(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2010-157230
(P2010-157230A)
(43)公開日　平成22年7月15日(2010.7.15)

(51)Int.Cl.		F I			テーマコード（参考）
G06F 21/22	(2006.01)	G06F	9/06	660Z	5B276
G06F 9/445	(2006.01)	G06F	9/06	610K	5B376

審査請求 有　請求項の数 23　ＯＬ　（全 15 頁）

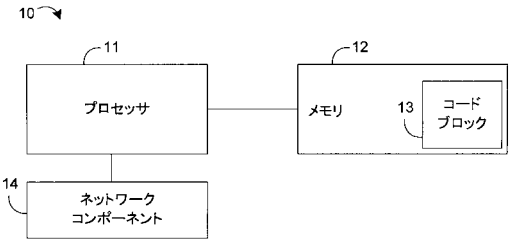
(21)出願番号　特願2009-292870 (P2009-292870)
(22)出願日　平成21年12月24日 (2009.12.24)
(31)優先権主張番号　12/317,852
(32)優先日　平成20年12月30日 (2008.12.30)
(33)優先権主張国　米国 (US)

(71)出願人　593096712
インテル　コーポレイション
アメリカ合衆国　９５０５２　カリフォルニア州　サンタ　クララ　ミッション　カレッジ　ブールバード　２２００
(74)代理人　100070150
弁理士　伊東　忠彦
(74)代理人　100091214
弁理士　大貫　進介
(74)代理人　100107766
弁理士　伊東　忠重
(72)発明者　ホームズド　エム.　コスラヴィ
アメリカ合衆国　９７２２９　オレゴン州　ポートランド　ノースウェスト　レイニア　テラス　４７３７

最終頁に続く

(54)【発明の名称】ランタイムインテグリティ検証のための装置及び方法

(57)【要約】
【課題】ランタイムファームウェアインテグリティ検証を効果的に実行するシステム及び方法を提供すること。
【解決手段】本発明の一特徴は、少なくとも１つのプロセッサと、前記少なくとも１つのプロセッサに接続される少なくとも１つのメモリと、コードブロックと、プロセッサベースシステムにより実行可能なコードとを有するプロセッサベースシステムであって、前記コードは、当該プロセッサベースシステムに、当該プロセッサベースシステムの再スタートに応答して、前記コードブロックのインテグリティ情報を生成させ、前記インテグリティ情報をセキュアに格納させ、前記セキュアに格納されているインテグリティ情報を用いて、当該プロセッサベースシステムの実行中に前記コードブロックのインテグリティを検証させるシステムに関する。
【選択図】図１



【特許請求の範囲】
【請求項１】
少なくとも１つのプロセッサと、
前記少なくとも１つのプロセッサに接続される少なくとも１つのメモリと、
コードブロックと、
プロセッサベースシステムにより実行可能なコードと、
を有するプロセッサベースシステムであって、
前記コードは、当該プロセッサベースシステムに、
当該プロセッサベースシステムの再スタートに応答して、前記コードブロックのインテグリティ情報を生成させ、
前記インテグリティ情報をセキュアに格納させ、
前記セキュアに格納されているインテグリティ情報を用いて、当該プロセッサベースシステムの実行中に前記コードブロックのインテグリティを検証させるシステム。
【請求項２】
前記コードブロックは、ファームウェアエレメント、ＢＩＯＳ（Ｂａｓｉｃ Ｉｎｐｕｔ Ｏｕｔｐｕｔ Ｓｙｓｔｅｍ）エレメント及びＳＭＭ（Ｓｙｓｔｅｍ Ｍａｎａｇｅｍｅｎｔ Ｍｏｄｅ）エレメントの１つに対応する、請求項１記載のシステム。
【請求項３】
前記コードはさらに、当該プロセッサベースシステムに、前記セキュアに格納されているインテグリティ情報を用いて、実行中に前記コードブロックのインテグリティを定期的に再検証させる、請求項１記載のシステム。
【請求項４】
前記コードはさらに、当該プロセッサベースシステムに、複数のコードブロックに対応するインテグリティ情報のリストをセキュアに格納させる、請求項１記載のシステム。
【請求項５】
前記コードはさらに、当該プロセッサベースシステムに、前記複数のコードブロックについて前記少なくとも１つのプロセッサに接続される少なくとも１つのメモリをスキャンさせ、前記セキュアに格納されているインテグリティ情報のリストを用いて、実行中に前記スキャンされたコードブロックのインテグリティを検証させる、請求項４記載のシステム。
【請求項６】
ネットワークコンポーネントをさらに有し、
前記コードはさらに、当該プロセッサベースシステムに、前記検証が失敗した場合にリモートアラートを送信させる、請求項５記載のシステム。
【請求項７】
少なくとも１つのプロセッサと、
前記少なくとも１つのプロセッサに接続されるシステム管理メモリと、
前記少なくとも１つのプロセッサと前記システム管理メモリとに接続される管理可能性エンジンを含むチップセットと、
プロセッサベースシステムにより実行可能なコードと、
を有するプロセッサベースシステムであって、
前記コードは、前記チップセットの管理可能性エンジンに当該プロセッサベースシステムの実行中に前記システム管理メモリのコンテンツを検証させるシステム。
【請求項８】
前記コードはさらに、当該プロセッサベースシステムに、
当該プロセッサベースシステムの再スタートに応答して、前記システム管理メモリのコンテンツのインテグリティ情報を生成させ、
前記インテグリティ情報を前記チップセットの管理可能性エンジンに転送させ、
前記管理可能性エンジンによりアクセス可能な位置に前記インテグリティ情報をセキュアに格納させる、請求項７記載のシステム。

