

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-44609

(P2010-44609A)

(43) 公開日 平成22年2月25日(2010.2.25)

(51) Int.Cl.
G06F 21/24 (2006.01)F I
G06F 12/14 560Cテーマコード (参考)
5B017

審査請求 有 請求項の数 10 O L (全 15 頁)

(21) 出願番号 特願2008-208480 (P2008-208480)
(22) 出願日 平成20年8月13日 (2008.8.13)(71) 出願人 000004237
日本電気株式会社
東京都港区芝五丁目7番1号
(74) 代理人 100079164
弁理士 高橋 勇
(72) 発明者 鈴木 和明
東京都港区芝五丁目7番1号 日本電気株
式会社内
Fターム(参考) 5B017 AA08 BA09 CA16

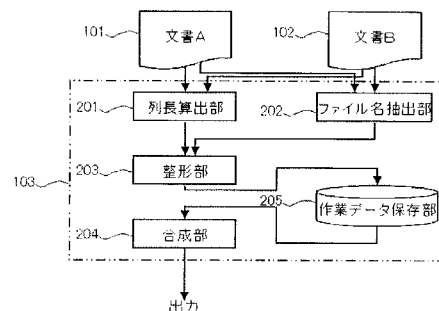
(54) 【発明の名称】 データ改ざん検出装置、データ改ざん検出方法、およびデータ改ざん検出プログラム

(57) 【要約】

【課題】複数の独立した異なるデータによって構成される一組のデータ対の関連付けを維持し、データの改ざんがあってもそのことを検出可能な改ざん検出装置などを提供する。

【解決手段】本発明に係る改ざん検出装置1は、一意的に関連づけられる必要のある第1および第2のデータ101～102の改ざんを検出する改ざん検出装置であって、第1のデータと第2のデータとをXOR（排他的論理和）演算して合成データ104を生成する演算部103と、合成データを記憶する記憶部205とを有し、演算部が第1のデータと第2のデータのうちのデータと合成データとをXOR演算して他方のデータを生成する。

【選択図】図2



【特許請求の範囲】**【請求項 1】**

一意的に関連づけられるべき第 1 および第 2 のデータの改ざんを検出するデータ改ざん検出装置であって、

前記第 1 のデータと第 2 のデータとを X O R (排他的論理和) 演算して合成データを生成する演算部と、前記合成データを記憶する記憶部とを有し、

前記演算部が、前記第 1 のデータと第 2 のデータのうちの一方のデータと前記合成データとを X O R 演算して前記他方のデータを生成することを特徴とするデータ改ざん検出装置。

【請求項 2】

10

前記演算部が、前記合成データを生成する際、

前記一方のデータが前記他方のデータより短い場合、前記他方のデータから前記一方のデータと同一のデータ長を第 1 の部分データとして切り出して残りを第 1 の残データとし、前記第 1 の部分データと前記一方のデータとを X O R 演算して得られた第 1 の合成データを前記記憶部に保存し、

以後、 n を 2 以上の自然数とすると、前記第 $n - 1$ の残データが前記一方のデータより長い場合、前記第 $n - 1$ の残データから前記一方のデータとデータ長が同一の第 n の部分データを切り出し、その残りを第 n の残データとして、前記第 n の部分データと前記一方のデータとを X O R 演算して得られた第 n の合成データを前記記憶部に保存する処理を繰り返す、

20

前記第 n の残データが前記一方のデータよりも短くなった場合に、前記一方のデータから前記第 n の残データと同一のデータ長を切り出した第 $n + 1$ の部分データと前記第 n の残データとを X O R 演算して得られた第 $n + 1$ の合成データを前記記憶部に保存し、

前記第 1 ないし第 $n + 1$ の合成データを切り出し順に従って連結することによって前記合成データを生成する合成部を有することを特徴とする、請求項 1 に記載のデータ改ざん検出装置。

【請求項 3】

前記合成部が、前記合成データから前記一方のデータと同一のデータ長を切り出して前記一方のデータと X O R 演算する処理を繰り返すと共に、前記 X O R 演算によって得られたデータを切り出し順に従って連結することによって前記他方のデータを生成するデータ復元機能を備えていることを特徴とする、請求項 2 に記載のデータ改ざん検出装置。

30

【請求項 4】

前記演算部が、

前記第 1 および第 2 の各データに、それぞれのデータ長およびファイル名を表す情報を連結する連結整形部を有し、

この連結整形部によってデータ長およびファイル名を表す情報を連結してなる前記第 1 および第 2 のデータに対して前記合成部が前記 X O R 演算を行うことを特徴とする、請求項 2 に記載のデータ改ざん検出装置。

【請求項 5】

前記第 2 のデータが生体認証データであり、前記合成データが前記第 1 のデータに添付されていることを特徴とする、請求項 1 に記載のデータ改ざん検出装置。

40

【請求項 6】

一意的に関連づけられるべき第 1 および第 2 のデータの改ざんを検出するデータ改ざん検出方法であって、

前記第 1 のデータと第 2 のデータとを X O R (排他的論理和) 演算して合成データを生成し、

前記合成データを記憶し、

前記第 1 のデータと第 2 のデータのうちの一方のデータと前記合成データとを X O R 演算して前記他方のデータを生成することを特徴とするデータ改ざん検出方法。

【請求項 7】

50

前記合成データを生成する処理が、

前記一方のデータが前記他方のデータより短い場合、前記他方のデータから前記一方のデータと同一のデータ長を第1の部分データとして切り出して残りを第1の残データとし、

前記第1の部分データと前記一方のデータとをXOR演算して得られた第1の合成データを保存し、

以後、 n を2以上の自然数とすると、前記第 $n-1$ の残データが前記一方のデータより長い場合、前記第 $n-1$ の残データから前記一方のデータとデータ長が同一の第 n の部分データを切り出し、その残りを第 n の残データとして、前記第 n の部分データと前記一方のデータとをXOR演算して得られた第 n の合成データを保存する処理を繰り返し、

10

前記第 n の残データが前記一方のデータよりも短くなった場合に、前記一方のデータから前記第 n の残データと同一のデータ長を切り出した第 $n+1$ の部分データと前記第 n の残データとをXOR演算して得られた第 $n+1$ の合成データを保存し、

