

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3880933号  
(P3880933)

(45) 発行日 平成19年2月14日(2007.2.14)

(24) 登録日 平成18年11月17日(2006.11.17)

(51) Int. Cl.		F I			
HO4L	9/10	(2006.01)	HO4L	9/00	621A
GO6F	21/22	(2006.01)	GO6F	9/06	660L
GO6F	12/08	(2006.01)	GO6F	12/08	559Z
GO6F	21/24	(2006.01)	GO6F	12/14	540A

請求項の数 8 (全 18 頁)

(21) 出願番号	特願2003-12558 (P2003-12558)	(73) 特許権者	000003078
(22) 出願日	平成15年1月21日(2003.1.21)		株式会社東芝
(65) 公開番号	特開2004-228786 (P2004-228786A)		東京都港区芝浦一丁目1番1号
(43) 公開日	平成16年8月12日(2004.8.12)	(74) 代理人	100083806
審査請求日	平成16年6月4日(2004.6.4)		弁理士 三好 秀和
		(74) 代理人	100100712
			弁理士 岩▲崎▼ 幸邦
		(74) 代理人	100100929
			弁理士 川又 澄雄
		(74) 代理人	100108707
			弁理士 中村 友之
		(74) 代理人	100095500
			弁理士 伊藤 正和
		(74) 代理人	100101247
			弁理士 高橋 俊一

最終頁に続く

(54) 【発明の名称】 耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法

(57) 【特許請求の範囲】

【請求項1】

暗号化されたプログラムの実行コードの読み出し要求を処理するキャッシュメモリ制御手段と、

前記キャッシュメモリ制御手段からの復号化要求に基づき、前記実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、

複数のキャッシュラインを有し、前記復号化手段にて復号化した前記実行コードを記憶するデータ領域と、前記実行コードの復号化に用いた暗号鍵を格納する秘密保護属性保持部とを、同一キャッシュライン上に有するキャッシュメモリと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタと

を具備し、

前記キャッシュメモリ制御手段は、

前記実行コードの読み出し要求を処理する際に、前記キャッシュメモリ内に前記実行コードが存在するときは、前記鍵値レジスタに記憶されている実行中のプログラムの暗号鍵と前記キャッシュメモリ内の実行コードが格納されたキャッシュラインの前記秘密保護属性保持部に格納された暗号鍵とを比較することによって、前記キャッシュメモリ内の実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断し、前記キャッシュメモリ内の実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、前記復号化手段へ前記復号化要求を出す代

10

20

わりに、前記キャッシュメモリ内の実行コードを使用するよう制御することを特徴とする耐タンパマイクロプロセッサ。

【請求項2】

暗号化されたプログラムのデータの読み出し要求を処理するキャッシュメモリ制御手段と、

前記キャッシュメモリ制御手段からの復号化要求に基づき、前記データを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、

複数のキャッシュラインを有し、前記復号化手段にて復号化した前記データを記憶するデータ領域と、前記データの復号化に用いた暗号鍵を格納する秘密保護属性保持部とを、同一キャッシュライン上に有するキャッシュメモリと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタと

を具備し、

前記キャッシュメモリ制御手段は、

前記データの読み出し要求を処理する際に、前記キャッシュメモリ内に前記データが存在するときは、前記鍵値レジスタに記憶されている実行中のプログラムの暗号鍵と前記キャッシュメモリ内のデータが格納されたキャッシュラインの前記秘密保護属性保持部に格納された暗号鍵とを比較することによって、前記キャッシュメモリ内のデータが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータである否かを判断し、前記キャッシュメモリ内のデータが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータであ

った場合には、前記復号化手段へ前記復号化要求を出す代わりに、前記キャッシュメモリ内のデータを使用するよう制御することを特徴とする耐タンパマイクロプロセッサ。

【請求項3】

前記キャッシュメモリ制御部は、前記秘密保護属性保持部内の暗号鍵を用いて前記データの処理結果を暗号化し、前記記憶装置に記憶させることを特徴とする請求項2に記載の耐タンパマイクロプロセッサ。

【請求項4】

前記キャッシュメモリ制御部は、前記データの処理結果をキャッシュメモリに書き込む際、前記鍵値レジスタの暗号鍵を前記秘密保護属性保持部に格納することを特徴とする請求項2または3に記載の耐タンパマイクロプロセッサ。

【請求項5】

暗号化されたプログラムの実行コードの読み出し要求を処理するステップと、

前記実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと

、  
復号化された前記実行コードを、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられたデータ領域の1つに記憶するステップと、

前記復号化に用いた暗号鍵を、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられた秘密保護属性保持部のうち、前記実行コードが格納されたキャッシュラインに設けられた秘密保護属性保持部に格納するステップと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップと、

前記実行コードの読み出し要求を処理する際に、前記キャッシュメモリ内に前記実行コードが存在するときは、前記鍵値レジスタに記憶されている実行中のプログラムの暗号鍵と前記キャッシュメモリ内の実行コードが格納されたキャッシュラインの前記秘密保護属性保持部に格納された暗号鍵とを比較することによって、前記キャッシュメモリ内の実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断し、前記キャッシュメモリ内の実行コードが前記所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、前記キャッシュメモリ内の実行コードを使用するよう制御するステップ

10

20

30

40

50

とを備えることを特徴とするキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項 6】

暗号化されたプログラムのデータの読み出し要求を処理するステップと、  
前記データを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、  
復号化された前記データを、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられたデータ領域の 1 つに記憶するステップと、

前記復号化に用いた暗号鍵を、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられた秘密保護属性保持部のうち、前記データが格納されたキャッシュラインに設けられた秘密保護属性保持部に格納するステップと、

前記暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップと、

前記データの読み出し要求を処理する際に、前記キャッシュメモリ内に前記データが存在するときは、前記鍵値レジスタに記憶されている実行中のプログラムの暗号鍵と前記キャッシュメモリ内のデータが格納されたキャッシュラインの前記秘密保護属性保持部に格納された暗号鍵とを比較することによって、前記キャッシュメモリ内のデータが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータである否かを判断し、前記キャッシュメモリ内のデータが前記所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、前記キャッシュメモリ内のデータを使用するよう制御するステップ

とを備えることを特徴とするキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項 7】

前記秘密保護属性保持部内の暗号鍵を用いて前記データの処理結果を暗号化し、前記記憶装置に記憶させるステップを含むことを特徴とする請求項 6 に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【請求項 8】

前記データの処理結果をキャッシュメモリに書き込む際、前記鍵値レジスタの暗号鍵を前記秘密保護属性保持部に格納するステップを含むことを特徴とする請求項 6 または 7 に記載のキャッシュメモリ搭載プロセッサによるデータアクセス制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マルチタスクのプログラム実行環境を支援する機能を持つマイクロプロセッサであって、キャッシュメモリ制御部及び暗号化・復号化機能を備え、プログラムの実行コード及び処理対象データの秘匿性の保護及び改変を防止する為の耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法に関する。

