(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6675227号
(P6675227)

(45)発行日　令和2年4月1日(2020.4.1)　(24)登録日　令和2年3月12日(2020.3.12)

(51)Int.Cl.		F I		
H04L 9/32	(2006.01)	H04L	9/00	675A
G06F 21/12	(2013.01)	H04L	9/00	675D
G06F 21/64	(2013.01)	G06F	21/12	
		G06F	21/64	

請求項の数 17　(全 28 頁)

(21)出願番号　特願2016-36312(P2016-36312)
(22)出願日　平成28年2月26日(2016.2.26)
(65)公開番号　特開2017-153044(P2017-153044A)
(43)公開日　平成29年8月31日(2017.8.31)
審査請求日　平成31年2月25日(2019.2.25)

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

最終頁に続く

(54)【発明の名称】情報処理装置、情報処理システム、情報処理方法、及びプログラム

(57)【特許請求の範囲】
【請求項１】
モジュール、データ、及び当該データについて生成されたハッシュ値である正解ハッシュ値を格納するメモリと、
前記メモリに格納されている前記モジュールのハッシュ値と、前記メモリに格納されている前記データのハッシュ値と、を算出する算出手段と、
前記算出手段が算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定する判定手段と、
前記算出手段が算出した前記モジュールのハッシュ値と、前記判定手段による判定結果を示す情報と、を前記メモリに格納されている前記モジュール及び前記データの完全性を検証するサーバに送信する送信手段と、
を備えることを特徴とする、情報処理装置。
【請求項２】
前記正解ハッシュ値は、完全性を有することが確認されたデータのハッシュ値であることを特徴とする、請求項１に記載の情報処理装置。
【請求項３】
前記モジュールが前記データを生成又は更新して前記メモリに格納したこと、又は前記モジュールが前記データを認証したことに応じて、前記算出手段は前記データのハッシュ値を算出して前記正解ハッシュ値として前記メモリに格納することを特徴とする、請求項１又は２に記載の情報処理装置。

【請求項４】
前記正解ハッシュ値は、前記モジュールの完全性が検証されない場合には更新されないことを特徴とする、請求項１乃至３の何れか１項に記載の情報処理装置。
【請求項５】
前記正解ハッシュ値は、前記算出手段が算出した前記モジュールのハッシュ値が所定値と異なる場合には更新されないことを特徴とする、請求項１乃至４の何れか１項に記載の情報処理装置。
【請求項６】
前記メモリは複数のデータと、それぞれのデータについての正解ハッシュ値とを格納し、
前記判定手段は、前記複数のデータのそれぞれについて、前記算出手段が算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定し、
前記送信手段は、前記複数のデータのそれぞれについての前記判定手段による判定結果を前記サーバに送信する
ことを特徴とする、請求項１乃至５の何れか１項に記載の情報処理装置。
【請求項７】
前記送信手段は、前記複数のデータのそれぞれについて、
前記データの識別子と、
前記算出手段が算出した前記データのハッシュ値と、
前記算出手段が算出した前記データのハッシュ値と前記判定手段による判定結果とから生成されたデータのハッシュ値と、
を前記サーバに送信することを特徴とする、請求項６に記載の情報処理装置。
【請求項８】
前記メモリが格納する複数のデータから、前記判定手段による完全性の検証対象となるデータを選択する選択手段をさらに備えることを特徴とする、請求項１乃至７の何れか１項に記載の情報処理装置。
【請求項９】
前記選択手段は、前記データへのアクセス権限を示す情報、前記モジュールの実行権限を示す情報、又は前記データの更新頻度を示す情報に従い、完全性の検証対象となるデータを選択することを特徴とする、請求項８に記載の情報処理装置。
【請求項１０】
前記送信手段は、前記算出手段が算出した前記モジュールのハッシュ値と、前記判定手段による判定結果を示す情報とを、耐タンパー性を有するメモリから読み出すことを特徴とする、請求項１乃至９の何れか１項に記載の情報処理装置。
【請求項１１】
前記送信手段は、
耐タンパー性がハードウェアにより実装された第１のメモリから、前記算出手段が算出した前記モジュールのハッシュ値を読み込み、
耐タンパー性がソフトウェアにより実装された第２のメモリから、前記データの少なくとも一部についての前記判定手段による判定結果を示す情報を読み込む
ことを特徴とする、請求項１乃至１０の何れか１項に記載の情報処理装置。
【請求項１２】
他の情報処理装置が有するモジュールのハッシュ値と、前記他の情報処理装置が有するデータの完全性を示す情報と、を受信する受信手段と、
前記受信したモジュールのハッシュ値と、他の情報処理装置が有するモジュールについて以前に生成されたハッシュ値である正解ハッシュ値とが一致し、かつ前記受信したデータの完全性を示す情報が、前記他の情報処理装置が有するデータが完全性を有していることを示す場合、前記他の情報処理装置が有するモジュール及びデータは完全性を有していると判定する判定手段と、
前記判定手段による判定結果を通知する通知手段と、

を備えることを特徴とする情報処理装置。
【請求項１３】
第１の情報処理装置と第２の情報処理装置とを備える情報処理システムであって、
前記第１の情報処理装置は、
モジュール、データ、及び当該データについて生成されたハッシュ値である正解ハッシュ値を格納するメモリと、
前記メモリに格納されている前記モジュールのハッシュ値と、前記メモリに格納されている前記データのハッシュ値と、を算出する算出手段と、
前記算出手段が算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定する判定手段と、
前記算出手段が算出した前記モジュールのハッシュ値と、前記判定手段による判定結果を示す情報と、を第２の情報処理装置に送信する送信手段と、を備え、
前記第２の情報処理装置は、
前記算出手段が算出した前記モジュールのハッシュ値と、前記判定手段による判定結果を示す情報と、を受信する受信手段と、
前記第１の情報処理装置が有するモジュールについて以前に生成されたハッシュ値である正解ハッシュ値を格納する格納手段と、
前記算出手段が算出した前記モジュールのハッシュ値と前記正解ハッシュ値とが一致し、かつ前記判定手段による判定結果を示す情報が、前記第１の情報処理装置が有するデータが完全性を有していることを示す場合、前記第１の情報処理装置が有するモジュール及びデータは完全性を有していると判定する判定手段と、
前記判定手段による判定結果を通知する通知手段と、を備える
ことを特徴とする、情報処理システム。
【請求項１４】
モジュール、データ、及び当該データについて生成されたハッシュ値である正解ハッシュ値をメモリに格納する情報処理装置が行う情報処理方法であって、
前記メモリに格納されている前記モジュールのハッシュ値と、前記メモリに格納されている前記データのハッシュ値と、を算出する算出工程と、
前記算出工程で算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定する判定工程と、
前記算出工程で算出した前記モジュールのハッシュ値と、前記判定工程における判定結果を示す情報と、を前記メモリに格納されている前記モジュール及び前記データの完全性を検証するサーバに送信する送信工程と、
を有することを特徴とする、情報処理方法。
【請求項１５】
情報処理装置が行う情報処理方法であって、
他の情報処理装置が有するモジュールのハッシュ値と、前記他の情報処理装置が有するデータの完全性を示す情報と、を受信する受信工程と、
前記受信したモジュールのハッシュ値と、他の情報処理装置が有するモジュールについて以前に生成されたハッシュ値である正解ハッシュ値とが一致し、かつ前記受信したデータの完全性を示す情報が、前記他の情報処理装置が有するデータが完全性を有していることを示す場合、前記他の情報処理装置が有するモジュール及びデータは完全性を有していると判定する判定工程と、
前記判定工程における判定結果を通知する通知工程と、
を備えることを特徴とする情報処理方法。
【請求項１６】
第１の情報処理装置と第２の情報処理装置とを備える情報処理システムが行う情報処理方法であって、
前記第１の情報処理装置は、モジュール、データ、及び当該データについて生成されたハッシュ値である正解ハッシュ値をメモリに格納しており、

前記情報処理方法は、
前記第１の情報処理装置が、前記メモリに格納されている前記モジュールのハッシュ値と、前記メモリに格納されている前記データのハッシュ値と、を算出する算出工程と、
前記第１の情報処理装置が、前記算出工程で算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定する第１の判定工程と、
前記第１の情報処理装置が、前記算出工程で算出した前記モジュールのハッシュ値と、前記第１の判定工程における判定結果を示す情報と、を第２の情報処理装置に送信する送信工程と、
前記第２の情報処理装置が、前記算出工程で算出した前記モジュールのハッシュ値と、前記第１の判定工程における判定結果を示す情報と、を受信する受信工程と、
前記第２の情報処理装置が、前記算出工程で算出した前記モジュールのハッシュ値と、前記第１の情報処理装置が有するモジュールについて以前に生成されたハッシュ値である正解ハッシュ値とが一致し、かつ前記第１の判定工程における判定結果を示す情報が、前記算出工程で算出した前記データのハッシュ値と前記正解ハッシュ値とが一致することを示す場合、前記第１の情報処理装置が有するモジュール及びデータは完全性を有していると判定する第２の判定工程と、
前記第２の情報処理装置が、前記第２の判定工程における判定結果を通知する通知工程と、
を有することを特徴とする、情報処理方法。
【請求項１７】
コンピュータを、請求項１乃至１２の何れか１項に記載の情報処理装置の各手段として機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、情報処理装置、情報処理システム、情報処理方法、及びプログラムに関する。
【背景技術】
【０００２】
コンピュータをサーバ等に接続する際、コンピュータ内部のモジュールが改竄されていないことを検証するための機器証明技術がある。機器証明技術を用いる場合、接続元のコンピュータ（以下、クライアントＰＣ(Personal Computer)と呼ぶ）は、内部に含まれる各モジュールから生成したハッシュ値をデジタル署名付きで接続先のサーバに送信する。サーバは、予めクライアントＰＣ内部に含まれる各モジュールのハッシュ値の正解値（又は期待値であり、以下、正解ハッシュ値と呼称する場合がある）をデータベースに保持しておく。そして、サーバは、クライアントＰＣから受信したハッシュ値と、データベース内の正解ハッシュ値を比較することで、クライアントＰＣが改竄されているか否かを判断する。
【０００３】
例えば、特許文献１及び非特許文献１には、クライアントＰＣが、ブート時に起動する各モジュールのハッシュ値をサーバに送信する技術が開示されている。サーバは、送信されたハッシュ値とデータベース内の正解ハッシュ値とを比較することで、クライアントＰＣの改竄を検知することができる。
【先行技術文献】
【特許文献】
【０００４】
【特許文献１】特許第４９５０１９５号公報
【非特許文献】
【０００５】
【非特許文献１】Trusted Computing Group(TCG) TPM Specification Version 1.2 (http

://www.trustedcomputinggroup.org/)
【発明の概要】
【発明が解決しようとする課題】
【０００６】
従来技術によれば、クライアントＰＣ内部に含まれる各モジュールに対する改竄が検知される。一方で、クライアントＰＣが格納しているデータに対する改竄を検知する需要も存在する。例えばモジュールが扱う、モジュールの挙動等を制御する各種設定が記載されたデータである設定ファイルが改竄された場合、モジュールが異常な挙動をすることが考えられる。この場合、データに対する改竄を検知できれば、クライアントＰＣの異常を検出することができる。また、例えば、メールの送受信を行うメールアプリが扱うアドレス帳データに対する改竄を検知することにより、予めサーバ上に保持されている正常なアドレス帳データを用いて、クライアントＰＣのアドレス帳データを復元することもできる。
【０００７】
一方で、モジュールの改竄を検知する上記の技術を、単純にデータに適用することは容易ではない。上記の技術においては、クライアントＰＣが有するモジュールの正解ハッシュ値を、サーバが管理する必要がある。したがって、この技術をデータに適用すると、クライアントＰＣに格納されたデータの正解ハッシュ値も、サーバが管理する必要があるため、クライアントＰＣ内のデータが更新されるたびに、サーバが管理する正解ハッシュ値も更新する必要がある。データはモジュールと比べて頻繁に更新されるため、このような方法を用いるとサーバの負荷が増大するという課題が存在する。
【０００８】
本発明は、サーバのような遠隔機器を用いてクライアントに対する改竄検知を行う構成において、サーバに対する負荷を抑えながら、クライアント上で頻繁に更新される情報に対する改竄をサーバに検知させることを目的とする。
【課題を解決するための手段】
【０００９】
本発明の目的を達成するために、例えば、本発明の情報処理装置は以下の構成を備える。すなわち、
モジュール、データ、及び当該データについて生成されたハッシュ値である正解ハッシュ値を格納するメモリと、
前記メモリに格納されている前記モジュールのハッシュ値と、前記メモリに格納されている前記データのハッシュ値と、を算出する算出手段と、
前記算出手段が算出した前記データのハッシュ値と、前記正解ハッシュ値とが一致するか否かを判定する判定手段と、
前記算出手段が算出した前記モジュールのハッシュ値と、前記判定手段による判定結果を示す情報と、を前記メモリに格納されている前記モジュール及び前記データの完全性を検証するサーバに送信する送信手段と、を備える
ことを特徴とする。
【発明の効果】
【００１０】
サーバのような遠隔機器を用いてクライアントに対する改竄検知を行う構成において、サーバに対する負荷を抑えながら、クライアント上で頻繁に更新される情報に対する改竄をサーバに検知させることができる。
【図面の簡単な説明】
【００１１】
【図１】情報処理装置の構成例を示すブロック図。
【図２】実施形態１に係る情報処理装置の機能構成例を説明するブロック図。
【図３】データベース１００３とリスト３０２を説明する図。
【図４】リスト３０２の生成及び更新手順を示すフローチャート。
【図５】実施形態１に係る改竄検知処理を説明するフローチャート。

