(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4610176号
(P4610176)
(45)発行日　平成23年1月12日(2011.1.12)　(24)登録日　平成22年10月22日(2010.10.22)

(51)Int.Cl.　F I
H04L 9/08 (2006.01)　H04L 9/00 601Z
H04L 9/10 (2006.01)　H04L 9/00 621Z

請求項の数 16　(全 32 頁)

(21)出願番号　特願2003-357910(P2003-357910)
(22)出願日　平成15年10月17日(2003.10.17)
(65)公開番号　特開2004-336702(P2004-336702A)
(43)公開日　平成16年11月25日(2004.11.25)
審査請求日　平成18年9月29日(2006.9.29)
(31)優先権主張番号　特願2003-110876(P2003-110876)
(32)優先日　平成15年4月15日(2003.4.15)
(33)優先権主張国　日本国(JP)

前置審査

(73)特許権者　399035766
エヌ・ティ・ティ・コミュニケーションズ株式会社
東京都千代田区内幸町一丁目１番６号
(74)代理人　100083806
弁理士　三好　秀和
(72)発明者　荻原　利彦
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内
(72)発明者　渡部　勝年
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内

最終頁に続く

(54)【発明の名称】データ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラム

(57)【特許請求の範囲】
【請求項１】
データを、その原本性を確保して保管するデータ原本性確保システムであって、
前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割部と、
分割された複数の分割データをそれぞれ記憶するハードウェア的に独立した複数の保管部と、を備え、
前記秘密分散法は、
前記データを所定の長さ毎に区切って、複数の部分データを生成し、
前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、
前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成する
ことを特徴とするデータ原本性確保システム。
【請求項２】
前記データ分割部は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成部と、
生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割部と

を備えたことを特徴とする請求項１記載のデータ原本性確保システム。
【請求項３】
前記識別データ生成部は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成する生成部を備えたことを特徴とする請求項２記載のデータ原本性確保システム。
【請求項４】
前記データ分割部は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、
前記複数の保管部は、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを特徴とする請求項２又は３記載のデータ原本性確保システム。
【請求項５】
前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、
復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認部と、
を備えたことを特徴とする請求項２又は３記載のデータ原本性確保システム。
【請求項６】
前記複数の保管部から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、
復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第１の確認部と、
前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第２の確認部と、
を備えたことを特徴とする請求項４記載のデータ原本性確保システム。
【請求項７】
前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認部をさらに備えたことを特徴とする請求項１記載のデータ原本性確保システム。
【請求項８】
データを、その原本性を確保して保管するデータ原本性確保方法であって、
前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するステップと、
分割された複数の分割データをハードウェア的に独立した複数の保管装置にそれぞれ記憶するステップと、を備え、
前記秘密分散法は、
前記データを所定の長さ毎に区切って、複数の部分データを生成し、
前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、
前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成する
ことを特徴とするデータ原本性確保方法。
【請求項９】

前記分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成するステップと、
生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割するステップと、
を備えたことを特徴とする請求項８記載のデータ原本性確保方法。
【請求項１０】
データを、その原本性を確保して保管するためのデータ原本性確保用プログラムであって、
コンピュータに、
前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割ステップと、
分割された複数の分割データを、該コンピュータが通信可能なハードウェア的に独立した複数の保管装置にそれぞれ送信して保管させる保管ステップと、を実行させ、
前記秘密分散法は、
前記データを所定の長さ毎に区切って、複数の部分データを生成し、
前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、
前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成する
ことを特徴とするデータ原本性確保用プログラム。
【請求項１１】
前記データ分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成ステップと、
生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割ステップと、
を備えることを特徴とする請求項１０記載のデータ原本性確保用プログラム。
【請求項１２】
前記識別データ生成ステップは、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成することを特徴とする請求項１１記載のデータ原本性確保用プログラム。
【請求項１３】
前記データ分割ステップは、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、
前記保管ステップは、前記複数の保管装置に、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ送信して保管させることを特徴とする請求項１１又は１２記載のデータ原本性確保用プログラム。
【請求項１４】
前記コンピュータに、
前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元ステップと、
復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認ステップと、
を実行させることを特徴とする請求項１１又は１２記載のデータ原本性確保用プログラム。
【請求項１５】
前記コンピュータに、
前記複数の保管装置から前記分割データ、前記データ分割に関する管理情報、および前

記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元ステップと、
復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第１の確認ステップと、
前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第２の確認ステップと、
を実行させることを特徴とする請求項１３記載のデータ原本性確保用プログラム。
【請求項１６】
前記コンピュータに、
前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認ステップを実行させることを特徴とする請求項１０記載のデータ原本性確保用プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、データを、その原本性を確保して保管するためのデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムに関する。
【背景技術】
【０００２】
ＩＴの発展に伴って、自らの端末をインターネット等の通信ネットワークに接続し、この通信ネットワークに接続されている他端末に対して通信ネットワークを介してデータを通信する機会が非常に増えている。
【０００３】
このとき、端末は通信ネットワークに接続されているため、悪意を持った第三者（ハッカー等）が通信ネットワークに接続された第三者側の端末を介して端末にアクセスし、端末に保管されているデータを改竄する恐れが生じており、改竄されたデータの原本性（故意または過失による改竄がなく、内容が完全であること）を確保する必要が生じていた。
【０００４】
この点、従来では、電子署名を用いてデータの改竄を検出可能にして、原本性を確保する技術が導入されている（例えば、非特許文献１参照）。
【０００５】
すなわち、電子署名技術を用いた原本性確保方法によれば、データ送信元のＡ氏は、予め認証局（ＣＡ;Certification Authority）により保証された公開鍵入りの電子証明書を受取る。
【０００６】
そして、Ａ氏は、送信したいデータ（データ本体とも呼ぶ）について電子署名を作成し、作成した電子署名を含むデータ（データ本体＋電子署名＋電子証明書）をＢ氏に送信する。
【０００７】
Ｂ氏は、受信したデータのうち、データ本体から生成したハッシュ値と、Ａ氏から受け取ったＡ氏の電子証明書に含まれる公開鍵で電子署名を復号化して得た値とを比較することにより、データが改竄されたか否かを判断している。
【非特許文献１】”ＰＫＩ関連技術説明”、［online］、平成１４年１２月１２日、情報処理振興事業協会、［平成１５年４月１１日検索］、インターネット＜URL:http://www.ipa.go.jp/security/pki/index.html＞
【発明の開示】
【発明が解決しようとする課題】

【０００８】
しかしながら、電子署名技術を用いた原本性確保方法によれば、Ｂ氏側において受け取ったデータを長期間保管する場合、秘密鍵の危殆化等からデータが改竄される恐れが生じている。
【０００９】
そこで、電子署名技術を用いた原本性確保方法において、認証局から発行される電子証明書には予め有効期限が定められており、有効期限到来前にＡ氏側において改めて秘密鍵と公開鍵とを生成し、公開鍵について電子証明書を認証局から発行してもらい、新しい秘密鍵を用いて再度電子署名付きのデータを作成する必要がある。
【００１０】
しかしながら、上述した電子証明書の有効期限に基づく認証局からの電子証明書の再発行処理、および新しい秘密鍵を用いた電子署名付きデータ作成処理は非常に手間および時間がかかり、また、有効期限毎に上述した電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返し行わなければならず、データを、その原本性をより簡易に確保しながら保管できる方法の開発が待望されていた。
【００１１】
本発明は、上述した事情に鑑みてなされたものであり、データを、その原本性をより簡易に確保して保管することのできるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムを提供することをその目的とする。
【課題を解決するための手段】
【００１２】
上記目的を達成するため、本発明は、請求項１記載に記載したように、データを、その原本性を確保して保管するデータ原本性確保システムであって、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割部と、分割された複数の分割データをそれぞれ記憶するハードウェア的に独立した複数の保管部と、を備え、前記秘密分散法は、前記データを所定の長さ毎に区切って、複数の部分データを生成し、前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成することを要旨とする。
【００１３】
請求項２に記載した発明は、前記データ分割部は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成部と、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割部とを備えたことを要旨とする。
【００１４】
請求項３に記載した発明は、前記識別データ生成部は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成する生成部を備えたことを要旨とする。
【００１５】
請求項４に記載した発明は、前記データ分割部は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、前記複数の保管部は、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを要旨とする。
【００１６】
請求項５に記載した発明は、前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認部と、を備えたことを要旨とする。

【００１７】
請求項６に記載した発明は、前記複数の保管部から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第１の確認部と、前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第２の確認部と、を備えたことを要旨とする。

【００１８】
請求項７に記載した発明は、前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認部をさらに備えたことを要旨とする。

【００１９】
請求項８に記載した本発明は、データを、その原本性を確保して保管するデータ原本性確保方法であって、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するステップと、分割された複数の分割データをハードウェア的に独立した複数の保管装置にそれぞれ記憶するステップと、を備え、前記秘密分散法は、前記データを所定の長さ毎に区切って、複数の部分データを生成し、前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成することを要旨とする。

【００２０】
請求項９に記載した発明は、前記分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成するステップと、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割するステップと、を備えたことを要旨とする。

【００２１】
請求項１０に記載した本発明は、データを、その原本性を確保して保管するためのデータ原本性確保用プログラムであって、コンピュータに、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割ステップと、分割された複数の分割データを、該コンピュータが通信可能なハードウェア的に独立した複数の保管装置にそれぞれ送信して保管させる保管ステップと、を実行させ、前記秘密分散法は、前記データを所定の長さ毎に区切って、複数の部分データを生成し、前記複数の部分データの各々に対応して、前記データと同じ長さの乱数または前記データより短い長さの乱数を前記所定の長さ毎に区切って、複数の乱数部分データを生成し、前記部分データと前記乱数部分データとの排他的論理和演算を行い、前記複数の分割データを生成することを要旨とする。

【００２２】
請求項１１に記載した本発明は、前記分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成ステップと、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割ステップと、を備えることを要旨とする。

【００２３】
請求項１２に記載した本発明は、前記識別データ生成ステップは、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成することを要旨とする。

