

(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2006-18528
(P2006-18528A)

(43) 公開日 平成18年1月19日(2006.1.19)

(51) Int. Cl.	F I	テーマコード (参考)
G06F 21/24 (2006.01)	G06F 12/14 540C	5B017
G06F 9/30 (2006.01)	G06F 12/14 530B	5B033
G06F 21/22 (2006.01)	G06F 9/30 390	5B076
H04L 9/10 (2006.01)	G06F 9/06 660D	5J104
	H04L 9/00 621A	
審査請求 未請求 請求項の数 20 O L (全 60 頁)		

(21) 出願番号 特願2004-194951 (P2004-194951)
(22) 出願日 平成16年6月30日 (2004. 6. 30)

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中4丁目1番1号
(74) 代理人 100074099
弁理士 大菅 義之
(74) 代理人 100067987
弁理士 久木元 彰
(72) 発明者 後藤 誠司
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
(72) 発明者 蒲田 順
神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

最終頁に続く

(54) 【発明の名称】 セキュアプロセッサ、およびセキュアプロセッサ用プログラム。

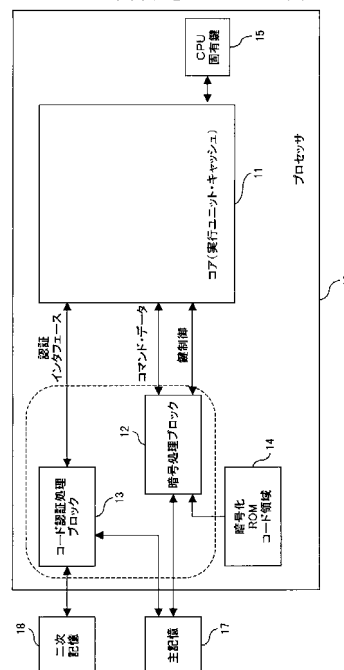
(57) 【要約】

【課題】 例えば暗号化された実行コードの認証を行い、認証に成功した実行コードのみを実行可能として情報処理の安全性を向上させる。

【解決手段】 命令コードを実行するコアと、コアに固有のCPU固有鍵と、暗号化された命令コードを書き換え不可能な形式で記憶する暗号化ROMコード領域と、この領域に記憶された命令コードを含む命令コードの認証を行うコード認証処理ブロックと、コアと外部との間で入出力されるデータを暗号化する暗号処理ブロックとを備える。

【選択図】 図2

第1の実施例におけるプロセッサの基本構成を示すブロック図



【特許請求の範囲】

【請求項 1】

命令コードを実行するコアを備えるプロセッサであって、
該コアに固有の鍵を記憶する鍵記憶手段と、
暗号化された命令コードを書き換え不可能な形式で記憶する命令コード記憶手段と、
該命令コード記憶手段に記憶された命令コードを含む命令コードの認証を前記固有鍵あるいは固有鍵による認証済鍵を用いて行う認証処理手段と、
該コアと外部との間で入出力されるデータを暗号化する暗号処理手段とを備えることを特徴とするセキュアプロセッサ。

【請求項 2】

前記暗号処理手段が、前記認証処理手段によって認証された命令コードを暗号化し、ページ単位で前記セキュアプロセッサに接続された記憶装置に格納することを特徴とする請求項 1 記載のセキュアプロセッサ。

【請求項 3】

前記記憶装置に格納されたページ単位の暗号化命令コードの実行時に不正命令が検出された時、該ページ単位の暗号化命令コードの実行を停止する不正命令実行停止手段を備えることを特徴とする請求項 2 記載のセキュアプロセッサ。

【請求項 4】

前記コアとして、
前記認証処理手段によって認証された命令コードのみを実行するセキュアコアと、
前記認証処理手段によって認証されていない通常の命令コードも実行可能なノーマルコアとを備えることを特徴とする請求項 1 記載のセキュアプロセッサ。

【請求項 5】

前記命令コード記憶手段に記憶された暗号化命令コードを用いて前記セキュアコアがブートされるとともに、
該セキュアコアが該ブート完了後に前記ノーマルコアのブートを行わせるノーマルコアブート手段を備えることを特徴とする請求項 4 記載のセキュアプロセッサ。

【請求項 6】

前記セキュアコアが、前記ノーマルコアのブート後に該ノーマルコアの動作を監視し、異常状態を検出した時、該ノーマルコアの動作停止、または特定処理への分岐を行わせるノーマルコア監視手段を備えることを特徴とする請求項 5 記載のセキュアプロセッサ。

【請求項 7】

プロセッサにおいて命令コードを実行するコアによって使用されるプログラムであって、
暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて自コアの起動処理を行う手順と、
該メモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックと、前記コア固有の鍵を管理する鍵管理処理と、該認証処理ブロックによって認証された命令コードの暗号化 / 復号処理に用いられる鍵が格納された鍵テーブルに対する操作処理とをセットアップする手順と、
前記認証処理ブロックを用いて二次記憶上のプログラムの認証処理を行う手順と、
起動されたオペレーティングシステムを含む該認証処理済みのプログラムの実行時に前記命令コードの暗号化 / 復号のための鍵処理を含む処理を実行する鍵処理モジュールとしての動作を行う手順とを計算機に実行させるためのセキュアプロセッサ用プログラム。

【請求項 8】

命令を実行する命令実行手段と、
該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード / ストアを制御するロード / ストア制御手段と、
該ロード / ストア制御手段と外部のメモリとの間でデータの暗号化 / 復号化を行う暗号処理手段とを備え、

10

20

30

40

50

前記命令実行手段が、実行中の命令に対応して該暗号処理手段に対してデータ暗号化 / 復号化に使用すべき鍵を指定することを特徴とするセキュアプロセッサ。

【請求項 9】

前記プロセッサにおいて複数の鍵を記憶する鍵記憶手段をさらに備え、

前記命令実行手段が、該鍵記憶手段に対して前記鍵を指定する鍵番号を出力し、該鍵記憶手段が該鍵番号に対応して前記暗号処理手段に対して、データ暗号化 / 復号化に使用すべき鍵を与えることを特徴とする請求項 8 記載のセキュアプロセッサ。

【請求項 10】

前記セキュアプロセッサにおいて、外部からロードされた命令フェッチデータの復号化に使用されるべき鍵を記憶する鍵記憶手段をさらに備え、

10

前記命令実行手段が命令フェッチ状態にある時、該鍵記憶手段が前記暗号処理手段に対して該復号化用の鍵を与えることを特徴とする請求項 8 記載のセキュアプロセッサ。

【請求項 11】

命令を実行する命令実行手段と、

該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード / ストアを制御するロード / ストア制御手段と、

該ロード / ストア制御手段と外部のメモリとの間でデータの暗号化 / 復号化を行う暗号処理手段とを備え、

前記命令実行手段が、実行中の命令によるデータ / 命令フェッチのアクセスアドレスに対応させて、該暗号処理手段に対してデータ暗号化 / 復号化に使用すべき鍵を指定する信号を与えることを特徴とするセキュアプロセッサ。

20

【請求項 12】

前記プロセッサにおいて複数の鍵を記憶する鍵記憶手段をさらに備え、

前記命令実行手段が、前記アクセスアドレスとしての論理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該論理アドレスに対応して前記データ暗号化 / 復号化用の鍵を前記暗号処理手段に与えることを特徴とする請求項 11 記載のセキュアプロセッサ。

【請求項 13】

前記セキュアプロセッサにおいて複数の鍵を記憶する鍵記憶手段をさらに備え、

前記ロード / ストア制御手段が、前記命令実行手段から与えられるコマンドに対応して前記アクセスアドレスとしての物理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該物理アドレスに対応して前記データ暗号化 / 復号化用の鍵を前記暗号処理手段に与えることを特徴とする請求項 11 記載のセキュアプロセッサ。

30

【請求項 14】

実行コードに対応するプロセスの実行に先立って、該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するためのセキュアプロセス識別子を、該プロセスの生成命令が発行された時点で生成するセキュアプロセス識別子生成手段と、

該生成されたセキュアプロセス識別子を該プロセスに関連する情報として保持するプロセス情報保持手段とを備えることを特徴とするセキュアプロセッサ。

【請求項 15】

40

前記プロセスに対応する実行コードに認証情報が付与されるとともに、

前記プロセス情報保持手段が、前記生成されたプロセスの生存期間中に行われる実行コード認証のための認証鍵をさらに保持することを特徴とする請求項 14 記載のセキュアプロセッサ。

【請求項 16】

前記実行コードに付与された認証情報がメモリにおけるページ単位の情報であることを特徴とする請求項 15 記載のセキュアプロセッサ

【請求項 17】

前記プロセスに対応する実行コードがメモリの空きページに格納され、該ページのアドレスに対応させて前記セキュアプロセス識別子が前記プロセッサ内のバッファに格納され

50

た後に前記ページ単位の認証鍵を用いた該実行コードの認証が成功した時、該バッファに前記セキュアページフラグをセットする認証手段をさらに備えることを特徴とする請求項16記載のセキュアプロセッサ。

【請求項18】

前記セキュアプロセッサにおいて、

前記実行コードの実際の実行に先立って前記バッファ内に格納されたセキュアプロセス識別子であって、対応する前記セキュアページフラグがセットされているセキュアプロセス識別子と、前記プロセス情報保持手段に保持され、実行すべき命令コードに対応するセキュアプロセス識別子とを比較し、両者が一致した時に前記実行コードが格納されたメモリ上のページへのアクセスを、命令を実行する命令実行部に許可するメモリアクセス制御手段をさらに備えることを特徴とする請求項17記載のセキュアプロセッサ。

10

【請求項19】

実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであって、

該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼する手順と

、
該転送の成功後に、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって

20

、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定する手順と、
前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。

【請求項20】

実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、メモリに読み込まれた該ページに対するハッシュ演算を行う手順と、

該ページに付与されている認証情報を復号する手順と、

該ハッシュ演算結果と該復号結果とを比較する手順と、

30

該比較の結果として一致が検出された時、該計算機のトランスレーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、計算機などの情報処理システムの安全性確保方式に係り、さらに詳しくは計算機や各種プロセッサ組込機器などにおいて悪意を持った実行コードの動作を防止することが可能なセキュアプロセッサ、およびセキュアプロセッサ用プログラムに関する。

40

【背景技術】

【0002】

プロセッサを使用するシステムでは、動作をプログラムによって記述することができ、ハードウェアによってすべてを構成するシステムに比べ、その動作の柔軟性が大きく、多種類の機能を実装することが容易である。これらの特徴のために、プロセッサはパーソナルコンピュータ、PDA、携帯電話、情報家電など多くのシステムに使用されており、またその普及とともに電子商取引のように高度にセキュリティを要求される処理も広範に行われるようになってきている。セキュリティを強固にするために回線データに対する暗号化、ユーザ認証などの各種のシステムの措置が施されているが、近年ではシステムレベルの安全性のみならず、コンピュータウィルスや不正アクセス等の蔓延に対応するために、

50

ソフトウェアレベル、プロセッサレベルの安全性が問題となっている。

【0003】

例えば携帯電話や情報家電など、各種のプロセッサ組込機器などがネットワークに接続されることにより、パーソナルコンピュータ等に対すると同等の脅威をこれらの機器も外部から受ける可能性が高くなっている。不正侵入などを細かく見ると、悪意を持った実行コードが端末内で動作することにその原因がある。悪意のあるコード、所望しないコードをプロセッサ上で動作させないようにすることが重要であるが、従来においては悪意のあるコードを動作させないようにするためのプロセッサ側の対策があまり十分ではなく、結果的に安全なソフトウェア実行環境が提供されていないという問題点があった。

【0004】

次に従来においては、例えばデータや命令の実行コードを主記憶装置や二次記憶装置に格納するに当り、安全性を確保するために暗号化を行って、実際の命令実行に当って暗号化されたデータなどを復号し、プロセッサ内のキャッシュメモリに格納して処理を実行することも行われているが、このような場合暗号処理を実行するハードウェアなどはプロセッサチップと別のチップに搭載され、外付けで使用されるために処理速度など、暗号処理性能が低くなってしまうという問題点があった。

【0005】

またこのような暗号化処理において、データなどの暗号化に用いられる暗号鍵は、外付けされたチップ上の暗号処理側で決定されており、プロセッサ側で実行される命令の種類やスーパーバイザ/ユーザモードなどの区別、あるいはデータや命令フェッチのアクセスアドレスなどに無関係であり、プロセッサ側の実行ユニットが暗号化、および復号において用いられるべき鍵を指定することができないために、実行中の命令に対応して適切な暗号鍵を選択することができないという問題点もあった。

【0006】

このようなソフトウェア実行環境の安全性に関する従来技術として次の文献がある。

【特許文献1】特開2002-353960号公報 「コード実行装置およびコード配布方法」

【0007】

この文献には暗号化された実行コードの認証を行って暗号化コードの有効性を確認し、セキュアプロセッサがその暗号化コードに対応する命令をフェッチし、セキュアタスクとして実行するコード実行装置が開示されている。

【0008】

しかしながらこのコード実行装置では実行コードに対応するプロセスと認証に用いられる鍵との間に関連が無く、例えばオペレーティングシステム(OS)に対して悪意のある操作が行われ、プログラムに別の認証鍵が割り付けられると、結果的に悪意のあるコードが動作してしまうという問題点を解決できなかった。

【発明の開示】

【発明が解決しようとする課題】

【0009】

本発明の第1の課題は、暗号化された命令コードを書き換え不可能な形式で記憶するメモリの記憶内容を基本として、例えば二次記憶装置上に格納されているプログラムの実行コードを次々と認証し、確実に信頼できるアプリケーションの範囲を段階的に広げ、信頼できる動作だけを実行することが可能なセキュアプロセッサを提供することである。

【0010】

本発明の第2の課題は、暗号処理ブロックを、例えばプロセッサと同一チップ上に備えて暗号処理性能を向上させるとともに、実行中の命令によってデータや実行コードの暗号化/復号に用いるべき鍵を選択可能とすることである。

【0011】

本発明の第3の課題は、プロセスの実行コードの主記憶への格納のタイミングでプロセスに対応する認証鍵を用いて実行コードの認証を行い、認証に成功した実行コードのみを

10

20

30

40

50

実行可能とすることによって、プロセッサによる情報処理の安全性を向上させることである。

【課題を解決するための手段】

【0012】

図1は本発明のセキュアプロセッサの原理構成ブロック図である。同図において本発明のセキュアプロセッサ1は、固有鍵記憶手段2、命令コード記憶手段3、認証処理手段4、および暗号処理手段5を備える。

【0013】

固有鍵記憶手段2は、セキュアプロセッサにおいて命令コードを実行するコアに固有の鍵、例えばCPU固有鍵を記憶するものであり、命令コード記憶手段3、例えば暗号化ROMコード領域は暗号化された命令コードを書き換え不可能な形式で記憶するものであり、認証処理手段4は固有鍵を用いて命令コード記憶手段3に記憶された命令コードを含む命令コードの認証を行うものであり、暗号処理手段5はコアと外部のメモリとの間で入出力されるデータを暗号化するものである。

10

【0014】

発明の実施の形態においては、暗号処理手段5が、認証された命令コードを暗号化してページ単位でセキュアプロセッサ1に接続された記憶装置、例えば主記憶に格納することもでき、また認証処理手段4が認証対象とする命令コードに認証情報が付加されていることもできる。

【0015】

次に図1のセキュアプロセッサ1において、命令コードを実行するコアとして認証処理手段4によって認証された命令コードのみを実行するセキュアコアと、認証されていない通常の命令コードを実行するノーマルコアとを備えることもできる。

20

【0016】

この場合、命令コード記憶手段3に記憶された暗号化された命令コードを用いてセキュアコアがブート(起動)されるとともに、セキュアコアがそのブート完了後にノーマルコアのブートを行わせるノーマルコアブート手段を備えることもでき、さらにセキュアコアがノーマルコアのブート後にノーマルコアの動作を監視し、異常状態を検出した時、ノーマルコアの動作停止、または特定処理への分岐を行わせるノーマルコア監視手段を備えることもできる。

30

【0017】

次に本発明のセキュアプロセッサ用プログラムは、暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて起動処理を行う手順と、そのメモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックと、プロセッサ固有の鍵を管理する鍵管理処理と、認証処理ブロックによって認証された命令コードの暗号化/復号処理に用いられる鍵が格納された鍵テーブルに対する操作処理とをセットアップする手順と、認証処理ブロックを用いて二次記憶上のプログラムの認証処理を行う手順と、起動されたオペレーティングシステムを含み、認証処理済のプログラムの実行時に必要となる鍵処理を実行する鍵処理モジュールとしての動作を行う手順とを計算機に実行させるものである。

40

【0018】

本発明のセキュアプロセッサは、命令を実行する命令実行手段、例えば実行ユニットと、命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード/ストアを制御するロード/ストア制御手段、例えばロードストアユニットと、ロード/ストア制御手段と外部のメモリとの間でデータの暗号化/復号化を行う暗号処理手段、例えば暗号化回路と復号化回路とを備え、命令実行手段が、実行中の命令に対応して暗号処理手段に対してデータ暗号化/復号化に使用すべき鍵を指定するものである。

【0019】

発明の実施の形態においては、このセキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段、例えば鍵テーブルメモリをさらに備え、命令実行手段が、鍵記憶手段に対し

50

て前述の鍵を指定する鍵番号を出力し、該鍵記憶手段がその鍵番号に対応して暗号処理手段に対して、データ暗号化/復号化に使用すべき鍵を与えることもできる。

【0020】

またこのセキュアプロセッサにおいて、外部からロードされた命令フェッチデータの復号に使用されるべき鍵を記憶する鍵記憶手段をさらに備え、命令実行手段が命令フェッチ状態にある時、鍵記憶手段が暗号処理手段に対してフェッチされた命令の復号に使用されるべき鍵を与えることもできる。

【0021】

また本発明のセキュアプロセッサは命令を実行する命令実行手段と、命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード/ストアを制御するロード/ストア制御手段と、ロード/ストア制御手段と外部のメモリとの間でデータの暗号化/復号化を行う暗号処理手段とを備え、命令実行手段が、実行中の命令によるデータ/命令フェッチのアクセスアドレスに対応させて、暗号処理手段に対してデータ、および命令の暗号化/復号化に使用すべき鍵を指定する信号を与えるものである。

10

【0022】

発明の実施の形態においては、このセキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、命令実行手段が、前述のアクセスアドレスとしての論理アドレスを出力し、鍵記憶手段がその論理アドレスに対応して暗号処理手段に対して暗号化/復号化に使用すべき鍵を与えることもできる。

【0023】

あるいはこのセキュアプロセッサにおいて、複数個の鍵を記憶する鍵記憶手段をさらに備え、ロード/ストア制御手段が、命令実行手段から与えられるコマンドに対応してアクセスアドレスとしての物理アドレスを出力し、鍵記憶手段がその物理アドレスに対応して暗号処理手段に対して暗号化/復号化に使用すべき鍵を与えることもできる。

20

【0024】

本発明のセキュアプロセッサは、実行コードに対応するプロセスの実行に先立ってその実行コードが正しく認証されたことを示すセキュアページフラグが設定されるページに対応するセキュアプロセス識別子と比較するためのセキュアプロセス識別子を、そのプロセスの生成命令が発行された時点で生成するセキュアプロセス(コンテキスト)識別子生成手段と、生成されたセキュアプロセス識別子をそのプロセスに関連する情報として保持するプロセス情報保持手段、例えばコンテキスト情報格納部とを備える。

30

【0025】

発明の実施の形態においては、前述のプロセスに対応する実行コードに認証情報が付与されるとともに、プロセス情報保持手段が生成されたプロセスの生存期間中に行われる実行コード認証のための認証鍵をさらに保持することもできる。

【0026】

またこのセキュアプロセッサは、前述のプロセスに対応する実行コードがメモリの空きページに格納され、そのページのアドレスに対応させてセキュアプロセス識別子がプロセッサ内のバッファに格納された後に、ページ単位の認証鍵を用いてその実行コードの認証を行い、認証が成功した時、そのバッファにセキュアページフラグをセットする認証手段をさらに備えることもできる。

40

【0027】

あるいはこのセキュアプロセッサは、実行コードの実際の実行に先立って前述のバッファ内に格納されたセキュアプロセス識別子であって、対応するセキュアページフラグがセットされているセキュアプロセス識別子と、プロセス情報保持手段に保持され、実行すべき命令コードに対応するセキュアプロセス識別子とを比較し、両者が一致した時に実行コードが格納されたメモリ上のページへのアクセスを、命令を実行する命令実行部に許可するメモリアクセス制御手段をさらに備えることもできる。

【0028】

またこのセキュアプロセッサは、それぞれ命令実行ユニットとキャッシュとを備えるコ

50

アとして、認証された実行コードのみを実行するセキュアコアと、認証されていない通常の実行コードを実行するノーマルコアとを備えることもできる。

【0029】

さらにこのセキュアプロセッサは、実行コードのメモリへの格納と並行して実行コードの認証に必要となる演算を行い、その演算の結果を保持して認証手段に与える直接メモリアクセス手段をさらに備えることもできる。

【0030】

次に本発明のセキュアプロセッサ用プログラムは、実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであり、計算機内の直接メモリアクセス機構に前述のページのメモリへの転送を依頼する手順と、その転送の成功後に計算機のトランザクション・ルックアサイド・バッファ内のページ・テーブル・エントリに、そのページ内の実行コードに対応するプロセスの実行に先立ってその実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、そのプロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、そのページについてのデータを設定する手順と、前述のページの認証と、認証の成功を示すセキュアページフラグのページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させるものである。

10

【0031】

さらに本発明のセキュアプロセッサ用プログラムは、実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、メモリに読み込まれたページに対するハッシュ演算を行う手順と、そのページに付加されている認証情報を復号する手順と、ハッシュ演算結果と復号結果とを比較する手順と、比較の結果として一致が検出された時、その計算機のトランザクション・ルックアサイド・バッファ内のページ・テーブル・エントリにページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させるものである。

20

【発明の効果】

【0032】

本発明によれば、プロセッサ内に保持されている書き換え不可能な形式の暗号化された命令コードを基本的な信頼点として、例えばオペレーティングシステムを含むプログラムの認証を行い、信頼できるプログラムの範囲を拡大していくことによって、システムのセキュリティレベルを本質的に向上させることが可能となる。

30

【0033】

また本発明によれば、暗号処理ブロックを、例えばプロセッサと同一チップ内に備え、暗号処理性能を向上させるとともに、実行中の命令に応じたデータや実行コードの暗号化を行うことが可能となる。実行中の命令に対応して、例えば暗号化のレベルを変化させることもでき、システムとしてのセキュリティレベルを向上させることができる。

【0034】

さらに本発明によれば、命令コードの実行の前にその命令コードの認証を行い、セキュアページフラグがセットされたプロセスに対応するプロセス識別子と実行中のプロセスのプロセス識別子との一致を検出してプロセスの実行を行うことにより、悪意を持って改ざんされた実行コードのプロセッサ上での動作を防止することができ、安全なソフトウェア実行環境が提供される。

40

【発明を実施するための最良の形態】

【0035】

以下、本発明の実施形態について詳細に説明するが、まず本発明のセキュアプロセッサの全体的な構成と、その処理の概要を第1の実施例として説明する。

図2は、第1の実施例としてのセキュアプロセッサの基本構成を示すブロック図である。同図においてプロセッサ10は、実行ユニットとキャッシュを含むコア11、外部インタフェースとのコマンド処理およびバスデータ(プログラムのコードまたはデータ)の暗

50

号化やその復号などを行う暗号処理ブロック 12、命令コードの認証を行うコード認証処理ブロック 13、プロセッサの起動時に用いられる最も基本的なプログラムなどが暗号化されて格納されている暗号化 ROM コード領域 14、およびこのコード領域 14 に格納されているプログラムなどの復号などを行うための CPU 固有鍵 15 を備えている。なお、暗号処理ブロック 12 の動作については後述の第 3 の実施例などで、コード認証処理ブロック 13 の動作については第 5 の実施例などでより詳細に説明する。

【0036】

そしてコア 11 と暗号処理ブロック 12 との間では、コマンド、およびデータのやり取りが行われるとともに暗号化のための鍵の制御が行われ、またコア 11 とコード認証処理ブロック 13 との間には認証インタフェースが備えられる。さらに暗号処理ブロック 12、およびコード認証処理ブロック 13 は主記憶 17 に対するアクセスを実行し、またコード認証処理ブロック 13 は二次記憶 18 に対するアクセスを実行するものとする。

10

【0037】

図 3 は、第 1 の実施例におけるセキュアプロセッサの全体処理フローチャートである。同図において電源が投入されると、図 2 のコア 11 はステップ S1 で暗号化 ROM コード領域 14 に格納されているプログラムを CPU 固有鍵 15 を用いて復号し、ブート（起動）処理を実行する。内蔵 ROM であるため、プログラムの改ざんは本来物理的に困難であるが、もし何らかの方法で改ざんされた場合でも、プログラムは暗号化されており、意味のある改ざんは困難である。従って正確にブートできた場合は、プログラムの改ざんがなかったと判断でき、暗号化 ROM コード領域 14 に格納されているプログラムは絶対的に信頼できるものといえることになり、この状態をプログラムの基本的な信頼点として定義することが可能となる。

20

【0038】

なお、暗号化 ROM コード領域 14 については、64 ビット単位の暗号化を行う DES（データ・エンクリプション・スタンダード）方式よりも秘匿強度の大きい AES（アドバンスド・エンクリプション・スタンダード）方式を用いる場合などはプロセッサ内部ではなく外付けとすることもできる。その場合には命令コードの NOP、データパターンの ALL0 / ALL1 など、頻発するパターンに対応して暗号鍵の推定を可能とさせないように、同一の平文に対して常に同一の暗号文が出力される ECB（エレクトリック・コードブック）以外のモードを使用することも可能である。

30

【0039】

続いてステップ S2 で暗号処理ブロック 12 内に備えられ、後述する鍵テーブル（メモリ）に対する操作処理、CPU 固有鍵 15 を用いた公開鍵や秘密鍵の生成などを行う鍵管理処理、およびコード認証処理ブロック 13 のセットアップなどが実行され、これらの処理の内容は同様な信頼点として定義される。

【0040】

続いてステップ S3 で二次記憶 18 に格納されているプログラムに対する認証処理が行われる。この第 1 の実施例ではオペレーティングシステム（OS）を含む一般のプログラムは、ハードディスクやネットワーク経由など二次記憶 18 上に格納されており、これらのプログラムに対する認証処理が実行される。この認証処理については、さらに後述する。

40

【0041】

前述の鍵テーブル操作処理などを実行するためのプログラム群はライブラリ化され、鍵処理モジュールと呼ばれる。暗号処理ブロック 12、コード認証処理ブロック 13、および CPU 固有鍵 15 等のセキュアハードウェア 20 に対するアクセスは、ステップ S4 でこの鍵処理モジュールの動作している区間だけに制限される。この鍵処理モジュールが動作し、セキュアハードウェア 20 にアクセス可能となる状態をアクセスレベル 1 と呼ぶことにする。アクセスレベル 1 は、プログラムカウンタが、固定領域にあるステップ S4 の鍵処理モジュールのアドレスを指しているかどうかの監視を行うハードウェアにより実現する。