【図６】実施形態２に係る改竄検知処理を説明するフローチャート。
【図７】実施形態２に係る改竄検知処理を説明するサブフローチャート。
【図８】実施形態２に係る情報処理装置の機能構成例を説明するブロック図。
【図９】アクセス制御リスト、実行権限表、及び更新頻度表を説明する図。
【図１０】各実施形態に係るシステム構成例を示すブロック図。
【図１１】データのハッシュ値計算ログ１１０３を説明する図。
【発明を実施するための形態】
【００１２】
以下、本発明の実施形態を図面に基づいて説明する。ただし、本発明の範囲は以下の実施形態に限定されるものではない。
【００１３】
［実施形態１］
（装置構成）
図１のブロック図を参照して、実施形態１に適用可能な情報処理装置１００の構成例を説明する。図１において、情報処理装置１００は、特に制限されないが、例えば一般に普及しているパーソナルコンピュータ、画像データのコピー、スキャン、若しくはプリント等を実行可能な画像処理装置、又はデジタル写真を撮影可能な撮像装置等でありうる。図１に示すように、本実施形態における情報処理装置１００は、ＲＯＭ１０１、ＨＤＤ１０２、ＴＰＭ１０３、ＲＡＭ１０４、及びＣＰＵ１０５を備える。
【００１４】
ＲＯＭ１０１は記憶装置であって、物理的又は論理的な書き換えが不可能な不揮発性メモリである。ＲＯＭ１０１は、ＢＩＯＳ１１０、各種モジュール、及びデータ等を記憶できる。ＢＩＯＳ１１０は情報処理装置１００全体を制御するモジュールである。また、ＢＩＯＳ１１０は、情報処理装置１００に電源が投入された際、情報処理装置内部で最初に起動されるモジュールである。
【００１５】
ＨＤＤ１０２は、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、モジュールＢ１１４、モジュールＡ１１３が扱うデータａ１１５及びデータｂ１１６、並びにモジュールＢ１１４が扱うデータｃ１１７等を記憶可能な記憶装置である。ここで、ブートローダ１１１はカーネル１１２の起動を制御するモジュールである。カーネル１１２は、各種モジュール（後述のモジュールＡ１１３及びモジュールＢ１１４）のロード、ＲＡＭ１０４のメモリ管理、及び不図示のキーボード又はディスプレイ等を用いた入出力機能を制御するモジュールである。
【００１６】
モジュールＡ１１３及びモジュールＢ１１４は、ワードプロセッサ、表計算、データベース管理、ネットワークブラウジング、メール送受信、映像・音声再生、印刷、及び通信等の、情報処理装置１００が実現する各種機能を提供するモジュールである。本実施形態では、ＨＤＤ１０２内にある各種機能を提供するモジュールが、モジュールＡ１１３及びモジュールＢ１１４から構成されているものとして説明する。しかしながら、本発明はこのような構成に限定されることはなく、情報処理装置１００はより多くのモジュールから構成されていてもよい。
【００１７】
本実施形態においては、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４に対する改竄が検知される。以下では、これらのプログラムをまとめてモジュールと呼ぶ。それぞれのモジュールに対しては、ＨＤＤ１０２上に記録されたプログラムの書き換え、又は情報処理装置１００に設けられたＲＯＭ１０１の交換等により、改竄が行われる可能性がある。
【００１８】
データａ１１５及びデータｂ１１６はモジュールＡ１１３が扱うデータであり、例えばモジュールの挙動を制御する設定ファイルでありうる。また、データｃ１１７はモジュー

ルＢ１１４が扱うデータである。モジュールと同様に、情報処理装置１００が扱うデータの数は限定されず、より多くのデータから構成されていてもよい。また、情報処理装置１００が扱うデータの種類も設定ファイルに限定されず、例えばデータａ１１５、データｂ１１６、及びデータｃ１１７の少なくとも１つが、アドレス帳データ又は文書データのようなモジュールにより作成されるデータであってもよい。以上のように、また、ＲＯＭ１０１及びＨＤＤ１０２のようなメモリは、モジュール及びデータを格納している。

【００１９】

ＴＰＭ１０３は、耐タンパー性を有するセキュリティチップである。耐タンパー性とは、外部からの解析を困難にすると共に、外部から解析しようとした場合に内部に記憶されているモジュール又はデータを破壊することにより自己防衛する特性である。ＴＰＭ１０３は、ＮＶＲＡＭ１１９、ＰＣＲ０（１２０）、ＰＣＲ１（１２１）、ＰＣＲ２（１２２）、ＰＣＲ３（１２３）、ＰＣＲ４（１２４）、及び制御部１１８を備える。

【００２０】

ＮＶＲＡＭ１１９は不揮発性メモリであり、デジタル署名の生成に必要な秘密鍵（クライアント秘密鍵及びサーバ秘密鍵）、公開鍵（クライアント公開鍵及びサーバ公開鍵）、及び公開鍵証明書等を記憶する。ＰＣＲ０～４は揮発性メモリであり、情報処理装置１００が備える各モジュール等のハッシュ値を記憶する。本実施形態ではＴＰＭ１０３は５つのＰＣＲを備えるが、ＰＣＲの数はこれに限定されず、例えばＰＣＲの数は５つより多くても良い。制御部１１８は、デジタル署名生成処理、及びＰＣＲ０～４へのハッシュ値保存処理などを実行する。

【００２１】

ここで、ＰＣＲへのハッシュ値保存処理について説明する。ハッシュ値保存処理において、制御部１１８は、所定のＰＣＲに既に保存されているハッシュ値Ｈａｓｈ１と、ＴＰＭ１０３の外部から入力されたモジュール又はデータのハッシュ値Ｈａｓｈ２と、を用いて次の式を計算する。そして、制御部１１８は、計算により得られた値Ｒｅｓｕｌｔ１をＰＣＲに保存する。

Ｒｅｓｕｌｔ１＝Ｈ（Ｈａｓｈ１｜Ｈａｓｈ２）　　（式１）

Ｈ（ｘ）は値ｘに対するハッシュ関数である。ハッシュ関数としては公知のＳＨＡ１、ＳＨＡ２５６、又はＳＨＡ５１２等のアルゴリズムを利用可能である。「ｘ｜ｙ」は値ｘと値ｙの連結を表す。

【００２２】

以上説明したＰＣＲへのハッシュ値保存処理は、情報処理装置１００が起動する際等に実行される。一方で、起動時等にＰＣＲに書き込まれたハッシュ値を改竄することは困難である。リセット後のＰＣＲにはデータのハッシュ値を書き込むことができる。しかしながら、既にハッシュ値が記録されているＰＣＲの値を書き換えようとすると、以前に記録されていたハッシュ値とも、新たに書き込もうとするデータのハッシュ値とも異なる値が、ＰＣＲには記録されることになる。

【００２３】

なお、以下で説明するように、第１の保存部２０２は、計算部２０１が算出したモジュールのハッシュ値をＴＰＭ１０３のＰＣＲに保存する。ここで、実際にＰＣＲに保存されるデータは、式（１）に示されるように、計算部２０１が算出したハッシュ値のハッシュ値である。しかしながら、所定の値に対してハッシュ関数を２回適用して得られた値もハッシュ値であることに変わりはないから、以下では、ＰＣＲに保存されるデータを単にモジュールのハッシュ値と呼ぶ。これは、第２の保存部２０４が保存する第１又は第２の所定値のハッシュ値、及び実施形態２で説明する結合ハッシュ値についても同様である。

【００２４】

ここで、情報処理装置１００の起動処理について説明する。情報処理装置１００に電源が投入されると、まずＢＩＯＳ１１０が実行される。その後、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４が、この純にロードされ及び実行される。モジュールＡ１１３及びモジュールＢ１１４は選択的にロード及び実行され

てもよい。すなわち、ロード及び実行されないモジュールが存在してもよい。また、モジュールＡ１１３及びモジュールＢ１１４のロード及び実行の順序は特に制限されない。すなわち、必要な時に必要なモジュールをロード及び実行することができる。また、モジュールのロード及び実行とは関係なく、任意の値のハッシュ値をＰＣＲに保存することもできる。

【００２５】

本実施形態では、前述したＰＣＲへのハッシュ値保存処理を、以上説明した情報処理装置１００の起動処理中に実行する。すなわち、ＢＩＯＳ１１０は自分自身のハッシュ値を算出し、算出したハッシュ値を式１に従ってＰＣＲ０へ保存する。そして、ＢＩＯＳ１１０はブートローダ１１１のハッシュ値を算出し、算出したハッシュ値を式１に従ってＰＣＲ１へ保存する。その後、ＢＩＯＳ１１０はブートローダ１１１を起動する。起動したブートローダ１１１はカーネル１１２のハッシュ値を算出し、算出したハッシュ値を式１に従ってＰＣＲ２へ保存する。その後、ブートローダ１１１はカーネル１１２を起動する。起動したカーネル１１２は、モジュール（モジュールＡ１１３及びモジュールＢ１１４）が必要となった場合にモジュールのハッシュ値を算出し、算出したハッシュ値を式１に従ってＰＣＲ３へ保存する。カーネル１１２は、モジュールが必要になった際にモジュールを起動する毎に、ハッシュ値保存処理を繰り返し実行する。さらにＴＰＭ１０３は、ＰＣＲに保存されたハッシュ値に対するデジタル署名を生成し、ＰＣＲに保存したハッシュ値とともに出力することができる。

【００２６】

本実施形態において、クライアントＰＣである情報処理装置１００は、ＰＣＲに保存されておりＴＰＭ１０３が出力したハッシュ値と、そのデジタル署名と、をサーバに送る。このサーバは、情報処理装置１００の、ＨＤＤ１０２のようなメモリに格納されているモジュール及びデータ等の完全性を検証する。例えば、このサーバは、送られたハッシュ値を正解ハッシュ値と比較することで、クライアントＰＣ内のモジュール及びデータ等に対する改竄の有無を検証することができる。なお、ハッシュ値がどのモジュール又はデータから算出されたのかを特定するために、サーバに送信するハッシュ値には、例えばモジュールのファイル名や識別子などを関連づけることができる。ここで、サーバが格納している正解ハッシュ値とは、情報処理装置１００のメモリに格納されているモジュールについて生成されたハッシュ値である。例えば、モジュールの正解ハッシュ値は、以前に、例えば完全性を有することが確認された際に生成されたモジュールのハッシュ値である。モジュールの正解ハッシュ値は、例えば、情報処理装置１００に対応づけて予めサーバに格納されていてもよいし、情報処理装置１００のモジュールがアップデートされた際にサーバに格納されてもよい。

【００２７】

（システム構成）

次に、図１０を参照して本実施形態におけるシステム構成例について説明する。図１０は本実施形態に適用可能なシステムの概要を示す図である。図１０に示すように、本実施形態における情報処理システム１０００は、クライアントＰＣ１００１、サーバ１００２、及びデータベース１００３を備える。クライアントＰＣ１００１とサーバ１００２は、有線又は無線の通信回線１００４を介して接続されており、互いにデータを通信可能である。また、クライアントＰＣ１００１及びサーバ１００２としては、前述した情報処理装置１００を用いることができる。また、サーバ１００２は、データベース１００３からのデータの読み込み及びデータベース１００３へのデータの書き込みが可能である。

