、一連のソフトウェア・コンポーネント４００－４０７が示される。本発明は、示された特定の

コンポーネントにも、コンポーネントのシーケンスにも制限されるものではない。示されたコンポーネントは、連続的に読み込むことができ、Ｃｏｒｅ　Ｒｏｏｔ　ｏｆ　Ｔｒｕｓｔ　ｆｏｒ　Ｍｅａｓｕｒｅｍｅｎｔ（ＣＲＴＭ）４００で開始し、ここでは単一のソフトウェア・コンポーネント４０７として一般化されたオペレーティング・システム（ＯＳ）４０７のコンポーネントで終了する。コンポーネントを読み込むことにより、コンポーネントが、メモリおよびＣＰＵのようなコンピュータのリソースにアクセスできるようになる、コンポーネントの命令をＣＰＵによって実行することができる。図４のコンポーネントが悪質なものであるかまたは壊れている場合には、コンポーネントが読み込まれると、これをセキュリティ手段の回避のために用いることができる。したがって、本発明に適合するコンピュータをブートするためのプロセスは、コンポーネントの実行を可能にする前に、コンポーネントまたは複数のコンポーネントを測定し、１つまたは複数のＰＣＲ３０３内に入れるステップを含む。ＴＰＭ内に秘匿される、信頼できる測定値３０４の組に秘匿される秘密３０２次第で、ブートを成功させることができる。しかしながら、本発明はまた、悪質なコードの測定値をＴＰＭ内に秘匿する可能性もあることに注意されたい。秘匿時に悪質なコードが動作している場合、ブートのためにこれらの測定値を必要とする。理想的には、秘密は、信頼できるコードの測定値３０４に秘匿される。ＰＣＲ３０３内に置かれた測定値が正しい場合には、３０２からの秘密を開示することができ、マシンが安全なブート状態で進行することが可能になる。秘密３０２を開示するプロセスが、図５に示される。

【００５１】

幾つかの使用シナリオにおいて、マシンの所有者は、自分達がそのマシンの構成を「ロック」したいと決定し、以前に検証されたものに加えて、ＲＯＭベースのコードがこれまでに実行されていないことを保証することができる。この場合、マシンの所有者は、使用されることになる付加的なＰＣＲ３０２を選択することによって、検証プロセスに含まれるより多くのソフトウェア・コンポーネント（ＢＩＯＳ、オプションＲＯＭ）を構成することができる。所有者は、自分達が、ＴＰＭ３０１によって検証されるマインのパスワードをさらに利用したいと決定することもできる。このことは、セキュリティが、本発明の標準的な実施形態において一般に提供できる上記のものを拡張することを可能にし、ユーザが、使いやすさと対照してマシンのセキュリティを評価することを可能にする。

【００５２】

図５は、次のソフトウェア・コンポーネントを読み取る前に、次のソフトウェア・コンポーネントの完全性を保証する、ＴＰＭを用いるための技術を示す。図４のコンポーネントのような一連のコンポーネント内に適切な命令を配置することによって、図５のステップを実行することができる。この点で、図５のプロセスは、ＣＲＴＭコンポーネントの実行で開始することができる５０８。ＣＲＴＭおよび図４の他のコンポーネントの一部または全てのようなコンポーネントは、別のコンポーネントを測定し、その結果を図３の３０３からのようなＰＣＲ内に置くための命令を支持することができる。そうした命令を支持するコンポーネントは、Ｒｏｏｔ　ｏｆ　Ｔｒｕｓｔ　ｆｏｒ　Ｍｅａｓｕｒｅｍｅｎｔ（ＲＴＭ）と呼ばれることもあり、上述のようなＳＲＴＭを使用するための命令を含むことができる。したがって、ブート・ブロックがブート・マネージャを測定する場合には、該ブート・ブロックは、該ブート・マネージャのためのＲＴＭとして働く。

【００５３】

ＲＴＭは、次のコンポーネントをメモリに読み込み５００、次に、次のコンポーネントの測定を行い５０１、その測定値をＰＣＲに加えることができる５０２。ＲＴＭが、鍵またはＴＰＭからＢＬＯＢのような秘密を必要とする場合には５０３、こうした秘密を要求することができ、ＴＰＭは、その秘密にアクセスするのに必要とされる全てのＰＣＲについて正しいＰＣＲ値が読み込まれた場合にのみ、要求された秘密を開示する。したがって、ＴＰＭから取り出された情報に基づいて、秘密の開示を試みることができる５０４。ステップ５０５において開示が成功した場合には、次のコンポーネントと、下記に説明される他の動作を読み取るステップを含む付加的なステップを取ることができる。開示が成功

しなかった場合には、ＰＣＲ内の値が正しくない可能性があり、よって、実行コードが壊れている可能性がある。ステップ５０７においてエラーが生じることがあり、例えば、コンピュータのディスク上のデータの暗号化を用い、解読鍵を出すのをやめることによって、コンピュータ上に格納された機密情報にアクセスできなくなることを保証するために、適切な手段を取ることができる。あるいはまた、例えば、システムを正しいＰＣＲ値を生成する状態に復旧させること、またはユーザが、図３からの値３０２内の秘匿された新しいＰＣＲ値を許可するのを認証することによって、システムを維持するためのプロセスを実施することができる。こうしたプロセスが、以下に説明される。示されるように、ステップ５０３において秘密が必要とされない場合には、如何なる秘密も必要とすることなく、次のコンポーネントを読み込むことができる。

【００５４】

図４および図５を一緒に参照し、本発明のシステムおよび方法に適合する例示的なブート・プロセスを示すことができる。最初に、基本入出力システム（ＢＩＯＳ）４０１を読み込み、ＣＲＴＭ４００を読み込み、測定することができる。この測定は、例えば、ＢＩＯＳ上でハッシュを行い、次にハッシュの測定値をＰＣＲにサブミットすることによって行うことができる。次に、ＢＩＯＳの実行が可能になり、ＢＩＯＳは、マスター・ブート・レコード（ＭＢＲ）４０２のためのＲＴＭとして働くことができる。ＭＢＲを測定してＰＣＲに格納することができ、次に、ＭＢＲ４０２の実行が許可される。ＭＢＲは、ブート・セクタ・コンポーネント４０３を測定することができ、次に該ブート・セクタ・コンポーネント４０３の実行が許可される。読み込み、測定し、ＰＣＲに書き込み、次のコンポーネントに移行するというこのパターンは、各コンポーネント４０４、４０５、４０６、および４０７、並びにオペレーティング・システム４０７内のいずれかのコンポーネントによって、必要に応じて繰り返すことができる。図５に示されるように、本発明の付加的な態様は、このプロセスの変形を含み、この変形は、途中のどの時点でも秘密を要求し、用いることができる。この点で、本発明の実施形態は、次のコンポーネントに移行する前に実行できる付加的なステップを通して、高度のセキュリティを提供する。これらの付加的なステップは、正しいＰＣＲ値を測定することによって取得される秘密を条件としてマシンのブートを成功させ、これにより、ブートに用いられるデータの一部または全てが、秘密が秘匿されたときのままであることが保証される。付加的なステップはまた、ブート中に必要であるがその後は必要でないブート後のリソースへのアクセスを防止する働きもできる。

