

(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2012-14197

(P2012-14197A)

(43) 公開日 平成24年1月19日(2012.1.19)

(51) Int.Cl.		F I		テーマコード (参考)
G09C	1/00	(2006.01)	G09C	1/00
H04L	9/08	(2006.01)	H04L	9/00
			660D	5J104
			601A	

審査請求 有 請求項の数 9 O L (全 16 頁)

(21) 出願番号	特願2011-220461 (P2011-220461)	(71) 出願人	398055691
(22) 出願日	平成23年10月4日 (2011. 10. 4)		株式会社ハイパーテック
(62) 分割の表示	特願2004-97087 (P2004-97087)		京都府京都市下京区中堂寺南町134 京
	の分割		都リサーチパーク (財) 京都高度技術研
原出願日	平成16年3月29日 (2004. 3. 29)	(74) 代理人	100109553
			弁理士 工藤 一郎
		(72) 発明者	小川 秀明
			京都府京都市下京区中堂寺南町134 京
			都リサーチパーク (財) 京都高度技術研
			究所内 株式会社ハイパーテック内
		Fターム(参考)	5J104 AA16 AA32 AA41 EA04 EA08
			EA17 NA02 NA37 PA14

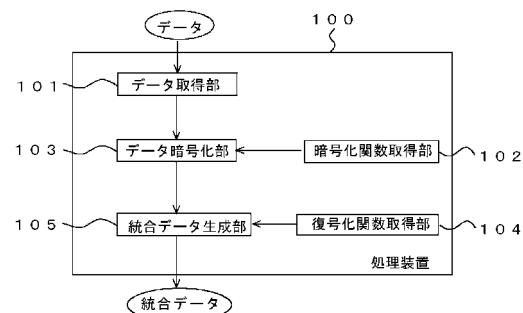
(54) 【発明の名称】 処理装置及び処理方法

(57) 【要約】

【課題】 暗号化データが容易に復号化されないように解析が行なわれ難いプログラムなどを生成する処理装置を提供する。

【解決手段】 データ取得部101で取得されたデータに対して、暗号化関数取得部102で取得された暗号化関数を、データ暗号化部103にて、適用して暗号化データを生成する。生成された暗号化データと、復号化関数取得部104で取得した復号化関数と、に基づいて、直接的には復号化関数を示す情報を含まない統合データを生成する。「直接的には復号化関数を示す情報を含まない」場合としては、例えば、復号化関数を暗号化して統合データに含ませる場合がある。

【選択図】 図1



【特許請求の範囲】**【請求項 1】**

データを取得するデータ取得部と、

前記データ取得部で取得したデータを暗号化するための関数である暗号化関数を取得する暗号化関数取得部と、

前記暗号化関数取得部で取得された暗号化関数により、前記データ取得部で取得されたデータを暗号化して暗号化データを生成するデータ暗号化部と、

前記データ暗号化部により生成される暗号化データを復号化するための復号化関数を取得する復号化関数取得部と、

前記暗号化部により生成された暗号化データと、前記復号化関数取得部で取得された復号化関数と、に基づいて、直接的には前記復号化関数を示す情報を含まない統合データを生成する統合データ生成部と、

を有し、

前記統合データ生成部で生成された統合データは、所定の操作を加えることにより前記復号化関数、及び、前記暗号化データを取得可能に構成されている処理装置。

【請求項 2】

前記暗号化関数取得部で取得される暗号化関数は、暗号化前のデータと暗号化後のデータとのサイズが等しい請求項 1 に記載の処理装置。

【請求項 3】

前記統合データ生成部は、前記復号化関数を暗号化する復号化関数暗号化手段と、

前記復号化関数暗号化手段により暗号化された前記復号化関数である暗号化復号化関数に基づいて前記統合データを生成する暗号化関数利用生成手段と、
を有する請求項 1 または 2 に記載の処理装置。

【請求項 4】

前記復号化関数と前記暗号化復号化関数とはサイズが等しい請求項 3 に記載の処理装置。

【請求項 5】

前記統合データはプログラムであり、前記所定の操作は、前記統合データを計算機において実行する操作である請求項 1 から 4 のいずれかに記載の処理装置。

【請求項 6】

前記所定の操作により得られる前記復号化関数、及び、前記暗号化データを前記復号化関数に適用して得られるデータは、前記計算機の主記憶にのみ保持される請求項 5 に記載の処理装置。

【請求項 7】

前記データ取得部で取得されるデータはプログラムであり、前記統合データは、前記復号化関数、及び、前記暗号化データを取得する取得ステップと、前記暗号化データを前記復号化関数に適用してデータを取得するデータ取得ステップと、前記データ取得ステップにより取得されたデータを実行する実行ステップと、を計算機に実行させるためのプログラムである請求項 5 または 6 に記載の処理装置。

【請求項 8】

前記復号化関数は、統合データが実行される計算機内にデバッガが存在するかどうかを確認する機能を有する請求項 6 または 7 に記載の処理装置。

【請求項 9】

データを取得するデータ取得ステップと、

前記データ取得ステップで取得したデータを暗号化するための関数である暗号化関数を取得する暗号化関数取得ステップと、

前記暗号化関数取得ステップで取得された暗号化関数により、前記データ取得ステップで取得されたデータを暗号化して暗号化データを生成するデータ暗号化ステップと、

前記データ暗号化ステップにより生成される暗号化データを復号するための復号化関数を取得する復号化関数取得ステップと、

前記データ暗号化ステップにより生成された暗号化データと前記復号化関数取得ステップにより取得された復号化関数とに基づいて直接的には前記復号化関数を示す情報を含まない統合データを生成する統合データ生成ステップと、

