(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6706278号
(P6706278)
(45)発行日　令和2年6月3日(2020.6.3)　(24)登録日　令和2年5月19日(2020.5.19)

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

請求項の数 13　(全 13 頁)

(21)出願番号　特願2018-60737(P2018-60737)
(22)出願日　平成30年3月27日(2018.3.27)
(65)公開番号　特開2019-175000(P2019-175000A)
(43)公開日　令和1年10月10日(2019.10.10)
審査請求日　平成31年3月22日(2019.3.22)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】情報処理装置、及び情報処理方法

(57)【特許請求の範囲】
【請求項１】
複数のモジュールを順次起動する情報処理装置であって、
ブートプログラムモジュールと、
第１のモジュールと、
第２のモジュールと、
第３のモジュールと
を備え、
前記第１のモジュールは検証された前記第２のモジュールを起動し、前記第２のモジュールは検証された第３のモジュールを起動し、
前記第１のモジュールは、
前記第２のモジュール及び前記第３のモジュールの両方を検証するために使用される検証情報を有し、
前記検証情報を用いて前記第２のモジュールを検証し、かつ、前記検証情報を用いて前記第３のモジュールを検証し、
前記第１、第２、及び第３のモジュールの各モジュールは、
自身の署名を予め記憶する記憶手段と、
次に起動するモジュールの署名の検証が成功すると、該次に起動するモジュールを起動する起動手段と
を備え、

前記第１のモジュールは、
前記第１のモジュールの前記記憶手段に記憶された前記検証情報を用いた前記第２のモジュールの前記検証に基づき、前記第２のモジュールの改ざんを検知し、
前記第１のモジュールの前記記憶手段に記憶された前記検証情報を用いた前記第３のモジュールの前記検証に基づき、前記第３のモジュールの改ざんを検知し、
前記ブートプログラムモジュールは、
前記第１のモジュールの署名を検証するために使用される検証情報を記憶する手段と、
前記第１のモジュールの署名を検証する検証情報を用いて、該第１のモジュールの改ざんを検知する手段と、
前記第１のモジュールの署名の検証が成功すると、該第１のモジュールを起動する手段と
を備えることを特徴とする情報処理装置。
【請求項２】
前記第２のモジュールは前記第２のモジュールの署名を有し、前記第３のモジュールは前記第３のモジュールの署名を有し、
前記第１のモジュールは、
前記第２のモジュールが有する前記署名と、前記第１のモジュールが有する前記検証情報とを用いて、前記第２のモジュールを検証し、
前記第３のモジュールが有する前記署名と、前記第１のモジュールが有する前記検証情報とを用いて、前記第３のモジュールを検証することを特徴とする請求項１に記載の情報処理装置。
【請求項３】
前記第３のモジュールは、
前記第３のモジュールの前記記憶手段に記憶された検証情報と、前記第３のモジュールの次に起動されるモジュールの署名との両方を用いて、前記第３のモジュールの次に起動されるモジュールを検証することを特徴とする請求項１又は２に記載の情報処理装置。
【請求項４】
前記第１のモジュールが該第１のモジュール前記記憶手段に記憶された前記検証情報を用いた前記第３のモジュールの前記検証に基づき、前記第３のモジュールの改ざんを検出すると、前記第２のモジュールは、前記第３のモジュールを起動しないことを特徴とする請求項１乃至３の何れか１項に記載の情報処理装置。
【請求項５】
前記第１のモジュールはＢＩＯＳ（Ｂａｓｉｃ Ｉｎｐｕｔ/Ｏｕｔｐｕｔ Ｓｙｓｔｅｍ）であり、前記第２のモジュールはローダーであり、前記第３のモジュールはカーネルであることを特徴とする請求項１乃至４の何れか１項に記載の情報処理装置。
【請求項６】
前記ＢＩＯＳはＲＯＭ（Ｒｅａｄ-Ｏｎｌｙ Ｍｅｍｏｒｙ）に保存され、前記ローダー、及び前記カーネルはフラッシュメモリに保存されることを特徴とする請求項５に記載の情報処理装置。
【請求項７】
前記第２のモジュールは、ユーザ入力に従って起動するカーネルを切り替えて起動し、
前記第１のモジュールは、前記ユーザ入力に従って起動可能な複数のカーネルのそれぞれの改ざんを検証することを特徴とする請求項５に記載の情報処理装置。
【請求項８】
各モジュールを制御するメインコントローラと、
前記メインコントローラとは別にプロセッサ及びメモリを有し、前記第１のモジュールの起動を制御する埋め込みコントローラとをさらに備えることを特徴とする請求項１乃至７の何れか１項に記載の情報処理装置。
【請求項９】
通知手段をさらに備え、