【００５５】

次のコンポーネントに移行する前に、コンポーネント４００－４０６の一部に、解読鍵、ＢＬＯＢ、または他の解読鍵などへのアクセスを可能にする保護情報とすることができる秘密を取り出すことを要求することによって、図４および図５の基本的プロセスを強化することができる。このように、本発明の実施形態は、ブート・プロセス内の戦略的時点において１つまたは複数の秘密にアクセスするときに、オペレーティング・システムによる有用な動作の性能を調整することができる。測定されるコード・モジュール４０１－４０６（ここではコンポーネントおよび／またはソフトウェア・プロセスとも呼ばれる）のいずれかが変更されたことが発見された場合には、重要な秘密を伏せることができる。伏せることができる秘密の例は、「ＳＹＳＫＥＹ」（サービスによって用いられるパスワードのような局所的な秘密を解読するために、ＬＳＡＳＳによって用いられる）、コンピュータのハードドライブまたはディスク・パーティション上に格納される実質的に全てを解読するためのボリューム暗号化鍵、およびＥＦＳのような高レベルのシステム保護によって要求される秘密である。次に、高レベルの保護は、ＳＲＴＭより汎用性のある方法で、カタログを用いて検証することができる。

【００５６】

マシンうまくブートできる状態にマシンを修理するために、ここに説明される安全なブート・プロセスに加えて、以下に説明されるシステムおよび方法を実施することができる。

【００５７】
例示的な付加的なブート保護技術
次のコンポーネントに移行する前に次のコンポーネントを測定するように、複数のソフトウェア・コンポーネントを構成することができる、図４および図５を参照して理解できるもののようなブート・シーケンスにおいて、コンピュータ上に格納されたデータのセキュリティをさらに高める幾つかの付加的な予防措置を取ることができる。これらの付加的な予防措置が、このセクションの主題である。ここに説明される予防措置のいずれかまたは全てを本発明の実施形態に組み込むことができる。１つの好ましい実施形態においては、以下に説明されるように、ここに説明される予防措置の全てが用いられる。しかしながら、本発明はこうした実施に限定されるものでない。
【００５８】
最初に図５ａを参照すると、こうしたプラットフォームの完全性において秘密の公開を調整することによって、コンピュータのブートを、オペレーティング・システムに先行するコンポーネントの完全性に結びつけることができる。最初に、図５ａの概念的概観が与えられ、次に、図５ａのより詳細な説明が与えられる。
【００５９】
最初に、ブート・マネージャのようなソフトウェア・コンポーネントの公に知られたハッシュ値を用いて、ＰＣＲを拡張することができる。このことは、開示可能なブート秘密をもたらすが、これは、全ての先行するソフトウェア・コンポーネントが信頼できる場合に満足の行くものである。ブート秘密を開示できる場合、全ての先行コンポーネントを信頼できる。この際に、例えばブート・マネージャのようなソフトウェア・コンポーネントの状態は知られていない。
【００６０】
次に、ブート秘密を解読することができ、ボリューム対称鍵を用いてオン・ザ・フライ式にシステム・パーティションを解読し、ブート・マネージャをメモリに読み込むことができる。
【００６１】
第３に、これから解読される、メモリ内のブート・マネージャのハッシュ値を公知のハッシュ値と照合することによって、認証前ステップを統合することができる。ハッシュ値が合致する場合には、ブートは正常に進行することができる。ハッシュ値が正しくない場合には、ＰＣＲを無効にすることができる。ハッシュ値が正しいことを確認する、少なくとも次の方法が存在する。
ａ．ブート・マネージャのハッシュ値を公に知られているハッシュ値と照合する。システムがブート秘密を開示できる場合には、公に知られたハッシュ値が有効であることが分かるので、含蓄的に、ブート・マネージャのハッシュ値が、ブートｂｌｏｂを開示するのに用いられるハッシュ値と合致する場合、該ブート・マネージャのハッシュ値が有効であることが分かる。
ｂ．ブート・マネージャのハッシュ値を秘匿された秘密に格納されたハッシュ値と照合する。
ｃ．公に知られたブート・マネージャのハッシュ値を用いて異なるＰＣＲを拡張し、２つのハッシュ値を比較する。
【００６２】
ここで図５ａをより詳細に参照すると、ステップ５５０において、現在のコンポーネントまたはＲＴＭが実行中である。現在のＲＴＭは、次のソフトウェア・コンポーネントに移行するために、次のステップを実行することができる。予想される次のコンポーネントの測定値を、例えばＰＣＲ［ａ］のようなＰＣＲ内に読み込むことができる５５１。次に、ＲＴＭコンポーネントは、秘密を取り出そうと試みることができる５５２。ＰＣＲ［ａ］が正しい値と共に読み込まれなかった場合には、現在のＲＴＭは有効でなく、秘密へのアクセスを拒否することができ、これにより図５を参照して説明されたような通常のブートが阻止される。秘密を用いて、次のコンポーネントを解読することができる５５３。次

のコンポーネントが解読されるので、将来の攻撃者が、次のコンポーネントを分解工学し、修正して、予期しない方法で実行することは不可能である。解読された次のコンポーネントを測定し５５４、その測定値を、ＰＣＲ［ｂ］のようなＰＣＲ内に置くことができる５５５。ＲＴＭは、次に、ＰＣＲ［ａ］および［ｂ］の値を比較することができる５５６。これらが合致する場合には、次のコンポーネントとすることができる次のコンポーネントに移行することができる５５８。これらが合致しない場合には、例えば、メモリの幾つかの所定部分を測定して、これらのＰＣＲ内に入れることによって、ターミナル値においてＰＣＲ［ａ］および［ｂ］に上限を設けることができ、通常のブートを中止することができる５５９。

【００６３】

図６を参照すると、示されるフローチャートは、図５ａに導入されたシステムおよび方法の実施形態を示し、ここでは、先行するコンポーネントの動作がうまくいった時に重要なブート・コンポーネントへのアクセスおよび該重要なブート・コンポーネントの完全性を調整するための、システムおよび方法を実施する多数のステップが示される。図６に用いられる例示的なブート・コンポーネントはブート・マネージャであるが、いずれのコンポーネントも図６に実証される技術の主題となり得る。本説明のために、図６のステップは、図４に照らして理解することができる。複数のソフトウェア・コンポーネントを、連続的な方法で実行することができ、これらの１つまたは複数は、次のコンポーネントに移行する前に次のコンポーネントを測定することができる。

【００６４】

この文脈において、ステップ６１２に示されるように、ブート・プロセスのある時点において、ブート・セクタのような第１のコンポーネントを読み込むことができる。次に、図５ａに述べられる技術に従って、ブート・セクタを測定してＰＣＲ内に格納することができる。ステップ６１１に用いられる例示的なＰＣＲはＰＣＲ［８］であるが、本発明は、いずれか特定のＰＣＲに限定されるものではない。次に、コンピュータは、ブート・セクタ６１２の実行に移行することができる。ここで、ブート・セクタは、ブート・ブロックのためのＲＴＭとして働くことができ、その際、該ブート・ブロックを測定してＰＣＲ［９］内に格納することができる６０８。次に、コンピュータは、ブート・ブロックの実行に移行することができる６００。ここで、ブート・ブロックは、ブート・マネージャのためのＲＴＭとして働き、次のステップにおける付加的なセキュリティ手段を実行することができる。

【００６５】

このように、ブート・ブロックは、ブート・マネージャの予想される測定値をＰＣＲに読み込むことができる６０１。図６に用いられる例示的なＰＣＲは、ＰＣＲ１０である。ＰＣＲ［８］、［９］、［１０］に読み込まれる値、および制御するように構成された前のまたは次のＰＣＲが正しい場合、こうした秘密がブート・ブロックによって要求されたとき、ＴＰＭは、秘密へのアクセスを許可することができる。この秘密は、ブート・マネージャが格納されているハードディスクの一部のようなメモリの一部を解読するための解読鍵とすることができる。ステップ６０２に示されるように、この鍵は、ブート・ブロック・コンポーネントによって取り出すこともできる。ブート・マネージャの正しい予想測定値を生成するためにブート・ブロックを要求することによって、第１の層のセキュリティが与えられ、正しくない値が与えられた場合には、ＴＰＭは、ブート・マネージャを解読するのに必要とされる鍵へのアクセスを拒否することができる。