【0002】

【従来の技術】

近年のコンピュータシステムにおいては、パーソナルコンピュータのように、様々なメーカーのハードウェア及びソフトウェアの組み合わせ構築が可能なオープンシステムが広く普及している。オープンシステムでは、ハードウェア及びシステムプログラム等から成るオペレーティングシステム（以下「OS」と記載）の情報が一般に開示されている。この為、ユーザが開示情報をもとにOSのプログラムを改変・改竄することは原理的に可能である。

【0003】

アプリケーションプログラムはこのOSの管理下で動作するので、OS自体が改竄されてハッカー等の第三者により攻撃された場合、これを避ける手段がない。よって、アプリケーションプログラム提供者が、アプリケーションプログラムを第三者の解析や改竄から完全に保護することは困難であった。

【0004】

10

20

30

40

50

そこで、オープンシステムのOSで動作するアプリケーションプログラムの改竄及び解析を防止する為に、暗号化という手法を用いる。プログラムを暗号化した場合、解析が困難となり、そのプログラムを改変した場合の動作が予測困難となる為改竄の防止にも有効である。ただし暗号化したアプリケーションプログラムを既存のコンピュータで実行することはできないので、これを復号化しつつ実行するようなマイクロプロセッサが必要となる。このマイクロプロセッサは、OSがアプリケーションプログラムに対して敵対動作をとることを前提として、プログラムの秘密を守り、プログラムだけでなくプログラムが扱う情報やデータも暗号化して、解析や改竄から保護する機能を備える(以下、このようなマイクロプロセッサを「耐タンパマイクロプロセッサ」と記載)。又、この耐タンパマイクロプロセッサは保護を受けたプログラムを複数同時に疑似並列実行するマルチタスクのプログラム実行環境を提供する(例えば特許文献1及び非特許文献1参照。)

10

## 【0005】

耐タンパマイクロプロセッサがキャッシュメモリを搭載する場合、暗号化や復号化を行なう暗号処理部は、プロセッサコアとキャッシュメモリの間若しくはキャッシュメモリとメインメモリ等の記憶装置の間に配置することが可能である。暗号処理部をキャッシュメモリとメインメモリの間に配置すると、キャッシュメモリには復号化後、又は暗号化前の平文の内容が格納される。このため前者よりも後者の配置手法が、暗号化・復号化の処理回数が少なく済むため、効率的である。

## 【0006】

暗号処理部が暗号化や復号化を行なう際、マルチタスクのプログラム実行環境であると、キャッシュメモリ内には複数のプログラムやそれらのデータが格納されている。この時、第三者によるOS等の改竄により、キャッシュメモリ内のプログラム間において、他のプログラムの秘密情報等の盗聴、改竄等が発生することがある。これを防止する為、キャッシュメモリに対するアクセスの制限が必要となる。

20

## 【0007】

従来提案されてきた耐タンパマイクロプロセッサでは、同時に動作する各プログラムに対して1つずつタスクIDを与え、これをキャッシュメモリに対するアクセス制限に用いる。キャッシュメモリの各キャッシュラインにはタスクIDを格納するためのメモリとして秘密保護フィールドが用意される。プロセッサコアがキャッシュメモリに平文の実行コードやデータを格納するときは、対応する秘密保護フィールドに現在実行中のプログラムのタスクIDを格納する。プロセッサコアがキャッシュメモリの内容を読み出す際は、読み出そうとしているキャッシュラインの秘密保護フィールドからタスクIDを取得する。そのタスクIDと、現在実行中のプログラムのタスクIDとを比較し、これらが一致する場合のみ読み出しを許可する。キャッシュメモリに格納されたデータをメインメモリ等の記憶装置に書き出すときには暗号化を行なう必要があるが、その際に用いる暗号鍵は、現在実行中のプログラムが保持する暗号鍵であるとは限らない。従来提案されてきた耐タンパマイクロプロセッサでは、キャッシュラインの秘密保護フィールドに格納されたタスクIDにより暗号鍵を取得する。このためタスクIDと暗号鍵の対応関係を格納した鍵値テーブルをプロセッサ内部に保持する。又、複数のプログラムが同時に協調動作をするとき、これらのプログラムの間で他のプログラムからは読み出せないようにデータを共有する機能がある。この機能を実現するにはプログラムどうして1つの暗号鍵の値を共有する。各々のプログラムがこの1つの暗号鍵をデータを読み書きするために使用すれば、これらのプログラムどうして共有されるメモリ領域の内容を共有することができ、この暗号鍵を知らない他のプログラムは、同じメモリ領域の内容を正しく暗号化・復号化して読み書きすることができない(例えば特許文献2参照。)

30

40

## 【0008】

## 【特許文献1】

特開2001-230770号公開公報

## 【0009】

## 【特許文献2】

50

特開 2002 - 202720 号 公開公報

【0010】

【非特許文献 1】

David Lie、Chandramohan Thekkath、Mark Mitchell、Patrick Lincoln、Dan Boneh、John Mitchell、Mark Horowitz 著「Architectural Support for Copy and Tamper Resistant Software」、ACM 発行「ASPLOS-IX Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, USA, November 12-15, 2000」、2000 年 11 月、p. 168 - 177

【0011】

【発明が解決しようとする課題】

しかしながら、上記のタスク ID を使用してキャッシュメモリに対するアクセスの制限をする方式には、以下に述べるような問題点が存在する。

【0012】

先ず、第 1 の問題点として、この方式の耐タンパマイクロプロセッサはタスク ID と暗号鍵の対応関係を格納した鍵値テーブルを保持するが、この鍵値テーブルの大きさにより同時起動可能なプログラムの数が制限される。

【0013】

第 2 の問題点として、あるタスク ID のプログラムを終了させてそのタスク ID を新たに起動する別のプログラムに割り当てるとき、前に実行されていたプログラムのデータがキャッシュメモリに残っていると、同じタスク ID を割り当てられた次のプログラムは暗号鍵を知らなくてもそのデータの読み出しが可能となる。これを回避するには、プログラム終了時にキャッシュメモリの全領域を走査し、終了したプログラムのタスク ID に対応する各キャッシュラインを無効化する必要があるが、この操作には時間が長くかかった。特に大きなキャッシュメモリを搭載するマイクロプロセッサではそれが顕著であった。

【0014】

第 3 の問題点として、複数のプログラムが同時に協調動作する際にメモリの内容を共有する場合、あるプログラムが書き込んだデータを別のプログラムが読み出すには、最初のプログラムが書き込んだデータをいったんキャッシュメモリからメインメモリ等の記憶装置に暗号化して書き出し、これを再度復号化してキャッシュメモリに読み込む必要がある。つまり、本来同一内容のデータであるにもかかわらず、それを一度暗号化し、再び同じ鍵で復号化するため、実行速度が低下する。