【００２８】

ここで、データベース１００３について、図３（Ａ）を参照して説明する。図３（Ａ）は、データベース１００３の一例である。データベース１００３において、列「クライアントＰＣのＩＤ」は、データベース１００３に登録されているクライアントＰＣの識別子を表す。図３（Ａ）の例では、ＩＤ「００１」又は「００２」をそれぞれ持つ２台のクライアントＰＣ１００１及び不図示のクライアントＰＣがデータベース１００３に登録され

ている。本実施形態では、データベース１００３に登録されているクライアントＰＣが有するモジュール及びデータに対する改竄の有無を、サーバ１００２が検証する。

【００２９】

データベース１００３において、列「検証対象」は、サーバ１００２が検証するクライアントＰＣのモジュール名又はデータ名を示す。「検証対象」がモジュールを指す場合、「検証対象」にはモジュールを一意に特定する情報、例えば、モジュールのファイル名又はモジュールの識別子等が記録される。「検証対象」がデータを指す場合、「検証対象」には単に検証の対象がデータであることを示す情報、例えば「データ」が記録される。本実施形態においては、検証の対象はクライアントＰＣ１００１が有する特定のデータではなく、クライアントＰＣ１００１内のデータ全体である。サーバ１００２は、「検証対象」列に記載された各モジュール及びデータに対する改竄の有無を検証することで、クライアントＰＣ１００１に対する改竄の有無を検証する。図３（Ａ）の例では、サーバ１００２は、クライアントＰＣ１００１が有するモジュールＡ、モジュールＢ、及びデータ全体（データａ，データｂ，データｃ）に対する改竄の有無を検証する。

【００３０】

データベース１００３において、「正解ハッシュ値」列には、「検証対象」列に登録されているモジュール又はデータのそれぞれについての正解ハッシュ値が登録されている。上述の通り、正解ハッシュ値とは、クライアントＰＣ１００１が有するモジュールについて以前に生成されたハッシュ値である。サーバ１００２は、この「正解ハッシュ値」とクライアントＰＣから受信したハッシュ値とを「検証対象」毎に比較し、一致していれば検証対象に対する改竄はないと判定し、不一致ならば検証対象に対する改竄があると判定する。

【００３１】

本実施形態において、「検証対象」が「データ」となっている行の「正解ハッシュ値」は、第１の所定値のハッシュ値である。例えば、第１の所定値が２進数の値「１」のとき、ＳＨＡ１に従って計算した第１の所定値のハッシュ値は「ｄａ３９・・・０９」となる。クライアントＰＣ１００１及びサーバ１００２の双方は、第１の所定値又は第１の所定値のハッシュ値を知っている。第１の所定値又は第１の所定値は、このような値であれば何でもよい。そして、サーバ１００２は、「検証対象」がデータとなっている行の「正解ハッシュ値」とクライアントＰＣ１００１から受信したデータのハッシュ値とが一致していれば、クライアントＰＣ１００１に含まれるデータに対する改竄はないと判定する。一方、これが不一致ならばサーバ１００２はクライアントＰＣ１００１に含まれるデータに対する改竄があると判定する。

【００３２】

このように、サーバ１００２は、ＩＤ「００１」を持つクライアントＰＣ１００１が有する「モジュールＡ」、「モジュールＢ」、及び「データ全体」について、それぞれの正解ハッシュ値とクライアントＰＣから受信したハッシュ値とを比較する。こうして、図３（Ａ）の場合、サーバ１００２は、クライアントＰＣ１００１が有する「モジュールＡ」、「モジュールＢ」、及び「データ全体」に対する改竄の有無を検証する。ＩＤ「００２」を持つクライアントＰＣに関しても同様であるため、説明は省略する。

【００３３】

（機能構成）

図２のブロック図により実施形態１の情報処理装置（クライアントＰＣ）１００１と情報処理装置（サーバ）１００２の機能構成例を説明する。この機能構成は、ＣＰＵ１０５が、例えばＨＤＤ１０２に格納されている、本実施形態の処理を実現する情報処理プログラムを実行することで実現することができる。以下の機能構成によれば、情報処理装置（クライアントＰＣ）１００１が改竄されているか否かを、情報処理装置（サーバ）１００２で検証することができる。

【００３４】

まず、情報処理装置（クライアントＰＣ）１００１の機能構成を説明する。計算部２０

１は、ＲＯＭ１０１及びＨＤＤ１０２等に格納されているモジュールのハッシュ値と、ＨＤＤ１０２等に格納されているデータのハッシュ値と、を算出する。例えば、本実施形態において計算部２０１は、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４からハッシュ値を計算し、第１の保存部２０２に出力する。また、計算部２０１は、データａ１１５、データｂ１１６、データｃ１１７、第１の所定値、及び第２の所定値のハッシュ値を計算し、第１の検証部２０３に出力する。

【００３５】

第２の所定値は、第１の所定値と異なる値であれば特に限定されない。例えば、第１の所定値を２進数の「１」に、第２の所定値を２進数の「０」にすることができる。また、第２の所定値は、第１の所定値のように、クライアントＰＣ１００１とサーバ１００２の双方が知っている必要はなく、クライアントＰＣ１００１のみが知っていてもよい。ハッシュ値の計算に用いるハッシュ関数は特に限定されず、公知のＳＨＡ１、ＳＨＡ２５６、及びＳＨＡ５１２等のアルゴリズムを利用可能である。

【００３６】

第１の保存部２０２は、計算部２０１により計算された、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４のハッシュ値をセキュリティチップ２０５に保存する。セキュリティチップ２０５へのハッシュ値保存処理は、前述したＴＰＭ１０３のＰＣＲへのハッシュ値保存処理と同様のため、説明は省略する。以下では、第１の保存部２０２に格納されるハッシュ値のことを、モジュールハッシュ値と呼ぶことがある。

【００３７】

第１の検証部２０３は、計算部２０１により計算されたデータのハッシュ値と、リスト３０２に含まれる正解ハッシュ値とが一致するか否かを判定する。本実施形態において第１の検証部２０３は、計算部２０１により計算されたデータａ１１５、データｂ１１６、及びデータｃ１１７のハッシュ値を、リスト３０２に含まれる正解ハッシュ値と比較する。ここで、リスト３０２に含まれる正解ハッシュ値は、それぞれのデータについて生成されたハッシュ値であり、以前に生成されたハッシュ値でありうる。

【００３８】

本実施形態において第１の検証部２０３は、全てのデータについて、計算されたハッシュ値がリスト３０２の正解ハッシュ値と一致した場合に、計算部２０１により計算された第１の所定値のハッシュ値を第２の保存部２０４に出力する。また、第１の検証部２０３は、いずれかのデータのハッシュ値がリスト３０２の正解ハッシュ値と不一致の場合は、計算部２０１により計算された第２の所定値のハッシュ値を第２の保存部２０４に出力する。この、第１の所定値のハッシュ値及び第２の所定値のハッシュ値は、第１の検証部２０３による判定結果を示している。すなわち、第１の所定値のハッシュ値は、データが完全性を有すること、具体的には計算部２０１により計算されたデータのハッシュ値がリスト３０２の正解ハッシュ値と一致することを示す。また、第１の所定値のハッシュ値は、データが完全性を有さないこと、具体的には計算部２０１により計算されたデータのハッシュ値がリスト３０２の正解ハッシュ値と一致しないことを示す。

【００３９】

ここで、図３（Ｂ）を参照して、正解ハッシュ値のリスト３０２について説明する。図３（Ｂ）に示すリスト３０２は、クライアントＰＣに含まれる各モジュール（モジュールＡ、モジュールＢ）が扱うデータ（データａ、データｂ、データｃ）の正解ハッシュ値を保持している。すなわち、第１の検証部２０３は、計算部２０１により計算された各データのハッシュ値が正解ハッシュ値と一致するか否かを検証することで、データに対する改竄の有無を検証する。

【００４０】

リスト３０２には、正しく生成又は更新されたデータのハッシュ値を格納することができる。本実施形態においては、モジュールがデータを生成又は更新してメモリに格納した

ことに応じて、計算部２０１は生成又は更新されたデータのハッシュ値を算出し、正解ハッシュ値としてリスト３０２に保存する。モジュールにより作成された直後のデータは、改竄されていない完全性を有するデータであると考えることができる。このような構成により、リスト３０２には、完全性を有することが確認されたデータのハッシュ値が正解ハッシュ値として格納される。もっとも、モジュール自体が改竄されることにより不正なデータが作成される可能性もあるが、モジュールの改竄はサーバ１００２により検知可能であるから、いずれにしろクライアントＰＣ１００１の改竄は検知可能である。

【００４１】

具体的には、計算部２０１は、データが更新又は新規作成されると、更新後のデータのハッシュ値を計算し、リスト３０２内の該当するデータのハッシュ値を更新する。例えばデータａが更新された場合には、計算部２０１は、更新後のデータａから計算したハッシュ値で、リスト３０２内のデータａの正解ハッシュ値を更新する。また、データが新規作成された場合には、計算部２０１は、新規作成されたデータのハッシュ値を計算し、そのハッシュ値をリスト３０２に新規行として追加する。例えば、新規にデータｄが作成された場合は、計算部２０１はデータｄのハッシュ値を計算し、リスト３０２にデータｄのハッシュ値を新規行として追加する。

【００４２】

計算部２０１は、モジュールがデータの正当性を認証したことに応じて、このデータのハッシュ値を算出し、正解ハッシュ値としてリスト３０２に保存してもよい。例えば、ユーザがモジュールの設定ファイルを修正した場合、モジュールが設定ファイルが不正な項目を含んでいないと判断した場合に、計算部２０１は設定ファイルの正解ハッシュ値を更新することができる。このような構成によれば、外部から入力されたデータについても、完全性を有する状態でのハッシュ値をリスト３０２に保存することができる。

【００４３】

ここで、図４のフローチャートを参照して、リスト３０２の生成及び更新処理をより詳細に説明する。以下では、モジュールＡ１１３がデータａを更新する場合及びデータｄを新規作成する場合について説明する。モジュールＢ１１４がデータを更新又は新規作成する場合も、同様の処理によりリスト３０２を更新することができる。

【００４４】

まず、計算部２０１はモジュールＡ１１３のハッシュ値を計算し、第１の保存部２０２に出力する（ステップＳ４０１）。第１の保存部２０２は計算部２０１が出力したモジュールＡ１１３のハッシュ値をセキュリティチップ２０５に保存する（ステップＳ４０２）。カーネル１１２はモジュールＡ１１３をロード及び実行する（ステップＳ４０３）。起動したモジュールＡ１１３は、データｄを新規作成する（ステップＳ４０４）。カーネル１１２はリスト３０２を読み込む（ステップＳ４０５）。読み込みに成功した場合、計算部２０１はデータｄのハッシュ値を計算し、リスト３０２に追加する（ステップＳ４０７）。読み込みに失敗した場合には、リスト３０２を更新せずに処理が終了する。

【００４５】

また、ステップＳ４０４でデータを更新する場合には、ステップＳ４０７で計算部２０１はリスト３０２の該当データのハッシュ値を更新する。例えば、ステップＳ４０４でモジュールＡ１１３がデータａを更新した場合、ステップＳ４０７で計算部２０１は更新後のデータａのハッシュ値を計算し、リスト３０２に登録されているデータａのハッシュ値を計算されたハッシュ値で更新する。

【００４６】

ここで、ステップＳ４０５において正解ハッシュ値が記録されたリスト３０２を読み込む処理について、より詳細に説明する。本実施形態において、モジュールの完全性が検証されない場合には正解ハッシュ値が更新されないように、リスト３０２はＨＤＤ１０２等のメモリに保存されている。より具体的には、計算部２０１が算出したモジュールのハッシュ値が、予め算出されているモジュールの正解ハッシュ値のような所定値と異なる場合には、正解ハッシュ値は更新されない。