【請求項９】
前記コードはさらに、前記管理可能性エンジンに、前記セキュアに格納されているインテグリティ情報を用いて、当該プロセッサベースシステムの実行中に前記システム管理メモリのコンテンツを定期的に再検証させる、請求項８記載のシステム。
【請求項１０】
前記コードはさらに、前記管理可能性エンジンに、前記システム管理メモリをスキャンさせ、前記セキュアに格納されているインテグリティ情報を用いて、前記スキャンされたメモリのインテグリティを検証させる、請求項９記載のシステム。
【請求項１１】
ネットワークコンポーネントをさらに有し、
前記コードはさらに、前記管理可能性エンジンに、前記検証が失敗した場合にリモートアラートを送信させる、請求項１０記載のシステム。
【請求項１２】
前記管理可能性エンジンによりアクセス可能な位置は、前記管理可能性エンジンのメモリサブシステムを含む、請求項１０記載のシステム。
【請求項１３】
前記管理可能性エンジンは、別のプロセッサを有し、
前記コードの少なくとも一部は、前記管理可能性エンジンの別のプロセッサにより実行可能である、請求項１０記載のシステム。
【請求項１４】
プロセッサベースシステムのためのランタイムインテグリティ検証を実行する方法であって、
前記プロセッサベースシステムの再スタートに応答して、コードブロックのインテグリティ情報を生成するステップと、
前記インテグリティ情報をセキュアに格納するステップと、
前記セキュアに格納されているインテグリティ情報を用いて、実行中に前記コードブロックのインテグリティを検証するステップと、
を有する方法。
【請求項１５】
前記コードブロックは、ファームウェアエレメント、ＢＩＯＳ（Ｂａｓｉｃ Ｉｎｐｕｔ Ｏｕｔｐｕｔ Ｓｙｓｔｅｍ）エレメント及びＳＭＭ（Ｓｙｓｔｅｍ Ｍａｎａｇｅｍｅｎｔ Ｍｏｄｅ）エレメントの１つに対応する、請求項１４記載の方法。
【請求項１６】
前記セキュアに格納されているインテグリティ情報を用いて、実行中に前記コードブロックのインテグリティを定期的に再検証するステップをさらに有する、請求項１４記載の方法。
【請求項１７】
複数のコードブロックに対応するインテグリティ情報のリストをセキュアに格納するステップをさらに有する、請求項１４記載の方法。
【請求項１８】
前記複数のコードブロックについて少なくとも１つのプロセッサに接続される少なくとも１つのメモリをスキャンし、前記セキュアに格納されているインテグリティ情報のリストを用いて、実行中に前記スキャンされたコードブロックのインテグリティを検証するステップをさらに有する、請求項１７記載の方法。
【請求項１９】
前記検証が失敗した場合にリモートアラートを送信するステップをさらに有する、請求項１８記載の方法。
【請求項２０】
プロセッサベースシステムのためのランタイムインテグリティ検証を実行する方法であって、

前記プロセッサベースシステムの再スタートに応答して、システム管理メモリのコンテンツのインテグリティ情報を生成するステップと、
前記インテグリティ情報を前記プロセッサベースシステムのチップセットの管理可能性エンジンに転送するステップと、
前記管理可能性エンジンによりアクセス可能な位置に前記インテグリティ情報をセキュアに格納するステップと、
を有する方法。
【請求項２１】
前記セキュアに格納されているインテグリティ情報を用いて、前記プロセッサベースシステムの実行中に前記管理可能性エンジンにより前記システム管理メモリのコンテンツを定期的に再検証するステップをさらに有する、請求項２０記載の方法。
【請求項２２】
前記システム管理メモリをスキャンし、前記セキュアに格納されているインテグリティ情報を用いて、前記スキャンされたメモリのインテグリティを検証するステップをさらに有する、請求項２１記載の方法。
【請求項２３】
前記検証が失敗した場合に前記管理可能性エンジンからリモートアラートを送信するステップをさらに有する、請求項２２記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、インテグリティ検証に関する。より詳細には、本発明の実施例は、ランタイムファームウェアインテグリティ検証に関する。
【背景技術】
【０００２】
コンピュータなどのプロセッサベースシステムは、ハードウェア及びソフトウェアセキュリティの各種レベルを要求又は利用する。
【発明の概要】
【発明が解決しようとする課題】
【０００３】
本発明の課題は、ランタイムファームウェアインテグリティ検証を効果的に実行するシステム及び方法を提供することである。
【課題を解決するための手段】
【０００４】
上記課題を解決するため、本発明の一特徴は、少なくとも１つのプロセッサと、前記少なくとも１つのプロセッサに接続される少なくとも１つのメモリと、コードブロックと、プロセッサベースシステムにより実行可能なコードとを有するプロセッサベースシステムであって、前記コードは、当該プロセッサベースシステムに、当該プロセッサベースシステムの再スタートに応答して、前記コードブロックのインテグリティ情報を生成させ、前記インテグリティ情報をセキュアに格納させ、前記セキュアに格納されているインテグリティ情報を用いて、当該プロセッサベースシステムの実行中に前記コードブロックのインテグリティを検証させるシステムに関する。
【発明の効果】
【０００５】
本発明によると、ランタイムファームウェアインテグリティ検証を効果的に実行するシステム及び方法を提供することができる。
【図面の簡単な説明】
【０００６】
【図１】図１は、本発明の実施例による電子システムのブロック図である。
【図２】図２は、本発明の実施例による他の電子システムのブロック図である。

【図３】図３は、本発明の実施例によるフロー図である。
【図４】図４は、本発明の実施例による他のフロー図である。
【図５】図５は、本発明の実施例による他のフロー図である。
【図６】図６は、本発明の実施例による他のフロー図である。
【図７】図７は、本発明の実施例による他の電子システムのブロック図である。
【図８】図８は、一例となるファイル構造の図である。
【発明を実施するための形態】
【０００７】
以下の説明では、限定することなく説明のため、特定の構造、アーキテクチャ、インタフェース、技術などの具体的詳細が、本発明の各種態様の完全な理解を提供するために与えられる。しかしながら、本発明の各種態様はこれらの具体的詳細から逸脱した他の実施例により実現可能であることは、本開示の利益を有する当業者に明らかであろう。ある実施例では、周知の装置、回路及び方法の説明は、不要な詳細により本発明の説明を不明りょうにしないように省略される。
【０００８】
図１を参照するに、本発明の実施例によると、プロセッサベースシステム１０は、少なくとも１つのプロセッサ１１と、少なくとも１つのプロセッサ１１に接続される少なくとも１つのメモリ１２と、コードブロック１３（メモリ１２などに格納される）とを有する。システム１０はさらに、プロセッサベースシステム１０にそれの再スタートに応答してコードブロック１３のインテグリティ情報を生成させ、インテグリティ情報をセキュアに格納させ、格納されているインテグリティ情報を用いてプロセッサベースシステム１０の実行中にコードブロック１３のインテグリティを検証させるため、プロセッサベースシステム１０により実行可能なコードを有する。例えば、コードブロック１３は、ファームウェアエレメント、ＢＩＯＳ（Ｂａｓｉｃ　Ｉｎｐｕｔ　Ｏｕｔｐｕｔ　Ｓｙｓｔｅｍ）エレメント、及びシステムマネージメントモード（ＳＭＭ）エレメントの１つに対応するものであってもよい。
【０００９】
本発明の実施例では、システム１０はさらに、プロセッサベースシステム１０にセキュアに格納されているインテグリティ情報を用いて実行中にコードブロック１３のインテグリティを定期的に再検証させるためのコードを有してもよい。本発明の実施例では、システム１０はさらに、プロセッサベースシステム１０に複数のコードブロック１３に対応するインテグリティ情報のリストをセキュアに格納させるためのコードを有してもよい。例えば、本発明の実施例では、システム１０はさらに、プロセッサベースシステム１０に複数のコードブロック１３について少なくとも１つのプロセッサ１１に接続される少なくとも１つのメモリ１２をスキャンさせ、セキュアに格納されているインテグリティ情報のリストを用いて実行中にスキャンされたコードブロック１３のインテグリティを検証させるためのコードを有してもよい。システム１０はさらに、ネットワークコンポーネント１４と、プロセッサベースシステム１０に検証が失敗した場合にリモートアラートを送信させるためのコードとを有してもよい。例えば、ネットワークコンポーネントは、イーサネット（登録商標）接続などの有線ネットワーク接続及び／又はＷｉＦｉ接続などの無線ネットワーク接続を含むものであってもよい。
【００１０】
図２を参照するに、本発明の実施例によると、プロセッサベースシステム２０は、少なくとも１つのプロセッサ２１と、少なくとも１つのプロセッサ２１に接続されるシステム管理メモリ２２と、少なくとも１つのプロセッサ２１及びシステム管理メモリ２２に接続される管理可能性エンジン２４を含むチップセット２３とを有する。システム２０はさらに、チップセット２３の管理可能性エンジン２４にプロセッサベースシステム２０の実行中にシステム管理メモリ２２のコンテンツを検証させるため、プロセッサベースシステム２０により実行可能なコードを有する。例えば、システム２０はさらに、プロセッサベースシステム２０にそれの再スタートに応答してシステム管理メモリ２２のコンテンツにつ