【００６６】

正しい予想値が与えられた場合には、暗号化鍵を取り出し、次に、ステップ６０３において、該暗号化鍵を用いて、ブート・マネージャ・コンポーネントを解読することができる。次に、ブート・ブロックは、ブート・マネージャを解読するのに用いられる「ブート・アクセス専用」鍵を恒久的に廃棄するように構成することもできる６０４。ブート・マネージャまたは次に読み込まれるコンポーネントが壊れていた場合、鍵にアクセスすることができなくなり、よってアクセスできるデータが厳しく制限されるので、ブート・マネ

ージャを読み込む前にブート・アクセス専用鍵を廃棄することによって、セキュリティの層を付加する。このことは、本発明の種々の実施形態について考慮されるように、コンピュータのハードディスクが殆ど完全に暗号化されているときに、特に言えることである。

【００６７】

次に、コンポーネントのハッシュ値の計算のような、ブート・マネージャ・コンポーネントを測定することができる６０５。測定値は、ＰＣＲ１３のような別のＰＣＲ内に格納することができる６０６。ステップ６０７に示されるように、ＰＣＲ１０およびＰＣＲ１３内に格納された値を比較し、これらが合致するかどうかを判断することができる。合致しない場合には、ブート・マネージャが変更され、壊れたコードまたは悪質のコードを含む可能性があるという結論に達することができる。まだブート・マネージャ・コンポーネントの実行に移行されておらず、よって、まだ如何なる危害も加えることができないことを思い出されたい。ブート・マネージャが壊れている場合には、ブート・ブロックによって、適切なセキュリティ手段を取ることができる。したがって、コンピュータのブートは、ブート・マネージャのような重要なソフトウェア・コンポーネントの解読および測定の成功を条件として行うことができる。

【００６８】

図７および図８を参照すると、後でコンピューティング・リソースを制御するプロセスからのブート中に用いられる秘密を秘匿するのに用いることができる、例示的なシステムおよび方法が示されている。図７および図８に示されるプロセスは、コンピュータ・ハード・ドライブ上に多数ディスク・パーティションが現存する状況において特に有用であるが、種々の設定において有用であることが理解される他の利点も有している。図７および図８に示されるプロセスの１つの利点は、単一のパーティションに対して、ソフトウェア・コンポーネントによってアクセスを制限するために、これらのプロセスを用いることができる点である。ブートの初期段階のコンポーネントは、全てのディスク・パーティションへのアクセスを要求することが多いが、後の段階およびブート後のコンポーネントは、単一のパーティションに制限することができる。図７および図８は、こうした制限を保証するための例示的なシステムおよび方法を示す。

【００６９】

図７は、図８に示されるプロセスのための設定を提供する。図面の左側には、パーティションＡ７００、パーティションＢ７０２、およびパーティションＣ７０４を含む、複数のディスク・パーティションが示されている。当業者には理解されるように、一般に、７０１、７０３、および７０５のような予約されたパーティション内に格納される、ブートの初期段階に必要とされる情報を除いて、各々のパーティションを完全に暗号化することができる。図４を参照して説明されるように連続的に読み込むことができるブート・ブロック７０６、ブート・マネージャ７０７、およびオペレーティング・システム（ＯＳ）ローダー７０８を含むソフトウェア・コンポーネントが、図７の下側に沿ってある。ＰＣＲｘ７０９、ＰＣＲｙ７１０、時間１のＰＣＲｚ７１１、および時間２のＰＣＲｚ７１２を含む、複数のＰＣＲが、図７の中央に示されている。ＰＣＲは、一般に、文字ではなく番号によって識別されるが、幾つかの実施形態は図８に説明されるＰＣＲを使用できるとはいえ、本発明が、使用される特定のＰＣＲに制限されないことを強調するために、ここでは文字が用いられている。図７のＰＣＲは、図５を参照して説明される機能を果たし、（値を内部に置くことができ）、ＴＰＭ７１３を信頼して、この値が正しいかどうかを示し、および／または正しい値が入力されたときに秘密へのアクセスを許可することができる。

【００７０】

図８に反映される実施形態のより詳細な説明に先立ち、図７を参照して一般的な概念を公式化することができる。ＴＰＭ７１３を通して、鍵またはＢＬＯＢ７１４のようなブート専用秘密にアクセスを獲得するために、時点１におけるＰＣＲｚの値のような１つまたは複数のＰＣＲの第１の値７１１を必要とする。コンピュータのブートの初期段階において要求されるように、ブート専用鍵またはＢＬＯＢ７１４は、複数のパーティションから

の情報を解読するのに有用である。ボリューム制限鍵（ｖｏｌｕｍｅ－ｂｏｕｎｄ　ｋｅｙ）またはＢＬＯＢ７１５へのアクセスを獲得するために、時点２におけるＰＣＲｚ７１２のような１つまたは複数の第２のＰＣＲ値を必要とする。ボリューム制限鍵またはＢＬＯＢは、パーティションＡ７００からのデータを解読するためだけといった、パーティションのサブセットに対してのみ有用である。したがって、それぞれ異なる時間に同じＰＣＲの異なる値を用い、これらの多数の値の適切な鍵への鍵またはＢＬＯＢのアクセスを調整することによって、下流側のソフトウェア・コンポーネントが、ブート・コンポーネントに使用可能な情報にアクセスするのを阻止することができる。ブートが適正に行われるためには、ボリューム制限鍵またはＢＬＯＢ７１５にアクセスしなければならず、これは、ブート専用鍵またはＢＬＯＢ７１４がもはやアクセス可能でなくなることを保証する。このシステムの付加的な利点が、特に図６に示されるシステムおよび方法と組み合わせて、当業者には明らかであろう。

【００７１】

図８を参照すると、図７に図式化されるようなシステムを実装するための種々の実施形態が示されている。したがって、ステップ８００において、ブート・マネージャ・コンポーネントを読み込むことができる。図６の技術を組み込むシステムにおいては、そこに示されるプロセスに従って、ブート・マネージャを読み込むことができる。例えば、ステップ８０１において、ブート・マネージャのハッシュ値を測定してＰＣＲ１０内に入れることができる。次に、ＴＰＭの使用には一般的なことだが、全ての以前の測定値だけでなく、例えばＰＣＲ［１１］および［１２］であり、まだ測定値と共に読み込まれていず、よって典型的にはゼロである初期値を保持するＰＣＲ［ｙ］および［ｚ］の値にも基づいて、ブート・アクセス専用鍵をＴＰＭから取り出すことができる。したがって、ＰＣＲ［ｙ］および［ｚ］の初期値に基づいて、ステップ８０２において秘密を取り出すことができる。