【0015】

最後に、第 4 の問題点として、タスク ID を使用してキャッシュメモリに対するアクセスの制限をする方式では鍵値テーブルを参照する機能及びプログラム終了時にキャッシュ領域を走査し無効化する機能等、多くの機能をハードウェアで実装する必要があり、マイクロプロセッサの構造が複雑化する。

【0016】

本発明はこれらの問題点を解決する為のものであり、キャッシュメモリに対するアクセスの制限方法において、同時起動可能なプログラムの数に制限を与えず、1つのプログラムが終了したときにそのプログラムが使用したキャッシュメモリの内容をキャッシュメモリの全領域を走査することなく他のプログラムから読み出し不可能とし、複数のプログラムが1つの暗号鍵を共有して共有メモリ領域を読み書きする際に、あるプログラムがキャッシュに書き込んだ内容を、それを暗号化して主メモリに書き出さずに他のプログラムから読み出すことができ、複雑なハードウェア機能が不要な耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法を提供することを目的とする。

【0017】

【課題を解決するための手段】

上記の問題点を鑑み、本発明の第 1 の特徴は、(イ)暗号化されたプログラムの実行コ

10

20

30

40

50

ードの読み出し要求を処理するキャッシュメモリ制御手段と、(ロ)キャッシュメモリ制御手段からの復号化要求に基づき、実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、(ハ)複数のキャッシュラインを有し、復号化手段にて復号化した実行コードを記憶するデータ領域と、実行コードの復号化に用いた暗号鍵を格納する秘密保護属性保持部とを、同一キャッシュライン上に有するキャッシュメモリと、(ニ)暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタとを具備し、(ホ)キャッシュメモリ制御手段は、実行コードの読み出し要求を処理する際に、キャッシュメモリ内に実行コードが存在するときは、鍵値レジスタに記憶されている実行中のプログラムの暗号鍵とキャッシュメモリ内の実行コードが格納されたキャッシュラインの秘密保護属性保持部に格納された暗号鍵とを比較することによって、キャッシュメモリ内の実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断し、キャッシュメモリ内の実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、復号化手段へ復号化要求を出す代わりに、キャッシュメモリ内の実行コードを使用するよう制御する耐タンパマイクロプロセッサであることを要旨とする。

10

## 【0018】

上記の発明によると、キャッシュメモリに対するアクセス制限が無い為、同時起動可能なプログラムの数に制限がない。又、1つのプログラムが終了したときにそのプログラムが使用していたキャッシュメモリの内容は、そのプログラムの暗号鍵を知らない限り読み出すことができない。その為、キャッシュメモリを無効化するためにキャッシュメモリ全領域を走査する必要が無い。さらに、アクセス制限を行う機能や、キャッシュメモリ全領域を走査する機能が不要であるため、ハードウェアの複雑化を防ぐことができる。

20

## 【0019】

本発明の第2の特徴は、(イ)暗号化されたプログラムのデータの読み出し要求を処理するキャッシュメモリ制御手段と、(ロ)キャッシュメモリ制御手段からの復号化要求に基づき、データを記憶装置より読み出し、所定の暗号鍵によって復号化する復号化手段と、(ハ)複数のキャッシュラインを有し、復号化手段にて復号化したデータを記憶するデータ領域と、データの復号化に用いた暗号鍵を格納する秘密保護属性保持部とを、同一キャッシュライン上に有するキャッシュメモリと、(ニ)暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵が更新されて記憶される鍵値レジスタとを具備し、(ホ)キャッシュメモリ制御手段は、データの読み出し要求を処理する際に、キャッシュメモリ内にデータが存在するときは、鍵値レジスタに記憶されている実行中のプログラムの暗号鍵とキャッシュメモリ内のデータが格納されたキャッシュラインの秘密保護属性保持部に格納された暗号鍵とを比較することによって、キャッシュメモリ内のデータが所定の暗号鍵と同一の暗号鍵によって復号化されたデータである否かを判断し、キャッシュメモリ内のデータが所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、復号化手段へ復号化要求を出す代わりに、キャッシュメモリ内のデータを使用するよう制御する耐タンパマイクロプロセッサであることを要旨とする。

30

## 【0020】

本発明の第3の特徴は、(イ)暗号化されたプログラムの実行コードの読み出し要求を処理するステップと、(ロ)実行コードを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、(ハ)復号化された実行コードを、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられたデータ領域の1つに記憶するステップと、(ニ)復号化に用いた暗号鍵を、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられた秘密保護属性保持部のうち、実行コードが格納されたキャッシュラインに設けられた秘密保護属性保持部に格納するステップと、(ホ)暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップと、(ヘ)実行コードの読み出し要求を処理する際に、キャッシュメモリ内に実行コードが存在するときは、鍵値レジスタに記憶されている実行中のプログラムの暗号鍵とキャッシュメモリ内の実行コードが格納されたキャッシュラインの秘密保護属性保持部に格納された暗号鍵とを比

40

50

較することによって、キャッシュメモリ内の実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードである否かを判断し、キャッシュメモリ内の実行コードが所定の暗号鍵と同一の暗号鍵によって復号化された実行コードであった場合には、キャッシュメモリ内の実行コードを使用するよう制御するステップとを備えるキャッシュメモリ搭載プロセッサによるデータアクセス制御方法であることを要旨とする。

【0021】

本発明の第4の特徴は、(イ)暗号化されたプログラムのデータの読み出し要求を処理するステップと、(ロ)データを記憶装置より読み出し、所定の暗号鍵によって復号化するステップと、(ハ)復号化されたデータを、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられたデータ領域の1つに記憶するステップと、(ニ)復号化に用いた暗号鍵を、キャッシュメモリ内の複数のキャッシュラインにそれぞれ設けられた秘密保護属性保持部のうち、データが格納されたキャッシュラインに設けられた秘密保護属性保持部に格納するステップと、(ホ)暗号化されたプログラムの実行毎に、復号化に使用する暗号鍵を更新し、鍵値レジスタに記憶するステップと、(ヘ)データの読み出し要求を処理する際に、キャッシュメモリ内にデータが存在するときは、鍵値レジスタに記憶されている実行中のプログラムの暗号鍵とキャッシュメモリ内のデータが格納されたキャッシュラインの秘密保護属性保持部に格納された暗号鍵とを比較することによって、キャッシュメモリ内のデータが所定の暗号鍵と同一の暗号鍵によって復号化されたデータである否かを判断し、キャッシュメモリ内のデータが所定の暗号鍵と同一の暗号鍵によって復号化されたデータであった場合には、キャッシュメモリ内のデータを使用するよう制御するステップとを備えるキャッシュメモリ搭載プロセッサによるデータアクセス制御方法であることを要旨とする。

10

20

【0022】

【発明の実施の形態】

キャッシュメモリ制御部及び暗号化復号化機能を備え、プログラムの実行コード及び処理対象データの秘匿性の保護及び改変を防止する為の耐タンパマイクロプロセッサ及びデータアクセス制御方法の実施の形態について以下に説明する。尚、本発明は以下の実施の形態にとらわれず、その趣旨を脱しない範囲で種々変更して実施できることは勿論である。