前記第1ないし第 $n+1$ の合成データを切り出し順に従って連結することによって前記合成データを生成する

ことを特徴とする、請求項6に記載のデータ改ざん検出方法。

【請求項8】

前記合成データから前記一方のデータと同一のデータ長を切り出してXOR演算する処理を繰り返すと共に、

前記XOR演算によって得られたデータを切り出し順に従って連結することによって前記他方のデータを復元することを特徴とする、請求項7に記載のデータ改ざん検出方法。

20

【請求項9】

一意的に関連づけられるべき第1および第2のデータの改ざんを検出する改ざん検出装置を形成するコンピュータに、

前記第1のデータと第2のデータとをXOR（排他的論理和）演算して合成データを生成する手順と、

前記合成データを記憶する手順と、

前記第1のデータと第2のデータのうちの一方のデータと前記合成データとをXOR演算して前記他方のデータを生成する手順と

を実行させることを特徴とするデータ改ざん検出プログラム。

30

【請求項10】

前記合成データを生成する手順が、

前記一方のデータが前記他方のデータより短い場合、前記他方のデータから前記一方のデータと同一のデータ長を第1の部分データとして切り出して残りを第1の残データとする手順と、

前記第1の部分データと前記一方のデータとをXOR演算して得られた第1の合成データを保存する手順と、

以後、 n を2以上の自然数とすると、前記第 $n-1$ の残データが前記一方のデータより長い場合、前記第 $n-1$ の残データから前記一方のデータとデータ長が同一の第 n の部分データを切り出し、その残りを第 n の残データとして、前記第 n の部分データと前記一方のデータとをXOR演算して得られた第 n の合成データを保存する処理を繰り返す手順と

40

前記第 n の残データが前記一方のデータよりも短くなった場合に、前記一方のデータから前記第 n の残データと同一のデータ長を切り出した第 $n+1$ の部分データと前記第 n の残データとをXOR演算して得られた第 $n+1$ の合成データを保存する手順と、

前記第1ないし第 $n+1$ の合成データを切り出し順に従って連結することによって前記合成データを生成する手順と

を実行させることを特徴とする、請求項9に記載のデータ改ざん検出プログラム。

【発明の詳細な説明】

【技術分野】

50

【 0 0 0 1 】

本発明は、複数の独立したデータによって構成されるデータ対に関し、特にそのようなデータ対の改ざんの検出に係るデータ改ざん検出装置等に関する。

【 背景技術 】

【 0 0 0 2 】

たとえば仕様書とレビュー記録、身分証明書と指紋データなどのように、複数の独立したデータを一对のものとして管理し、それらのデータが改ざんされた場合にそのことを容易に検出可能であることが必要である場合がある。そのために専用のデータ管理システムを構築するのは、該データを利用する全てのユーザが利用可能なものである必要があるので、関係者が多い場合にはコスト面での課題が大きい。このため、既存のデータ保存システムを有効に活用し、かつ巨大な計算機システムを必要とせずに、データ対の改ざんを検出可能な管理方式が求められている。

10

【 0 0 0 3 】

これに関連する技術文献として、次の各々が開示されている。特許文献 1 には、メモリ素子上で 2 進数表現されたデータに対してデータ秘匿性を向上するスクランブル回路が記載されている。特許文献 2 および 3 には、登録文書に含まれる文字をデータベース化して検索可能とする文書管理装置が記載されている。

【 0 0 0 4 】

特許文献 4 には、データの内容を示す識別子が同一内容を示すか否かを判定する同定処理について記載されている。特許文献 5 には、電子メール本文の内容に関連する文書を当該電子メールに添付するための方法について記載されている。

20

【 0 0 0 5 】

【特許文献 1】特開 2 0 0 4 - 3 6 1 9 8 6 号公報

【特許文献 2】特開平 0 8 - 1 6 1 3 5 7 号公報

【特許文献 3】特開平 1 0 - 0 4 0 2 5 6 号公報

【特許文献 4】特開平 1 1 - 2 0 3 1 8 5 号公報

【特許文献 5】特許第 4 0 6 2 5 7 9 号公報

【 発明の開示 】

【 発明が解決しようとする課題 】

【 0 0 0 6 】

30

複数の独立した異なるデータによって構成される一組のデータ対で、その関連付けを維持しなければならない場合、通常の各々のデータの識別子の組を管理するという方法がとられる。この方法自体は特別なものではなく、特殊な処理を特に必要とはしない。

【 0 0 0 7 】

しかし、データ対を構成するデータ自体はそれぞれに独立したデータであるため、それぞれを編集して更新することが可能である。そのため、人為的なミス、あるいは故意等により、本来のデータの関連づけを維持できなくなる危険性がある。

【 0 0 0 8 】

たとえば、仕様書 A とデータ対を構成するレビュー記録表 α がある場合、仕様書 A を更新して A' とした場合にそれに対応してレビュー記録表 α が更新されないと、この時点で正しい関連付けが破壊されたことになる。このように、独立した異なるデータによってデータ対を構成した場合の関連づけ情報は、一方に改ざんが発生した場合にそのことを検出できず、その関連付けを完全に維持することができるものではなかった。

40

【 0 0 0 9 】

ハッシュ関数を用いるなどしてデータそのものの改ざんを検知する方法はある。しかしながら、ハッシュ値の組を管理するだけではデータ対の関連付けまで完全に維持できるものではない。たとえば、ハッシュ値を管理する組み合わせデータそのものを改ざんされた場合には、その関連付けを維持することはできない。

【 0 0 1 0 】

上述の特許文献 1 は、特定のハードウェアの構成を前提とするものであるので、上記の

50

問題を解決することはできない。また特許文献 2 ～ 5 はいずれも、文書内に文字として表現された内容の「関連性」についてのものである。上記のように複数の独立した異なるデータによって構成されるデータ対の「関連性」は、文字として表現された内容の「関連性」とは異なるものである。即ち、特許文献 1 ～ 5 のいずれの技術も、上記の問題に対して適用できるものではない。