【００７２】

さらに図８を参照すると、ステップ８０３およびステップ８０４において、ＯＳローダーは、メモリ内に読み込み、ブート・マネージャによって測定することができる。ＯＳローダーのハッシュ値をＰＣＲ［ｙ］内に置くことができる８０５。このＰＣＲ［ｙ］への変更は、ブート・アクセス専用秘密への将来のアクセスを事実上取り消すことになり、よって、秘密がブート・マネージャによって廃棄される場合、秘密は、下流側のコンポーネントに失われることに注意されたい。次に、ＰＣＲ［ｙ］を、ブート・アクセス専用秘密内に格納されている値と比較することができる８０６。例えば、ブート・アクセス専用秘密がＢＬＯＢである場合には、ＰＣＲ値を該ＢＬＯＢと共に格納することができる。比較がうまくいった場合には、ブート・アクセス専用ＢＬＯＢからボリューム制限鍵を抽出することができる８０７。ボリューム制限鍵を測定してＰＣＲ［ｚ］内に入れることができる８０８。ＰＣＲ［ｚ］を通して、新しいＰＣＲ値に基づいたボリューム制限秘密へのアクセスを許可するように、ＴＰＭを構成することができる８０９。したがって、ステップ８０９においてボリューム制限ＢＬＯＢを獲得することにより、ブート・アクセス専用ＢＬＯＢのアクセスしにくさを調整することができる。この技術を利用する本発明の実施において、全ての次のプロセスを、ボリューム制限鍵またはＢＬＯＢに関連付けられたパーティションのサブセットに、事実上制限することができる。

【００７３】

システム・データを保護するための例示的なブート検証プロセス

本発明の実施形態は、ユーザ・インターフェース（ＵＩ）を通して、ユーザのコマンドで動作し、構成することができるブート検証プロセスを提供することができる。したがって、コントロールパネル・アプレットのようなプログラムを用いて、ユーザが本発明によるブート保護プロセスの動作をイネーブルにできるＵＩを使用可能にすることができる。マシンのユーザがそのマシンのＴＰＭの所有権を取得していなかった場合には、ＵＩは、まず所有権を取得するかまたはキャンセルするかの選択肢を提示することができる。特定のブート・パーティションの選択をユーザに要求する、同様の選択肢を提示することもで

きる。保護されたブートが、Ｎｅｗ Ｔｅｃｈｎｏｌｏｇｙ Ｆｉｌｅ Ｓｙｓｔｅｍ（ＮＴＦＳ）のような特定のファイル・システムのみで動作するように構成される場合には、ユーザは、そのファイル・システムを利用するブート・パーティションの選択を要求することができる。

【００７４】

保護されたブートがＵＩによってイネーブルにされると、自動プロセスは、セキュリティ保護されるべきトップレベルの秘密が、可能であれば再生成され、次に、秘密の開示に必要とされる予想ＰＣＲレジスタ値に秘匿されることを保証できる。好ましい実施形態は、この操作のために、ＰＣＲ［４］、ＰＣＲ［８］から場合によってはＰＣＲ［１５］までを利用することができる。パスワードに開示操作を代行させ、これを公に見える場所に格納することができる。したがって、選択されたパスワードは、秘匿操作に使用されたものと異なるものにすることができる。この操作をサポートするために、ＴＣＧ（登録商標）１．２ＴＰＭが好ましい。より高いセキュリティを提供できるこのプロセスの変形により、より多くのＰＣＲを指定することが可能になり、マシンの所有者がア開示用パスワードを指定し、これをブート・プロセスの初期に入力することが可能になる。

【００７５】

従来のＰＣまたはＡＴコンピュータ（ＰＣＡＴ）システム、即ち従来技術のＢＩＯＳを用いるｘ８６ベース・システムにおいては、ＭＢＲブート・セクタ、ＮＴＦＳブート・セクタ、ＮＴＦＳブート・ブロック、およびブート・マネージャを用いて、予想されるＰＣＲ値を判断することができる。予想されるＰＣＲ値についてのさらなる詳細が、例示的なブート・シーケンスと併せて以下に説明される。拡張ファームウェア・インターフェース（ＥＦＩ）システムについて、ＥＦＩＵシステム・パーティション（ＥＳＰ）内の関連ファイルが測定される。ブート・ボリューム暗号化を含む本発明の変形について、ＮＴＦＳブート・ブロックを含む、そこまでのブートの初期部分について、ディスク暗号化鍵をＰＣＲに秘匿することができる。

【００７６】

リカバリー・シナリオを助けるために、ＣＤＲＯＭを介するリカバリー、およびこうしたパーティションが存在する場合に特定のリカバリー・パーティションを介するリカバリー、および取り外し可能メディアおよび／またはパスワードのような第２の認証方法を介するリカバリーを含む、上記の秘密の他のコピーをブートするように秘匿することができる。

【００７７】

ＰＣＡＴシステムのための例示的なブート・プロセスが以下に提供される。ここに表示されるプロセスもまた、図８および図９を参照して理解することができる。

・ＴＣＧ（登録商標）１．２仕様によって要求されるように、ＢＩＯＳを測定してＰＣＲ［０］内に入れるのを担う、ＲＯＭの読取り専用部分を実行することができる。

・ＢＩＯＳ構成パラメータを測定してＰＣＲ［１］内に入れる。

・オプションＲＯＭを測定してＰＣＲ［２］内に入れる。

・オプションＲＯＭパラメータを測定してＰＣＲ［３］内に入れる。

・ＭＢＲを測定してＰＣＲ［４］内に入れる。

・パーティション・テーブルを測定してＰＣＲ［５］内に入れる。

・ＭＢＲを測定した後、ＢＩＯＳが実行をＭＢＲに転送する。

・ＭＢＲが、アクティブ・パーティションのＮＴＦＳブート・セクタを読み込み、これを測定してＰＣＲ［８］内に入れる。次に、ＭＢＲは、実行をこのブート・セクタ転送する。

・ブート・セクタは、ブート・ブロックをメモリ内に読み込む（一般に、８Ｋ）。ブート・ブロックを測定してＰＣＲ［９］内に入れる（暗号化情報を除く）。ボリュームが暗号化される場合には、この時点で暗号化情報が開示され、これを用いて、ディスクから読み込まれたいずれかの将来のセクタを解読する。

・ブート・マネージャがディスクからメモリ内に読み込まれる。ブート・マネージャを

測定してＰＣＲ［１０］に入れる。実行をブート・マネージャに転送する。（上述のような１つの変形は、予想されるＰＣＲ［１０］の測定値を秘匿されたデータの中に格納し、これを用いて、正しいブート・マネージャが測定されたことを検証することができる。）

・ブート・マネージャは、重要なデータを測定してＰＣＲ［１１］に入れる。重要なデータは、例えば、デバッガがイネーブルにされようとしているかどうかといったセキュリティに影響を与え得る情報を含むことができる。幾つかの実施形態においては、ＰＣＲ［１１］が情報を用いて拡張されるまで、この情報に従って動作することができない。

・ブート・マネージャは、ＯＳローダーを選択してメモリに入れ、これを測定してＰＣＲ［１２］に入れ、実行を該ＰＣＲ［１２］に転送する。

・ＯＳローダーは、重要なデータを測定してＰＣＲ［１３］に入れる。

・ＯＳローダーは、ＰＣＲ［４］、ＰＣＲ［８］乃至［１３］、および随意的にいずれかの付加的ＰＣＲを用いて、ＯＳローダーによって用いられる秘密を開示する。

・ＯＳローダーは、「コード完全性（Ｃｏｄｅ　Ｉｎｔｅｇｒｉｔｙ）」を転送し、システムのさらなる検証を行う。

・コード完全性は、フェーズ－０ドライバ、ＮＴＫＲＮＬ、およびＨＡＬのようなシステムによって読み込まれる将来のバイナリの各々を検証する。

・ＮＴＫＲＮＬは、ＬＳＡＳＳおよびＷｉｎＬｏｇｉｎを含む初期のシステム・プロセスを開始する。

・ＬＳＡＳＳは、ＰＣＲ［４］、ＰＣＲ［８］乃至［１３］、および随意的にいずれかの付加的ＰＣＲを用いて、ＳＹＳＫＥＹを開示する。開示に失敗した場合、ＬＳＡＳＳは、原因を判断し、是正措置を示唆し、および／またはリカバリー情報を要求し、二次的な方法によって秘密を取得する。