【0042】

50

このアクセスレベル 1 に比較して前述の OS を含む一般のプログラムによる動作はアクセスレベル 2、またはアクセスレベル 3 に分類される。第 1 の実施例では OS はアクセスレベル 2 に分類され、ステップ S 5 で OS などの起動が行われると、ステップ S 6 で認証済みプログラムの実行が行われる。アクセスレベル 2 における認証済みプログラムは、アクセスレベル 1 におけるステップ S 4、すなわち鍵処理モニタに対し、鍵処理を依頼する事が出来、自空間の暗号化、データの暗号化や復号など鍵処理モニタ経由で間接的に行うことができる。このように CPU 外部からのプログラムであっても、認証されたものについてはアクセスレベル 2 として位置づけられ、鍵処理を行うことが出来るが、公開鍵以外のすべての鍵またはセキュアハードウェア 20 には直接アクセスすることは出来ないため、レベル 2 のプログラムに何らかの障害が生じたとしても公開鍵を除く鍵情報が外部に流出することはない。

【 0 0 4 3 】

アクセスレベル 3 における未認証のプログラムの実行は、ステップ S 7 においてステップ S 5 の OS などの起動の後に行われるが、このアクセスレベル 3 のプログラムは公開鍵以外のすべての鍵へのアクセス、および鍵処理モニタへの鍵処理の依頼などを行うことは一切できないものとする。なおステップ S 4 からステップ S 7 の処理は各アクセスレベルのプログラム間のプロセス間通信を利用して実行される。

【 0 0 4 4 】

以上説明したように第 1 の実施例においては、まずプロセッサの起動時に暗号化 ROM コード領域 14 に格納されたプログラムを用いて行われるブート処理の成功の時点でプログラムの基本的な信頼点が確立され、その後その信頼点を用いて OS を含む各種のプログラムの認証を行いながら信頼できるプログラム範囲を拡大していくことによって、システムのセキュリティレベルをプロセッサ自らが段階的に向上させるという目的が達成される。また運用開始後は認証単位毎にコードやデータの暗号化を行うことも可能となり、プログラムの間の秘密性保持に関して十分な信頼性を維持することができる。なおこの第 1 の実施例では、アクセスレベル 1 の処理をプロセッサのコアが実行するソフトウェアとして実現する方式を説明したが、このレベル 1 の処理の一部、または全てをマイクロコード、あるいはワイヤードロジックとして実現することも可能である。

【 0 0 4 5 】

図 4 は、図 2 のコード認証処理ブロック 13 と暗号処理ブロック 12 による処理の概要を示すフローチャートである。同図においてまずステップ S 10 のコード認証処理ブロックによる処理に続いて、ステップ S 11 で暗号処理ブロックによる処理が行われるものとする。

【 0 0 4 6 】

図 4 においてまずステップ S 12 で、例えば主記憶 17、または二次記憶 18 に格納されているプログラムに対するコード認証処理が実行される。この処理の詳細については後述する。そしてステップ S 13 で認証が成功したか失敗したかが判定され、失敗した場合にはステップ S 14 でコード実行に対する停止処理が実行される。

【 0 0 4 7 】

認証が成功した場合には、暗号処理ブロックによる処理が開始され、ステップ S 16 で暗号化のための鍵が、例えばページ単位に指定されているか否かが判定され、指定されていない場合にはステップ S 17 で乱数生成器などを用いてランダム鍵が生成され、指定されている場合にはステップ S 18 でその指定鍵が取り出される。ここで、鍵が指定されていない場合とは、新規にそのページが生成される場合などが含まれ、鍵が指定されている場合とは、生成されたページが一度ページアウト後、再度ページインする場合や、外部からの暗号化ページの格納などが含まれる。鍵が確定後、ステップ S 19 で暗号化ページエントリ、すなわち後述するトランスレーション・ルックアサイド・バッファ (TLB) 内のページ・テーブル・エントリ (PTE) が生成され、暗号化ページが割り当てられてコードまたはデータの暗号化が行われる。

【 0 0 4 8 】

10

20

30

40

50

図5は、同一プロセスの命令領域とデータ領域とにそれぞれ異なる暗号鍵が割り当てられてコードの暗号化が行われる場合の、コード認証とその暗号化処理の全体フローチャートである。同図においてステップS10、すなわちコード認証処理ブロックによる処理は図4における場合と同様である。

【0049】

図5においてコード認証が成功すると、ステップS21で命令領域に対する鍵としての命令鍵の指定があるか否かが判定され、指定がない場合にはステップS22でランダム鍵が生成され、指定がある場合にはステップS23でその指定鍵が取り出され、ステップS24でランダム鍵、または指定鍵が使用されて、暗号化命令ページ・テーブル・エントリ、すなわちPTEが生成され、暗号化ページが命令領域に割り当てられて命令領域の暗号化が行われる。

10

【0050】

その後ステップS26でデータ領域に対する暗号鍵としてのデータ鍵の指定があるか否かが判定され、指定がない場合にはステップS27でランダム鍵が生成され、指定がある場合にはステップS28で指定された鍵が取り出され、ステップS29でデータに対するページ・テーブル・エントリが生成され、暗号化ページが割り当てられてデータ領域に対する暗号化が実行される。

【0051】

続いて第1の実施例における暗号鍵の取得動作について図6から図9を用いて説明する。図6、および図7は、暗号鍵取得動作例(その1)におけるプロセッサ内部の構成例とその処理のフローチャートである。この例ではプロセッサ固有のRSA秘密鍵が予め安全な方式によってプロセッサ内部に保持されており、対応するRSA公開鍵は何らかの方法によってプロセッサの外部に出力され、外部から与えられる暗号鍵はこの公開鍵によって暗号化されているものとする。すなわち、例えばページ単位の暗号化、および復号用の暗号鍵は共通鍵であり、その秘密性保持のために公開鍵による再暗号化が必須のものとなっている。

20

【0052】

図6は、プロセッサ内部への暗号鍵設定処理実行のためのプロセッサ10の構成を示し、プロセッサの内部には必要なブロックとして暗号鍵設定部21、復号部22、プロセッサ固有RSA秘密鍵23、およびトランслーション・ルックアサイド・バッファ(TLB)24が備えられ、TLBの内部には前述のページ・テーブル・エントリ(PTE)に相当する論理アドレステーブル25、物理アドレステーブル26、および鍵テーブル27が備えられている。そして暗号鍵設定部21に対して、プロセッサ固有のRSA公開鍵で暗号化された暗号鍵を含む暗号鍵設定要求が外部から与えられる。

30

【0053】

図7は、暗号鍵取得処理のフローチャートである。同図において処理が開始されると、まずステップS31で暗号鍵設定部21によって暗号鍵設定要求が受け付けられ、ステップS32で復号部22によって受け取った暗号化暗号鍵がプロセッサ固有RSA秘密鍵23を用いて復号され、ステップS33で暗号鍵設定部21によって復号された暗号鍵がTLB24の内部の鍵テーブル27に格納されて処理を終了する。

40

【0054】

図8は、暗号鍵取得動作例(その2)におけるプロセッサの構成例である。同図においては、その1の例における図6と比較すると、プロセッサ10の内部に復号部22に代わって署名検証部28が備えられ、またプロセッサ固有RSA秘密鍵23の代わりに認証局公開鍵としての認証局証明書29が格納されている点が異なっている。この認証局証明書29はその不正な置き換えが不可能なようにプロセッサ内部に記録されているものとし、暗号鍵設定部21に対しては、認証局の署名が付けられた暗号鍵を含む暗号鍵設定要求が与えられるものとする。

【0055】

図9は、暗号鍵取得動作例(その2)における処理のフローチャートである。同図にお

50

いて処理が開始されると、まずステップ S 3 6 で暗号鍵設定部 2 1 によって暗号鍵が署名とともに受け取られ、ステップ S 3 7 で署名検証部 2 8 によって受け取った暗号鍵が署名と認証局公開鍵を用いて検証され、ステップ S 3 8 で検証が成功したか否かが判定され、成功した場合にはステップ S 3 9 で暗号鍵設定部 2 1 によって受け取った暗号鍵が T L B 2 4 の内部の鍵テーブル 2 7 に格納された後に、また検証が失敗した場合には直ちに処理を終了する。なおさらに暗号鍵の信頼性を向上させるためにその 1 の動作例、すなわち暗号鍵の秘密性保持と、動作例その 2、すなわち暗号鍵の身元証明とを組み合わせることも当然可能である。

【 0 0 5 6 】

図 1 0 は、第 1 の実施例において暗号化された命令領域の命令を実行中に不正命令を検出した場合の不正命令対応処理のフローチャートである。同図においてステップ S 4 1 で不正命令が検出されると、ステップ S 4 2 でその不正命令が暗号化ページ内の命令であるか否かが判定され、非暗号化ページ内の命令である場合にはステップ S 4 3 で通常的不正命令対応処理が行われるが、暗号化ページ内の命令であると判定されるとステップ S 4 4 で命令改ざんが行われたと判定され、その改ざんに対する命令改ざん対応処理としてプロセスのロックダウンや、保留中処理のキャンセルなどの処理が実行され、その命令コードの実行は停止される。

【 0 0 5 7 】

図 1 1 は、図 5 で説明したように同一プロセスの命令領域とデータ領域とに異なる暗号鍵が割り当てられている場合に、例えばデータ領域に格納されている命令コードの実行に先立ってその命令が不正命令として検出されることを防ぐための鍵付け替え処理のフローチャートである。このような命令コードのデータ領域への格納はプログラム D I O (P I O)、すなわちプログラムによる命令のコピーの実行時に起こるものである。

【 0 0 5 8 】

図 1 1 においてまずステップ S 4 6 で命令コードが P I O によってデータ領域にコピーされたものとし、ステップ S 4 7 で鍵付け替え処理が起動される。この処理ではステップ S 4 8 で命令が格納されていたデータページに対応するデータ P T E が読み出され、ステップ S 4 9 でそのエントリに格納されていた暗号鍵が取り出された後にその P T E が消去され、ステップ S 5 0 でそのデータ P T E の内容、すなわち暗号鍵を用いて、その暗号鍵が、例えば図 6 の鍵テーブル 2 7 に格納された命令 P T E が生成され、ステップ S 5 1 でその命令 P T E が T L B に書き込まれた後にステップ S 5 2 で命令がコピーされた領域への分岐が行われ、その後の処理、すなわちコピー領域に格納された命令の実行処理が行われる。

【 0 0 5 9 】

このように第 1 の実施例では、図 2 で説明したようにプロセッサ 1 0 の内部に実行ユニットとキャッシュを含むコア 1 1 が 1 つだけ備えられ、そのコア 1 1 がセキュアコアとしてセキュアプロセッサとしての動作の中心的役割を果たすものとしたが、マルチプロセッサシステム、あるいはマルチコアシステムと呼ばれるシステムでは、例えば複数のコアを、セキュア動作を実行するセキュアコアと、ノーマル動作を実行するノーマルコアとに分類し、処理を分担させることも可能である。そのようなプロセッサシステムを次に第 2 の実施例として説明する。

【 0 0 6 0 】

図 1 2 は、第 2 の実施例としてのプロセッサの基本構成ブロック図である。同図においては第 1 の実施例を示す図 2 と比較すると、コア 1 1 に代わってセキュアコア 3 1 とノーマルコア 3 2 とが備えられ、またこれらの 2 つのコア 3 1、3 2 と暗号処理ブロック 1 2、およびコード認証処理ブロック 1 3 との間にバスインタフェース 3 3 が備えられ、またセキュアコア 3 1 と暗号処理ブロック 1 2 との間では鍵制御が行われ、セキュアコア 3 1 とコード認証処理ブロック 1 3 との間で認証制御が行われ、さらに C P U 固有鍵 1 5 はセキュアコア 3 1 だけに接続されている点が異なっている。すなわち第 2 の実施例では、図 3 で説明したセキュアハードウェア 2 0 としての暗号処理ブロック 1 2、コード認証処理

10

20

30

40

50

ブロック 13、および CPU 固有鍵 15 に対する制御がセキュアコア 31 だけによって行われるという点に基本的な特徴がある。

【0061】

この第 2 の実施例では、セキュアハードウェア 20 に対するアクセスはセキュアコア 31 のみに限定される。第 1 の実施例ではセキュア動作としての図 2 のステップ S4 における鍵処理モニタの動作においてユーザのソフトウェアが介在する余地があり、そのため前述のようにプログラムカウンタのハードウェア監視によるアクセス制限などが行われるが、第 2 の実施例ではこのようなソフトウェアの介在がなく、ソフトウェアバグによる問題も発生しない。

【0062】

また、第 1 の実施例では例えば同一のコアを使用して時分割方式でアクセスレベル間の共有を行うことも必要となるが、第 2 の実施例ではコアが別になるため、アクセスレベル切替時点におけるレジスタクリアなどのソフトウェアに対する要求処理量も少なくなる。

【0063】

図 13 は、第 2 の実施例におけるプロセッサの基本処理フローチャートである。第 1 の実施例における図 3 と比較して異なる処理を中心にその処理を説明する。図 12 におけるセキュアコア 31 とノーマルコア 32 とが基本的に対等な関係にあるものとする、電源が投入された時点でそれぞれのコアは、暗号化 ROM コード領域 14 に格納されたプログラムを用いてブート処理を実行する。すなわち前述のようにステップ S1 でセキュアコアによるブート処理において CPU 固有鍵 15 を用いて暗号化されたプログラムが復号され、ブート処理が実行される。このブート処理が成功した場合には、前述のようにその状態がプログラムの基本的な信頼点として定義され、セキュアコアはその後、例えばもっぱら鍵処理モニタとしての動作を継続することになる。

【0064】

これに対してノーマルコア 32 は、主として OS などのアクセスレベル 2 に相当する処理を担当することになる。図 13 のステップ S3 ではセキュアコア側で二次記憶上のプログラムの認証処理が実行されるのに対応してノーマルコア 32 側で電源がオンとなり、ステップ S55 で暗号化 ROM コード領域 14 内のプログラムによるブート処理が実行される。この時点では、セキュアコア 31 によって暗号化 ROM コード領域 14 内のプログラムは絶対的に信頼可能であることが判明しているものとする、ノーマルコア側のブート処理は基本的に問題なく終了し、ステップ S5 における OS などの起動処理が続いて実行されることになる。

【0065】

図 14 は、第 2 の実施例においてセキュアコアとノーマルコアとが対等な関係でなく、セキュリティの厳密な適用を行うために、セキュアコアによってノーマルコアの制御が行われる場合のプロセッサの構成ブロック図である。セキュアコア 31 とノーマルコア 32 とが基本的に対等な関係にある図 12 と比較してプロセッサの構成要素は同一であるが、セキュアコア 31 からノーマルコア 32 に対してコア制御信号が与えられる点が異なっている。このコア制御信号の具体的な例としてはリセット信号や割込み信号などが挙げられる。

【0066】

図 15 は、図 14 のプロセッサによる全体処理のフローチャートである。同図においてセキュアコア 31 側では、ステップ S1 のブート処理に続いてステップ S57 でステップ S2 の代わりに、鍵テーブル操作処理、鍵管理処理、および認証処理ブロックのセットアップに加えてシステム監査が行われる。このシステム監査では、システム構成の変更の有無や、二次記憶上のプログラムの変更の有無などの検証が行われ、システムのセキュリティ機能、およびシステム構成に問題がないことが確認される。

【0067】

その後ステップ S58 でセキュアコア 31 側からノーマルコアの起動が行われ、これに対応してノーマルコア 32 側では、ステップ S59 で暗号化 ROM コード領域 14 に格納

10

20

30

40

50

されたプログラムを用いた起動処理が実行される。その後の処理は、例えば図3における場合と同様である。

【0068】

図16は、図14のプロセッサにおけるセキュアコア31によるノーマルコア32の制御処理の1つとしてのノーマルコアの停止制御処理の説明図である。同図において、例えばノーマルコア側でステップS6の認証済みプログラムの実行において、セキュアコア31側にデータなどの認証のための鍵処理が依頼され、ステップS4における鍵処理モニタの動作において認証の失敗や、セキュリティ基準への違反が検出された場合には、セキュアコア31側からの指示によってノーマルコア32による処理、すなわちステップS6の認証済みプログラムの実行、およびステップS7の未認証プログラムの実行が停止される。

10

【0069】

図17は、図14におけるセキュアコア31によるノーマルコア32の制御処理のフローチャートである。セキュアコア31側ではステップS61でブート処理が実行され、ステップS62でその処理が完了すると、ノーマルコア32側への起動制御が行われ、ステップS63でノーマルコアの起動が行われ、ステップS64で公開鍵以外の鍵や認証処理を必要としない通常処理がノーマルコア側で実行される。セキュアコア31側では、常にステップS65でノーマルコア32側から送られる監視情報を用いた認証・監視処理を行っており、ステップS66でエラー発生があったか否かを判定し、発生がない場合にはステップS65以降の処理を続行し、エラー発生があった場合にはノーマルコア32側に対する停止要求、または割り込みを行ってノーマルコア32側の処理を停止させる。セキュアコアによるノーマルコアの制御については、前述のようにリセット信号をその例として用いるものとしたが、その他の方法としてはCPUに対するNMI(マスク不可割り込み)を利用することもできる。

20

【0070】

図18は、第2の実施例において鍵生成機構を有するプロセッサの構成ブロック図である。同図において図12の構成に加えて、鍵生成機構34を備えている点が異なっている。

【0071】

図19は、第2の実施例におけるセキュアコアによる鍵の生成と、生成された鍵を用いた暗号化処理の説明図である。同図においてプロセッサ内のセキュアコア31はCPU固有鍵15、および鍵生成機構34を用いて公開鍵Ke、N、および秘密鍵Kd35を生成し、例えばノーマルコア32を経由して公開鍵Ke、Nを外部に通知するものとする。このときノーマルコア32側には秘密鍵Kdが渡されることはなく、ノーマルコア32は前述のように公開鍵以外の鍵処理を実行することはできない。

30

【0072】

そして例えば外部において公開鍵と原文Pを用いて暗号化された暗号文Cがノーマルコア32に入力されると、ノーマルコア32は秘密鍵Kdを保持していないためセキュアコア31に復号処理を依頼する。セキュアコア31は秘密鍵Kdを用いて原文Pを復号する。

40

【0073】

次に本発明の第3の実施例について説明する。図20は、第3の実施例におけるプロセッサの基本構成ブロック図である。同図においてプロセッサ40は実行ユニット41、ロードストアユニット42、暗号化回路43、および復号化回路44を備え、またロードストアユニット42は、キャッシュメモリ45とメモリ管理ユニット46を備えている。

【0074】

この第3の実施例は前述の第1、および第2の実施例と同様に基本的にはセキュアな動作を実行するプロセッサであるが、プロセッサ40の内部で第1の実施例における暗号処理ブロックと同様にストアデータの暗号化を行う暗号化回路43と、フェッチされる命令を含むロードデータを復号する復号化回路44とに対して、実行ユニット41からストア

50

用の暗号化鍵とロード用の復号化鍵の指定が行われるところに基本的な特徴がある。

【0075】

図20の第3の実施例において、実行ユニット41からロードストアユニット42に対しては、コマンドとストアデータとしての平文が与えられ、ロードストアユニット42から実行ユニット41に対しては平文としてのロードデータが与えられる。このうちコマンドはロードストアユニット42を介して、例えば図2で説明した主記憶や二次記憶に与えられるが、平文としてのストアデータは暗号化回路43に与えられ、暗号化されたストアデータとして、例えば主記憶に出力され、また例えば主記憶から入力される暗号化されたロードデータは復号化回路44によって復号され、平文としてのロードデータとしてロードストアユニット42に与えられる。

10

【0076】

図21は、第3の実施例において暗号化鍵、復号化鍵を格納する鍵テーブルメモリを備えるプロセッサの構成ブロック図である。同図において鍵テーブルメモリ47はストアデータ暗号化用の暗号化鍵を格納するものであり、また鍵テーブルメモリ48はロードデータを復号するための復号化鍵を格納するものである。実行ユニット41からは鍵テーブルメモリ47に対してストア用の鍵番号指示とストア用の暗号化鍵の更新の指示が与えられ、また鍵テーブルメモリ48に対してはロード用の鍵番号の指示とロード用復号化鍵の更新の指示が与えられる。鍵テーブルメモリの構成については後述する。

【0077】

図22は、第3の実施例においてフェッチされる命令の復号を行うための命令フェッチ用復号化鍵を格納する鍵テーブルメモリを備えるプロセッサの構成ブロック図である。同図において実行ユニット41は、例えば主記憶に格納されている命令をフェッチする命令アクセス状態の処理を行っており、例えば主記憶からのロードデータとして命令フェッチデータが復号化回路44に与えられ、このとき実行ユニット41は鍵テーブルメモリ48に対して命令アクセス状態フラグを与え、復号化回路44は鍵テーブルメモリ48から出力される命令フェッチ用復号化鍵を用いて命令フェッチデータの復号を行い、平文としての命令フェッチデータはロードストアユニット42を介して実行ユニット41に与えられる。実行ユニット41からは鍵テーブルメモリ48に対して、必要に応じて命令フェッチ用復号化鍵の更新の指示が与えられる。

20

【0078】

図23は、第3の実施例において鍵テーブルメモリに対して使用すべき鍵番号の指示を与える鍵選択レジスタを備えるプロセッサの構成ブロック図である。同図においてストア用暗号化鍵を格納する鍵テーブルメモリ47と実行ユニット41の間にストア用鍵番号指示を鍵テーブルメモリ47に与える鍵選択レジスタ51が、またロード用復号化鍵を格納する鍵テーブルメモリ48と実行ユニット41の間に鍵テーブルメモリ48にロード用鍵番号指示を与える鍵選択レジスタ52が備えられる。実行ユニット41から鍵選択レジスタ51に対してはストア用鍵選択レジスタの更新指示が与えられ、また鍵選択レジスタ52に対してはロード用鍵選択レジスタの更新指示が与えられる。

30

【0079】

すなわち図21では実行ユニット41からは実行命令のそれぞれに対応して鍵番号の指示が出力されるのに対して、図23では命令のある区間毎にレジスタの更新指示が与えられ、次の更新指示が与えられるまでは同一の鍵を使用して暗号化/復号化が行われる。なお実行ユニットから鍵テーブルメモリに対して直接に鍵番号指示を与える経路と、鍵選択レジスタを経由した経路との両方を設け、例えば実行ユニット41が実行命令に対応してどちらの経路の指示を用いるべきかの信号を鍵テーブルメモリに与えるような構成も当然可能である。

40

【0080】

図24は、第3の実施例において実行ユニットの命令アクセス状態に対応する鍵選択レジスタを備えるプロセッサの構成ブロック図である。同図においては図22における同様に、実行ユニット41は、例えば主記憶から命令フェッチを行うべき命令アクセス状態

50

であり、実行ユニット 4 1 から鍵選択レジスタ 5 2 に対して命令アクセス状態フラグが与えられ、鍵選択レジスタ 5 2 は、それに対応して命令フェッチ用の鍵番号指示を、命令フェッチ用復号化鍵を格納する鍵テーブルメモリ 4 8 に与える。

【 0 0 8 1 】

図 2 5 は、第 3 の実施例における鍵テーブルメモリの構成例の説明図である。同図において鍵テーブルメモリには、暗号鍵とその属性とが対応して格納されており、実行ユニット 4 1 から直接に、あるいは鍵選択レジスタを介して鍵番号の指示が与えられ、その鍵番号がリードアドレスとして用いられ、暗号化鍵、または復号化鍵が暗号化回路 4 3、または復号化回路 4 4 に対する暗号化方式の指定情報や、暗号化の可否を指示する属性データとともに与えられる。また実行ユニット 4 1 から与えられる鍵更新番号指示がライトアドレスとして用いられ、鍵更新データの書込みが行われる。

10

【 0 0 8 2 】

各エントリの属性データはエントリの有効/無効、暗号化のオン/オフ、暗号化方式および暗号化モードなどを示し、暗号鍵は指定された暗号化方式に依存するものとなる。なお暗号化のオン/オフを指示するデータは後述するように暗号化/復号化を行うことなく平文データをロード、ストアする場合の指示に対応する。

【 0 0 8 3 】

図 2 6 は、第 3 の実施例における暗号化回路、および復号化回路の構成例の説明図である。例えば図 2 0 の復号化回路 4 4 は、復号パイプ 5 5 とバスアービタ 5 7 とによって基本的に構成され、復号パイプ 5 5 は実行ユニット 5 1 からのコマンドバッファ 5 9 を介したコマンド情報の入力に対応して動作する。復号パイプ 5 5 は、例えば主記憶からバスを介して入力される暗号データを平文データに復号するための N 段のパイプであり、この N 段のパイプは共通鍵暗号処理一段の概念的な例である処理 5 6 が N 段接続されたものである。復号パイプ 5 5 から出力される平文データは、バスアービタ 5 7 を介して、例えば図 2 0 のキャッシュメモリ 4 5 に格納される。

20

【 0 0 8 4 】

暗号化回路 4 3 は暗号パイプ 6 0 とバスアービタ 6 1 とによって基本的に構成される。暗号パイプ 6 0 に対しては、キャッシュメモリ 4 5 から例えば 3 2 b i t の平文データが与えられ、実行ユニット 4 1 から指定される暗号化の鍵を用いて N 段のパイプによって暗号化された暗号データが出力され、その暗号データはバスアービタ 6 1 を介して、例えば主記憶に接続されたバスに与えられる。暗号パイプ 6 0 の動作は復号パイプ 5 5 と同様にコマンドバッファ 5 9 を介して実行ユニット 4 1 から与えられるコマンド情報によって制御される。また暗号パイプ 6 0 の各段における処理の基本構造は復号パイプ 5 5 におけると同様である。さらに暗号方式としては A E S 1 2 8、D E S、および S C 2 0 0 0 などの各種の暗号化方式を用いることができる。なお A E S 方式としては A E S 1 9 2、A E S 2 5 6 の仕様も規定されている。

30

【 0 0 8 5 】

なお例えばバスアービタ 6 1 は主記憶装置や二次記憶装置に接続されたバスに対する調停を行うものであり、本発明におけるセキュアプロセッサの動作とは基本的に無関係である。

40

【 0 0 8 6 】

図 2 7 は、第 3 の実施例において全てのデータを暗号化するのではなく、一部のデータを平文データのままで、例えば主記憶との間で入出力するためのデータ追い越し機能付き暗号化回路と復号化回路の構成を示すブロック図である。同図において暗号化回路と復号化回路の基本的な構成は図 2 6 におけると同様であるが、例えば暗号化回路側では、キャッシュメモリ 4 5 から与えられる平文データのうちで暗号化の必要がないデータについては暗号パイプ 6 0 を介することなく、直接にバイパスセレクタ 6 3 にそのデータが与えられ、暗号パイプ 6 0 から出力された暗号化データとともに複数のバイパスバッファ 6 4 の何れかに格納され、バスアービタ 6 1 を介して、例えば主記憶に接続されたバスに与えられる。