【 0 0 1 1 】

本発明の目的は、複数の独立した異なるデータによって構成される一組のデータ対の関連付けを維持し、一方のデータに改ざんがあってもそのことを検出可能なデータ改ざん検出装置、データ改ざん検出方法およびそのプログラムを提供することにある。

【課題を解決するための手段】

【 0 0 1 2 】

上記目的を達成するため、本発明に係るデータ改ざん検出装置は、一意的に関連づけられる必要のある第 1 および第 2 のデータの改ざんを検出するデータ改ざん検出装置であって、第 1 のデータと第 2 のデータとを X O R (排他的論理和) 演算して合成データを生成する演算部と、合成データを記憶する記憶部とを有し、演算部が、第 1 のデータと第 2 のデータのうちの一方のデータと合成データとを X O R 演算して他方のデータを生成することを特徴とする。

【 0 0 1 3 】

上記目的を達成するため、本発明に係るデータ改ざん検出方法は、一意的に関連づけられる必要のある第 1 および第 2 のデータの改ざんを検出するデータ改ざん検出方法であって、第 1 のデータと第 2 のデータとを X O R (排他的論理和) 演算して合成データを生成し、合成データを記憶し、第 1 のデータと第 2 のデータのうちの一方のデータと合成データとを X O R 演算して他方のデータを生成することを特徴とする。

【 0 0 1 4 】

上記目的を達成するため、本発明に係るデータ改ざん検出プログラムは、一意的に関連づけられる必要のある第 1 および第 2 のデータの改ざんを検出する改ざん検出装置を形成するコンピュータに、第 1 のデータと第 2 のデータとを X O R (排他的論理和) 演算して合成データを生成する手順と、合成データを記憶する手順と、第 1 のデータと第 2 のデータのうちの一方のデータと合成データとを X O R 演算して他方のデータを生成する手順とを実行させることを特徴とする。

【発明の効果】

【 0 0 1 5 】

本発明は、上述したように第 1 および第 2 のデータを X O R (排他的論理和) 演算して生成した合成データに、さらに第 1 のデータと第 2 のデータのうちの一方のデータを X O R 演算して他方のデータを生成するので、特殊な計算を特に必要とはせずに、記憶保持している他方のデータと生成されたデータとを比較することによって、第 1 もしくは第 2 のデータに改ざんがあったことを検出可能である。これによって、複数の独立した異なるデータによって構成される一組のデータ対の関連付けを維持し、データの改ざんがあってもそのことを検出可能であるという、従来にない優れたデータ改ざん検出装置、データ改ざん検出方法およびそのプログラムを提供することができる。

【発明を実施するための最良の形態】

【 0 0 1 6 】

[第 1 の実施形態]

以下、本発明の実施形態の構成について添付図に基づいて説明する。

最初に、本実施形態の基本的な内容について説明し、その後でより具体的な内容について説明する。

本実施形態に係るデータ改ざん検出装置 1 は、図 1 ～ 2 に示すように、第 1 のデータ (文書 A 1 0 1) と第 2 のデータ (文書 B 1 0 2) とを X O R (排他的論理和) 演算して合成データ (C O M (A , B) 1 0 4) を生成する演算部 (演算装置 1 0 3) と、合成データを記憶する記憶部 (作業用データ保存部 2 0 5) とを有する。その上で演算部 (演算装

10

20

30

40

50

置 1 0 3) は、合成データ (COM (A, B) 1 0 4) と第 1 のデータと第 2 のデータのうちの一方のデータとを XOR 演算して他方のデータを生成する。

【0 0 1 7】

そして、演算部 (演算装置 1 0 3) は、次の処理を行う合成部 2 0 4 を備える。合成部 2 0 4 は、一方のデータ (文書 B 1 0 2) が他方のデータ (文書 A 1 0 1) より短い場合、他方のデータ (文書 A 1 0 1) から一方のデータ (文書 B 1 0 2) と同一のデータ長を第 1 の部分データとして切り出し、残りを第 1 の残データとし、と一方のデータ (文書 B 1 0 2) とを XOR 演算して得られた第 1 の合成データを記憶部 (作業用データ保存部 2 0 5) に保存する。

以後は、 n を 2 以上の自然数とすると、第 $n - 1$ の残データが一方のデータより長い場合、第 $n - 1$ の残データから一方のデータ (文書 B 1 0 2) とデータ長が同一の第 n の部分データを切り出し、その残りを第 n の残データとして、第 n の部分データと一方のデータ (文書 B 1 0 2) とを XOR 演算して得られた第 n の合成データを記憶部に保存するという処理を繰り返す。

第 n の部分データが一方のデータ (文書 B 1 0 2) よりも短くなれば一方のデータ (文書 B 1 0 2) から第 n の部分データと同一のデータ長を切り出した第 $n + 1$ の部分データと第 n の部分データとを XOR 演算して得られた第 $n + 1$ の合成データを記憶部に保存する。

そして、第 1 ~ 第 $n + 1$ の合成データを切り出し順に従って連結して、合成データ (COM (A, B) 1 0 4) を生成する。

【0 0 1 8】

そして、この合成部 2 0 4 は、合成データ (COM (A, B) 1 0 4) から一方のデータ (文書 B 1 0 2) と同一のデータ長を切り出して一方のデータ (文書 B 1 0 2) と XOR 演算する処理を繰り返し、これらの XOR 演算によって得られたデータを切り出し順に従って連結することによって他方のデータ (文書 A 1 0 1) を生成することができる。この処理は第 1 および第 2 のデータ 1 0 1 ~ 1 0 2 から合成データ (COM (A, B) 1 0 4) を生成する際の処理と同一でよい。

【0 0 1 9】

ここで演算部 (演算装置 1 0 3) は、第 1 および第 2 のデータ 1 0 1 ~ 1 0 2 に、それぞれのデータ長およびファイル名を表すデータを連結する整形部 2 0 3 を備えるようにし、合成部 2 0 4 は整形部 2 0 3 がデータ長およびファイル名を表すデータを連結したデータに対して前述の処理を行うようにすることもできる。

【0 0 2 0】

このようにすることにより、合成データ (COM (A, B) 1 0 4) と一方のデータ (文書 B 1 0 2) とがあれば、他方のデータ (文書 A 1 0 1) を生成することのできるデータ改ざん検出装置 1 を得ることができる。以下、これをさらに詳述する。