【0023】

(耐タンパマイクロプロセッサ)

本発明の実施の形態に係る耐タンパマイクロプロセッサ100は、図1に示すように、プロセッサコア10、キャッシュメモリ制御部20、コードデータ暗号復号処理部30、鍵値レジスタ40及び外部バスインタフェース50等を備える。耐タンパマイクロプロセッサ100は、外部バスインタフェース50に接続されたバス70を介して記憶装置60と接続されている。

30

【0024】

プロセッサコア10は、キャッシュメモリ制御部20から渡された平分コード及び平文データの演算処理を行う。

【0025】

コードデータ暗号復号処理部30は、復号化手段であり、キャッシュメモリ制御部20と外部バスインタフェース50の間に配置され、バス70側から送信された暗号化済み実行コードやデータの復号を行うモジュールである。またコードデータ暗号復号処理部30は、キャッシュメモリ制御部20から受信したデータを暗号化しバス70側に出力する。

40

【0026】

鍵値レジスタ40は、コードデータ暗号復号処理部30が暗号化及び復号化の実行時に使う暗号鍵を格納する。1つのプログラムは図5のように、複数の実行コード及びデータから構成され、これら複数の実行コード及びデータは、鍵値レジスタ40内の1つの暗号鍵によって暗号化若しくは復号化される。なお、本実施の形態ではプログラムの持つ実行コード及びデータは同一の暗号鍵で暗号化されるものとするが、実行コードとデータを別々の鍵で暗号化するために、鍵値レジスタ40に実行コード用とデータ用の2つの暗号鍵を

50

保持できるようにしても良い。鍵値レジスタ40と図2の秘密保護属性保持部25a~25dは同じ鍵値を格納する為、これらの領域と同じ大きさであることが好ましい。鍵値レジスタ40の領域は、新たなプログラムを実行する際には新たなプログラムの鍵値に上書きされ、更新される。

#### 【0027】

記憶装置60は、プログラムのコンパイル後の実行コード及びデータを記憶する為の主メモリ、記憶装置等を指す。又記憶装置60は、バス70を介して図示しない通信制御装置を用い、外部の補助記憶装置若しくは外部バスとのデータのやり取りを行っても良い。尚、記憶装置60が記憶する実行コード及びデータは既に相手の暗号処理装置等によって暗号化された状態で受信したものを記憶している。

10

#### 【0028】

キャッシュメモリ制御部20は、キャッシュメモリ制御手段であり、実行コード及びデータに付随するアドレス情報や秘密保護属性等の情報の比較処理、演算処理等の制御を行う。キャッシュメモリ制御部20は、キャッシュメモリ21を内部に備え、キャッシュメモリ21は実行コード及びデータを平文コード及び平文データとして記憶する。

#### 【0029】

図2は、キャッシュメモリ21のデータ構造を示す。キャッシュメモリ21は複数のキャッシュライン22a、22b、22c、22d... (以下「キャッシュライン22a~22d」と記載)から構成される。キャッシュライン22a~22dは、タグ領域23a、23b、23c、23d... (以下「タグ領域23a~23d」と記載)、データ領域24a、24b、24c、24d... (以下「データ領域24a~24d」と記載)、秘密保護属性保持部25a、25b、25c、25d... (以下「秘密保護属性保持部25a~25d」と記載)及び制御情報保持部26a、26b、26c、26d... (以下「制御情報保持部26a~26d」と記載)等から構成される。タグ領域23a~23dは、実行コード及びデータが読み出されたアドレスに関する情報を記憶する領域である。データ領域24a~24dは、読み出された実行コード及びデータを記憶する領域である。秘密保護属性保持部25a~25dは、実行コード及びデータの復号化に使用した暗号鍵、データの暗号化に使用するべき暗号鍵を記憶する領域である。制御情報保持部26a~26dは、各キャッシュライン22a~22dが保持する実行コード及びデータの制御情報を記憶する領域である。

20

30

#### 【0030】

本発明の実施の形態に係る耐タンパマイクロプロセッサが、鍵値レジスタの値の初期設定、実行コードの読み出し、データの読み出し及びデータの書き出しを行う際の動作について以下に説明する。

#### 【0031】

(鍵値レジスタの値の初期設定)

記憶装置60にロードされたプログラムは、暗号化された実行コードによる実行を開始する前に、プログラム実行の初期設定として、暗号鍵を耐タンパマイクロプロセッサ100の鍵値レジスタ40に設定する。具体的には、暗号鍵を符合化したデータをパラメータとしてプロセッサコア10に与える。プロセッサコア10は、与えられた暗号鍵を鍵値レジスタ40に設定する。例えば、図5のように、プログラム1において実行コード若しくはデータ「A」「B」「C」を実行する前に、鍵値「X」を鍵値レジスタ40に設定し、プログラム2において実行コード若しくはデータ「D」「E」「F」を実行する前に、鍵値「Y」を鍵値レジスタ40に設定する。又は、鍵値が0の、暗号化されていない実行コードを、図5中の「A」の前に挿入し、その実行コードを実行する際の動作内容として鍵値レジスタ40を設定してもよい。

40

#### 【0032】

尚、暗号化されたプログラムの実行中に、割り込み等により実行が中断されて他のプログラムに制御が移る場合には、通常のコテキスト保存操作の一部として鍵値レジスタの値を保存する。また、中断されていたプログラムの実行を再開する際も、通常のコテキスト

50

ト復元操作の一部として鍵値レジスタの値を復元する。ここでコンテキストの暗号化を行なっても良い。鍵値レジスタをコンテキストとして保存することにより、別々の暗号鍵で暗号化された複数のプログラムが疑似並列的に実行されるマルチタスク環境が実現される。

【0033】

耐タンパマイクロプロセッサ100の電源投入時やリセットの直後、及びコンテキスト保存の直後（ただしコンテキスト保存の直後に別のコンテキストの復元が行なわれる場合は除く）の制御として、耐タンパマイクロプロセッサ100は鍵値レジスタ40に特別な値の設定する。これは例えば、値がゼロの暗号鍵、鍵値レジスタの1ビットをフラグとする等の手法がある。鍵値レジスタ40にこれらの特別な値が設定されているとき、耐タンパマイクロプロセッサ100は記憶装置60等から読み出した実行コードの復号化を行わず、読み出したものをそのまま実行する。このことによりプログラマ等は暗号化復号化を行うプログラムや、暗号化復号化を行うタイミング等を自由に設定することが可能となる。

10

【0034】

プロセッサコア10により鍵値レジスタ40に暗号鍵が設定された後、キャッシュメモリ制御部20は実行コード及びデータの読み出しを以下で述べる手順により実行する。

【0035】

（実行コード及びデータの読み出し）

本発明の実施の形態に係る耐タンパマイクロプロセッサ100が、暗号化されたプログラムの実行コード又はデータを記憶装置60又はキャッシュメモリ21より読み出す動作について、図3のフローチャートを参照し、以下に説明する。