50

【0087】

バイパスセクタ63による平文データ、または暗号化データの選択もコマンドバッファ59を介した実行ユニット41からのコマンド情報によって制御される。暗号パイプ60による処理は時間を要するため、暗号化の必要のない平文データは、バイパスセクタ63の制御によって暗号化データを追い越して主記憶側に与えられることが可能となる。なお図27で暗号化に必要な鍵は鍵レジスタ69を介して暗号パイプ60に与えられている。

【0088】

例えば主記憶に接続されたバスからのデータのうち、暗号化されていない平文データは復号パイプ55を経由することなく直接にバイパスセクタ66に与えられ、復号パイプ55によって復号された平文データとともにバイパスセクタ66によって複数のバイパスバッファ67のいずれかに格納され、バスアービタ57を介してキャッシュメモリ45に出力される。

【0089】

図28は、第3の実施例においてライトスルーキャッシュ方式に対応するためのリード・モディファイ・ライト方式の説明図である。キャッシュメモリ45がライトスルー方式を採用している場合には、ストア時にキャッシュミスが発生するとキャッシュメモリ45にそのデータが格納されず主記憶にそのままデータの格納が行われる。ストアすべきデータが、例えば1バイトしか無いような場合には、主記憶に1バイトデータの格納が行われる。しかしながら第3の実施例においては、基本的にストアデータは暗号化回路43によって暗号化された後に主記憶に格納されるものであり、一般に暗号化の処理においてはストアデータとしてある程度の量のデータを必要とし、1バイトだけのデータを暗号化して主記憶に格納したとしてもその正しい復号は困難である。

【0090】

図28のロードストアユニット42は、例えば1バイトのデータを主記憶に格納する必要があるときに、暗号化の処理に必要な長さのデータを主記憶からロードし、ロードされたデータとストアすべき1バイトのデータを結合し、結合されたデータを暗号化して主記憶に格納するリード・モディファイ・ライト動作を行うものである。

【0091】

すなわち、例えば1バイトのデータをキャッシュにストアすべきキャッシュストア命令(1)が(2)でキャッシュミスと判定されると、(3)でキャッシュメモリ45から主記憶に対してコマンドとしてのロードが発行され、(4)で復号化回路44を介して平文のロードデータがリード・モディファイ・ライト(RMW)バッファ71に格納され、(5)でストアすべきデータがRMWバッファ71に与えられ、(6)でストアすべきデータとロードデータとを結合したデータが暗号化回路43に与えられ、(7)でコマンドとしてのストアが主記憶に対して発行される。

【0092】

次に本発明の第4の実施例について説明する。この第4の実施例では、第3の実施例において暗号化回路によって使用される暗号化鍵、および復号化回路によって使用される復号化鍵の例えば鍵番号が実行ユニット41によって指定されるのに対して、実行ユニット41による命令実行時にストア、またはロード対象となるデータのアクセスアドレスが実行ユニット41によって指定され、そのアドレスに応じて暗号化鍵、または復号化鍵が選択される点が異なっている。

【0093】

図29は、第4の実施例におけるプロセッサの基本構成ブロック図である。同図においてプロセッサ40は実行ユニット41、暗号化回路43、復号化回路44に加えて、実行ユニット41から与えられるアドレスに対応してストア用暗号化鍵を暗号化回路43に与え、ロード用復号化鍵を復号化回路44に与える鍵テーブルメモリ73を備えている。

【0094】

図30は、実行ユニットから指定されるストアデータ、またはロードデータの論理アド

レスに対応して鍵が選択されるプロセッサの構成ブロック図である。同図においてプロセッサ40は、図29と異なってストア用暗号化鍵を格納する鍵テーブルメモリ74と、ロード用復号化鍵を格納する鍵テーブルメモリ75とを備えるとともに、キャッシュメモリ45、メモリ管理ユニット46を備えるロードストアユニット42を、例えば図20とおけると同様に備えている。実行ユニット41からロードストアユニット42に与えられるアドレス、すなわちストアデータ、またはロードデータのアドレスは論理アドレスであり、この論理アドレスが鍵テーブルメモリ74、または75に与えられてストア用暗号化鍵、またはロード用復号化鍵が選択され、それぞれ暗号化回路43、または復号化回路44に与えられる。また実行ユニット41から鍵テーブルメモリ74に対してはストア用暗号化鍵の更新の指示が、鍵テーブルメモリ75に対してはロード用復号化鍵の更新の指示が与えられる。

【0095】

図31は、第4の実施例においてデータの物理アドレスに対応して鍵が選択されるプロセッサの構成ブロック図である。同図を図30と比較すると、鍵テーブルメモリ74と75のそれぞれに対してロードストアユニット42からストアデータの物理アドレス、またはロードデータの物理アドレスが与えられ、そのアドレスに対応してストア用暗号化鍵が暗号化回路43に与えられ、ロード用復号化鍵が復号化回路44にそれぞれ与えられることになる。

【0096】

図32は、第4の実施例における鍵テーブルメモリの構成図である。同図を第3の実施例における図25と比較すると、実行ユニット側からデータのアクセスアドレスとして0bit目から31bit目までの32bitが与えられると、そのアドレスをリードアドレスとして格納されている暗号鍵が選択され、暗号化回路43、または復号化回路44に暗号属性とともに与えられる。メモリのリードアドレスとして4kバイト毎に異なる鍵が使用される場合には、アドレスの12bit目から31bit目が使用されて暗号鍵の選択が行われる。なおこの4kバイトは後述するように例えば主記憶における1ページの大きさに相当する。またこの4kバイトを暗号化のアドレス単位と呼ぶことにすると、鍵テーブルメモリのエントリデータには全エントリ数×アドレス単位分を除いたアドレスタグが含まれる。例えば全エントリ数が32(5ビット)であれば、アドレスの17ビット目から31ビット目までがタグとなる。

【0097】

図33は、第3の実施例において複数のウェイの構成を持つ鍵テーブルメモリの説明図である。同図において鍵テーブルメモリは、鍵テーブル1から鍵テーブル4までの複数のテーブルによって構成されており、実行ユニット側から与えられるアクセスアドレスに対応して4つのテーブルのうちの何れかに格納されている鍵と暗号属性とが選択され、暗号化回路43、または復号化回路44に与えられる。

【0098】

図34は、連想記憶方式を用いる鍵テーブルメモリの構成例の説明図である。同図においてはアクセスアドレス32bitが比較選択器77によって格納されている暗号鍵のそれぞれに対応する対象アドレス範囲の何れかに分類され、その範囲に対応する暗号鍵が選択されて、暗号属性とともに暗号化回路43、または復号化回路44に与えられる。なお図34では全エントリ数に無関係にアドレス単位分を除いたアドレスタグがエントリに含まれる。アドレス単位が4kバイトのときには、アドレスの12ビット目から31ビット目がタグとなる。

【0099】

図35は、第4の実施例においてデータの論理アドレス、または物理アドレスの何れかに対応して鍵を選択するプロセッサの構成ブロック図である。同図において実行ユニット41からはデータの論理アドレスが、またロードストアユニット42からは物理アドレスがそれぞれ鍵テーブルメモリ74、75に与えられ、また実行ユニット41から鍵テーブルメモリ74に対してはストアデータの論理アドレスと物理アドレスの選択指示が与えら

れ、鍵テーブルメモリ75に対してはロードデータの論理アドレスと物理アドレスの選択指示が与えられる。そしてこれらの選択指示に対応して論理アドレス、または物理アドレスの何れかに対応する鍵が選択されて暗号化回路43、復号化回路44に与えられる。

【0100】

図36は、鍵テーブルメモリに対して論理アドレスと物理アドレスの選択指示を与える鍵選択レジスタを備えるプロセッサの構成例である。同図を図35と比較すると実行ユニット41と鍵テーブルメモリ74、75との間にそれぞれ鍵選択レジスタ78、79が備えられ、それぞれストアデータに対する論理アドレスと物理アドレスの選択指示、ロードデータに対する論理アドレスと物理アドレスの選択指示を鍵テーブルメモリ74、75に出力する。実行ユニット41から鍵選択レジスタ78、79に対してはそれぞれ鍵選択レジスタの更新指示が与えられる。

10

【0101】

図37は、図35と図36における鍵テーブルメモリの構成例を示す。同図において鍵テーブルメモリは、物理アドレス鍵テーブルと論理アドレス鍵テーブルとを備え、それぞれ物理アドレスと論理アドレスに対応して物理鍵、論理鍵を出力し、実行ユニット41側からの鍵選択指示、または鍵選択レジスタからの選択指示に対応して物理・論理鍵選択部81によって物理鍵、または論理鍵の何れかが暗号属性とともに暗号化回路43、または復号化回路44に出力される。

【0102】

図38は、第4の実施例において鍵テーブルメモリの内容を鍵テーブルとしてロードストアユニット42の内部のメモリ管理ユニット(MMU)46に備えたプロセッサの構成例である。

20

【0103】

図39、および図40は、このメモリ管理ユニット内の鍵情報の格納形式とキャッシュメモリアクセス方式の説明図である。一般的にMMU46の内部のトランスレーション・ルックアサイド・バッファ(TLB)には論理アドレスと物理アドレスとの対応が、例えば物理メモリ内の各ページに対応して各エントリに格納されているが、図39ではTLBの各エントリにそのページに対応する鍵情報を格納し、例えばデータアクセスアドレスが論理アドレスである場合には論理アドレスが一致するエントリが選択され、そのエントリに対応するデータの属性とアクセス属性とが属性チェック83によってチェックされ、キャッシュコマンド生成84によって生成されたコマンドがキャッシュメモリ45に送られる。

30

【0104】

キャッシュメモリ45側では、受け取ったコマンドの内容に対応してタグを検索し、キャッシュヒットの場合には直ちにデータ応答を実行ユニット41側に返すことになるが、キャッシュミスの場合には暗号化回路43、復号化回路44を含む暗号化・復号化バスインタフェース85に対してタグに対応するコマンドが発行される。このときエントリから読み出された鍵情報や物理アドレスが使用され、例えば主記憶からの応答データが復号化された後にキャッシュメモリに格納され、実行ユニット41にデータ応答が返されることになる。

40

【0105】

図40は、メモリ管理ユニット内にTLBに代わるアドレス・マップ・レジスタ(AMR)を設けた場合の鍵情報格納形式の説明図である。同図においてはTLBの格納内容に対応する情報がメモリでなくレジスタに格納されており、例えばページサイズを可変とすることができ、大きなデータ領域を1つのエントリでカバーすることも可能となる。

【0106】

図41は、ロードストアユニットの内部のメモリ管理ユニット(MMU)が動作停止の状態、すなわちOFFの状態において実行ユニット41から暗号化鍵を暗号化回路43に与え、復号化鍵を復号化回路44に与えるプロセッサの構成例である。同図においてMMUのON/OFF信号は暗号化回路43と復号化回路44に与えられ、その信号がOFF

50

のときには暗号化回路 4 3、復号化回路 4 4 は実行ユニット 4 1 から与えられる鍵、ON のときにはメモリ管理ユニット 4 6 の内部の TLB 8 7、または AMR 8 8 から与えられる鍵を用いて暗号化、または復号化の処理を実行することになる。

【0107】

図 4 2 は、図 4 1 の暗号化回路、復号化回路における鍵の切り替え方式の説明図である。同図において暗号化回路、および復号化回路の構成は第 3 の実施例における図 2 6 と基本的に同じであるが、鍵セクタ 9 0 が追加され、実行ユニットから与えられる MMU ON/OFF 信号の値に応じて OFF の時には実行ユニットから与えられる鍵が、ON の時には TLB、または AMR から与えられる鍵の何れかが鍵セクタ 9 0 によって選択され、暗号パイプ 6 0、復号パイプ 5 5 に与えられる。

10

【0108】

図 4 3 は、第 3 の実施例、および第 4 の実施例における実行ユニットの入出力信号の説明図である。まず第 3 の実施例における図 2 0 では、出力信号としてロード復号化鍵、ストア暗号化鍵、ストアデータ、およびコマンド、入力信号としてロードデータが必須の信号（印）であり、アクセスアドレス、ロード・ストア状態信号などは構成的には存在する信号（印）である。

【0109】

図 2 1 に対しては、ロード復号化鍵の代わりにロード鍵番号指示、ストア暗号化鍵の代わりにストア鍵番号指示の出力信号が必須のものとなる。また図 2 1 では、鍵テーブルメモリに対する更新が、実行ユニットから見てレジスタアクセスと等価であるため、レジスタ関連の入出力信号も必須となる。

20

【0110】

図 2 2 に対しては、命令アクセス状態に対応する入出力信号が必要であり、出力信号として実行状態信号、入力信号として命令フェッチデータが必須となる。

図 2 3、図 2 4 では、図 2 0、図 2 1 に加え鍵選択レジスタを使用するため、レジスタ関連の入出力信号も必須となる。

【0111】

以下説明を簡略化し、特徴的な部分について説明すると図 2 1 + 図 2 2 + 図 2 3 では 3 つの図を組み合わせた場合の入出力信号に加えて、スーパーバイザ、あるいはユーザのいずれに対応するプロセスの実行であることを示すスーパーバイザ・ユーザ状態信号とコンテキスト、すなわちプロセスの ID（識別子）のデータが追加された場合が示されている。これらスーパーバイザ/ユーザ状態信号とコンテキスト ID データは、第 3 の実施例において実行ユニットから出力される鍵番号指示の信号に加えて暗号化鍵、復号化鍵の選択のために用いられる。

30

【0112】

図 2 9 以降は第 4 の実施例に対応するものであり、データへのアクセスアドレスが必須の出力信号となるとともに、図 3 5、図 3 6 では論理アドレスと物理アドレスの何れかを選択するための鍵選択指示信号が出力される。

【0113】

図 3 8 に対しては、メモリ管理ユニットの内部の TLB に鍵テーブルが併設されるために、レジスタ関係の信号が構成的には存在する信号となり、またスーパーバイザ・ユーザ状態信号とコンテキスト ID データとが追加された場合が示されている。これら追加された信号も第 3 の実施例と同様にアクセスアドレスとともに暗号化鍵、復号化鍵の選択に用いられる。

40

【0114】

図 4 1 に対しては、メモリ管理ユニット（MMU）の ON/OFF を示す状態信号の値に対応して実行ユニットから出力される鍵が用いられる場合と、例えば TLB から出力される鍵が用いられる場合とがあり、スーパーバイザ・ユーザ状態信号、コンテキスト ID データも追加されて、すべての入出力信号が必須のものとなっている。

【0115】

50

以上のように第3、第4の実施例ではデータや命令コードの暗号化/復号化用の鍵が実行ユニットから指定されるために、実行される命令に対応したレベルで暗号化処理を行うことも可能となる。また、鍵選択レジスタ、あるいはアクセスアドレスによる暗号化/復号化鍵の指定により、プログラム単位またはアクセス単位での暗号化処理も可能であり、様々な状況に応じて使い分けることが可能である。

【0116】

続いて本発明の第5の実施例について説明する。この第5の実施例としては、例えば第1の実施例としてのセキュアプロセッサのセキュア動作を実現するためのより詳細な構成を示し、その構成に対応させてプロセス(プログラム)の信頼点を拡大していくための認証鍵の設定や、プロセスの認証などの動作について詳細に説明する。

10

【0117】

図44は、第5の実施例を説明するためのプロセッサ内の必要な機能構成図である。同図においてプロセッサ100は、物理メモリ101、例えば図2では主記憶17と、I/O装置102、例えば二次記憶18と接続されている。

【0118】

プロセッサ100は、物理メモリ101、およびI/O装置102へのアクセスを制御するメモリアクセス制御部105、実行すべき命令を解釈する命令解釈部106、実行コードの格納されているページの認証などを行う認証部107、例えば認証後のページの暗号化/復号などを行う暗号化/復号、署名生成/検証部108、プロセスの生成時にそのプロセス、すなわちコンテキストに対応するセキュアコンテキスト識別子を生成するセキュアコンテキスト識別子生成部109、プロセスの消滅時に対応する識別子を消滅させるセキュアコンテキスト識別子消滅部110、暗号化などに用いられるプロセッサ固有鍵111、例えば物理メモリ101に格納されている物理ページに対応する認証情報を格納する認証情報一次格納部112、メモリアクセス時に直接メモリアクセスを行うためのセキュアDMA113を備えている。

20

【0119】

プロセッサ100の内部には、さらに例えば図39で説明したトランスレーション・ルックアサイド・バッファ(TLB)114とコンテキスト情報格納部115が備えられている。TLB114には、例えば物理ページに対応させて論理アドレスと物理アドレスとの対応などを示すページ・テーブル・エントリ(PTE)122が格納され、コンテキスト情報格納部115にはプログラムカウンタ(の値を保持するカウンタ)117、セキュアコンテキスト識別子を格納するセキュアコンテキスト識別子レジスタ118、認証に必要な鍵を格納する認証鍵レジスタ119、およびレジスタ群120を備えている。

30

【0120】

また物理メモリ101には、例えば実行コードが物理ページ124の単位で格納されており、I/O装置102には実行コードやデータがページ125の単位で認証情報126が付加された形式で格納されている。なお第7の実施例では、セキュアコンテキスト識別子レジスタ118にセキュアコンテキスト識別子が格納されたコンテキスト(プロセス)の実行コードの認証は認証鍵レジスタ119に格納される認証鍵を用いて行われる。

【0121】

図45は、プロセッサ上で動作するプログラム、例えばユーザによって使用されるプログラムが起動されて、コンテキスト生成命令を発行した時点でのそのコンテキストに対応するセキュアコンテキスト識別子の生成方法の説明図である。コンテキスト生成命令は、プロセッサ内部の命令解釈部106に与えられ、その解釈結果に対応してセキュアコンテキスト識別子生成部109によってセキュアコンテキスト識別子が生成され、セキュアコンテキスト識別子レジスタ118にその値がセットされる。第5の実施例ではセキュアコンテキスト識別子レジスタ118への値のセットはこの方法によってのみ可能となるように構成される。これによってセキュアコンテキスト識別子を改ざんし、他のコンテキストになりすますことは不可能となる。なおコンテキストは基本的にオブジェクト指向プログラミングにおける概念であり、より一般的にはプロセス、すなわちプログラムの実行状態

40

50

に対応するが、この第5の実施例ではプロセスの代わりにコンテキストという用語を用いる。

【0122】

図46は、セキュアコンテキスト識別子の具体的な生成方法の説明図である。その生成には図に示すように乱数発生器127を用いることも、また単調増加カウンタ128を用いることもできる。乱数として同じ値が生成される確率は0でなく、カウンタの値も一巡すると同じ値になるため、厳密には識別子としてのユニーク性が保証できないが、十分長いビット長のセキュアコンテキスト識別子を用いることによって実質的に問題が起きないようにすることができる。

【0123】

あるいは図に示すようにプロセッサが持つ既存のコンテキストIDと乱数発生器127の出力とを結合部129で結合するか、単調増加カウンタ128の出力と結合することによってセキュアコンテキスト識別子を生成しても良い。なお既存のコンテキストIDは、例えばOSによって設定される任意の値であり、一般にユニーク性が保証されているものではない。

【0124】

図47は、セキュアコンテキスト識別子消滅方法の説明図である。同図においてプロセッサ上で動作するプログラムがコンテキスト消滅命令を発行した場合に、プロセッサはセキュアコンテキスト識別子レジスタ118の内容を無効とする。無効とする方法は0クリアでも良く、レジスタ中に有効/無効を表すフラグの格納領域を設け、そのフラグを無効にセットしても良い。

【0125】

図48は、図44において例えばページ単位の実行コード125に付加される認証情報126の説明図である。プロセッサ内のコンテキスト情報115の内部の認証鍵レジスタ119には認証情報126を用いて実行コード125を認証処理するための鍵が格納される。認証情報として、例えばRSAによる電子署名を用いる場合には、認証鍵はRSA公開鍵となり、共通鍵系のSHA(セキュアハッシュアルゴリズム)-1 HMAC(ハッシュベースド・メッセージ・オーセンティケーション・コード)を用いる場合には認証鍵は20バイトの値となる。

【0126】

認証鍵レジスタ119への鍵の格納は、OSによるコンテキストの生成時、すなわちコンテキスト情報が初期化される段階で行われ、同時に認証鍵の正当性のチェックがおこなわれる。認証鍵自体が悪意を持った者によって生成され、その鍵によって悪意を持った実行コードに対応する認証情報が生成されてしまうと、認証処理自体は何の問題もなく成功し、プロセッサによる認証機能が働かなくなることになる。従って認証鍵の正当性をいかに保証するかは重要な課題である。

【0127】

図49は、認証鍵が公開鍵の場合の認証鍵レジスタへの鍵設定方式の説明図である。同図において認証鍵はRSA公開鍵であるものとし、認証局証明書、すなわち認証局公開鍵134は、例えばプロセッサ内に工場出荷時に埋め込まれ、それ以降その置き換えや変更は不可能となっているものとする。認証鍵レジスタ119に設定すべき認証鍵は認証局秘密鍵による署名が付与された形で、例えばコンテキスト生成の時点でプロセッサに対して認証鍵設定命令として与えられ、その命令は命令解釈部106によって解釈され、署名検証部108によって認証鍵の検証が行われた後に認証鍵レジスタ119に格納される。これにより認証局のお墨付きのある公開鍵のみが認証鍵レジスタに設定される。

【0128】

図50は、図49における認証鍵設定処理のフローチャートである。同図においてまずステップS71で命令解釈部106によって認証鍵設定命令がフェッチされ、ステップS72で署名検証部108によってフェッチされた公開鍵が署名と認証局公開鍵を用いて検証され、ステップS73で検証が成功した否かが判定され、成功した場合には命令解釈部

10

20

30

40

50

106によってフェッチした設定命令に含まれる公開鍵が認証鍵レジスタ119に格納された後に、検証が失敗した場合には直ちに処理を終了する。

【0129】

図51は認証鍵が共通鍵の場合の鍵設定方式を示し、図52は鍵設定処理のフローチャートである。証明鍵として共通鍵を用いる場合には、実行コードに対して認証情報を付加する側から安全な方法で認証鍵を受け取る必要がある。ここではRSA公開鍵を用いて暗号化されたHMAC鍵を認証鍵設定命令とともに受け取るものとし、プロセッサ側でプロセッサ固有RSA秘密鍵137を用いて復号部108によって復号した後に認証鍵レジスタ119に格納するものとする。

【0130】

図52のフローチャートにおいて、まずステップS76で命令解釈部106によって認証鍵設定命令がフェッチされ、ステップS77でその命令に含まれる暗号化されたHMAC鍵がプロセッサ固有RSA秘密鍵137を用いて復号部108によって復号され、ステップS78で復号されたHMAC鍵が命令解釈部106によって認証鍵レジスタ119に格納されて処理を終了する。

【0131】

図53は既にセキュアコンテキスト識別子が生成されているコンテキストの実行コードを主記憶、すなわち物理メモリ101の物理ページに格納し、その物理ページの認証を行って処理の実行開始を可能とするためのページイン方式の説明図であり、図54はこのページインにおける処理のフローチャートである。

【0132】

このページインの処理では、まずOSによって実行コードの物理ページへの格納やページ・テーブル・エントリ(PTE)内の各種データの設定が行われた後に、OSから認証部107に対してコンテキスト、すなわち物理ページの認証要求としてのセキュアページフラグフィールドのセット要求がなされ、その要求に対応して認証部107によって物理ページの認証が行われた後にセキュアページフラグフィールドのフラグがセットされ、以後PTE使用が可能となる。

【0133】

図54の処理フローチャートにおいて処理が開始されると、ステップS80でOSによって空き物理ページに実行コードが格納され、ステップS81でOSによってその物理ページの先頭アドレスと対応する論理ページの先頭アドレスとが物理アドレス、論理アドレスとしてTLBの内部のPTEにセットされ、ステップS82でセキュアコンテキスト識別子の値がそのPTEにセットされる。例えば図45、46で説明したようにコンテキスト生成命令の発行時点で生成され、セキュアコンテキスト識別子レジスタ118に格納されたセキュアコンテキスト識別子がOSによって読み出し可能となっているものとして、OSは読み出したセキュアコンテキスト識別子をPTEに設定する。

【0134】

その後ステップS83でOSによって、必要に応じてそのページに対するリード/ライト属性などがPTEにセットされ、ステップS84でOSから、例えばハードウェアとしての認証部107に対してセキュアページフラグフィールドのセットが要求される。なお、OS自体は既に認証済みであることは大前提であり、セキュアページフラグフィールドのセットは基本的にOSの仕事であるが、ここでは認証済みのOSからハードウェアに対してフラグのセット要求が行われる。

【0135】

ステップS85で認証部107による認証処理が実行される。この処理の詳細については後述する。この処理ではセキュアコンテキスト識別子に対応するコンテキストの認証鍵と、認証情報一次格納部112に格納された認証情報とを用いて物理ページの認証が行われ、ステップS86で認証が成功したか否かが判定され、成功した場合にはセキュアページフラグフィールドのフラグがセットされ、以後そのPTEが使用可能となる。これに対して失敗した場合にはそのフィールドのフラグはリセットされ、そのPTEは使用不可と

10

20

30

40

50

され、ステップ S 8 9 で OS によるリカバリまたはエラー処理が行われる。

【 0 1 3 6 】

なお例えば図 5 4 の処理は T L B 内の P T E に対して直接に値の設定可能なプロセッサを対象としているが、例えば主記憶上の P T E に値が設定され、T L B はそのキャッシュとして働くようなプロセッサにおいては、ステップ S 8 0 からステップ S 8 3 までの処理が主記憶上の P T E に対して行われ、その内容を T L B にキャッシュするタイミングでステップ S 8 4 以降の動作が行われることになる。

【 0 1 3 7 】

図 5 5 は図 5 3 の認証部 1 0 7 の構成例であり、図 5 6 は図 5 4 におけるステップ S 8 5 の認証処理のフローチャートである。ここではページ全体から S H A - 1 ハッシュ値が計算され、電子署名の復号結果と比較されるものとする。なお図 5 6 に示すように認証部 1 0 7 の動作をハードウェアによってではなく、ソフトウェアによる処理として実現することも当然可能である。

10

【 0 1 3 8 】

図 5 5 において物理ページ 1 2 5 は 6 4 バイトずつに分割され、S H A - 1 ハッシュ演算器 1 4 0 に与えられ、ページ全体のハッシュ値が計算されて比較器 1 4 2 に与えられる。一方、認証情報一次格納部 1 1 2 に格納された R S A 電子署名は、認証鍵レジスタ 1 1 9 に格納された R S A 公開鍵とともに R S A 復号器 1 4 1 に与えられ、その出力としての復号済みハッシュ値が比較器 1 4 2 によって S H A - 1 ハッシュ演算器 1 4 0 の出力と比較されて一致する場合には認証成功、不一致の場合には失敗と判定される。