前記第１のモジュールは前記検証が成功した前記第２のモジュールを起動し、前記第２のモジュールは前記検証が成功した前記第３のモジュールを起動し、
前記通知手段は、前記第２のモジュールの検証が失敗したか、又は、前記第３のモジュールの検証が失敗した場合に、ユーザにエラーを通知することを特徴とする請求項１乃至８の何れか１項に記載の情報処理装置。
【請求項１０】
前記通知手段は、表示部であることを特徴とする請求項９に記載の情報処理装置。
【請求項１１】
前記第１のモジュールの検証が失敗した場合に、ユーザにエラーを通知する他の通知手段をさらに備えることを特徴とする請求項１０に記載の情報処理装置。
【請求項１２】
前記他の通知手段は、ＬＥＤであることを特徴とする請求項１１に記載の情報処理装置。
【請求項１３】
ブートプログラムモジュールと、第１のモジュールと、第２のモジュールと、第３のモジュールとを備える情報処理装置の情報処理方法であって、
前記ブートプログラムモジュールによって前記第１のモジュールを検証する工程と、
前記ブートプログラムモジュールによって、前記ブートプログラムモジュールによって検証された前記第１のモジュールを起動する工程と、
前記第１のモジュールによって第２のモジュールを検証する工程と、
前記第１のモジュールによって、該第１のモジュールによって検証された前記第２のモジュールを起動する工程と、
前記第１のモジュールによって第３のモジュールを検証する工程と、
前記第２のモジュールによって、該第１のモジュールによって検証された前記第３のモジュールを起動する工程と、
を含み、
前記第１、第２、及び第３のモジュールの各モジュールは、自身の署名を予め記憶しており、
前記第１のモジュールは、
前記第２のモジュール及び前記第３のモジュールの両方を検証するために使用される検証情報を有し、
前記第２のモジュールの署名と前記検証情報とを用いて前記第２のモジュールを検証して前記第２のモジュールの改ざんを検知し、かつ、前記第３のモジュールの署名と前記検証情報を用いて前記第３のモジュールを検証して前記第３のモジュールの改ざんを検知し、
前記ブートプログラムモジュールは、
前記第１のモジュールの署名を検証するために使用される検証情報を記憶し、
前記第１のモジュールの署名と前記ブートプログラムモジュールが記憶する検証情報とを用いて前記第１のモジュールを検証して前記第１のモジュールの改ざんを検知し、
前記第１のモジュールの署名の検証が成功すると、該第１のモジュールを起動することを特徴とする情報処理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、情報処理装置、及び情報処理方法に関する。
【背景技術】
【０００２】
コンピュータシステムにおけるソフトウェアの脆弱性をついて、ソフトウェアを改ざんし、コンピュータを悪用する攻撃が問題となっている。これらの攻撃対策として、プログラムに署名を施して保存しておき、起動するたびにプログラムの署名を検証することで改

ざんの有無を検知する方法が考えられている。

【０００３】

特許文献１では、プログラムを部分的に交換可能にするためにモジュール化し、各モジュールに対して署名を検証する処理や検証するのに必要な鍵情報をそれぞれに内包して格納する方法が提案されている。

【先行技術文献】

【特許文献】

【０００４】

【特許文献１】米国特許第２０１４／００８９６５１号明細書

【発明の概要】

【発明が解決しようとする課題】

【０００５】

しかしながら、上記従来技術には以下に記載する課題がある。例えば、上記従来技術では各モジュールの署名に用いるアルゴリズムや鍵情報が同一である場合、同一の署名検証処理や鍵情報を各モジュールにおいて重複して格納することとなる。このため、メモリ資源等の利用効率が低く、サイズやコストを縮小するデバイスにおいては、貴重なメモリ資源を有効に利用できていないのが現状である。

【０００６】

本発明は、上述の問題の少なくとも一つに鑑みて成されたものであり、メモリ資源を有効に利用しつつ、起動時にシステムの改ざんを検知する仕組みを提供することを目的とする。

【０００７】

本発明のさらなる別の目的は、より堅牢な、システムにおける改ざんを検知する仕組みを提供することにある。

【課題を解決するための手段】

【０００８】

本発明は、例えば、複数のモジュールを順次起動する情報処理装置であって、ブートプログラムモジュールと、第１のモジュールと、第２のモジュールと、第３のモジュールとを備え、前記第１のモジュールは検証された前記第２のモジュールを起動し、前記第２のモジュールは検証された第３のモジュールを起動し、前記第１のモジュールは、前記第２のモジュール及び前記第３のモジュールの両方を検証するために使用される検証情報を有し、前記検証情報を用いて前記第２のモジュールを検証し、かつ、前記検証情報を用いて前記第３のモジュールを検証し、前記第１、第２、及び第３のモジュールの各モジュールは、自身の署名を予め記憶する記憶手段と、次に起動するモジュールの署名の検証が成功すると、該次に起動するモジュールを起動する起動手段とを備え、前記第１のモジュールは、前記第１のモジュールの前記記憶手段に記憶された前記検証情報を用いた前記第２のモジュールの前記検証に基づき、前記第２のモジュールの改ざんを検知し、前記第１のモジュールの前記記憶手段に記憶された前記検証情報を用いた前記第３のモジュールの前記検証に基づき、前記第３のモジュールの改ざんを検知し、前記ブートプログラムモジュールは、前記第１のモジュールの署名を検証するために使用される検証情報を記憶する手段と、前記第１のモジュールの署名を検証する検証情報を用いて、該第１のモジュールの改ざんを検知する手段と、前記第１のモジュールの署名の検証が成功すると、該第１のモジュールを起動する手段とを備えることを特徴とする。

【発明の効果】

【０００９】

本発明によれば、メモリ資源を有効に利用しつつ、起動時にシステムの改ざんを検知することができる。また、本発明の他の側面によれば、より堅牢な、システムにおける改ざんを検知する仕組みを提供することができる。