いてインテグリティ情報を生成させ、インテグリティ情報をチップセット２３の管理可能性エンジン２４に転送させ、管理可能性エンジン２４によりアクセス可能な位置にインテグリティ情報を安全に格納させるためのコードを有してもよい。

【００１１】

本発明の実施例では、システム２０はさらに、管理可能性エンジン２４にセキュアに格納されているインテグリティ情報を用いてプロセッサベースシステム２０の実行中にシステム管理メモリ２２のコンテンツを定期的に再検証させるためのコードを有してもよい。例えば、システム２０はさらに、管理可能性エンジン２４にシステム管理メモリ２２をスキャンさせ、セキュアに格納されているインテグリティ情報を用いてスキャンされたメモリ２２のインテグリティを検証させるためのコードを有してもよい。例えば、システム２０はさらに、ネットワークコンポーネント２５と、管理可能性エンジン２４に検証が失敗した場合にリモートアラートを送信させるためのコードとを有してもよい。本発明の実施例では、管理可能性エンジン２４によりアクセス可能な位置は、管理可能性エンジン２４のメモリサブシステム２６を含むものであってもよい。本発明の実施例では、管理可能性エンジン２４は、別のプロセッサ２７を有してもよく、コードの少なくとも一部は、管理可能性エンジン２４のプロセッサ２７により実行可能であってもよい。

【００１２】

図３を参照するに、本発明の実施例によると、プロセッサベースシステムのランタイムインテグリティ検証を実行する方法は、プロセッサベースシステムの再スタートに応答してコードブロックのインテグリティ情報を生成し（ブロック３０などにおいて）、インテグリティ情報をセキュアに格納し（ブロック３１などにおいて）、セキュアに格納されているインテグリティ情報を用いて実行中にコードブロックのインテグリティを検証する（ブロック３２などにおいて）ことを含む。例えば、コードブロックは、ファームウェアエレメント、ＢＩＯＳ（Ｂａｓｉｃ　Ｉｎｐｕｔ　Ｏｕｔｐｕｔ　Ｓｙｓｔｅｍ）エレメント及びシステムマネージメントモード（ＳＭＭ）エレメント（ブロック３３などにおいて）の１つの対応するものであってもよい。

【００１３】

例えば、本発明の実施例はさらに、セキュアに格納されているインテグリティ情報を用いて実行中にコードブロックのインテグリティを定期的に再検証する（ブロック３４などにおいて）ことを含むものであってもよい。例えば、本発明の実施例はさらに、複数のコードブロックに対応するインテグリティ情報のリストをセキュアに格納する（ブロック３５などにおいて）ことを含むものであってもよい。例えば、本発明の実施例はさらに、複数のコードブロックについて少なくとも１つのプロセッサに接続される少なくとも１つのメモリをスキャンし、セキュアに格納されているインテグリティ情報のリストを用いて実行中にスキャンされたコードブロックのインテグリティを検証する（ブロック３６などにおいて）ことを含むものであってもよい。本発明の実施例はさらに、検証が失敗した場合にリモートアラートを送信する（ブロック３７などにおいて）ことを含むものであってもよい。

【００１４】

図４を参照するに、本発明の実施例によると、プロセッサベースシステムのランタイムインテグリティ検証を実行する方法は、プロセッサベースシステムの再スタートに応答してシステム管理メモリのコンテンツのインテグリティ情報を生成し（ブロック４０などにおいて）、インテグリティ情報をプロセッサベースシステムの管理可能性エンジンに送信し（ブロック４１などにおいて）、管理可能性エンジンによりアクセス可能な位置にインテグリティ情報をセキュアに格納し（ブロック４２などにおいて）、セキュアに格納されているインテグリティ情報を用いてプロセッサベースシステムの実行中に管理可能性エンジンによるシステム管理メモリのコンテンツを検証する（ブロック４３などにおいて）ことを含む。

【００１５】

例えば、本発明の実施例はさらに、セキュアに格納されているインテグリティ情報を用

いてプロセッサベースシステムの実行中に管理可能性エンジンによりシステム管理メモリのコンテンツを定期的に再検証する（ブロック４４などにおいて）を含むようにしてもよい。例えば、本発明の実施例はさらに、システム管理メモリをスキャンし、セキュアに格納されているインテグリティ情報を用いてスキャンされたメモリのインテグリティを検証する（ブロック４５などにおいて）ことを含むようにしてもよい。例えば、本発明の実施例はさらに、検証が失敗した場合に管理可能性エンジンからリモートアラートを送信する（ブロック４６などにおいて）ことを含むようにしてもよい。