20

【 0 1 3 9 】

図 5 6 の認証処理において、まずステップ S 9 0 で物理ページが 6 4 バイト単位で読み込まれ、ステップ S 9 1 でハッシュ演算が行われ、ステップ S 9 2 でページ終端に達したか否かが判定され、達していない場合にはステップ S 9 0 以降の処理が繰り返される。

【 0 1 4 0 】

終端に達した場合にはステップ S 9 3 で R S A 公開鍵を用いて電子署名が復号処理され、ステップ S 9 4 で復号結果とハッシュ演算の結果が比較され、一致している場合にはステップ S 9 5 でセキュアページフラグフィールドがセットされ、不一致の場合にはステップ S 9 6 でセキュアページフラグフィールドがリセットされて処理を終了する。

【 0 1 4 1 】

以上のように第 5 の実施例では、実行コードの物理メモリ（主記憶）へのページインにあたって実行コードの認証が行われ、認証が成功したことを示すセキュアページフラグのセットが行われる。

30

【 0 1 4 2 】

次に本発明における物理ページ上の命令実行時のメモリアクセス制御について第 6 の実施例として説明する。図 5 7 は、物理ページ上の命令実行時のメモリアクセス制御方式の説明図である。同図においてセキュアコンテキスト識別子レジスタ 1 1 8 に意味のある値が入っており、また P T E 1 2 2 のセキュアページフラグフィールドがセットされており、さらにセキュアコンテキスト識別子レジスタ 1 1 8 内に格納されている識別子の値と、P T E 1 2 2 上のコンテキスト識別子の値が一致する場合に物理ページ 1 2 4 上の命令の実行が許可される。この制御はメモリアクセス制御部 1 0 5 によって行われる。なお物理ページに対するデータのリード/ライト属性やスーパーバイザ属性などのチェックは本発明の内容と直接の関係はなく、別途行われているものとする。

40

【 0 1 4 3 】

図 5 8 は、メモリアクセス制御部 1 0 5 の動作例の説明図である。同図において太い一点鎖線の中味がメモリアクセス制御部 1 0 5 に相当し、また T L B 1 1 4 の内部の P T E の属性データとしてセキュアページフラグフィールドやセキュアコンテキスト識別子が含まれるものとする。

【 0 1 4 4 】

図 3 9 と同様に、例えば論理アドレスをアクセスアドレスとしたアクセスが行われると

50

、そのアドレスによって選択された P T E の属性データが読み出され、アクセス属性と属性チェック 1 4 6 によって比較され、チェック結果が O K であれば論理アドレスに対応して読み出された物理アドレスと属性チェック結果を用いてキャッシュコマンド生成 1 4 7 が行われ、例えば図 2 0 のキャッシュメモリ 4 5 の内部のタグ 1 4 8 が検索されて、キャッシュヒット時の場合にはそのままデータ応答が返され、キャッシュミス時の場合にはキューおよびバスインタフェース 1 4 9 を介して、例えば主記憶からロードされたデータがキャッシュメモリに格納されるとともにデータ応答が実行ユニットに返される。

【 0 1 4 5 】

図 5 9 は、命令フェッチ時のメモリアクセス制御部 1 0 5 の処理フローチャートである。図 5 7 の命令実行部 1 4 4 によって命令フェッチのための論理アドレスが出力されると、ステップ S 9 8 で指定された論理アドレスに対応する P T E の属性データが選択され、ステップ S 9 9 でカレントコンテキスト、すなわち現在実行すべきコンテキストがセキュアコンテキストであるか否か、すなわち有効なセキュアコンテキスト識別子を持っているかどうかチェックされる。セキュアコンテキストである場合には、ステップ S 1 0 0 でそのコンテキストに対応する P T E のセキュアページフラグフィールド (S P F) がセットされているか否かがチェックされ、セットされている場合にはステップ S 1 0 1 でカレントコンテキストのセキュアコンテキスト識別子、すなわちセキュアコンテキスト識別子レジスタ 1 1 8 に格納されている識別子と P T E に格納されているセキュアコンテキスト識別子が一致しているか否かが判定される。

10

【 0 1 4 6 】

一致している場合にはステップ S 1 0 2 で、例えばコンテキストに対応するページの属性としてのリード/ライト属性やスーパーバイザ属性などのチェックが行われ、O K の場合にはステップ S 1 0 3 で命令フェッチのための物理アドレスをキャッシュに出力するためのキャッシュコマンドが生成されて処理を終了する。

20

【 0 1 4 7 】

ステップ S 9 9 でカレントコンテキストが有効なセキュアコンテキスト識別子を持っていない場合には、対応する P T E の中のセキュアページフラグフィールド (S P F) がセットされているか否かがステップ S 1 0 4 で判定され、セットされていない場合にはセキュアコンテキスト識別子が設定されておらず、また認証の行われていない従来と同様の実行コードを処理すべきことになり、ステップ S 1 0 2 の処理に移行する。ステップ S 1 0 0 、 S 1 0 1 における判定結果が N o である場合、またステップ S 1 0 4 の判定結果が Y e s である場合、ステップ S 1 0 2 の判定結果が N o である場合にはいずれもステップ S 1 0 5 でエラー処理が行われて処理を終了する。なおここでは論理アドレスを論理ページの先頭アドレスとページ内のオフセットの値とに分解する処理と、物理ページの先頭アドレスとそのオフセットの値を加算する処理が必要であるが、これらの処理については本発明と直接の関係はなくその説明を省略する。

30

【 0 1 4 8 】

図 6 0 は、セキュアコアとノーマルコアとが備えられたプロセッサにおけるメモリアクセス制御方式の説明図である。同図においてノーマルコア 1 5 2 は、第 1 の実施例における図 1 2 と同様に暗号処理ブロック 1 2 とコード認証処理ブロック 1 3 による処理とは無関係の従来と同様の処理だけを行うものであり、セキュアコア 1 5 1 は図 4 4 においては説明しない暗号処理ブロックによる動作の制御を含めて、コード認証処理ブロックの認証制御を含むセキュア動作を実行可能なものである。

40

【 0 1 4 9 】

図 6 0 においてメモリアクセス制御部 1 0 5 の制御によって、セキュアコア 1 5 1 からセキュアページフラグフィールドのセットされた P T E に対応する物理ページの使用は許可されるが、ノーマルコアからはそのページが利用できないようにする制御が行われる。

【 0 1 5 0 】

なお図 4 4 において、セキュアコアから制御されるコード認証処理ブロックはメモリア

50

クセス制御部 105 を含み、認証部 107、暗号化 / 復号、署名生成 / 検証部 108、セキュアコンテキスト識別子生成部 109、セキュアコンテキスト識別子消滅部 110、プロセッサ固有鍵 111、認証情報一次格納部 112、セキュアDMA 113、セキュアコンテキスト識別子レジスタ 118、認証鍵レジスタ 119、および PTE 122 の内部のセキュアページフラグフィールドとセキュアコンテキスト識別子に相当する。

【0151】

なお図 60 においてセキュアコア 151 は、認証処理が終了し、セキュアページフラグフィールドが PTE にセットされた物理ページ内の実行コードのみを実行し、ノーマルコア 152 は認証されていない通常コードのみを実行することを基本とするが、ノーマルコアが通常コードに加えて認証されたコードを実行可能とするよう構成することも可能である。

10

【0152】

図 61 は、セキュアモードとノーマルモードとを切り替えるコアを備えるプロセッサの構成ブロック図である。同図においてはコア 154 の内部にモードレジスタ 155 が備えられ、セキュアモードの場合だけセキュアページフラグフィールドが設定されたページを利用可能とするものである。なおセキュアモードとノーマルモードの切り替えは、例えば通常のユーザモードとカーネルモードとの切り替えのように割込みをトリガとする方法でもよく、他の方法を用いても良い。

【0153】

図 62 はメモリアクセス制御方式としての、図 44 におけるセキュアDMA 113 による物理メモリ 101 へのページデータ転送方式の説明図であり、図 63 はセキュアDMA 113 によるデータ転送処理のフローチャートである。例えば図 53 や図 54 のページイン処理では、物理ページに実行コードとしてのページデータが格納された後にその実行コードの認証を行うものとしたが、認証処理においてはハッシュ値の計算などの処理を必要とするため、ここではページデータの転送単位毎にハッシュ値を計算し、その結果をハッシュ演算の中間結果として保持する動作を繰返し、転送終了時点ではハッシュ演算を終了して、その結果をその後の認証処理に利用するものである。

20

【0154】

図 62 においてセキュアDMA 113 は、コア 154 からのデータの転送元アドレス、転送先アドレス、転送サイズを受け取る転送管理部 157、I/O 装置 102 からデータを読み出すデータ読み出し器 158、ハッシュ演算を行うハッシュ演算器 159、物理メモリ 101 にページデータを書き込むデータ書き込み器 160、物理ページの先頭アドレスとそのページに対するハッシュ値を保持する物理ページ先頭アドレス保持部 161 を備えている。

30

【0155】

図 63 において処理が開始されると、コア 154 上で動作するプログラム、一般には OS から転送元アドレスなどの指示を受けた転送管理部 157 によって、データ読み出し器 158 に対して I/O 装置 102 から次の 64 バイトのデータの読み出しが指示され、ステップ S111 でデータ読み出し器 158 によって 64 バイトのデータが読み出され、ステップ S112 で転送管理部 157 によってハッシュ演算器 159 に対してハッシュ演算が指示され、ステップ S113 でハッシュ演算器 159 によってハッシュ演算が行われ、その中間結果が内部に保持され、ステップ S114 で転送管理部 157 からデータ書き込み器 160 に対して物理メモリ 101 への 64 バイトのデータの書き込みが指示され、ステップ S115 でデータ書き込み器 160 によって 64 バイトのデータが物理メモリ 101 に書き込まれ、ステップ S116 で 1 ページのデータ転送が終了したか否かが判定され、終了していない場合にはステップ S110 からの処理が繰り返され、終了している場合には転送管理部 157 によって転送先アドレスとしての物理ページ先頭アドレスとハッシュ値のペアが保持部 161 に与えられて処理を終了する。

40

【0156】

図 64 はメモリアクセスの制御を含むコード実行時の処理フローチャートである。同図

50

は代表的にはOSによるページイン時の処理のフローチャートであり、本発明の特徴は太線で囲まれた処理にある。処理が開始されるとまずステップS120でセキュアDMA113に対して転送元/先アドレス、転送サイズなどが指示され、ステップS121で転送が成功したか否かが判定され、成功した場合にはステップS122で図54のステップS81からS83におけると同様にTLB内のPTEに各種情報が設定され、ステップS123でステップS84と同様にセキュアページフラグフィールドのセットが要求され、認証部による認証処理が実行された後にステップS124でフラグのセットが成功したか否かが判定され、成功した場合には処理を終了する。またステップS121で転送が失敗した場合、ステップS124でセットが失敗した場合には直ちに処理を終了する。

【0157】

以上のように第6の実施例によれば、既に認証が成功した実行コードへのアクセスに対してもセキュアコンテキスト識別子やセキュアページフラグフィールドのチェックが行われた後にアクセスが許可される。

【0158】

最後に本発明の第7の実施例について図65から図74を用いて説明する。この第7の実施例では、例えばコンテキストスイッチに対応したコンテキスト情報やPTEの、例えば主記憶への退避に当って、データを保護するための暗号化、または改ざん検出情報の付加が行われる。例えば第1の実施例では認証された実行コードが暗号化されて物理メモリに格納されるものとしたが、第7の実施例では、例えばコンテキスト情報が暗号化されて物理メモリに格納される。

【0159】

図65は、そのコンテキスト情報暗号化方式の説明図である。同図においては図44で説明したコンテキスト情報格納部115に格納されているコンテキスト情報のすべてが暗号器165によってプロセッサ固有鍵111を用いて暗号化され、暗号化コンテキスト情報166として物理メモリ101に格納される。

【0160】

図66は、図65に対応するコンテキスト情報の復号方式の説明図である。物理メモリ101に格納されている暗号化コンテキスト情報166は、コンテキストスイッチによって必要となった時点でプロセッサ固有鍵111を用いて復号器168によって復号され、コンテキスト情報格納部115に格納される。

【0161】

図67は、コンテキスト情報への改ざん検出情報の付加方式の説明図である。同図においてコンテキスト情報格納部115に格納されているコンテキスト情報に対して、プロセッサ固有鍵111を用いて改ざん検出情報170が改ざん検出情報生成器169によって生成され、物理メモリ101にコンテキスト情報とともに格納される。

【0162】

図68は、図67に対応する改ざん検出情報を用いたコンテキスト情報に対する改ざん検出方式の説明図である。同図においてコンテキスト情報に付加された改ざん検出情報170を用いて、改ざん検出器172によってプロセッサ固有鍵111を用いた改ざん検出が行われる。

【0163】

図69は、コンテキスト情報格納部115に格納されたコンテキスト情報のうちでセキュアな動作に必要となるコンテキスト情報と、通常のコンテキスト情報とを区分し、セキュア動作用コンテキスト情報175だけを暗号化するコンテキスト情報暗号化方式の説明図である。この方式では、プロセッサの核となる部分の変更は極力行わないように通常のコンテキスト情報176、すなわち既存コンテキストIDなどのコンテキスト情報は従来と同様に暗号化せずに扱うこととし、認証鍵レジスタ119、セキュアコンテキスト識別子レジスタ118の格納内容をセキュア動作用コンテキスト情報175として暗号化するものである。

【0164】

10

20

30

40

50

既存コンテキストIDとしては、例えばOSの動作としてセキュアコンテキスト識別子と同じ値を格納することも可能であるものとする。例えばOSが悪意をもったコードに書き換えられたような場合には2つの識別子の値が同じとなる保証がなくなるが、同じ値であるときのみプロセッサを動作可能とさせるように構成することで、同じ値でない場合には動作しないという安全サイドに倒れ、問題は発生しない。

【0165】

図69においてセキュア動作コンテキスト情報175だけがプロセッサ固有鍵111を用いて暗号器/復号器174によって暗号化され、暗号化コンテキスト情報177として物理メモリ101に格納され、通常のコンテキスト情報176は平文コンテキスト情報176としてそのまま物理メモリ101に格納される。

10

【0166】

図70は、セキュア動作コンテキスト情報175に対して改ざん検出情報を付加して物理メモリ101に格納するコンテキスト情報格納方式の説明図である。同図においては改ざん検出情報生成器/改ざん検出器179によって、プロセッサ固有鍵111を用いてセキュア動作コンテキスト情報175に対する改ざん検出情報180が生成され、物理メモリ101にセキュア動作コンテキスト情報175と通常のコンテキスト情報、すなわち平文コンテキスト情報176とともに格納される。なおここでは通常のコンテキスト情報としてのプログラムカウンタの値やレジスタ群の値に対しては暗号化などを行わないものとしたが、さらに信頼性を向上させるためにはこのような通常コンテキスト情報についても暗号化、あるいは改ざん検出情報の付加を行うことも当然可能である。

20

【0167】

図71から図74は、ページ・テーブル・エントリ(PTE)122の格納内容の保護方式の説明図である。図71はPTEの暗号化方式を示し、PTE122の格納内容、すなわちセキュアページフラグフィールド、セキュアコンテキスト識別子、論理アドレス、および物理アドレスの値がプロセッサ固有鍵111を用いて暗号器165によって暗号化され、暗号化PTE183として物理メモリ内のページテーブル182に格納される。

【0168】

図72は、図71に対応する暗号化PTEの復号方式の説明図である。同図において物理メモリ101に格納されている暗号化PTE183は、プロセッサ固有鍵111を用いて復号器168によって復号され、TLB114の内部にPTEとして格納される。

30

【0169】

図73はPTEへの改ざん検出情報付加方式、図74はPTEに対する改ざん検出方式の説明図である。図73においては改ざん検出情報生成器169によってプロセッサ固有鍵111を用いてPTE122に対する改ざん検出情報185が生成され、PTE122とともにページテーブル182に格納される。

【0170】

図74においては改ざん検出情報185とプロセッサ固有鍵111とを用いて、改ざん検出器172によってページテーブル182に格納されているPTE122に対する改ざん検出が行われる。

【0171】

以上のように第7の実施例では、セキュアプロセッサによって使用されるコンテキスト情報とPTEに対しても暗号化や改ざん検出の処理が行われ、情報処理の安全性がさらに向上する。

40

【0172】

以上において本発明のセキュアプロセッサ、およびセキュアプロセッサ用プログラムについてその詳細を説明したが、このセキュアプロセッサを一般的なコンピュータシステムの基本要素とすることが可能である。図75はそのようなコンピュータシステム、すなわちハードウェア環境の構成ブロック図である。

【0173】

図75においてコンピュータシステムは中央処理装置(CPU)200、リードオンリ

50

メモリ (ROM) 201、ランダムアクセスメモリ (RAM) 202、通信インタフェース 203、記憶装置 204、入出力装置 205、可搬型記憶媒体の読取り装置 206、およびこれらの全てが接続されたバス 207によって構成されている。

【0174】

記憶装置 204としてはハードディスク、磁気ディスクなど様々な形式の記憶装置を使用することができ、このような記憶装置 204、またはROM 201に図3～図5、図7、図9～図11、その他のフローチャートに示されたプログラムや、本発明の特許請求の範囲の請求項7、19、および20のプログラムなどが格納され、そのようなプログラムがCPU 200によって実行されることにより、本実施形態におけるセキュアプロセッサの動作、暗号鍵の設定、コード認識処理、および暗号処理などが可能となる。

10

【0175】

このようなプログラムは、プログラム提供者 208からネットワーク 209、および通信インタフェース 203を介して、例えば記憶装置 204に格納されることも、また市販され、流通している可搬型記憶媒体 210に格納され、読取り装置 206にセットされて、CPU 200によって実行されることも可能である。可搬型記憶媒体 210としてはCD-ROM、フレキシブルディスク、光ディスク、光磁気ディスク、DVDなど様々な形式の記憶媒体を使用することができ、このような記憶媒体に格納されたプログラムが読取り装置 206によって読取られることにより、本実施形態におけるセキュアプロセッサの動作が可能となる。

【0176】

20

(付記1)

命令コードを実行するコアを備えるプロセッサであって、
該コアに固有の鍵を記憶する鍵記憶手段と、
暗号化された命令コードを書き換え不可能な形式で記憶する命令コード記憶手段と、
該命令コード記憶手段に記憶された命令コードを含む命令コードの認証を前記固有鍵あるいは固有鍵による認証済鍵を用いて行う認証処理手段と、
該コアと外部との間で入出力されるデータを暗号化する暗号処理手段とを備えることを特徴とするセキュアプロセッサ。

【0177】

(付記2)

30

前記暗号処理手段が、前記認証処理手段によって認証された命令コードを暗号化し、ページ単位で前記セキュアプロセッサに接続された記憶装置に格納することを特徴とする付記1記載のセキュアプロセッサ。

【0178】

(付記3)

前記記憶装置に格納されたページ単位の暗号化命令コードの実行時に不正命令が検出された時、該ページ単位の暗号化命令コードの実行を停止する不正命令実行停止手段を備えることを特徴とする付記2記載のセキュアプロセッサ。

【0179】

(付記4)

40

前記認証処理手段が認証対象とする命令コードに認証情報が付加されていることを特徴とする付記1記載のセキュアプロセッサ。

【0180】

(付記5)

前記認証情報内に暗号化の鍵が指定されている時、前記暗号処理手段が該指定されている鍵を使用してさらに前記命令コードの暗号化を行うことを特徴とする付記4記載のセキュアプロセッサ。

【0181】

(付記6)

前記認証情報内に暗号化の鍵が指定されていない時、前記暗号処理手段が任意のページ

50

鍵を使用してさらに前記命令コードの暗号化を行うことを特徴とする付記 4 記載のセキュアプロセッサ。

【0182】

(付記 7)

前記暗号処理手段が、前記認証された命令コードに対応する同一プロセスのデータに対して、該命令コードに対する暗号鍵と異なる暗号鍵を用いて該データの暗号化を行うことを特徴とする付記 1 記載のセキュアプロセッサ。

【0183】

(付記 8)

前記セキュアプロセッサに接続された記憶装置内でデータの格納領域に格納された命令コードの実行時に前記異なる暗号鍵に代わって命令コードに対する暗号鍵を使用することを特徴とする付記 7 記載のセキュアプロセッサ。 10

【0184】

(付記 9)

前記セキュアプロセッサにおいて、

前記認証処理手段による認証が失敗した命令コードの実行を停止させるコード実行停止処理手段をさらに備えることを特徴とする付記 1 記載のセキュアプロセッサ。

【0185】

(付記 10)

前記コアとして、 20

前記認証処理手段によって認証された命令コードのみを実行するセキュアコアと、

前記認証処理手段によって認証されていない通常の命令コードも実行可能なノーマルコアとを備えることを特徴とする付記 1 記載のセキュアプロセッサ。

【0186】

(付記 11)

前記命令コード記憶手段に記憶された暗号化命令コードを用いて前記セキュアコアがブートされるとともに、

該セキュアコアが該ブート完了後に前記ノーマルコアのブートを行わせるノーマルコアブート手段を備えることを特徴とする付記 10 記載のセキュアプロセッサ。

【0187】 30

(付記 12)

前記セキュアコアが、前記ノーマルコアのブート後に該ノーマルコアの動作を監視し、異常状態を検出した時、該ノーマルコアの動作停止、または特定処理への分岐を行わせるノーマルコア監視手段を備えることを特徴とする付記 11 記載のセキュアプロセッサ。

【0188】

(付記 13)

前記セキュアコアが、前記ノーマルコアに対してコア制御信号を与え、ノーマルコアの動作を制御することを特徴とする付記 10 記載のセキュアプロセッサ。

【0189】

(付記 14) 40

前記コア固有鍵に対するアクセスが、前記セキュアコアに対して許可され、前記ノーマルコアに対して禁止されることを特徴とする付記 10 記載のセキュアプロセッサ。

【0190】

(付記 15)

前記セキュアコアの制御のもとで、前記コア固有鍵を用いて公開鍵と秘密鍵のペア、および共通鍵を生成する鍵生成手段をさらに備えることを特徴とする付記 14 記載のセキュアプロセッサ。

【0191】

(付記 16)

前記セキュアコアが、前記ノーマルコアを経由して前記鍵生成手段によって生成された 50

公開鍵を外部に通知し、

外部から該公開鍵によって暗号化された原文をノーマルコアを經由して受け取り、前記秘密鍵を用いて原文を復号することを特徴とする付記 15 記載のセキュアプロセッサ。

【0192】

(付記 17)

前記原文が情報の暗号化に使用された鍵であることを特徴とする付記 16 記載のセキュアプロセッサ。

【0193】

(付記 18)

プロセッサにおいて命令コードを実行するコアによって使用されるプログラムであって 10

、暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて自コアの起動処理を行う手順と、

該メモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックと、前記コア固有の鍵を管理する鍵管理処理と、該認証処理ブロックによって認証された命令コードの暗号化/復号処理に用いられる鍵が格納された鍵テーブルに対する操作処理とをセットアップする手順と、

前記認証処理ブロックを用いて二次記憶上のプログラムの認証処理を行う手順と、

起動されたオペレーティングシステムを含む該認証処理済みのプログラムの実行時に前記命令コードの暗号化/復号のための鍵処理を含む処理を実行する鍵処理モニタとしての動作を行う手順とを計算機に実行させるためのセキュアコア用プログラム。 20

【0194】

(付記 19)

プロセッサにおいて命令コードを実行するコアによって使用されるプログラムであって、

暗号化された命令コードが書き換え不可能な形式で記憶されたメモリ内のプログラムを用いて、自コアの起動処理を行う手順と、

オペレーティングシステムを起動する手順と、

該プロセッサ内で前記メモリ内に記憶された命令コードを含む命令コードの認証処理を行う認証処理ブロックによって認証されたプログラム、または認証されていないプログラムを実行し、該認証されたプログラムの実行処理としては、該認証された実行コードに対応して、暗号化/復号用の鍵を用いた処理を含む鍵処理を実行する鍵処理モニタに対する鍵処理の依頼を含みうる処理を実行する手順とを計算機に実行させることを特徴とするノーマルコア用プログラム。 30

【0195】

(付記 20)

命令を実行する命令実行手段と、

該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード/ストアを制御するロード/ストア制御手段と、

該ロード/ストア制御手段と外部のメモリとの間でデータの暗号化/復号化を行う暗号処理手段とを備え、 40

前記命令実行手段が、実行中の命令に対応して該暗号処理手段に対してデータ暗号化/復号化に使用すべき鍵を指定することを特徴とするセキュアプロセッサ。

【0196】

(付記 21)

前記プロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、

前記命令実行手段が、該鍵記憶手段に対して前記鍵を指定する鍵番号を出力し、該鍵記憶手段が該鍵番号に対応して前記暗号処理手段に対して、データ暗号化/復号化に使用すべき鍵を与えることを特徴とする付記 20 記載のセキュアプロセッサ。

【0197】

(付記 2 2)

前記セキュアプロセッサにおいて、外部からロードされた命令フェッチデータの復号化に使用されるべき鍵を記憶する鍵記憶手段をさらに備え、

前記命令実行手段が命令フェッチ状態にある時、該鍵記憶手段が前記暗号処理手段に対して該復号化用の鍵を与えることを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 1 9 8 】

(付記 2 3)

前記セキュアプロセッサにおいて、
複数の鍵を記憶する鍵記憶手段と、

前記命令実行手段によって出力され、前記鍵を指定するための鍵番号を記憶する鍵番号記憶手段とを備え、

該鍵記憶手段が、該鍵番号記憶手段から与えられる鍵番号に対応して前記データ暗号化 / 復号化に使用すべき鍵を前記暗号処理手段に対して与えることを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 1 9 9 】

(付記 2 4)

前記セキュアプロセッサにおいて、

外部からロードされる命令フェッチデータの復号化に使用されるべき鍵を含む複数の鍵を記憶する鍵記憶手段と、

外部からロードされた命令フェッチデータの復号化に使用されるべき鍵の鍵番号を記憶する鍵番号記憶手段とをさらに備え、

前記命令実行手段が命令フェッチ状態にある時、該鍵番号記憶手段から出力される鍵番号に対応して該鍵記憶手段が、命令フェッチデータの復号化に使用されるべき鍵を前記暗号処理手段に与えることを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 2 0 0 】

(付記 2 5)

前記命令実行手段が、前記鍵を指定するための信号として鍵の番号に加えて命令に対応するスーパーバイザ / ユーザ切り替え信号を出力することを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 2 0 1 】

(付記 2 6)

前記命令実行手段が、前記鍵を指定するための信号として鍵の番号に加えて実行中の命令が含まれるプロセスの識別子を出力することを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 2 0 2 】

(付記 2 7)

前記ロード / ストア制御手段が、
ライトスルー方式のキャッシュメモリと、

外部のメモリにストアすべきデータと該外部メモリから前記暗号処理手段を介してロードされたデータとを結合して暗号処理手段に与えるリードモディファイライト手段とをさらに備えることを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 2 0 3 】

(付記 2 8)

