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

特許第5703391号
(P5703391)

(45)発行日　平成27年4月15日(2015.4.15)　(24)登録日　平成27年2月27日(2015.2.27)

(51)Int.Cl.		F I		
G06F 21/57	(2013.01)	G06F	21/57	350
G06F 21/64	(2013.01)	G06F	21/64	
G06F 21/12	(2013.01)	G06F	21/12	310

請求項の数 12　(全 18 頁)

(21)出願番号	特願2013-549562 (P2013-549562)
(86)(22)出願日	平成24年1月13日 (2012.1.13)
(65)公表番号	特表2014-505943 (P2014-505943A)
(43)公表日	平成26年3月6日 (2014.3.6)
(86)国際出願番号	PCT/US2012/021215
(87)国際公開番号	WO2012/097231
(87)国際公開日	平成24年7月19日 (2012.7.19)
審査請求日	平成25年7月12日 (2013.7.12)
(31)優先権主張番号	13/007,529
(32)優先日	平成23年1月14日 (2011.1.14)
(33)優先権主張国	米国(US)

(73)特許権者　503260918
アップル　インコーポレイテッド
アメリカ合衆国　９５０１４　カリフォルニア州　クパチーノ　インフィニット　ループ　1
(74)代理人　100092093
弁理士　辻居　幸一
(74)代理人　100082005
弁理士　熊倉　禎男
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100086771
弁理士　西島　孝喜

最終頁に続く

(54)【発明の名称】耐タンパー性ブート処理のためのシステム及び方法

(57)【特許請求の範囲】

【請求項１】

フルディスク暗号化を利用して暗号化され、オペレーティングシステムを記憶する記憶媒体を有する、コンピュータデバイスをブート処理する方法であって、

前記コンピュータデバイスにおいて、ブート時間にユーザから、前記暗号化された記憶媒体に関するボリュームキーに関連する資格証明書を受け取る段階と、

暗号化されていない記憶媒体からカーネルキャッシュ及び前記オペレーティングシステムが生成したカーネルキャッシュダイジェストを取り込む段階と、

前記資格証明書及び前記カーネルキャッシュダイジェストに基づいて前記カーネルキャッシュが真正であることを確認する段階と、

前記カーネルキャッシュが真正である場合、前記オペレーティングシステムの開始及び実行、並びに前記暗号化された記憶媒体の復号化を可能にする段階と、

を含む方法。

【請求項２】

ユーザログイン時に、前記資格証明書の少なくとも一部を前記オペレーティングシステムに提供する段階を更に含む、請求項１に記載の方法。

【請求項３】

前記コンピュータデバイスのファームウェアは、カーネルキャッシュが真正であることを確認する、請求項１に記載の方法。

【請求項４】

前記オペレーティングシステムは、暗号化アルゴリズム、ユーザ資格証明書、及びカーネルキャッシュを使用して前記カーネルキャッシュダイジェストを生成する、請求項１に記載の方法。

【請求項５】

命令を記憶するコンピュータ読み取り可能な記録媒体であって、前記命令はプロセッサによって実行されると前記プロセッサに複数の手順を実行させるためのものであり、当該複数の手順は、　コンピュータデバイスにおいて、ブート時間にユーザから、暗号化された記憶媒体に関するボリュームキーに関連する資格証明書を受け取る手順と、

暗号化されていない記憶媒体からカーネルキャッシュ及びオペレーティングシステムが生成したカーネルキャッシュダイジェストを取り込む手順と、

前記資格証明書及び前記カーネルキャッシュダイジェストに基づいて前記カーネルキャッシュが真正であることを確認する手順と、

前記カーネルキャッシュが真正である場合、前記オペレーティングシステムの開始及び実行、並びに前記暗号化された記憶媒体の復号化を可能にする手順と、

を含む、コンピュータ読み取り可能な記録媒体。

【請求項６】

前記複数の手順は、ユーザログイン時に、前記資格証明書の少なくとも一部を前記オペレーティングシステムに提供する手順を更に含む、請求項５に記載のコンピュータ読み取り可能な記録媒体。

【請求項７】

前記コンピュータデバイスのファームウェアは、カーネルキャッシュが真正であることを確認する、請求項５に記載のコンピュータ読み取り可能な記録媒体。

【請求項８】

前記オペレーティングシステムは、暗号化アルゴリズム、ユーザ資格証明書、及びカーネルキャッシュを使用して前記カーネルキャッシュダイジェストを生成する、請求項５に記載のコンピュータ読み取り可能な記録媒体。

【請求項９】

データ処理システムであって、

プロセッサと、

前記プロセッサに接続され、命令を記憶するためのメモリであって、前記命令はプロセッサによって実行されると前記プロセッサに複数の手順を実行させるためのものであり、当該複数の手順は、

ブート時間にユーザから、暗号化された記憶媒体に関するボリュームキーに関連する資格証明書を受け取る手順と、

暗号化されていない記憶媒体からカーネルキャッシュ及びオペレーティングシステムが生成したカーネルキャッシュダイジェストを取り込む手順と、

前記資格証明書及び前記カーネルキャッシュダイジェストに基づいて前記カーネルキャッシュが真正であることを確認する手順と、

前記カーネルキャッシュが真正である場合、前記オペレーティングシステムの開始及び実行、並びに前記暗号化された記憶媒体の復号化を可能にする手順と、

を含む、メモリと、

を備える、データ処理システム。

【請求項１０】

前記複数の手順は、ユーザログイン時に、前記資格証明書の少なくとも一部を前記オペレーティングシステムに提供する手順を更に含む、請求項９に記載のデータ処理システム。

【請求項１１】

前記データ処理システムのファームウェアは、カーネルキャッシュが真正であることを確認する、請求項９に記載のデータ処理システム。

【請求項１２】

前記オペレーティングシステムは、暗号化アルゴリズム、ユーザ資格証明書、及びカーネルキャッシュを使用して前記カーネルキャッシュダイジェストを生成する、請求項９に記載のデータ処理システム。

【発明の詳細な説明】

【技術分野】

【０００１】

（関連出願）

本出願は、２０１１年１月１４日出願の米国特許出願番号１３／００７，５２９「耐タンパー性ブート処理のためのシステム及び方法」の優先権を主張するものであり、その開示内容全体は、引用により本明細書に組み込まれている。

【０００２】

（技術分野）

本開示は、暗号化に関し、詳細にはコンピュータデバイスの耐タンパー性ブート処理に関する。

【背景技術】

【０００３】

現在、大部分のコンピュータデバイスは、ある程度の情報保護を必要とする。暗号手法はコンピュータデバイスを保護するために利用される１つの方法である。暗号手法は、情報の暗号化及び復号化の両方に関連する。暗号化は、理解できる情報（平テキスト）を理解できない情報（暗号化テキスト）に変換する処理であり、復号化は、暗号化テキストを平テキストに戻す処理である。

【０００４】

コンピュータデバイスは、種々のサイズのデータをデータの小さなセットから大きなブロックに暗号化することができる。フルディスク暗号化（ＦＤＥ）は、コンピュータデバイスのディスクボリューム全体を暗号化する方法である。ディスクボリューム全体の復号化は、ディスクボリュームがアクセス可能になる前に必要になる。フルディスク暗号化は、全てのファイル（一時ファイルを含む）が暗号化されるのでファイルレベル暗号化よりも安全と考えられる。