【００２４】
請求項１３に記載した本発明は、前記データ分割ステップは、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、前記保管ステップは、前記複数の保管装置に、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ送信して保管させることを要旨とする。
【００２５】
請求項１４に記載した本発明は、前記コンピュータに、前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元ステップと、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認ステップと、を実行させることを要旨とする。
【００２６】
請求項１５に記載した本発明は、前記コンピュータに、前記複数の保管装置から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元ステップと、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第１の確認ステップと、前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第２の確認ステップと、を実行させることを要旨とする。
【００２７】
請求項１６に記載した本発明は、前記コンピュータに、前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認ステップを実行させることを要旨とする。
【発明の効果】
【００２８】
本発明に係わるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムによれば、データを、秘密分散法を用いて複数の分割データに分割してハードウェア的に独立した複数の保管部（保管装置）にそれぞれ保管しているため、そのデータの内の少なくとも一部が改竄された場合でも、改竄データを含む分割データから、その分割データに対する改竄の有無を容易に確認することができる。
【００２９】
この結果、例えばデータを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返し行うことなく、簡易にデータを保管することができる。
【発明を実施するための最良の形態】
【００３０】
本発明に係わるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムの実施の形態について、添付図面を参照して説明する。
【００３１】
（第１の実施の形態）
図１は、本発明の第１の実施の形態に係わるデータ原本性確保システム１の概略構成を示すブロック図である。
【００３２】
図１に示すように、データ原本性確保システム１は、インターネット等の通信用ネットワークＮに対して接続して通信できるクライアントコンピュータ（以下、単にクライアン

トとする）２と、ネットワークＮに対して通信可能に接続されており、互いにハードウェア的に独立した複数（本実施の形態では４とする）のデータ保管用サーバコンピュータ（以下、単に保管サーバとする）３ａ１～３ａ４とを備えている。

【００３３】

クライアント２は、演算処理部であるＣＰＵ、このＣＰＵに接続され該ＣＰＵおよびネットワークＮ間の通信を可能にするインタフェース、ＣＰＵに接続されたデータ入力用の入力部およびＣＰＵに接続されたメモリ１０をそれぞれ備えている。

【００３４】

メモリ１０には、原本性を確保して保管する対象となるデータ｛例えば、Ｍ（Ｍは自然数）バイトのデジタルデータ；以下、元データとも呼ぶ｝Ｂが格納されている。また、メモリ１０には、クライアント２（そのＣＰＵ）が読み取り可能であり、後述する図２および図３に示す原本性確保処理（データ保管処理）をクライアント２（そのＣＰＵ）に実行させるためのデータ原本性確保用プログラムＰが搭載されている。

【００３５】

なお、このデータ原本性確保用プログラムＰは、磁気メモリや半導体メモリ等の各種記録媒体に搭載され、必要に応じてその記録媒体からクライアント２に読み出されてメモリ１０にロードされるように構成することも可能である。

【００３６】

クライアント２は、データ原本性確保用プログラムＰにより実現される機能として、メモリ１０に格納された元データＢから、その元データＢをユニークに識別できる例えばハッシュ値を生成するハッシュ値生成部１１と、元データＢおよびハッシュ値Ｈを含むデータを秘密分散法を用いて分割して分割データＤ（１）～Ｄ（４）（本実施形態では、分割数を４とする）を生成する分割データ生成部１３と、分割データＤ（１）～Ｄ（４）から元データＢおよびハッシュ値Ｈを生成する元データ復元部１５と、分割データＤ（１）～Ｄ（４）をネットワークＮを介して通信する通信部１７とを備えている。

【００３７】

各保管サーバ３ａ１～３ａ４は、演算処理部であるＣＰＵ、このＣＰＵに接続され該ＣＰＵおよびネットワークＮ間の通信を可能にするインタフェース、ＣＰＵに接続されたデータ入力用の入力部、ＣＰＵに接続されたメモリおよびハードディスク等の記憶装置をそれぞれ備えている。

【００３８】

次に、本実施の形態に係わるデータ原本性確保システム１の全体処理について説明する。

【００３９】

図２および図３に示すように、クライアント２は、データ原本性確保用プログラムＰに従って動作し、ハッシュ値生成部１１の機能として、メモリ１０に記憶された、原本性を確保して保管したい元データＢをメモリ１０から読み込み、読み込んだ元データＢを、所定のハッシュ関数を用いてその元データＢのハッシュ値Ｈを生成する（ステップＳ１）。

【００４０】

このハッシュ値Ｈは、その元データＢが１ビットでも変更されると全く異なる値を示す性質、すなわち、元データＢをユニークに識別できる性質を有している。

【００４１】

次いで、クライアント２は、分割データ生成部１３の機能として、生成したハッシュ値Ｈを含む元データＢを秘密分散法を用いて４つのデータ（分割データ）Ｄ（１）～Ｄ（４）に分割する（ステップＳ２）。

【００４２】

以下、ステップＳ２の秘密分散法に基づく分割データＤ（１）～Ｄ（４）生成処理について詳細に説明する。

【００４３】

例えば、２次多項式Ｆ（ｘ）＝ａｘ^{2}＋ｂｘ＋Ｂ（ｍｏｄ　ｐ；ｐで割った時の余りを

表す）を基にしたＳｈａｍｉｒの秘密分散法｛（ｋ、ｎ）閾値法；但し、分割数を表すｎを４とし、復元できる数を表すｋを３とする｝で考える。ここでＢは元データ、Ｆ（ｘ）は分割データである。ａ、ｂ、ｐは、元データＢの分割に際して任意に決定される。但し、ｐは、ａ、ｂ、Ｂよりも大きい素数とする。

【００４４】

このとき、クライアント２の分割データ生成処理により、分割データＦ（１）～Ｆ（４）｛上記分割データＤ（１）～Ｄ（４）に対応｝は、次式（１）～（４）のように作成される。

【００４５】

Ｆ（１）＝ａ＋ｂ＋Ｂ（ｍｏｄ　ｐ）　　・・・（１）

Ｆ（２）＝４ａ＋２ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（２）

Ｆ（３）＝９ａ＋３ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（３）

Ｆ（４）＝１６ａ＋４ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（４）

この分割データＦ（１）～Ｆ（４）の内、ｋ＝３以上の分割データ｛例えば、Ｆ（１）、Ｆ（２）、Ｆ（４）｝が集まれば、この分割データ

Ｆ（１）＝ａ＋ｂ＋Ｂ（ｍｏｄ　ｐ）　　・・・（１）

Ｆ（２）＝４ａ＋２ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（２）

Ｆ（４）＝１６ａ＋４ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（４）

を連立して元データＢを求めることができる。

【００４６】

そして、ｋ－１以下の分割データが集まっても、元データＢを復元することはできない。

【００４７】

なお、元データＢが長いデータ列である場合には、クライアント２は、例えば元データＢの先頭から１バイト毎にＦ（１）からＦ（４）を順次作成して分割データＤ（１）（Ｆ（１））～Ｄ（４）（Ｆ（４））を作成する。

【００４８】

そして、クライアント２は、通信部１７の機能として、作成した分割データＤ（１）～Ｄ（４）を保管サーバ３ａ１～３ａ４にネットワークＮを介してそれぞれ送信する（ステップＳ３）。

【００４９】

各保管サーバ３ａ１～３ａ４は、ネットワークＮを介して送信されてきた分割データＤ（１）～Ｄ（４）を、それぞれのハードディスク等の記憶装置に記憶する（ステップＳ５）。

【００５０】

このようにして、元データＢを、そのハッシュ値Ｈを含んで分割された分割データＤ（１）～Ｄ（４）として保管サーバ３ａ１～３ａ４に保管することができる。

【００５１】

次に、保管サーバ３ａ１～３ａ４に保管された分割データＤ（１）～Ｄ（４）が改竄されているか否かを確認する場合、クライアント２は、分割データＤ（１）～Ｄ（４）のダウンロード要求を保管サーバ３ａ１～３ａ４にそれぞれ送信する（ステップＳ１０）。

【００５２】

各保管サーバ３ａ１～３ａ４は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各分割データＤ（１）～Ｄ（４）を各記憶装置から読み出し、読み出した各分割データＤ（１）～Ｄ（４）をネットワークＮを介してクライアント２に送信する（ステップＳ１１）。

【００５３】

クライアント２は、データ原本性確保用プログラムＰに従って動作し、元データ復元部１５の機能として、ネットワークＮを介して送信されてきた分割データＤ（１）～Ｄ（４）を受信し、受信した分割データＤ（１）～Ｄ（４）に基づいて秘密分散法により元デー

タＢ１およびハッシュ値Ｈ１をそれぞれ復元する（ステップＳ１２）。

【００５４】

次いで、クライアント２は、元データ復元部１５の機能として、復元した元データＢ１からハッシュ値Ｈ２を再生成し、再生成したハッシュ値Ｈ２と復元したハッシュ値Ｈ１とを比較して一致するか否かを確認する（ステップＳ１３）。

【００５５】

このステップＳ１２およびＳ１３の処理を具体的に説明する。

【００５６】

例えば、上記分割データＦ（１）、Ｆ（２）、Ｆ（３）、Ｆ（４）の内、一部の分割データが改竄｛例えば、Ｆ（２）～Ｆ（４）がＦａ（２）～Ｆａ（４）に改竄｝されたと仮定する。

【００５７】

このとき、クライアント２は、下式

Ｆ（１）　＝ａ＋ｂ＋Ｂ（ｍｏｄ　ｐ）　　・・・（１）

Ｆａ（２）＝４ａ＋２ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（２）

Ｆａ（３）＝９ａ＋３ｂ＋Ｂ（ｍｏｄ　ｐ）　・・・（３）

Ｆａ（４）＝１６ａ＋４ｂ＋Ｂ（ｍｏｄ　ｐ）・・・（４）

の中から少なくとも３つの式を連立させてハッシュ値Ｈを含む元データＢを復元することになる。

【００５８】

しかしながら、分割データＦ（２）～Ｆ（４）が分割データＦａ（２）～Ｆａ（４）に改竄されているため、この分割データＦａ（２）～Ｆａ（４）を連立しても、復元された元データＢ１は、原本性確保対象となる元データＢとは異なるため、その元データＢ１から再生成されたハッシュ値Ｈ２も、復元されたハッシュ値Ｈ１とは異なる。この結果、保管サーバ３ａ１～３ａ４に保管されている分割データＤ（１）～Ｄ（４）の少なくとも一部が改竄されていることをクライアント２側において確認することができる。

【００５９】

以上述べたように、本実施形態によれば、元データＢを、秘密分散法を用いて複数の分割データＤ（１）～Ｄ（４）に分割して保管サーバ３ａ１～３ａ４にそれぞれ保管しているため、その分割データＤ（１）～Ｄ（４）の内の少なくとも一部が改竄された場合でも、改竄データを含む分割データＤ（１）～Ｄ（４）から、その分割データＤ（１）～Ｄ（４）に対する改竄の有無を容易に確認することができる。