【図面の簡単な説明】

【００１０】

【図１】一実施形態に係る複合機のハードウェア構成図。
【図２】一実施形態に係る複合機のソフトウェア構成図。
【図３】一実施形態に係る起動時の動作を示す模式図。
【図４】一実施形態に係る処理手順を示すフローチャート。
【図５】一実施形態に係る処理手順を示すフローチャート。
【発明を実施するための形態】
【００１１】
以下に本発明の一実施形態を示す。以下で説明される個別の実施形態は、本発明の上位概念、中位概念及び下位概念など種々の概念を理解するために役立つであろう。また、本発明の技術的範囲は、特許請求の範囲によって確立されるのであって、以下の個別の実施形態によって限定されるわけではない。なお、実施形態に係る情報処理装置として複合機（デジタル複合機／ＭＦＰ／Ｍｕｌｔｉ　Ｆｕｎｃｔｉｏｎ　Ｐｅｒｉｐｈｅｒａｌ）を例に説明する。しかしながら適用範囲は複合機に限定はせず、情報処理装置であればよい。
【００１２】
＜第１の実施形態＞
以下では、添付図面を参照して、本発明の第１の実施形態について説明する。なお、以下の実施形態は特許請求の範囲に係る本発明を限定するものでなく、また本実施形態で説明されている特徴の組み合わせの全てが本発明の解決手段に必須のものとは限らない。また、実施形態に係る情報処理装置として複合機（デジタル複合機／ＭＦＰ／Ｍｕｌｔｉ　Ｆｕｎｃｔｉｏｎ　Ｐｅｒｉｐｈｅｒａｌ）を例に説明する。しかしながら適用範囲は複合機に限定はせず、情報処理装置であればよい。
【００１３】
＜情報処理装置のハードウェア構成＞
まず、図１を参照して、本実施形態に係る情報処理装置である複合機１００と埋め込みコントローラ（Ｅｍｂｅｄｄｅｄ　Ｃｏｎｔｒｏｌｌｅｒ）１１３のハードウェア構成を説明する。複合機１００は、ＣＰＵ１０１、ＲＯＭ（Ｒｅａｄ-Ｏｎｌｙ Ｍｅｍｏｒｙ）１０２、ＲＡＭ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｍｅｍｏｒｙ）１０３、ＨＤＤ（Ｈａｒｄ Ｄｉｓｋ Ｄｒｉｖｅ）１０４、ネットワークＩ／Ｆ制御部１０５、スキャナＩ／Ｆ制御部１０６、プリンタＩ／Ｆ制御部１０７、パネル制御部１０８、スキャナ１１１、プリンタ１１２、埋め込みコントローラ１１３、フラッシュメモリ１１４、及びＬＥＤ１１７を備える。また、埋め込みコントローラ１１３は、ＣＰＵ１１５及びＲＡＭ１１６を備える。
【００１４】
ＣＰＵ１０１は、複合機１００のソフトウェアプログラムを実行し、装置全体の制御を統括的に行う。ＲＯＭ１０２はリードオンリーメモリであり、複合機１００のＢＩＯＳ（Ｂａｓｉｃ Ｉｎｐｕｔ/Ｏｕｔｐｕｔ Ｓｙｓｔｅｍ）、固定パラメータ等を格納している。ＲＡＭ１０３はランダムアクセスメモリであり、ＣＰＵ１０１が複合機１００を制御する際に、プログラムや一時的なデータの格納などに使用される。ＨＤＤ１０４はハードディスクドライブであり、一部のアプリケーションや、各種データを格納する。フラッシュメモリ１１４は、ローダー、カーネル、アプリケーションなどの各種モジュールを格納する。
【００１５】
埋め込みコントローラ１１３のＣＰＵ１１５は、埋め込みコントローラ１１３のソフトウェアプログラムを実行し、複合機１００における一部の制御を行う。ＲＡＭ１１６はランダムアクセスメモリであり、ＣＰＵ１１５が複合機１００を制御する際に、プログラムや一時的なデータの格納などに使用される。複合機１００は、埋め込みコントローラ１１３に対して、システムを統括的に制御するメインコントローラを備える。当該メインコントローラは、少なくともＣＰＵ１０１、ＲＯＭ１０２、及びＲＡＭ１０３を含んで構成される。

【００１６】
ネットワークＩ／Ｆ制御部１０５は、ネットワーク１１８とのデータの送受信を制御する。スキャナＩ／Ｆ制御部１０６は、スキャナ１１１による原稿の読み取り制御する。プリンタＩ／Ｆ制御部１０７は、プリンタ１１２による印刷処理などを制御する。パネル制御部１０８は、タッチパネル式の操作パネル１１０を制御し、各種情報の表示、使用者からの指示入力を制御する。バス１０９は、ＣＰＵ１０１、ＲＯＭ１０２、ＲＡＭ１０３、ＨＤＤ１０４、ネットワークＩ／Ｆ制御部１０５、スキャナＩ／Ｆ制御部１０６、及びプリンタＩ／Ｆ制御部１０７を相互に接続する。さらに、バス１０９は、パネル制御部１０８、埋め込みコントローラ１１３、及びフラッシュメモリ１１４も相互に接続する。このバス１０９を介して、ＣＰＵ１０１からの制御信号や各装置間のデータ信号が送受信される。ＬＥＤ１１７は必要に応じて点灯し、ソフトウェアやハードウェアの異常を外部に伝えるために利用される。
【００１７】
＜情報処理装置のソフトウェア構成＞
次に、図２（ａ）を参照して、本実施形態に係る複合機１００が有するソフトウェアモジュールを説明する。複合機１００は、ソフトウェアモジュールとして、埋め込みコントローラ１１３内にブートプログラム２０９を含む。さらに、複合機１００は、ＢＩＯＳ２１０、ローダー２１１、カーネル２１２、Ｎａｔｉｖｅプログラム２１３、Ｊａｖａ（登録商標）プログラム２１４、ＵＩ制御部２０３、及び通信管理部２０７を含む。
【００１８】
通信管理部２０７は、ネットワーク１１８に接続されるネットワークＩ／Ｆ制御部１０５を制御して、ネットワーク１１８を介して外部とデータの送受信を行う。ＵＩ制御部２０３は、パネル制御部１０８を介して操作パネル１１０への入力を受け取り、入力に応じた処理や操作パネル１１０への画面出力を行う。
【００１９】
ブートプログラム２０９は複合機１００の電源を入れると埋め込みコントローラ１１３のＣＰＵ１１５で実行されるプログラムであり、起動に関わる処理を実行するほかにＢＩＯＳの改ざん検知を行うＢＩＯＳ改ざん検知部２０１を含む。ＢＩＯＳ２１０はブートプログラム２０９の実行後にＣＰＵ１０１で実行されるプログラムであり、起動に関わる処理を実行するほかにローダー２１１及びカーネル２１２の改ざん検知を行うローダー／カーネル改ざん検知部２０２を含む。
【００２０】
ローダー２１１はＢＩＯＳ２１０の処理が完了した後にＣＰＵ１０１で実行されるプログラムであり、起動に関わる処理を実行する。カーネル２１２はローダー２１１の処理が完了した後にＣＰＵ１０１で実行されるプログラムであり、起動に関わる処理を実行するほかにＮａｔｉｖｅプログラム２１３の改ざん検知を行うＮａｔｉｖｅ改ざん検知部２０５を有する。
【００２１】
Ｎａｔｉｖｅプログラム２１３はＣＰＵ１０１で実行されるプログラムであり、複合機１００のＪａｖａプログラム２１４と連携して各機能を提供する複数のプログラムからなる。例えば、スキャナＩ／Ｆ制御部１０６やプリンタＩ／Ｆ制御部１０６を制御するプログラムや起動プログラムなどである。カーネル２１２によってＮａｔｉｖｅプログラムの中から起動プログラムが呼び出され、起動処理を実行する。またプログラムの中の一つとしてＪａｖａプログラムの改ざん検知を行うＪａｖａプログラム改ざん検知部を有する。
【００２２】
Ｊａｖａプログラム２１４はＣＰＵ１０１で実行されるプログラムであり、複合機１００のＮａｔｉｖｅプログラム２１３と連携して各機能を提供するプログラムである。例えば、操作パネル１１０に画面を表示するプログラムなどがある。
【００２３】
＜起動手順＞