【００１６】

効果的には、本発明の実施例は、管理可能性エンジン（ＭＥ）／チップセットを用いたシステム管理モード（ＳＭＭ）ランタイムインテグリティ検証のための方法及び／又は装置を提供する。各種のセキュリティの労力は、ＳＭＭ／ＢＩＯＳ及びオペレーティングシステム（ＯＳ）のための信頼されるブート又はブートタイムインテグリティに注力されてきた。例えば、プレＯＳ　ＵＥＦＩ（Ｕｎｉｆｉｅｄ　Ｅｘｔｅｎｓｉｂｌｅ　Ｆｉｒｍｗａｒｅ　Ｉｎｔｅｒｆａｃｅ）／ＢＩＯＳプラットフォームインテグリティ検証は、インストール時（署名されたアップデートなど）及び／又はブート時（検証されたロード）に実行されてもよい。効果的には、本発明の実施例は、Ｔｉｍｅ　Ｏｆ　Ｃｈｅｃｋ（プレＯＳへのコードのロード中）／Ｔｉｍｅ　Ｏｆ　Ｕｓｅ（例えば、ＳＭＩイベントに応答した実行中のコードの呼び出し時など）（ＴＯＣＴＯＵ）アタック及び／又はコード投入アタックなどのランタイムアタックに対するセキュリティを向上させる。Ａｕｔｈｅｎｔｉｃｏｄｅ又は埋め込まれたシグネチャブロックによる他の公開鍵ベース機構を用いたＵＥＦＩ　ＳＭＭ実行可能ＰＥ　ＣＯＦＦイメージによる署名されたアップデートに対して、ＭＥはまた、イメージに係る公開検証鍵が無効にされているか確認するため、Ｃｅｒｔｉｆｉｃａｔｅ　Ａｕｔｈｏｒｉｔｙによりリモート検証することが可能である。

【００１７】

例えば、本発明の実施例は、チップセット管理可能性エンジンを用いて、ＳＭＭハンドラ（ＢＩＯＳコード）のランタイムインテグリティ検証を提供するようにしてもよい。効果的には、本発明の実施例は、ＳＭＭに対するランタイムアタックを無効にし、ＯＳに独立し、異なるＳＭＭコードの改訂に対してスケーラビリティを提供するものであってもよい。効果的には、本発明の実施例はまた、実行中のオペレーティングシステムのカーネルと同じ位置にあるＯＥＭのＵＥＦＩ（Ｕｎｉｆｉｅｄ　Ｅｘｔｅｎｓｉｂｌｅ　Ｆｉｒｍｗａｒｅ　Ｉｎｔｅｒｆａｃｅ）におけるランタイムアタックに対してガードするものであってもよい。効果的には、本発明の実施例によるランタイムインテグリティ検証を提供するため、ハードウェアベースのチップセットの管理可能性エンジンを利用することは、ソフトウェアベースのセキュリティと比較して、プラットフォームにおいてよりセキュアで隔離された位置を提供する。

【００１８】

ＳＭＭルートキットは、アンチウイルスソフトウェアによる検出が困難なコンピュータメモリのプロテクト部分において実行される悪意的なルートキットソフトウェアである。例えば、ＳＭＭルートキットは、秘密情報を隠すのに利用可能なキーロギング及び／又は通信ソフトウェアなどの悪質な機能を有するかもしれない。効果的には、本発明の実施例は、ＳＭＭルートキットアタックに対してガードするようにしてもよい。

【００１９】

本発明の実施例によると、ＳＭＩハンドラは以下のように実現されてもよい。すなわち、ＳＭＭに入ると、プロセッサは物理アドレスＳＭＢＡＳＥ＋０ｘ８０００（これは、ＳＭＩハンドラがＳＭＲＡＭの内部のオフセット０ｘ８０００にある必要があることを意味する）にジャンプする。Ｄ＿ＯＰＥＮビットが設定されているとき、コードはＳＭＲＡＭにあるかもしれないため、ＳＭＩトリガーは当該コードを実行させるよう強制される。