【００４７】
一例として、リスト３０２は、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３のハッシュ値が正解ハッシュ値と一致した時のみ復号可能となるように暗号化された状態で、ＨＤＤ１０２に保存されている。例えば、ＴＰＭ１０３のＰＣＲに保存されているＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３のハッシュ値が、リスト３０２を暗号化した時のハッシュ値と同じである場合に、リスト３０２が復号可能となる。
【００４８】
したがって、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３に対する改竄がなければ、カーネル１１２はリスト３０２を復号化することができる。一方で、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３のいずれかが改竄されている場合は、カーネル１１２はリスト３０２の復号に失敗するため、リスト３０２を読み出すことができない。このように、ステップＳ４０５では、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３が改竄されていない場合にのみ、リスト３０２が更新可能となるように制御が行われる。
【００４９】
なお、前述したリスト３０２の復号条件は一例であり、例えば、ＢＩＯＳ１１０、ブートローダ１１１、及びカーネル１１２のハッシュ値が暗号化時と一致することを復号条件としてもよい。また、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４のハッシュ値が暗号化時と一致することを復号条件としてもよい。上述した、暗号化時にＰＣＲに保存されていたハッシュ値と、現在のＰＣＲに保存されているハッシュ値とが一致した時にのみリスト３０２を復号可能とする暗号化機能を、以下では、ＴＰＭのシール機能と呼ぶ場合がある。
【００５０】
また、リスト３０２の保護のためには、上述のシール機能を用いる代わりに、ＴＰＭ１０３のＮＶＲＡＭ１１９にリスト３０２を保存してもよい。ＴＰＭ１０３のＮＶＲＡＭ１１９には、上述の復号条件と同様のアクセス条件を設定可能である。これにより、リスト３０２をＮＶＲＡＭ１１９に保存した時にＰＣＲに保存されていたハッシュ値と、ＮＶＲＡＭ１１９へのアクセス時にＰＣＲに保存されているハッシュ値とが一致した場合にのみ、リスト３０２の読み込みや書き換えが可能となる。例えば、リスト３０２をＮＶＲＡＭ１１９に保存した時にＴＰＭ１０３のＰＣＲに保存されていたＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３のハッシュ値を、ＮＶＲＡＭへのアクセス条件として設定できる。
【００５１】
ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３のいずれか改竄されると、リスト３０２をＮＶＲＡＭ１１９に保存した時にＰＣＲに保存されていたハッシュ値と、現在のＰＣＲに保存されているハッシュ値とが不一致となる。このために、リスト３０２をＮＶＲＡＭ１１９から読み出せなくなる。一方で、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、及びモジュールＡ１１３に改竄がない場合は、リスト３０２をＮＶＲＡＭ１１９から読み出すことができる。
【００５２】
なお、前述したＮＶＲＡＭ１１９へのアクセス条件は一例であり、例えば、ＢＩＯＳ１１０、ブートローダ１１１、及びカーネル１１２のハッシュ値が、リスト３０２の保存時のハッシュ値と一致することをアクセス条件としてもよい。また、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４のハッシュ値が、リスト３０２の保存時のハッシュ値と一致することをアクセス条件にしてもよい。
【００５３】
上述のアクセス制御機能を、以下では、ＴＰＭのＮＶＲＡＭ機能と呼称する場合がある。ＮＶＲＡＭ機能によれば、上述のように、ＮＶＲＡＭ１１９へのリスト３０２の保存時

にＰＣＲに保存されていたハッシュ値と、現在のＰＣＲに保存されているハッシュ値とが一致した時にのみ、ＮＶＲＡＭ１１９へのアクセスが許可される。

【００５４】

以上のように、ＨＤＤ１０２又はＴＰＭ１０３のようなメモリは、正解ハッシュ値を含むリスト３０２を格納する。そして、リスト３０２は、上述のＴＰＭのシール機能またはＮＶＲＡＭ機能で保護される。このため、クライアントＰＣ１００１に含まれるモジュール等に改竄がない場合にのみ、リスト３０２を読み込むことができる。

【００５５】

第２の保存部２０４は、第１の検証部２０３が出力した第１の所定値のハッシュ値又は第２の所定値のハッシュ値を、セキュリティチップ２０５に保存する。セキュリティチップ２０５へハッシュ値を保存する処理は、前述したＴＰＭ１０３のＰＣＲへハッシュ値を保存する処理と同様のため、説明は省略する。なお、上述のようにモジュールのハッシュ値をＰＣＲ０～３に保存している場合は、第１の所定値のハッシュ値又は第２の所定値のハッシュ値をＰＣＲ４に保存することができる。以下では、第２の保存部２０４に格納されるハッシュ値の事を、フラグハッシュ値と呼ぶことがある。

【００５６】

セキュリティチップ２０５は、第１の保存部２０２が保存したモジュールハッシュ値、及び第２の保存部２０４が保存したフラグハッシュ値に対して、デジタル署名を生成する。そして、セキュリティチップ２０５は、生成したデジタル署名と、モジュールハッシュ値と、フラグハッシュ値とを含む検証データを、送信部２０６に出力する。セキュリティチップ２０５としては、例えば前述のＴＰＭ１０３を利用することができる。

【００５７】

送信部２０６は、セキュリティチップ２０５が生成した検証データを、情報処理装置（サーバ）１００２の受信部２０７に送信する。上述のように、送信部２０６が送信する検証データには、計算部２０１が算出したモジュールのハッシュ値と、第１の検証部２０３による判定結果を示す情報と、が含まれる。

【００５８】

次に、情報処理装置（サーバ）１００２の機能構成を説明する。受信部２０７は、クライアントＰＣ１００１が有するモジュールのハッシュ値であるモジュールハッシュ値と、クライアントＰＣ１００１が有するデータの完全性を示す情報であるフラグハッシュ値と、を受信する。具体的には、受信部２０７は、情報処理装置（クライアントＰＣ）１００１の送信部２０６が送信した検証データを受信し、第２の検証部２０８に出力する。

【００５９】

第２の検証部２０８は、受信部２０７が受信した検証データを検証することで、情報処理装置（クライアントＰＣ）１００１が改竄されているか否かを検証する。なお、前述したように、検証データには、第１の保存部２０２が保存したモジュールハッシュ値、第２の保存部２０４が保存したフラグハッシュ値、及びそれらに対するデジタル署名が含まれる。第２の検証部２０８は、後述のように、クライアントＰＣ１００１が有しているモジュール及びデータの完全性を検証することにより、クライアントＰＣ１００１が改竄されているか否かを検証する。この際に、第２の検証部２０８は、計算部２０１が算出したモジュールのハッシュ値と、第１の検証部２０３による判定結果を示す情報と、モジュールについて以前に生成されたハッシュ値である正解ハッシュ値とを参照する。

【００６０】

第２の検証部２０８は、まず、検証データのデジタル署名を検証することで、検証データに含まれるモジュールハッシュ値及びフラグハッシュ値が改竄されているか否かを検証する。

【００６１】

第２の検証部２０８は、次に、クライアントＰＣ１００１が有するモジュール及びデータが完全性を有しているか否かを判定する。具体的には、第２の検証部２０８は、受信部２０７が受信したモジュールのハッシュ値と正解ハッシュ値とが一致するか否かを判定す

る。また、第２の検証部２０８は、受信部２０７が受信したデータの完全性を示す情報が、クライアントＰＣ１００１が有するデータが完全性を有していることを示すか否かを判定する。双方が満たされる場合、第２の検証部２０８は、クライアントＰＣ１００１が有するモジュール及びデータが完全性を有していると判定する。

【００６２】

具体的には、第２の検証部２０８は、検証データに含まれるモジュールハッシュ値と、データベース１００３に含まれる正解ハッシュ値とを比較することで、情報処理装置（クライアントＰＣ）１００１内に含まれるモジュールに対する改竄の有無を検証する。例えば、第２の検証部２０８は、検証データに含まれるモジュールＡ１１３のハッシュ値と、データベース１００３に登録されているモジュールＡ１１３の正解ハッシュ値とを比較することで、モジュールＡ１１３が改竄されているか否かを検証できる。検証データに含まれるモジュールＡ１１３のハッシュ値と、データベース１００３に登録されているモジュールＡ１１３の正解ハッシュ値とが一致する場合、第２の検証部２０８は「改竄なし」と判定できる。また、不一致の場合、第２の検証部２０８は「改竄あり」と判断できる。

【００６３】

第２の検証部２０８は、さらに、検証データに含まれるフラグハッシュ値と、データベース１００３の正解ハッシュ値とを比較することで、情報処理装置（クライアントＰＣ）１００１内のモジュールが扱うデータに対する改竄を検知する。第１の検証部２０３に関して説明したように、第２の保存部２０４は、データから計算したハッシュ値が、リスト３０２内の正解ハッシュ値と一致した場合に、第１の所定値のハッシュ値をセキュリティチップ２０５に保存する。したがって、検証データに含まれるフラグハッシュ値が第１の所定値のハッシュ値である場合、第２の検証部２０８は、情報処理装置（クライアントＰＣ）１００１内のデータに対する改竄はないと判断することができる。

【００６４】

そして、上述のようにデータベース１００３の「データ」行には第１の所定値のハッシュ値が登録されている。第２の検証部２０８は、検証データに含まれるフラグハッシュ値が、データベース１００３の「データ」行に登録されている正解ハッシュ値と一致する場合、情報処理装置（クライアントＰＣ）１００１内のデータに対する改竄はないと判断することができる。一方、第２の保存部２０４が第２の所定値のハッシュ値をセキュリティチップ２０５に保存した場合、検証データに含まれるフラグハッシュ値はデータベース１００３の「データ」行のハッシュ値と一致しない。この場合、第２の検証部２０８は、情報処理装置（クライアントＰＣ）１００１内のデータに対する改竄があると判断することができる。

【００６５】

通知部２０９は、第２の検証部２０８による判定結果を通知する。通知部２０９は、判定結果をクライアントＰＣ１００１に通知してもよいし、サーバ１００２の何らかの処理部に通知してもよいし、その他の外部機器に通知してもよい。

【００６６】

（改竄検知処理）

図５のフローチャートを参照して、本実施形態に係る改竄検知処理を説明する。クライアントＰＣ１００１の計算部２０１は、モジュールのハッシュ値を計算し、第１の保存部２０２に出力する（ステップＳ５０１）。第１の保存部２０２は、計算部２０１が出力したモジュールのハッシュ値をセキュリティチップ２０５に保存する（ステップＳ５０２）。

【００６７】

次に、計算部２０１はデータのハッシュ値を計算し、第１の検証部２０３に出力する（ステップＳ５０３）。第１の検証部２０３は、リスト３０２を読み込む（ステップＳ５０４）。第１の検証部２０３はリスト３０２を読み込めたかどうかを判定し（ステップＳ５０５）、読み込めた場合、計算部２０１が出力したデータのハッシュ値と、リスト３０２に記録された正解ハッシュ値とを比較する（ステップＳ５０６）。