【００６０】

この結果、例えば元データＢを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返し行うことなく、簡易に元データＢを保管することができる。

【００６１】

また、本実施形態によれば、元データＢを分割データＤ（１）～Ｄ（４）に分割して保管サーバ３ａ１～３ａ４に保管しているため、例えば、分割データＤ（１）～Ｄ（４）の内の１つの分割データが改竄された場合でも、残りの分割データを用いて元データＢを復元することができる。

【００６２】

この結果、元データＢの原本性をより確実に確保することができる。

【００６３】

（第２の実施の形態）

図４は、本発明の第２の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保用プログラムに基づく原本性確保処理の一例を示す概略フローチャートである。なお、本実施の形態におけるデータ原本性確保システムの構成については、そのデータ原本性確保用プログラムの内容およびクライアントの処理が異なり、他の構成については、図１に示す構成と略同等であるため、その説明は省略する。

【００６４】
本実施の形態において、クライアント２は、分割データ生成部１３の機能として、メモリ１０に記憶された、原本性を確保して保管したい元データＢをメモリ１０から読み込み、読み込んだ元データＢを秘密分散法を用いて４つのデータ（分割データ）Ｄ（１）～Ｄ（４）に分割する（ステップＳ２１）。
【００６５】
なお、ステップＳ２１の処理は、ステップＳ２の処理と略同等である。
【００６６】
次いで、クライアント２は、通信部１７の機能として、作成した分割データＤ（１）～Ｄ（４）を保管サーバ３ａ１～３ａ４にネットワークＮを介してそれぞれ送信する（ステップＳ２２）。
【００６７】
各保管サーバ３ａ１～３ａ４は、ネットワークＮを介して送信されてきた分割データＤ（１）～Ｄ（４）を、それぞれのハードディスク等の記憶装置に記憶する（ステップＳ２３）。
【００６８】
このようにして、元データＢを、分割データＤ（１）～Ｄ（４）として保管サーバ３ａ１～３ａ４に保管することができる。
【００６９】
次に、保管サーバ３ａ１～３ａ４に保管された分割データＤ（１）～Ｄ（４）が改竄されているか否かを確認する場合、クライアント２は、分割データＤ（１）～Ｄ（４）のダウンロード要求を保管サーバ３ａ１～３ａ４にそれぞれ送信する（ステップＳ３０）。
【００７０】
各保管サーバ３ａ１～３ａ４は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各分割データＤ（１）～Ｄ（４）を、その記憶装置から読み出し、読み出した各分割データＤ（１）～Ｄ（４）をネットワークＮを介してクライアント２に送信する（ステップＳ３１）。
【００７１】
クライアント２は、元データ復元部１５の機能として、ネットワークＮを介して送信されてきた分割データＤ（１）～Ｄ（４）を受信し、受信した分割データＤ（１）～Ｄ（４）を秘密分散法により異なる組合せにより組み合わせて複数の元データＢ１～Ｂｍ（本実施の形態では、ｍ＝２）を復元し、復元した元データＢ１～Ｂ２が互いに一致するか否かを確認する（ステップＳ３２）。
【００７２】
このステップＳ３２の処理を具体的に説明する。
【００７３】
クライアント２は、上記分割データＦ（１）、Ｆ（２）、Ｆ（３）、Ｆ（４）の内、例えば、所定の組合せ、例えば分割データＦ（１）～Ｆ（３）を用いて秘密分散法により元データＢ１を復元し、次いで、上記組合せとは異なる組合せ、例えば、分割データＦ（２）～Ｆ（４）を用いて秘密分散法により元データＢ２を復元する。
【００７４】
このとき、上記分割データＦ（１）～Ｆ（４）の内の一部に改竄が発生した場合、上述したように、復元された元データＢ１およびＢ２は、原本性確保対象となる元データＢとは異なり、かつ元データＢ１およびＢ２は分割データＦ（１）～Ｆ（４）を互いに異なる組合せにより組み合わせているため、復元した元データＢ１およびＢ２は互いに異なる。
【００７５】
したがって、クライアント２は、復元した元データＢ１および元データＢ２との不一致により、保管サーバ３ａ１～３ａ４に保管されている分割データＤ（１）～Ｄ（４）の少なくとも一部が改竄されていることをクライアント２側において確認することができる。
【００７６】

なお、本実施の形態では、分割データＦ（１）～Ｆ（４）の中から異なる組合せとして、分割データＦ（１）～Ｆ（３）と分割データＦ（２）～Ｆ（４）とをそれぞれ選択したが、本発明はこの組合せに限定されるものではなく、４つの分割データＤ（１）～Ｄ（４）それぞれが少なくとも１回復元に使用される複数の組合せを選択すればよく、互いの組合せに基づいて復元された元データが一致しているか否かを確認することができる。

【００７７】

（第３の実施の形態）

図５は、本発明の第３の実施の形態に係わるデータ原本性確保システム４の概略構成を示すブロック図である。

【００７８】

図５に示すように、データ原本性確保システム４は、インターネット等の通信用ネットワークＮに対して接続して通信できる端末５および分割装置６と、ネットワークＮに対して通信可能に接続されており、互いにハードウェア的に独立した複数（本実施の形態では３とする）のデータ保管用サーバコンピュータ（以下、単に保管サーバとする）３ａ１～３ａ３とを備えている。そして、上記システム構成により、分割装置６は、ネットワークＮを介して端末５からのデータ保管要求に応じてデータを複数の分割データに分割し、この分割した複数の分割データをネットワークＮを介して複数の保管サーバ３ａ１～３ａ３に保管するようになっている。尚、端末５と分割装置６との間の通信、および分割装置６と各保管サーバ３ａ１～３ａ３との間の通信は、通信内容の漏洩を防止するため、ＳＳＬ（Secure Socket Layer）、ＩＰ－ＶＰＮ（Virtual Private Network）などにより、通信データが暗号化されている。

【００７９】

端末５および分割装置６は、演算処理部であるＣＰＵ、このＣＰＵに接続され該ＣＰＵおよびネットワークＮ間の通信を可能にするインタフェース、ＣＰＵに接続されたデータ入力用の入力部およびＣＰＵに接続されたメモリをそれぞれ備えている。

【００８０】

端末５は、利用者が原本性を確保して保管するデータ｛例えば、Ｍ（Ｍは自然数）バイトのデジタルデータ；以下、元データとも呼ぶ｝Ｂを分割装置６に送信するとともに、保管したデータを分割装置６から受信するデータ送受信部３１を備えている。

【００８１】

分割装置６のメモリ１０には、端末５から送信された元データＢが格納される。また、メモリ１０には、分割装置６（そのＣＰＵ）が読み取り可能であり、後述する図６および図７に示すデータ原本性確保処理を分割装置６（そのＣＰＵ）に実行させるためのデータ原本性確保用プログラムＰ３が搭載されている。

【００８２】

なお、このデータ原本性確保用プログラムＰ３は、磁気メモリや半導体メモリ等の各種記録媒体に搭載され、必要に応じてその記録媒体から分割装置６に読み出されてメモリにロードされるように構成することも可能である。

【００８３】

分割装置６は、データ原本性確保用プログラムＰ３により実現される機能として、メモリ１０に格納された元データＢから、その元データＢをユニークに識別できる例えばハッシュ値Ｈを生成するとともに、後述するハッシュ値ｈ１～ｈ３を生成するハッシュ値生成部３２と、元データＢおよびハッシュ値Ｈを含むデータを後述する秘密分散法（第１および第２の実施の形態とは異なる秘密分散法）Ｓを用いて分割して分割データ（以下、分割データ本体ともよぶ）Ｄ（１）～Ｄ（３）（本実施形態では、分割数を３とする）を生成するとともに、該分割データ本体Ｄ（１）～Ｄ（３）、ヘッダ情報、および分割データ本体とヘッダ情報の内容全体から生成されたハッシュ値ｈ１～ｈ３から構成された分割データ（以下、送信分割データともよぶ）ＤＤ（１）～ＤＤ（３）を生成する分割データ生成部３３と、分割データ本体Ｄ（１）～Ｄ（３）を生成する際に用いられる乱数データを発生させる乱数発生部３４と、保管サーバ３ａ１～３ａ３から取得した分割データ本体Ｄ（

１）～Ｄ（３）およびそのヘッダ情報からハッシュ値ｈ１～ｈ３を再生成するとともに、復元された元データＢからハッシュ値Ｈを再生成し、データの真正性を確認するハッシュ値確認部３５と、分割データ本体Ｄ（１）～Ｄ（３）から元データＢおよびハッシュ値Ｈを秘密分散法Ｓを用いて生成する元データ復元部３６と、元データＢおよび送信分割データＤＤ（１）～ＤＤ（３）をネットワークＮを介して送受信するデータ送受信部３７とを備えている。

【００８４】

各保管サーバ３ａ１～３ａ３は、演算処理部であるＣＰＵ、このＣＰＵに接続され該ＣＰＵおよびネットワークＮ間の通信を可能にするインタフェース、ＣＰＵに接続されたデータ入力用の入力部およびＣＰＵに接続されたメモリおよびハードディスク等の記憶装置をそれぞれ備えている。

【００８５】

次に、本実施の形態に係わるデータ原本性確保システム４の全体処理を図６および図７を用いて説明する。ここで、図６は、本実施の形態におけるデータ原本性確保用プログラムＰ３に基づくデータ原本性確保処理を示すフローチャートであり、図７は、図６に示すデータ原本性確保処理のデータシーケンスである。

【００８６】

図６および図７に示すように、利用者が保管したい元データＢを指定し、端末５から分割装置６に該元データＢを送信すると、分割装置６は、データ原本性確保用プログラムＰ３に従って動作し、ハッシュ値生成部３２の機能として、メモリ１０に記憶された、原本性を確保して保管したい元データＢをメモリ１０から読み込み、読み込んだ元データＢを、所定のハッシュ関数を用いてその元データＢのハッシュ値Ｈを生成する（ステップＳ１０１，Ｓ１０２）。