【００２０】

【表１】

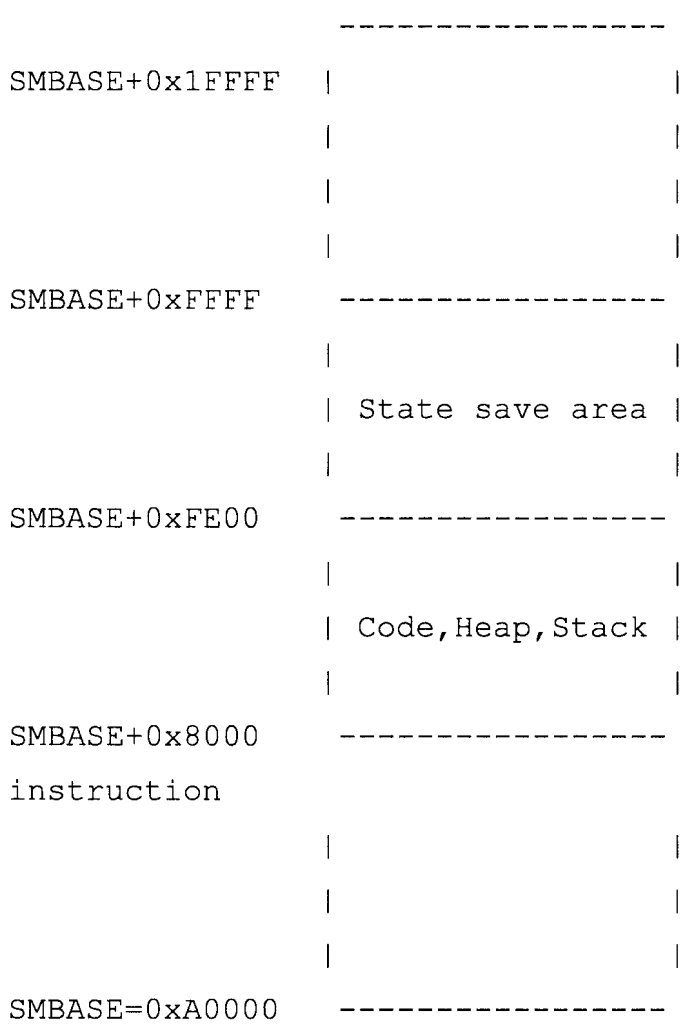


図５を参照するに、本発明の実施例では、管理可能性エンジン（ＭＥ）は、システムブート中にＨＥＣＩ（Ｈｏｓｔ Ｅｍｂｅｄｄｅｄ Ｃｏｎｔｒｏｌ Ｉｎｔｅｒｆａｃｅ）を介しＢＩＯＳからＳＭＭホワイトリストを受信する（ブロック５１などにおいて）。例えば、ホワイトリストは、認証されたコードモジュールのリストに対応するものであってもよい。例えば、管理可能性エンジンは、ＰＥＣＯＦＦ（Ｐｏｒｔａｂｌｅ Ｅｘｅｃｕｔｉｏｎ ａｎｄ Ｃｏｍｍｏｎ Ｏｂｊｅｃｔ Ｆｉｌｅ Ｆｏｒｍａｔ）ファイルを使用して、マニフェストを生成し、それらをフラッシュにセキュアに格納する（ブロック５２などにおいて）。管理可能性エンジンは、ランタイムインテグリティ検証チェックを定期的に実行する。例えば、管理可能性エンジンは、ランタイムインテグリティチェックタイマーを設定し、タイマーが経過したとき（ブロック５２などにおいて）、管理可能性エンジンは、ＳＭＭハンドラとＵＥＦＩランタイムテーブルについて、ホストメモリ及び／又はシステム管理ランダムアクセスメモリ（ＳＭＲＡＭ）をスキャンしてもよい（ブロック５３などにおいて）。管理可能性エンジンは、格納されているマニフェストを用いてＳＭＭ／ＢＩＯＳコードを検証する（ブロック５４などにおいて）。管理可能性エンジンは、ＳＭＭハンドラのポインタがテーブルポイントを有効なコード領域にディスパッチすることを検証する（ブロック５５などにおいて）。効果的には、本発明の実施例は、管理可能性エンジンがそのコンテンツをスキャンするため、ＳＭＲＡＭへのアクセスをチップセットに開放する。チップセットは、ＳＭＲＡＭに部分的にあるＳＭＭについてインテグリティ検証を提供する。

【００２１】
インテグリティチェックが失敗した場合（ブロック５７などにおいて）、管理可能性エンジンは、アラートをリモートＩＴ／ＡＶサーバ及び／又は他のリモートサービスに送信する。例えば、管理可能性エンジンはまた、システムが破壊する可能性があることをユーザに警告するようにしてもよい。例えば、管理可能性エンジンはまた、実行を停止してもよい。例えば、管理可能性エンジンはまた、システムを既知の信頼される状態に再格納し、及び／又はシステムを再スタートしてもよい。インテグリティチェックが成功した場合、管理可能性エンジンは、タイマーをリセットし、次のタイマーの中断を待機する（ブロック５８などにおいて）。

【００２２】
例えば、ｗｗｗ．ｕｅｆｉ．ｏｒｇのＵＥＦＩ　Ｐｌａｔｆｏｒｍ　Ｉｎｉｔｉａｌｉｚａｔｉｏｎ（ＰＩ）仕様書のＶｏｌｕｍｅ　４又はｗｗｗ．ｉｎｔｅｌ．ｃｏｍ／ｔｅｃｈｎｏｌｏｇｙ／ｆｒａｍｅｗｏｒｋのＳＭＭ－ＣＩＳに記載されるようなＵＥＦＩベースのＳＭＭインフラストラクチャについて、ポリシーベースのディスパッチャがある。このポリシーベースのディスパッチャは、ＵＥＦＩ　ＰＥ／ＣＯＦＦ（ｈｔｔｐ：／／ｍｉｃｒｏｓｏｆｔ．ｃｏｍ／ｗｈｄｃ／ｓｙｓｔｅｍ／ｐｌａｔｆｏｒｍ／ｆｉｒｍｗａｒｅ／ＰＥＣＯＦＦ．ｍｓｐｘ）　ｅｘｅｃｕｔａｂｌｅの系列を検出し、ＳＭＲＡＭにロードする。本発明の実施例は、以下の擬似コードを実現してもよい。

【００２３】
【表２】

```
For each UEFI SMM Driver D_raw_i in Firmware Volume FV_j (
    If D_raw_i meets integrity constraint (
        Load D_raw_i into SMRAM
        Relocate D_raw_i for its SMRAM-based execution
address
        Now have D_relocated_i in SMRAM
        If exist HECI interface (
            Pass D_relocated_i to ME via HECI in
            order to store for runtime whitelisting
        )
    )
)
```

例えば、本発明の実施例は、以下の擬似コードを実現してもよい。

【００２４】

【表３】

```
from [SmmBase.c]
 // Load UEFI DXE SMM driver into SMRAM
 Status = mPeLoader->LoadPeImage (
                    mPeLoader, mImageHandle, Args->FilePath,
                    Args->SrcBuffer, Args->SrcSize, Address,
                    &Index, Args->ImageHandle, NULL,
                    EFI_LOAD_PE_IMAGE_ATTRIBUTE_NONE
                    );

   if (!EFI_ERROR (Status) || Status ==
EFI_BUFFER_TOO_SMALL) {
     return EFI_INVALID_PARAMETER;
   }
  // Register the Image w/ the ME
   CheckAndFixHeciForAccess ();
   if (!EFI_ERROR (Status) || Status ==
EFI_BUFFER_TOO_SMALL) {
     return EFI_INVALID_PARAMETER;
   }
   HeciIntegrityPacket->ImageBase =  Args->SrcBuffer;
   HeciIntegrityPacket->ImageSize =   Args->SrcSize;

    Status = HeciPacketSend(HeciIntegrityPacket);

   if (!EFI_ERROR (Status) || Status ==
EFI_BUFFER_TOO_SMALL) {
     return EFI_INVALID_PARAMETER;
   }
```

上記フローとは対照的に、ＳＭＭルートキットは、上記ロードシーケンスの後の何れかの時点において、コードをＳＭＲＡＭに追加しようとする。ルートキットは、ＳＭＲＡＭをアクセス不可にするＤ＿ＬＣＫビットのチップセットにおける非設定（オプションのＲＯＭの実行前又はＯＳのブート前にＢＩＯＳが通常設定する）などのＢＩＯＳバグを介し（ｈｔｔｐ：／／ｗｗｗ．ｃｓ．ｕｃｆ．ｅｄｕ／～ｃｚｏｕ／ｒｅｓｅａｒｃｈ／ＳＭ

Ｍ－Ｒｏｏｔｋｉｔｓ－Ｓｅｃｕｒｅｃｏｍ０８．ｐｄｆ）、又はＴ－ＳＥＧがメモリアタックコードにエイリアスされるキャッシュアタックなどのハードウェアアタックを介し（ｈｔｔｐ：／／ｒｉｓｅｓｅｃｕｒｉｔｙ．ｏｒｇ／ｐａｐｅｒｓ／ｓｍｍ＿ｈａｃｋ．ｔｘｔ）、自らをインストールすることが可能である。あるいは、ＳＭＭルートキットは、終了されていないＢＩＯＳフレームワークＡＰＩを使用することによりＳＭＲＡＭに入ることが可能である（ｈｔｔｐｓ：／／ｗｗｗ．ｂｌａｃｋｈａｔ．ｃｏｍ／ｐｒｅｓｅｎｔａｔｉｏｎｓ／ｂｈ－ｕｓａ－０７／Ｈｅａｓｍａｎ／Ｐｒｅｓｅｎｔａｔｉｏｎ／ｂｈ－ｕｓａ－０７－ｈｅａｓｍａｎ．ｐｄｆ）。