【００６８】
第１の検証部２０３は、全データについてハッシュ値がリスト３０２の正解ハッシュ値と一致するかどうかを判定し（ステップＳ５０７）、一致する場合、第１の所定値のハッシュ値を第２の保存部２０４に出力する。この場合、第２の保存部２０４は第１の所定値のハッシュ値をセキュリティチップ２０５に保存する（ステップＳ５０８）。ステップＳ５０７で何れかのデータのハッシュ値がリスト３０２に記録された正解ハッシュ値と不一致の場合、第１の検証部２０３は、第２の所定値のハッシュ値を第２の保存部２０４に出力する。この場合、第２の保存部２０４は第２の所定値のハッシュ値をセキュリティチップ２０５に保存する（ステップＳ５０９）。
【００６９】
セキュリティチップ２０５は、第１の保存部２０２及び第２の保存部２０４が保存したハッシュ値に対するデジタル署名を生成する。そして、第１の保存部２０２が保存したハッシュ値、第２の保存部２０４が保存したハッシュ値、及びそれらのデジタル署名を含む検証データを生成する（ステップＳ５１０）。なお、ステップＳ５０５でリスト３０２の読み込みに失敗したと判定された場合、セキュリティチップ２０５は、第１の保存部２０２が保存したハッシュ値とそのデジタル署名を含む検証データを生成する。送信部２０６は、セキュリティチップ２０５が生成した検証データをサーバの受信部２０７に送信する（ステップＳ５１１）。
【００７０】
サーバ１００２の受信部２０７は、クライアントＰＣ１００１の送信部２０６が送信した検証データを受信し、第２の検証部２０８に出力する（ステップＳ５１２）。第２の検証部２０８は、検証データに含まれるデジタル署名を検証する（ステップＳ５１３）。第２の検証部２０８はデジタル署名の検証に成功したかどうかを判定し（ステップＳ５１４）、成功した場合、検証データに含まれるハッシュ値とデータベース１００３に含まれる正解ハッシュ値とを比較する（ステップＳ５１５）。上述したように、第２の検証部２０８は、検証データに含まれるモジュールハッシュ値とデータベース１００３に含まれる正解ハッシュ値とを比較することで、クライアントＰＣ１００１に含まれるモジュールに対する改竄の有無を検証する。また、第２の検証部２０８は、検証データに含まれるフラグハッシュ値とデータベース１００３に含まれる正解ハッシュ値とを比較することで、クライアントＰＣ１００１に含まれるデータに対する改竄の有無を検証する。
【００７１】
通知部２０９は、ステップＳ５１５におけるモジュール及びデータに対する改竄の検証結果をクライアントＰＣに通知し（ステップＳ５１６）、クライアントＰＣ１００１は検証結果を受信する（ステップＳ５１７）。なお、ステップＳ５１４でデジタル署名の検証に失敗したと判定された場合、ステップＳ５１６で通知部２０９は、デジタル署名検証に失敗したことをクライアントＰＣに通知することができる。
【００７２】
以上のように、本実施形態においては、クライアントＰＣ１００１によりデータの正解ハッシュ値リストが保持される。そして、データを生成又は更新する際には、クライアントＰＣが保持する正解ハッシュ値リストが更新される。したがって、更新されたデータのハッシュ値をサーバ１００２に送信することは必要ではない。また、データが生成又は更新されるたびに、サーバ１００２がデータベース１００３を更新することも必要ではない。さらに、本実施形態においては、クライアントＰＣ１００１は、各データのハッシュ値をサーバ１００２に送信する代わりに、各データに対する改竄の有無を検証し、その検証結果をサーバに送信する。このため、データの改竄検知も含めた機器証明を、サーバに大きな負荷をかけることなく実現できる。さらには、本実施形態においては、各データのハッシュ値をＰＣＲに保存する代わりに、全データについての改竄検証結果を示す第１の所定値のハッシュ値又は第２の所定値のハッシュ値がＰＣＲに保存される。このため、利用するＰＣＲの数を節約できる。
【００７３】

［実施形態２］
以下、本発明の実施形態２において行われる情報処理を説明する。なお、実施形態２において、実施形態１と略同様の構成については、同一符号を付して、その詳細説明を省略する。
【００７４】
実施形態１では、データ毎に改竄を検知するのではなく、データ全体について改竄が検知された。つまり、実施形態１では、クライアントＰＣ１００１内に改竄されたデータが１つもなければデータの「改竄なし」と判定され、改竄されたデータが１つでもあればデータの「改竄あり」と判定された。したがって、例えばクライアントＰＣ１００１内にデータａ、データｂ、及びデータｃの３つのデータが存在する場合、実施形態１では、データａ、データｂ、及びデータｃのうち、どのデータが改竄されたかは特定されなかった。
【００７５】
一方で、実施形態２では、ソフトＴＰＭを用いてデータ毎に改竄の有無を検証することにより、データ毎に改竄検知が行われる。すなわち実施形態２では、クライアントＰＣ１００１のメモリに格納されている複数のデータのそれぞれについて、完全性が検証される。前述の例では、クライアントＰＣ１００１は、データａ、データｂ、及びデータｃのうち、どのデータが改竄されているかを、メモリに格納されているそれぞれのデータについて正解ハッシュ値を参照して特定する。
【００７６】
実施形態１では、データ全体についての改竄検知結果に相当する第１の所定値のハッシュ値又は第２の所定値のハッシュ値がＴＰＭ１０３のＰＣＲに格納された。実施形態２においても、同様に、データ毎の改竄検知結果をＴＰＭ１０３のＰＣＲに格納することができる。一方で、ＴＰＭ１０３のＰＣＲ数には限りがあるため、データ毎の改竄検知結果をＴＰＭ１０３のＰＣＲに保存することはできないかもしれない。そこで、以下の説明において、データ毎の改竄検知結果、すなわち第１の検証部８０２による判定結果を示す結合ハッシュ値は、ソフトＴＰＭのＰＣＲに保存される。ソフトＴＰＭとは実施形態１のＴＰＭ１０３と同等の機能を持ち、耐タンパー性を実装するソフトウェアのことである。ソフトＴＰＭはソフトウェアとして、例えばＨＤＤ１０２に保存される。このような構成によれば、ＰＣＲの数を、ＨＤＤ１０２の容量が許す限り増やすことができる。一方で、計算部８０１が算出したモジュールのハッシュ値は、耐タンパー性がハードウェアにより実装されているＴＰＭ１０３に格納される。
【００７７】
［機能構成］
図８のブロック図を参照して実施形態２の情報処理装置（クライアントＰＣ）１００１と情報処理装置（サーバ）１００２の機能構成例を説明する。実施形態２の機能構成は、図８に示すように、実施形態１と類似しているが、異なる機能も有している。図８において、実施形態１とは異なる機能には異なる符号が付されており、以下ではこれらの機能について説明する。なお、これらの機能構成は、ＣＰＵ１０５が、例えばＨＤＤ１０２に保存されている、実施形態２の情報処理を実現するプログラムを実行することで実現される。
【００７８】
計算部８０１は、実施形態１の計算部２０１の機能に加えて、データ毎にデータを識別する値及びそのハッシュ値が記載されたログファイルを、計算ログとして送信部８０６に出力する機能を持つ。計算ログには、クライアントＰＣ１００１のメモリに格納されたデータのそれぞれについて、データの識別子と、計算部８０１が算出したデータのハッシュ値と、が記録されている。
【００７９】
ここで、計算部８０１が出力する計算ログについて図１１を参照して説明する。図１１に示すように、計算ログ１１０１の「データ」列には、計算部８０１によるハッシュ値計算の対象となったデータが何であるかを識別する値が保存されている。データを識別する

値は例えば、ファイル名又は識別子等である。そして、計算ログ１１０１の「ハッシュ値」列には、計算部８０１によりデータから計算されたハッシュ値が保存されている。例えば、計算部８０１が計算したデータａのハッシュ値が「４８２５・・・ａｆ」である場合、図１１に示すように、「データ」列には「データａ」が登録され、対応する「ハッシュ値」列にはデータａのハッシュ値「４８２５・・・ａｆ」が保存される。

【００８０】

第１の検証部８０２は、実施形態１の第１の検証部２０３と同様に、複数のデータのそれぞれについて計算部８０１が計算したハッシュ値と、リスト３０２に含まれる正解ハッシュ値とを比較する。そして、第１の検証部８０２は、複数のデータのそれぞれについて、計算部８０１が計算したハッシュ値と、リスト３０２に含まれる正解ハッシュ値とが一致するか否かを判定する。

【００８１】

具体的には、計算部８０１が計算したデータのハッシュ値がリスト３０２の正解ハッシュ値と一致している場合、第１の検証部８０２は、計算部８０１にデータのハッシュ値と第１の所定値のハッシュ値とを用いた結合ハッシュ値を計算させる。結合ハッシュ値とは、計算部８０１が算出したデータのハッシュ値と、第１の検証部８０２による判定結果と、から生成されたデータのハッシュ値である。本実施形態においては、結合ハッシュ値として、データのハッシュ値と、ハッシュ値比較結果を示す第１又は第２の所定値のハッシュ値とを結合して得られるデータのハッシュ値が用いられる。そして、第１の検証部８０２は、それぞれのデータについて、計算部８０１により計算された結合ハッシュ値を、第２の保存部８０３に出力する。

【００８２】

例えば、データａのハッシュ値をＨ（ａ）、第１の所定値のハッシュ値をＨ（ｖ１）、第２の所定値のハッシュ値をＨ（ｖ２）とする。このとき、データａのハッシュ値Ｈ（ａ）がリスト３０２の正解ハッシュ値と一致する場合、計算部８０１は、データａのハッシュ値と第１の所定値のハッシュ値とが結合されたデータＨ（ａ）｜Ｈ（ｖ１）を生成する。そして、計算部８０１は、このデータのハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ１））を、結合ハッシュ値として計算する。

【００８３】

一方で、データａのハッシュ値Ｈ（ａ）が正解ハッシュ値と不一致の場合は、計算部８０１は、データａのハッシュ値と第２の所定値のハッシュ値とが結合されたデータＨ（ａ）｜Ｈ（ｖ２）を生成する。そして、計算部８０１は、このデータのハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ２））を、結合ハッシュ値として計算する。

【００８４】

第１の検証部８０２は、このような処理をデータａ、データｂ、及びデータｃに対して行うことにより、それぞれのデータについての３つの結合ハッシュ値を第２の保存部８０３に出力する。後述するように、データａのハッシュ値Ｈ（ａ）と結合ハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ２））とを用いることにより、データａについての第１の検証部８０２による判定結果を知ることができる。このように、この結合ハッシュ値は、それぞれのデータについての第１の検証部８０２による判定結果を示している。なお、リスト３０２は、実施形態１と同様にＴＰＭのシール機能やＴＰＭのＮＶＲＡＭ機能で保護することができる。

【００８５】

第２の保存部８０３は、第１の検証部８０２が出力した結合ハッシュ値を第２のセキュリティチップ８０５に出力する。なお、後述するが、第２のセキュリティチップ８０５としては例えばソフトＴＰＭを利用可能である。ここで、結合ハッシュ値は、第２のセキュリティチップ８０５のそれぞれのＰＣＲに保存される。例えば、クライアントＰＣ１００１にデータが１００個存在する場合、それぞれのハッシュ値比較結果である結合ハッシュ値も１００個存在することになる。このとき、各データの結合ハッシュ値は第２のセキュリティチップ８０５内の１００個のＰＣＲ（例えば、ＰＣＲ０～ＰＣＲ９９）のそれぞれ

に保存される。

【００８６】

第１のセキュリティチップ８０４は、実施形態１のセキュリティチップ２０５と略同様の機能を有する。すなわち、第１のセキュリティチップ８０４は、ＰＣＲに保存されているハッシュ値に対するデジタル署名を生成し、ハッシュ値及びデジタル署名を第１の検証データとして送信部８０６に出力する。なお、第１のセキュリティチップ８０４のＰＣＲには、第１の保存部２０２が保存したＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４のハッシュ値が保存されている。従って、第１の検証データには、ＢＩＯＳ１１０、ブートローダ１１１、カーネル１１２、モジュールＡ１１３、及びモジュールＢ１１４のハッシュ値と、デジタル署名とが含まれる。

【００８７】

第２のセキュリティチップ８０５は、ＰＣＲに保存されているデータ毎の結合ハッシュ値に対するデジタル署名を生成し、結合ハッシュ値及びデジタル署名を第２の検証データとして送信部８０６に出力する。

【００８８】

本実施形態においては、データ毎に、ハッシュ値の比較結果を示す結合ハッシュ値が保存される。第２のセキュリティチップ８０５としてソフトＴＰＭを用いることにより、結合ハッシュ値を格納するＰＣＲを多数利用することができる。例えば、データａ、データｂ、及びデータｃに対する改竄検知を行う場合、第２のセキュリティチップ８０５のＰＣＲ０にデータａの結合ハッシュ値を、ＰＣＲ１にデータｂの結合ハッシュ値を、ＰＣＲ２にデータｃの結合ハッシュ値を保存することができる。このとき、第２の検証データには、データａの結合ハッシュ値、データｂの結合ハッシュ値、データｃの結合ハッシュ値、及びこれらの結合ハッシュ値に対するデジタル署名が含まれる。

【００８９】

また、第１のセキュリティチップ８０４を用い、第２のセキュリティチップ８０５を保護することができる。具体的には、第１のセキュリティチップ８０４が持つＴＰＭのシール機能で、ソフトＴＰＭである第２のセキュリティチップ８０５を暗号化することにより、第２のセキュリティチップ８０５を保護することができる。他の方法として、第２のセキュリティチップ８０５を第１のセキュリティチップ８０４のＮＶＲＡＭに保存することで、ＴＰＭのＮＶＲＡＭ機能を利用して第２のセキュリティチップ８０５へのアクセス制御を施すことができる。