【０００５】

ＦＤＥはファイルレベル暗号化よりも安全と考えられているが、ＦＤＥを使用するシステムは、依然として攻撃に対して脆弱である。単純な１つの攻撃において、ハッカーは、正常なブートシーケンスを実行する代わりにコンピュータデバイスをハッカー自身の悪意のあるコードでブートするように、コンピュータデバイスのブートシーケンスを変更することができる。ＦＤＥシステムは、ブート時にＦＤＥボリュームをアンロックするためにユーザに対してパスワードの入力を促す必要がある。ハッカーの悪意のあるコードは。ＦＤＥパスワードを要求するスクリーンと全く同じ見かけであるが、ＦＤＥボリュームをアンロックするためにパスワードを使用するのではなくこれを盗むスクリーンを表示することができる。ユーザがパスワードを入力すると、ハッカーはパスワードを記録してコンピュータデバイスにアクセスする。従って、ハッカーは、こっそりとユーザのパスワードを盗むことができる。ブースターからオペレーティングシステムカーネルへの信頼チェーンが無いと、この攻撃はフルディスク暗号化コンピュータデバイスであっても破ることができる。

【発明の概要】

【課題を解決するための手段】

【０００６】

本開示の更なる特徴及び利点は、以下の説明に記載され、この説明から部分的に自明であり、又は本明細書に記載の原理を実施することで知ることができる。本開示の更なる特徴及び利点は、特に請求項に指摘される手段及び組み合わせによって実現すること及び得ることができる。本開示の前述の及び他の特徴は、以下の説明及び請求項から更に理解できるようになり、又は本明細書の原理を実施することで知ることができるはずである。

【０００７】
耐タンパー性ブート処理と呼ばれる、フルディスク暗号化を使用して暗号化された記憶媒体を有するコンピュータデバイスをブート処理するためのシステム、方法、非一時的コンピュータ可読記憶媒体が開示される。前述の攻撃に対処する１つの方法は耐タンパー性ブートと呼ばれる。コンピュータデバイスがＦＤＥといった安全性の高いモードを使用する場合、安全性の低いモードで作動する場合よりも、ハッカーは、ブートシーケンスを変更することが一層困難になるはずである。しかしながら、ＦＤＥ可能記憶媒体を備えるコンピュータデバイスにおいて、オペレーティングシステムカーネルは暗号化されない。これは、ブート環境がＦＤＥボリュームを復号するように十分に複雑ではないという理由からである。その代わりに、ブート環境は、復号化を行うためにオペレーティングシステムカーネルに依存する必要がある。暗号化されていないオペレーティングシステムカーネルは、攻撃者によって悪意のあるパスワード盗用コードにこっそりと置き換えることができるので、ＦＤＥセキュリティモデル全体において明らかな脆弱リンクである。コンピュータデバイスに関するこの欠陥に対処するための１つの方法は、ファームウェアからのオペレーティングシステムのカーネルに対する信頼チェーンを確立することによって、ブートシーケンスの信頼性を確認することである。図１は、本明細書に記載の方法を実施する例示的なシステム１００を示す。システム１００は、ブート時間にユーザからＦＤＥ可能暗号化記憶媒体に関するボリュームキーに関連する資格証明書を受信する。システムは、記憶媒体から暗号化されていないカーネル及び暗号化されていないカーネルキャッシュダイジェストを取り込む。次に、システムは、取り込んだカーネルキャッシュダイジェストを演算したダイジェストと比較することによってカーネルキャッシュが真正であることを確認する。オペレーティングシステムの開始及び実行は、カーネルキャッシュが真正であるとシステムが判定した場合に行われる。システムは、カーネルキャッシュが真正ではない場合にはエラーを発生する。

【０００８】
本方法を実行するシステムは、最初にユーザから受け取ったフルディスク暗号化パスワードに基づいてボリュームキーを発生することで、耐タンパー性ブート処理プロセスを開始する。システムは、ボリュームキーで記憶媒体を暗号化して、フルディスク暗号化記憶媒体をもたらす。次に、システムは、フルディスク暗号化ボリュームキーでカーネルキャッシュを暗号化して、カーネルキャッシュダイジェストをもたらし、システムは、これをカーネルキャッシュ自体と一緒に暗号化されていない記憶媒体又はそうでなければ暗号化される記憶媒体の暗号化されていない部分に記憶する。システムがブートすると、システムは、カーネルキャッシュダイジェスト及び演算ダイジェストに基づいてカーネルキャッシュの完全性を確認する。

【０００９】
システムは、ファームウェアを用いてシステムをブート処理すること、及びファームウェアによってブーターの完全性を確かめることで、オペレーティングシステムが不正変更されていないことを確認する。システムは、制御をブーターに渡し、ブーターは、記憶されたカーネルキャッシュダイジェスト及び演算されたダイジェストに基づいてオペレーティングシステムのカーネルキャッシュを確かめる。オペレーティングシステムが不正変更されていないとシステムが判定すると、システムは、開始及び実行及び何らかの残りのタスクのための制御をオペレーティングシステムに渡す。

【００１０】
１つの実施形態において、システムは、ファームウェアレベルで、パスワード検証子を生成して、これをパスワード証明書と比較することで耐タンパー性ブート処理を無効にする。システムは、暗号化アルゴリズムの複数回の反復をサルト値及びパスワードに適用することでパスワード検証子を生成する。システムが耐タンパー性ブート処理を無効にするユーザからのリクエストを受け取ると、オペレーティングシステムは、部分的なパスワード証明書を生成する。部分的なパスワード証明書の生成は、暗号化アルゴリズムの複数回の反復の一部をパスワード及びサルト値に適用することで実現され、次に、コンピュータ

デバイスをリブート処理する。システムがリブートすると、ファームウェアは、部分的なパスワード証明書を取り込み、暗号化アルゴリズムの複数回の反復の残りを遂行して完全なパスワード証明書を生成する。パスワード証明書がパスワード検証子と一致すると、システムは、耐タンパー性ブート処理を無効にする。このようにして、システムは、リクエストを真正として耐タンパー性ブートを無効にする。

【００１１】

更に、システムは、パスワード検証子を記憶するデータベースを確立することで耐タンパー性ブート処理を初期化する。オペレーティングシステムは、耐タンパー性ブート処理を無効にする権限のあるユーザリストを生成し、このリストをファームウェアに送る。オペレーティングシステムは、各ユーザに関するパスワード検証子を生成し、これを同様にファームウェアに送る。ファームウェアは、これを不揮発性ランダムアクセスメモリ（ＮＶＲＡＭ）データベースに記憶する。また、システムは、ファームウェアにおいてシステムサルトを生成し、システムサルトを後で使用するためにＮＶＲＡＭに記憶する。本明細書に開示する原理は、フルディスク暗号化を用いて暗号化された記憶媒体を有するコンピュータデバイスに適用する。

【００１２】

本開示の前述及び他の利点及び特徴を得ることができる方法を説明するために、前述の簡潔な原理のより詳細な説明は、添付図面に示される特定の実施形態を参照して与えられる。図面は、本開示の例示的な実施形態を示すだけであり、この範囲を限定することを意図しておらず、本明細書の原理は添付図面を利用して、追加の特定性及び詳細内容を用いて記載及び説明されることを理解されたい。

【図面の簡単な説明】

【００１３】

【図１】例示的なシステムの実施形態を示す。
【図２】耐タンパー性ブート処理のための例示的な方法の実施形態を示す。
【図３】カーネルキャッシュの認証を示す。
【図４】例示的なシステムの実施形態の確立を示す。
【図５】耐タンパー性ブート処理を確立するための例示的な方法の実施形態を示す。
【図６】耐タンパー性ブートを無効にするための例示的な方法の実施形態を示す。
【図７】パスワード検証子の発生を示す。
【図８】パスワード検証子を発生するための例示的な論理の流れを示す。
【図９】部分的なパスワード証明書の生成を示す。
【図１０】完全なパスワード証明書の生成を示す。
【図１１】部分的なパスワード証明書を生成する例示的な論理の流れを示す。
【図１２】完全なパスワード証明書を生成する例示的な論理の流れを示す。
【図１３】フルディスク暗号化の確立を示す。
【図１４】フルディスク暗号化を確立するための例示的な方法の実施形態を示す。