【００８７】

このハッシュ値Ｈは、その元データＢが１ビットでも変更されると全く異なる値を示す性質、すなわち、元データＢをユニークに識別できる性質を有している。

【００８８】

次いで、分割装置６は、分割データ生成部３３の機能として、生成したハッシュ値Ｈを含む元データＢを秘密分散法Ｓを用いて３つのデータ（分割データ）Ｄ（１）～Ｄ（３）に分割する（ステップＳ１０３）。そして、各分割データＤ（ｉ）（ｉ＝１，２，３）について、図８に示す形式のデータ、即ち、ハッシュ値ｈｉ（ｉ＝１，２，３）、データ分割に関する管理情報を含むヘッダ情報Ａｉ（ｉ＝１，２，３）、および分割データＤ（ｉ）（ｉ＝１，２，３）から構成される送信分割データＤＤ（ｉ）を生成する（ステップＳ１０４，Ｓ１０５）。

【００８９】

ここで、ハッシュ値ｈｉは、ハッシュ値生成部３２の機能として、ヘッダ情報Ａｉおよび分割データＤ（ｉ）を合わせた内容全体から、所定のハッシュ関数を用いて生成されるものである。尚、ヘッダ情報Ａｉのデータ識別子Ａｉ１には、本実施の形態の秘密分散法Ｓにより分割されたデータであることを示す情報、バージョン情報Ａｉ２には、データフォーマットのバージョンを識別する情報、およびヘッダサイズＡｉ３には、元データサイズから分割データサイズまでの領域の合計データの長さを示す情報が格納されるようになっている。また、元データサイズＡｉ４は、元データ（ハッシュ値Ｈを含む元データＢ）の長さを示す情報、分割数Ａｉ５には、データの分割数（本実施の形態においては３）を示す情報、および処理単位ビット長Ａｉ６には、秘密分散法Ｓによる分割処理において用いられる後述する処理単位ビット長を示す情報が格納されるようになっている。また、乱数データサイズＡｉ７には、秘密分散法Ｓによる分割処理において用いられる乱数データの長さを示す情報、分割データタイプＡｉ８には、いくつ目の分割データであるかを示す情報（例えば、３分割の場合は、分割データがＤ（１），Ｄ（２），Ｄ（３）のいずれかであるかを示す情報）、および分割データサイズＡｉ９には、分割データＤ（ｉ）の長さを示す情報が格納されるようになっている。

【００９０】
そして、分割装置６は、データ送受信部３７の機能として、作成した送信分割データＤＤ（１）～ＤＤ（３）を保管サーバ３ａ１～３ａ３にネットワークＮを介してそれぞれ送信する（ステップＳ１０６）。
【００９１】
各保管サーバ３ａ１～３ａ３は、ネットワークＮを介して送信されてきた送信分割データＤＤ（１）～ＤＤ（３）を、それぞれのハードディスク等の記憶装置に記憶する（ステップＳ１０７）。
【００９２】
このようにして、元データＢをそのハッシュ値Ｈを含んで分割した分割データＤ（１）～Ｄ（３）、ヘッダ情報Ａ１～Ａ３、およびハッシュ値ｈ１～ｈ３から構成される送信分割データＤＤ（１）～ＤＤ（３）をそれぞれ保管サーバ３ａ１～３ａ３に保管することができる。
【００９３】
次に、保管サーバ３ａ１～３ａ３に保管された送信分割データＤＤ（１）～ＤＤ（３）を取り出す場合には、利用者が取り出したい元データＢを指定し、端末５から分割装置６に取り出し指示を送信すると、分割装置６は、送信分割データＤＤ（１）～ＤＤ（３）のダウンロード要求を保管サーバ３ａ１～３ａ３にそれぞれ送信する（ステップＳ１０８，Ｓ１０９）。
【００９４】
各保管サーバ３ａ１～３ａ３は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各送信分割データＤＤ（１）～ＤＤ（３）を各記憶装置から読み出し、読み出した各送信分割データＤＤ（１）～ＤＤ（３）をネットワークＮを介して分割装置６に送信する（ステップＳ１１０）。
【００９５】
分割装置６は、データ原本性確保用プログラムＰ３に従って動作し、ハッシュ値確認部３５の機能として、ネットワークＮを介して送信されてきた送信分割データＤＤ（１）～ＤＤ（３）を受信し、受信した送信分割データＤＤ（１）～ＤＤ（３）の分割データ本体Ｄ（１）～Ｄ（３）およびヘッダ情報Ａ１～Ａ３からハッシュ値ｈ１’～ｈ３’を再生成する（ステップＳ１１１）。そして、再生成されたハッシュ値ｈ１’～ｈ３’と受信した送信分割データＤＤ（１）～ＤＤ（３）のハッシュ値ｈ１～ｈ３とを比較して一致するか否かを確認する（ステップＳ１１２）。これにより、各保管サーバ３ａ１～３ａ３から取得したデータの内容が欠落したり、改竄されていないことを確認することができる。
【００９６】
次に、ステップＳ１１２でハッシュ値が一致する送信分割データＤＤ（１）～ＤＤ（３）が少なくとも２以上ある場合には、元データ復元部３６の機能として、分割データＤ（１）～Ｄ（３）およびヘッダ情報Ａ１～Ａ３に基づいて秘密分散法Ｓにより元データＢ１およびハッシュ値Ｈ１をそれぞれ復元する（ステップＳ１１３）。これは、保管サーバ３ａ１～３ａ３から分割データを取得できない場合やハッシュ値の一致が確認できない場合があっても、データが取得でき、ハッシュ値の一致が確認できたものがデータの復元に必要な個数以上（本実施の形態においては、後述するように２つ以上）あった場合には、後続処理（データ復元処理）を行うものである。
【００９７】
次いで、分割装置６は、ハッシュ値確認部３５の機能として、復元した元データＢ１からハッシュ値Ｈ２を再生成し、再生成したハッシュ値Ｈ２と復元したハッシュ値Ｈ１とを比較して一致するか否かを確認する（ステップＳ１１４，Ｓ１１５）。これにより、復元された元データＢおよびハッシュ値Ｈの内容が欠落したり、改竄されていないことを確認することができる。また、ヘッダ情報Ａ１～Ａ３および分割データ本体Ｄ（１）～Ｄ（３）の内容を改竄した後、改竄した内容に合わせてハッシュ値ｈ１～ｈ３を算出し、書き換えた場合においても（この場合、ステップＳ１１２においてハッシュ値は一致してしまう

）、改竄を検出をすることが可能となる。

【００９８】

そして、分割装置６は、復元された元データＢをネットワークＮを介して端末５に送信し、これにより、端末５は元データＢを取得する（ステップＳ１１６，Ｓ１１７）。

【００９９】

ここで、本実施の形態における秘密分散法Ｓについて詳しく説明する。尚、以後の説明における元データとは、データ分割の対象となる元データ、即ち、本実施の形態においては、ハッシュ値生成部３３で生成されたハッシュ値Ｈを含む元データＢをいう。

【０１００】

本実施形態における元データの分割および復元では、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するが、この場合の処理単位ビット長は任意の値に設定することができ、元データを処理単位ビット長毎に区分けして、この元部分データから分割部分データを分割数より１少ない数ずつ生成するので、元データのビット長が処理単位ビット長の（分割数-1）倍の整数倍に一致しない場合は、元データの末尾の部分に０を埋めるなどして元データのビット長を処理単位ビット長の（分割数-1）倍の整数倍に合わせることにより本実施形態を適用することができる。

【０１０１】

また、上述した乱数も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして乱数発生部３４から生成される。すなわち、乱数は処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして生成される。更に、元データは処理単位ビット長に基づいて所望の分割数の分割データに分割されるが、この分割データの各々も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。すなわち、分割データの各々は、処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。

【０１０２】

なお、以下の説明では、上述した元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれB,R,D,nおよびbで表すとともに、また複数のデータや乱数などのうちの１つを表わす変数としてi(=1～n)およびj(=1～n-1)を用い、（分割数n-1）個の元部分データ、（分割数n-1）個の乱数部分データ、および分割数n個の分割データDのそれぞれのうちの１つをそれぞれB (j),R(j)およびD(i)で表記し、更に各分割データD(i)を構成する複数(n-1)の分割部分データをD(i,j)で表記するものとする。すなわち、B (j)は、元データBの先頭から処理単位ビット長毎に区分けして１番から順に採番した時のj番目の元部分データを表すものである。

【０１０３】

この表記を用いると、元データ、乱数データ、分割データとこれらをそれぞれ構成する元部分データ、乱数部分データ、分割部分データは、次のように表記される。

【０１０４】

$$
\begin{aligned}
&\text{元データ}B=(n-1)\text{個の元部分データ}B(j)\\
&\qquad =B(1),B(2),\cdots,B(n-1)\\
&\text{乱数}R=(n-1)\text{個の乱数部分データ}R(j)\\
&\qquad =R(1),R(2),\cdots,R(n-1)\\
&n\text{個の分割データ}D(i)=D(1),D(2),\cdots,D(n)\\
&\text{各分割部分データ}D(i,j)\\
&\qquad =D(1,1),D(1,2),\cdots,D(1,n-1)\\
&\qquad\ \ D(2,1),D(2,2),\cdots,D(2,n-1)\\
&\qquad\ \ \cdots\quad\ \cdots\quad\ \cdots\\
&\qquad\ \ D(n,1),D(n,2),\cdots,D(n,n-1)\\
&\qquad\qquad\qquad (i=1\sim n),\ (j=1\sim n-1)
\end{aligned}
$$

本実施形態は、上述したように処理単位ビット長毎に区分けされる複数の部分データに対して元部分データと乱数部分データの排他的論理和演算（XOR）を行って、詳しくは、元部分データと乱数部分データの排他的論理和演算（XOR）からなる定義式を用いて、元データの分割を行うことを特徴とするものであり、上述したデータ分割処理に多項式や剰余演算を用いる方法（第１および第２の実施の形態における方法）に比較して、コンピュータ処理に適したビット演算である排他的論理和（XOR）演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを生成することができるとともに、また分割データの保管に必要となる記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。更に、任意に定めた一定の長さ毎にデータの先頭から順に演算処理を行うストリーム処理により分割データが生成される。

【０１０５】

なお、本実施形態で使用する排他的論理和演算（XOR）は、以下の説明では、「＊」なる演算記号で表すことにするが、この排他的論理和演算のビット毎の演算規則での各演算結果は下記のとおりである。