以下では、図３（ａ）及び図３（ｂ）を参照して、複合機１００の起動手順について説明する。図３（ａ）は、改ざん検知を行わずに複合機１００が起動する順序を示す。ブートプログラム２０９がＢＩＯＳ２１０を起動し、ＢＩＯＳ２１０がローダー２１１を起動し、ローダー２１１がカーネル２１２を起動し、カーネル２１２がＮａｔｉｖｅプログラム２１３の中から起動プログラムを起動する。起動プログラムの中でＪａｖａプログラム２１４が起動され、以降はＮａｔｉｖｅプログラム２１３とＪａｖａプログラム２１４が連携して複合機１００の機能を提供する。このように各モジュールは所定の順序で起動制御が行われ、前のモジュールの起動が完了すると次のモジュールの起動処理が実行される。

【００２４】

図３（ｂ）は、改ざん検知を行いながら複合機１００が起動する順序を示す。図示するように、ブートプログラム２０９から、ＢＩＯＳ２１０、ローダー２１１、カーネル２１２、Ｎａｔｉｖｅプログラム２１３、Ｊａｖａプログラム２１４の順に改ざん検知を行いながら起動する。起動するモジュールの改ざん検知は、直前に起動されたモジュールが行う。例えば、ＢＩＯＳ２１０の改ざん検知は、ブートプログラム２０９が行う。また、図３（ｂ）は各プログラムの保存場所、デジタル署名（以下、署名と称する。）と署名を検証するための公開鍵（検証情報）の保存場所を表している。図３（ｂ）に示すように、署名については各モジュールが自身の署名を保存している。一方で、公開鍵については、所定のモジュール（第１のモジュールなど）は保存しているものの、保存していないモジュールも存在する。これは、本実施形態において、公開鍵が同一のモジュールについては所定のモジュールが保存し、当該モジュールにおいて複数の他のモジュールの改ざんの検知を連続的に行うためである。これにより、メモリ資源を有効に利用することができる。

【００２５】

以下では、ＲＯＭ１０２にブートプログラム２０９とＢＩＯＳ２１０が保存され、フラッシュメモリ１１４にローダー２１１とカーネル２１２とＮａｔｉｖｅプログラム（第１プログラム）２１３とが保存されているものとする。さらに、ＨＤＤ１０４にＪａｖａプログラム２１４（第２プログラム）が保存されているものとする。

【００２６】

ブートプログラム２０９にはＢＩＯＳの署名検証用の公開鍵３００が保存され、ＢＩＯＳ２１０にはＢＩＯＳの署名３０２とローダー／カーネル検証用の公開鍵３０３が保存される。ローダー２１１にはローダー署名３０４とカーネル検証用の公開鍵３０５が保存される。また、カーネル２１２にはカーネル署名３０６とＮａｔｉｖｅプログラム検証用の公開鍵３０７が保存され、Ｎａｔｉｖｅプログラム２１３にはＮａｔｉｖｅプログラムの署名３０８とＪａｖａプログラム検証用の公開鍵３０９が保存される。さらに、Ｊａｖａプログラム２１４には、Ｊａｖａプログラムの署名３１０が保存される。これらの公開鍵と署名は予め複合機１００の工場出荷前にプログラムに対して付与されることが望ましい。本実施形態に係る複合機１００においては、２０１、２０２、２０５、２０６の各検知部が、次に起動する各プログラム（各モジュール）を検証し、問題がなければ次のプログラムを起動することで改ざん検知を行う。

【００２７】

＜処理手順＞

次に、図４を参照して、本実施形態に係る複合機１００の起動時における処理手順を説明する。複合機１００に電源が投入されると、ＲＯＭ１０２からＲＡＭ１１６にブートプログラム２０９が読み込まれ、ＣＰＵ１１５によって実行される。

【００２８】

Ｓ４０１で、ブートプログラム２０９に含まれるＢＩＯＳ改ざん検知部２０１は、ＢＩＯＳの署名検証を行い、成功したか否かを判定する。具体的には、ＢＩＯＳ改ざん検知部２０１は、フラッシュメモリ１１４からＢＩＯＳ２１０とローダー／カーネル検証用の公開鍵３０３、ＢＩＯＳの署名３０２をＲＡＭ１１６に読み込む。さらに、ＢＩＯＳ改ざん検知部２０１は、ＢＩＯＳ検証用の公開鍵３００を用いてＢＩＯＳの署名３０２の検証を