[基本原理]

関連付けるべき計算機上のデータ対を A および B とする。この二つのデータはコンピュータ上では 2 進数列として表現されている。データ A、B の 2 進数表現 (バイナリデータ) を各々 BIN (A)、BIN (B) とする。また、二つのデータの 2 進数列の長さを LEN (BIN (A))、LEN (BIN (B)) とする。説明を容易にするために、LEN (BIN (A)) = LEN (BIN (B)) であるとする。

【0 0 2 1】

ここで、この二つのデータに対して排他的論理和 (演算子 XOR) を計算する。データ A およびデータ B に対する合成を $COM (A, B) = BIN (A) \text{ XOR } BIN (B)$ と表現する。このとき、COM (A, B) は二つのデータ A および B のおのおのの情報を完全に含む情報であり、XOR の代数的性質を用いることでデータ A と COM (A, B) を用いることでデータ B を復元できる。具体的には、 $BIN (B) = COM (A, B) \text{ XOR } BIN (A)$ である。

【0 0 2 2】

10

20

30

40

50

同様に、データBとCOM(A、B)を用いることでデータAを復元できる。逆にデータA、Bと異なるデータでは、対となるデータを復元することができない。たとえば、データAの変更A'とCOM(A、B)を用いてもデータBを復元することはできない。このCOM(A、B)を用いて、二つのデータを管理することで、従来の課題を解決する。

【0023】

以下に示す本発明の実施形態では、仕様書とそのレビュー記録票を例に取って説明する。たとえば、仕様書Aのある特定のバージョン(以下単に仕様書Aと記する)とそれに対するレビュー記録表Bはその内容も含めて対として管理されていなければ意味を成さない。そこで、仕様書Aに対してレビュー記録表Bを関連付ける。具体的には、仕様書Asレビュー記録表Bの各々のバイナリデータであるBIN(A)およびBIN(B)に対して、次の演算を行う。

$$f = \text{COM}(A, B) = \text{BIN}(A) \text{ XOR } \text{BIN}(B)$$

【0024】

仕様書Aについて、そのレビュー記録の存在およびレビュー記録表の内容を確認しようとする場合は、該仕様書Aのデータと前述のfとを用いて以下の演算を実行する。

$$B = \text{COM}(f, A) = \text{COM}(A, B) \text{ XOR } \text{BIN}(A)$$

【0025】

このように管理対象となる文書を計算機上に表現された二進数列として排他的論理和を直接演算することで、関連する二つの文書の完全に一元的管理できる。本実施形態では、説明を簡単にするため、管理対象を文書データとして説明するが、管理対象となるデータは文書に限らず、コンピュータ上でバイナリデータとして表現されるデータなら何でもよい。

【0026】

なお、BIN(A) XOR BIN(B)というXOR演算を行う場合、BIN(A)とBIN(B)の2進数列の長さは必ず同一である必要がある。BIN(A)とBIN(B)の2進数列の長さが異なる場合の処理は後述する。

【0027】

以下、これをさらに詳述する。

図1は本発明の第1の実施形態に係るデータ改ざん検出装置1の構成を示す概略構成図である。データ改ざん検出装置1は、データ対をなす文書A101と文書B102を管理対象とし、それらから合成データを算出するコンピュータ装置である演算装置103を要部として構成されている。符号104は、演算装置103によって算出された合成データとしてのCOM(A, B)104を示す。この合成データCOM(A, B)は、上述の通りCOM(A, B) = BIN(A) XOR BIN(B)である。

【0028】

図2は、図1に示した演算部(演算装置103)の構成をより詳細に表す概略構成図である。演算装置103は、CPU、RAM、OSなどを含むコンピュータで、コンピュータプログラムを実行する主体であり、列長算出部201、ファイル名抽出部202、整形部203、合成部204、作業用データ保存部205といった各機能部を実行する。作業用データ保存部205は不揮発性メモリ(ハードディスク、フラッシュメモリなど)によって構成される。

【0029】

列長算出部201は、管理対象(処理対象)となる文書A101および文書B102の二進数列の長さ(ビット長)、即ち前述のLEN(BIN(A))およびLEN(BIN(B))を算出する。ファイル名抽出部202は、文書A101および文書B102の文書名を抽出する。ここで、文書A101および文書B102の文書名を各々NAM(A)およびNAM(B)と表記する。

【0030】

整形部203は、列長算出部201によって算出されたLEN(BIN(A))および

LEN(BIN(B))と、ファイル名抽出部202によって抽出されたNAM(A)およびNAM(B)と、各文書の2進数表現BIN(A)およびBIN(B)とを連結して、作業用データ300を生成し、不揮発性メモリである作業用データ保存部205に保存する。

【0031】

図3は、図2で示した整形部203が生成する作業用データ300の構成を示す説明図である。列長301は、列長算出部201によって算出された $L1 = \text{LEN}(\text{BIN}(A))$ もしくは $\text{LEN}(\text{BIN}(B))$ である。ファイル名302は、ファイル名抽出部202によって抽出されたNAM(A)もしくはNAM(B)である。ボディ303は、各文書の2進数表現BIN(A)もしくはBIN(B)である。列長301およびファイル名302は原則として固定長データであるのに対して、ボディ303はデータに応じて長さが変わる可変長データである。

10

【0032】

以後、文書A101に対して生成された作業用データ300を整形済文書A101bといい、文書B102に対して生成された作業用データ300を整形済文書B102bという。整形部203は逆に、作業用データ300から列長301およびファイル名302を取り除き、整形前のデータであるボディ303にファイル名302を付けて元のファイル(文書A101もしくは文書B102)を生成することも可能である。

【0033】

合成部204は、文書A101および文書B102の作業用データ300に対して後述のような排他的論理和演算と判断処理とを実行し、最終的な結果を出力する。

20

【0034】

なお、列長算出部201の行うビット長の算出、ファイル名抽出部202の行うファイル名の抽出、整形部203の行うデータの連結、合成部204の行う排他的論理和演算、および作業用データ保存部205の行うデータの保存については、通常の演算手法を利用して行うことができる。

【0035】

(合成データの生成)