前記セキュアプロセッサにおいて、

前記ロード / ストア制御手段と外部メモリとの間で、前記暗号処理手段をバイパスして、暗号化 / 復号化を行うことなく、平文データの転送を行うデータバイパス手段をさらに備えることを特徴とする付記 2 0 記載のセキュアプロセッサ。

【 0 2 0 4 】

(付記 2 9)

命令を実行する命令実行手段と、

10

20

30

40

50

該命令実行手段からのコマンドに対応して外部のメモリに対するデータのロード/ストアを制御するロード/ストア制御手段と、

該ロード/ストア制御手段と外部のメモリとの間でデータの暗号化/復号化を行う暗号処理手段とを備え、

前記命令実行手段が、実行中の命令によるデータ/命令フェッチのアクセスアドレスに対応させて、該暗号処理手段に対してデータ暗号化/復号化に使用すべき鍵を指定する信号を与えることを特徴とするセキュアプロセッサ。

【0205】

(付記30)

前記プロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、

前記命令実行手段が、前記アクセスアドレスとしての論理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該論理アドレスに対応して前記データ暗号化/復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記29記載のセキュアプロセッサ。

【0206】

(付記31)

前記セキュアプロセッサにおいて複数個の鍵を記憶する鍵記憶手段をさらに備え、

前記ロード/ストア制御手段が、前記命令実行手段から与えられるコマンドに対応して前記アクセスアドレスとしての物理アドレスを該鍵記憶手段に対して出力し、該鍵記憶手段が該物理アドレスに対応して前記データ暗号化/復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記29記載のセキュアプロセッサ。

【0207】

(付記32)

前記セキュアプロセッサにおいて、前記アクセスアドレスとしての論理アドレスと物理アドレスとのそれぞれに対応させてそれぞれ複数の鍵を記憶する鍵記憶手段をさらに備え、

該鍵記憶手段に対して前記ロード/ストア制御手段から与えられる前記アクセスアドレスとしての物理アドレスと、前記命令実行手段から与えられる論理アドレスとのいずれを選択すべきかを示す該命令実行手段からの指示に対応して、該鍵記憶手段が選択したアドレスに対応する前記データ暗号化/復号化用の鍵を前記暗号処理手段に与えることを特徴とする付記29記載のセキュアプロセッサ。

【0208】

(付記33)

前記セキュアプロセッサにおいて、

前記アクセスアドレスとしての論理アドレスと物理アドレスとのそれぞれに対応させてそれぞれ複数の鍵を記憶する鍵記憶手段と、

前記命令実行手段によって出力され、前記暗号処理手段に対して論理アドレスと物理アドレスのいずれに対応する鍵を与えるべきかを示すアドレス選択指示のデータを記憶するアドレス選択指示記憶手段とをさらに備え、

該鍵記憶手段が、該アドレス選択指示記憶手段の記憶内容に従って論理アドレスと物理アドレスとのいずれかに対応する鍵を前記データ暗号化/復号化用の鍵として前記暗号処理手段に与えることを特徴とする付記29記載のセキュアプロセッサ。

【0209】

(付記34)

前記ロード/ストア制御手段が、前記アクセスアドレスに対応して複数の鍵を記憶する鍵記憶手段をさらに備え、

該ロード/ストア制御手段が、前記命令実行手段から命令実行中に与えられたアクセスアドレスに対応して該鍵記憶手段に記憶された鍵を選択し、前記データ暗号化/復号化用の鍵として前記暗号処理手段に与えることを特徴とする付記29記載のセキュアプロセッサ。

【0210】

10

20

30

40

50

(付記 35)

前記命令実行手段が前記暗号処理手段に対して、前記鍵記憶手段の ON / OFF を示す信号と、該鍵記憶手段が OFF の時に前記データ暗号化 / 復号化に使用されるべき鍵を与える信号とを出力し、

該暗号処理手段が前記 ON / OFF 信号に対応して、前記鍵記憶手段が ON の時には鍵記憶手段から与えられる鍵を、OFF の時には該命令実行手段から与えられる鍵を前記データ暗号化 / 復号化用の鍵として使用することを特徴とする付記 34 記載のセキュアプロセッサ。

【0211】

(付記 36)

前記命令実行手段が前記鍵を指定するための信号として、前記アクセスアドレスに加えて、実行中の命令に対応するスーパーバイザ / ユーザ切り替え信号を出力することを特徴とする付記 29 記載のセキュアプロセッサ。

【0212】

(付記 37)

前記命令実行手段が、前記鍵を指定するための信号として前記アクセスアドレスに加えて、実行中の命令が含まれるプロセスの識別子を出力することを特徴とする付記 29 記載のセキュアプロセッサ。

【0213】

(付記 38)

前記ロード / ストア制御手段が、
ライトスルー方式のキャッシュメモリと、
外部のメモリにストアすべきデータと該外部メモリから前記暗号処理手段を介してロードされたデータを結合して該暗号処理手段に与えるリードモディファイライト手段とをさらに備えることを特徴とする付記 29 記載のセキュアプロセッサ。

【0214】

(付記 39)

前記セキュアプロセッサにおいて、前記ロード / ストア制御手段と外部メモリとの間で前記暗号処理手段をバイパスして、暗号化 / 復号化を行うことなく、平文データの転送を行うデータバイパス手段をさらに備えることを特徴とする付記 29 記載のセキュアプロセッサ。

【0215】

(付記 40)

実行コードに対応するプロセスの実行に先立って、該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するためのセキュアプロセス識別子を、該プロセスの生成命令が発行された時点で生成するセキュアプロセス識別子生成手段と、

該生成されたセキュアプロセス識別子を該プロセスに関連する情報として保持するプロセス情報保持手段とを備えることを特徴とするセキュアプロセッサ。

【0216】

(付記 41)

前記生成され、前記プロセス情報保持手段に保持されているセキュアプロセス識別子を、前記プロセスの消滅時に消去するセキュアプロセス識別子消去手段をさらに備えることを特徴とする付記 40 記載のセキュアプロセッサ。

【0217】

(付記 42)

前記プロセスに対応する実行コードに認証情報が付与されるとともに、
前記プロセス情報保持手段が、前記生成されたプロセスの生存期間中に行われる実行コード認証のための認証鍵をさらに保持することを特徴とする付記 40 記載のセキュアプロセッサ。

10

20

30

40

50

【0218】

(付記43)

前記実行コードに付与された認証情報がメモリにおけるページ単位の情報であることを特徴とする付記42記載のセキュアプロセッサ。

【0219】

(付記44)

前記プロセスに対応する実行コードがメモリの空きページに格納され、該ページのアドレスに対応させて前記セキュアプロセス識別子が前記プロセッサ内のバッファに格納された後に前記ページ単位の認証鍵を用いた該実行コードの認証が成功した時、該バッファに前記セキュアページフラグをセットする認証手段をさらに備えることを特徴とする付記43記載のセキュアプロセッサ。

10

【0220】

(付記45)

前記セキュアプロセッサにおいて、

前記実行コードの実際の実行に先立って前記バッファ内に格納されたセキュアプロセス識別子であって、対応する前記セキュアページフラグがセットされているセキュアプロセス識別子と、前記プロセス情報保持手段に保持され、実行すべき命令コードに対応するセキュアプロセス識別子とを比較し、両者が一致した時に前記実行コードが格納されたメモリ上のページへのアクセスを、命令を実行する命令実行部に許可するメモリアクセス制御手段をさらに備えることを特徴とする付記44記載のセキュアプロセッサ。

20

【0221】

(付記46)

前記セキュアプロセッサにおいて、

前記実行コードのメモリへの格納に並行して前記実行コードの認証に必要な演算を行い、該演算の結果を保持して前記認証手段に与える直接メモリアクセス手段をさらに備えることを特徴とする付記44記載のセキュアプロセッサ。

【0222】

(付記47)

前記セキュアプロセッサにおいて、該プロセッサに固有の暗号化/復号化用の鍵と、

前記バッファ内に格納されたセキュアページフラグ、セキュアプロセス識別子、および実行コードが格納されたメモリページのアドレスの情報を外部に退避、または外部から復帰するに当り、該プロセッサ固有鍵を用いて該情報の暗号化/復号化を行う暗号処理手段とをさらに備えることを特徴とする付記44記載のセキュアプロセッサ。

30

【0223】

(付記48)

前記セキュアプロセッサにおいて、

該プロセッサに固有の鍵と、

前記バッファ内に格納されたセキュアページフラグ、セキュアプロセス識別子、および実行コードが格納されたメモリページのアドレスの情報を外部に退避するに当り、該プロセッサ固有鍵を用いて該情報に対する改ざん検出情報を生成して付与し、外部から復帰するに当り該固有鍵を用いて該情報に対する改ざん検出を行う改ざん検出手段とをさらに備えることを特徴とする付記44記載のセキュアプロセッサ。

40

【0224】

(付記49)

前記セキュアプロセッサにおいて、それぞれ命令実行ユニットとキャッシュとを備えるコアであって、

前記認証された実行コードのみを実行するセキュアコアと、

該認証されていない通常の実行コードを実行するノーマルコアとを備えることを特徴とする付記40記載のセキュアプロセッサ。

【0225】

50

(付記50)

前記ノーマルコアが前記通常コードに加えて前記認証されたコードをも実行することを特徴とする付記49記載のセキュアプロセッサ。

【0226】

(付記51)

前記セキュアプロセッサにおいて、

実行ユニットとキャッシュメモリとを備えるコアが、前記認証された実行コードのみを実行すべきセキュアモードと、認証されていない通常の実行コードのみを実行すべきノーマルモードとのいずれかの指示が設定されるモード指定手段をさらに備え、

該コアが該指示に対応してセキュアモード、またはノーマルモードのいずれかを実行することを特徴とする付記40記載のセキュアプロセッサ。 10

【0227】

(付記52)

前記セキュアプロセッサにおいて、

該プロセッサに固有の暗号化/復号化用の鍵と、

前記プロセス情報保持手段に保持され、前記セキュアプロセス識別子を含む情報を外部に退避、または外部から復帰するに当り、該プロセッサ固有鍵を用いて該情報の暗号化/復号を行う暗号処理手段とをさらに備えることを特徴とする付記40記載のセキュアプロセッサ。

【0228】

20

(付記53)

前記セキュアプロセッサにおいて、

該プロセッサに固有の鍵と、

前記プロセス情報保持手段に保持され、前記セキュアプロセス識別子を含む情報を外部に退避するに当り、該プロセッサ固有鍵を用いて該情報に対する改ざん検出情報を生成して付与し、外部から復帰するに当り該固有鍵を用いて該情報に対する改ざん検出を行う改ざん検出手段とをさらに備えることを特徴とする付記40記載のセキュアプロセッサ。

【0229】

(付記54)

実行コードを含むページをメモリにページインする計算機によって使用されるプログラムであって、 30

該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼する手順と、

該転送の成功後に、該計算機のトランслーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定する手順と、

前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求する手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。 40

【0230】

(付記55)

実行コードを含むページをメモリにページインする計算機によって使用される記憶媒体であって、

該計算機内の直接メモリアクセス機構に前記ページのメモリへの転送を依頼するステップと、

該転送の成功後に、該計算機のトランслーション・ルックアサイド・バッファ内のページ・テーブル・エントリに、該ページ内の実行コードに対応するプロセスの実行に先立 50

って該実行コードを格納するページが正しく認証されたことを示すセキュアページフラグが設定されたページに対応するセキュアプロセス識別子と比較するための識別子であって、該プロセスの生成命令が発行された時点で生成されたセキュアプロセス識別子を含み、該ページについてのデータを設定するステップと、

前記ページの認証と、該認証の成功を示すセキュアページフラグの該ページ・テーブル・エントリへのセットとをハードウェアに要求するステップとを計算機に実行させるセキュアプロセッサ用プログラムを格納した計算機読出し可能可搬型記憶媒体。

【0231】

(付記56)

実行コードを含むページの認証を行う計算機によって使用されるプログラムであって、メモリに読み込まれた該ページに対するハッシュ演算を行う手順と、
該ページに付与されている認証情報を復号する手順と、
該ハッシュ演算結果と該復号結果とを比較する手順と、
該比較の結果として一致が検出された時、該計算機のトランслーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットする手順とを計算機に実行させることを特徴とするセキュアプロセッサ用プログラム。

【0232】

(付記57)

実行コードを含むページの認証を行う計算機によって使用される記憶媒体であって、メモリに読み込まれた該ページに対するハッシュ演算を行うステップと、
該ページに付与されている認証情報を復号するステップと、
該ハッシュ演算結果と該復号結果とを比較するステップと、
該比較の結果として一致が検出された時、該計算機のトランслーション・ルックアサイド・バッファ内のページ・テーブル・エントリに該ページの認証が成功したことを示すセキュアページフラグをセットするステップとを計算機に実行させるセキュアプロセッサ用プログラムを格納した計算機読出し可能可搬型記憶媒体。

【図面の簡単な説明】

【0233】

【図1】本発明のセキュアプロセッサの原理構成ブロック図である。 30

【図2】第1の実施例におけるプロセッサの基本構成を示すブロック図である。

【図3】第1の実施例におけるプロセッサの基本処理フローチャートである。

【図4】コード認証処理ブロックと暗号処理ブロックによる処理のフローチャートである。

【図5】命令領域とデータ領域とによって異なる鍵が指定されている場合の暗号処理ブロックの処理フローチャートである。

【図6】公開鍵で暗号化された暗号鍵の格納方式の説明図である。

【図7】公開鍵で暗号化された暗号鍵の格納処理フローチャートである。

【図8】認証局の署名が付与された暗号鍵の格納方式の説明図である。

【図9】認証局の署名が付与された暗号鍵の格納処理フローチャートである。 40

【図10】不正命令検出時の処理フローチャートである。

【図11】データ領域に格納された命令に対する鍵付け替え処理のフローチャートである。

【図12】第2の実施例におけるプロセッサの基本構成を示すブロック図である。

【図13】第2の実施例におけるプロセッサの基本処理フローチャートである。

【図14】セキュアコアとノーマルコアを備えるプロセッサの基本構成を示すブロック図である。

【図15】図14のプロセッサにおける処理の基本フローチャートである。

【図16】図14のプロセッサにおけるセキュアコアによるノーマルコアの動作の停止制御方式の説明図である。

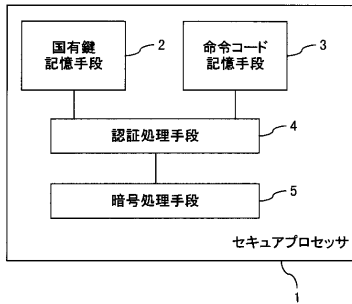
- 【図 17】図 14 のプロセッサにおけるセキュアコアによるノーマルコアの動作の停止制御処理のフローチャートである。
- 【図 18】セキュアコアに対応する鍵生成機構を備えるプロセッサの構成ブロック図である。
- 【図 19】図 18 のプロセッサにおける鍵処理方式の具体例の説明図である。
- 【図 20】第 3 の実施例におけるプロセッサの基本構成を示すブロック図である。
- 【図 21】第 3 の実施例において鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
- 【図 22】第 3 の実施例において命令アクセス状態にあるプロセッサの構成を示すブロック図である。 10
- 【図 23】鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図である。
- 【図 24】命令アクセス状態にあり、鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図である。
- 【図 25】鍵テーブルメモリの構成例を示す図である。
- 【図 26】暗号化回路、復号化回路の構成例を示すブロック図である。
- 【図 27】データ追い越し機能付き暗号化回路、復号化回路の構成例を示す図である。
- 【図 28】キャッシュスルー方式のロードストアユニットに対応するリードモディファイライト方式の説明図である。
- 【図 29】第 4 の実施例におけるプロセッサの基本構成を示すブロック図である。 20
- 【図 30】論理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
- 【図 31】物理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図である。
- 【図 32】第 4 の実施例における鍵テーブルメモリの構成例（その 1）を示す図である。
- 【図 33】第 4 の実施例における鍵テーブルメモリの構成例（その 2）を示す図である。
- 【図 34】第 4 の実施例における鍵テーブルメモリの構成例（その 3）を示す図である。
- 【図 35】論理アドレスと物理アドレスとが与えられる鍵テーブルメモリを備えるプロセッサの構成を示すブロック図である。
- 【図 36】図 35 の鍵テーブルメモリに対してアドレス選択指示を与える鍵選択レジスタ 30
を備えるプロセッサの構成ブロック図である。
- 【図 37】図 35、図 36 における鍵テーブルメモリの構成例を示す図である。
- 【図 38】メモリ管理ユニット内に鍵テーブルが備えられるプロセッサの構成を示すブロック図である。
- 【図 39】図 38 におけるデータアクセス方式の説明図である。
- 【図 40】アドレスマップレジスタに鍵テーブルが併設されている場合のデータアクセス方式の説明図である。
- 【図 41】メモリ管理ユニットの ON / OFF 状態に応じて鍵を切り替えるプロセッサの構成を示すブロック図である。
- 【図 42】メモリ管理ユニットの ON / OFF 状態に応じて鍵を切り替える暗号化 / 復号 40
方式の説明図である。
- 【図 43】第 3、および第 4 の実施例における実行ユニットの入出力信号の説明図である。
- 【図 44】第 5 の実施例におけるプロセッサシステムの詳細構成ブロック図である。
- 【図 45】セキュアコンテキスト識別子生成方式の説明図である。
- 【図 46】セキュアコンテキスト識別子生成方法の説明図である。
- 【図 47】セキュアコンテキスト識別子消滅方式の説明図である。
- 【図 48】実行コードに付加された認証情報の説明図である。
- 【図 49】公開鍵の認証鍵レジスタへの格納方式の説明図である。
- 【図 50】公開鍵の認証鍵レジスタへの格納処理のフローチャートである。 50

- 【図 5 1】暗号化された共通鍵の認証鍵レジスタへの格納方式の説明図である。
- 【図 5 2】暗号化された共通鍵の認証鍵レジスタへの格納処理フローチャートである。
- 【図 5 3】物理メモリへのページイン時の処理方式の説明図である。
- 【図 5 4】物理メモリへのページイン時の処理フローチャートである。
- 【図 5 5】認証部の構成を示すブロック図である。
- 【図 5 6】認証部の動作フローチャートである。
- 【図 5 7】第 5 の実施例におけるページ利用時のメモリアクセス制御部によるアクセスチェック方式の説明図である。
- 【図 5 8】メモリアクセス制御部の動作例を説明する図である。
- 【図 5 9】命令フェッチ時のメモリアクセス制御部の処理フローチャートである。 10
- 【図 6 0】セキュアコアとノーマルコアからのページ利用時のアクセス制御方式を説明する図である。
- 【図 6 1】セキュアモードとノーマルモードを切り替えるためのモードレジスタを備えるプロセッサの構成図である。
- 【図 6 2】セキュアDMAの構成を示すブロック図である。
- 【図 6 3】セキュアDMAによるデータ転送処理のフローチャートである。
- 【図 6 4】OSによるページイン時の処理のフローチャートである。
- 【図 6 5】第 7 の実施例におけるコンテキスト情報暗号化方式の説明図である。
- 【図 6 6】コンテキスト情報の復号方式の説明図である。
- 【図 6 7】コンテキスト情報に対する改ざん検出情報付加方式の説明図である。 20
- 【図 6 8】コンテキスト情報に対する改ざん検出方式の説明図である。
- 【図 6 9】セキュア動作コンテキスト情報の暗号化方式の説明図である。
- 【図 7 0】セキュア動作コンテキスト情報に対する改ざん検出情報付加方式の説明図である。
- 【図 7 1】ページ・テーブル・エントリの暗号化方式の説明図である。
- 【図 7 2】ページ・テーブル・エントリの復号方式の説明図である。
- 【図 7 3】ページ・テーブル・エントリへの改ざん検出情報付加方式の説明図である。
- 【図 7 4】ページ・テーブル・エントリに対する改ざん検出方式の説明図である。
- 【図 7 5】本発明を実現するためのプログラムのコンピュータへのローディングを説明する図である。 30
- 【符号の説明】
- 【 0 2 3 4 】
- 1 0、4 0、1 0 0 プロセッサ
- 1 1、1 5 4 コア
- 1 2 暗号処理ブロック
- 1 3 コード認証処理ブロック
- 1 4 暗号化ROMコード領域
- 1 5 CPU固有鍵
- 1 7 主記憶
- 1 8 二次記憶 40
- 2 0 セキュアハードウェア
- 2 1 暗号鍵設定部
- 2 2 復号部
- 2 3、1 3 7 プロセッサ固有RSA秘密鍵
- 2 4、1 1 4 トランслーション・ルックアサイド・バッファ(TLB)
- 2 5 論理アドレステーブル
- 2 6 物理アドレステーブル
- 2 7 鍵テーブル
- 2 8 署名検証部
- 2 9、1 3 4 認証局証明書(認証局公開鍵) 50

3 1、 1 5 1	セキュアコア	
3 2、 1 5 2	ノーマルコア	
3 4	鍵生成機構	
4 1	実行ユニット	
4 2	ロードストアユニット	
4 3	暗号化回路	
4 4	復号化回路	
4 5	キャッシュメモリ	
4 6	メモリ管理ユニット	
4 7、 4 8、 7 3、 7 4、 7 5	鍵テーブルメモリ	10
5 1、 5 2、 7 8、 7 9	鍵選択レジスタ	
7 1	リードモデファイライトバッファ	
8 8	アドレスマップレジスタ (A M R)	
1 0 1	物理ページ	
1 0 2	I / O 装置	
1 0 5	メモリアクセス制御部	
1 0 6	命令解釈部	
1 0 7	認証部	
1 0 8	暗号化 / 復号、署名生成 / 検証部	
1 0 9	セキュアコンテキスト識別子生成部	20
1 1 0	セキュアコンテキスト識別子消滅部	
1 1 1	プロセッサ固有鍵	
1 1 2	認証情報一次格納部	
1 1 3	セキュア D M A	
1 1 5	コンテキスト情報格納部	
1 1 7	プログラムカウンタ	
1 1 8	セキュアコンテキスト識別子レジスタ	
1 1 9	認証鍵レジスタ	
1 2 0	レジスタ群	
1 2 4	物理ページ	30
1 2 5	ページ	
1 2 6	認証情報	
1 4 0	S H A - 1 ハッシュ演算器	
1 4 1	R S A 復号器	
1 4 2	比較器	
1 4 4	命令実行部	
1 5 5	モードレジスタ	
1 6 5	暗号器	
1 6 8	復号器	
1 6 9	改ざん検出情報生成器	40
1 7 2	改ざん検出器	
1 7 4	暗号器 / 復号器	
1 7 9	改ざん検出情報生成器 / 改ざん検出器	