【０１０６】

０＊０の演算結果は０
０＊１の演算結果は１
１＊０の演算結果は１
１＊１の演算結果は０

また、XOR演算は交換法則、結合法則が成り立つ。すなわち、

a*b=b*a
(a*b)*c=a*(b*c)が成り立つことが数学的に証明される。

【０１０７】

また、a*a=0,a*0=0*a=aが成り立つ。

【０１０８】

ここでa,b,cは同じ長さのビット列を表し、0はこれらと同じ長さですべて「0」からなるビット列を表す。

【０１０９】

次に、フローチャートなどの図面も参照して、本実施の形態における秘密分散法Ｓの作用について説明するが、この説明の前に図９乃至１２のフローチャートに示す記号の定義について説明する。

【０１１０】

（１）$\Pi_{i=1}^{n}A(i)$は、A(1)*A(2)*…A(n)を意味するものとする。

【０１１１】

（２）c(j,i,k)を(n-1)×(n-1)行列であるU[n-1,n-1]×(P[n-1,n-1])＾(j-1)のi行k列の値と定義する。

【０１１２】

このときQ(j,i,k)を下記のように定義する。

【０１１３】

c(j,i,k)=1 のとき Q(j,i,k)=R((n-1)×m+k)
c(j,i,k)=0 のとき Q(j,i,k)=0

（３）U[n,n]とは、n×n行列であって、i行j列の値をu(i,j)で表すと、

i+j<=n+1 のとき u(i,j)=1
i+j>n+1 のとき u(i,j)=0

である行列を意味するものとし、「上三角行列」ということとする。具体的には下記のような行列である。

【数１】

$$U[3,3]=\begin{pmatrix}1&1&1\\1&1&0\\1&0&0\end{pmatrix}\qquad U[4,4]=\begin{pmatrix}1&1&1&1\\1&1&1&0\\1&1&0&0\\1&0&0&0\end{pmatrix}$$

（４）P[n,n]とは、n×n行列であって、i行j列の値をp(i,j)で表すと、
j=i+1　　のとき p(i,j)=1
i=n,j=1　のとき p(i,j)=1
上記以外のとき　p(i,j)=0
である行列を意味するものとし、「回転行列」ということとする。具体的には下記のような行列であり、他の行列の右側からかけると当該他の行列の１列目を２列目へ、２列目を３列目へ、…,n-1列目をn列目へ、n列目を１列目へ移動させる作用がある。つまり、行列Pを他の行列に右側から複数回かけると、その回数分だけ各列を右方向へ回転させるように移動させることができる。

【数２】

$$P[3,3]=\begin{pmatrix}0&1&0\\0&0&1\\1&0&0\end{pmatrix}\qquad U[4,4]=\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{pmatrix}$$

（５）A,Bをn×n行列とすると、A×Bとは行列AとBの積を意味するものとする。行列の成分同士の計算規則は通常の数学で用いるものと同じである。

【０１１４】

（６）Aをn×n行列とし、iを整数とすると、A＾iとは行列Aのi個の積を意味するものとする。また、A＾0とは単位行列Eを意味するものとする。

【０１１５】

（７）単位行列E[n,n]とは、n×n行列であって、i行j列の値をe(i,j)で表すと、
i=j　　のとき e(i,j)=1
上記以外のとき e(i,j)=0
である行列を意味するものとする。具体的には下記のような行列である。Aを任意のn×n行列とすると
A×E=E×A=A
となる性質がある。

【数３】

$$E[3,3]=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}\qquad E[4,4]=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

次に、図９に示すフローチャートおよび図１０および図１１に示す具体的データなどを

参照して、まず元データBの分割処理について説明する。

【０１１６】

まず、元データBを分割装置６に供給する（図９のステップＳ２０１）。なお、本例では、元データBは、１６ビットの「10110010 00110111」とする。

【０１１７】

次に、利用者は端末５から分割数nとして３を分割装置６に指示する（ステップＳ２０３）。この分割数nは分割装置６において予め定められた値を用いてもよい。なお、この分割数n=3に従って分割装置６で生成される３個の分割データをD(1),D(2),D(3)とする。この分割データD(1),D(2),D(3)は、すべて元データのビット長と同じ１６ビット長のデータである。

【０１１８】

それから、元データBを分割するために使用される処理単位ビット長bを８ビットと決定し、また元データと同じビット長である１６ビットの乱数Rを乱数発生部３４から取得して生成する（ステップＳ２０５）。この処理単位ビット長bは、利用者が端末５から分割装置６に対して指定してもよいし、または分割装置６において予め定められた値を用いてもよい。なお、処理単位ビット長bは、任意のビット数でよいが、ここでは元データBを割り切れることができる８ビットとしている。従って、上記１６ビットの「10110010 00110111」の元データBは、８ビットの処理単位ビット長で区分けされた場合の２個の元分割データB(1)およびB(2)は、それぞれ「10110010」および「00110111」となる。

【０１１９】

次のステップＳ２０７では、元データBのビット長が8×2の整数倍であるか否かを判定し、整数倍でない場合には、元データBの末尾を０で埋めて、8×2の整数倍に合わせる。なお、本例のように処理単位ビット長bが８ビットおよび分割数nが３に設定された場合における分割処理は、元データBのビット長として１６ビットに限られるものでなく、処理単位ビット長b×(分割数n-1)=8×2の整数倍の元データBに対して有効なものである。

【０１２０】

次に、ステップＳ２０９では、変数m、すなわち上述した整数倍を意味する変数mを０に設定する。本例のように、元データBが処理単位ビット長b×(分割数n-1)=8×2=16ビットである場合には、変数mは０であるが、２倍の３２ビットの場合には、変数mは１となり、３倍の４８ビットの場合には、変数mは２となる。

【０１２１】

次に、元データBの8×2×m+1ビット目から8×2ビット分のデータが存在するか否かが判定される（ステップＳ２１１）。これは、このステップＳ２１１以降に示す分割処理を元データBの変数mで特定される処理単位ビット長b×(分割数n-1)=8×2=16ビットに対して行った後、元データBとして次の１６ビットがあるか否かを判定しているものである。本例のように元データBが１６ビットである場合には、１６ビットの元データBに対してステップＳ２１１以降の分割処理を１回行うと、後述するステップＳ２１９で変数mが+1されるが、本例の元データBでは変数mがm+1の場合に相当する１７ビット以降のデータは存在しないので、ステップＳ２１１からステップＳ２２１に進むことになるが、今の場合は、変数mは０であるので、元データBの8×2×m+1ビット目は、8×2×0+1=1となり、元データBの１６ビットの１ビット目から8×2ビット分にデータが存在するため、ステップＳ２１３に進む。

【０１２２】

ステップＳ２１３では、変数jを１から２(=分割数n-1)まで変えて、元データBの8×(2×m+j-1)+1ビット目から８ビット分(=処理単位ビット長)のデータを元部分データB(2×m+j)に設定し、これにより元データBを処理単位ビット長で区分けした２(分割数n-1)個の元部分データB(1),B(2)を次のように生成する。

【０１２３】

元データB=B(1),B(2)

第１の元部分データB(1)=「10110010」

第２の元部分データB(2)=「00110111」

次に、変数jを１から２(=分割数n-1)まで変えて、乱数部分データR(2×m+j)に乱数発生部３４から発生する８ビットの長さの乱数を設定し、これにより乱数Rを処理単位ビット長で区分けした２(分割数n-1)個の乱数部分データR(1),R(2)を次のように生成する（ステップＳ２１５）。

【０１２４】

乱数R=R(1),R(2)

第１の乱数部分データR(1)=「10110001」

第２の乱数部分データR(2)=「00110101」

次に、ステップＳ２１７において、変数iを１から３(=分割数n)まで変えるとともに、更に各変数iにおいて変数jを１から２(=分割数n-1)まで変えながら、ステップＳ２１７に示す分割データを生成するための元部分データと乱数部分データの排他的論理和からなる定義式により複数の分割データD(i)の各々を構成する各分割部分データD(i,2×m+j)を生成する。この結果、次に示すような分割データDが生成される。

【０１２５】

分割データD

=３個の分割データD(i)=D(1),D(2),D(3)

第１の分割データD(1)

=２個の分割部分データD(1,j)=D(1,1),D(1,2)

=「00110110」,「10110011」

第２の分割データD(2)

=２個の分割部分データD(2,j)=D(2,1),D(2,2)

=「00000011」,「00000010」

第３の分割データD(3)

=２個の分割部分データD(3,j)=D(3,1),D(3,2)

=「10110001」,「00110101」

なお、各分割部分データ(i,j)を生成するためのステップＳ２１７に示す定義式は、本例のように分割数n=3の場合には、具体的には図１１に示す表に記載されているものとなる。図１１に示す表から、分割部分データD(1,1)を生成するための定義式はB(1)*R(1)*R(2)であり、D(1,2)の定義式はB(2)*R(1)*R(2)であり、D(2,1)の定義式はB(1)*R(1)であり、D(2,2)の定義式はB(2)*R(2)であり、D(3,1)の定義式はR(1)であり、D(3,2)の定義式はR(2)である。また、図１１に示す表にはm>0の場合の任意の整数についての一般的な定義式も記載されている。

【０１２６】

このように整数倍を意味する変数m=0の場合について分割データDを生成した後、次に変数mを１増やし（ステップＳ２１９）、ステップＳ２１１に戻り、変数m+1に該当する元データBの１７ビット以降について同様の分割処理を行おうとするが、本例の元データBは１６ビットであり、１７ビット以降のデータは存在しないので、ステップＳ２１１からステップＳ２２１に進み、上述したように生成した分割データD(1),D(2),D(3)を分割装置６のメモリ１０に一時保存して、分割処理を終了する。なお、このように保管された分割データD(1),D(2),D(3)はそれぞれ単独では元データが推測できない。

【０１２７】

ここで、上述した図９のフローチャートのステップＳ２１７における定義式による分割データの生成処理、具体的には分割数n=3の場合の分割データの生成処理について詳しく説明する。

【０１２８】

まず、整数倍を意味する変数m=0の場合には、ステップＳ２１７に示す定義式から各分割データD(i)=D(1)～D(3)の各々を構成する各分割部分データD(i,2×m+j)=D(i,j)(i=1～3,j=1～2)は、次のようになる。

【０１２９】

D(1,1)=B(1)*Q(1,1,1)*Q(1,1,2)
D(1,2)=B(2)*Q(2,1,1)*Q(2,1,2)
D(2,1)=B(1)*Q(1,2,1)*Q(1,2,2)
D(2,2)=B(2)*Q(2,2,1)*Q(2,2,2)
D(3,1)=R(1)
D(3,2)=R(2)