【００９０】

送信部８０６は、第１のセキュリティチップ８０４が出力した第１の検証データ、第２のセキュリティチップ８０５が出力した第２の検証データ、及び計算部８０１が出力した計算ログ１１０１を、情報処理装置（サーバ）１００２の受信部８０７に送信する。上述のように、送信部８０６は、耐タンパー性がハードウェアにより実装された第１のセキュリティチップ８０４から、第１の検証データに含まれるモジュールのハッシュ値を読み込む。また、送信部８０６は、耐タンパー性がソフトウェアにより実装された第２のセキュリティチップ８０５から、第２の検証データに含まれる結合ハッシュ値を読み込む。さらに、計算ログ１１０１には、クライアントＰＣ１００１が有する複数のデータのそれぞれについて、データの識別子と、計算部８０１が算出したデータのハッシュ値と、が含まれている。また、第２の検証データには、クライアントＰＣ１００１が有する複数のデータのそれぞれについて、第１の検証部８０２による完全性の判定結果を示す、結合ハッシュ値が含まれている。

【００９１】

受信部８０７は、情報処理装置（クライアントＰＣ）１００１の送信部８０６が送信した第１の検証データ、第２の検証データ、及び計算ログ１１０１を受信し、第２の検証部８０８に出力する。

【００９２】

第２の検証部８０８は、受信部８０７から受信した第１の検証データからモジュールに対する改竄を検知し、第２の検証データ及び計算ログ１１０１から各データに対する改竄を検知する。モジュール（ＢＩＯＳやブートローダ等も含む）に対する改竄の検知は、実施形態１の第２の検証部２０８と同様に行われる。すなわち、第２の検証部８０８は、まずデジタル署名を用いて第１の検証データに対する改竄の有無を検証する。そして、第１の検証データに対する改竄がないと判定した場合、第２の検証部８０８は、第１の検証データに含まれるモジュールのハッシュ値とデータベース１００３内の正解ハッシュ値とを比較する。これらが一致すれば、第２の検証部８０８はモジュールに「改竄なし」と判断し、不一意ならばモジュールに「改竄あり」と判断する。

【００９３】

データに対する改竄の検知に関しては、第２の検証部８０８は、まずデジタル署名を用いて第２の検証データに対する改竄の有無を検証する。そして、第２の検証データに対する改竄がないと判定した場合、結合ハッシュ値の計算を行う。すなわち、第２の検証部８０８は、計算ログ１１０１に記載されている各データについて、計算ログ１１０１に記載されているデータのハッシュ値と、データベース１００３に記録されている「データ」の正解ハッシュ値と、の結合ハッシュ値を計算する。実施形態１と同様、データベース１００３には、「データ」の正解ハッシュ値として第１の所定値のハッシュ値が登録されている。そして、第２の検証部８０８は、データ毎に、第２の検証部８０８が計算した結合ハッシュ値と、第２の検証データに含まれる結合ハッシュ値と、を比較する。第２の検証部８０８は、結合ハッシュ値が一致すればデータに対する「改竄なし」と判断し、不一致ならば「改竄あり」と判断する。

【００９４】

以下では、データａに対する改竄の有無を検証する場合の具体例を説明する。第２の検証部８０８は、計算ログ１１０１に記載されているデータａのハッシュ値Ｈ（ａ）と、データベース１００３に「データ」の正解ハッシュ値として保存されている第１の所定値のハッシュ値Ｈ（ｖ１）と、を結合する。そして、第２の検証部８０８は、得られた値のハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ１））を、データａの結合ハッシュ値として計算する。そして、第２の検証部８０８は、計算した結合ハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ１））と、第２の検証データに含まれるデータａの結合ハッシュ値とを比較する。

【００９５】

データａが改竄されている場合、第２の検証データに含まれるデータａの結合ハッシュ値はＨ（Ｈ（ａ）｜Ｈ（ｖ２））であるから、第２の検証部８０８で計算した結合ハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ１））と一致しない。この場合、第２の検証部８０８は、データａに対して「改竄あり」と判断できる。一方、データａが改竄されていない場合、第２の検証データに含まれるデータａの結合ハッシュ値はＨ（Ｈ（ａ）｜Ｈ（ｖ１））となるため、第２の検証部８０８で計算した結合ハッシュ値Ｈ（Ｈ（ａ）｜Ｈ（ｖ１））と一致する。この場合、第２の検証部８０８は、データａに対して「改竄なし」と判断できる。

【００９６】

［改竄検知処理］

図６のフローチャートを参照して、実施形態２における改竄検知処理を説明する。クライアントＰＣ１００１の計算部８０１は、それぞれのモジュールのハッシュ値を計算し、第１の保存部２０２に出力する（ステップＳ６０１）。第１の保存部２０２は、計算部８０１が出力したモジュールのハッシュ値を第１のセキュリティチップ８０４に保存する（ステップＳ６０２）。次に、計算部８０１はそれぞれのデータのハッシュ値を計算し、計算したハッシュ値を第１の検証部８０２に出力し、同時にハッシュ値の計算ログ１１０１を送信部８０６に出力する（ステップＳ６０３）。

【００９７】

第１の検証部８０２は、リスト３０２を読み込む（ステップＳ６０４）。第１の検証部８０２は、リストを読み込みできたかどうかを判定し（ステップＳ６０５）、読み込みできた場合、第１の検証部８０２は、計算部８０１が出力したデータのハッシュ値と、リス

ト３０２の正解ハッシュ値とを比較する。そして、第１の検証部８０２は、比較結果に応じた結合ハッシュ値を計算部８０１に計算させ、第２の保存部８０３は結合ハッシュ値を第２のセキュリティチップ８０５に保存する（ステップＳ６０６）。

【００９８】

第１のセキュリティチップ８０４は、第１の保存部２０２が保存したハッシュ値に対するデジタル署名を生成し、第１の保存部２０２が保存したハッシュ値とそのデジタル署名とを含む第１の検証データを生成する（ステップＳ６０７）。第２のセキュリティチップ８０５は、第２の保存部８０３が保存した結合ハッシュ値に対するデジタル署名を生成し、第２の保存部８０３が保存した結合ハッシュ値とそのデジタル署名とを含む第２の検証データを生成する（ステップＳ６０８）。なお、ステップＳ６０５でリスト３０２の読み込みに失敗したと判定された場合、ステップＳ６０８で第２の検証データは生成されない。

【００９９】

送信部８０６は、第１のセキュリティチップ８０４が生成した第１の検証データ、第２のセキュリティチップ８０５が生成した第２の検証データ、及び計算部８０１が生成した計算ログ１１０１をサーバ１００２に送信する（ステップＳ６０９）。

【０１００】

サーバ１００２の受信部８０７は、クライアントＰＣ１００１の送信部８０６が送信した第１の検証データ、第２の検証データ、及び計算ログ１１０１を受信し、第２の検証部８０８に出力する（ステップＳ６１０）。第２の検証部８０８は、第１の検証データに含まれるデジタル署名を検証する（ステップＳ６１１）。第２の検証部８０８は、デジタル署名の検証に成功したかどうかを判定し（ステップＳ６１２）、デジタル署名の検証に成功した場合、第２の検証部８０８はデータベース１００３の正解ハッシュ値と第１の検証データに含まれるハッシュ値を比較する。こうして、第２の検証部８０８は、それぞれのモジュールに対する改竄の有無を検証する（ステップＳ６１３）。

【０１０１】

次に、第２の検証部８０８は、第２の検証データに含まれるデジタル署名を検証する（ステップＳ６１４）。そして、第２の検証部８０８は、デジタル署名の検証に成功したかどうかを判定する（ステップＳ６１５）。デジタル署名の検証に成功した場合、第２の検証部８０８は、計算ログ１１０１に含まれるハッシュ値とデータベース１００３に含まれるデータの正解ハッシュ値とを用いて、各データについて結合ハッシュ値を計算する。さらに、第２の検証部８０８は、計算した結合ハッシュ値と第２の検証データに含まれる結合ハッシュ値とを比較することで、各データに対する改竄の有無を検証する（ステップＳ６１６）。

【０１０２】

通知部２０９は、ステップＳ６１３における各モジュールに対する改竄有無の検証結果とステップＳ６１６における各データに対する改竄有無の検証結果を、クライアントＰＣ１００１に送信する（ステップＳ６１７）。クライアントＰＣ１００１は検証結果を受信する（ステップＳ６１８）。なお、ステップＳ６１２で第１の検証データに対するデジタル署名の検証に失敗した場合、通知部２０９は、第１の検証データのデジタル署名の検証に失敗したことを示す情報を検証結果としてクライアントＰＣ１００１に送信する（ステップＳ６１７）。同様に、ステップＳ６１６でデジタル署名の検証に失敗した場合、通知部２０９は、第２の検証データのデジタル署名の検証に失敗したことを示す情報を、検証結果としてクライアントＰＣ１００１に送信する（ステップＳ６１７）。

【０１０３】

ここで、ステップＳ６０６の処理を図７（Ａ）のサブフローチャートを参照して詳細に説明する。第１の検証部８０２は、計算部８０１が計算したデータのハッシュ値が、ステップＳ６０５で読み込んだリスト３０２に含まれるデータの正解ハッシュ値と一致するか否かを判定する（ステップＳ７０１）。ハッシュ値が一致する場合、計算部８０１は、計算部８０１が計算したデータのハッシュ値と、第１の所定値のハッシュ値とを結合する。

そして、計算部８０１は、結合された値のハッシュ値を結合ハッシュ値として計算する。第２の保存部８０３は、計算部８０１により計算された結合ハッシュ値を第２のセキュリティチップ８０５に保存する（ステップＳ７０２）。

【０１０４】

一方、ステップＳ７０１でデータのハッシュ値が正解ハッシュ値と一致しないと判定された場合、計算部８０１は、計算部８０１が計算したデータのハッシュ値と、第２の所定値のハッシュ値とを結合する。そして、計算部８０１は、結合された値のハッシュ値を結合ハッシュ値として計算する。第２の保存部８０３は、計算部８０１により計算された結合ハッシュ値を第２のセキュリティチップ８０５に保存する（ステップＳ７０３）。

【０１０５】

ステップＳ７０４で、第１の検証部８０２は、検証対象となっている全データに対してステップＳ７０１～ステップＳ７０３の処理が実施されたかどうかを判定する（ステップＳ７０４）。全データに対して処理が実施されていない場合、処理はステップＳ７０１に戻り、他のデータに対して結合ハッシュ値が計算される。こうして、全データについての結合ハッシュ値が第２のセキュリティチップ８０５に保存される。

【０１０６】

次に、ステップＳ６１６の処理を図７（Ｂ）のサブフローチャートを参照して詳細に説明する。第２の検証部８０８は、計算ログ１１０１に含まれるデータのハッシュ値と、データベース１００３に含まれる第１の所定値のハッシュ値と、を結合する。そして、第２の検証部８０８は、結合された値のハッシュ値を結合ハッシュ値として計算する（ステップＳ７１０）。第２の検証部８０８は、計算した結合ハッシュ値と、第２の検証データに含まれるデータの結合ハッシュ値が一致するかどうかを検証する（ステップＳ７１１）。結合ハッシュ値が一致する場合、第２の検証部８０８は、対象のデータに対する「改竄なし」と判定する（ステップＳ７１２）。一方、結合ハッシュ値が一致しない場合、第２の検証部８０８は、対象のデータに対する「改竄あり」と判定する（ステップＳ７１３）。

【０１０７】

ステップＳ７１４で、第２の検証部８０８は、検証対象となっている全データに対してステップＳ７１０～ステップＳ７１３の処理が実施されたかどうかを判定する（ステップＳ７１４）。全データに対して処理が実施されていない場合、処理はステップＳ７１１に戻り、他のデータに対して改竄の有無が判定される。こうして、検証対象となっている全データに対して、改竄の有無が検証される（ステップＳ７１４）。

【０１０８】

このように、本実施形態においては、クライアントＰＣ１００１による各データに対する改竄検知の結果に対応する結合ハッシュ値が、ソフトＴＰＭに保存される。このような方法によれば、サーバに大きな負荷をおけることなく、データ毎に改竄検知を行うことができる。