図4は、図1および図2に示した演算装置103が実行する処理を表すフローチャートである。なお、図4では管理対象を2つの文書であるとして説明しているが、3つ以上の管理対象文書に対しては、2つの管理対象に対して実施した本発明の動作の結果得られた合成データをひとつの管理対象文書とみなして、3つ目の文書との間で処理を行い、以降同様に必要な文書数だけ処理を繰り返せばよい。

30

【0036】

まず演算部(演算装置103)は、管理対象である文書A101および文書B102を処理系内に取り込む(ステップS401)。取り込まれたこれらの文書に対してファイル名抽出部202がファイル名の抽出(ステップS402)、続いて列長算出部201が列長の算出の処理を行う(ステップS403)。なお、ステップS402とS403は、実行される順番が逆になってもよい。

【0037】

整形部203が、管理対象である文書A101および文書B102が作業用データ保存部205に読み込まれたことを確認し(ステップS404)、読み込まれていなければステップS401に戻る。確認後、文書A101および文書B102に対して整形部203が図3に示した形式の作業用データ300(整形済文書A101bおよびB102b)を作成し、作業用データ保存部205に保存する(ステップS405)。

40

【0038】

合成部204は、管理対象である文書A101および文書B102に対して整形処理が完了したことを確認し(ステップS406)、完了していなければステップS405に戻る。確認後、文書A101および文書B102を整形した整形済文書A101bおよびB102bに対して合成部204が排他的論理和により後述の演算処理を実行する(ステッ

50

プ S 4 0 7)。

【 0 0 3 9 】

図 5 は、図 4 で示したステップ S 4 0 7 の処理をより詳細に説明するフローチャートである。図 6 は、作業用データ保存部 2 0 5 に保存されているステップ S 4 0 7 の処理対象となる整形済文書 A 1 0 1 b および B 1 0 2 b に対する処理の状態を表す説明図である。図 6 (a) は、整形済文書 A 1 0 1 b および B 1 0 2 b の初期状態を示し、図 6 (b) は図 5 のステップ S 5 0 6 ~ 5 0 7 の 1 回目の処理について示す。また図 6 (c) は図 5 のステップ S 5 1 1 の処理について示す。

【 0 0 4 0 】

以後、説明の便宜上、整形済文書 A 1 0 1 b の方が、整形済文書 B 1 0 2 b より長いものとして説明する。整形済文書 B 1 0 2 b の方が長い場合は、以下の説明で整形済文書 A 1 0 1 b と B 1 0 2 b とが入れ替わるだけで、方法そのものに違いはない。先頭の文字を 0 文字目とし、整形済文書 A 1 0 1 b と整形済文書 B 1 0 2 b の各々の最終文字を Max A , Max B 字目とする。

【 0 0 4 1 】

合成部 2 0 4 がまず、図 6 に示した整形済文書 A 1 0 1 b および B 1 0 2 b を作業用データ保存部 2 0 5 から読み込む (ステップ S 5 0 1)。合成部 2 0 4 は整形済文書 A 1 0 1 b および B 1 0 2 b が正常に読み込まれたことを確認し (ステップ S 5 0 2)、正常に読み込まれなければステップ S 5 0 1 に戻る。確認後、読み込まれた整形済文書 A 1 0 1 b と B 1 0 2 b との間でデータ長を比較する (ステップ S 5 0 3)。この比較の結果、は整形済文書 A 1 0 1 b および B 1 0 2 b のうち、データ長のより長い方を単純に BIN と定義し、より短い方を pivot と定義する (ステップ S 5 0 4 または 5 0 5)。ここでは、整形済文書 A 1 0 1 b が BIN、整形済文書 B 1 0 2 b が pivot となる。

【 0 0 4 2 】

続いて合成部 2 0 4 は、pivot の大きさに合わせて BIN を部分数列として切り出し (ステップ S 5 0 6)、切り出された BIN の一部と pivot との間の XOR を演算し、その結果を Com (n) として作業用データ保存部 2 0 5 に記憶する (ステップ S 5 0 7)。ここで、BIN に対して i 文字目から k 文字目までの部分数列を切り出す処理を Part 関数として定義すると、ステップ S 5 0 6 の処理は以下の数 1 として表現できる。なお、i の初期値は 0、k の初期値は Max B = LEN (BIN (pivot))、n の初期値は 0 である。また、BIN、pivot とともに、先頭の文字を 0 文字目とする。

【 0 0 4 3 】

【 数 1 】

$$p = \text{Part}_i^k(\text{BIN})$$

【 0 0 4 4 】

続いて合成部 2 0 4 は、BIN の未処理部分数列の長さが pivot より短くなったか否かを判断し (ステップ S 5 0 8)、短くなっていないならばステップ S 5 0 7 で算出された XOR の値を作業用データ保存部 2 0 5 に記憶し (ステップ S 5 0 9)、BIN から処理済みの部分を取り除き、 $i = k$ 、 $k = k + \text{LEN}(\text{BIN}(\text{pivot}))$ 、 $n = n + 1$ として i、k および n の値を置き換え (ステップ S 5 1 0)、ステップ S 5 0 6 からの処理を繰り返す。

【 0 0 4 5 】

ステップ S 5 0 8 で、BIN の未処理部分数列の長さが pivot より短くなっていれば、今度は pivot の長さを BIN の未処理部分数列の長さに切断して、BIN の未処理部分数列と切断された pivot に対して排他的論理和を実行して、算出された XOR の値を作業用データ保存部 2 0 5 に記憶する (ステップ S 5 1 1)。

【 0 0 4 6 】

10

20

30

40

50

ステップS 5 1 1まででB I Nの全ての部分に対して排他的論理和C o m (n)の算出が終わった後、作業用データ保存部2 0 5に一時記憶されていたC o m (n)の値を取り出し、処理した時間順に連結する処理を行う(ステップS 5 1 2)。ここで連結された数値が、合成処理結果C O M (A、B)である。なお、C o m (n)の各々の値のデータ長は、最後にステップS 5 1 1で実行されたC o m (n)を除いて、全てp i v o tのデータ長と同一である。従って、データの連結に当たっては、区切りのために特定の文字を挿入する必要は特にない。