を含む処理方法であって、

前記統合データ生成ステップで生成された統合データは、所定の操作を加えることにより前記復号化関数、及び、前記暗号化データを取得可能に構成されている処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データを暗号化して含み、所定の操作を加えることにより復号化をすることが可能な統合データを生成する処理装置及び処理方法に関する。 10

【背景技術】

【0002】

データを暗号化して統合データに含ませ、データが必要となった場合にのみ、暗号化されたデータを符号化することは、コンテンツの配信やプログラムの配布において行なわれている。

【0003】

例えば、コンテンツの配信においては、著作権などの保護のためにコンテンツを暗号化して配信し、コンテンツを再生する間だけ暗号化されたコンテンツが復号化され、保持される。また、プログラムの配布においては、権原を有する者だけがそのプログラムを実行可能とするため、様々な保護手段が用いられるが、その保護手段を破られないようにするために、プログラムを暗号化して配布し、実行時に暗号化されたプログラムを復号化することが行なわれる。例えば、非特許文献1の第5章には、プログラムを暗号化し、実行時に復号化を行なってそのプログラムを実行するプログラムを生成するフリーウェアなどが紹介されている。 20

【0004】

図15は、上記のフリーウェアなどにより、データであるプログラムを暗号化した状態を例示する。生成されたプログラム1501の中に、復号化関数1502と暗号化データ1503とが含まれている。プログラム1501を実行すると、復号化関数1502を暗号化データ1503に適用して、暗号化データが復号化され、データが得られる。 30

【先行技術文献】

【非特許文献】

【0005】

【非特許文献1】Pavol ▲C▼erve▲n▼著、"CRACKPROOF YOUR SOFTWARE"、NO STARCH PRESS、米国、2002年

【発明の概要】

【発明が解決しようとする課題】

【0006】

図15に例示されたような形式でプログラムなどを暗号化した場合、例えば、復号化関数1502を逆アセンブルなどすることにより、復号化関数の動作が容易に解析されてしまう。これにより、暗号化データ部分を取り出して、解析した結果に基づいて復号化をすることにより、暗号化データが容易に復号化されてしまうという課題がある。 40

【0007】

そこで、本発明は、暗号化データが容易に復号化されないようなプログラムなどを生成する処理装置を提供することを目的とする。

【課題を解決するための手段】

【0008】

かかる課題を解決するために、本発明が提供する処理装置は、データを取得し、取得したデータを暗号化するための暗号化関数を取得し、その暗号化関数によりデータを暗号化して暗号化データを生成し、また、暗号化データを復号化するための復号化関数を取得し 50

、暗号化データと復号化関数に基づいて、直接的には復号化関数を示す情報を含まない統合データを生成する。なお、その統合データは、所定の操作を加えることにより、復号化関数と暗号化データとを取得可能に構成されているものとする。

【0009】

このような処理装置により、生成される統合データには、直接的には復号化関数が含まれていないので、復号化関数を解析されることがなく、課題が解決される。

【0010】

なお、取得したデータを暗号化する暗号化関数は、暗号化前と暗号化後のデータのサイズが等しくなるようなものであってもよい。これにより、統合データに含まれる暗号化データを復号化して得られるデータで置き換えることが可能となり、復号化して得られるデータがハードディスクなどに別途記録される可能性が少なくなる。

10

【0011】

また、復号化関数は、暗号化されるようになっていてもよい。

【0012】

また、復号化関数の暗号化は、暗号化前と暗号化後とでサイズが等しくなるようなものを用いてもよい。

【0013】

また、復号化関数、復号化で得られるデータは、計算機の主記憶にのみ保持されるようになっていてもよい。

【0014】

また、暗号化関数により暗号化されるデータはプログラムであってもよく、統合データは、復号化関数を取得して、復号化関数を暗号化データに適用し、得られたデータを実行するプログラムであってもよい。

20

【発明の効果】

【0015】

本発明により、生成される統合データには、直接的には復号化関数が含まれていないので、復号化関数を解析されることがなく、権原無く復号化がされることを防止することができる。

【図面の簡単な説明】

【0016】

【図1】本発明の実施形態1に係る処理装置の機能ブロック図

【図2】統合データと暗号化データと復号化関数との関係の一例図

【図3】本発明の実施形態1に係る処理装置の処理の流れ図

【図4】可逆的なインストラクションの一例図

【図5】可逆的なインストラクションで構成される暗号化関数と復号化関数との関係を示す図

30

【図6】発明の効果を説明する一例図

【図7】本発明の実施形態3に係る処理装置の機能ブロック図

【図8】統合データの構造の一例図

【図9】暗号化関数を復号化するための情報を含む統合データの構造の一例図

40

【図10】本発明の実施形態5に係る処理装置により生成された統合データの処理の流れ図

【図11】本発明の実施形態7に係る処理装置が生成する統合データの構造

【図12】プログラムである統合データの動作を説明する流れ図

【図13】ランニングラインを説明するための図

【図14】統合データの変化を説明するための図

【図15】従来の暗号化の形式の一例図

【発明を実施するための形態】

【0017】

以下、本発明を実施するために最良の形態について、実施形態として図を用いて説明す

50

る。なお、本発明はこれら実施形態に何ら限定されるものでなく、その要旨を逸脱しない範囲において、種々なる態様で実施し得る。

【0018】

(実施形態1：主に請求項1、9について説明する)

【0019】

本発明の実施形態1として、データを取得し、暗号化データを生成し、その暗号化データと、その暗号化データを復号化する関数と、に基づいて、直接的には復号化する関数を含まない統合データを生成する処理装置について説明する。