行い成功したか判定する。署名の検証に失敗した場合、Ｓ４０３に進み、ＢＩＯＳ改ざん検知部２０１は、ＬＥＤ１１７を点灯させ、処理を終了する。一方、署名の検証に成功した場合、ＢＩＯＳ改ざん検知部２０１はＣＰＵ１０１に通電し、ブートプログラムの処理を終了する。その後、本フローチャートはＣＰＵ１０１によって実行されるＳ４０２以降の処理に移行する。

【００２９】

ＣＰＵ１０１は通電されると、Ｓ４０２で、フラッシュメモリ１１４からＢＩＯＳ２１０とローダー／カーネル検証用の公開鍵３０３をＲＡＭ１０３に読み込み、ＢＩＯＳ２１０を起動する。以降の処理は、全てＣＰＵ１０１によって処理されるものとして説明する。

【００３０】

ＢＩＯＳ２１０が起動されるとＳ４０４に移行する。Ｓ４０４で、ＢＩＯＳ２１０は、各種初期化処理を実行し、ＢＩＯＳ２１０に含まれるローダー／カーネル改ざん検知部２０２がフラッシュメモリ１１４からローダー２１１、ローダー署名３０４をＲＡＭ１０３に読み込む。さらに、ローダー／カーネル改ざん検知部２０２は、ローダー／カーネル検証用の公開鍵３０３を用いてローダー署名３０４の検証を行い成功したか判定する。署名の検証に失敗した場合は、Ｓ４１２に進み、ローダー／カーネル改ざん検知部２０２は、操作パネル１１０にエラーメッセージを表示し、起動を停止して処理を終了する。一方、署名の検証に成功した場合は、ローダー／カーネル改ざん検知部２０２はフラッシュメモリ１１４からカーネル２１２とＮａｔｉｖｅプログラム検証用の公開鍵３０７とカーネル署名３０６をＲＡＭ１０３に読み込んで、Ｓ４０５に進む。

【００３１】

Ｓ４０５で、ローダー／カーネル改ざん検知部２０２は、ローダー／カーネル検証用の公開鍵３０３を用いて、カーネル署名３０６の検証を行い成功したか判定する。署名の検証に失敗した場合はＳ４１２に進み、カーネル改ざん検知部２０４は操作パネル１１０にエラーメッセージを表示し、起動を停止して処理を終了する。一方、署名の検証に成功した場合、ローダー／カーネル改ざん検知部２０２は処理を終了し、Ｓ４０６に進む。

【００３２】

Ｓ４０６で、ＢＩＯＳ２１０は、ＲＡＭ１０３に読み込まれたローダー２１１を起動する。ローダー２１１は起動されると、各種初期化処理を実行し、フラッシュメモリ１１４からカーネル２１２とＮａｔｉｖｅプログラム検証用の公開鍵３０７とカーネル署名３０６をＲＡＭ１０３に読み込む。Ｓ４０７に進み、ローダー２１１はＲＡＭ１０３に読み込まれたカーネル２１２を起動する。

【００３３】

カーネル２１２が起動されるとＳ４０８に移行する。Ｓ４０８で、カーネル２１２は、各種初期化処理を実行する。さらに、カーネル２１２に含まれるプログラム改ざん検知部２０５がフラッシュメモリ１１４からＮａｔｉｖｅプログラム２１３とＪａｖａプログラム検証用の公開鍵３０８とＮａｔｉｖｅプログラムの署名３０９をＲＡＭ１０３に読み込む。プログラム改ざん検知部２０５は、Ｓ４０８でＮａｔｉｖｅプログラム検証用の公開鍵３０７を用いて、Ｎａｔｉｖｅプログラムの署名３０９の検証を行い成功したか判定する。署名の検証に失敗した場合、Ｓ４１２にてプログラム改ざん検知部２０５は操作パネル１１０にエラーメッセージを表示し、起動を停止して処理を終了する。一方、署名の検証に成功した場合、プログラム改ざん検知部２０５は処理を終了し、Ｓ４０９にてＮａｔｉｖｅプログラム２１３を起動する。

【００３４】

Ｎａｔｉｖｅプログラム２１３のうち、改ざん検知の処理を実行するＪａｖａプログラム改ざん検知部２０６が起動されるとＳ４１０に移行する。Ｓ４１０で、Ｊａｖａプログラム改ざん検知部２０６は、ＨＤＤ１０４からＪａｖａプログラム２１４とＪａｖａプログラムの署名３１０をＲＡＭ１０３に読み込む。Ｊａｖａプログラム改ざん検知部２０６はＳ４１０でＪａｖａプログラム検証用の公開鍵３０８を用いて、Ｊａｖａプログラムの

署名３１０の検証を行い成功したか判定する。署名の検証に失敗した場合はＳ４１２に進み、Ｊａｖａプログラム改ざん検知部２０６は、操作パネル１１０にエラーメッセージを表示し、起動を停止して処理を終了する。署名の検証に成功した場合、Ｊａｖａプログラム改ざん検知部２０５は処理を終了し、Ｓ４１１でＮａｔｉｖｅプログラム２１３はＪａｖａプログラム２１４を起動する。