20

【0036】

（a）まず、キャッシュメモリ制御部20は、ステップS101において、プロセッサコア10より実行コード又はデータ取得の要求を受け取ると、ステップS102にて、第1の確認として、この実行コード又はデータがキャッシュメモリ21上にあるかを判断する。

【0037】

（b）実行すべきアドレスの実行コード又はデータがキャッシュメモリ21上にないと判断された場合は、ステップS103にて、バス70及び外部バスインタフェース50を経由して記憶装置60からそのアドレスの実行コード又はデータを読み出す。

30

【0038】

（c）ステップS104では、コードデータ暗号復号処理部30が、記憶装置60より読み出された実行コード又はデータを、初期設定された鍵値レジスタ40の暗号鍵で復号化する。

【0039】

（d）ステップS105にて、この実行コード又はデータはキャッシュメモリ21に転送され、図2のキャッシュライン22a～22d上のデータ領域24a～24dのいずれかに格納される。この際、ステップS106にて、実行コード又はデータが読み出されたアドレスに関する情報がタグ領域23a～23dのいずれかに格納され、制御情報保持部26a～26dのいずれかの内容が更新される。また、実行コード又はデータの復号化に用いた鍵値レジスタ40の暗号鍵は秘密保護属性保持部25a～25dのいずれかに格納される。ステップS107にて、復号化された実行コード又はデータは実行のためプロセッサコアにも転送される。

40

【0040】

（e）一方、ステップS102において、第1の確認の際、実行すべきアドレスの実行コード又はデータがキャッシュメモリに存在すると判断された場合は、ステップS108において、キャッシュメモリ制御部20が、第2の確認を行う。即ち、キャッシュライン22a～22dに含まれる秘密保護属性保持部25a～25dの内、これから実行すべきアドレスの実行コード又はデータのキャッシュライン中の秘密保護属性保持部より暗号鍵を

50

取得する。更に、ステップS 1 0 9において、現在演算処理が実行されているプログラムが固有する暗号鍵を格納する鍵値レジスタ4 0より暗号鍵を取得する。ステップS 1 1 0にて、これらの暗号鍵を比較する。もし2つの暗号鍵が一致した場合、これから実行すべきアドレスの実行コード又はデータとして、キャッシュメモリ2 1内に既に存在する同一の実行コード又はデータの内容を使用することを許可される。その後ステップS 1 1 1にて、キャッシュライン2 2 a ~ 2 2 dのデータ領域2 4 a ~ 2 4 dにある実行コード又はデータ内に存在する同一の実行コード又はデータが再びプロセッサコア1 0に転送される。反対にステップS 1 1 0において、2つの暗号鍵が一致しない場合は、現在実行中のプログラムはキャッシュメモリの内容を使用することを許可されず、ステップS 1 0 3 ~ S 1 0 7の、実行すべきアドレスの実行コード又はデータがキャッシュメモリにない場合と同様の動作を行なう。又は、プログラムの実行を中断して異常終了を示す例外を発生しても良い。

10

**【0041】**

上記の耐タンパマイクロプロセッサ1 0 0の動作の具体例として、図5のプログラム1及びプログラム2を耐タンパマイクロプロセッサ1 0 0が実行する動作について説明する。プログラム1及びプログラム2を実行する際には、プログラム1の終了後にプログラム2を実行する場合（実施例1）、プログラム1の実行中にプログラム2が割り込んで実行する場合（実施例2）、同じプログラム内で実行コード若しくはデータを実行する場合（実施例3）、異なるプログラム間にて実行コード若しくはデータを実行する場合（実施例4）とが考えられる。以下、それぞれの実施例について詳述する。

20

**【0042】**

（実施例1）

先ず、通常の処理として、図5のプログラム1の終了後にプログラム2を実行する場合について図6を用いて説明する。

**【0043】**

（a）先ず、耐タンパマイクロプロセッサ1 0 0は、ステップS 3 0 1において、プログラム1の初期設定として、鍵値レジスタ4 0に「鍵値X」を格納する。この後、ステップS 3 0 2において、記憶装置6 0よりプログラム1の実行コード若しくはデータ「A」を呼び出し、「鍵値X」にて復号化し、図2のキャッシュライン2 2 aに格納する。この際、秘密保護属性保持部2 5 aには「鍵値X」が格納される。又、実行コード若しくはデータ「A」はプロセッサコア1 0にて演算処理される。プロセッサコア1 0は図示しないプログラムカウンタを1つずつ進め、これと同様の処理を実行コード若しくはデータ「B」「C」に対しても行う（ステップS 3 0 3 ~ S 3 0 4）。

30

**【0044】**

（b）プログラム1が完了すると、待機していたプログラム2の処理に入る。ステップS 3 0 5において、プログラム2の初期設定として、鍵値レジスタ4 0に「鍵値Y」を格納する。この後、ステップS 3 0 6において、記憶装置6 0よりプログラム2の実行コード若しくはデータ「D」を呼び出し、「鍵値Y」にて復号化し、図2のキャッシュライン2 2 dに格納する。この際、秘密保護属性保持部2 5 dには「鍵値Y」が格納される。又、実行コード若しくはデータ「D」はプロセッサコア1 0にて演算処理される。プロセッサコア1 0は図示しないプログラムカウンタを1つずつ進め、これと同様の処理を実行コード若しくはデータ「E」「F」に対しても行う（ステップS 3 0 7 ~ S 3 0 8）。

40

**【0045】**

実施例1によると、暗号化されたプログラム1及びプログラム2を、暗号鍵を用いて平文化することにより、悪意の第三者の盗聴及び改竄等より防止し、安全に実行することができる。

**【0046】**

（実施例2）

次に、図5のプログラム1の実行中にプログラム2が割り込んで実行する場合について図7を用いて説明する。

50

## 【 0 0 4 7 】

( a ) 先ず、耐タンパマイクロプロセッサ 1 0 0 は、ステップ S 4 0 1 において、プログラム 1 の初期設定として、鍵値レジスタ 4 0 に「鍵値 X」を格納する。この後、ステップ S 4 0 2 において、記憶装置 6 0 よりプログラム 1 の実行コード若しくはデータ「A」を呼び出し、「鍵値 X」にて復号化し、図 2 のキャッシュライン 2 2 a に格納する。この際、秘密保護属性保持部 2 5 a には「鍵値 X」が格納される。又、実行コード若しくはデータ「A」はプロセッサコア 1 0 にて演算処理される。プロセッサコア 1 0 は図示しないプログラムカウンタを 1 つずつ進め、これと同様の処理を実行コード若しくはデータ「B」に対しても行う(ステップ S 4 0 3 )。