【0020】

(実施形態1：構成)

【0021】

図1は、本発明の実施形態1に係る処理装置の機能ブロック図を例示する。処理装置100は、データ取得部101と、暗号化関数取得部102と、データ暗号化部103と、復号化関数取得部104と、統合データ生成部105と、を有する。

【0022】

「データ取得部」101は、データを取得する。データは特定の種類や形式のものに限定されることはない。本発明に係る処理装置が、主にデジタルデータを扱う装置であれば、計算機にファイルとして格納できるデータであればよい。例えば、テキストデータ、プログラム（コンパイル済のものを含む）、静止画データ、音声データ、動画データなどがある。データの取得の態様には様々あり、例えば、ファイル名を指定したり、データそのものあるいは、BASE64などでテキストデータにエンコードしたデータを入力ウィンドウに入力したりする。また、処理装置に接続されたカメラやマイクなどにより、撮影などにより、データが得られるようになっていてもよい。また、処理装置内でのシミュレーションなどを行なった結果のデータを得るようになっていてもよい。この意味で、本明細書においては、「取得」には「生成」という意味も含まれるものとする。

【0023】

「暗号化関数取得部」102は、データ取得部101で取得したデータを暗号化するための関数である暗号化関数を取得する。本実施形態では、暗号化関数については特に限定はしない。いずれの暗号化関数であってもよい。なお、「暗号化関数」と述べているが、これは暗号化を行なうためのプログラム全般を意味するために「関数」という名称を使用している。プログラミング言語によっては、「手続き（procedure）」という名前である場合もあるが、手続きという言葉も本明細書の関数という言葉の意味に含まれるものとする。

【0024】

「データ暗号化部」103は、暗号化関数取得部102で取得された暗号化関数により、データ取得部101で取得されたデータを暗号化して暗号化データを生成する。すなわち、暗号化関数をデータに対して適用する部である。

【0025】

「復号化関数取得部」104は、データ暗号化部103により生成される暗号化データを復号化するための復号化関数を取得する。すなわち、暗号化関数取得部102で取得された暗号化関数と対になる復号化関数を取得する。「対になる」とは、暗号化関数で暗号化されたデータを復号化関数に適用することにより、暗号化関数を適用する前のデータが得られることをいう。例えば、暗号化関数が特定のビットパターンでビット毎にXOR（排他的論理和）を適用する場合には、復号化関数は同じビットパターンでXORを適用する関数である。また、暗号化関数が、公開鍵暗号方式の秘密鍵で暗号化を行なう関数であれば、復号化関数は、秘密鍵に対応する公開鍵で復号化を行なう関数となる。

【0026】

「統合データ生成部」105は、データ暗号化部103により生成された暗号化データと、復号化関数取得部104で取得された復号化関数と、に基づいて、直接的には前記復号化関数を示す情報を含まない統合データを生成する。

10

20

30

40

50

【0027】

「前記復号化関数」とは、復号化関数取得部104で取得された復号化関数を意味する。また、「直接的には前記復号化関数を示す情報を含まない」とは、復号化関数そのものを含まず、また、統合データのみによっては復号化関数を導出できないことを意味する。例えば、統合データが復号化関数やそれを示す情報を全く含まない場合がある。また、予め、復号化関数に識別子を付与しておき、統合データには、復号化関数に付与した識別子を含ませるようにしておき、復号化関数と識別子の関係は秘密に保持されるようになっていてもよい。

【0028】

本実施形態においては、統合データ生成部105で生成された統合データは、所定の操作を加えることにより、前記復号化関数、及び、前記暗号化データを取得可能に構成されている。

【0029】

図2は、統合データと暗号化データと復号化関数との関係を模式的に示す。例えば、統合データ200に、暗号化データ201と、復号化関数の識別子である復号化関数識別子202が含まれており、所定の操作を行なうことにより、暗号化データ201と、復号化関数203と、が得られるようになっている。

【0030】

「所定の操作」とは、予め定められた操作を意味する。例えば、予め定められたプログラムに統合データを入力することを意味する。そのプログラムは、統合データを読み取り、統合データより暗号化データを取得し、また、例えば、自身に含まれる復号化関数を取得する。あるいは、統合データに図2に例示されるように復号化関数識別子が含まれている場合には、復号化関数識別子で識別される復号化関数を取得する。例えば、インターネットなどの通信網により通信が可能なサーバとSSLなどでセキュアな通信路を確保して、復号化関数識別子をそのサーバに送信して、復号化関数を取得するようになっていてもよい。

【0031】

(実施形態1：処理)

【0032】

図3は、本実施形態に係る処理装置の処理の流れ図を例示する。ステップS301において、データ取得部101などにより、データを取得する(データ取得ステップ)。ステップS302において、暗号化関数取得部102などにより、暗号化関数を取得する(暗号化関数取得ステップ)。ステップS303において、データ暗号化部103などにより、暗号化データを生成する(データ暗号化ステップ)。ステップS304において、復号化関数取得部104などにより、復号化関数を取得する(復号化関数取得ステップ)。ステップS305において、統合データ生成部105などにより、統合データを生成する(統合データ生成ステップ)。

【0033】

以上の説明のように、本実施形態に係る処理装置は、データ取得ステップ、暗号化関数取得ステップ、データ暗号化ステップ、復号化関数取得ステップ、統合データ生成ステップを含む処理方法を使用するための装置と見することもできる。ただし、本実施形態に係る処理方法を使用する装置は、本実施形態に係る処理装置に限定されることはない。