【００３５】

以上説明したように、本実施形態に係る情報処理装置は、ブートプログラムの起動に続いて複数のモジュールを順次起動する。複数のモジュールの各モジュールは、自身の署名をメモリ等に予め記憶しており、次に起動するモジュールの署名の検証が成功すると、当該次に起動するモジュールを起動する。また、複数のモジュールのうち第１のモジュールは、次に起動する第２のモジュールの署名を検証する検証情報であって、メモリ等に予め記憶した検証情報を用いて、第２のモジュールの改ざんを検知する。さらに、第１のモジュールは、上記検証情報を用いて、第２のモジュールの後に起動する少なくとも１つの第３のモジュールの改ざんを検知する。さらに、第２のモジュールは、第１のモジュールによって起動されると、次に起動する第３のモジュールを起動する。このように、本実施形態によれば、各モジュールが自身の署名情報を予め記憶しておくものの、検証情報については、各モジュールが予め記憶する必要がない。つまり、所定のモジュール（第１のモジュール）が検証情報を記憶しておき、当該検証情報を用いて、一部の複数のモジュールの改ざんを検知する。従って、複数のモジュールの全てが検証情報を記憶する必要がなくなり、メモリ資源を有効に利用することができる。また、本実施形態によれば、より堅牢な、システムにおける改ざんを検知する仕組みを提供することができる。

【００３６】

＜第２の実施形態＞

以下では、本発明の第２の実施形態について説明する。図３（ｃ）に示すように、複合機１００に複数のカーネルや複数のプログラムを保持し、ローダーで起動するカーネルやプログラムを切り替える構成の場合もある。このような構成の場合、上記第１の実施形態の構成だとカーネル２１２以外のカーネルＢ２２０を起動しようとした場合、署名が存在せず、改ざんされていないにも関わらず改ざん検知してしまい、起動しないという問題がある。そこで、本実施形態では異なるカーネル、異なるプログラムを保持する構成であっても改ざん検知して起動する方法について説明する。

【００３７】

＜ソフトウェア構成＞

まず、図２（ｂ）を参照して、本実施形態に係る複合機１００が有するソフトウェアモジュールの構成例を説明する。２０１～２１４は図２（ａ）と同等であるため説明を省略する。

【００３８】

ローダー２２３はＢＩＯＳ２１０の処理が完了した後にＣＰＵ１０１で実行されるプログラムであり、起動に関わる処理を実行するほかに、操作パネル１１０を介したユーザ入力に従って起動するカーネルを切り替える。

【００３９】

カーネルＢ２２０は、ＣＰＵ１０１で実行されるカーネル２１２とは異なるプログラムであり、起動に関わる処理を実行するほかにＮａｔｉｖｅプログラムＢ２２２の改ざん検知を行うプログラム改ざん検知部Ｂ２２１を有する。ＮａｔｉｖｅプログラムＢ２２２は、ＣＰＵ１０１で実行されるプログラムであり、複合機１００のアップデート機能を提供する。カーネルＢ２２０によって呼び出されて、カーネル２１２やＮａｔｉｖｅプログラム２１３、Ｊａｖａプログラム２１４をアップデートする機能を提供する。なお、ＮａｔｉｖｅプログラムＢ２２２はアップデート機能に限らず、ほかの機能を提供するプログラムであってもよい。

【００４０】

＜起動手順＞

次に、図３（ｄ）を参照して、ローダー２２３によって起動するカーネルがカーネル２１２か、又はカーネルＢ２２０かによって改ざん検知される対象が切り替わって起動する処理の流れを説明する。

【００４１】

ローダー２２３には、ローダー２２３の署名３０４が含まれているものとする。さらに、カーネルＢ２２０には、カーネルＢの署名３４１とＮａｔｉｖｅプログラムＢ検証用の公開鍵３４２が含まれ、ＮａｔｉｖｅプログラムＢ２２２にはＮａｔｉｖｅプログラムＢの署名３４３が含まれる。これらの公開鍵と署名は予め複合機１００の工場出荷前にプログラムに対して付与されることが望ましい。このように、ローダー２２３には、次に起動可能な複数のカーネルそれぞれの公開鍵（検証情報）が含まれる。

【００４２】

＜処理手順＞

次に、図５を参照して、本実施形態に係る複合機１００の起動時における処理手順を説明する。複合機１００に電源が投入されると、ＲＯＭ１０２からＲＡＭ１１６にブートプログラム２０９が読み込まれ、ＣＰＵ１１５によって実行される。Ｓ４０１～Ｓ４０５の処理は図４と同等であるため説明を省略する。

【００４３】

Ｓ４０５に続くＳ６００で、ローダー／カーネル改ざん検知部２０２は、ローダー／カーネル検証用の公開鍵３０３を用いて、カーネルＢ署名３４１の検証を行い成功したか判定する。署名の検証に失敗した場合、Ｓ４１２に処理を遷移する。署名の検証に成功した場合、ローダー／カーネル改ざん検知部２０２は処理を終了し、Ｓ４０６でＢＩＯＳ２１０がＲＡＭ１０３に読み込まれたローダー２１１を起動する。

【００４４】

Ｓ４０６でローダー２２３が起動されると、ローダー２２３は、各種初期化処理を実行する。続いて、Ｓ６０１で、ローダー２２３は、操作パネル１１０を介したユーザ入力に従って、カーネル２１２が起動対象として選択されたか否かを判定する。カーネル２１２が起動対象として選択された場合は、Ｓ４０７の処理に遷移する。以降のＳ４０７～Ｓ４１２の処理は図４と同等であるため説明を省略する。一方、カーネルＢ２２０が選択されるとＳ６０２に進み、ローダー２２３は、ＲＡＭ１０３に読み込まれたカーネルＢ２２０を起動する。