【発明を実施するための形態】

【００１４】

以下に、本開示の種々の実施形態を詳細に示す。特定の実施例が記載されるが、これは例示目的であることを理解されたい。当業者であれば、本開示の精神及び範囲を逸脱することなく他の構成要素及び機器構成を使用できることを理解できるはずである。

【００１５】

本開示は、ブートシーケンスの真正性の確認に関する従来のニーズに対処するものである。フルディスク暗号化を用いて暗号化された記憶媒体を有するコンピュータデバイスをブートするシステム、方法、及び非一時的コンピュータ可読媒体が開示される。図１には基本的な汎用システム又はコンピュータデバイスが説明され、本明細書に開示の原理を実行するために使用できる。以下に、フルディスク暗号化（ＦＤＥ）を用いて、コンピュータデバイス上で耐タンパー性ブート処理を確立、管理、使用、及び除去するための詳細な説明を行う。これらの変形形態は、本明細書では種々の実施形態として説明される。

【００１６】
図１を参照すると、例示的なシステム１００は、プロセッシングユニット（ＣＰＵ又はプロセッサ）１２０、及びリードオンリメモリ（ＲＯＭ）１４０及びランダムアクセスメモリ（ＲＡＭ）１５０等のシステムメモリ１３０を含む種々のシステム構成要素をプロセッサ１２０に接続するシステムバス１１０を含む汎用コンピュータデバイス１００を備える。システム１００は、プロセッサ１２０に直接、ごく接近して接続されるか、又はその一部として統合される高速メモリのキャッシュ１２２を含むことができる。システム１００は、プロセッサ１２０による迅速なアクセスのためにメモリ１３０及び／又は記憶装置１６０からキャッシュにデータをコピーする。このようにして、キャッシュは、プロセッサ１２０がデータを待つ間の遅れを回避して性能向上を可能にする。これらの及び他のモジュールは、プロセッサ１２０を制御すること又はプロセッサ１２０を制御するように構成することができ、種々の動作を実行するようになっている。他のシステムメモリ１３０を利用でき同様に使用することができる。メモリ１３０は、性能特性が異なる複数の異なる種類のメモリを含むことができる。本開示は、高い処理性能をもたらすようにネットワーク化された２つ以上のプロセッサ１２０、或いはコンピュータデバイスのグループ又はクラスタを有するコンピュータデバイス１００上で作動できることを理解されたい。プロセッサ１２０は、任意の汎用プロセッサ、及び記憶装置１６０に記憶されプロセッサ１２０並びに特定用途向けプロセッサを制御するように構成される、モジュール１　１６２、モジュール２　１６４、及びモジュール３　１６６といったハードウェアモジュール又はソフトウェアモジュールを含むことができ、ソフトウェア命令は、実際のプロセッサデザインに組み込まれる。プロセッサ１２０は、本質的に、複数のコア又はプロセッサ、バス、メモリコントローラ、キャッシュ等を含む完全内蔵式コンピュータシステムとすることができる。マルチコアプロセッサは、対称又は非対称とすることができる。

【００１７】
システムバス１１０は、任意の種々のバスアーキテクチャを使用するメモリバス又はメモリコントローラ、周辺バス、ローカルバスを含む、任意の種々の形式のバス構成とすることができる。ＲＯＭ１４０等に記憶される基本入出力（ＢＩＯＳ）は、起動時等にコンピュータデバイス１００内の各要素の間の情報転送を助ける基本ルーチンを提供することができる。更に、コンピュータデバイス１００は、ハードディスクドライブ、磁気ディスクドライブ、光ディスクドライブ、テープドライブ等の記憶装置１６０を含む。記憶装置１６０は、プロセッサ１２０を制御するためのソフトウェアモジュール１６２、１６４、１６６を含むことができる。他のハードウェア又はソフトウェアモジュールも想定できる。記憶装置１６０は、ドライブインタフェースによってシステムバス１１０に接続される。ドライブ及び関連のコンピュータ可読記憶媒体は、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、及びコンピュータデバイス１００に関する他のデータの非一時的記憶を可能にする。１つの態様において、特定の機能を実行するハードウェアモジュールは、プロセッサ１２０、バス１１０、ディスプレイ７０等の必須のハードウェア構成要素に関連する、非一時的コンピュータ可読媒体に記憶されるソフトウェア構成要素を含む。基本構成要素は、当業者には公知であり、装置１００が小型携帯式コンピュータデバイス、ディスクトップコンピュータ、又はコンピュータサーバであるか否かの装置の形式に応じて、適切な変形形態が想定できることを理解できるはずである。

【００１８】
本明細書に記載の例示的な実施形態は、ハードディスク１６０を使用するが、当業者であれば、例示的な作動環境において、磁気カセット、フラッシュメモリカード、デジタル多用途ディスク、カートリッジ、ランダムアクセスメモリ（ＲＡＭ）１５０、リードオンリメモリ（ＲＯＭ）１４０、ビット流を含む有線又は無線信号等の、コンピュータがアクセス可能なデータを記憶することができる他のコンピュータ可読媒体を使用できることを理解できるはずである。非一時的コンピュータ可読記憶媒体は、特にエネルギ、搬送信号、電磁波、信号等の媒体を本質的に除外する。

【００１９】

ユーザがコンピュータデバイス１００と相互作用すること可能にするために、入力デバイス１９０は、会話用のマイク、ジェスチャ又は図形入力用のタッチセンサ式スクリーン、キーボード、マウス、モーション入力、会話等の任意数の入力機構を表す。また、出力デバイス１７０は、当業者には公知の１つ又はそれ以上の複数の出力機構とすることができる。いくつかの例では、マルチモーダルシステムにより、ユーザは、複数の形式の入力を行ってコンピュータデバイス１００と通信することができる。一般的に、通信インタフェース１８０は、ユーザ入力及びシステム出力を制御及び管理する。何らかの特定のハードウェア構成上での作動に制限されないので、基本的な特徴は、改良されたハードウェア又はファームウェア構成に置き換えることが容易である。

【００２０】

説明を明瞭化するために、システムの実施形態は、「プロセッサ」又はプロセッサ１２０と表記される機能ブロックを含む個々の機能ブロックを含むように示される。これらのブロックの機能は、共有又は専用ハードウェアのいずれかでもたらすことができ、限定されるものではないが、プロセッサ１２０のようなソフトウェア及びハードウェアを実行することができるハードウェアを含み、これは汎用プロセッサ上で実行されるソフトウェアと同等に作動する専用のものである。例えば、図１に示す１つ又はそれ以上のプロセッサの機能は、単一の共用プロセッサ又は複数のプロセッサで提供することができる（用語「プロセッサ」の使用は、ソフトウェアを実行することができるハードウェアのみを呼ぶと解釈すべきではない）。例示的な実施形態は、マイクロプロセッサ及び／又はデジタル信号プロセッサ（ＤＳＰ）ハードウェア、以下に説明する動作を行うソフトウェアを記憶するリードオンリメモリ（ＲＯＭ）１４０、及び結果を記憶するランダムアクセスメモリ（ＲＡＭ）１５０を含むことができる。汎用ＤＳＰ回路と組み合わせて超大規模集積回路（ＶＬＳＩ）ハードウェアの実施形態及びカスタムＶＬＳＩ回路も可能である。