【0034】

また、本実施形態に限らず、本発明に係る処理装置の構成要素である各部、各手段は、ハードウェア、ソフトウェア、ハードウェアとソフトウェア(プログラム)の両者、のいずれかによって構成することが可能である。たとえば、これらを実現する一例として、計算機を利用する場合には、CPU、メモリ、バス、インターフェース、周辺装置などから構成されるハードウェアと、これらのハードウェア上にて実行可能なソフトウェアを挙げることができる。

【0035】

(実施形態 1 : 主な効果)

【0036】

本実施形態によれば、統合データには、直接的には復号化関数を示す情報が含まれないので、統合データに含まれる暗号化データの復号化を解析（リバースエンジニアリング）することが困難となり、権原無く復号化がされることを防止することができる。

(実施形態 2 : 主に請求項 2 について説明する)

【0037】

本発明の実施形態 2 として、暗号化関数として、暗号化前と暗号化後とでデータのサイズが等しくなる処理装置について説明する。

【0038】

10

(実施形態 2 : 構成)

【0039】

本発明の実施形態 2 は、実施形態 1 において、暗号化関数取得部 102 で取得される暗号化関数は、暗号化前のデータと暗号化後のデータとのサイズが等しくなるようにした処理装置である。

【0040】

例えば、一バイトずつ、あるいは、一ビットずつ暗号化を行なうストリーム暗号が用いられてもよい。また、もし、データがブロック単位のサイズであれば、DES (Data Encryption Standard) のようなブロック単位で暗号化を行なう暗号化関数が用いられてもよい。

20

【0041】

ストリーム暗号の例としては、CPU の可逆的なインストラクションを用いるものがある。図 4 は、可逆的なインストラクションの例を、インストラクションのサイズ（バイト数）ごとにまとめたものである。「可逆的なインストラクション」とは、インストラクションによってデータを操作しても、元に戻すインストラクション（対となるインストラクション）が存在するようなインストラクションをいう。例えば、レジスタの値に 1 を加えるインストラクションは、レジスタの値から 1 を引くインストラクションが存在するので、可逆的なインストラクションである。

【0042】

そこで、暗号化関数取得部 102 は、図 4 に例示されたインストラクションをランダムに選び、また、インストラクションに Immediate があれば、その値を生成し、そのインストラクションを並べることにより、暗号化関数を生成するようになっていてもよい。そして、復号化関数取得部 104 は、暗号化関数を構成するインストラクションを逆に並べ、対となるインストラクションに置き換える。

30

【0043】

図 5 は、このように生成される暗号化関数と復号化関数との関係を例示する。暗号化関数の逆順にインストラクションを並べつつ、対となるインストラクションに置換を行なうことにより、復号化関数が生成されることが例示されている。

【0044】

統合データには、直接的には復号化関数を示す情報が含まれないが、例えば、暗号化関数のハッシュ値を復号化関数識別子として保持するようになっていてもよい。所定の操作を加える際には、ハッシュ値である復号化関数識別子に基づいて、暗号化関数のサーチを行なって、復号化関数を生成する。もちろん、全くランダムに暗号化関数を生成するとサーチの処理が困難となるので、暗号化関数のパターンをいくつか決めておき、サーチの計算量を減少するようになっていてもよい。

40

【0045】

(実施形態 2 : 主な効果)

【0046】

本実施形態では、実施形態 1 の効果に加えて、データと暗号化データのサイズが等しいので、統合データを生成する際には、データが保持される領域を暗号化データが保持され

50

る領域として使用でき、処理装置の作業スペースを増加させないようにすることができる。例えば、図 6 に例示されるように、まず、データ 602 を統合データ 601 に含ませておき、次にデータを暗号化することにより暗号化データ 603 が統合データに含まれるようにすることができる。

【0047】

(実施形態 3：主に請求項 3 について説明する)

【0048】

本発明の実施形態 3 として、復号化関数を暗号化して統合データに含ませる処理装置について説明する。

【0049】

(実施形態 3：構成)

【0050】

図 7 は、本発明の実施形態 3 に係る処理装置の機能ブロック図を例示する。処理装置 700 は、データ取得部 101 と、暗号化関数取得部 102 と、データ暗号化部 103 と、復号化関数取得部 104 と、統合データ生成部 105 と、を有し、統合データ生成部 105 は、復号化関数暗号化手段 701 と、暗号化関数利用生成手段 702 と、を有する。したがって、本実施形態に係る処理装置 700 は、実施形態 1 または 2 に係る処理装置の統合データ生成部 105 が、復号化関数暗号化手段 701 と、暗号化関数利用生成手段 702 と、を有する構成となっている。

【0051】

「復号化関数暗号化手段」701 は、前記復号化関数を暗号化する。例えば、図 4 と図 5 を用いて説明したような暗号化方法を使用して暗号化を行なう。

【0052】

「暗号化関数利用生成手段」702 は、復号化関数暗号化手段 701 により暗号化された前記復号化関数である暗号化復号化関数に基づいて、前記統合データを生成する。暗号化復号化関数は、直接的には復号化関数を示す情報を含まないので、統合データに暗号化復号化関数を含ませるようにする。

【0053】

図 8 は、本実施形態に係る処理装置で生成される統合データの構造を模式的に例示する。統合データ 801 の中に、暗号化復号化関数 802 と暗号化データ 803 とが含まれている。所定の操作を行なう場合には、統合データ 801 の中から暗号化復号化関数を取り出し、暗号化復号化関数を復号する関数を取得して、復号化関数を得る処理を行なうことになる。例えば、暗号化関数のパターンがある程度限定されており、また、復号化関数のパターンもある程度限定されていれば、暗号化復号化関数のパターンもある程度限定されるので、暗号化復号化関数より、復号を行なう関数を取得して、復号化関数を得ることが可能である。