## 【 0 0 4 8 】

( b ) 実行コード若しくはデータ「B」の処理最中若しくは終了後に、プログラム 2 の割り込みが発生する。ステップ S 4 0 4 において、キャッシュメモリ制御部 2 0 は、通常のコンテキスト保存として「鍵値 X」を保存する。

## 【 0 0 4 9 】

( c ) 割り込みしたプログラム 2 は、ステップ S 4 0 5 において、初期設定として、鍵値レジスタ 4 0 に「鍵値 Y」を格納する。この後、ステップ S 4 0 6 において、記憶装置 6 0 よりプログラム 2 の実行コード若しくはデータ「D」を呼び出し、「鍵値 Y」にて復号化し、図 2 のキャッシュライン 2 2 d に格納する。この際、秘密保護属性保持部 2 5 d には「鍵値 Y」が格納される。又、実行コード若しくはデータ「D」はプロセッサコア 1 0 にて演算処理される。プロセッサコア 1 0 は図示しないプログラムカウンタを 1 つずつ進め、これと同様の処理を実行コード若しくはデータ「E」「F」に対しても行う(ステップ S 4 0 7 ~ S 4 0 8 )。

## 【 0 0 5 0 】

( d ) プログラム 2 が終了すると、割り込みされたプログラム 1 は処理を再開する。ステップ S 4 0 9 として、保存されていた「鍵値 X」を鍵値レジスタ 4 0 に再設定し、一時中断されていたプログラム 1 の実行コード若しくはデータ「C」の処理を再開する。

## 【 0 0 5 1 】

実施例 2 によると、暗号化されたプログラム 1 及びプログラム 2 を復号するための暗号鍵をコンテキストとして保存することにより、複数のプログラムが擬似並列的に実行されるマルチタスク環境を実現することができる。

## 【 0 0 5 2 】

( 実施例 3 )

次に、図 5 のプログラム 1 内で実行コード若しくはデータ「A」「B」「C」「A」を実行する場合について図 8 を用いて説明する。尚、変数 CNT の初期値は 1 とする。

## 【 0 0 5 3 】

( a ) 先ず、耐タンパマイクロプロセッサ 1 0 0 は、ステップ S 5 0 1 において、プログラム 1 の初期設定として、鍵値レジスタ 4 0 に「鍵値 X」を格納する。この後、ステップ S 5 0 2 において、記憶装置 6 0 よりプログラム 1 の実行コード若しくはデータ「A」を呼び出し、「鍵値 X」にて復号化し、図 2 のキャッシュライン 2 2 a に格納する。この際、秘密保護属性保持部 2 5 a には「鍵値 X」が格納される。又、実行コード若しくはデータ「A」はプロセッサコア 1 0 にて演算処理される。プロセッサコア 1 0 は図示しないプログラムカウンタを 1 つずつ進め、これと同様の処理を実行コード若しくはデータ「B」「C」に対しても行う(ステップ S 5 0 3 ~ S 5 0 5 )。

## 【 0 0 5 4 】

( b ) 次に、ステップ S 5 0 6 においてカウンタが 1 つ進められ、ステップ S 5 0 7 として、再度 S 5 0 2 の処理が行われる。この時には、鍵値レジスタ 4 0 は「鍵値 X」を格納しており、実行コード若しくはデータ「A」の秘密保護属性保持部 2 5 a の値も「鍵値 X」であるため、キャッシュメモリ制御部 2 0 は、キャッシュメモリ 2 1 に存在する実行コード若しくはデータ「A」の使用を許可し、プロセッサコア 1 0 へ対しキャッシュメモリ 2 1 に存在する実行コード若しくはデータ「A」を再送信する。

10

20

30

40

50

## 【 0 0 5 5 】

(c) プログラム 1 が完了すると、ステップ S 5 0 8 の分岐命令によって、待機していたプログラム 2 の処理に入る。プログラム 2 の処理ステップ S 5 0 9 ~ S 5 1 2 についてはステップ S 3 0 5 ~ S 3 0 8 と同様である為説明を省略する。

## 【 0 0 5 6 】

実施例 3 によると、同一暗号鍵を有するプログラム、つまり同一プログラムにおいては、キャッシュメモリ内に既存する実行コード及びメモリを使用することができる。よって、演算処理効率が向上する。

## 【 0 0 5 7 】

(実施例 4)

最後に、図 5 のプログラム 1 及びプログラム 2 間で実行コード若しくはデータを実行する場合について図 9 を用いて説明する。

## 【 0 0 5 8 】

(a) 先ず、耐タンパマイクロプロセッサ 1 0 0 は、プログラム 1 の初期設定と実行コード又はデータ「A」「B」「C」に関する処理をステップ S 6 0 1 ~ S 6 0 4 にて行う。この処理はステップ S 3 0 1 ~ S 3 0 4 と同様である為説明を省略する。

## 【 0 0 5 9 】

(b) プログラム 1 が完了すると、待機していたプログラム 2 の処理に入る。プログラム 2 は、ステップ S 6 0 5 において、初期設定として、鍵値レジスタ 4 0 に「鍵値 Y」を格納する。この後、ステップ S 6 0 6 において、記憶装置 6 0 よりプログラム 2 の実行コード若しくはデータ「D」を呼び出し、「鍵値 Y」にて復号化し、図 2 のキャッシュライン 2 2 d に格納する。この際、秘密保護属性保持部 2 5 d には「鍵値 Y」が格納される。又、実行コード若しくはデータ「D」はプロセッサコア 1 0 にて演算処理される。プロセッサコア 1 0 は図示しないプログラムカウンタを 1 つずつ進める。

## 【 0 0 6 0 】

(c) ステップ S 6 0 7 において、次に処理を行う実行コード若しくはデータ「E」が、JUMP命令、GOTO命令等で実行コード若しくはデータ「C」に関わるものであるとする。この場合、キャッシュメモリ制御部 2 0 は、キャッシュメモリ 2 1 内に実行コード若しくはデータ「C」が存在するか否かを判断する。プログラム 1 の直後にプログラム 2 を実行した為、キャッシュメモリ 2 1 内の、例えば、キャッシュライン 2 2 c には実行コード若しくはデータ「C」が存在するとする。次にキャッシュメモリ制御部 2 0 は、鍵値レジスタ 4 0 の現在の値「鍵値 X」と、秘密保護属性保持部 2 5 c に格納されている「鍵値 Y」とを比較する。この場合、鍵値は一致しない為、ステップ S 6 0 8 にて、キャッシュメモリ制御部 2 0 は、キャッシュメモリ 2 1 内のキャッシュライン 2 2 c を使用することを許可しない。この後、実行コード若しくはデータ「C」をステップ S 1 0 3 ~ 1 0 7 の手順により取得したり、プログラム 2 の実行を中断し、異常終了を示したりする。