【００２１】

種々の実施形態の論理演算は、（１）汎用コンピュータ内のプログラム可能な回路上で実行する一連のコンピュータにより実現されたステップ、動作、又は手順、（２）専用のプログラム可能な回路上で実行する一連のコンピュータにより実現されたステップ、動作、又は手順、及び／又は（３）プログラム可能な回路内の相互接続されたマシンモジュール、又はプログラムエンジンとして実現される。図１に示されたシステム１００は、列挙された方法の全て、又は一部を実施し、列挙されたシステムの一部であり、及び／又は列挙された非一時的コンピュータ可読記憶媒体における命令に従って動作する。そのような論理演算は、モジュールのプログラミングに従って特定の機能を実行するようプロセッサ１２０を制御するように構成されたモジュールとして実現される。例えば、図１は、プロセッサ１２０を制御するように構成されたモジュールである３つのモジュールＭｏｄ１　１６２、Ｍｏｄ２　１６４及びＭｏｄ３　１６６を示す。これらのモジュールは、記憶装置１６０上に格納されるか、実行時にＲＡＭ１５０、又はメモリ１３０にロードされるか、又は当技術分野において既知であるように他のコンピュータ可読記憶場所に格納される。

【００２２】

いくつかの基本的なシステム構成要素を開示したが、次に、本開示は、図２の例示的な方法の実施形態に示すような、耐タンパー性ブート処理の説明に戻る。明瞭化のために、本方法を実行するように構成された図１に示すような例示的システムに関して本方法を説明する。

【００２３】

図２は、耐タンパー性ブート処理とし知られる、フルディスク暗号化を使用して暗号化された記憶媒体を有するコンピュータデバイスをブート処理するための例示的な方法の実施形態を示す。耐タンパー性ブート処理により、ハッカーは、認可されたブートシーケンスを認可されていない方法で変更すること、又は自身の認可されていないブートシーケンスを挿入することがより困難になる。ブートシーケンスを保護する１つの方法は、ファームウェアからオペレーティングシステムカーネルへの信頼チェーンを確立することである

。ファームウェアは、デバイスの低レベル基本動作を制御する固定プログラムを呼ぶ。しかしながら、ファームウェアは「固定」プログラムと呼ばれるが、例えば、ファームウェア更新によりファームウェアが提供する機能性を変更することができる。典型的に、ファームウェアは、ＢＩＯＳ（基本入出力システム）ベース又はＥＦＩ（エクステンシブル・ファームウェア・インタフェース）ベースのコンピュータブートアップ実装としてＲＯＭ、ＥＥＰＲＯＭ等に記憶される。カーネルは、中央処理装置（ＣＰＵ）、メモリ、及び他のデバイスを含む、コンピュータデバイスのハードウェアのようなコンピューティング資源にアクセスできるソフトウェアアプリケーションを提供するオペレーティングシステムの主要な構成要素である。信頼チェーンの確立において、ファームウェアはブーター（ブーター）の完全性を確かめることができるが、ブーターはカーネルの完全性を確かめることはできない。コンピュータは製造工場からの出荷後に自身のカーネルを更新し、ブート時の特定のマシンに関するカーネル拡張を含む。カーネルは、特定のユーザのマシンに対してカスタマイズされる。ブーターはカーネルの完全性を確かめてブートシーケンス全体を保護する必要がある。

【００２４】

システム１００がフルディスク暗号化を使用する場合、暗号化されるボリュームに関するボリュームキーは、ブーターとカーネルとの間の共有秘密キーとして機能する。ブーターは、パスワードを確認してこれをオペレーティングシステムカーネルに送るために、ボリュームキーを有する必要があり、その後、オペレーティングシステムカーネルはディスクを復号することができる。システムの実行時、カーネルは、メモリ内にフルディスク暗号化ボリュームキーを有することもできる。ブート時間において、ブーターは、ボリュームキーを使用してカーネルの完全性チェックを行う。ボリュームキーは、パスワードと同一とすること、又は少なくともパスワードの一部に基づいて形成することができる。

【００２５】

システム１００は、ファームウェアからカーネルへの信頼チェーンを確立して、コンピュータデバイスのブートシーケンスを保護する。システムは、ブート時間において、ユーザからボリュームキーに関する資格証明書を受信する（２１０）。暗号手法において、資格証明書は、ユーザ識別を証明するために使用される。資格証明書のいくつかの例として、パスワード証明書、又は指紋や声紋等の生体認証を挙げることができる。ボリュームキー資格証明書と同じか、又はその派生物である。例えば、ボリュームキーは、ユーザパスワード、暗号化されたパスワード、又はユーザパスワードを入力とするアルゴリズムの出力とすることができる。システム１００がボリュームキーを証明すると、システムは、カーネル及び暗号化されていない記憶媒体からオペレーティングシステムが生成するカーネルキャッシュダイジェストを取り込む（２３０）。カーネルキャッシュダイジェストは、暗号学的ハッシュ関数、秘密キー、及びカーネルキャッシュを用いて生成される。カーネルキャッシュは、カーネルコード及びカーネル拡張を含む。ダイジェストは、ハッシュ値、メッセージダイジェスト、又はハッシュ付きメッセージ認証コード（ＨＭＡＣ）と呼ばれる場合もある。ＨＭＡＣは、メッセージの真正性及び完全性の両方を同時に確認するために使用できる。暗号学的ハッシュ関数は、データブロックを取得して、ハッシュ値又はメッセージダイジェストと呼ばれる、固定サイズのビットストリングを戻す手続きである。データブロックの何らかの変更は、ダイジェストを変更することになる。カーネルキャッシュダイジェストは、カーネルキャッシュのキー付きハッシュであり、キーはＦＤＥボリュームキーである。

【００２６】

システムが記憶されたカーネルキャッシュダイジェストを取り込むと、システムは、カーネルキャッシュが、ユーザ入力の資格証明書に基づいて真正であることを確認し（２４０）、カーネルキャッシュが真正であるとシステムが判定する場合、オペレーティングシステムの開始及び実行を可能にする（２５０）。システムは、カーネルキャッシュが真正でない場合、エラーを生成し（２６０）、オペレーティングシステムを実行しない。システムは、カーネルの完全性を確認できない場合、コンピュータデバイスが不正変更されて

いると仮定する。この処理は、完全性チェックがブート処理の各段階で行われるという理由から耐タンパー性ブートと呼ばれ、不正変更に対するシステム耐性をもたらし、ブート処理の重要部分が許可なく変更された場合を検出することができる。

【００２７】

ユーザが手動でフルディスク暗号化を行うこと、又はいくつかの自動処理がフルディスク暗号化を開始することができるが、大部分の自動処理は、システムがボリュームキーを生成するパスワードの入力といった特定のユーザ入力を必要とする。耐タンパー性ブート処理は、フルディスク暗号化が可能になった場合及び／又は何らかの適切な共有秘密キーがフルディスク暗号化ボリュームキーの変わりとして利用可能になった場合の他の状況において使用できる。