【００４５】

カーネルＢ２２０は起動されると、各種初期化処理を実行し、カーネルＢ２２０に含まれるプログラム改ざん検知部Ｂ２２１がフラッシュメモリ１１４からＮａｔｉｖｅプログラムＢ２２２とＮａｔｉｖｅプログラムＢの署名３４３をＲＡＭ１０３に読み込む。続いて、Ｓ６０３で、プログラム改ざん検知部Ｂ２２１は、ＮａｔｉｖｅプログラムＢ検証用の公開鍵３４２を用いて、ＮａｔｉｖｅプログラムＢの署名３４３の検証を行い成功したか判定する。署名の検証に失敗した場合、Ｓ４１２の処理に遷移する。一方、署名の検証に成功した場合、プログラム改ざん検知部Ｂ２２１は処理を終了し、Ｓ６０４で、カーネルＢ２２０はＮａｔｉｖｅプログラムＢ２２２を起動する。ＮａｔｉｖｅプログラムＢ２２２は起動するとアップデート機能をユーザに提供する。

【００４６】

以上説明したように、本実施形態によれば、複数のカーネル、複数のプログラムを保持する構成であってもそれらの改ざんを検知して起動することができ、上記第１の実施形態と同等の効果を奏することができる。

【００４７】

＜変形例＞

本発明は上記実施形態に限らず様々な変形が可能である。上記第１及び第２の実施形態では公開鍵が異なるものがあるとして説明したが、同じものがあってもよい。また、各プログラムの保存場所としてＲＯＭ１０２、フラッシュメモリ１１４、ＨＤＤ１０４があるものとして説明したが、保存場所を限定するものではなく、別の記憶媒体であってもよい

。またプログラムの保存場所が説明した箇所になくてもよく、例えばＲＯＭ１０２上にローダー２２３を記憶する構成であってもよい。

【００４８】

＜その他の実施形態＞

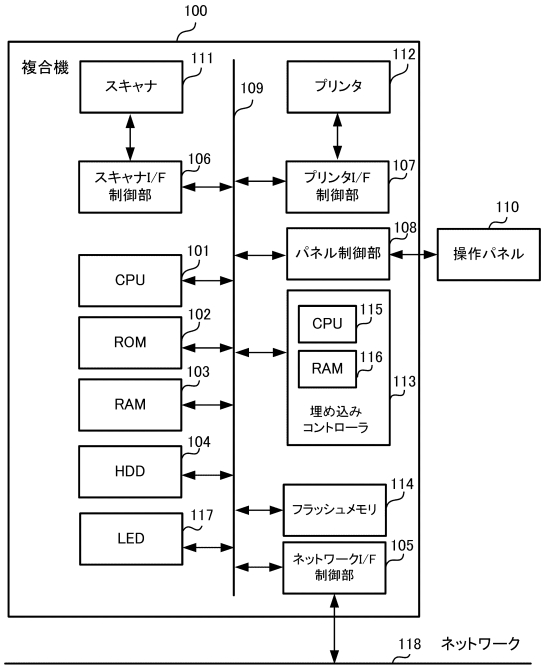
本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【符号の説明】

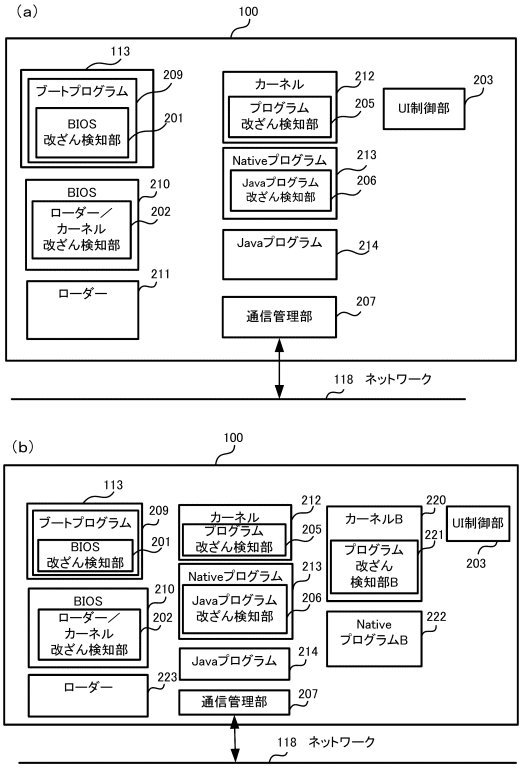
【００４９】

１００：複合機、１０１：ＣＰＵ、１０２：ＲＯＭ、１０３：ＲＡＭ、１０４：ＨＤＤ、１０５：ネットワークＩ／Ｆ制御部、１０６：スキャナＩ／Ｆ制御部、１０７：プリンタＩ／Ｆ制御部、１０８：パネル制御部、１０９：バス、１１０：操作パネル、１１１：スキャナ、１１２：プリンタ、１１３：埋め込みコントローラ、１１４：フラッシュメモリ、１１５：ＣＰＵ、１１６：ＲＡＭ、１１７：ＬＥＤ、１１８：ネットワーク、２０１：ＢＩＯＳ改ざん検知部、２０２：ローダー／カーネル改ざん検知部、２０５：プログラム改ざん検知部、２０６：Ｊａｖａプログラム改ざん検知部