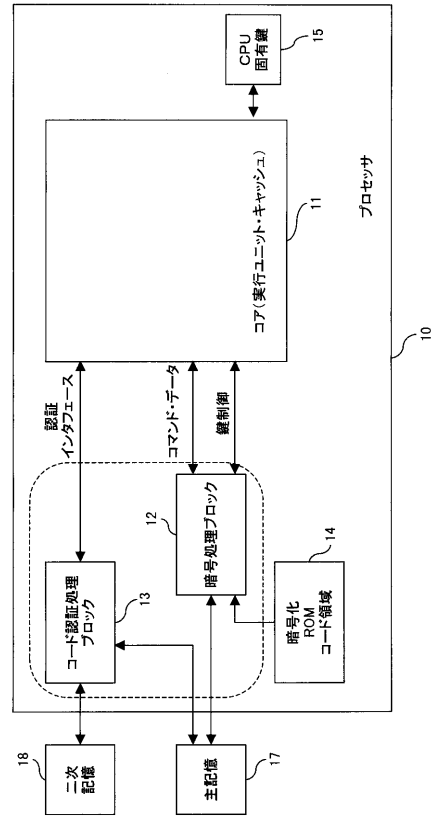
【 図 1 】

本発明のセキュアプロセッサの原理構成ブロック図



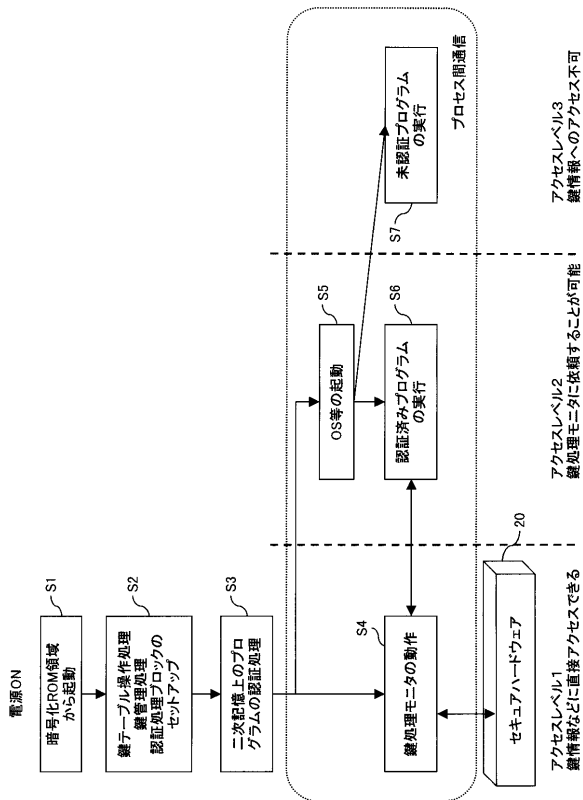
【 図 2 】

第1の実施例におけるプロセッサの基本構成を示すブロック図



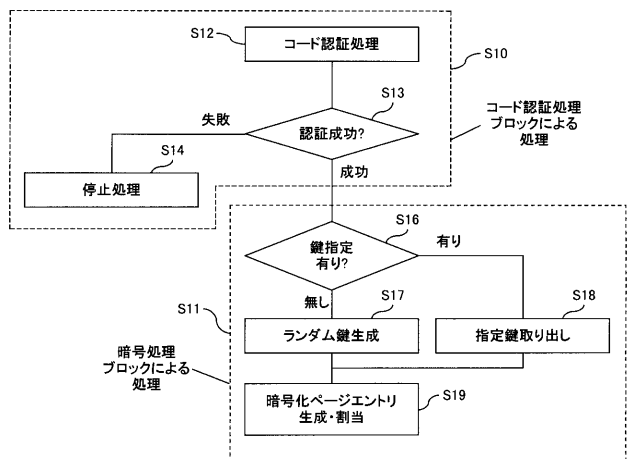
【 図 3 】

第1の実施例におけるプロセッサの基本処理フローチャート



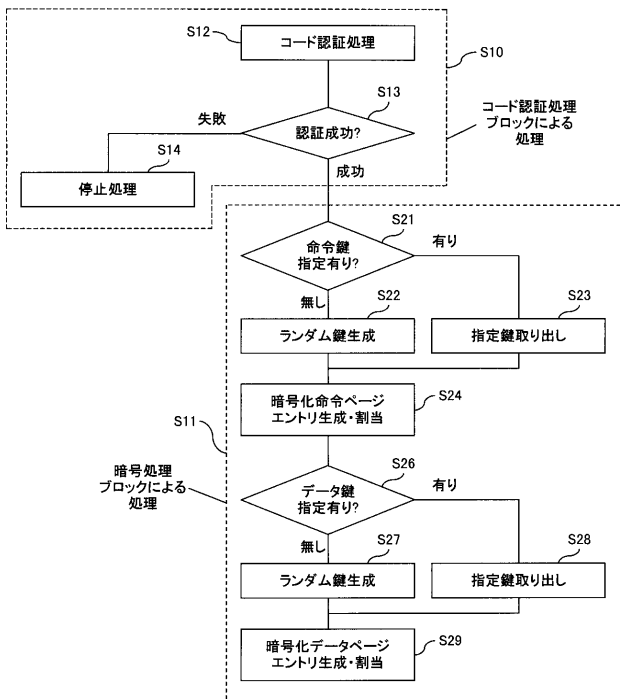
【 図 4 】

コード認証処理ブロックと暗号処理ブロックによる処理のフローチャート



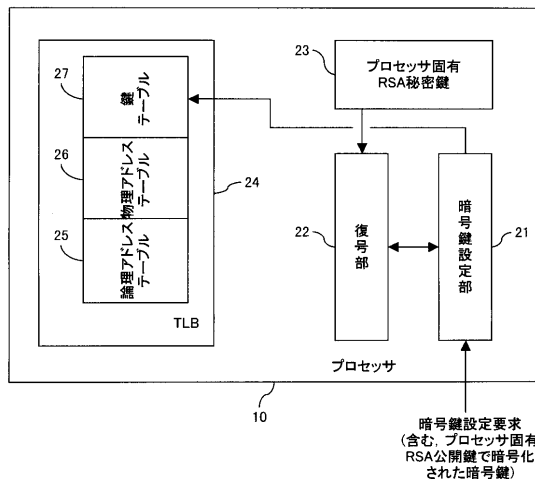
【 図 5 】

命令領域とデータ領域とによって異なる鍵が指定されている場合の暗号処理ブロックの処理フローチャート



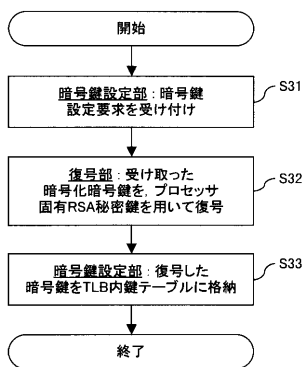
【 図 6 】

公開鍵で暗号化された暗号鍵の格納方式の説明図



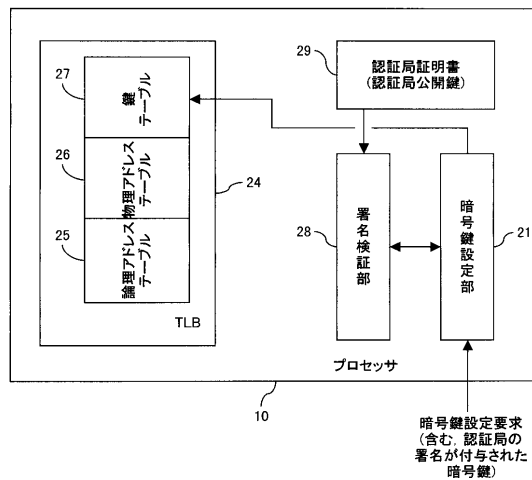
【 図 7 】

公開鍵で暗号化された暗号鍵の格納処理フローチャート



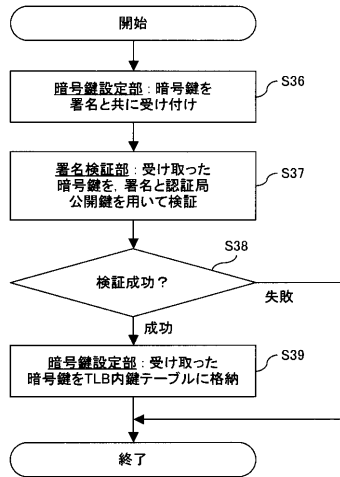
【 図 8 】

認証局の署名が付与された暗号鍵の格納方式の説明図



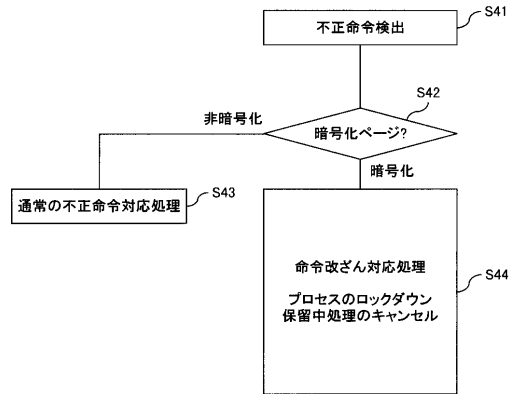
【 図 9 】

認証局の署名が付与された暗号鍵の格納処理フローチャート



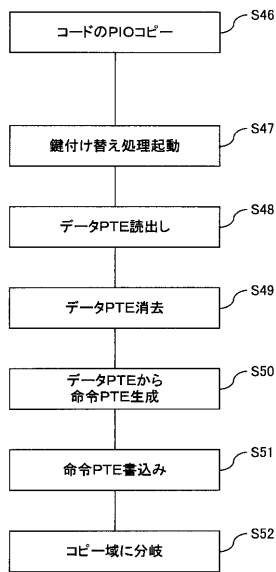
【 図 10 】

不正命令検出時の処理フローチャート



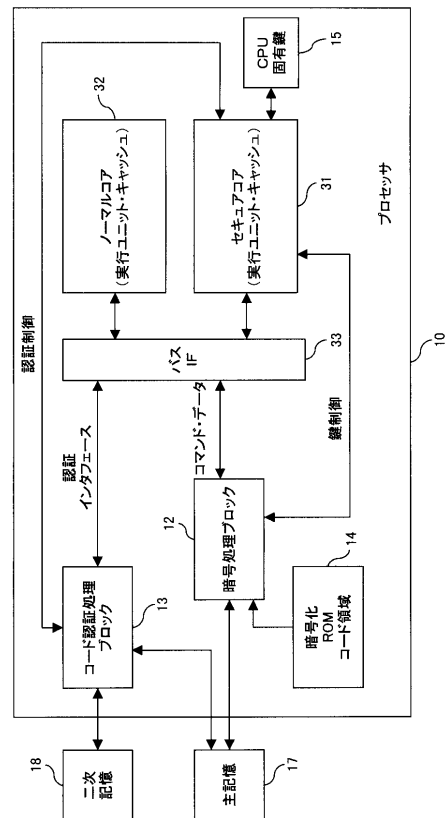
【 図 11 】

データ領域に格納された命令に対する鍵付け替え処理のフローチャート

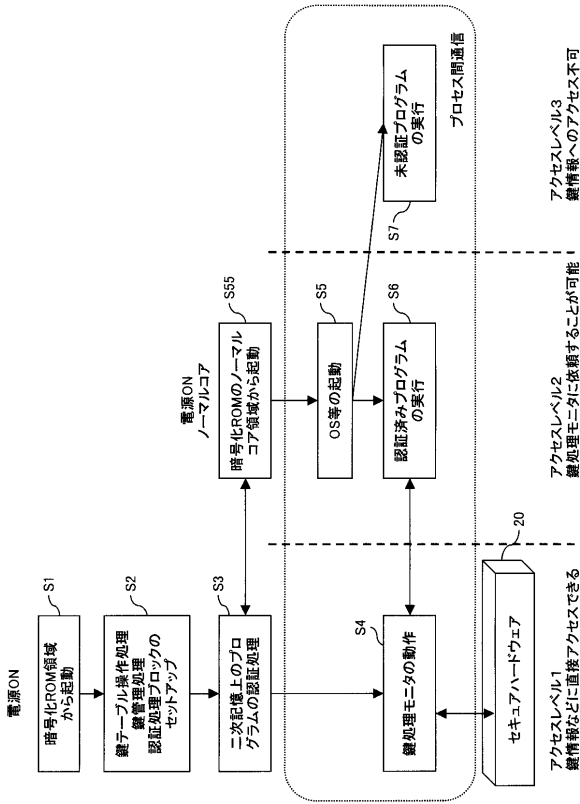


【 図 12 】

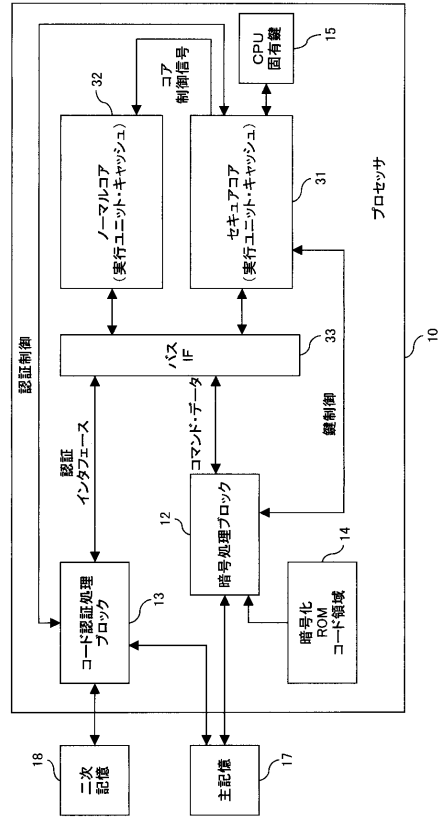
第2の実施例におけるプロセッサの基本構成を示すブロック図



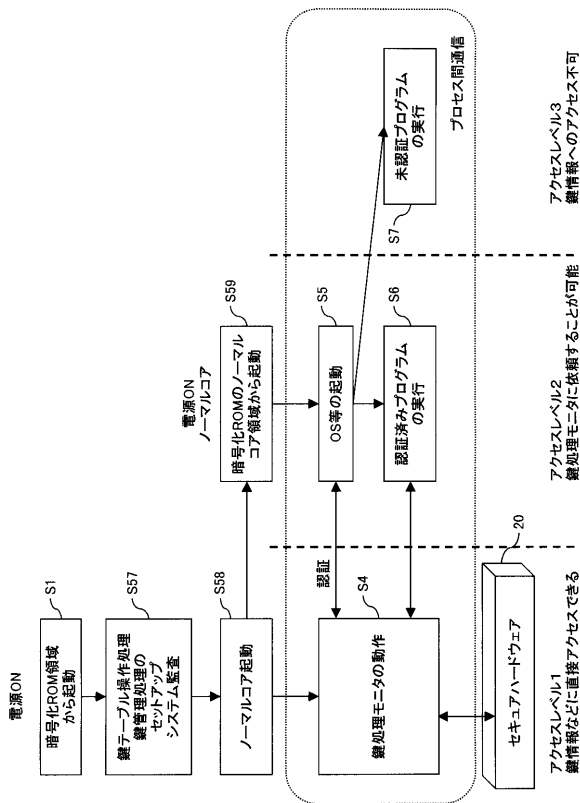
【図13】 第2の実施例におけるプロセッサの基本処理フローチャート



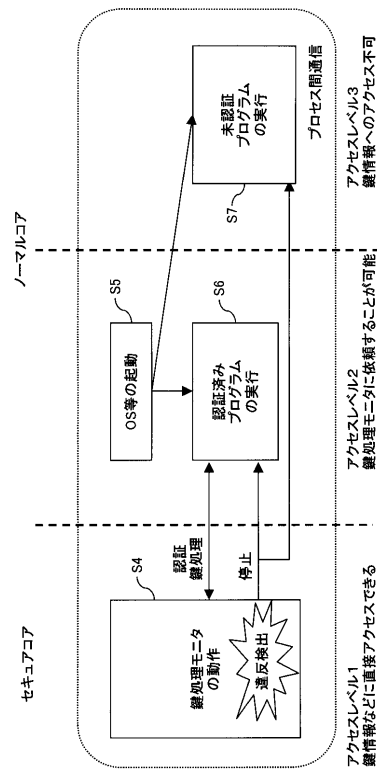
【図14】 セキュアコアとノーマルコアを備えるプロセッサの基本構成を示すブロック図