上記の６つの式のうち上から４つの式に含まれるQ(j,i,k)を具体的に求める。

【０１３０】

これはc(j,i,k)を2×2行列であるU[2,2]×(P[2,2])＾(j-1)のi行k列の値としたとき下記のように定義される。

【０１３１】

c(j,i,k)=1 のとき Q(j,i,k)=R(k)
c(j,i,k)=0 のとき Q(j,i,k)=0

ここで、

j=1のときは

【数４】

$$U[2,2]\times(P[2,2])^{(j-1)} = U[2,2]\times(P[2,2])^{0}$$

$$= U[2,2]\times E[2,2]$$

$$= U[2,2]$$

$$= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

j=2のときは

【数５】

$$U[2,2]\times(P[2,2])^{(j-1)} = U[2,2]\times(P[2,2])^{1}$$

$$= U[2,2]\times P[2,2]$$

$$= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

これを用いると、各分割部分データD(i,j)は次のような定義式により生成される。

【０１３２】

D(1,1)=B(1)*Q(1,1,1)*Q(1,1,2)=B(1)*R(1)*R(2)
D(1,2)=B(2)*Q(2,1,1)*Q(2,1,2)=B(2)*R(1)*R(2)
D(2,1)=B(1)*Q(1,2,1)*Q(1,2,2)=B(1)*R(1)*0=B(1)*R(1)
D(2,2)=B(2)*Q(2,2,1)*Q(2,2,2)=B(2)*0*R(2)=B(2)*R(2)

上述した各分割部分データD(i,j)を生成するための定義式は、図１０にも図示されている。

【０１３３】

図１０は、上述したように１６ビットの元データBを８ビットの処理単位ビット長に基づいて分割数n=3で３分割する場合の各データと定義式および各分割部分データから元デ

ータを復元する場合の計算式などを示す表である。

【０１３４】

ここで、上述した定義式により分割データD(1),D(2),D(3)および各分割部分データD(1,1),D(1,2),D(2,1),D(2,2),D(3,1),D(3,2)を生成する過程と定義式の一般形について説明する。

【０１３５】

まず、第１の分割データD(1)に対しては、第１の分割部分データD(1,1)は、上述した定義式B(1)*R(1)*R(2)で定義され、第２の分割部分データD(1,2)は定義式B(2)*R(1)*R(2)で定義される。なお、この定義式の一般形は、D(1,j)に対してはB(j)*R(j)*R(j+1)であり、D(1,j+1)に対してB(j+1)*R(j)*R(j+1)である（jは奇数とする）。定義式に従って計算すると、D(1,1)は00110110, D(1,2)は10110011となるので、D(1)は00110110 10110011である。なお、定義式の一般形は、図１１にまとめて示されている。

【０１３６】

また、第２の分割データD(2)に対しては、D(2,1)はB(1)*R(1)で定義され、D(2,2)はB(2)*R(2)で定義される。この定義式の一般形は、D(2,j)に対してはB(j)*R(j)であり、D(2,j+1)に対してはB(j+1)*R(j+1)である（jは奇数とする）。定義式に従って計算すると、D(2,1)は00000011, D(2,2)は00000010となるので、D(2)は00000011 00000010である。

【０１３７】

更に第３の分割データD(3)に対しては、D(3,1)はR(1)で定義され、D(3,2)はR(2)で定義される。この定義式の一般形は、D(3,j)に対してはR(j)であり、D(3,j+1)に対してはR(3,j+1)である（jは奇数とする）。定義式に従って計算すると、D(3,1)は10110001, D(3,2)は00110101となるので、D(3)は10110001 00110101である。

【０１３８】

上記説明は、B,R,D(1),D(2),D(3)の長さを１６ビットとしたが、データの先頭から上記分割処理を繰り返すことにより、どのような長さの元データBからでも分割データD(1),D(2),D(3)を生成することができる。また、処理単位ビット長bは任意にとることができ、元データBの先頭から順にb×2の長さ毎に上記分割処理を繰り返すことにより任意の長さの元データ、具体的には処理単位ビット長b×2の整数倍の長さの元データに対して適用することができる。なお、元データBの長さが処理単位ビット長b×2の整数倍でない場合は、例えば、データ末尾の部分を０で埋めるなどして元データBの長さを処理単位ビット長b×2の整数倍に合わせることにより上述した本実施形態の分割処理を適用することができる。

【０１３９】

次に、図１０の右側に示す表を参照して、分割データから元データを復元する処理について説明する。

【０１４０】

まず、分割装置６に元データBの復元を要求する。分割装置６は、この元データBの復元要求を受け取ると、この元データBに対応する分割データD(1),D(2),D(3)が保管サーバ３ａ１，３ａ２，３ａ３に保管されていることを記憶しているので、ネットワークＮを介して保管サーバ３ａ１，３ａ２，３ａ３から分割データD(1),D(2),D(3)を取得し、この取得した分割データD(1),D(2),D(3)から次に示すように元データBを復元する。

【０１４１】

まず、分割部分データD(2,1),D(3,1)から第１の元部分データB(1)を次のように生成することができる。

【０１４２】

$$
\begin{aligned}
D(2,1)*D(3,1) &= (B(1)*R(1))*R(1) \\
&= B(1)*(R(1)*R(1)) \\
&= B(1)*0 \\
&= B(1)
\end{aligned}
$$

具体的に計算すると、D(2,1)は00000011, D(3,1)は10110001なので、B(1)は10110010と

なる。

【０１４３】

また、別の分割部分データから次のように第２の元部分データB(2)を生成することができる。

【０１４４】

D(2,2)*D(3,2)=(B(2)*R(2))*R(2)
=B(2)*(R(2)*R(2))
=B(2)*0
=B(2)

具体的に計算すると、D(2,2)は00000010, D(3,2)は00110101なので、B(2)は00110111となる。

【０１４５】

一般に、jを奇数として、

D(2,j)*D(3,j)=(B(j)*R(j))*R(j)
=B(j)*(R(j)*R(j))
=B(j)*0
=B(j)

であるから、D(2,j)*D(3,j)を計算すれば、B(j)が求まる。

【０１４６】

また、一般に、jを奇数として、

D(2,j+1)*D(3,j+1)=(B(j+1)*R(j+1))*R(j+1)
=B(j+1)*(R(j+1)*R(j+1))
=B(j+1)*0
=B(j+1)

であるから、D(2,j+1)*D(3,j+1)を計算すれば、B(j+1)が求まる。

【０１４７】

次に、D(1),D(3)を取得してBを復元する場合には、次のようになる。

【０１４８】

D(1,1)*D(3,1)*D(3,2)=(B(1)*R(1)*R(2))*R(1)*R(2) =B(1)*(R(1)*R(1))*(R(2)*R(2))
=B(1)*0*0
=B(1)

であるから、D(1,1)*D(3,1)*D(3,2)を計算すれば、B(1)が求まる。具体的に計算すると、D(1,1)は00110110, D(3,1)は10110001, D(3,2)は00110101なので、B(1)は10110010となる。

【０１４９】

また同様に、

D(1,2)*D(3,1)*D(3,2)=(B(2)*R(1)*R(2))*R(1)*R(2)
=B(2)*(R(1)*R(1))*(R(2)*R(2))
=B(2)*0*0
=B(2)

であるから、D(1,2)*D(3,1)*D(3,2)を計算すれば、B(2)が求まる。具体的に計算すると、D(1,2)は10110011, D(3,1)は10110001, D(3,2)は00110101なので、B(2)は00110111となる。

【０１５０】

一般に、jを奇数として、

D(1,j)*D(3,j)*D(3,j+1)=(B(j)*R(j)*R(j+1))*R(j)*R(j+1)
=B(j)*(R(j)*R(j))*(R(j+1)*R(j+1))
=B(j)*0*0
=B(j)

であるから、D(1,j)*D(3,j)*D(3,j+1)を計算すれば、B(j)が求まる。

【０１５１】
また、一般に、jを奇数として、

$$D(1,j+1)*D(3,j)*D(3,j+1)=(B(j+1)*R(j)*R(j+1))*R(j)*R(j+1)$$
$$=B(j+1)*(R(j)*R(j))*(R(j+1)*R(j+1))$$
$$=B(j+1)*0*0$$
$$=B(j+1)$$

であるから、D(1,j+1)*D(3,j)*D(3,j+1)を計算すれば、B(j+1)が求まる。
【０１５２】
次に、D(1),D(2)を取得してBを復元する場合には、次のようになる。
【０１５３】

$$D(1,1)*D(2,1)=(B(1)*R(1)*R(2))*(B(1)*R(1))$$
$$=(B(1)*B(1))*(R(1)*R(1))*R(2)$$
$$=0*0*R(2)$$
$$=R(2)$$

であるから、D(1,1)*D(2,1)を計算すれば、R(2)が求まる。具体的に計算すると、D(1,1)は00110110, D(2,1)は00000011なので、R(2)は00110101となる。
【０１５４】
また同様に、

$$D(1,2)*D(2,2)=(B(2)*R(1)*R(2))*(B(2)*R(2))$$
$$=(B(2)*B(2))*R(1)*(R(2)*R(2))$$
$$=0*R(1)*0$$
$$=R(1)$$

であるから、D(1,2)*D(2,2)を計算すれば、R(1)が求まる。具体的に計算すると、D(1,2)は10110011, D(2,2)は00000010なので、R(1)は10110001となる。
【０１５５】
このR(1),R(2)を使用してB(1),B(2)を求める。
【０１５６】

$$D(2,1)*R(1)=(B(1)*R(1))*R(1)$$
$$=B(1)*(R(1)*R(1))$$
$$=B(1)*0$$
$$=B(1)$$

であるから、D(2,1)*R(1)を計算すれば、B(1)が求まる。具体的に計算すると、D(2,1)は00000011, R(1)は10110001なので、B(1)は10110010となる。
【０１５７】
また同様に、

$$D(2,2)*R(2)=(B(2)*R(2))*R(2)$$
$$=B(2)*(R(2)*R(2))$$
$$=B(2)*0$$
$$=B(2)$$

であるからD(2,2)*R(2)を計算すればB(2)が求まる。具体的に計算するとD(2,2)は00000010, R(2)は00110101なので、B(2)は00110111となる。
【０１５８】
一般に、jを奇数として、

$$D(1,j)*D(2,j)=(B(j)*R(j)*R(j+1))*(B(j)*R(j))$$
$$=(B(j)*B(j))*(R(j)*R(j))*R(j+1)$$
$$=0*0*R(j+1)$$
$$=R(j+1)$$