【0 0 4 7】

前述のように図6 (c)は、作業用データ保存部2 0 5に保存されている整形済文書A 1 0 1 bおよびB 1 0 2 bに対する、図5のステップS 5 1 1の処理の状態を表す。この図6 (c)で説明している例では、ステップS 5 0 6 ~ 5 1 0のループの2回目で、B I Nの未処理部分の長さが、p i v o tよりも短くなっている。ここでステップS 5 0 8の判断に基づいて、ステップS 5 1 1の処理に進んだ例について、図6 (c)では説明しているのである。

10

【0 0 4 8】

換言すれば、図5に示した処理は、B I Nを先頭からp i v o tと同一のデータ長に切り出してはp i v o tとの間のX O Rを演算して結果をC o m (n)として保存するという処理を繰り返し、B I Nの残りがp i v o tよりも短くなった場合に限って、今度はp i v o tをB I Nの残りと同一のデータ長に切り出してp i v o tとの間のX O Rを演算してこれもC o m (n)として保存するというものである。そして、X O Rを演算して得られた値であるC o m (n)を、処理した時間順に連結して、最終的にB I Nと同一データ長の合成データC O M (A、B) 1 0 4を得る。

20

【0 0 4 9】

(データの復元)

図7は、図4 ~ 5で述べた処理で合成部2 0 4によって生成された合成処理結果C O M (A、B) 1 0 4を利用して、図1 ~ 2に示した改ざん検出装置1によって元のデータを復元する処理について説明するフローチャートである。ここでは、文書B 1 0 2のバイナリデータであるB I N (B)と、合成処理結果C O M (A、B) 1 0 4とを利用して、B I N (A)を再生する。

【0 0 5 0】

まず演算部(演算装置1 0 3)は、B I N (B)と、合成処理結果C O M (A、B) 1 0 4のデータを作業用データ保存部2 0 5に読み込む(ステップS 7 0 1)。ここで読み込まれたデータが未整形のデータである場合は、これに対して作業用データである整形済文書B 1 0 2 bを作成する。この処理では、図4の4 0 2 ~ 4 0 5と同一の処理を実行して、整形済文書B 1 0 2 bを生成する(ステップS 7 0 2 ~ 7 0 4)。未整形データでなければ、そのまま作業用データ保存部2 0 5にとどめる。

30

【0 0 5 1】

この、整形済文書B 1 0 2 bとC O M (A、B)という2つのデータに対して、合成部2 0 4がステップS 4 0 7(または図5)と同一の合成処理を実行する。排他的論理和(X O R)の性質より、C O M (A、B)に整形済文書B 1 0 2 bをX O R演算すれば整形済文書A 1 0 1 bのデータが再現される。

40

【0 0 5 2】

図5 ~ 6で説明したようにC O M (A、B)が整形済文書B 1 0 2 bよりも長い場合、C O M (A、B)をB I Nとし、整形済文書B 1 0 2 bをp i v o tとして、図5で説明した処理と同様に、p i v o tの長さごとに合成処理結果C O M (A、B)を切断して、それらに対して順番にX O R p i v o tの演算を行っていき、その結果として得られた文字列を連結すれば整形済文書A 1 0 1 bが得られる。

【0 0 5 3】

再現された整形済文書A 1 0 1 bから、整形部2 0 3が、列長3 0 1とファイル名3 0 2を取り除き(ステップS 7 0 5 ~ 7 0 8)、ボディ3 0 3の部分をファイル名データ3

50

02のファイル名でファイルとすれば(ステップS709)、元の文書A101が生成される。列長301とファイル名302は、いずれも図3で示したように固定長データであるので、整形部203は、整形済文書A101bの先頭から所定のデータ長を取り除けば、ボディ303、つまり元の文書A101を得ることができる。

【0054】

(全体的な動作)

次に、上記の第1の実施形態の全体的な動作について説明する。本発明の第1の実施形態に係る改ざん検出装置1は、第1のデータ(文書A101)と第2のデータ(文書B102)とをXOR(排他的論理和)演算して合成データ(COM(A, B)104)を生成し、この合成データ(COM(A, B)104)を記憶し、合成データ(COM(A, B)104)と第2のデータ(文書B102)とをXOR演算して第1のデータ(文書A101)を生成するものである。

10

【0055】

ここで、一方のデータ(文書B102)が他方のデータ(文書A101)より短い場合、他方のデータ(文書A101)から一方のデータ(文書B102)と同一のデータを第1の部分データとして切り出して残りを第1の残データとし、第1の部分データと一方のデータとをXOR演算し(図5:ステップS506~507)、その結果として得られた第1の合成データを保存する(図5:ステップS509~510)。

以後、nを2以上の自然数とすると、第n-1の残データが一方のデータより長い場合、第n-1の残データから一方のデータとデータ長が同一の第nの部分データを切り出し、その残りを第nの残データとして、第nの部分データと一方のデータとをXOR演算して得られた第nの合成データを保存するという処理を繰り返す(図5:ステップS506~510)。

20

ここで、第nの残データが一方のデータよりも短くなった場合には(図5:ステップS508)、一方のデータから第nの残データとデータ長が同一の第n+1の部分データを切り出し、これと第nの残データとをXOR演算して得られた第n+1の合成データを保存する(図5:ステップS511)。

そして、これらの第1~第n+1の合成データを切り出し順に従って連結することによって合成データ(COM(A, B)104)を生成する(図5:ステップS512)。

【0056】

30

上記と同様の処理で、合成データ(COM(A, B)104)から第2のデータ(文書B102)と同一のデータ長を切り出してXOR演算する処理を繰り返し、それらのXOR演算によって得られたデータを連結することによって第1のデータ(文書A101)を生成する(図7:ステップS407、または図5)ようにすることもできる。

【0057】

ここで、上記各動作ステップについては、これをコンピュータで実行可能にプログラム化し、これらを前記各ステップを直接実行するコンピュータである改ざん検出装置1に実行させるようにしてもよい。

【0058】

以上説明したように、本発明においては、以上のような動作を行うことにより、次に記載するような効果を奏する。

40

第1のデータと第2のデータのうちの一方のデータをXOR演算して他方のデータを生成することにより、特殊な計算を特に必要とはせず、記憶保持している他方のデータと生成されたデータとを比較することによって、第1もしくは第2のデータに改ざんがあったことを検出可能である。