・解読されたブート・ボリュームにアクセスする全てのコードが、ＰＣＲ［４］、ＰＣＲ［８］および［９］、並びにいずれかの付加的な指定のＰＣＲを用いて、ブート・ボリューム解読秘密を開示する。

【００７８】

ＥＦＩシステムにおいて、上記のプロセスに対する幾つかの変形が、有利である。例えば、ＭＢＲを測定し、実行をこれに転送する代わりに、以下の動作を取ることができる。すなわち、

・オプションＲＯＭのものに加えて、ＲＯＭベースのドライバを測定し、ＰＣＲ［２］に入れる。

・ブート・マネージャを含むディスク・ベースのドライバおよびモジュールを測定して、ＰＣＲ［４］に入れる。

・ＮＴＦＳを理解するいずれかのＥＦＩドライバが、ブート・ボリューム解読秘密を開示するための付加的な能力を有する。

【００７９】

上記のプロセスおよびその変形は、標準的なコンピュータ・ブートの範囲を超える目的のために用いることができる。具体的には、２つの付加的な目的が考えられるが、本発明の付加的な使用もまた可能であり、本発明は、特定の設定または目的に限定されるものではない。第１に、上記のプロセスは、ハイバネーション・ファイルの保護を含むように拡張することができる。第２に、上記のプロセスは、オペレーティング・システムの動作に必要とされるブート・ボリュームおよび他のボリュームの保護を含むように拡張することができる。

【００８０】

ハイバネーション・ファイルの保護について、これは、ハイバネーション・ファイルの暗号化鍵および解読鍵を、使用可能な秘密の中に格納することによって達成することができる。暗号化鍵および解読鍵は、単一の対称鍵とすることができ、あるいは別の対称鍵をシールするのに用いられる非対称鍵とすることもできる。マシンが休止状態になるとき、ハイバネーション・ファイルを暗号化することができる。マシンが検証されたブート・コード経路を介してブートしない限り、ハイバネーション・ファイルを解読することはでき

ず、よって、ハイバネーション・ファイル内に格納されたいずれの秘密も維持される。マシンが検証されたコード経路を介してブートした場合には、ハイバネーション・ファイルは、検証されたコードによって解読され、良好に定義された実行経路で実行が再開され、実行環境のセキュリティを復旧させる。

【００８１】

オペレーティング・システムの動作に必要とされるブート・ボリュームまたは他のボリュームの保護を達成することもできる。この場合、全てのブート・ボリュームは、暗号化するおよび／または全体の完全性チェックを含むことができる。解読に必要とされる鍵は、検証されたブート・コードに対してのみ利用可能であり、次にそうした鍵を用いて、システムのブートを再開するのに必要とされる更なるコードおよびデータを解読する。ディスクの完全性情報を更新するのに必要とされる鍵もまた、検証されたブート・コードに対してのみ利用可能である。全体の完全性チェックを含むシステムは、ひとたび該システムが検証されたコードを実行していることが保証されると、更なる動作のために検証された完全性コードおよびデータだけを選択することができる。検証されたコードだけがこうした使用可能ビットを開示できるので、攻撃者がこうしたシステムを騙し、その完全性が有効であると信じさせることはできない。

【００８２】

保護されたブート・プロセスを修復し、アップグレードするための例示的なシステムおよび方法

本発明の実施形態は、コンピュータを安全にブートするためのシステムおよび方法を診断し、修復し、アップグレードするためのプロセスを組み込むことができる。この目的のために、ブート・プロセス内の問題を診断するための第１の観察は、上述の保護されたブート・プロセスにおいて、秘密を開示するプロセスが、測定値が正しいかどうかを判断するための手段を提供するという点である。したがって、２つの状態、すなわち測定されるコードのものを示す秘密が開示され、検証されたコードだけが実行される状態、または秘密が開示されず、検証されていないコードが実行された可能性があることを示す状態、のいずれかが存在する。診断のために、ＴＣＧ準拠ＢＩＯＳによって作成されたログを検査することによって、何が失敗したかを判断することが可能である。次に、エラーが意図的なものではなく偶発的なものであったときに、この情報を用いて問題を診断し、より多くの情報をフィードバックすることができる。

【００８３】

上述の保護されたブート・プロセスは、ＴＰＭの利用によるシステムの自己検証に依存している。幾つかの実施形態においては、こうしたシステムが、実際には依然として有効であるときに、無効に見えることがある。システムが無効に見えるときには、本発明の種々の実施形態において、その一方または両方を使用可能にできる２つのレゾリューション・パスが存在する、すなわち、第１のものは、ログを検査することから得られた情報を用いて、システムを有効とみなすことができる状態に戻すことができ、第２のものは、システムを有効と見なすべきであることをユーザが認証できる。

【００８４】

システムを有効とみなすことができる状態に戻すために、ログ情報を用いて、ＴＰＭが測定値を無効とみなした理由を診断することができる。変更された何らかのコードを元の状態に戻すこともできる。あるいはまた、ユーザが、システム・ディスクをブートオフする前にネットワーク・ブートを試みることなどによって、普通でない方法でブートした場合に、コンピュータをリブートし、予想される方法でブートしようと試みる。

【００８５】

製品に組み込み、システムを有効な状態に戻す実施形態を補うことができる多数の付加的な機能が存在する。例えば、マシン上のハードウェアが壊れ、ディスクが他の方法で同一の種であるがマシンにマイグレートされる場合には、ＴＰＭの秘密鍵が異なっていてもよい。この場合、マシンの代わりにユーザを認証することができる。多数の機構が、二次認証と呼ばれるものを行うことができる。このための証明書は、容易にアクセス可能なも

のにする必要はなく、例えば、マシンを再びイネーブルにするために電話をかけることを要求することができる。二次認証は、一次ＴＰＭ方法によって解読されるものと同じ秘密を提供し、これを別の方法で取得することができる。こうした実施形態は、一次認証方法と同じ方法を用いる場合と比べて、より強力なセキュリティを提供することができる。例えば、マシンのパスワードを覚えやすい形にする必要はなく、全くランダムに生成することができる。マシンがこの二次の方法による認証を必要とするとき、そのマシンのユーザは、ＩＴ部門に電話をかける。ＩＴ部門は、最適なシステムを用いて、発信者の身元を検証し、該発信者にパスワードを読み上げる。パスワードが入力されたとき、上述のマイグレーション機構をこのシナリオに用い、秘密を新しいＴＰＭ　ＰＣＲ値に再秘匿することができる。さらに、こうしたシステムは、一度しか使用できないパスワードをもたらすパスワード・システムを用いることができ、二次認証機構のための新しいパスワードに再び秘匿された秘密には、新たな電話連絡を必要とする。

【００８６】

コンピュータを安全にブートするためのシステムおよび方法の実施形態は、容易にアップグレードできるように構成することかできる。本発明の実施形態によって監視されるコードは滅多に変更されないが、これらのコード・モジュールの１つが最終的に変更されることは避けられない。さらに、安全なブート・プロセスに用いられる秘密は、システムが最初に構成されたときまたは上述のようなリカバリー後、ＴＰＭに秘匿することができる。