であるからD(1,j)*D(2,j)を計算すればR(j+1)が求まる。
【０１５９】
また同様に、

D(1,j+1)*D(2,j+1)=(B(j+1)*R(j)*R(j+1))*(B(j+1)*R(j+1))
=(B(j+1)*B(j+1))*R(j)*(R(j+1)*R(j+1))
=0*R(j)*0
=R(j)

であるからD(1,j+1)*D(2,j+1)を計算すればR(j)が求まる。

【０１６０】

このR(j),R(j+1)を使用してB(j),B(j+1)を求める。

【０１６１】

D(2,j)*R(j)=(B(j)*R(j))*R(j)
=B(j)*(R(j)*R(j))
=B(j)*0
=B(j)

であるからD(2,j)*R(j)を計算すればB(j)が求まる。

【０１６２】

また同様に、

D(2,j+1)*R(j+1)=(B(j+1)*R(j+1))*R(j+1)
=B(j+1)*(R(j+1)*R(j+1))
=B(j+1)*0
=B(j+1)

であるからD(2,j+1)*R(j+1)を計算すればB(j+1)が求まる。

【０１６３】

上述したように、元データの先頭から処理単位ビット長bに基づいて分割処理を繰り返し行って、分割データを生成した場合には、３つの分割データD(1),D(2),D(3)のすべてを用いなくても、３つの分割データのうち、２つの分割データを用いて上述したように元データを復元することができる。

【０１６４】

本発明の他の方法として、乱数Rのビット長を元データBのビット長よりも短いものを使用して、元データの分割処理を行うことができる。

【０１６５】

すなわち、上述した乱数RはB,D(1),D(2),D(3)と同じビット長のデータとしたが、乱数Rを元データBのビット長より短いものとし、分割データD(1),D(2),D(3)の生成にこの短いビット長の乱数Rを繰り返し用いるものである。

【０１６６】

なお、分割データD(3)は乱数Rのみから生成されるので、分割データD(3)は乱数Rを繰り返して保管しておく必要はない。例えば、元データBのビット長を１６００ビット（200バイト）としたとき、乱数Rは任意にとった１６０ビット（20バイト）のデータの繰り返しとする。つまり、R(1)～R(20)はランダムに生成し、R(21)～R(200)はR(21)=R(1),R(22)=R(2),…,R(40)=R(20),R(41)=R(1),R(42)=R(2),…,R(60)=R(20),R(61)=R(1),R(62)=R(2),…,R(80)=R(20),……,R(181)=R(1),R(182)=R(2),…,R(200)=R(20)とする。

【０１６７】

先の説明では、分割部分データD(3,j)を乱数部分データR(j)と定義してD(3)を生成しているが、D(3,20)まで保管すれば十分である。つまり、D(3)の長さはD(1),D(2)の１０分の１となる。従って、保管すべきデータの総量は先の実施形態では元データBの３倍であるが、この実施形態では２．１倍とすることができる。

【０１６８】

乱数Rにおける繰り返し部分のデータの長さは、短すぎると、D(1)またはD(2)のみからRが解読されてしまうことも考えられるため、適切な長さを選択することが望ましい。

【０１６９】

次に、図１２に示すフローチャートを参照して、分割数がnで、処理単位ビット長がbである場合の一般的な分割処理について説明する。

【０１７０】
まず、元データBを分割装置６に供給する（ステップＳ４０１）。また、利用者は端末５から分割数n(n≧3である任意の整数)を分割装置６に指示する（ステップＳ４０３）。この分割数nは分割装置６において予め定められた値を用いてもよい。処理単位ビット長bを決定する（ステップＳ４０５）。なお、bは０より大きい任意の整数である。次に、元データBのビット長がb×(n-1)の整数倍であるか否かを判定し、整数倍でない場合には、元データBの末尾を０で埋める（ステップＳ４０７）。また、整数倍を意味する変数mを０に設定する（ステップＳ４０９）。

【０１７１】
次に、元データBのb×(n-1)×m+1ビット目からb×(n-1)ビット分のデータが存在するか否かが判定される（ステップＳ４１１）。この判定の結果、データが存在しない場合は、ステップＳ４２１に進むことになるが、今の場合は、ステップＳ４０９で変数mは０に設定された場合であるので、データが存在するため、ステップＳ４１３に進む。

【０１７２】
ステップＳ４１３では、変数jを１からn-1まで変えて、元データBのb×(n-1)×m+j-1)+1ビット目からbビット分のデータを元部分データB((n-1)×m+j)に設定する処理を繰り返し、これにより元データBを処理単位ビット長bで区分けした(n-1)個の元部分データB(1),B(2),…B(n-1)が生成される。

【０１７３】
次に、変数jを１からn-1まで変えて、乱数部分データR((n-1)×m+j)に乱数発生部３４から発生する処理単位ビット長bの乱数を設定し、これにより乱数Rを処理単位ビット長bで区分けしたn-1個の乱数部分データR(1),R(2),…R(n-1)が生成される（ステップＳ４１５）。

【０１７４】
次に、ステップＳ４１７において、変数iを１からnまで変えるとともに、更に各変数iにおいて変数jを１からn-1まで変えながら、ステップＳ４１７に示す分割データを生成するための定義式により複数の分割データD(i)の各々を構成する各分割部分データD(i,(n-1)×m+j)を生成する。この結果、次に示すような分割データDが生成される。

【０１７５】
分割データD
=n個の分割データD(i)=D(1),D(2),…D(n)
第１の分割データD(1)
=n-1個の分割部分データD(1,j)=D(1,1),D(1,2),…D(1,n-1)
第２の分割データD(2)
=n-1個の分割部分データD(2,j)=D(2,1),D(2,2),…D(2,n-1)
… … …
… … …
第nの分割データD(n)
=n-1個の分割部分データD(n,j)=D(n,1),D(n,2),…D(n,n-1)

このように変数m=0の場合について分割データDを生成した後、次に変数mを１増やし（ステップＳ４１９）、ステップＳ４１１に戻り、変数m=1に該当する元データBのb×(n-1)ビット以降について同様の分割処理を行う。最後にステップＳ４１１の判定の結果、元データBにデータがなくなった場合、ステップＳ４１１からステップＳ４２１に進み、上述したように生成した分割データD(1), …,D(n)を分割装置６のメモリ１０に一時保存して、分割処理を終了する。

【０１７６】
以上述べたように、本実施形態によれば、元データＢを、元データＢのハッシュ値Ｈを含めて秘密分散法Ｓにより複数の分割データＤ（１）～Ｄ（３）に分割し、また、分割データＤ（１）～Ｄ（３）、ヘッダ情報Ａ１～Ａ３、並びに分割データＤ（１）～Ｄ（３）およびヘッダ情報Ａ１～Ａ３から算出されたハッシュ値ｈ１～ｈ３から構成される送信分

割データＤＤ（１）～ＤＤ（３）を生成し、該送信分割データＤＤ（１）～ＤＤ（３）を保管サーバ３ａ１～３ａ３にそれぞれ保管しているため、その送信分割データＤＤ（１）～ＤＤ（３）の内の少なくとも一部が改竄された場合でも、改竄データを含む送信分割データＤＤ（１）～ＤＤ（３）から、その送信分割データＤＤ（１）～ＤＤ（３）に対する改竄の有無を容易に確認することができる。

【０１７７】

この結果、例えば元データＢを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返し行うことなく、簡易に元データＢを保管することができる。

【０１７８】

また、本実施形態によれば、元データＢを送信分割データＤＤ（１）～ＤＤ（３）に分割して保管サーバ３ａ１～３ａ３に保管しているため、例えば、送信分割データＤＤ（１）～ＤＤ（３）の内の１つの分割データが改竄された場合でも、残りの分割データを用いて元データＢを復元することができる。

【０１７９】

この結果、元データＢの原本性をより確実に確保することができる

また、第１および第２の実施の形態における秘密分散法は、公開鍵暗号方式の秘密鍵の分割管理などで利用されており、比較的データ容量の小さいデータに対して実用的な方法であるが、本実施の形態における秘密分散法Ｓは、多項式演算・剰余演算などを含む多倍長整数の演算処理を必要としないので、大容量データを多数処理する場合においても簡単かつ迅速にデータの分割および復元を行うことができるという効果を得ることができる。

【０１８０】

なお、第１および第２の実施の形態では、元データＢを分割した分割データＤ（１）～Ｄ（４）を、クライアント２に対してネットワークＮを介して接続された保管サーバ３ａ１～３ａ４に保管し、第３の実施の形態では、元データＢを分割した送信分割データＤＤ（１）～ＤＤ（３）を、分割装置６に対してネットワークＮを介して接続された保管サーバ３ａ１～３ａ３に保管するように構成したが、本発明はこの構成に限定されるものではない。

【０１８１】

例えば、データ原本性確保システム１の変形例を図１３に示す。

【０１８２】

図１３に示すように、変形例に係わるデータ原本性確保システム１Ａは、クライアント側単独でのクライアントシステム２５として構成されており、このクライアントシステム２５は、クライアント２と、このクライアント２に対してＬＡＮ等を介して接続された互いにハードウェア的に独立した複数の保管用記憶装置２０ａ１～２０ａ４とを備えている。

【０１８３】

図１３に示すようにデータ原本性確保システム１Ａを構成しても、元データＢを分割データＤ（１）～Ｄ（４）として複数の保管用記憶装置２０ａ１～２０ａ４に保管することができ、元データＢの原本性を簡易かつ確実に確保しながら保管することができる。

【０１８４】

また、同様に、データ原本性確保システム４の変形例として、端末５、分割装置６、および保管サーバ３ａ１～３ａ３をクライアント側単独でのクライアントシステムとして構成してもよいものである。

【０１８５】

さらに、上記実施の形態で説明した各システム構成は、あくまでも好適な一具体例を示したものであり、機能的に同一であるのならば、他のシステム構成、例えば、データ原本性確保システム１のシステム構成をデータ原本性確保システム４に示すようなシステム構成としてもよい。これは、データ原本性確保システム１のクライアント２の機能を端末５と分割装置６に機能分散したシステム構成とするものである。同様にして、例えば、デー