【0059】

即ち、仕様書とレビュー記録などのような必ず対とならなければ意味をなさないデータに対して、排他的論理和の演算が可能な演算装置であれば可能である動作で、偽造などを容易に検出することができる。この動作は、巨大な計算機システムなどを必要とするものではない。ここでいう「対とならなければ意味をなさないデータ」は、必ずしも文書内に

50

文字として表現された内容において「関連性」を持ったものである必要はない。

【 0 0 6 0 】

たとえば、仕様書を原本のまま管理し、人間の目で判読可能な状況を保ちつつ、レビュー記録表を合成として管理することで、仕様書の版数に一致したレビュー記録表を確実に保持することができるようになる。

【 0 0 6 1 】

また、本実施形態によれば簡単な代数的演算により一対となるデータの生成が可能で、かつ生成されたデータは通常のファイルとして管理可能であるので、ファイルシステムや言語などに依存することはなく、またデータの種類などについても問わずに適用することが可能である。

10

【 0 0 6 2 】

[第 2 の実施形態]

図 8 は、本発明の第 2 の実施形態に係るデータ改ざん検出装置 8 0 0 を示す概略構成図である。本実施形態に係るデータ改ざん検出装置 8 0 0 は、身分証明書 8 0 1 とそれに対応する生体認証データ 8 0 2 とからなるデータ対を管理対象とする。生体認証データ 8 0 2 としては、たとえば指紋、虹彩などを利用することができる。生成された合成データ 8 0 4 は、身分証明書 8 0 1 に添付される。

【 0 0 6 3 】

身分証明書 8 0 1 と生体認証データ 8 0 2 は、いずれもバイナリデータとして表現可能であるので、前述の第 1 の実施形態と全く同一の構成の演算装置 1 0 3 によって、合成データ 8 0 4 を生成して、前述の第 1 の実施形態と全く同一の処理を行い、それによって身分証明書 8 0 1 と生体認証データ 8 0 2 のうちのいずれかまたは両方が改ざんされたことを検知することができる。この効果もまた、前述の第 1 の実施形態と全く同一である。

20

【 0 0 6 4 】

合成データ 8 0 4 そのものは人間の目で判読可能なものではないので、合成自体を有意に改ざん / 改変することは困難である。従って、合成データ 8 0 4 自体は秘匿される必要がないので、管理対象となるデータ、ここでは身分証明書 8 0 1 自身に合成データ 8 0 4 を記録として保持させても特に問題はない。具体的には、磁気ストライプや 2 次元バーコードなどを利用して、合成データ 8 0 4 を身分証明書 8 0 1 に添付するようにすることができる。

30

【 0 0 6 5 】

たとえば身分証明書 8 0 1 がパスポートである場合、当該パスポートそのものが改ざん / 成りすまし検知機能を備えることになるので、発行国の管理するデータを参照するという手間をかけずに正当性の確認が可能となる。

【 0 0 6 6 】

これら以外にも、バイナリデータとして表現可能なデータであり、かつ一対一で対応させる必要があるデータであれば、本発明の第 1 および第 2 の実施形態を適用することができる。ファイルシステムや言語などには特に依存しない。

【 0 0 6 7 】

これまで本発明について図面に示した特定の実施形態をもって説明してきたが、本発明は図面に示した実施形態に限定されるものではなく、本発明の効果を奏する限り、これまで知られたいかなる構成であっても採用することができる。

40

【 産業上の利用可能性 】

【 0 0 6 8 】

バイナリデータとして表現可能で、かつ一対一で対応させる必要があるデータ対をなすデータであれば、種類を問わずに適用することができる。また、ファイルシステムや言語などにも依存しない。たとえば文書管理、情報管理、ID 管理などの用途に特に適している。

【 図面の簡単な説明 】

【 0 0 6 9 】

50

【図 1】本発明の第 1 の実施形態に係るデータ改ざん検出装置 1 の構成を示す概略構成図である。

【図 2】図 1 に示した演算装置 103 の構成をより詳細に表す概略構成図である。

【図 3】図 2 で示した整形部 203 が生成する作業用データ 300 の構成を示す説明図である。

【図 4】図 1 および図 2 に示した演算装置 103 が実行する処理を表すフローチャートである。

【図 5】図 4 で示したステップ S407 の処理をより詳細に説明するフローチャートである。

【図 6】作業用データ保存部 205 に保存されているステップ S407 の処理対象となる整形済文書 A101b および B102b に対する処理の状態を表す説明図である。図 6 (a) は整形済文書 A101b および B102b の初期状態を示し、図 6 (b) は図 5 のステップ S506 ~ 507 の 1 回目の処理について示し、図 6 (c) は図 5 のステップ S511 の処理について示す。

10

【図 7】図 4 ~ 5 で述べた処理で合成部 204 によって生成された合成処理結果 COM (A、B) 104 を利用して、図 1 ~ 2 に示したデータ改ざん検出装置 1 によって元のデータを復元する処理について説明するフローチャートである。