【０１０９】

［実施例１，２の変形例１］

実施形態１，２では、クライアントＰＣ１００１内の全データを対象に改竄検知を行った。しかしながら、特定のデータのみを対象に改竄検知を行うこともできる。また、改竄検知の対象とするデータを動的に決定することもできる。本変形例では、クライアントＰＣ１００１は、メモリが格納する複数のデータから、第１の検証部２０３，８０２による完全性の検証対象となるデータを選択する。このような処理は、例えば、クライアントＰＣ１００１が備える選択部（不図示）が行うことができる。

【０１１０】

完全性の検証対象となるデータの選択方法は特に限定されない。例えば、カーネル１１２が直接扱うデータのみを改竄検知の対象にしてもよいし、アプリケーションの設定ファイルのみを改竄検知の対象にしてもよい。以下では、データへのアクセス権限を示す情報、モジュールの実行権限を示す情報、又はデータの更新頻度を示す情報に従い、完全性の検証対象となるデータを選択する構成について説明する。本変形例では、完全性の検証対

象とされなかったデータについては、改竄検知が行われない。

【０１１１】

例えば、図９（Ａ）に示すアクセス制御リスト９０１に応じて、改竄検知の対象にするデータと改竄検知の対象にしないデータとを決定できる。図９（Ａ）のアクセス制御リスト９０１は、カーネル（図中のｓｙｓｔｅｍ）、クライアントＰＣ管理者（図中のａｄｍｉｎ）、及び一般ユーザ（図中のｕｓｅｒ）が、対象データに対してどのようなアクセス権を持つかを示している。例えば、図９（Ａ）のアクセス制御リスト９０１は、カーネルとＰＣ管理者はデータａの読み込み（Ｒｅａｄ）と書き込み（Ｗｒｉｔｅ）の両方が可能で、一方、ユーザはデータａの読み込みはできるが書き込みはできないことを示している。また、アクセス制御リスト９０１は、カーネル及びＰＣ管理者に加えてユーザもデータｂの読み込み及び書き込みが可能であることを示している。

【０１１２】

このように、ユーザによる書き込みが禁止されているデータａは、データｂよりも重要なデータである可能性が高い。したがって、重要なデータのみを保護したい場合は、ユーザによるアクセスが制限されているデータを改竄検知の対象とすることができる。具体例として、上述のアクセス制御リスト９０１に従って、ユーザが書き込み権限を有さないデータであるデータａを改竄検知の対象とし、データｂは改竄検知の対象から外すことができる。しかしながら、これは一例にすぎず、例えば、カーネルだけが読み書きできるデータのみが、重要なデータとして保護されてもよい。

【０１１３】

また、図９（Ｂ）に示すモジュールの実行権限表９０２を参照して、動的に改竄検知の対象とするデータを決定することもできる。図９（Ｂ）に示すモジュールの実行権限表９０２は、カーネル、ＰＣ管理者、及びユーザが、それぞれのモジュールを実行できるか否かを示している。例えば、図９（Ｂ）に示すモジュールの実行権限表９０２は、カーネル及び管理者はモジュールＡの実行が可能だが、一方でユーザはモジュールＡの実行が不可能であることを示している。

【０１１４】

このとき、ユーザにモジュールＡの実行権限がないことから、モジュールＡはクライアントＰＣの動作に影響を与える、重要なモジュールである可能性が高い。この場合、モジュールＡが扱うデータも重要なデータである可能性が高い。このため、モジュールＡが扱うデータを改竄検知の対象とすることで、重要なデータの保護が実現できる。一方、図９（Ｂ）によれば、モジュールＢはユーザも実行可能である。このため、モジュールＢが扱うデータは改竄検知対象としなくてもよい。しかしながら、これは一例にすぎず、例えば、カーネルだけが実行できるモジュールが扱うデータのみが、重要なデータとして保護されてもよい。このように、ユーザによる実行が制限されているモジュールが生成するデータを改竄検知の対象とすることができる。

【０１１５】

さらには、変形例２について説明するように、更新頻度の低いデータを完全性の検証対象とすることもできる。更新頻度の低いデータは、更新頻度の高いデータよりも、システムの動作に関わるために重要性が高い可能性がある。一方で、更新時におけるデータの意図しない破損を検出する目的では、更新頻度の高いデータを完全性の検証対象とすることもできる。

【０１１６】

上述の方法により改竄検知の対象とされた重要なデータに関しては、実施形態１，２と同様に、クライアントＰＣ１００１の第１の検証部２０３，８０２がデータのハッシュ値と正解ハッシュ値とを比較する。そして、サーバ１００２の第２の検証部２０８，８０８は、クライアントＰＣ１００１から送られた検証データとデータベース１００３とを参照してデータに対する改竄の検知を行う。

【０１１７】

上述の方法で改竄検知の対象とされたデータの数が少ない場合、実施形態２において、

データについての結合ハッシュ値を、第２のセキュリティチップ８０５ではなく、第１のセキュリティチップ８０４に保存してもよい。ハードウェアＴＰＭである第１のセキュリティチップ８０４は、ソフトＴＰＭである第２のセキュリティチップ８０５と比較して、有するＰＣＲの数は少ないが、より安全である。このため、第１のセキュリティチップ８０４は、少数の重要なデータについての結合ハッシュ値を保存するのに適している。この場合、それぞれのデータについてのハッシュ値比較結果を示す結合ハッシュ値は、それぞれ第１のセキュリティチップ８０４のＰＣＲに保存される。そして、実施形態２と同様に、サーバ１００２はクライアントＰＣ１００１から送られた検証データに含まれる結合ハッシュ値を用いて、それぞれのデータに対する改竄を検知する。なお、改竄検知の対象とされたデータのそれぞれについてのハッシュ値を格納するリスト３０２を、第１のセキュリティチップ８０４に保存してもよい。

【０１１８】

実施形態２において、重要なデータとそれ以外のデータを分けて保護することもできる。具体例としては、重要なデータの結合ハッシュ値を第１のセキュリティチップ８０４のＰＣＲに保存し、それ以外のデータの結合ハッシュ値を第２のセキュリティチップ８０５のＰＣＲに保存することができる。この場合でも、クライアントＰＣ１００１から送られたそれぞれの結合ハッシュ値をサーバ１００２が検証することにより、重要なデータ及びそれ以外のデータの双方について改竄を検知することができる。

【０１１９】

［実施形態１，２の変形例２］

実施形態１，２では、改竄検知の対象とされた全てのデータについて、クライアントＰＣ１００１がデータのハッシュ値と正解ハッシュ値との比較を行った。しかしながら、一部のデータについてクライアントＰＣ１００１がデータのハッシュ値と正解ハッシュ値との比較を行い、他のデータについてサーバ１００２がデータのハッシュ値と正解ハッシュ値との比較を行ってもよい。また、クライアントＰＣ１００１が比較を行うデータと、サーバ１００２が比較を行うデータとは、動的に決定することもできる。本変形例では、クライアントＰＣ１００１は、メモリが格納する複数のデータから、クライアントＰＣ１００１による完全性の検証対象となるデータと、サーバ１００２による完全性の検証対象となるデータと、を選択する。このようなデータの選択は、例えば、データへのアクセス権限を示す情報、モジュールの実行権限を示す情報、又はデータの更新頻度を示す情報に従って行うことができる。

【０１２０】

例えば、更新頻度が多いデータについてはクライアントＰＣ１００１はデータのハッシュ値と正解ハッシュ値との比較を行い、更新頻度が少ないデータについてはサーバ１００２がデータのハッシュ値と正解ハッシュ値との比較を行うことができる。つまり、更新頻度が多いデータについては、実施形態１，２と同様に、クライアントＰＣ１００１が計算されたデータのハッシュ値と正解ハッシュ値とを比較し、比較結果をサーバ１００２に送信する。そして、サーバ１００２は受信した比較結果とデータベース１００３とを参照して、更新頻度が多いデータに対する改竄の検知を行う。一方、更新頻度が少ないデータに関しては、クライアントＰＣ１００１は各データ（更新頻度が少ないデータ）から計算したハッシュ値をサーバ１００２に送信する。そして、サーバ１００２は、受信した各データのハッシュ値と、予めデータベース１００３に登録されている各データの正解ハッシュ値とを比較することで、各データの改竄を検知する。

【０１２１】

この場合、更新頻度が少ないデータに関しては、クライアントＰＣ１００１においてデータが更新される度に、サーバ１００２が有するデータベース１００３に格納された正解ハッシュ値を更新する必要がある。しかしながら、このような処理の対象が更新頻度が少ないデータに限定されているため、サーバ１００２の負荷を少なく抑えることができる。なお、更新頻度が少ないデータに関しては、サーバ１００２がハッシュ値の比較を行うため、各データについてハッシュ値がＰＣＲに格納される。この場合には、ＰＣＲの数が多

い第２のセキュリティチップ８０５（ソフトＴＰＭ）のＰＣＲに、更新頻度が少ないデータのハッシュ値を保存することができる。

【０１２２】

「更新頻度が多いデータ」と「更新頻度が少ないデータ」の識別方法としては、例えば、図９（Ｃ）に示すデータ毎の更新頻度表９０３を用いる方法が挙げられる。図９（Ｃ）に示す更新頻度表９０３は、データ毎に、所定の時間間隔内におけるデータ更新回数の平均値（以下、更新頻度）を示す。例えば、図９（Ｃ）は、データａには１日平均１回（１回／日）のデータ更新が発生すること、データｂには１年平均１回（１回／年）のデータ更新が発生することを示す。

【０１２３】

そして、更新頻度に関する所定の閾値を設定することにより、この閾値を用いて「更新頻度が多いデータ」と「更新頻度が少ないデータ」とを識別することができる。例えば、更新頻度が閾値以上であるデータを「更新頻度が多いデータ」と識別することができ、更新頻度が閾値未満であるデータを「更新頻度が少ないデータ」と識別することができる。一例として、更新頻度の閾値を１週間に１回（１回／週）に設定することができる。このとき、図９（Ｃ）に示すデータａの更新頻度は１回／日であるため、閾値（１回／週）以上であるため、データａは「更新頻度が多いデータ」として識別される。一方で、データｂの更新頻度は１回／年で、閾値（１回／週）未満であるため、データｂは「更新頻度が少ないデータ」として識別される。

【０１２４】

上述の閾値は一例であり、閾値を１回／月や１回／日としてもよい。また、それぞれのデータの更新頻度を算出する方法の一例としては、データ毎に更新回数とその更新日とを記録したログファイルを用いる方法が挙げられる。この場合、所定の時間間隔内（例えば、１日、１週、１月、又は１年）におけるデータ更新回数の平均値を更新頻度として算出できる。

【０１２５】

上述した「更新頻度が多いデータ」と「更新頻度が少ないデータ」の識別方法は一例であり、それ以外の方法を採用することもできる。例えば、現在の日時とデータの最終更新日時との差が所定の閾値以上なら、このデータを「更新頻度が少ないデータ」と判定し、所定の閾値未満なら、このデータを「更新頻度が多いデータ」と識別することもできる。

【０１２６】

また、ユーザによるアクセスが制限されているデータを重要なデータであるかもしれないため、このようなデータをサーバ１００２による改竄検知の対象とし、その他のデータをクライアント１００１による改竄検知の対象とすることもできる。この場合、サーバ１００２による改竄検知の対象となるデータの選択は、変形例１と同様に、データへのアクセス権限を示す情報に従って行うことができる。一方、ユーザによる実行が制限されているモジュールが生成するデータは重要なデータであるかもしれないため、このようなデータをサーバ１００２による改竄検知の対象とし、その他のデータをクライアント１００１による改竄検知の対象とすることもできる。この場合、サーバ１００２による改竄検知の対象となるデータの選択は、変形例１と同様に、モジュールの実行権限を示す情報に従って行うことができる。

【０１２７】

［実施形態１，２の変形例３］

実施形態１，２では、クライアントＰＣ１００１とサーバ１００２の双方が、第１の所定値のハッシュ値を共有していた。ここで、第１の所定値、又は第１の所定値のハッシュ値の代わりに、ナンスを使うこともできる。ナンスは例えば１６ｂｙｔｅの乱数で、クライアントＰＣ１００１とサーバ１００２とが通信する度に異なる値をとる。ナンスを利用することで、例えば、データが改竄されているにも関わらず攻撃者が第１の所定値をＴＰＭに保存することで、データに改竄がないように見せかける攻撃を防止できる。