【００８７】

１つまたは複数のブート・コンポーネントをアップグレードする第１の方法は、リカバリー後またはコード修正後に可能になるマイグレーションを利用することができ、ＴＰＭ　ＰＣＲ値が求められるまで、一時的ストレージ内に格納することができる。多くの実施形態において、ＰＣＲ値は現在のブートにおいて知られているので、このことは、リブートを必要としない。しかしながら、コード・モジュールが変更された場合には、リブートすることは、新しいコード・モジュールが測定され、値がＴＰＭ　ＰＣＲに格納されることを保証する。

【００８８】

制御されたコード修正環境においては、１つまたは複数のコンポーネントをアップグレードする第２の方法を用いることができる。この場合、新しいコード修正のために予想されるＰＣＲ値が予め求められ、システムがリブートされる前に予想されるＰＣＲ値に秘密を秘匿することができる。

【００８９】

実行中のシステムは、限定でない次の選択肢、すなわち、

・変更前、例えば、サービスパックは、それがＯＳローダーを変更することを知ることができる。

・変更直後、例えば、ディスクが書式設定された後。

・検証されたシステムについての変更検出後、例えば、シャットダウン時に、システムは、コンポーネントが適法に修正されたことを通知し、マイグレーションをサイレントに行うことができる。

・リカバーの一部として。例えば、システムの起動時に、システムは、リカバーがなされたかどうかを判断し、マイグレーションを行うことができるので、次のブート後にリカバー機構を必要としない。

の１つまたは複数に従って上述のマイグレーションを行うことができる。

【００９０】

安全なブート・プロセスを維持するための更に別のシステムは、ＴＰＭの外部に作成される多数の異なる鍵を提供することができる。こうした鍵の各々は、同じＲＳＡキーイング材料を用いることができるが、各々の鍵の使用は、異なるＰＣＲの組および／またはパスワードにバインドしなければならない。事実、こうした付加的な鍵は、全く何にもバインドされなくてもよい。こうした実施形態においては、少なくとも１つのＢＬＯＢを、全

く何にもバインドされていない各々のディスク・ボリューム（例えば、パーティション）に関連付けることもできる。各々の鍵を異なるブート・コンポーネントから用い、ＢＬＯＢのプライバシーを保証することができる。パスワード型鍵をリカバーのために用いることができ、ＲＳＡキーイング材料を預託することもできる。

【００９１】

この手法は、上述の安全なブート・プロセスとは僅かしか違わないが、保守およびサービスの点で著しい利点が明らかになる。すなわち、ＲＳＡキーイング材料がＴＰＭの外部で生成され、あらゆる鍵において同一であるという事実によって、このＲＳＡ材料は、一事業部門の従業員または全組織の従業員といった多数のユーザに対して、より大きな規模で使用することができる。その結果、その組織のいずれかのマシンの開始およびサービスを可能にするマスター鍵を作成することができる。鍵は、依然として各々のＴＰＭのＳＲＫによって保護されているので、依然として鍵が安全であるとみなすことができる。しかしながら、この実施形態においては、情報技術（ＩＴ）部門のような中枢部は、マシンごとに１つの鍵を格納する必要はなく、寧ろ、論理グループごとに１つの鍵を格納する必要がある。また、多数のＢＬＯＢにわたって多数の鍵を格納するために、中枢部はブート・ブロック内にわずかに少ない記憶空間だけを必要とする。

【００９２】

最後に、上述された実施形態において、管理者は、今やポリシーおよび新しいＲＳＡ鍵をプッシュダウンすることができるので、各マシンにおいて鍵が頻繁に変更される。このことにより、マシンの保守費用が低減される。

【００９３】

完全ボリューム暗号化および保護されたブートを用いたデータへのアクセスの恒久的破壊

上述された安全なブート・プロセスの副産物は、完全ボリュームの暗号化、即ちパーティション内のほとんど全てのデータの暗号化を効率的且つ効果的にサポートできるという点である。このことは、秘密を破壊し、これによりコンピュータ上のデータにアクセスするのに必要とされる重要な情報の破壊に必要とされる労力を矮小化することができる。この有効なデータ破壊は、特定の設定、特に機密データの廃棄が望まれる場合、より具体的にはこうしたデータの迅速な破棄が望まれる場合に、有用である。

【００９４】

本発明を実装するコンピュータを動作させるのに必要とされる秘密を削除することにより、ソフトウェアを再インストールすることなく、コンピュータが使用不能になり、該コンピューティング上のデータへのアクセスを恒久的に防止することができる。これを達成するために、ＴＰＭの内部に格納された秘密を再設定することができる。このことは、ＴＰＭの所有者を変更することにより、自明に行うことができる。ＴＰＭによって秘匿された秘密は、もはや有効でなくなる。二次的リカバリー機構を破壊する必要もある。しかしながら、リカバリー機構が現場から離れている場合には、この機構を破壊するまでの短い期間、マシンを一時的にディスエーブルにし、その後該マシンをリカバーさせる方法を提供することができる。

【００９５】

ＴＰＭ内に格納された秘密およびいずれかのリカバー機構の両方が変更されたとき、マシンのコンテンツすなわちコードおよびデータの両方が、取得不能になる。このことにより、マシンのセキュリティの一掃が、極めて迅速に達成される。このような有効にセキュリティを一掃することの１つの利点は、これによりマシンの再販がより現実的になることである。

【図面の簡単な説明】

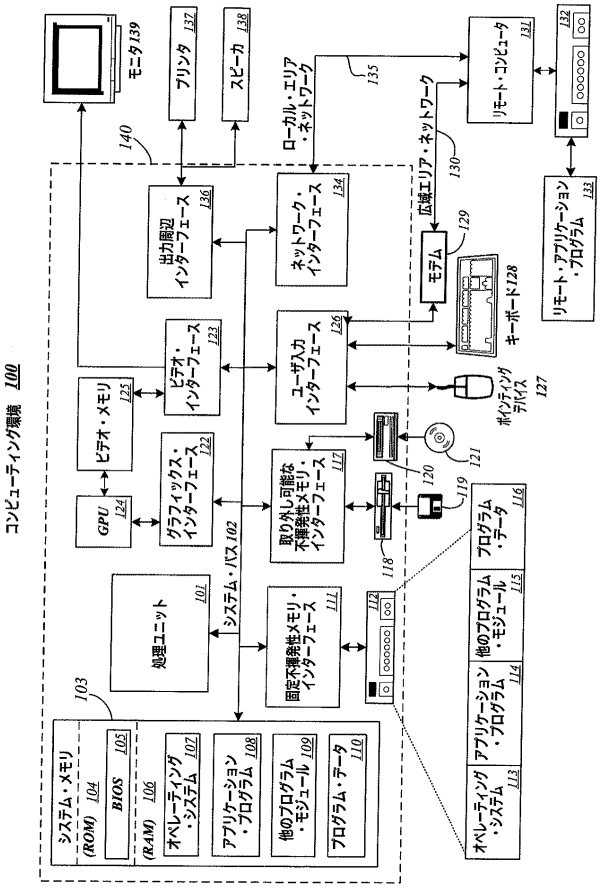
【００９６】

【図１】本発明に関連したソフトウェアおよび／またはハードウェア技術を実装するのに適したコンピューティング環境を示す図である。

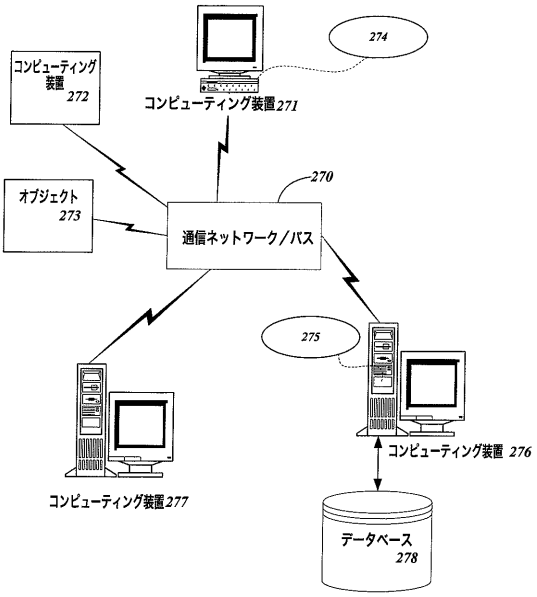
【図２】多数のネットワーク化装置にわたって現代のコンピューティング技術を実行できることを強調するために、図１の基本的なコンピューティング環境の拡張を提供する図で