【００２８】

図３は、ファームウェアからカーネルへの信頼チェーンの確立の一部である、例示的なカーネルキャッシュ認証を示す。システム１００は、暗号化アルゴリズム、ユーザ入力パスワード、及びカーネルキャッシュを用いてＨＭＡＣを演算することで（３１０）、カーネルキャッシュが真正でないことを確認する（２４０）。システム１００は、ＨＭＡＣを出力する暗号学的ハッシュ関数への入力としてカーネルキャッシュ及びパスワードを使用する。システム１００は、演算したＨＭＡＣを記憶されたカーネルキャッシュダイジェストと比較して（３２０）、オペレーティングシステムが不正変更されたか否かを判定する。比較された値が一致する場合（３３０）、オペレーティングシステムは不正変更されておらず真正とみなし、システムはブートを継続できる。比較された値が異なる場合（３４０）、カーネルキャッシュダイジェストが生成され真正ではないのでオペレーティングシステムは変更されている。この時点で、システムはブート処理を停止して、ユーザにパスワードの再入力を指示し、リモートデバイスに警告を送ること、及び／又は何らかの他の所望のアクションを起動することができる。

【００２９】

図４は、耐タンパー性ブート処理を示す。システム１００は、ファームウェア４１０を用いてシステムをブート処理して、ファームウェア４１０によってブーター４２０の完全性を確かめることによって、オペレーティングシステムが不正変更されていないことを確認する。システムがファームウェアよってブーターの完全性を確かめると、システムは、制御をブーター４２０に渡す。ブーターは、記憶されたカーネルキャッシュダイジェスト４３０及びブート時間４２０に入力されたパスワードに基づいてオペレーティングシステムのカーネルキャッシュ４４０を確かめる。カーネルキャッシュの真正性は、カーネルキャッシュ及びユーザ入力パスワードを用いてＨＭＡＣを演算することで（４５０）判定される。ブート処理がオペレーティングシステムは不正変更されてないと判定すると、開始及び実行のために、ブート処理は制御をオペレーティングシステムに渡す。

【００３０】

図５は、耐タンパー性ブート処理の開始を示す。システム１００は、ユーザから受け取ったフルディスク暗号化パスワードに基づいて最初にボリュームキーを生成することで、耐タンパー性ブート処理を開始する（５１０）。システム１００は、ボリュームキーを用いて記憶媒体を暗号化してフルディスク暗号化記憶媒体を得る（５２０）。ボリュームキーは、フルディスク暗号化及び／又はユーザ登録処理に関するセットアップ処理の一部として生成することができる。例えば、ユーザが記憶ボリュームのフルディスク暗号化のセットアップを決めた場合、システムは、ユーザに対してフルディスク暗号化パスワード又はパスワードに等価なものの入力を指示する。次に、システムは、フルディスク暗号化パスワードに基づいてボリュームキーを生成し、ボリュームキーを用いてボリュームを暗号化し、同時に耐タンパー性ブート処理に関して同じボリュームキーを使用することができる。次に、システム１００は、カーネルキャッシュ及びフルディスクボリュームキーを用いてカーネルキャッシュダイジェストを生成し（５３０）、これをシステム１００は暗号化されていないボリュームに記憶する（５４０）。システムがブートすると、システムは、記憶されたカーネルキャッシュダイジェスト及び演算されダイジェストに基づいてカー

ネルキャッシュの完全性を確認する。少なくとも１つの変形形態において、システム１００は、暗号化された記憶ボリューム以外の場所にカーネルキャッシュダイジェストを記憶するが、この理由は、ブート環境が、アンロック処理が非常に複雑でフル機能のカーネルを必要とするのでフルディスク暗号化ボリュームをアンロックできないからである。従って、カーネルキャッシュは、ブーターがフルディスク暗号化ボリュームをアンロックできないのでの、暗号化ボリューム内に存在できない。その結果、カーネルキャッシュは、フルディスク暗号化ボリュームとは別のボリューム上に暗号化されずに存在し、ここではブーターは、該ブーターでは不可能な復号化を行うことなくカーネルキャッシュに達することができる。データはＦＤＥボリューム上で暗号化されるが、カーネルキャッシュは全く保護されず、攻撃者による置き換えにさらされる可能性がある。

【００３１】

カーネルは、キー付きＨＭＡＣダイジェスト（フルディスク暗号化ボリュームキーがダイジェストキーとして使用される）がカーネルの隣に存在する限り、保護されていないボリューム上に存在することができる。従って、攻撃者は、カーネルを悪意のあるものに置き換えることができるが、攻撃者はフルディスク暗号化パスワードを知らないので、攻撃者はマッチングダイジェストファイルを作ることができず、ブーターは、悪意のあるカーネルのブートを拒否することになる。

【００３２】

耐タンパー性ブート処理の開始及び実行を説明したが、本開示は、以下に、コンピュータデバイスの耐タンパー性ブート処理を無効にする方法を説明する。システム１００は、最初に耐タンパー性ブート処理を無効にするリクエストを認証して、リクエストが有効でハッカーからのものではないことを確認する必要がある。システムは、パスワード証明書（ｐａｓｓｗｏｒｄ　ｐｒｏｏｆ）を生成することで、耐タンパー性ブート処理を無効にするリクエストが有効であることを確認する。ファームウェア及びオペレーティングシステムは同時に実行されないので、従来のやり方は、２つのプロセスが相互通信するための実行可能な方法を提示しない。オペレーティングシステムは、ファームウェアに情報を送ってマシンをブートすることができ、ファームウェアは、ブート時間にオペレーティングシステムからのメッセージを受け取ることができる。しかしながら、いくつかの状況では、複数のリブートは許されないので、システムは、コマンドをファームウェアに対して一度だけ送ることができる。ファームウェアは、オペレーティングシステムによって書き込みできないが読み出し可能な不揮発性ランダムアクセスメモリ（ＮＶＲＡＭ）チップに記憶される。オペレーティングシステムは、平テキストで記憶されオペレーティングシステムで読み出し可能なので、元のパスワードを送ることはできない。また、オペレーティングシステムは、攻撃者がハッシュを読み取ってアンロック機構を始動させることができるので、パスワードのハッシュを送ることはできない。オペレーティングシステムは、保護されていないが、依然として無効リクエストの真正性を確認するデータを送る必要がある。このことは、パスワード検証子（ｐａｓｓｗｏｒｄ　ｖｅｒｉｆｉｅｒ）及びパスワード証明書を使用して実現できる。

【００３３】

図６は、ファームウェアレベルでパスワード検証子を生成してパスワード証明書と比較することで、耐タンパー性ブート処理を無効にする方法を示す。システム１００は、パスワード検証子を生成して、後で使用するために記憶する（６１０）。システム１００は、ユーザからリクエスト及びパスワードを受け取り、耐タンパー性ブートを無効にする（６２０）。オペレーティングシステムは、暗号化アルゴリズムの複数回の反復の一部を要求パスワード及びソルト値に適用することで部分的なパスワード証明書を生成し（６３０）、次に、コンピュータデバイスを再起動する（６４０）。システムがリブートすると、ファームウェアは、部分的なパスワード証明書を取り込んで（６５０）、暗号化アルゴリズムの残りの反復回数を実行して、完全なパスワード証明書を得る（６６０）。パスワード証明書がパスワード検証子に一致する場合、システム１００は、耐タンパー性ブート処理を無効にする（６７０）。システムは、リクエストが真正であることが確認される場合の