【0054】

また、図 9 に示すように統合データ 901 に、暗号化関数 902 を復号化するための情報 901 を含ませておいてもよい。情報 901 は、例えば、暗号化復号化関数を復号化するための復号鍵であったり、暗号化復号化関数を復号するプログラムであってもよい。また、そのような復号鍵やプログラムを所定の方法で暗号化したものであってもよい。

【0055】

(実施形態 3：主な効果)

【0056】

本実施形態により、実施形態 1 または 2 の効果に加え、統合データに暗号化復号化関数を含ませることができるので、所定の処理において、暗号化データを復号化する処理を簡略化できる。

【0057】

(実施形態 4：主に請求項 4 について説明する)

【0058】

10

20

30

40

50

本発明の実施形態 4 として、復号化関数と暗号化復号化関数とのサイズが等しい処理装置について説明する。

【0059】

(実施形態 4：構成)

【0060】

本発明の実施形態 4 に係る処理装置は、実施形態 3 に係る処理装置を、復号化関数と暗号化復号化関数とは、サイズが等しくなるようにしたものである。実施形態 2 で説明したように、ストリーム暗号を用いたり、図 4 と図 5 とを用いて説明した暗号化方法を用いたり、復号化関数のサイズがブロック単位のサイズであれば、ブロック単位で暗号化を行なう暗号化関数を用いたりする。

10

【0061】

(実施形態 4：主な効果)

【0062】

本実施形態においては、実施形態 3 の効果に加え、復号化関数と暗号化復号化関数とのサイズが等しいので、所定の操作を行なう場合に、暗号化復号化関数の領域を復号化関数の領域として使用することができるで、作業スペースを増加させないようにすることができる。

【0063】

(実施形態 5：主に請求項 5 について説明する)

【0064】

20

本発明の実施形態 5 として、統合データがプログラムとなっている処理装置について説明する。

【0065】

(実施形態 5：構成)

【0066】

本発明の実施形態 5 に係る処理装置は、実施形態 1 から 4 のいずれかの実施形態に係る処理装置において、統合データがプログラムになるようにしたものであり、所定の操作とは、そのプログラムを実行することとしたものである。

【0067】

すなわち、統合データは実行することが可能であり、統合データの実行により、復号化関数と暗号化データとが取得され、復号化関数を暗号化データに適用することにより、データが得られることになる。

30

【0068】

もし、本実施形態に係る処理装置が実施形態 1 または 2 を基にしている場合には、プログラムである統合データは、例えば、インターネットなどの通信網に接続されたサーバより復号化関数を取得するようになっていてもよい。

【0069】

また、本実施形態に係る処理装置が実施形態 3 または 4 を基にしている場合には、プログラムである統合データは、自身に含まれる暗号化復号化関数を取得し、復号化することにより、復号化関数を得て、それを暗号化データに適用することになる。

40

【0070】

図 10 は、本実施形態に係る処理装置により生成された統合データからデータを得る処理の流れ図を例示する。ステップ S 1001 において、暗号化復号化関数から復号化する関数を用いて、復号化関数を得る。ステップ S 1002 において、復号化関数を用いて暗号化データよりデータを得る。

【0071】

(実施形態 5：主な効果)

【0072】

本実施形態により、実施形態 1 から 4 のいずれかの実施形態の主な効果に加えて、統合データから直接、データを得ることが可能となる。

50

【0073】

(実施形態6：主に請求項6について説明する)

【0074】

本発明の実施形態6として、所定の操作により得られる復号化関数と、データと、が計算機の主記憶にのみ保持されるようにした処理装置について説明する。

【0075】

(実施形態6：構成)

【0076】

本発明の実施形態6に係る処理装置は、実施形態5の処理装置を、前記所定の操作により得られる前記復号化関数、及び、前記暗号化データを前記復号化関数に適用して得られるデータは、前記計算機の主記憶にのみ保持されるようにしたものである。「前記所定の操作」とは、統合データを実行することである。「前記復号化関数」とは、復号化関数取得部104で取得されたのと同じ復号化関数を意味する。「前記暗号化データ」とは、統合データが生成されるとき基となった暗号化データである。また、「前記計算機」とは、プログラムである統合データが実行される計算機を意味する。

【0077】

したがって、プログラムである統合データは、データを得るための作業用の領域としてファイルなどを使用せず、主記憶のメモリだけを用いて作業を行なうことになる。

【0078】

また、復号化関数は、プログラムである統合データを実行するプロセスのメモリが読めないような処理を行なってもよい。通常の計算機のオペレーティングシステムでは、他のプロセスのメモリが読めないようになっているが、デバッガ用を実現するためなどの特殊なシステムコールなどを使用することにより、他のプロセスのメモリを読むことができる。そこで、特殊なデバイスドライバをオペレーティングシステムに組み込み、他のプロセスのメモリを読むシステムコールをフック(hook)し、特定のプロセスのメモリを読めないようにする。そして、復号化関数は、その特殊なデバイスドライバに対して設定を行ない(例えば、ioctlシステムコールを用いて設定する)、自身のプロセスをその特定のプロセスとして設定する。これにより、デバッガなどを用いても、復号化関数やデータが他人に取得されないようにすることができる。

【0079】

(実施形態6：主な効果)

【0080】

本実施形態により、実施形態5の主な効果に加えて、復号化関数と、データと、が主記憶にのみ保持されるので、復号化関数やデータを入手することが困難となる。

【0081】

(実施形態7：主に請求項7について説明する)