ある。
【図３】信頼できるプラットフォーム・モジュール（ＴＰＭ）を利用するコンピューティング・プラットフォームを示す図である。
【図４】次のプロセスに移行する前に、複数のソフトウェア・コンポーネントが次のプロセスを測定する例示的なブート・プロセスを示す図である。
【図５】次のコンポーネントの実行を可能にする前に、次のソフトウェアまたはプロセスの完全性を保証する、ＴＰＭのようなハードウェア・セキュリティ・モジュール（ＨＳＭ）を用いるための一般的な技術を示す図である。
【図５ａ】プロセスに用いられるデータの測定値がＴＰＭによって検証されない限り、ブート・プロセスを進行できないことを確実にするためのシステムおよび方法を示す。
【図６】コンピュータをうまくブートすることが、例示的なコンポーネント即ちブート・マネージャをうまく解読し、測定することに結びついている、図５ａに示されるシステムおよび方法の一例を示す図である。
【図７】一定時間リソースにアクセスするブート・コンポーネントを提供し、次にオペレーティング・システムを起動する前に、該リソースへのアクセスを無効にするためのアーキテクチャの動作を示す図である。
【図８】図７のようなアーキテクチャにおいて実行される例示的なステップのためのフローチャートである。
【符号の説明】
【００９７】
１００:コンピューティング・システム環境
１２１：コンピュータ
１３０：広域エリア・ネットワーク
１３５：ローカル・エリア・ネットワーク
３００：ＣＰＵ
３０１、５０３、５０４、７１３：ＴＰＭ
３０２：秘密
３０３、３０４、５０２：ＰＣＲ
３０７：ログ

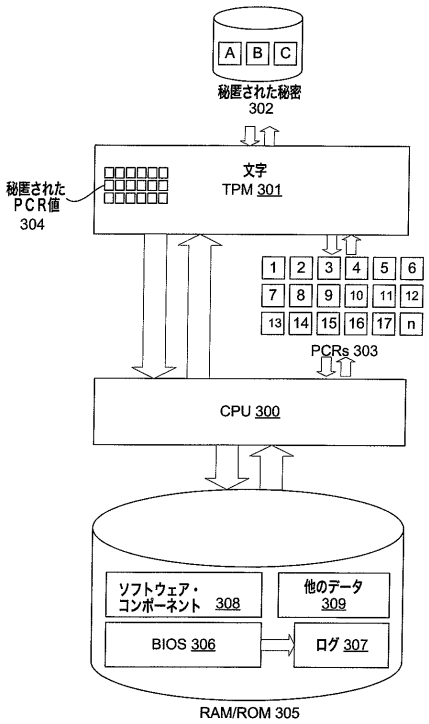
【図１】


コンピューティング環境 100
モニタ 139
プリンタ 137
スピーカ 138
ローカル・エリア・ネットワーク 135
広域エリア・ネットワーク 130
リモート・コンピュータ 131
リモート・アプリケーション・プログラム 133
出力周辺インターフェース 136
ネットワーク・インターフェース 134
モデム 129
キーボード 128
ポインティング・デバイス 127
ビデオ・インターフェース 123
ユーザ入力インターフェース 126
ビデオ・メモリ 125
GPU 124
グラフィックス・インターフェース 122
システム・バス 102
取り外し可能な不揮発性メモリ・インターフェース 117
固定不揮発性メモリ・インターフェース 111
処理ユニット 101
システム・メモリ (ROM) 104
BIOS 105
(RAM) 106
オペレーティング・システム 107
アプリケーション・プログラム 108
他のプログラム・モジュール 109
プログラム・データ 110
プログラム・データ 116
他のプログラム・モジュール 115
アプリケーション・プログラム 114
オペレーティング・システム 113

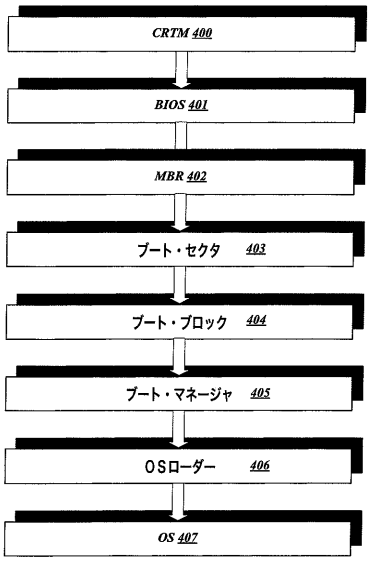
【図２】


コンピューティング装置 272
コンピューティング装置 271
274
オブジェクト 273
270
通信ネットワーク／バス
275
コンピューティング装置 276
コンピューティング装置 277
データベース 278

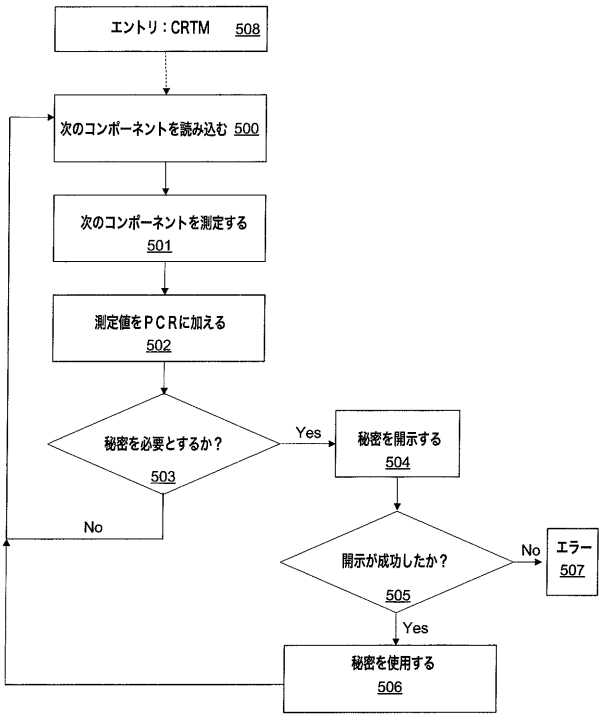
【図３】


A B C
秘匿された秘密 302
秘匿された PCR値 304
文字
TPM 301
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 n
PCRs 303
CPU 300
ソフトウェア・コンポーネント 308
他のデータ 309
BIOS 306
ログ 307
RAM/ROM 305