み耐タンパー性ブート処理を無効にする。

【００３４】

図７及び８は、パスワード検証子の生成方法を示す。システム１００は、先行する反復の出力を現在の反復の入力として利用して、暗号化アルゴリズムの複数回の反復を実行することでパスワード検証子を生成する（７２０）。例えば、本プロセスは、PV_1を使用してPV_2を生成し、PV_{99}を使用してPV_{100}を生成する。最初の反復の場合に（７１０）、先行する反復からの出力がないのでパスワードＰが入力として使用される。システム１００は、例えば１００回の反復を行うことでパスワード検証子を生成する（７４０）。パスワード証明書は、中間点の反復といった、１つ又はそれ以上の中間の反復の出力から得られる（７３０）。先行の反復からの出力を暗号化アルゴリズムの入力として使用することに加えて、サルト値を使用する。サルト値は、通常暗号学的ハッシュ関数であるキー派生関数への１つの入力として使用されるランダムビットを含む。サルトデータは、記憶量及びパスワードを決定するのに必要な演算量を増大させるので攻撃を困難にする。盗んだハッシュからパスワードを特定するために、攻撃者はランダム特性のハッシュを計算する必要があるので、所要の演算時間が増加する。いくつかの状況では、このプロセスはフルディスク暗号化ボリュームを有するデバイス上のプロセッサでの実行に限定される、結果的に、反復回数は、プロセッサ速度に基づいて選択又は決定することができ、何らかの攻撃に必要な時間は、プロセッサの速度の関数であり、総当たり攻撃で見つけることは容易ではない。ＨＭＡＣを演算する際にＳＨＡ－１又はＭＤ５のような任意の暗号学的ハッシュ関数を使用することができる。

【００３５】

パスワード検証子の生成は、パスワードＰ、サルト値Ｓ、実行反復回数ｎ、及びインデックスｉを必要とする（８１０）。システムは、インデックスｉと反復回数ｎを比較して所要の反復回数が終了しているか否かをチェックする（８１０）。両者が等しくない場合、プロセスは終了しておらず追加の反復が必要である。次に、システムは、現在のラウンドｉに関するラウンドパスワード検証子を生成する（８３０）。ラウンドパスワード検証子ＰＶｉの生成は、暗号化アルゴリズム、先行するラウンドＰＶｉ－１からのパスワード検証子（現在にラウンドが１ではない場合）、及びサルト値Ｓを用いて行われる（８３０）。最初のラウンドに関しては先行するラウンド検証子が存在しないので、パスワードＰを使用してPV_0を初期設定する（８１０）。次に、システムはインデックスをインクリメントして（８３０）、このインデックスを反復回数と比較して所要の反復回数が終了したか否かをチェックする（８２０）。インデックスが反復回数と一致する場合、プロセスは終了してシステムはパスワード検証子を出力する（８４０）。

【００３６】

システムは、パスワード検証子及び同じプロセスに続いてパスワード証明書を生成するが、パスワード証明書を生成する段階はコンピュータデバイスをブート処理すること以外で同じプロプロセスである。システムは、通常は半分である全反復回数の所要の一部だけで遂行して、自身がリブートする。システムが再起動されると、残りの反復回数が遂行されて最終的なパスワード証明書がもたらされる。図９及び１０は、パスワード証明書の生成方法を示す。例えば、パスワード証明書を生成するのに必要な全反復回数は１００であり、システムは、５０回の反復を使用して一部のパスワード証明書を生成し、その結果をリブート処理の前に記憶する（９１０）。システムが再起動すると、記憶された部分パスワード証明書を読み込んで残りの５０回の反復を遂行して最終的なパスワード証明書をもたらす（１０１０）。部分的なパスワード証明書は、中間点で又は特定の反復セットの他の点で生成することができる。部分的なパスワード証明書は、元のパスワード、反復回数、パスワード検証子、又はパスワード証明書に関する何らかの有用な情報を示さないので、平テキスト及び／又は他のプロセスが読み出すことができるメモリ領域に記憶しても安全である。

【００３７】

図１１及び１２は、パスワード証明書の生成方法を示す。システム１００は、先行する

反復の出力を現在の反復の入力として利用して、暗号化アルゴリズムの複数回の反復を遂行することでパスワード証明書を生成する。例えば、このプロセスは、PV_1を使用してPV_2を生成し、PV_{99}を使用してPV_{100}を生成する。最初の反復の場合には先行する反復からの出力がないのでパスワードＰを使用する。先行する反復からの出力を暗号化アルゴリズムの入力として使用することに加えて、サルト値を使用する。

【００３８】

パスワード証明書の生成は、パスワードＰ、サルト値Ｓ、実行反復回数ｎ、インデックスｉ、及びンデックスｊを必要とするが（１１１０）、ｊは反復回数を２で除算したものである。システムは、インデックスｉをインデックスｊと比較して所要の反復回数が終了したか否かをチェックする（１１２０）。両者が等しくない場合、プロセスは終了しておらず追加の反復が必要である。次に、システムは、現在のラウンドｉに関するラウンドパスワード検証子を生成する（１１３０）。ラウンドパスワード検証子ＰＶｉの生成は、暗号化アルゴリズム、先行するラウンドＰＶｉ－１からのパスワード検証子（現在にラウンドが１ではない場合）、及びサルト値Ｓを用いて行われる（１１３０）。最初のラウンドに関しては先行するラウンド検証子が存在しないので、パスワードＰを使用してPV_0を初期設定する（１１１０）。次に、システムはインデックスをインクリメントして（１１３０）、このインデックスを反復回数と比較して所要の反復回数が終了したか否かをチェックする（１１２０）。インデックスｉがインデックスｊと一致する場合、プロセスは部分的なパスワード証明書を記憶してコンピュータデバイスをリブートする（１１４０）。

【００３９】

コンピュータデバイスがリブートすると、ファームウェアは残りの反復を演算して完全なパスワード証明書をもたらす。反復の半分は既に遂行されているので、インデックスｉはｊ＋１に初期化される（１２１０）。システムは、インデックスｉと反復回数ｎとを比較して、所要の反復回数が終了したか否かをチェックする（１２２０）。両者が一致しない場合、プロセスは終了しておらず追加の反復が必要である。次に、システムは、現在のラウンドｉに関するラウンドパスワード検証子を生成する（１２３０）。ラウンドパスワード検証子ＰＶｉの生成は、暗号化アルゴリズム、先行するラウンドＰＶｉ－１からのパスワード検証子、及びサルト値Ｓを用いて行われる（１２３０）。次に、システムはインデックスをインクリメントして（１２３０）、このインデックスを反復回数と比較して所要の反復回数が終了したか否かをチェックする（１２２０）。インデックスｉが反復回数と等しい場合、プロセスは終了し、システムは完全なパスワード証明書ＰＶｎを出力する（１２４０）。

【００４０】

図１３は、パスワード証明書の生成方法を示す。部分的なパスワード証明書は、元のパスワードではないがパスワードの派生物なので安全である。システム１００は、これを何らかのプロセス（悪意のあるプロセスも含む）で読み出せるようにＮＶＲＡＭ１３１０に記憶する。プロセスは、中間点での証明書を得るために元のパスワードを保持する必要がある（１３２０）。次に、ファームウェアは、残りに反復を繰り返して、証明書が元のパスワードを所持するプロセスから来たことを確認することができる（１３３０）。このプロセス自体は、一定時間に到達するまでループ内でＨＭＡＣ反復を実行し、新しい高速ハードウェアに容易に適合するので、ハードウェアが改良された場合の自動改善に役立つ。高速ハードウェアは、単純に多数の反復を実行して同様の性能特性を実現することができる。反復回数は、所望の実行性能、安全性、及び／又は他の特性閾値に基づいて選択又は決定することができる。

【００４１】

図１４は、フルディスク暗号化の無効化方法を示す。システムは、耐タンパー性ブート処理を無効にする権限を与えられたユーザのリストを生成し（１４１０）、このリストをファームウェアへ送る（１４２０）。システムは、各ユーザに関するパスワード検証子を生成し、この検証子をソフトウェア１３４０としてＮＶＲＡＭ１３１０に記憶する（１４３０）。システムがユーザについてのパスワード証明書に関するコマンドを送出する場合