【図 8】本発明の第 2 の実施形態に係るデータ改ざん検出装置 800 を示す概略構成図である。

【符号の説明】

20

【0070】

1、800 データ改ざん検出装置

101 文書 A

102 文書 B

101b 整形済文書 A

102b 整形済文書 B

103 演算装置

104 COM (A, B)

201 列長算出部

202 ファイル名抽出部

203 整形部

204 合成部

205 作業用データ保存部

300 作業用データ

301 列長

302 ファイル名

303 ボディ

801 身分証明書

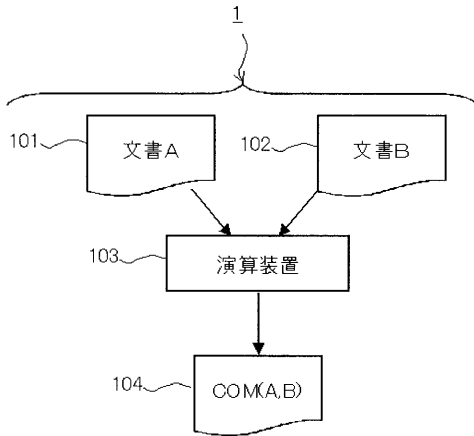
802 生体認証データ

804 合成データ

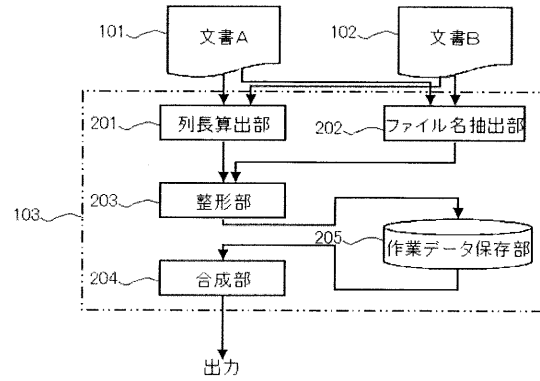
30

40

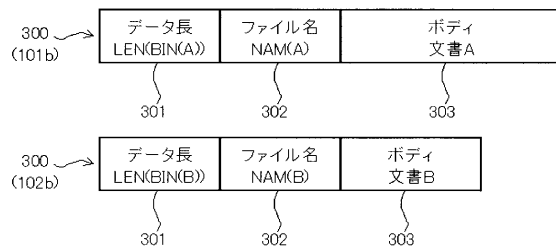
【図 1】



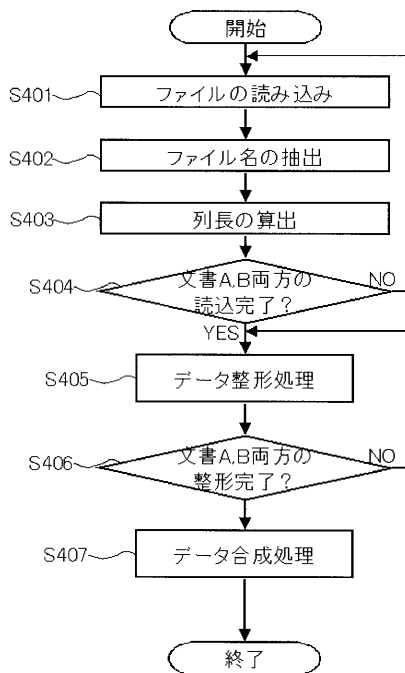
【図 2】



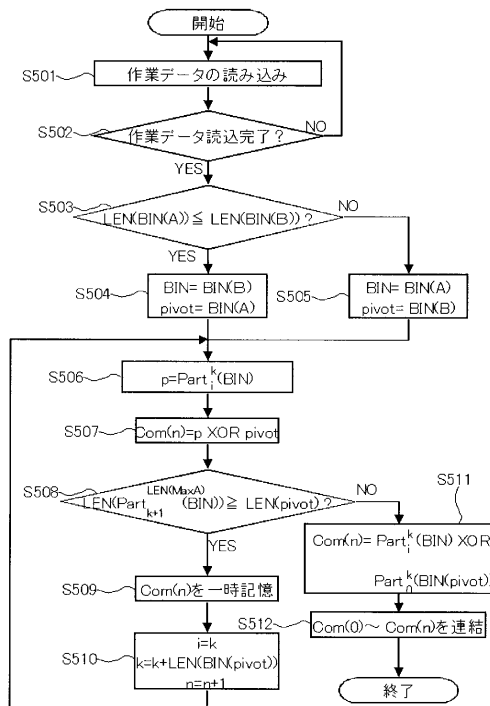
【図 3】



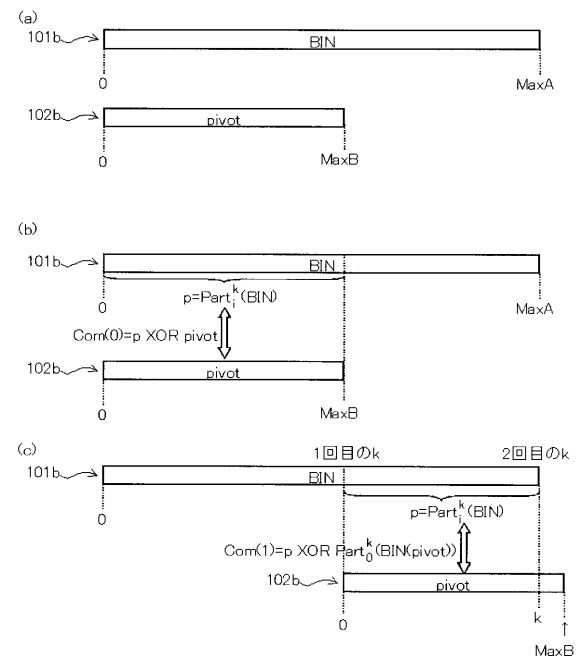
【図 4】



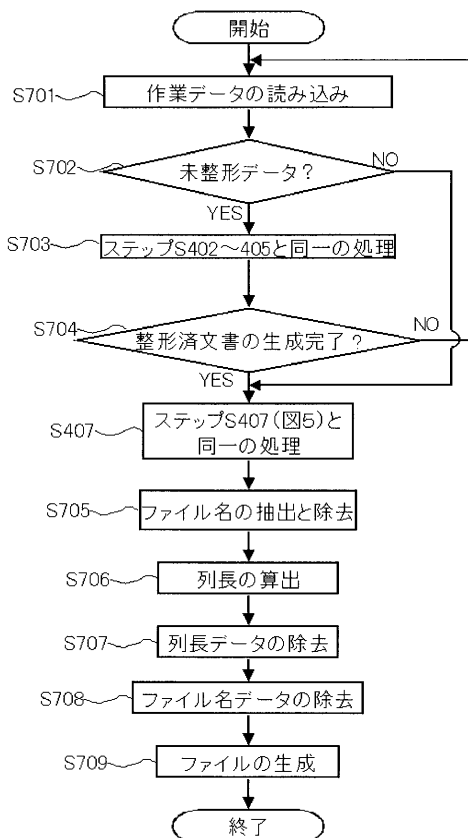
【図5】



【図6】



【図7】



【図8】