【００２５】

図６を参照するに、本発明の実施例による一例となるＵＥＦＩブートフロー全体は、電源オン（又は再スタート）と、ｐｒｅＥＦＩステージ及びＤＸＥ（Ｄｒｉｖｅｒ Ｅｘｅｃｕｔｉｏｎ Ｅｎｖｉｒｏｎｍｅｎｔ）ステージを含むプラットフォーム初期化とを含む。ＤＸＥステージにおけるこのブートフローのサブセットは、ＳＭＭドライバのロードである。本発明の実施例によると、ＳＭＭドライバに関するインテグリティ情報は、システムブート中に生成され、管理可能性エンジンに転送される。例えば、ルートキーにより認証された署名付きのＵＥＦＩ ＤＸＥ ＳＭＭ ｅｘｅｃｕｔａｂｌｅに関する情報が、管理可能性エンジンに格納されてもよい。例えば、リロケートされたＵＥＦＩ ＤＸＥ ＳＭＭイメージのベース及びハッシュが、管理可能性エンジンに記録されてもよい。

【００２６】

図７を参照するに、本発明の実施例による電子システム７０は、メモリコントロールハブ（ＭＣＨ）７２とＩ／Ｏコントロールハブ（ＩＣＨ）７３とに接続されるＣＰＵ７１を有する。例えば、メインメモリＤＲＡＭ７４は、ＭＣＨ７２に接続されてもよい。例えば、ＩＣＨ７３は、ＳＡＴＡ及びＵＳＢを含む各種Ｉ／Ｏインタフェースをサポートするようにしてもよい。例えば、ＢＩＯＳコード、管理可能性エンジン（ＭＥ）コード及び他のシステムコードを格納するフラッシュメモリ７５が、ＩＣＨ７３に接続されてもよい。例えば、管理可能性エンジン（ＭＥ）を含むチップセット７６が、ＩＣＨ７３に接続されてもよい。ＤＲＡＭ７４のセグメントは、ＳＭＲＡＭとして指定される。例えば、ＳＭＲＡＭの先頭セグメントは、Ｔ－ＳＥＧと呼ばれるかもしれない。本発明の実施例では、Ｔ－ＳＥＧ ＳＭＲＡＭは、チップセット７６の管理可能性エンジン（ＭＥ）により実行時にモニタされる。

【００２７】

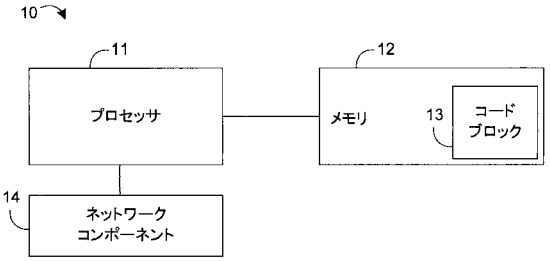
図８を参照するに、一例となるＰＥＣＯＦＦ実行可能ファイルは、図示されるようなファイル構造を有する。例えば、ＰＥＣＯＦＦ ｅｘｅｃｕｔａｂｌｅは、ＭＥによる観察中のオブジェクトを構成してもよい。例えば、Ｔ－ＳＥＧは、ＵＥＦＩ ＰＩ ＳＭＭインフラストラクチャのＰＥＣＯＦＦ実行可能イメージの系列により占有されてもよい。上記実行可能コンテンツ又はＳＭＲＡＭにおける認証されたリストを超えた実行可能コンテンツをハイジャック又は変更しようとする試みは、アタックを構成する。

【００２８】

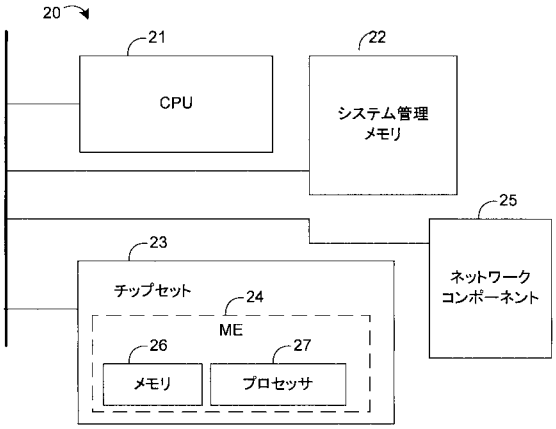
効果的には、実行時のホワイトリスティングＳＭＭに加えて、本発明の実施例は、実行時に他のエンティティをモニタする。例えば、本発明の実施例は、ＵＥＦＩ＿ＳＹＳＴＥＭ＿ＴＡＢＬＥ＿ＲＵＮＴＩＭＥ＿ＳＥＲＶＩＣＥＳデータオブジェクトと関連するファンクションポインタ及びルーチンとをモニタするようにしてもよい。例えば、これらのエンティティは、ＧｅｔＴｉｍｅ（）、ＵＥＦＩ変数のＧｅｔＳｅｔ（）、ＣａｐｓｕｌｅＵｐｄａｔｅ（）などのＯＳカーネルランタイムファームウェアインタラクション／コールを実現するため、プラットフォームファームウェアからＯＳカーネルランタイムにわたされてもよい。例えば、これらのエンティティの一部は、実行時にＯＳカーネルと一緒に配置され、ＯＳカーネルプロテクションがこれらのオブジェクトに対して利用されてもよい。効果的には、これらのコード及びデータオブジェクトをプロテクトするため、ＭＥによるランタイムインテグリティ検証の実行は、よりロウバストなＵＥＦＩの実現を可能にする。