【0082】

本発明の実施形態7として、プログラムをデータとして取得し、統合データを実行することによりそのプログラムの実行が開始される処理装置について説明する。

【0083】

(実施形態7：構成)

【0084】

本発明の実施形態7に係る処理装置は、実施形態5または6の処理装置において、データ取得部101で取得されるデータはプログラムである。また、前記統合データは、取得ステップと、データ取得ステップと、実行ステップと、を計算機に実行させるためのプログラムである。ここに、「前記統合データ」とは、統合データ生成部105で生成される統合データである。また、「取得ステップ」とは、前記復号化関数、及び、前記暗号化データを取得するステップである。「前記復号化関数」と「前記暗号化データ」は、前記統合データを生成する基となった復号化関数と暗号化データである。「データ取得ステップ」は、前記暗号化データを前記復号化関数に適用してデータを取得するステップである。

「実行ステップ」とは、データ取得ステップで取得されたデータを実行するステップである。

【0085】

取得ステップは、例えば、復号化関数を取得し、メモリなどに保持し、また、統合データに含まれる暗号化データの位置や大きさなどを特定する。

【0086】

データ取得ステップは、取得ステップで取得された復号化関数に対して、暗号化データの位置と大きさなどを与えて、暗号化データを復号化してデータを得る。得られたデータは、メモリなどに保持される。

【0087】

実行ステップは、データ取得ステップで得られたデータがメモリ中にあれば、プログラムカウンタを、エントリポイントのアドレスに設定などする。また、データがファイルにあれば、そのファイルを実行するためのシステムコール（例えば、`execle`）などを実行する。

【0088】

なお、本実施形態において、データ取得部でプログラムであるデータが取得された場合には、シンボルテーブルをクリアしたり、共有ライブラリのインポートテーブルをクリアするなどの操作を行なってもよい。例えば、シンボルテーブルがクリアされることにより、万一データが他人に取得されたとしても、逆アセンブルなどのプログラムの解析を行なっても関数名などが表示されないで解析が困難となる。また、インポートテーブルのクリアにより、プログラムがどのようなライブラリを使っているかを調べるのが困難となり、どのような形でプログラムが書かれているかを知られにくくすることができる。

【0089】

また、暗号化復号化関数を復号化する際には、トラップハンドラを用いたランニングラインと呼ばれる手法を用いてもよい。ランニングラインとは、CPUがインストラクションを一つ実行するごとにCPUに割り込みをかけてトラップハンドラを呼び、次に実行するインストラクションのアドレスを決定することを繰り返すことである（図13参照）。これにより、インストラクションの間に無意味なインストラクションの列を挿入することができ、プログラムの解析を困難なものとする。ランニングラインにおいては、トラップハンドラの解析がされるとプログラムの解析がされるので、トラップハンドラを予め暗号化しておき、実行時に復号化するのが好ましい。

【0090】

図11は、本実施形態に係る処理装置が生成する統合データの構造を模式的に例示する。統合データ1101には、トラップハンドラの復号化関数が含まれており、これにより、暗号化されたトラップハンドラ1103が復号できるようになっている。また、暗号化復号化関数を復号化する関数1104も含まれており、これにより暗号化復号化関数1105が復号されるが、このときに、ランニングラインを用いてもよい。なお、暗号化復号化関数を復号化する関数1104の実行の全体でランニングラインを行なってもよいが、ランニングラインではトラップハンドラが一インストラクションごとに呼ばれて処理速度が低下する。そこで、暗号化復号化関数を復号化する関数1104の一部でランニングラインを実行するようになっていてもよい。例えば、暗号化復号化関数を復号化するための復号鍵や図5に例示された復号化関数が暗号化されているときに、暗号化された復号鍵などを復号化するときだけにランニングラインを用いるようになっていてもよい。

【0091】

図12は、プログラムである統合データの動作を説明する流れ図である。ステップS1201において、暗号化されたトラップハンドラを復号化する。例えば、特定のビットパターンにより、ビット毎にXORを計算することにより復号化を行なう。ステップS1202において、トラップハンドラをセットする。「セットする」とは、所定の条件が成就した場合に、トラップハンドラが呼ばれるようにすることである。ランニングラインにおいては、所定の条件とは、一インストラクションが実行されることである。ステップS1

10

20

30

40

50

203において、復号化する関数を用いて暗号化復号化関数より復号化関数を得る。ステップS1204において、トラップハンドラをリセットする。ステップS1205において、復号化関数を用いて暗号化データを復号化する。ステップS1206において、データを実行する。

【0092】

図14は、所定の操作を行なうことにより発生する図11に例示された統合データの変化を例示する。まず、ステップS1201において、暗号化されたトラップハンドラが復号化され、図14の上に例示される統合データが得られる。次に、ステップS1203により、暗号化復号化関数が復号化され、図14の中央に例示される統合データが得られる。ステップS1205により、暗号化データが復号化され、図14の下に例示される統合データが得られる。

10

【0093】

(実施形態7：主な効果)

【0094】

本実施形態により、実施形態5または6の主な効果に加えて、プログラムを暗号化して統合データに含ませることにより、解析が困難な状態でプログラムを安全に配布することができる。

【0095】

(実施形態8：主に請求項8について説明する)

【0096】

本発明の実施形態8として、復号化関数がデバッガの存在を確認する機能を有した処理装置について説明する。

20

【0097】

(実施形態8：構成)