、システムはパスワード証明書を認証コマンドインタフェース（ＡＣＩ）のための「ドロップボックス」に入れ、ここではＮＶＲＡＭ１３１０は次のブートでコマンドを読み出す。ＮＶＲＡＭはオペレーティングシステム及び全てのユーザが読み出し可能なので、いずれかのユーザはこのＮＶＲＡＭ内のコマンドを読むことができ、パスワード証明書を盗むことができる。従来、サルトはこの問題を解決するために使用されている。しかしながら、これを実現するのは容易ではない。従って、システムは、マシンが最初にブートする場合に、システムサルトを生成して（１４４０）、システムサルトをＮＶＲＡＭに記憶する（１４５０）。ＮＶＲＡＭに記憶されたシステムサルトは、ＡＣＩインタフェースを経由してファームウェアに送出される次のコマンドに利用される。ファームウェアは、コマンドが送出される度に新しいランダムサルトをＮＶＲＡＭに入れる。

【００４２】

本明細書で説明する手法で実現できる１つの新しい結果は、フルディスク暗号化可能システムにおけるより安全なブート処理であり、ブート信頼チェーンの無許可の変更を検出できる。この手法の他の結果は、最小数のシステムリブートでこの安全なブートを有効及び／又は無効にできることである。

【００４３】

本開示の範囲内の実施形態は、記憶されるコンピュータ実行可能命令又はデータ構造を実行するか又は格納する有形の及び／又は非一時的なコンピュータ可読記憶媒体を含むことができる。そのような非一時的なコンピュータ可読記憶媒体は、汎用コンピュータ、又は前述の任意の専用プロセッサの機能デザインを含む専用コンピュータによりアクセスされる任意の使用可能な媒体とすることができる。例示的に、限定されるものではないが、そのような非一時的なコンピュータ可読記憶媒体は、ＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ、ＣＤ－ＲＯＭ、或いは他の光ディスク記憶装置、磁気ディスク記憶装置、又は他の磁気記憶装置、或いはコンピュータが実行可能な命令、データ構造、又はプロセッサチップ設計の形態の所望のプログラムコード手段を保持、或いは格納するために使用される他の任意の媒体を含む。情報がネットワーク、又は別の通信接続（ハードワイヤード、無線、又はそれらの組合せのいずれか）を介してコンピュータに転送、又は提供される場合、コンピュータは、正しく接続をコンピュータ可読媒体として見なす。従って、そのような任意の接続を正しくコンピュータ可読媒体と呼ぶ。前述の組合せは、コンピュータ可読媒体の範囲内にも含まれるべきである。

【００４４】

例えば、コンピュータが実行可能な命令は、ある特定の機能、又は機能のグループを汎用コンピュータ、専用コンピュータ、或いは専用処理装置に実行させる命令及びデータを含む。コンピュータが実行可能な命令は、スタンドアロン環境、又はネットワーク環境においてコンピュータにより実行されるプログラムモジュールを更に含む。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、或いは特定の抽象データ型を実現する専用プロセッサの設計に固有のルーチン、プログラム、構成要素、データ構造、オブジェクト及び機能等を含む。コンピュータが実行可能な命令、関連したデータ構造及びプログラムモジュールは、本明細書において開示した方法のステップを実行するプログラムコード手段の例を示す。そのような実行可能な命令、または関連したデータ構造の特定のシーケンスは、そのようなステップにおいて説明した機能を実現する対応する動作の例を示す。

【００４５】

本発明の他の実施形態がパーソナルコンピュータ、ハンドヘルド装置、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購買者向けの電子機器、またはプログラム可能な購買者向けの電子機器、ネットワークＰＣ、ミニコンピュータ及びメインフレームコンピュータ等を含む多くの種類のコンピュータシステム構成でネットワークコンピューティング環境において実施されることは、当業者により理解されるだろう。実施形態は、通信ネットワークを介してリンクされる（ハードワイヤードリンク、無線リンク、又はそれらの組合せのいずれかにより）ローカル処理装置及びリモート処理装置によりタスクが実行される分散コンピューティング環境において更に実施される。分散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、ローカルメモリ記憶装置及びリモートメモリ記憶装置の双方に配置される。

【００４６】

前述の種々の実施形態は例示的であり、本開示の範囲を限定すると解釈すべきではない。例えば、本明細書の原理は、ディスクトップコンピュータ、ラップトップコンピュータ、携帯デバイス、又は安全で耐タンパー性のブート処理が望まれる、ボリュームキーといった適切な共有秘密キーを有する任意の他のコンピュータ環境に適用することができる。当業者であれば、本明細書に記載して説明される例示的な実施形態及び応用例に追従することなく、及び本開示の精神及び範囲から逸脱することなく、種々の変形及び変更を本明細書に記載の原理から行い得ることを容易に理解できるはずである。

【図１】

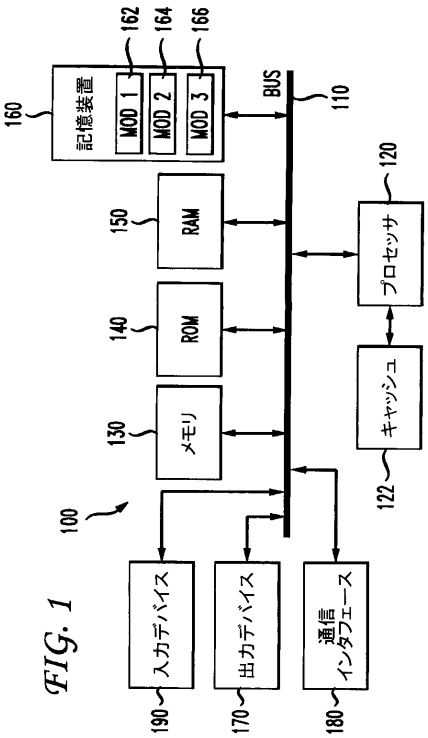


FIG. 1

【図２】

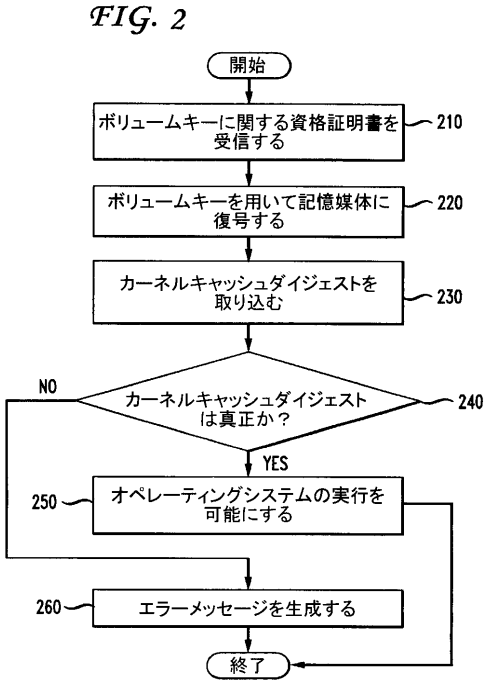
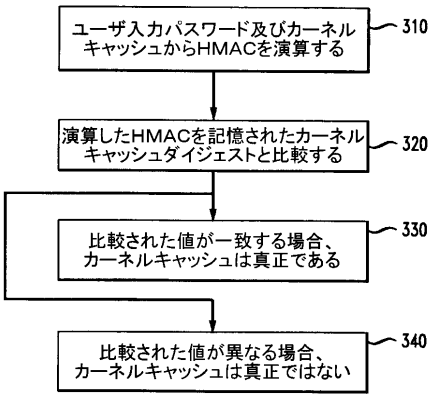