【００２９】
ここに記載される実施例の多くは特定の実行環境に関する用語を使用するが、当業者は、本発明がこれらの特定の実施例に限定されず、均等なルーチン及び／又は構造が、セキュリティ機能が所望される他のプロセッサベース環境により実現可能であることを理解するであろう。同様に、ここで言及された各種コンポーネント、コントローラ、バーチャルモジュール、プロトコル及びレジスタは、等価な構成を提供し、等価な機能を実行しながら、他のプラットフォームにおける他の用語により記載されてもよい。
【００３０】
当業者は、本開示の利益が与えられると、他の多数の回路とハードウェア及び／又はソフトウェアの組み合わせが、ここに記載される実施例及び本発明の他の実施例に従って各種方法、回路及びシステムを実現するよう構成されてもよいことを理解するであろう。図１～８の実施例は、適切な実施例の非限定的な例である。
【００３１】
本発明の上記及び他の特徴は、個別に及び組み合わせて実現される。本発明は、特定の請求項により明示的に要求されない場合、このような特徴の２以上を要求するものとして解釈されるべきでない。さらに、本発明は好適な実施例であると現在考えられているものに関して説明されたが、本発明は開示された実施例に限定されず、本発明の趣旨及び範囲内に含まれる各種変更及び均等な構成をカバーするものであると理解されるべきである。
【符号の説明】
【００３２】
１０，２０　システム
１１，２１　プロセッサ
１２，２２　メモリ
１３　コードブロック
２３　チップセット
２４　管理可能性エンジン

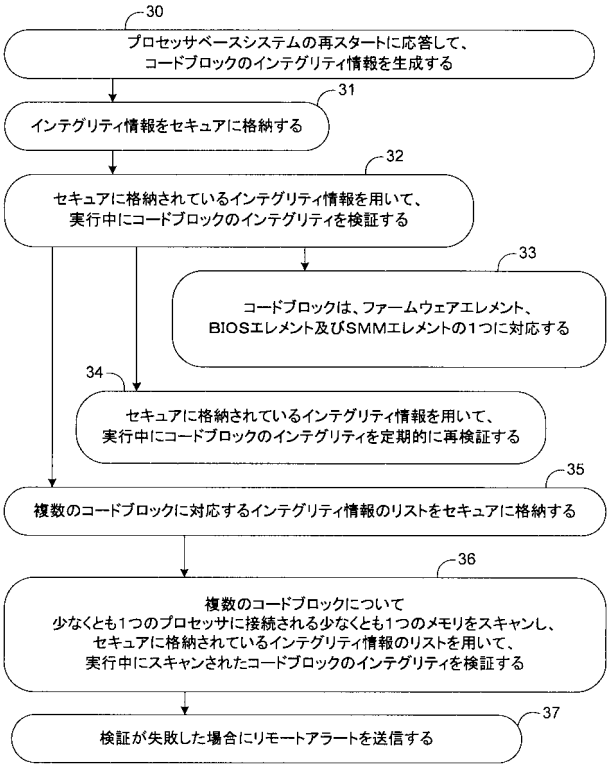
【図１】


10
11
プロセッサ
12
メモリ
13
コード
ブロック
14
ネットワーク
コンポーネント

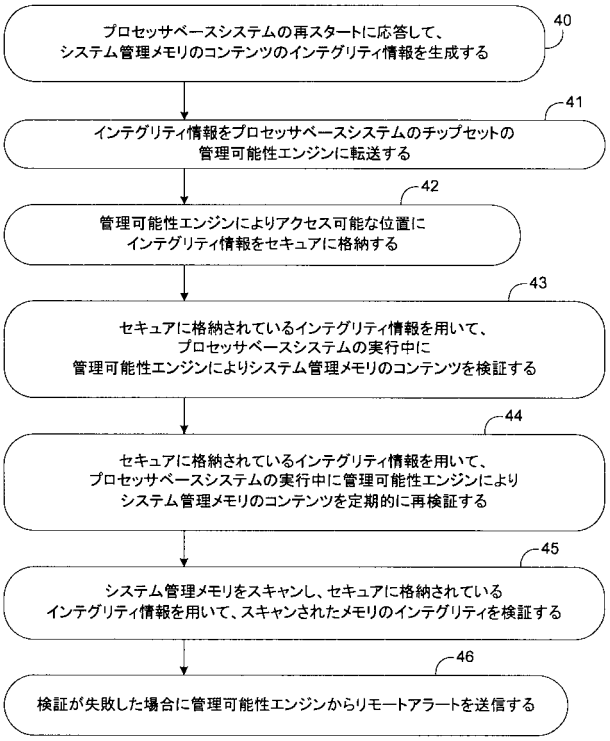
【図２】


20
21
CPU
22
システム管理
メモリ
25
ネットワーク
コンポーネント
23
チップセット
24
ME
26
メモリ
27
プロセッサ

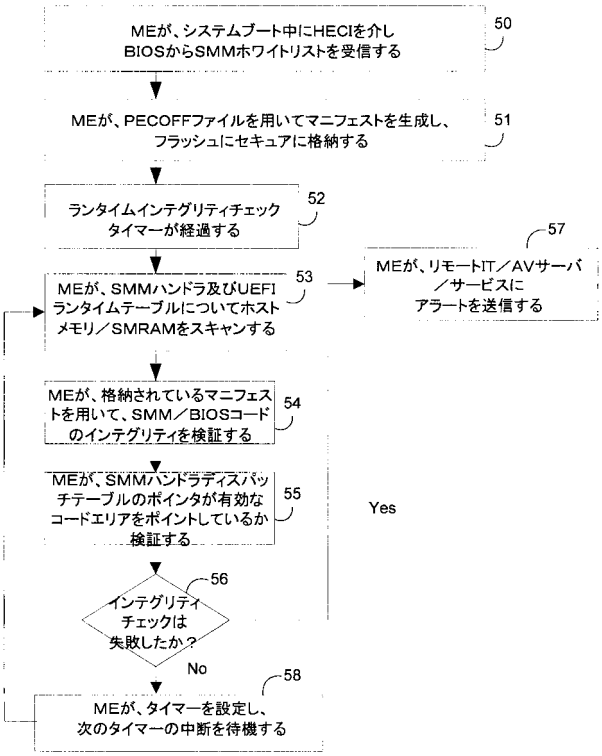
【図３】


30
プロセッサベースシステムの再スタートに応答して、
コードブロックのインテグリティ情報を生成する
31
インテグリティ情報をセキュアに格納する
32
セキュアに格納されているインテグリティ情報を用いて、
実行中にコードブロックのインテグリティを検証する
33
コードブロックは、ファームウェアエレメント、
BIOSエレメント及びSMMエレメントの1つに対応する
34
セキュアに格納されているインテグリティ情報を用いて、
実行中にコードブロックのインテグリティを定期的に再検証する
35
複数のコードブロックに対応するインテグリティ情報のリストをセキュアに格納する
36
複数のコードブロックについて
少なくとも1つのプロセッサに接続される少なくとも1つのメモリをスキャンし、
セキュアに格納されているインテグリティ情報のリストを用いて、
実行中にスキャンされたコードブロックのインテグリティを検証する
37
検証が失敗した場合にリモートアラートを送信する