【0098】

本発明の実施形態8に係る処理装置は、実施形態6または7に係る処理装置において、復号化関数が、統合データの実行される計算機内にデバッガが存在するかどうかを確認する機能を有するようにしたものである。デバッガが計算機に存在することにより、統合データの実行が解析されることを防止する目的を持つ。

【0099】

デバッガを検出する方法としては、デバッガが通常インストールされているディレクトリやフォルダにデバッガのコマンド名のファイルが存在するかどうかで検出する方法がある。また、オペレーティングシステムによっては、レジストリと呼ばれる場所にデバッガの情報が記録されているので、その記録の有無を調べてもよい。また、Softwareというデバッガが存在すると、イベントハンドラのアドレスの差が所定の値になったり、特殊なデバイスが存在するので、それを検出してもよい。Softwareの様々な検出方法については、非特許文献1の7章などに説明されている。

30

【0100】

もし、デバッガが存在することが検出されれば、暗号化データの復号化は行なわないようにすることができる。

40

【0101】

(実施形態8：主な効果)

【0102】

本実施形態においては、実施形態6または7の主な効果に加えて、デバッガなどの存在する場合には、暗号化データの復号化を行なわないようにすることができるので、解析がされにくくすることができる。

【産業上の利用可能性】

【0103】

本発明に係る処理装置及び処理方法により生成される統合データには、直接的には復号化関数が含まれていないので、復号化関数を解析されることがなく、権原無く復号化がさ

50

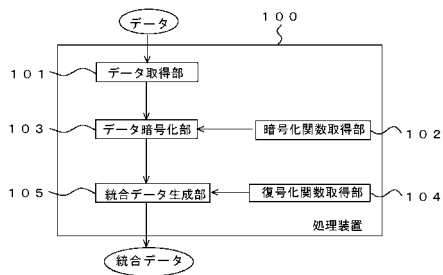
れることを防止することができるので、有用である。

【符号の説明】

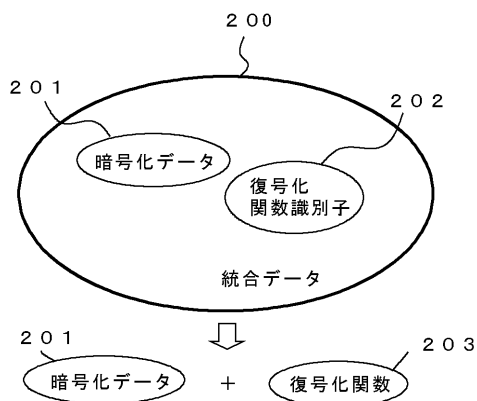
【0104】

- 100 処理装置
- 101 データ取得部
- 102 暗号化関数取得部
- 103 データ暗号化部
- 104 復号化関数取得部
- 105 統合データ生成部

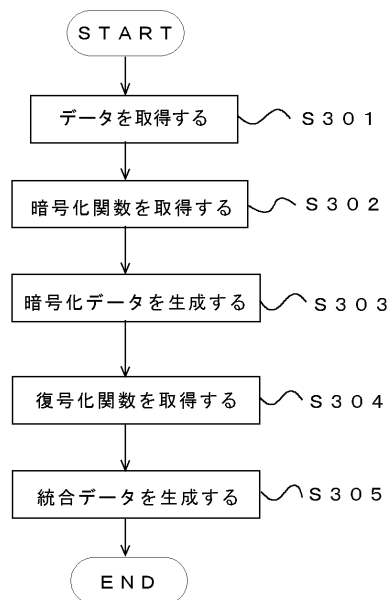
【図1】



【図2】



【図3】



【図 4】

1 バイトオペレーション

オペレーション	逆オペレーション
NOP	NOP
STC	CLC
CLC	STC

401

2 バイトオペレーション

オペレーション	逆オペレーション
INC AL	DEC AL
ADD AL, Immediate	SUB AL, Immediate
SUB AL, Immediate	ADD AL, Immediate
ADD AL, CL	SUB AL, CL
SUB AL, CL	ADD AL, CL

402

3 バイトオペレーション

オペレーション	逆オペレーション
ROR AL, Immediate	ROL AL, Immediate
ROL AL, Immediate	ROR AL, Immediate

403

【図 5】

暗号化関数

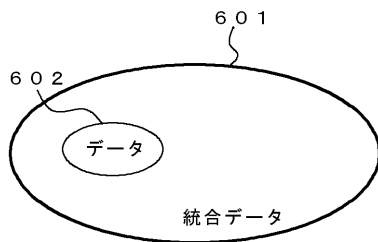
INC AL
ADD AL, 032h
STC
...
ROL AL, 013h
NOP
SUB AL, 0dah

復号化関数

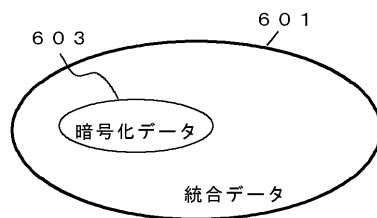
ADD AL, 0dah
NOP
ROR AL, 013h
...
CLC
SUB AL, 032h
DEC AL

【図 6】

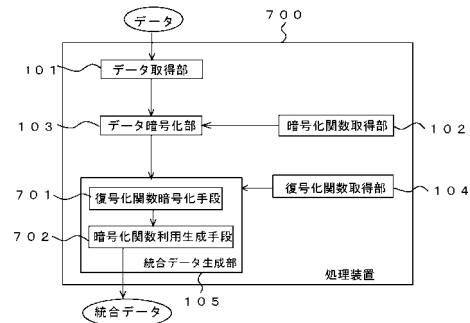
(1)