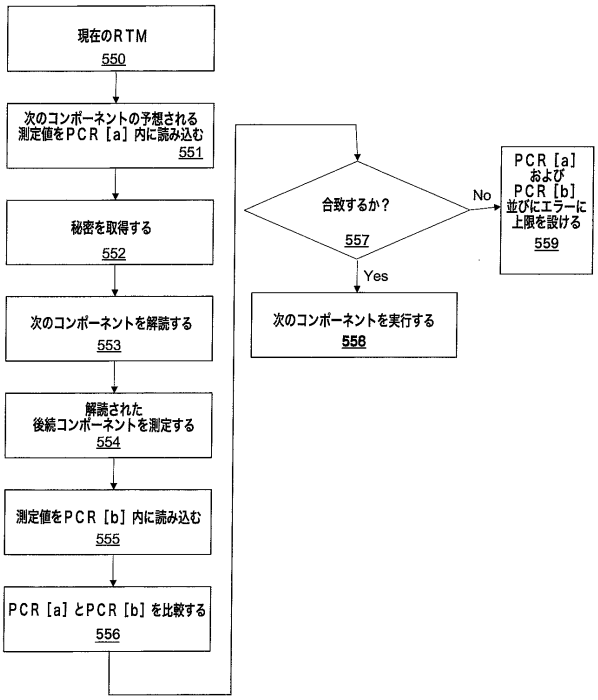
【図４】


CRTM 400
BIOS 401
MBR 402
ブート・セクタ 403
ブート・ブロック 404
ブート・マネージャ 405
OSローダー 406
OS 407

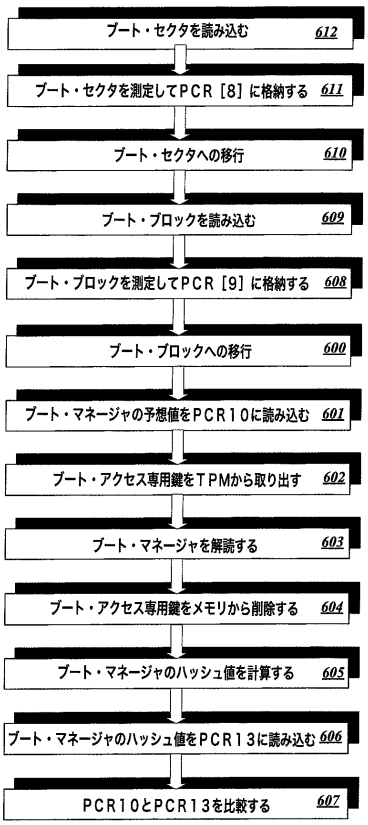
【図５】


エントリ：CRTM 508
次のコンポーネントを読み込む 500
次のコンポーネントを測定する 501
測定値をPCRに加える 502
秘密を必要とするか？ 503
Yes
秘密を開示する 504
No
開示が成功したか？ 505
No
エラー 507
Yes
秘密を使用する 506
No

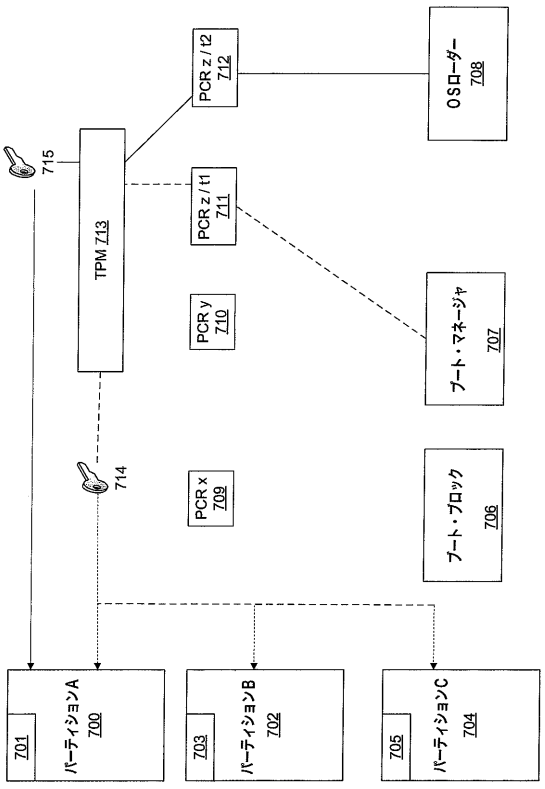
【図５ａ】


現在のRTM 550
次のコンポーネントの予想される測定値をPCR [a] 内に読み込む 551
秘密を取得する 552
次のコンポーネントを解読する 553
解読された後続コンポーネントを測定する 554
測定値をPCR [b] 内に読み込む 555
PCR [a] とPCR [b] を比較する 556
合致するか？ 557
No
PCR [a] および PCR [b] 並びにエラーに上限を設ける 559
Yes
次のコンポーネントを実行する 558

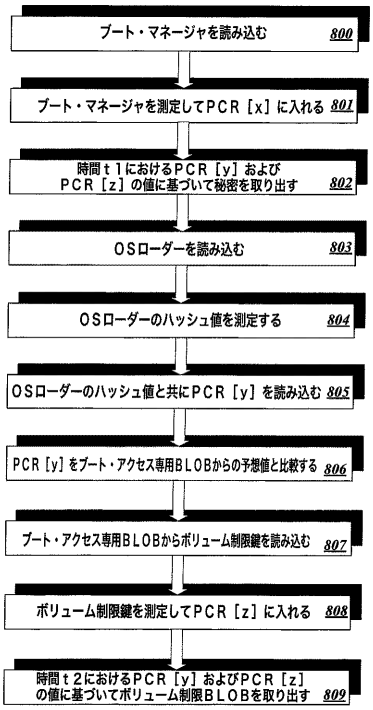
【図６】


ブート・セクタを読み込む 612
ブート・セクタを測定してPCR [8] に格納する 611
ブート・セクタへの移行 610
ブート・ブロックを読み込む 609
ブート・ブロックを測定してPCR [9] に格納する 608
ブート・ブロックへの移行 600
ブート・マネージャの予想値をPCR10に読み込む 601
ブート・アクセス専用鍵をTPMから取り出す 602
ブート・マネージャを解読する 603
ブート・アクセス専用鍵をメモリから削除する 604
ブート・マネージャのハッシュ値を計算する 605
ブート・マネージャのハッシュ値をPCR13に読み込む 606
PCR10とPCR13を比較する 607

【図７】


PCR z / t2 712
OSローダー 708
715
TPM 713
PCR z / t1 711
PCR y 710
ブート・マネージャ 707
714
PCR x 709
ブート・ブロック 706
701
パーティションA 700
703
パーティションB 702
705
パーティションC 704

【図８】


ブート・マネージャを読み込む 800
ブート・マネージャを測定してPCR［x］に入れる 801
時間ｔ１におけるPCR［y］および
PCR［z］の値に基づいて秘密を取り出す 802
OSローダーを読み込む 803
OSローダーのハッシュ値を測定する 804
OSローダーのハッシュ値と共にPCR［y］を読み込む 805
PCR［y］をブート・アクセス専用BLOBからの予想値と比較する 806
ブート・アクセス専用BLOBからボリューム制限鍵を読み込む 807
ボリューム制限鍵を測定してPCR［z］に入れる 808
時間ｔ２におけるPCR［y］およびPCR［z］
の値に基づいてボリューム制限BLOBを取り出す 809

フロントページの続き

(72)発明者 ジェミー ハンター
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ジョナサン ディー．シュワルツ
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ケニス ディー．レイ
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ポール イングランド
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ラッセル ハンフィリーズ
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内
(72)発明者 ステファン トム
アメリカ合衆国 ９８０５２ ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

審査官 後藤 彰

(56)参考文献 特開２００４－３５５５６１（ＪＰ，Ａ）
特開２００４－０１３９０５（ＪＰ，Ａ）
特開２００３－１０８２５７（ＪＰ，Ａ）
国際公開第２００４／０９０７０１（WO，Ａ２）
特表２００６－５２２３７７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／２２
Ｇ０６Ｆ ２１／２４