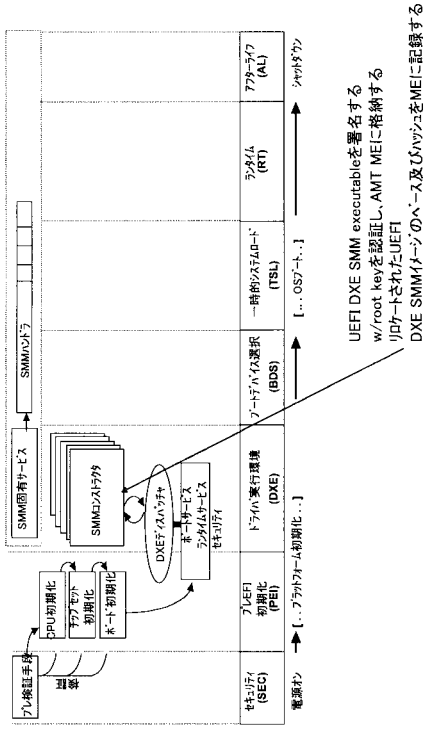
【図４】


プロセッサベースシステムの再スタートに応答して、
システム管理メモリのコンテンツのインテグリティ情報を生成する
40
インテグリティ情報をプロセッサベースシステムのチップセットの
管理可能性エンジンに転送する
41
管理可能性エンジンによりアクセス可能な位置に
インテグリティ情報をセキュアに格納する
42
セキュアに格納されているインテグリティ情報を用いて、
プロセッサベースシステムの実行中に
管理可能性エンジンによりシステム管理メモリのコンテンツを検証する
43
セキュアに格納されているインテグリティ情報を用いて、
プロセッサベースシステムの実行中に管理可能性エンジンにより
システム管理メモリのコンテンツを定期的に再検証する
44
システム管理メモリをスキャンし、セキュアに格納されている
インテグリティ情報を用いて、スキャンされたメモリのインテグリティを検証する
45
検証が失敗した場合に管理可能性エンジンからリモートアラートを送信する
46

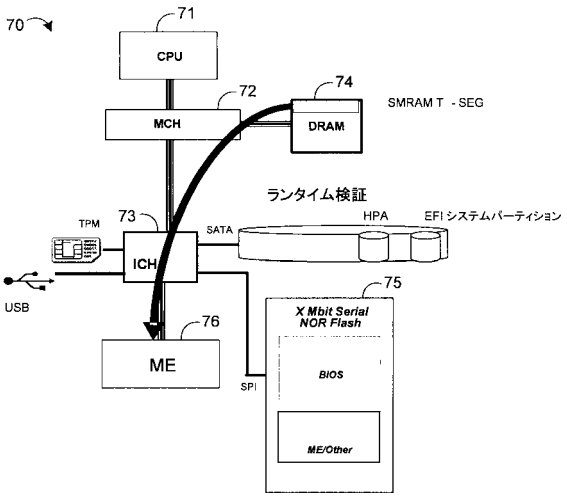
【図５】


MEが、システムブート中にHECIを介し
BIOSからSMMホワイトリストを受信する
50
MEが、PECOFFファイルを用いてマニフェストを生成し、
フラッシュにセキュアに格納する
51
ランタイムインテグリティチェック
タイマーが経過する
52
MEが、SMMハンドラ及びUEFI
ランタイムテーブルについてホスト
メモリ／SMRAMをスキャンする
53
MEが、リモートIT／AVサーバ
／サービスに
アラートを送信する
57
MEが、格納されているマニフェス
トを用いて、SMM／BIOSコード
のインテグリティを検証する
54
MEが、SMMハンドラディスパッ
チテーブルのポインタが有効な
コードエリアをポイントしているか
検証する
55
Yes
インテグリティ
チェックは
失敗したか？
56
No
MEが、タイマーを設定し、
次のタイマーの中断を待機する
58

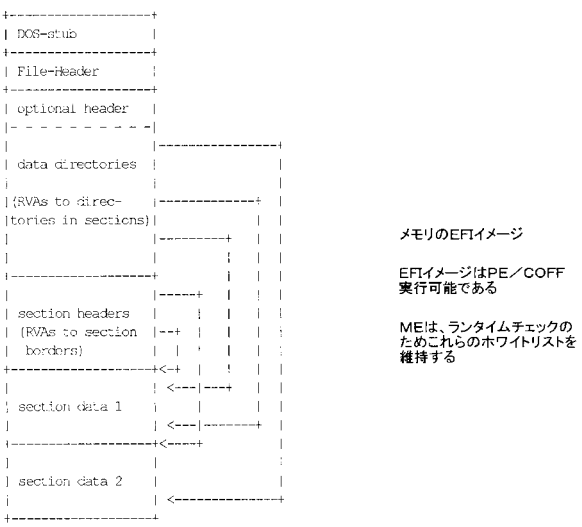
【図６】


UEFI DXE SMM executableを署名する
w/root keyを認証し、AMT MEに格納する
リロケートされたUEFI
DXE SMMイメージのベース及びハッシュをMEに記録する
SMMハンドラ
SMM固有サービス
SMMコンテクスト
DXEディスパッチャ
ボードサービス
ランタイムサービス
セキュリティ
CPU初期化
チップセット初期化
ボード初期化
プレ検証手段
検証
アフターライフ (AL)
ランタイム (RT)
一時的システムロード (TSL)
ブートデバイス選択 (BDS)
ドライバ実行環境 (DXE)
プレEFI初期化 (PEI)
セキュリティ (SEC)
電源オン
[...OSブート...]
[...プラットフォーム初期化...]

【図７】


70
71
CPU
72
MCH
74
DRAM
SMRAM T - SEG
ランタイム検証
TPM
73
ICH
SATA
HPA
EFI システムパーティション
USB
75
X Mbit Serial NOR Flash
BIOS
ME/Other
76
ME
SPI

【図８】


DOS-stub
File-Header
optional header
data directories
{RVAs to directories in sections}
section headers
{RVAs to section borders}
section data 1
section data 2
メモリのEFIイメージ
EFIイメージはPE／COFF実行可能である
MEは、ランタイムチェックのためこれらのホワイトリストを維持する

フロントページの続き

(72)発明者　ヴィンセント　ジェイ．　ジンマー
アメリカ合衆国　９８００３　ワシントン州　フェデラルウェイ　サウス　３６９番　ストリート　１９３７
(72)発明者　ディヴヤ　ナイドゥ　コーラー　サンダー
アメリカ合衆国　９７１２４　オレゴン州　ヒルズボロ　ノースイースト　アレックス　ウェイ　１３１４　ナンバー　１８７

Ｆターム(参考) 5B276 FD00
5B376 AD24 AD25