## 【 0 0 6 1 】

実施例 4 においては、異なる暗号鍵を有するプログラム間、つまり、複数の別々のプログラム間においては、キャッシュメモリ内の実行コード及びデータの使用を許可しない。このことによって、1 つのプログラムの終了時にキャッシュメモリ全領域を走査する必要がなくなり、ひいては、演算処理効率が向上する。

## 【 0 0 6 2 】

(データの書き出し)

本発明の実施の形態に係る耐タンパマイクロプロセッサ 1 0 0 がプログラムの要求により、データをキャッシュメモリ 2 1 及び記憶装置 6 0 に書き出す動作について、図 4 のフローチャートを参照し、以下に説明する。

## 【 0 0 6 3 】

(a) 先ずステップ S 2 0 1 では、キャッシュメモリ制御部 2 0 が、プロセッサコア 1 0 より、書き出すべきデータを平文のままキャッシュメモリ 2 1 内に取得し、図 2 のキャッシュライン 2 2 a ~ 2 2 d のデータ領域 2 4 a ~ 2 4 d の該当領域に格納する。

10

20

30

40

50

## 【0064】

(b)この際、ステップS202では、データを書き込むアドレスに関する情報をタグ領域23a~23dの該当領域に格納し、制御情報保持部26a~26dの該当領域の内容を更新する。また、このデータを記憶装置60に書き出す際に用いるべき暗号鍵が鍵値レジスタ40より読み出され、秘密保護属性保持部25a~25dの該当領域に格納される。

## 【0065】

(c)次に、データをキャッシュメモリ21から記憶装置60に書き出す。この動作は、キャッシュメモリ21の該当するキャッシュライン22a~22dに別のデータを格納する場合等に必要となる。詳細を説明すると、ステップS203で、キャッシュメモリ制御部20がキャッシュライン22a~22dのデータ領域24a~24dの該当領域より平文データを取得する。又、ステップS204で、秘密保護属性保持部25a~25dの該当領域より暗号鍵を取得する。これらの平文データ及び暗号鍵はコードデータ暗号復号処理部30に転送される。

10

## 【0066】

(d)ステップS205では、コードデータ暗号復号処理部30が暗号鍵を用いて、平文データを暗号化する。ステップS206で、暗号化済みのデータは外部バスインタフェース50及びバス70を経由して記憶装置60に書き出される。

## 【0067】

上記によると、耐タンパマイクロプロセッサ100において演算処理され、変形されたデータは、秘密保護属性保持部の暗号鍵を用いて暗号化される。この為、悪意の第三者に盗聴、改竄されることなく、安全な状態で記憶装置に渡され、データを所望する装置、ネットワーク等に送信することができる。

20

## 【0068】

本発明の実施の形態に係る耐タンパマイクロプロセッサによると、キャッシュメモリに対するアクセスの際に鍵値テーブルが不要であるため、第1の問題点である、同時起動可能なプログラムの数に制限を与えることを回避することができる。

## 【0069】

1つのプログラムが終了したときにそのプログラムが使用していたキャッシュメモリは、そのプログラムが保持していた暗号鍵の値を知らない限り読み出すことができない。そのため、このキャッシュメモリの内容を残したまま他のプログラムを起動しても、その内容が他のプログラムに読み出される恐れはなく、キャッシュメモリを無効化するために全領域を走査しなくても良い。よって、第2の問題点である全領域走査時間を短縮し、ひいては、プログラムの実行速度を上げることができる。

30

## 【0070】

複数のプログラムが1つの暗号鍵を共有している場合、どのプログラムも前記共有された暗号鍵を用いてキャッシュメモリの内容を参照することができ、1つのプログラムが書き込んだ内容を別のプログラムが読み出すために、その内容をいったん暗号化しメインメモリ等の記憶装置に書き出す必要はない。この為、第3の問題点である、暗号化復号化によるプログラムの実行速度の低下を防ぐことができる。

40

## 【0071】

この耐タンパマイクロプロセッサを用いた手法によると、鍵値テーブルを用いる手法と異なり、鍵値テーブルを参照する機能や、プログラム終了時にキャッシュ領域を走査し無効化する機能は不要であり、第4の問題点であるハードウェアの複雑化を回避することができる。ひいては製作コストを下げることができる。

## 【0072】

## 【発明の効果】

本発明によると、同時起動可能なプログラムの数に制限を与えず、1つのプログラムが終了したときにそのプログラムが使用したキャッシュメモリの内容をキャッシュメモリの全領域を走査することなく他のプログラムから読み出し不可能とし、複数のプログラムが1

50

つの暗号鍵を共有して共有メモリ領域を読み書きする際に、あるプログラムがキャッシュに書き込んだ内容を、それを暗号化して主メモリに書き出さずに他のプログラムから読み出すことができ、複雑なハードウェア機能が不要な耐タンパマイクロプロセッサ及びキャッシュメモリ搭載プロセッサによるデータアクセス制御方法を提供することができる。

【図面の簡単な説明】

【図 1】本発明の実施の形態に係る耐タンパマイクロプロセッサの構成図である。

【図 2】本発明の実施の形態に係る耐タンパマイクロプロセッサのキャッシュメモリのデータ構成を示す図である。

【図 3】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの手順を示すフローチャートである。

10

【図 4】本発明の実施の形態に係る耐タンパマイクロプロセッサのデータの書き出しの手順を示すフローチャートである。

【図 5】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータ構造の概略を示す図である。

【図 6】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例 1 を示すフロー図である。

【図 7】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例 2 を示すフロー図である。

【図 8】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例 3 を示すフロー図である。

20

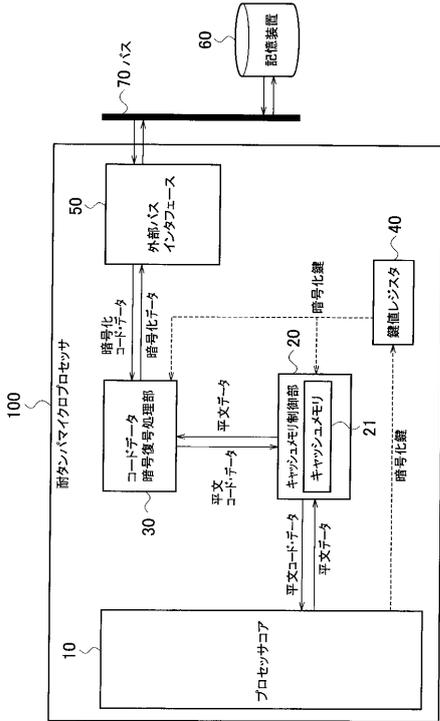
【図 9】本発明の実施の形態に係る耐タンパマイクロプロセッサの実行コード及びデータの読み出しの実施例 4 を示すフロー図である。