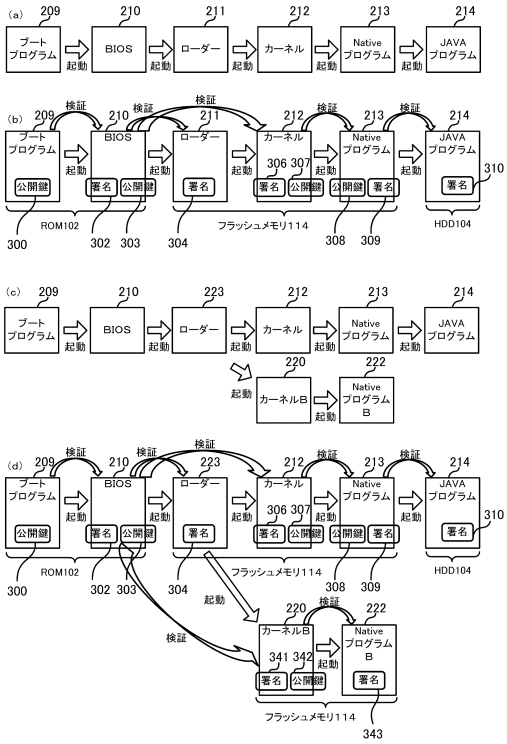
【図１】



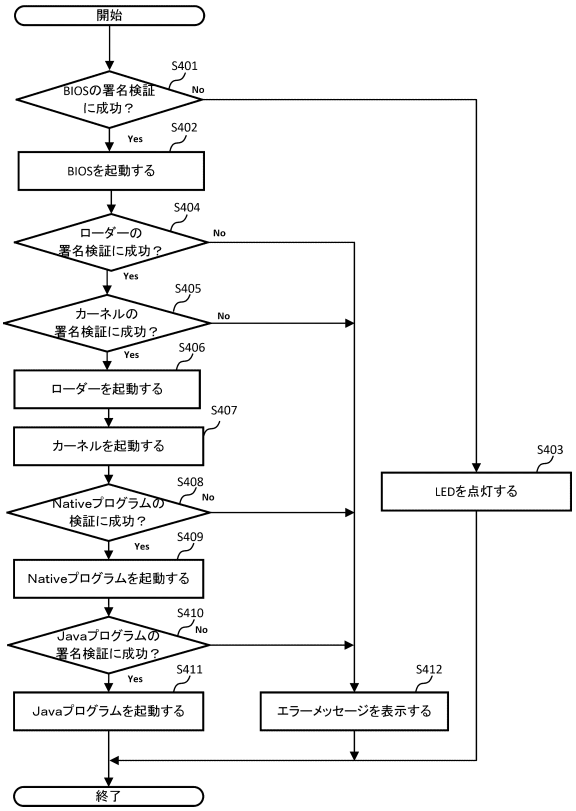
【図２】



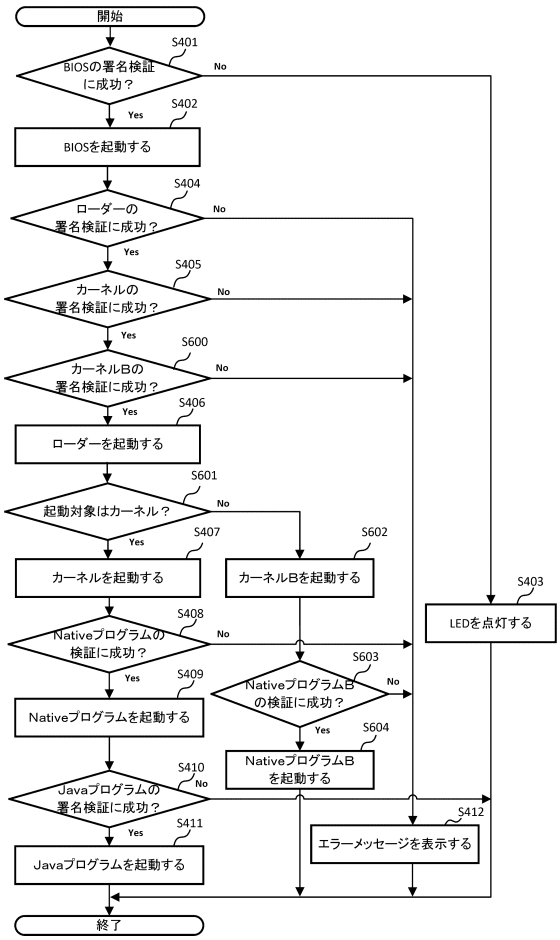
【図３】


(a)
209 ブートプログラム
起動
210 BIOS
211 ローダー
212 カーネル
213 Nativeプログラム
214 JAVAプログラム
(b)
検証
公開鍵
署名
306 307
310
ROM102
フラッシュメモリ114
HDD104
300 302 303 304 308 309
(c)
223
220 カーネルB
222 NativeプログラムB
(d)
341 342
343

【図４】


開始
S401 BIOSの署名検証に成功？
No
Yes
S402 BIOSを起動する
S404 ローダーの署名検証に成功？
S405 カーネルの署名検証に成功？
S406 ローダーを起動する
S407 カーネルを起動する
S408 Nativeプログラムの検証に成功？
S409 Nativeプログラムを起動する
S410 Javaプログラムの署名検証に成功？
S411 Javaプログラムを起動する
S412 エラーメッセージを表示する
S403 LEDを点灯する
終了

【図５】


開始
S401 BIOSの署名検証に成功？
No
Yes
S402 BIOSを起動する
S404 ローダーの署名検証に成功？
S405 カーネルの署名検証に成功？
S600 カーネルBの署名検証に成功？
S406 ローダーを起動する
S601 起動対象はカーネル？
S407 カーネルを起動する
S602 カーネルBを起動する
S403 LEDを点灯する
S408 Nativeプログラムの検証に成功？
S603 NativeプログラムBの検証に成功？
S409 Nativeプログラムを起動する
S604 NativeプログラムBを起動する
S410 Javaプログラムの署名検証に成功？
S412 エラーメッセージを表示する
S411 Javaプログラムを起動する
終了

フロントページの続き

(72)発明者　伊藤　祥晴
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

審査官　和平　悠希

(56)参考文献　特開２００５－１４８９３４（ＪＰ，Ａ）
特開２０１６－００６６５９（ＪＰ，Ａ）
特開２０１０－１８２１９６（ＪＰ，Ａ）
米国特許出願公開第２０１４／０１１５３１４（ＵＳ，Ａ１）
特開２０１７－１４６６９４（ＪＰ，Ａ）
特開平０８－３３９２９５（ＪＰ，Ａ）
特表２００８－５３７２２４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５７
Ｇ０６Ｆ　２１／５１
Ｇ０６Ｆ　２１／６４