【図15】 図14のプロセッサにおける処理の基本フローチャート

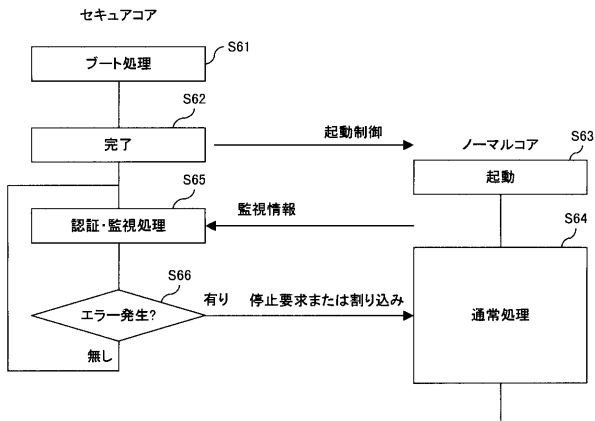


【図16】 図14のプロセッサにおけるセキュアコアによるノーマルコアの動作の停止制御方式の説明図



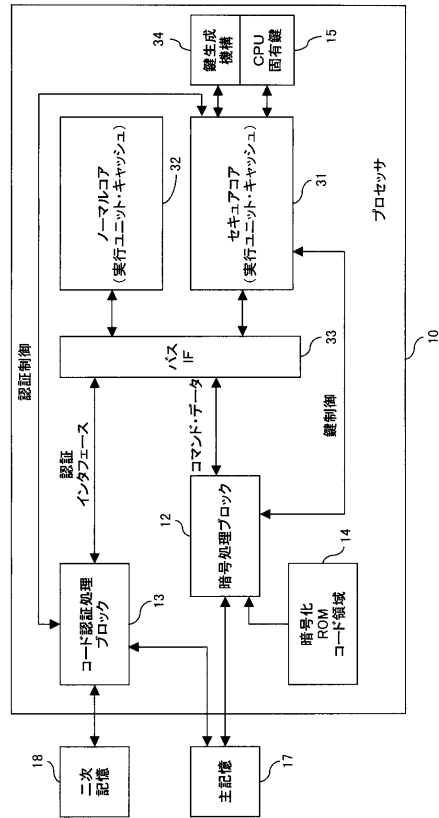
【 図 1 7 】

図14のプロセッサにおけるセキュアコアによる
ノーマルコアの動作の停止制御処理のフローチャート



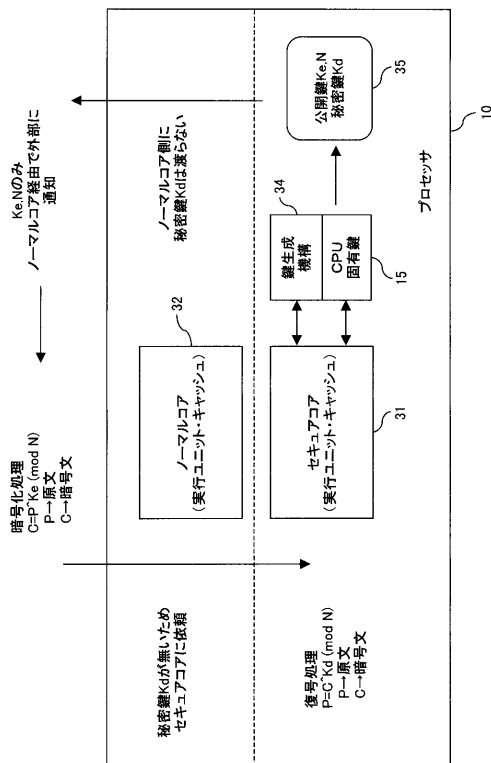
【 図 1 8 】

セキュアコアに対応する鍵生成機構を備える
プロセッサの構成ブロック図



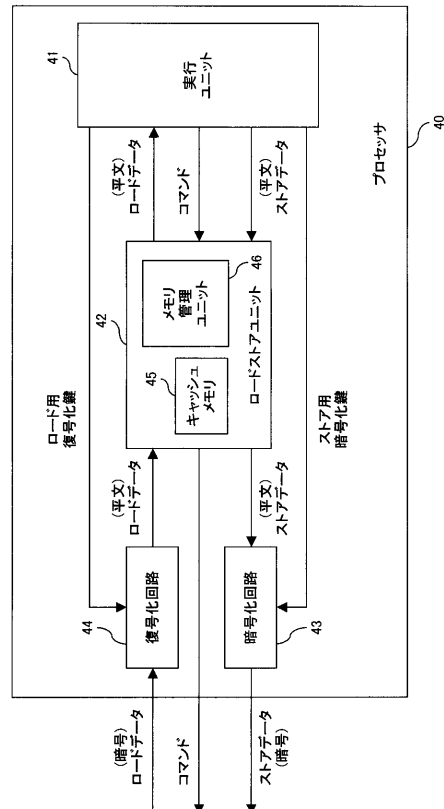
【 図 1 9 】

図18のプロセッサにおける鍵処理方式の
具体例の説明図



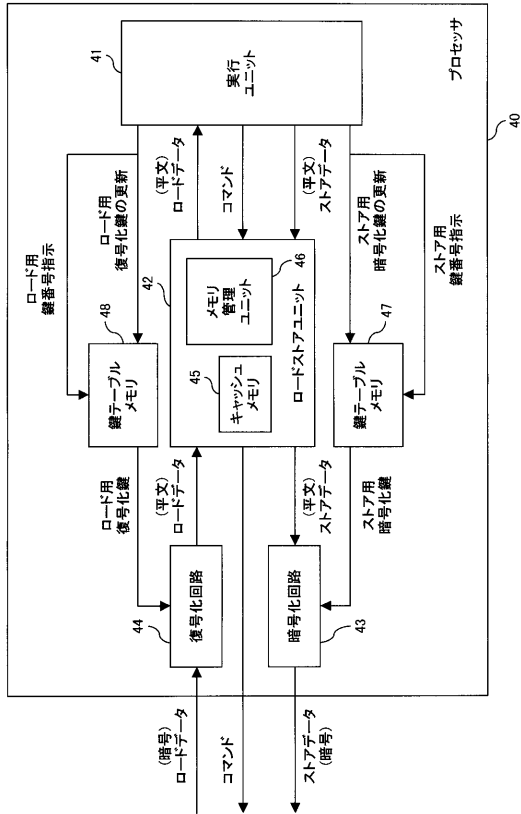
【 図 2 0 】

第3の実施例におけるプロセッサの
基本構成を示すブロック図



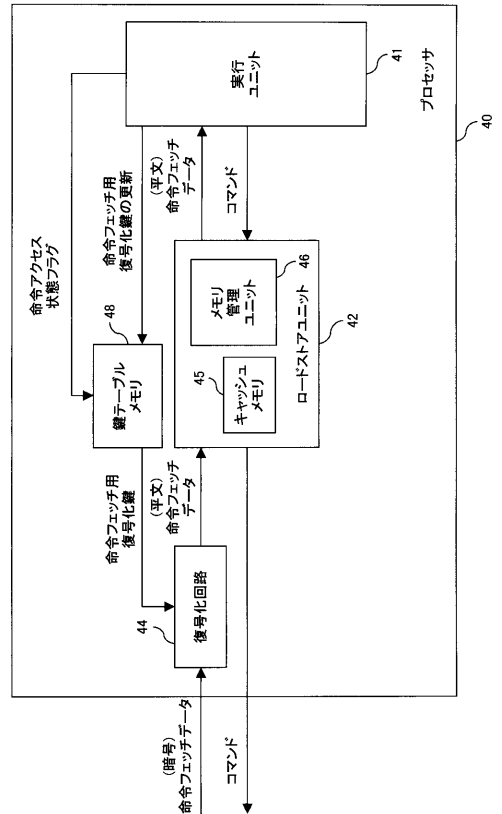
【図 2 1】

第3の実施例において鍵テーブルメモリを備えるプロセッサの構成ブロック図



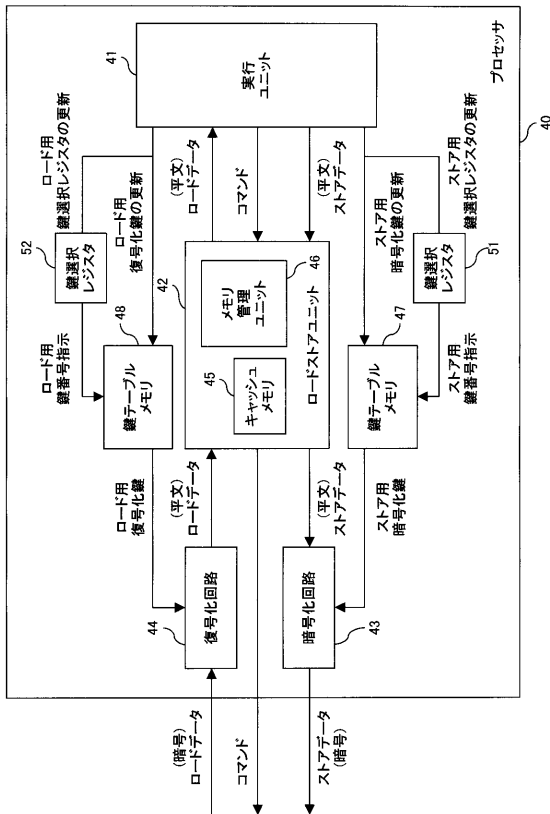
【図 2 2】

第3の実施例において命令アクセス状態にあるプロセッサの構成を示すブロック図



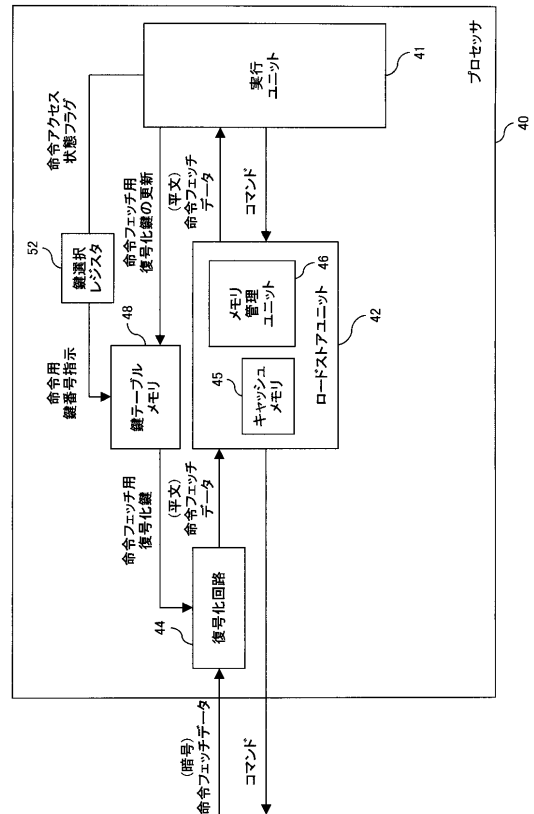
【図 2 3】

鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図



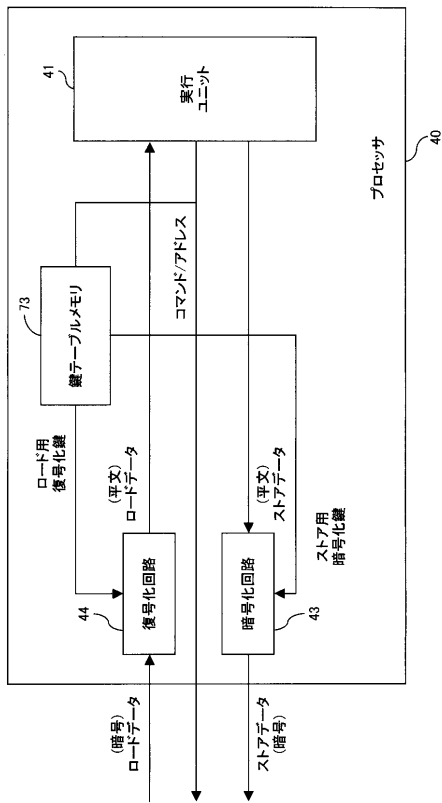
【図 2 4】

命令アクセス状態にあり、鍵テーブルメモリに対する鍵選択レジスタを備えるプロセッサの構成ブロック図



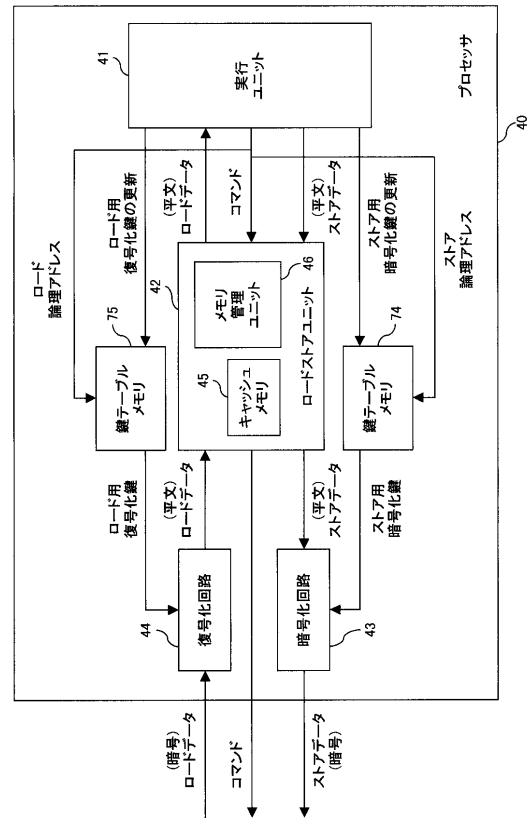
【図 29】

第4の実施例におけるプロセッサの基本構成を示すブロック図



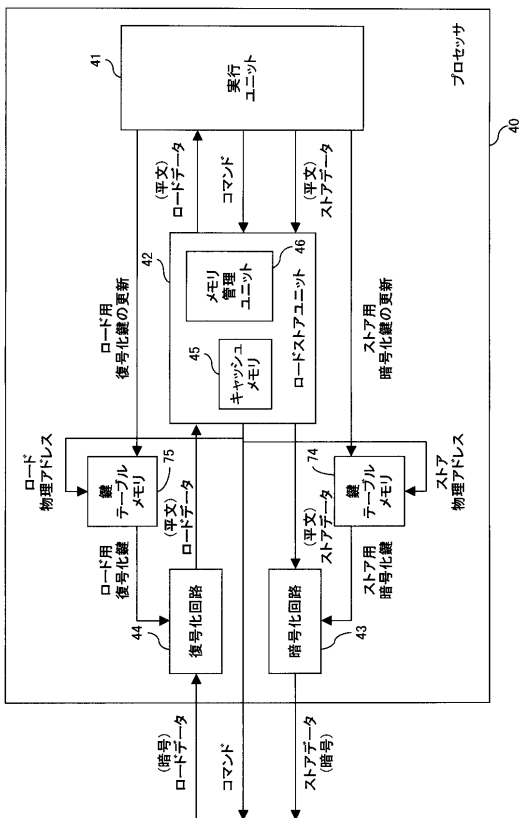
【図 30】

論理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図



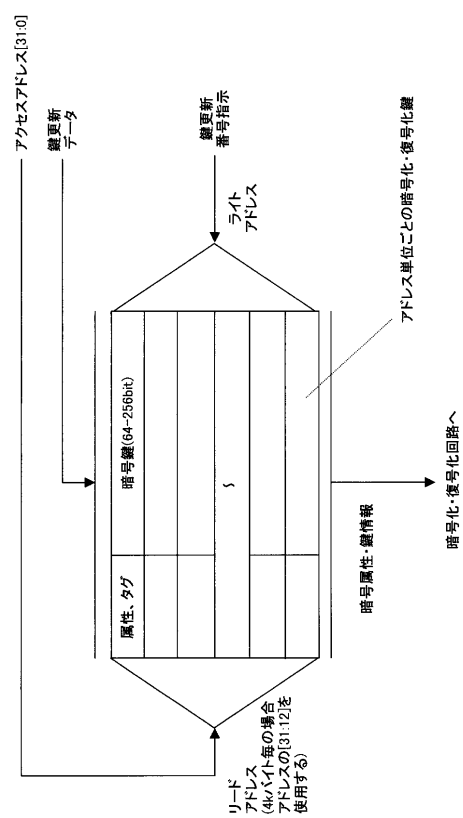
【図 31】

物理アドレスが与えられる鍵テーブルメモリを備えるプロセッサの構成ブロック図



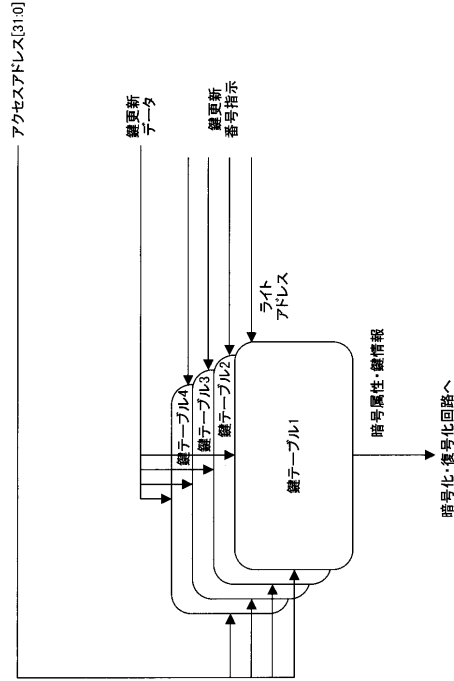
【図 32】

第4の実施例における鍵テーブルメモリの構成例(その1)を示す図



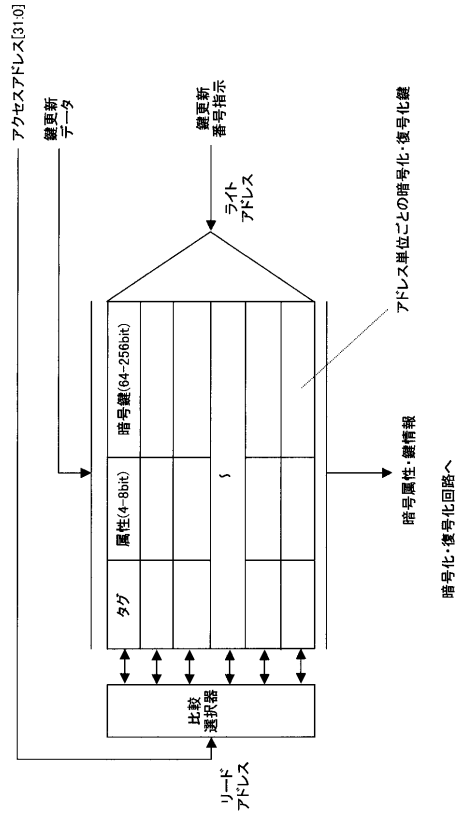
【 図 3 3 】

第4の実施例における鍵テーブルメモリ
の構成例(その2)を示す図



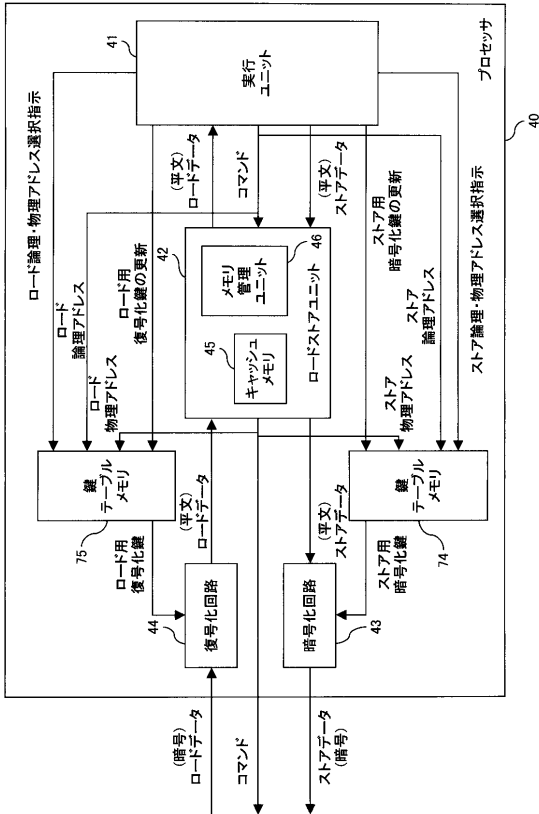
【 図 3 4 】

第4の実施例における鍵テーブルメモリ
の構成例(その3)を示す図



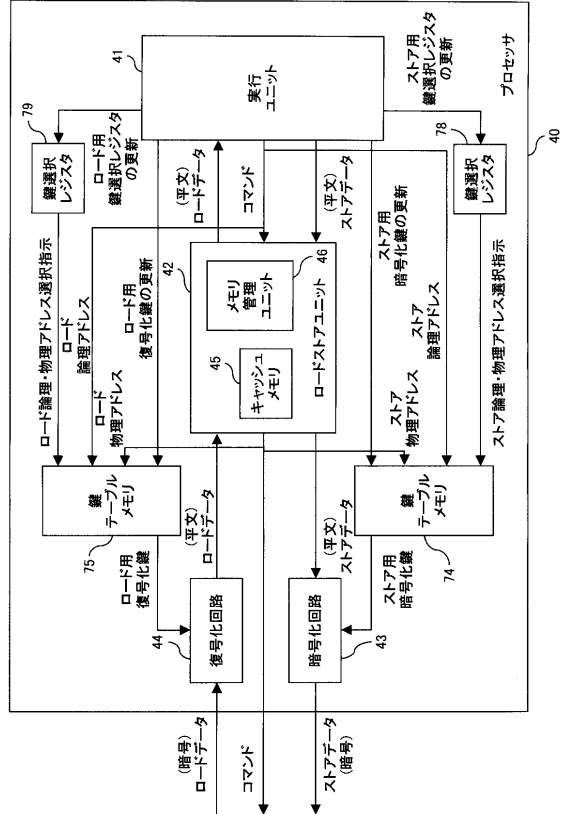
【 図 3 5 】

論理アドレスと物理アドレスとが与えられる鍵テーブル
メモリを備えるプロセッサの構成を示すブロック図



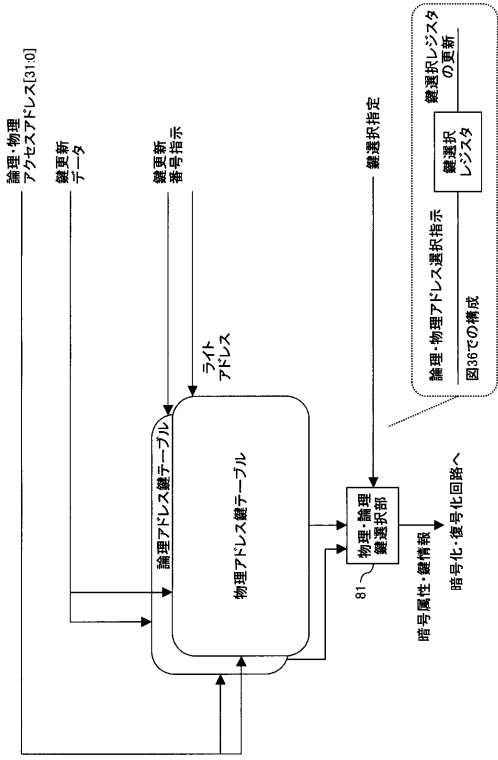
【 図 3 6 】

図35の鍵テーブルメモリに対してアドレス選択指示を
与える鍵選択レジスタを備えるプロセッサの構成ブロック図



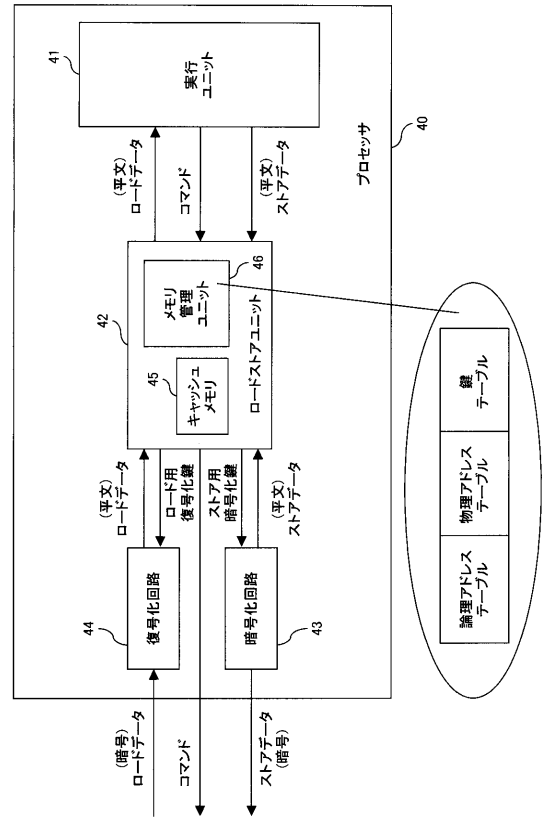
【 図 3 7 】

図35、図36における鍵テーブルメモリの構成例を示す図



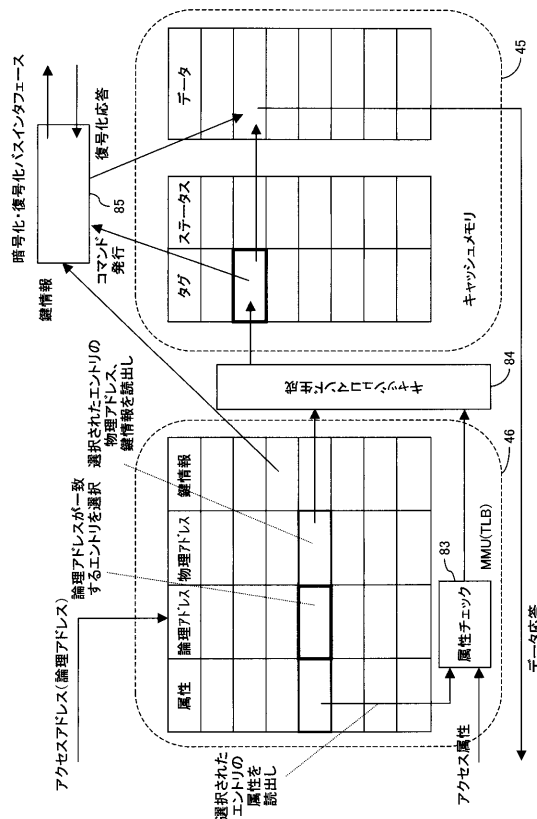
【 図 3 8 】

メモリ管理ユニット内に鍵テーブルが備えられるプロセッサの構成を示すブロック図



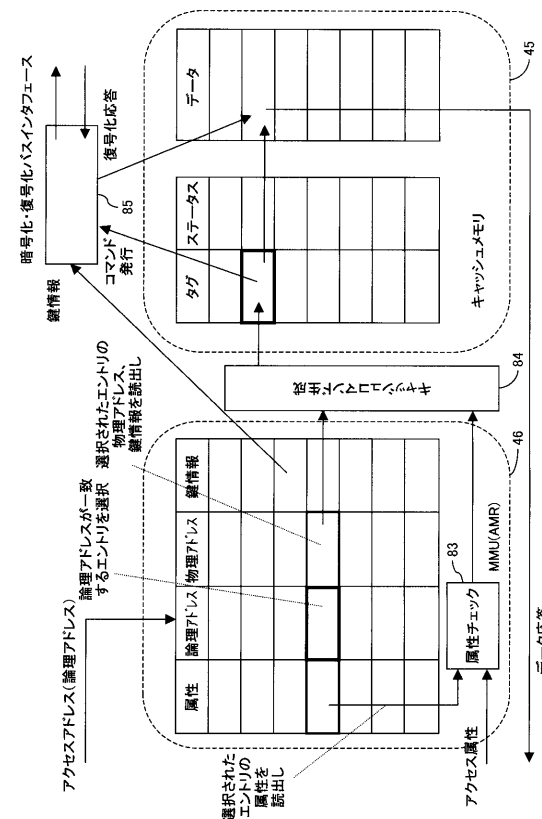
【 図 3 9 】

図38におけるデータアクセス方式の説明図



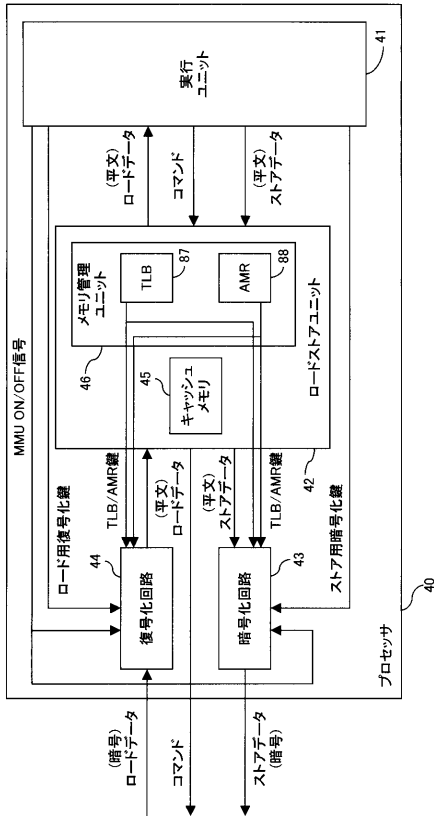
【 図 4 0 】

アドレスマップレジスタに鍵テーブルが併設されている場合のデータアクセス方式の説明図



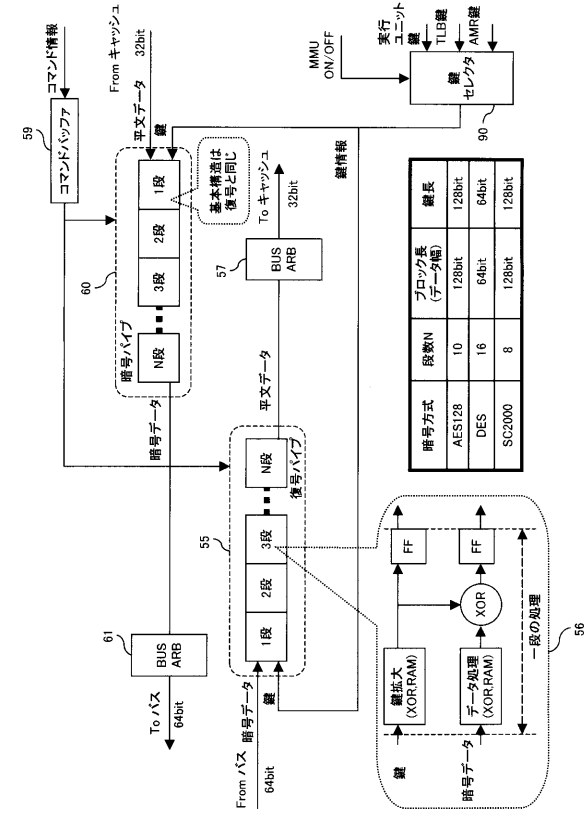
【 図 4 1 】

メモリ管理ユニットのON/OFF状態に応じて鍵を切り替えるプロセッサの構成を示すブロック図



【 図 4 2 】

メモリ管理ユニットのON/OFF状態に応じて鍵を切り替える暗号化/復号方式の説明図



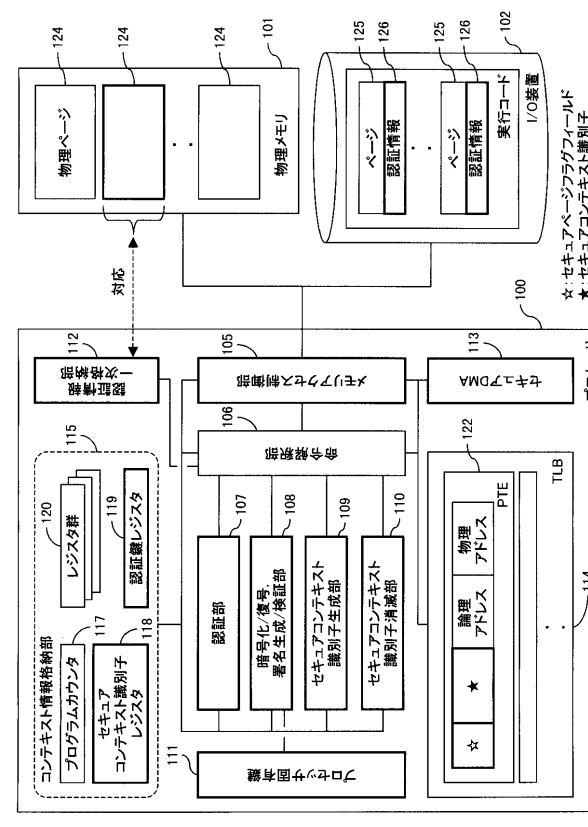
【 図 4 3 】

第3、および第4の実施例における実行ユニットの入出力信号の説明図

項目	図番号															
	20	21	22	23	24	21+22+23	27	29	30	31	35	36	38	40	41	備考
ロード復号化鍵	出力	○														
ロードデータ	入力	○														
ロード鍵番号指示	出力	○														
命令アドレスデータ	出力	○														
ストア復号化鍵	出力	○														
ストアデータ	出力	○														
ストア鍵番号指示	出力	○														
コマンド	出力	○														
実行状態番号	出力	○														
レジスタライト番号	出力	○														
レジスタリードデータ	入力	○														
レジスタライトデータ	出力	○														
アクセラドレス	出力	△	△	△	△	△	△	△	△	△	△	△	△	△	△	
ロードストア状態番号	出力	△	△	△	△	△	△	△	△	△	△	△	△	△	△	
鍵選択指示	出力															
MMU 状態番号	出力															
スーパーバイザ/ユーザ状態番号	出力															
コンテキストIDデータ	出力															