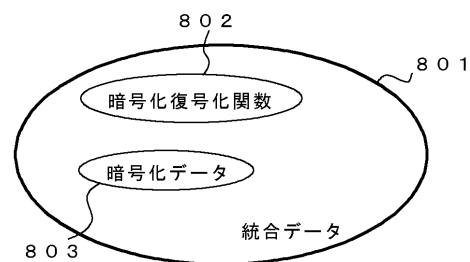
(2)



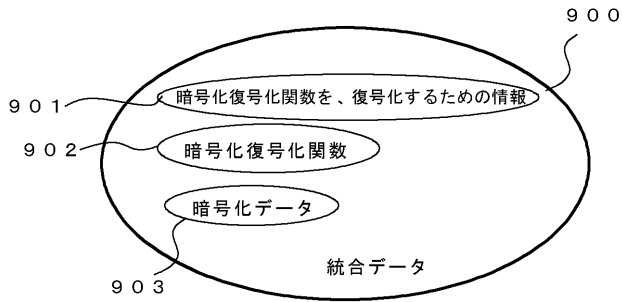
【図 7】



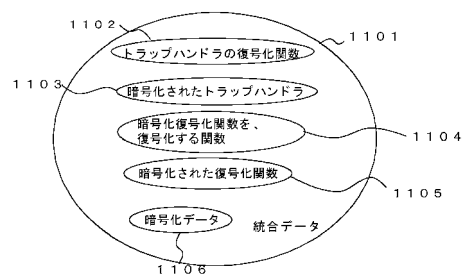
【図 8】



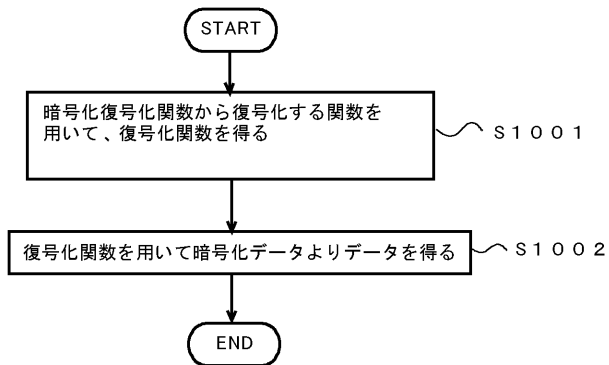
【図 9】



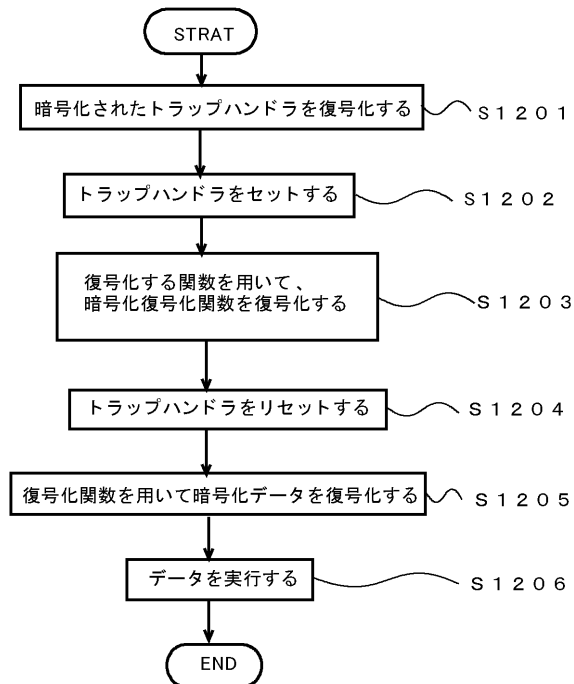
【図 11】



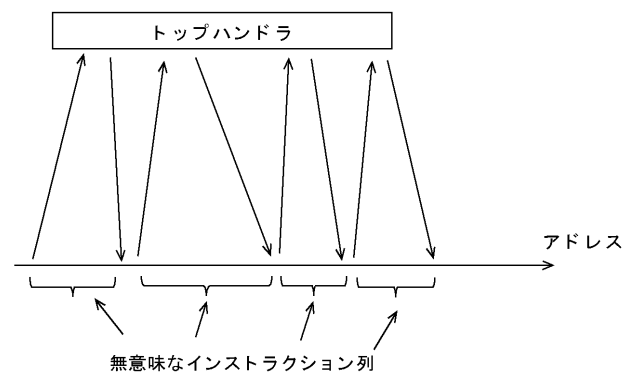
【図 10】



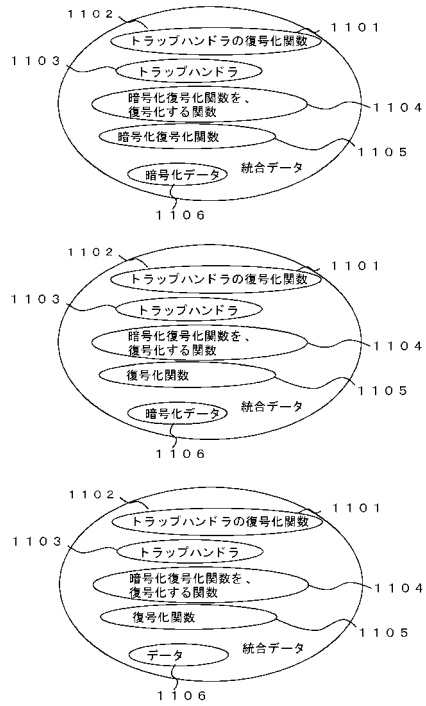
【図 12】



【図 13】



【図 14】



【図 15】