【０１２８】

（その他の実施例）

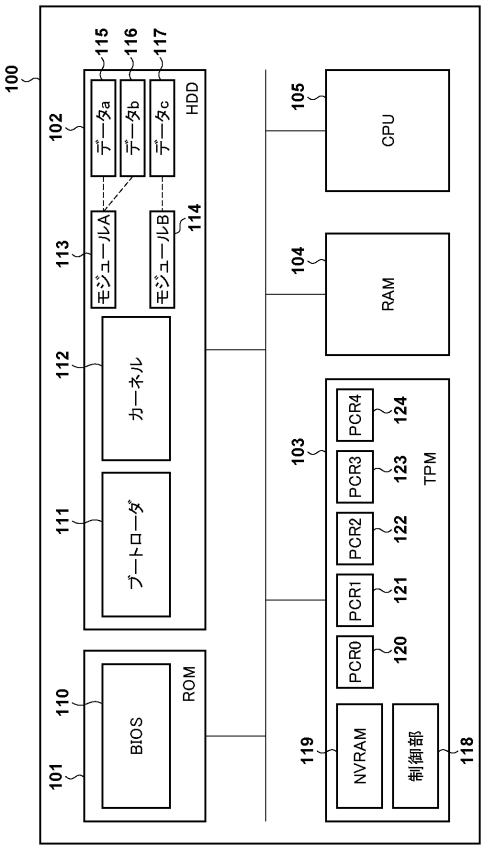
本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【符号の説明】

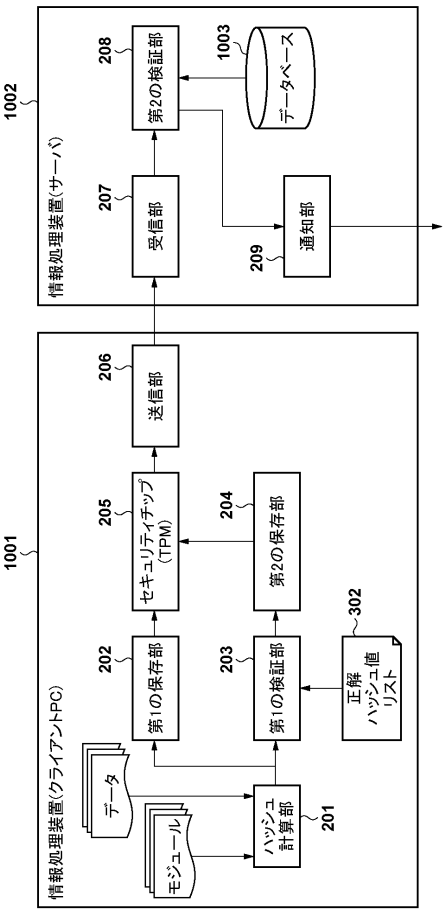
【０１２９】

１０１：ＲＯＭ、１０２：ＨＤＤ、１０３：ＴＰＭ、１１０：ＢＩＯＳ、１１１：ブートローダ、１１２：カーネル、１１３：モジュールＡ、１１４：モジュールＢ、１１５：データａ、１１６：データｂ、１１７：データｃ、２０１：ハッシュ計算部、２０４：第２の保存部、２０６：送信部、１００１：クライアントＰＣ、１００２：サーバ

【図１】



【図２】



【図３】

(A)

1003

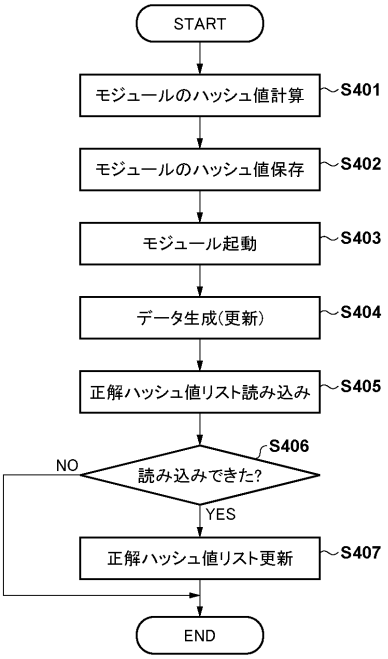
クライアントPCのID	検証対象	正解ハッシュ値
001	モジュールA	7f85・・・6a
	モジュールB	b311・・・8f
	データ	da39・・・09
002	モジュールC	9ed4・・・c1
	データ	da39・・・09

(B)

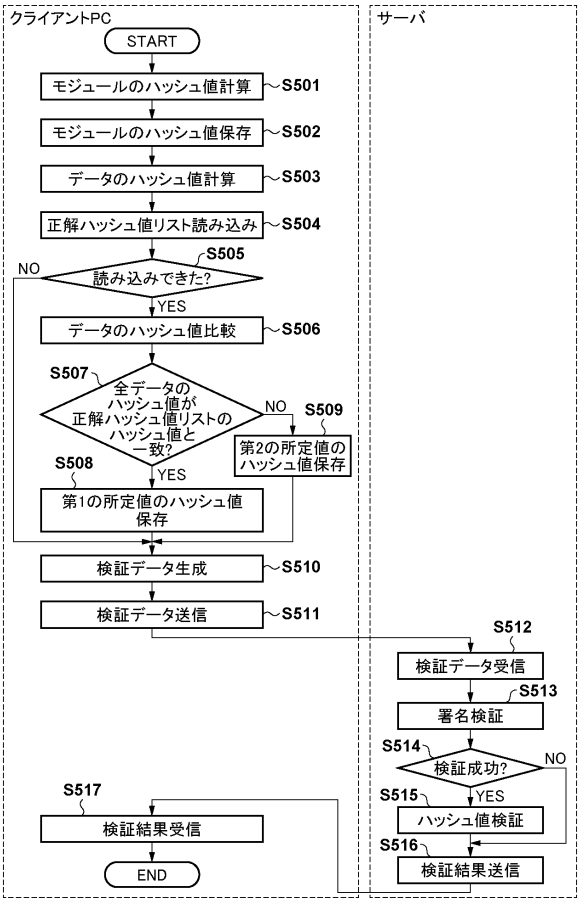
302

データ	正解ハッシュ値
データa	4825・・・af
データb	bf7a・・・e3
データc	cd23・・・62

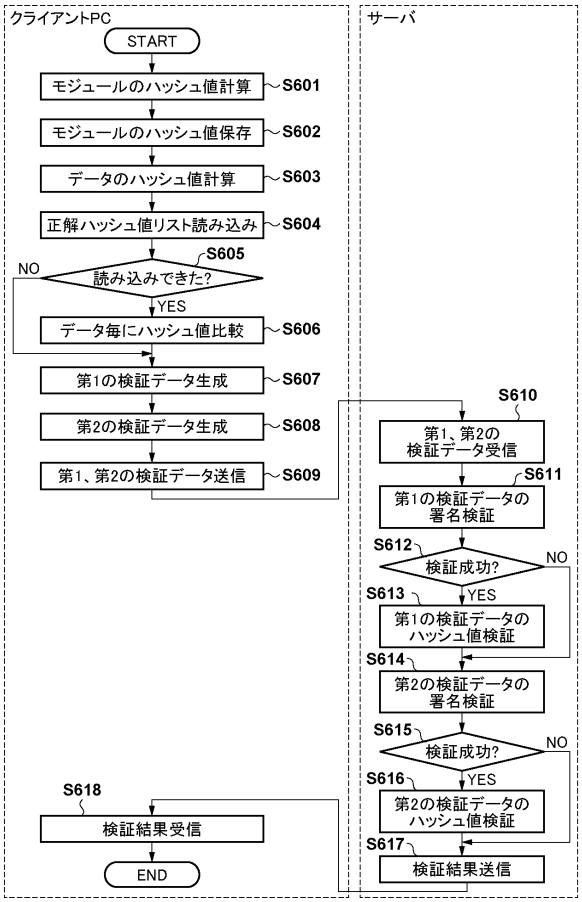
【図４】



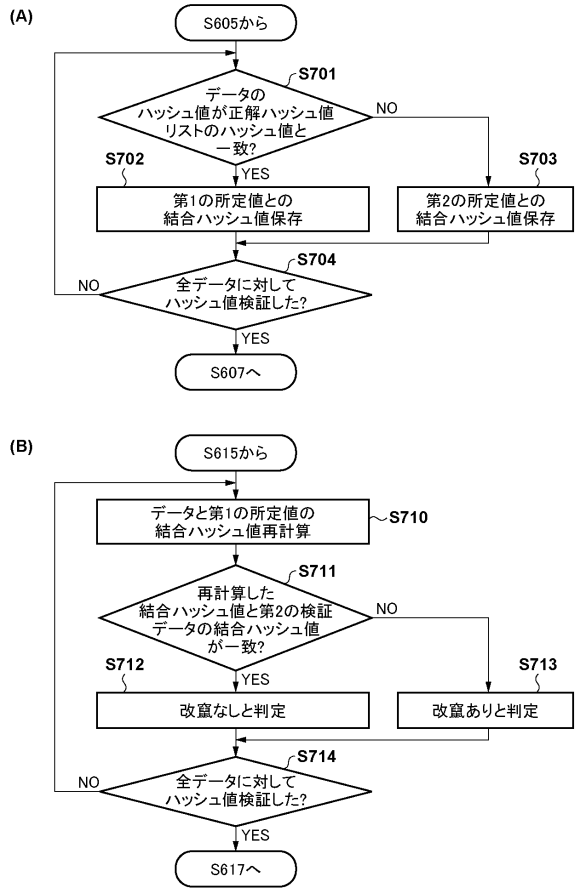
【図５】



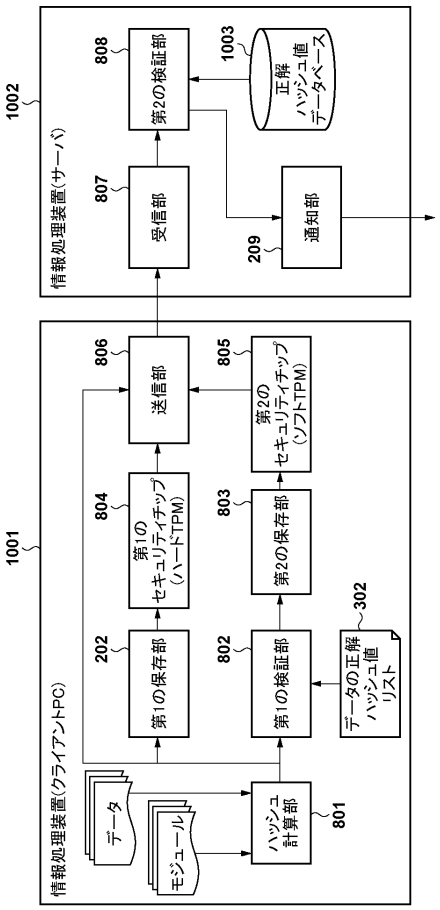
【図６】



【図７】



【図８】



【図９】

(A)

901

	system	admin	user
データa	Read/Write	Read/Write	Read
データb	Read/Write	Read/Write	Read/Write

(B)

902

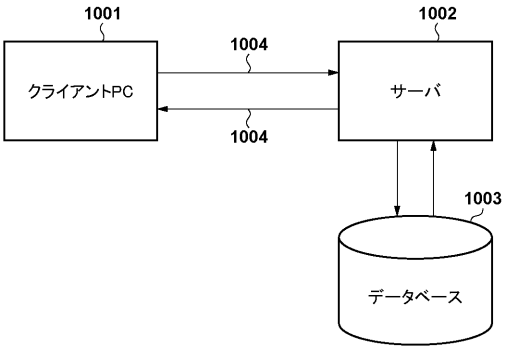
	system	admin	user
モジュールA	実行可	実行可	実行不可
モジュールB	実行可	実行可	実行可

(C)

903

	更新頻度
データa	1回/日
データb	1回/年

【図１０】



【図１１】

1101

データ	ハッシュ値
データa	4825･･･af
データb	bf7a･･･e2
データc	cd23･･･62

フロントページの続き

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

審査官　行田　悦資

(56)参考文献　米国特許出願公開第２０１５／０１１３５８７（ＵＳ，Ａ１）
国際公開第２００８／０２６２８７（WO，Ａ１）
特開２０１４－０９８９５１（ＪＰ，Ａ）
特開２００４－００５５８５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／３２
Ｇ０６Ｆ　２１／１２
Ｇ０６Ｆ　２１／６４