【符号の説明】

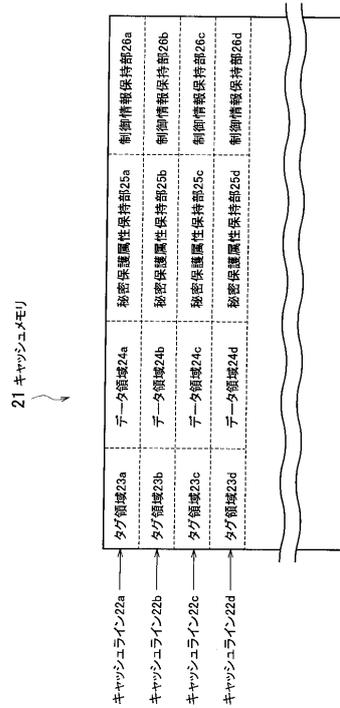
- 1 0 ... プロセッサコア
- 2 0 ... キャッシュメモリ制御部
- 2 1 ... キャッシュメモリ
- 2 2 a ~ 2 2 d ... キャッシュライン
- 2 3 a ~ 2 3 d ... タグ領域
- 2 4 a ~ 2 4 d ... データ領域
- 2 5 a ~ 2 5 d ... 秘密保護属性保持部
- 2 6 a ~ 2 6 d ... 制御情報保持部
- 3 0 ... コードデータ暗号復号処理部
- 4 0 ... 鍵値レジスタ
- 5 0 ... 外部バスインタフェース
- 6 0 ... 記憶装置
- 7 0 ... バス
- 1 0 0 ... 耐タンパマイクロプロセッサ

30

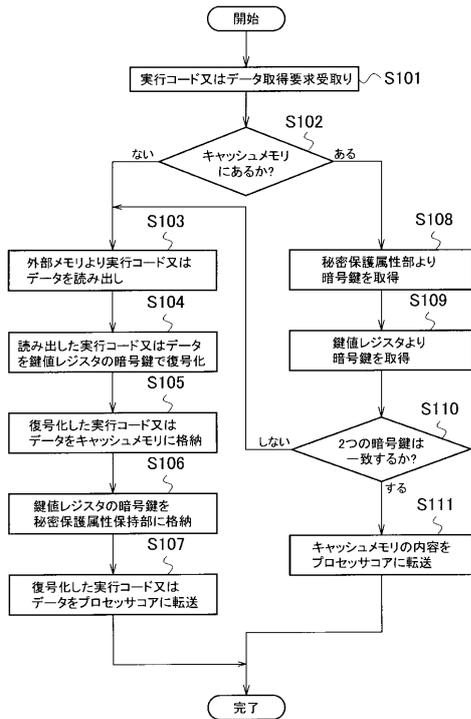
【 図 1 】



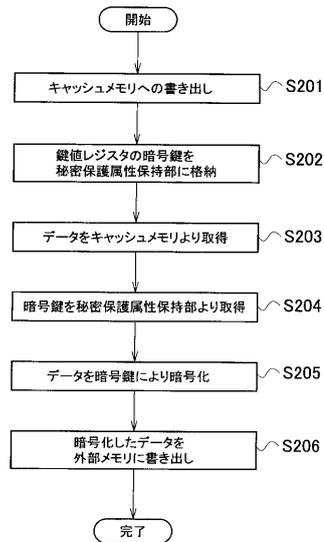
【 図 2 】



【 図 3 】



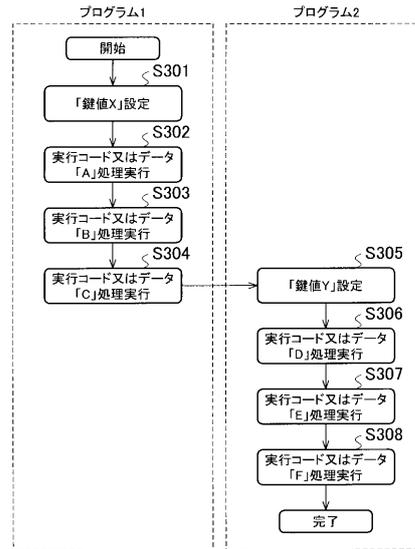
【 図 4 】



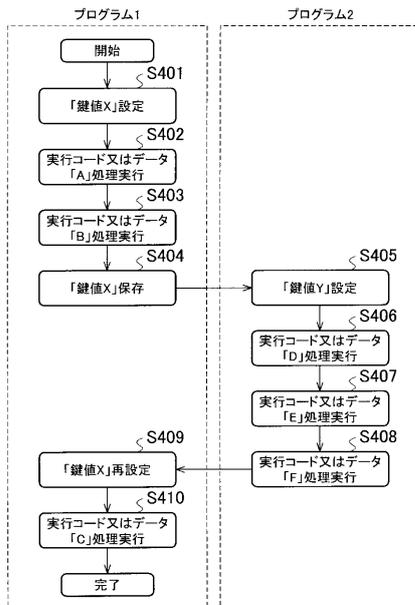
【 図 5 】

プログラム	実行コード又はデータ	鍵
1	A	X
	B	X
	C	X
2	D	Y
	E	Y
	F	Y

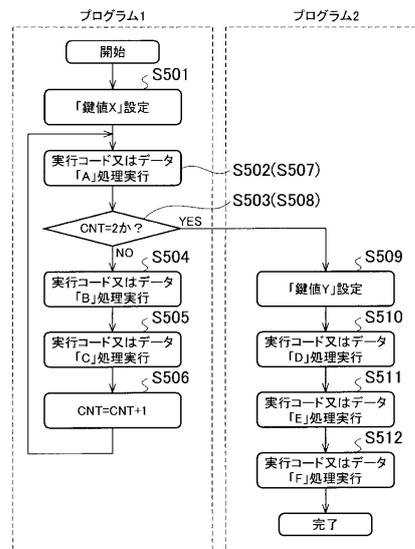
【 図 6 】



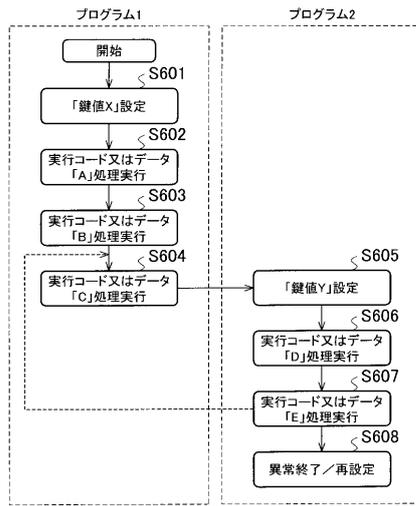
【 図 7 】



【 図 8 】



【 図 9 】



---

フロントページの続き

(74)代理人 100098327

弁理士 高松 俊雄

(72)発明者 山口 健作

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝 研究開発センター内

(72)発明者 橋本 幹生

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝 研究開発センター内

審査官 中里 裕正

(56)参考文献 特開2002-232417(JP,A)

(58)調査した分野(Int.Cl.,DB名)

H04L 9/10

G06F 12/08

G06F 21/22

G06F 21/24

JSTPlus(JDream2)