FIG. 2

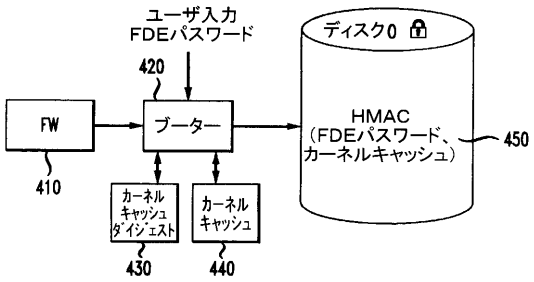
【図３】

FIG. 3



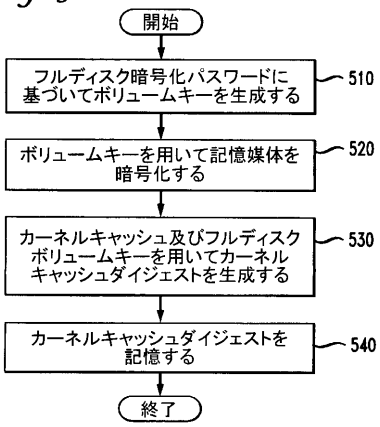
【図４】

FIG. 4



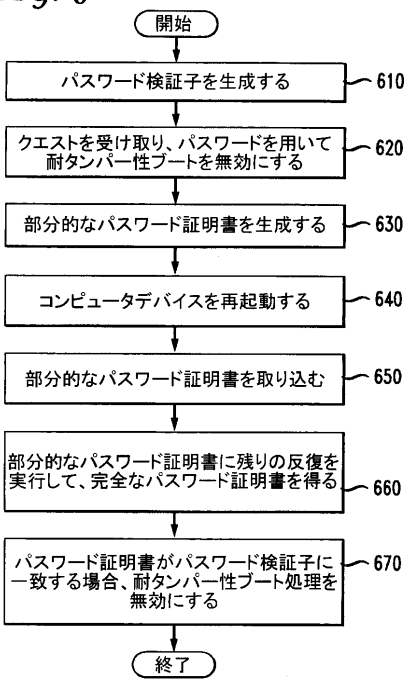
【図５】

FIG. 5



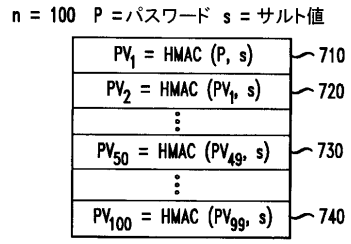
【図６】

FIG. 6



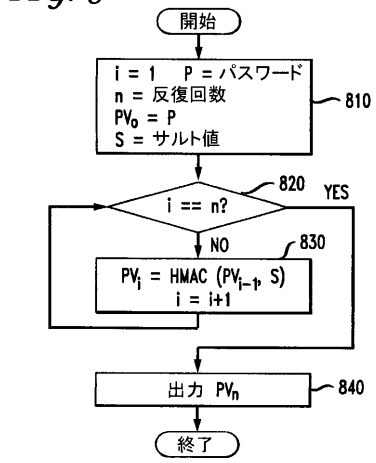
【図７】

FIG. 7



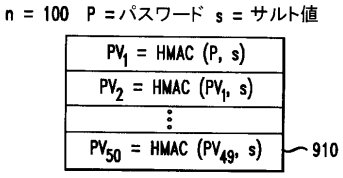
【図８】

FIG. 8



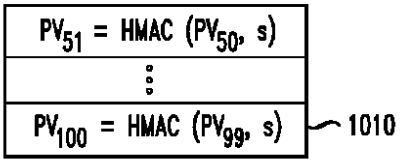
【図９】

FIG. 9



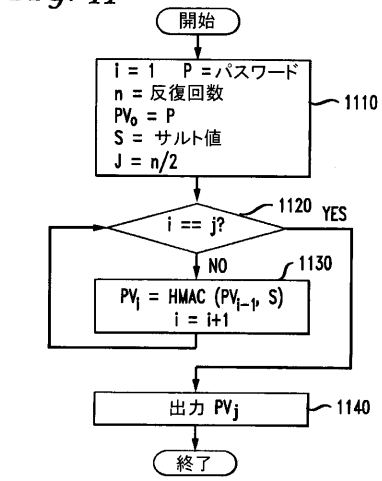
【図１０】

FIG. 10



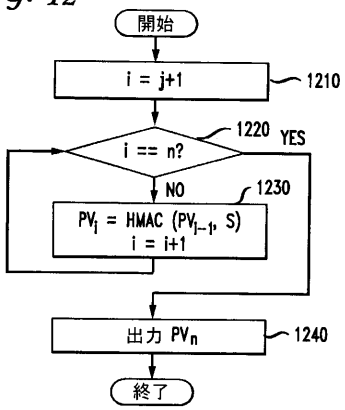
【図１１】

FIG. 11



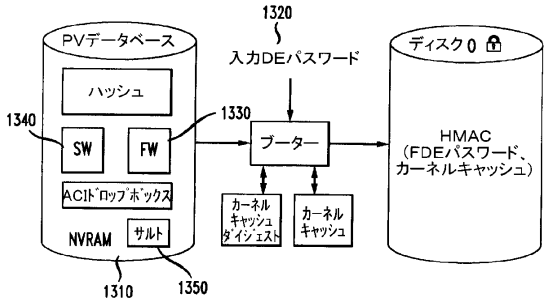
【図１２】

FIG. 12



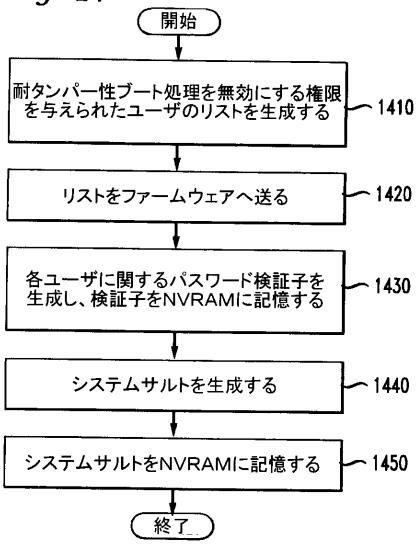
【図１３】

FIG. 13



【図１４】

FIG. 14



フロントページの続き

(72)発明者　クルスティック　イヴァン
アメリカ合衆国　カリフォルニア州　９４１１７　サンフランシスコ　フルトン　ストリート　９８８　アパートメント　３１９
(72)発明者　イーヴン　ジョエル
アメリカ合衆国　カリフォルニア州　９５１１０　サン　ホセ　パーク　アベニュー　４１１　ユニット　１３６

審査官　宮司　卓佳

(56)参考文献　特開２００６－３２３８１４（ＪＰ，Ａ）
特表２０１０－５１１２０９（ＪＰ，Ａ）
米国特許第０７３６００７３（ＵＳ，Ｂ１）
特開２０１０－２３７９７４（ＪＰ，Ａ）
特開２０１０－１８２１９６（ＪＰ，Ａ）
特開２００４－２６５２８６（ＪＰ，Ａ）
特表２０１１－５２７７７７（ＪＰ，Ａ）
特開２０００－３５７１５６（ＪＰ，Ａ）
特開平１０－３３３９０２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ２１／００－２１／８８