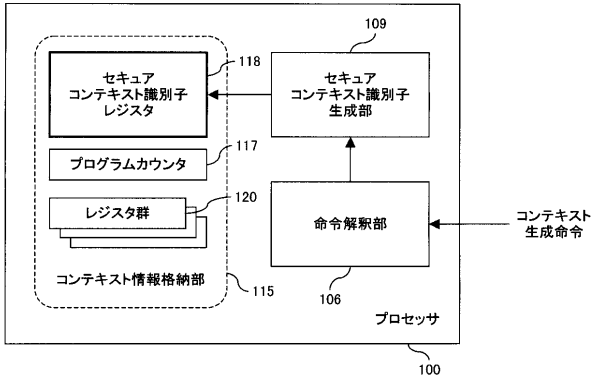
【 図 4 4 】

第5の実施例におけるプロセッサシステムの詳細構成ブロック図



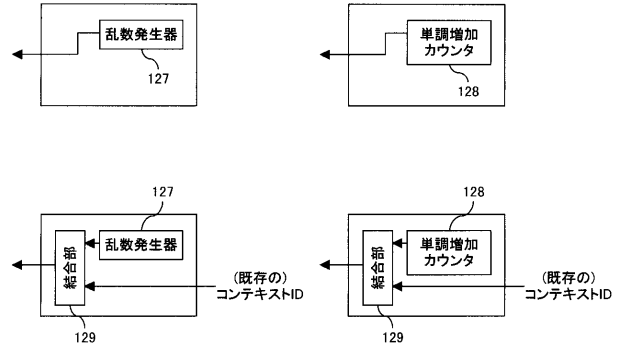
【 図 4 5 】

セキュアコンテキスト識別子生成方式の説明図



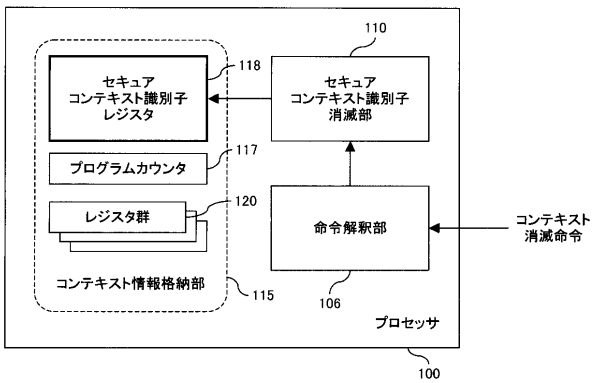
【 図 4 6 】

セキュアコンテキスト識別子生成方法の説明図



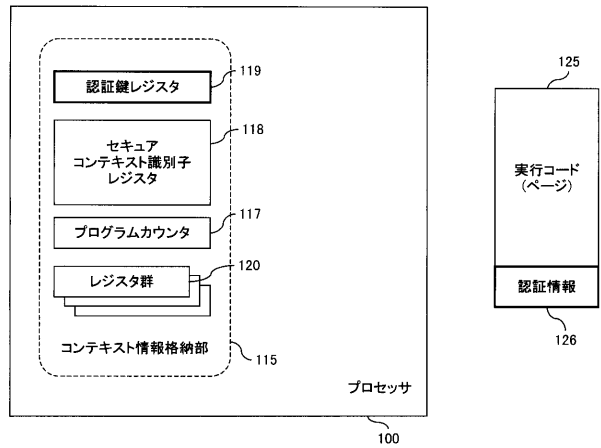
【 図 4 7 】

セキュアコンテキスト識別子消滅方式の説明図



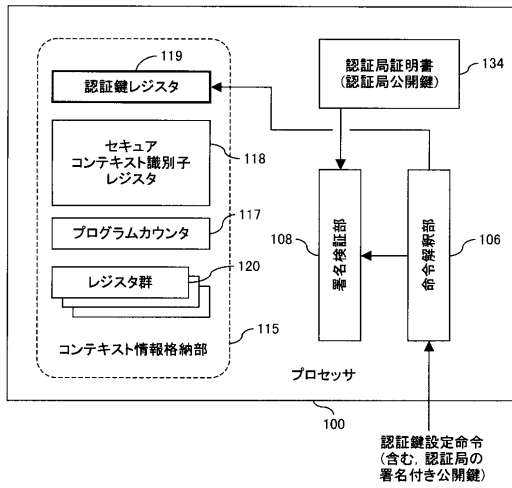
【 図 4 8 】

実行コードに付加された認証情報の説明図



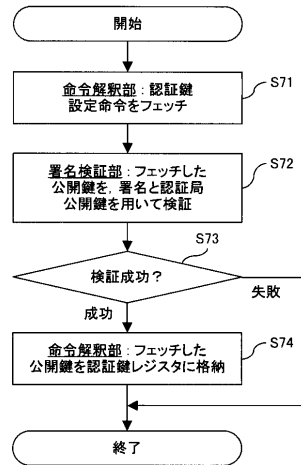
【図49】

公開鍵の認証鍵レジスタへの格納方式の説明図



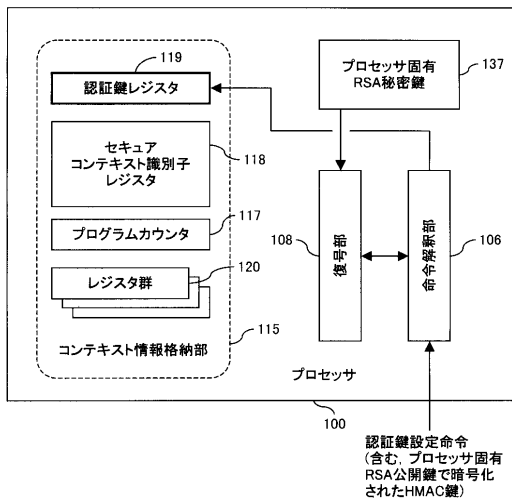
【図50】

公開鍵の認証鍵レジスタへの格納処理のフローチャート



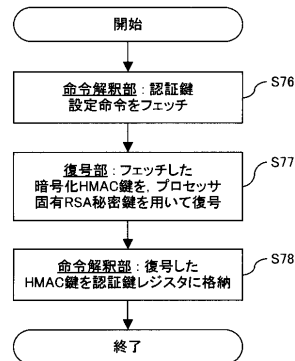
【図51】

暗号化された共通鍵の認証鍵レジスタへの格納方式の説明図



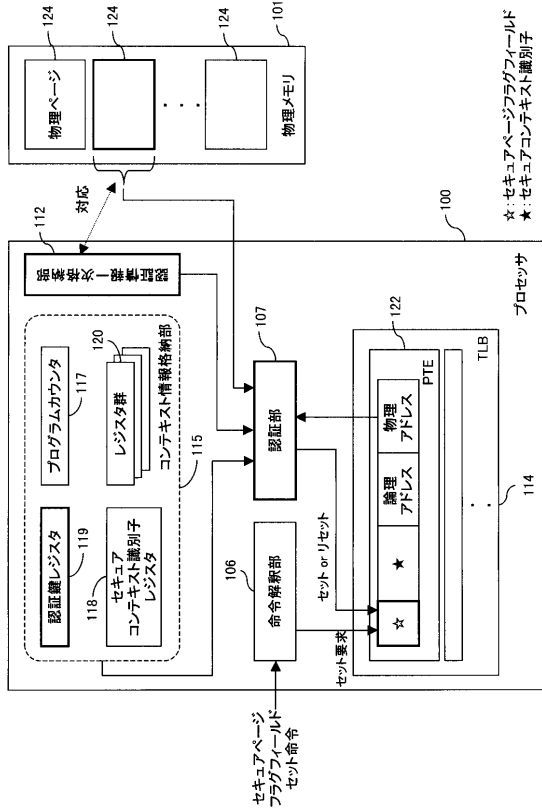
【図52】

暗号化された共通鍵の認証鍵レジスタへの格納処理フローチャート



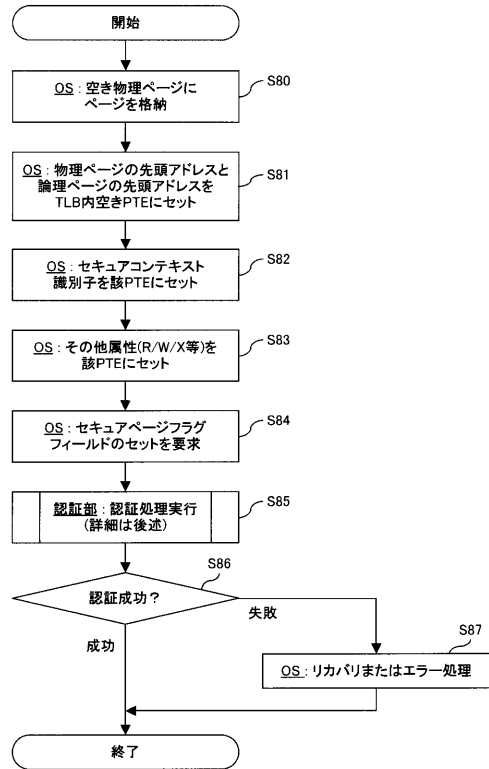
【 図 5 3 】

物理メモリへのページイン時の
処理方式の説明図



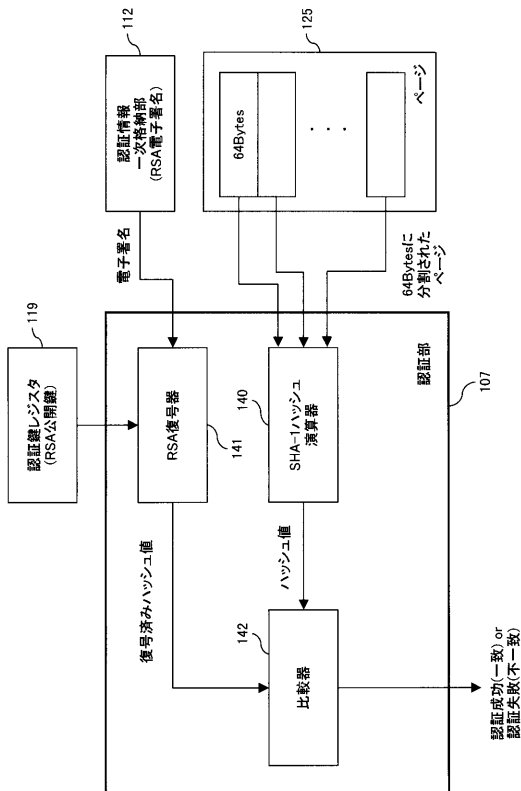
【 図 5 4 】

物理メモリへのページイン時の
処理フローチャート



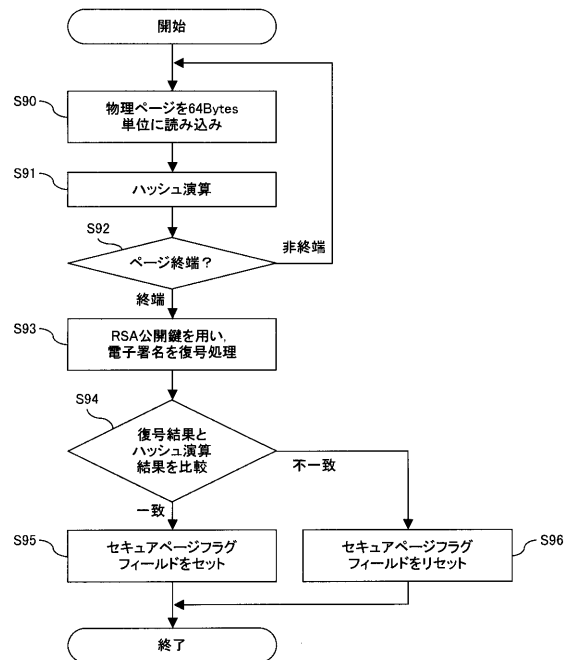
【 図 5 5 】

検証部の構成を示すブロック図



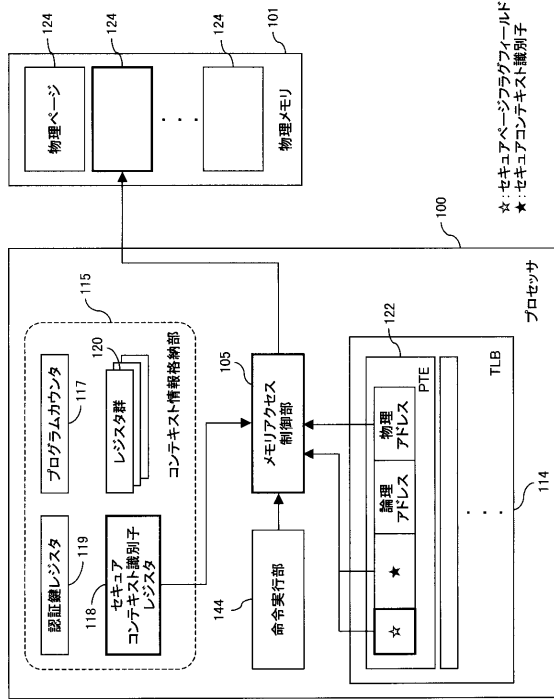
【 図 5 6 】

検証部の動作フローチャート



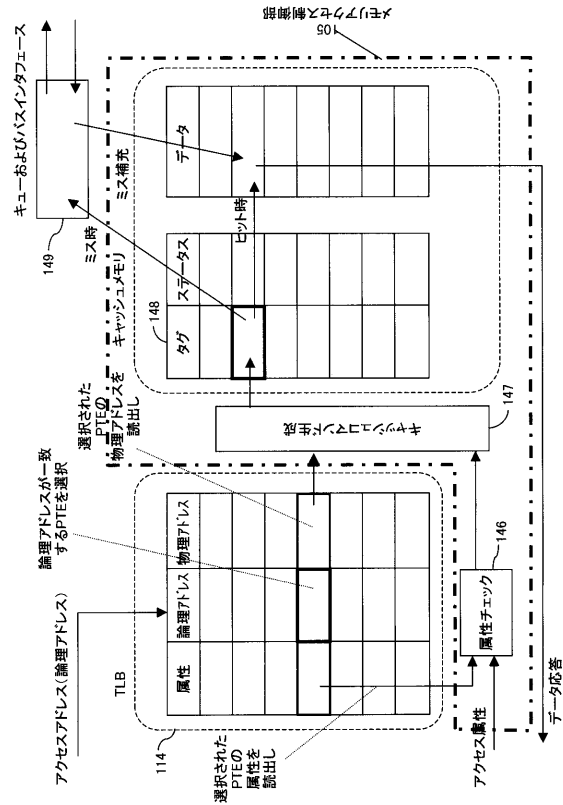
【図57】

第5の実施例におけるページ利用時の
メモリアクセス制御部による
アクセスチェック方式の説明図



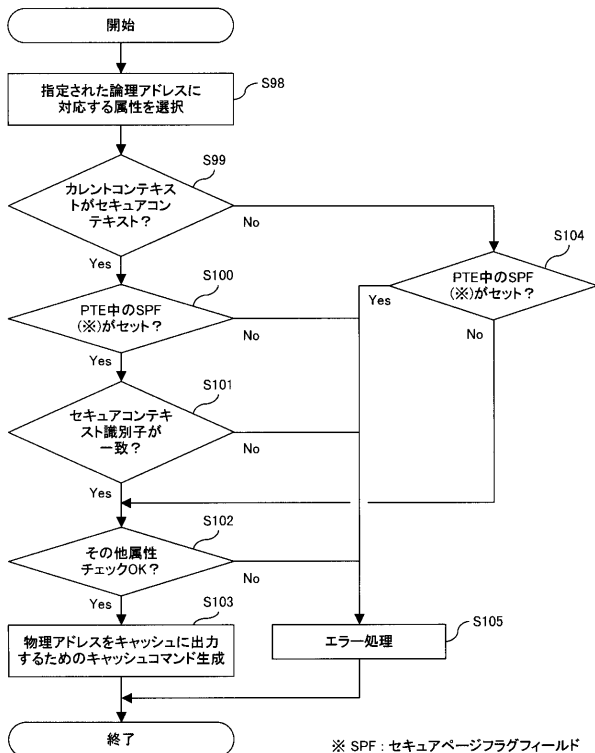
【図58】

メモリアクセス制御部の動作例を説明する図



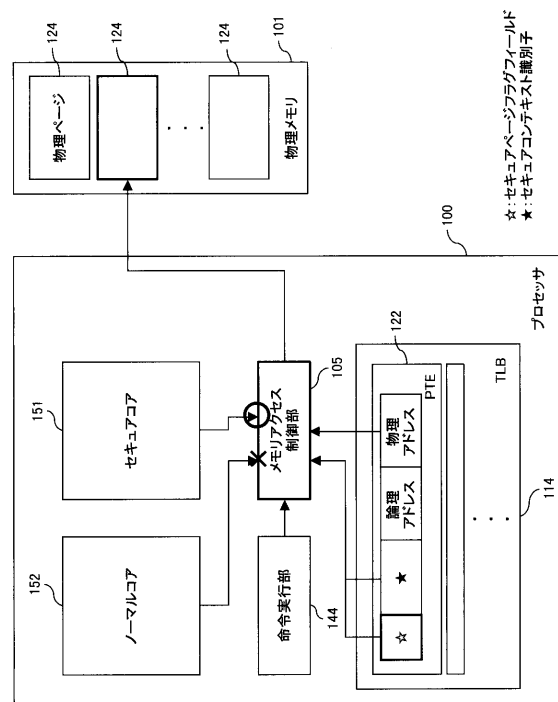
【図59】

命令フェッチ時のメモリアクセス
制御部の処理フローチャート

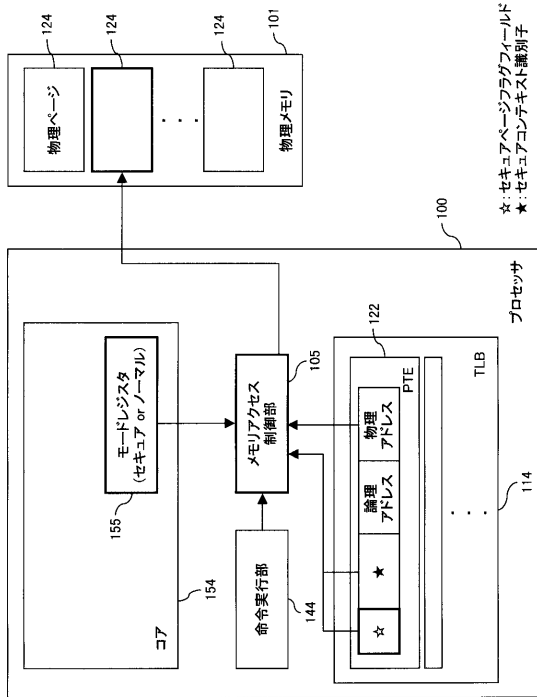


【図60】

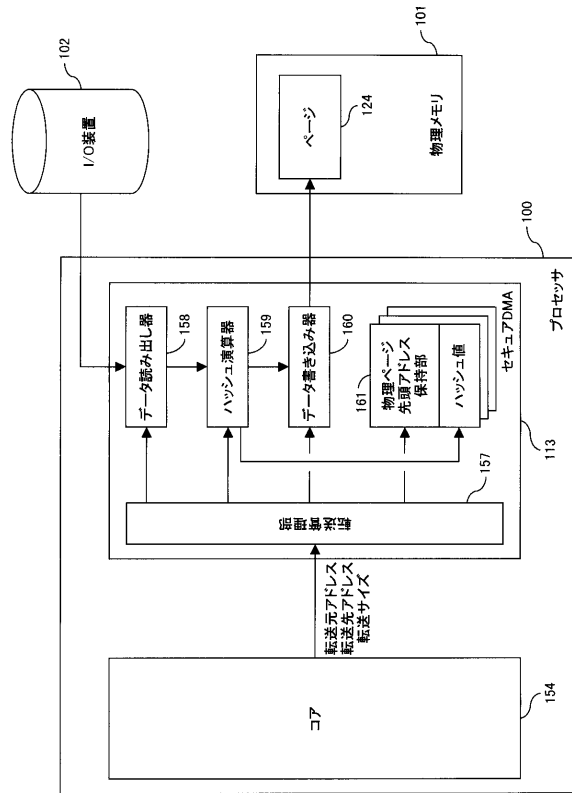
セキュアコアとノーマルコアからのページ
利用時のアクセス制御方式を説明する図



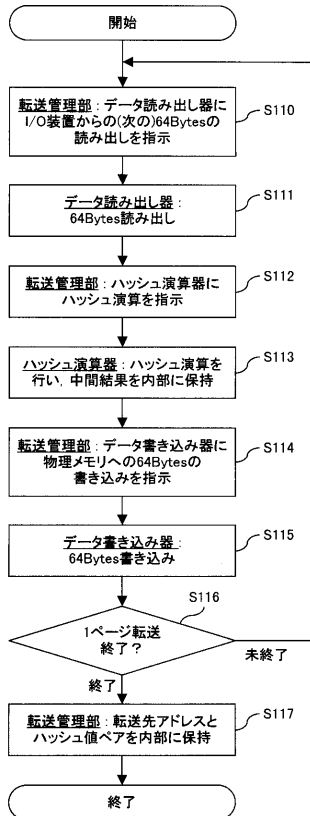
【 図 6 1 】
 セキュアモードとノーマルモードを切り替えるための
 モードレジスタを備えるプロセッサの構成図



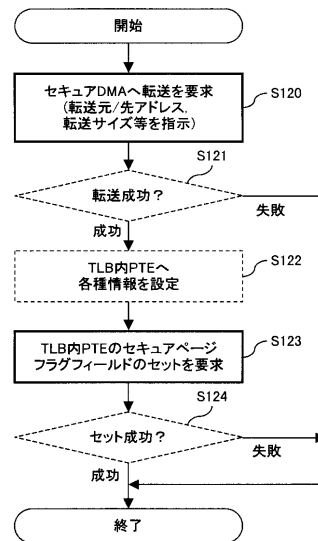
【 図 6 2 】
 セキュアDMAの構成を示すブロック図



【 図 6 3 】
 セキュアDMAによるデータ
 転送処理のフローチャート

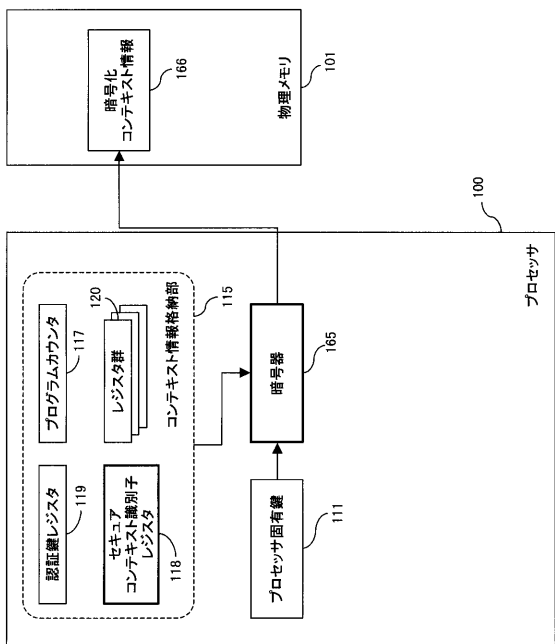


【 図 6 4 】
 OSによるページイン時の
 処理のフローチャート



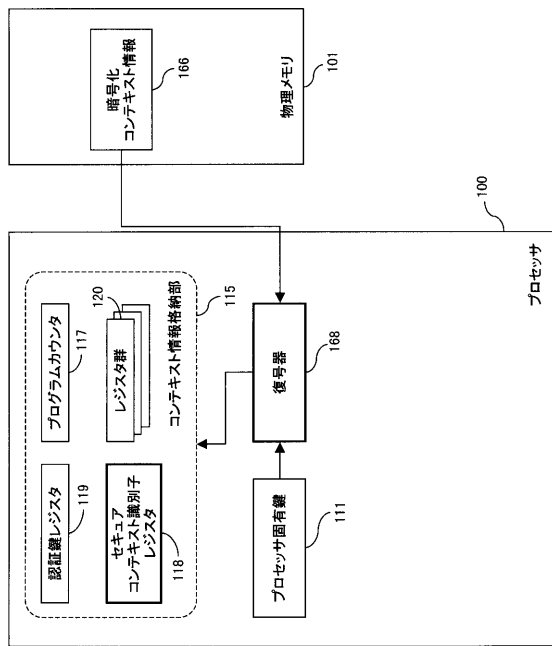
【図 65】

第7の実施例におけるコンテキスト
情報暗号化方式の説明図



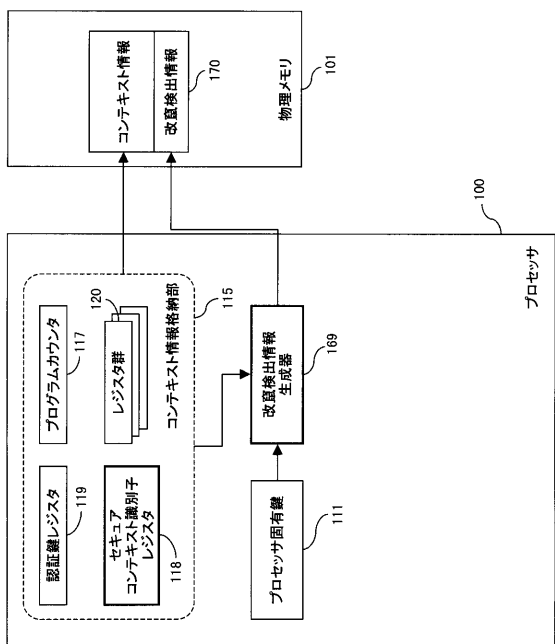
【図 66】

コンテキスト情報の復号方式の説明図



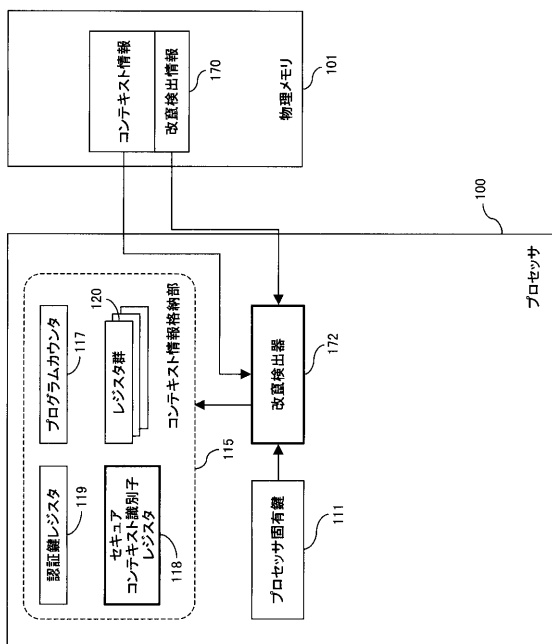
【図 67】

コンテキスト情報に対する改ざん
検出情報付加方式の説明図



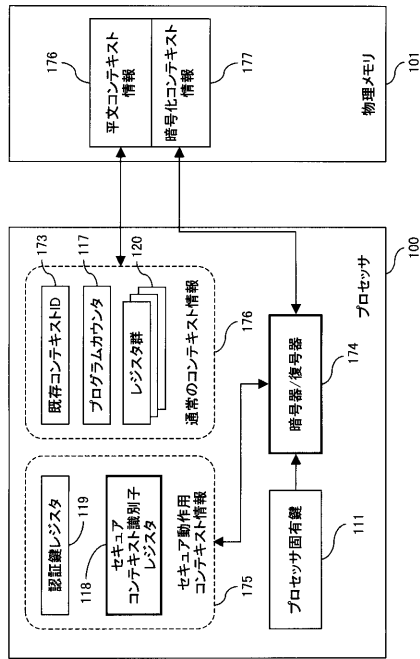
【図 68】

コンテキスト情報に対する改ざん
検出方式の説明図



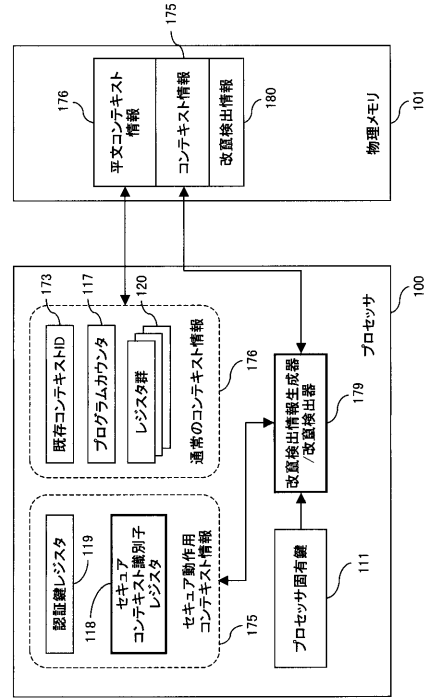
【図69】

セキュア動作コンテキスト情報の暗号化方式の説明図



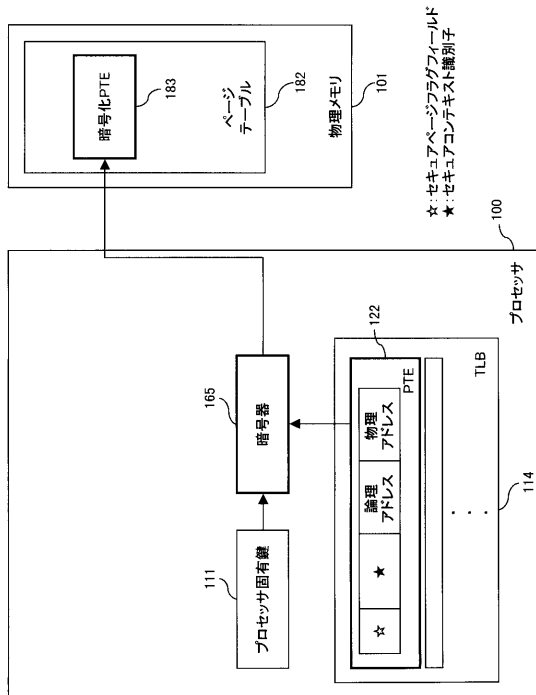
【図70】

セキュア動作コンテキスト情報に対する改ざん検出情報付加方式の説明図



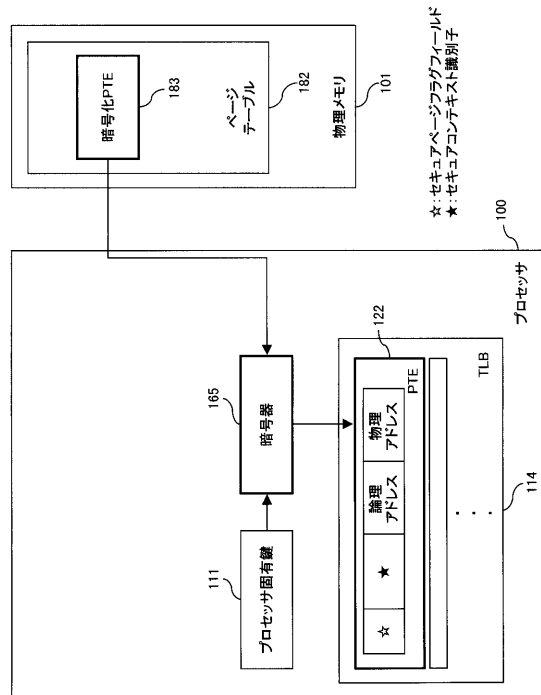
【図71】

ページ・テーブル・エントリの暗号化方式の説明図

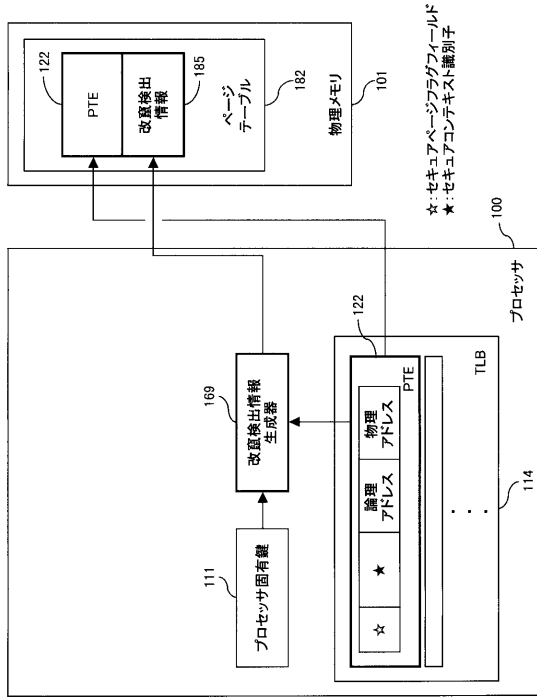


【図72】

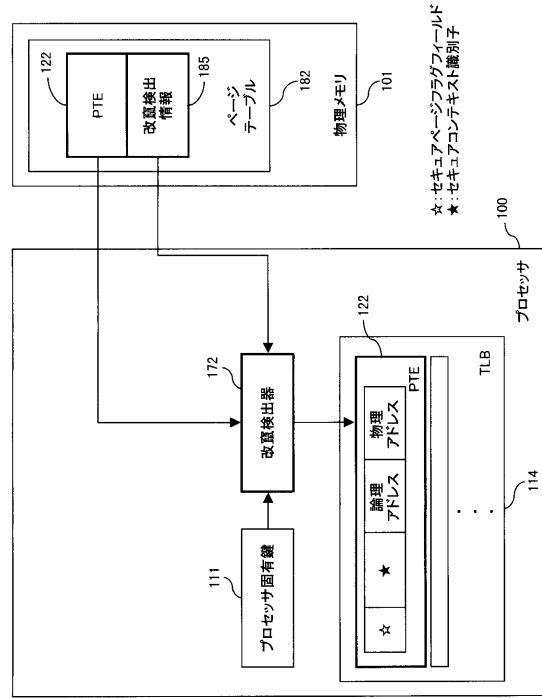
ページ・テーブル・エントリの復号方式の説明図



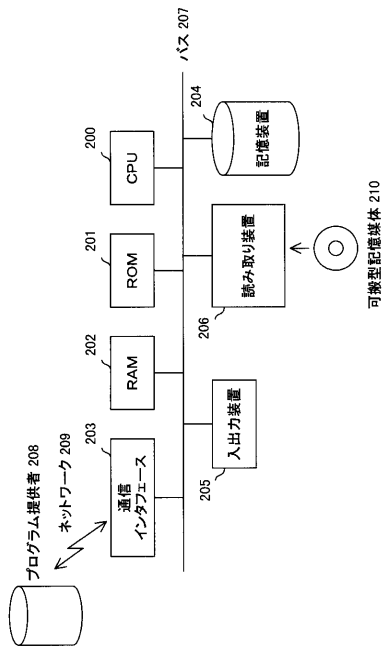
【 図 7 3 】
 ページ・テーブル・エントリへの
 改ざん検出情報付加方式の説明図



【 図 7 4 】
 ページ・テーブル・エントリに対する
 改ざん検出方式の説明図



【 図 7 5 】
 本発明を実現するためのプログラムのコ
 ンピュータへのローディングを説明する図



フロントページの続き

(72)発明者 田宮 大司

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

Fターム(参考) 5B017 AA03 AA08 BA07 BA09 CA01 CA12 CA15

5B033 FA20

5B076 FA01 FB01 FD08

5J104 NA37 NA39 NA42 PA07