タ原本性確保システム４のシステム構成をデータ原本性確保システム１に示すようなシステム構成としてもよい。これは、データ原本性確保システム４の端末５と分割装置６の機能を集約してクライアント２の機能とするものである。

【０１８６】

また、上記実施の形態、ならびにその変形例によれば、秘密分散法としてＳｈａｍｉｒの秘密分散法｛（ｋ、ｎ）閾値法；但し、分割数を表すｎを４とし、復元できる数を表すｋを３とする｝、および秘密分散法Ｓを用いたが、本発明はこの構成に限定されるものではなく、上記実施の形態で述べた秘密分散法以外の他の秘密分散法を用いることも可能である。

【０１８７】

さらに、上記実施の形態、ならびにその変形例によれば、元データのユニークに識別できる識別データとしてハッシュ値を用いたが、ハッシュ値以外の識別データ、例えばパリティ、チェックサム（元データをブロックに分割し、分割されたブロックを数値とみなして全てのブロックを合計した値）等であってもよい。

【図面の簡単な説明】

【０１８８】

【図１】本発明の第１の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図２】本発明の第１の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保用プログラムに基づくデータ原本性確保処理の一例を示す概略フローチャート。

【図３】図２に示すデータ原本性確保処理におけるデータシーケンスを概略的に示す図。

【図４】本発明の第２の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図５】本発明の第３の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図６】本発明の第３の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保用プログラムに基づくデータ原本性確保処理の一例を示す概略フローチャート。

【図７】図６に示すデータ原本性確保処理におけるデータシーケンスを概略的に示す図。

【図８】本発明の第３の実施の形態における分割データの構成を示す図。

【図９】本発明の第３の実施の形態における分割数n=3の場合の分割処理を示すフローチャート。

【図１０】本発明の第３の実施の形態において、１６ビットの元データを８ビットの処理単位ビット長に基づいて分割数n=3で３分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表。

【図１１】本発明の第３の実施の形態において、分割数n=3の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表。

【図１２】本発明の第３の実施の形態における分割数がnで単位ビット長がbである場合の一般的な分割処理を示すフローチャート。

【図１３】本発明の変形例に係わるデータ原本性確保システムの概略構成を示すブロック図。

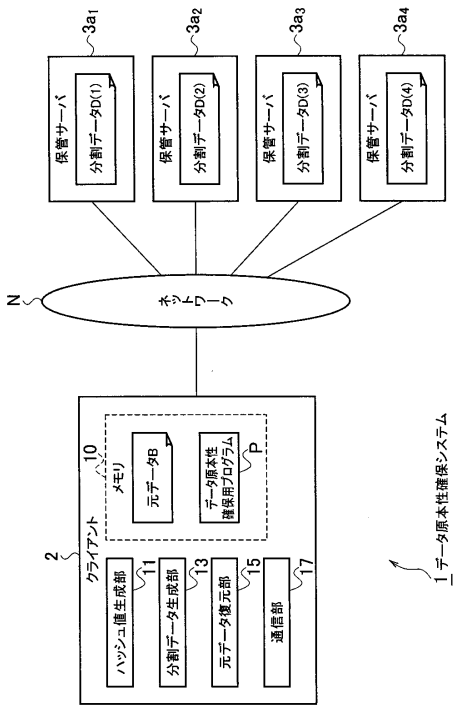
【符号の説明】

【０１８９】

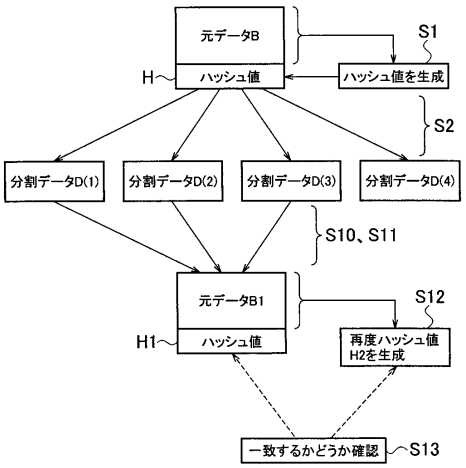
１、１Ａ、４…データ原本性確保システム
２…クライアント
３ａ１～３ａ４…保管サーバ
５…端末
６…分割装置
１０…メモリ
１１…ハッシュ値生成部
１３…分割データ生成部

１５…元データ復元部
１７…通信部
２０ａ１～２０ａ４…保管用記憶装置
２５…クライアントシステム
３１、３７…データ送受信部
３２…ハッシュ値生成部
３３…分割データ生成部
３４…乱数発生部
３５…ハッシュ値確認部
３６…元データ復元部
３７…データ送受信部

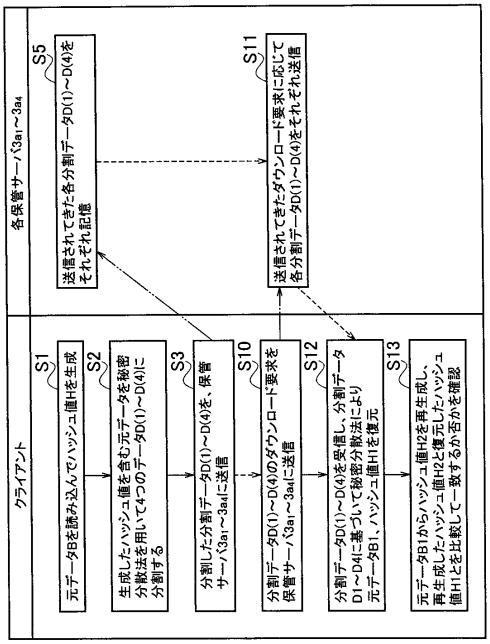
【図１】



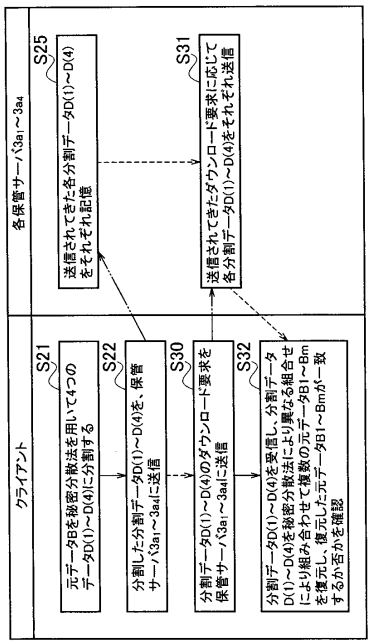
【図２】



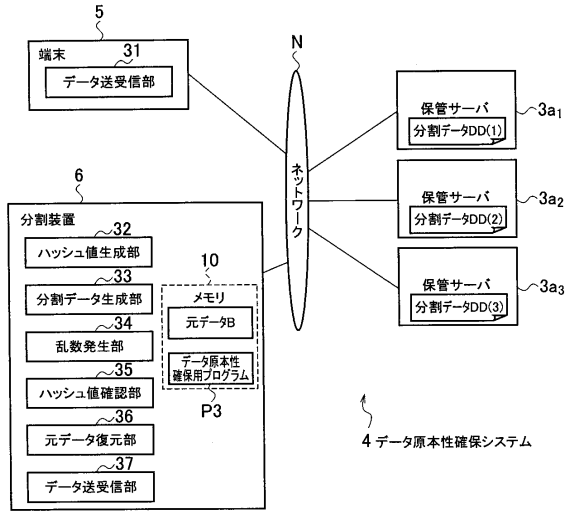
【図３】


クライアント
各保管サーバ3a1~3a4
S1
元データBを読み込んでハッシュ値Hを生成
S2
生成したハッシュ値を含む元データを秘密分散法を用いてn個のデータD(1)~D(n)に分割する
S3
分割した分割データD(1)~D(n)を、保管サーバ3a1~3a4に送信
S5
送信されてきた各分割データD(1)~D(4)をそれぞれ記憶
S10
分割データD(1)~D(4)のダウンロード要求を保管サーバ3a1~3a4に送信
S11
送信されてきたダウンロード要求に応じて各分割データD(1)~D(4)をそれぞれ送信
S12
分割データD(1)~D(4)を受信し、分割データD1~D4に基づいて秘密分散法により元データB1、ハッシュ値H1を復元
S13
元データB1からハッシュ値H2を再生成し、再生成したハッシュ値H2と復元したハッシュ値H1とを比較して一致するか否かを確認

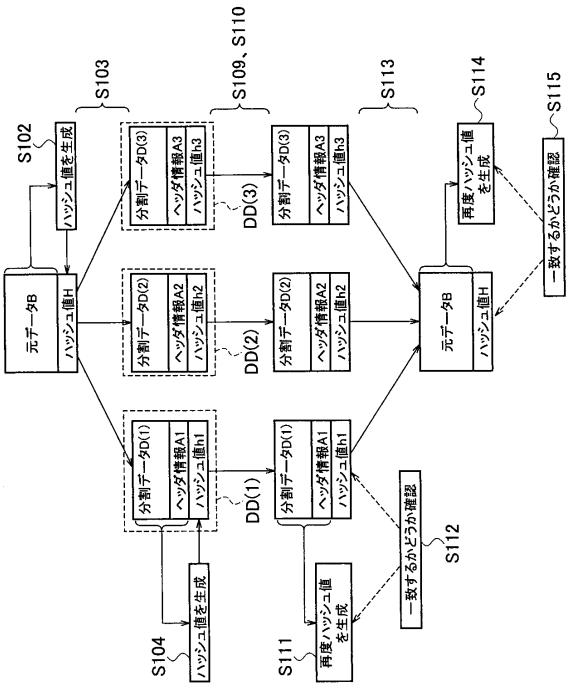
【図４】


クライアント
各保管サーバ3a1~3a4
S21
元データBを秘密分散法を用いてn個のデータD(1)~D(n)に分割する
S22
分割した分割データD(1)~D(n)を、保管サーバ3a1~3a4に送信
S25
送信されてきた各分割データD(1)~D(4)をそれぞれ記憶
S30
分割データD(1)~D(4)のダウンロード要求を保管サーバ3a1~3a4に送信
S31
送信されてきたダウンロード要求に応じて各分割データD(1)~D(4)をそれぞれ送信
S32
分割データD(1)~D(4)を受信し、分割データD(1)~D(4)を秘密分散法により異なる組合せにより組み合わせて複数の元データB1~Bmを復元し、復元した元データB1~Bmが一致するか否かを確認

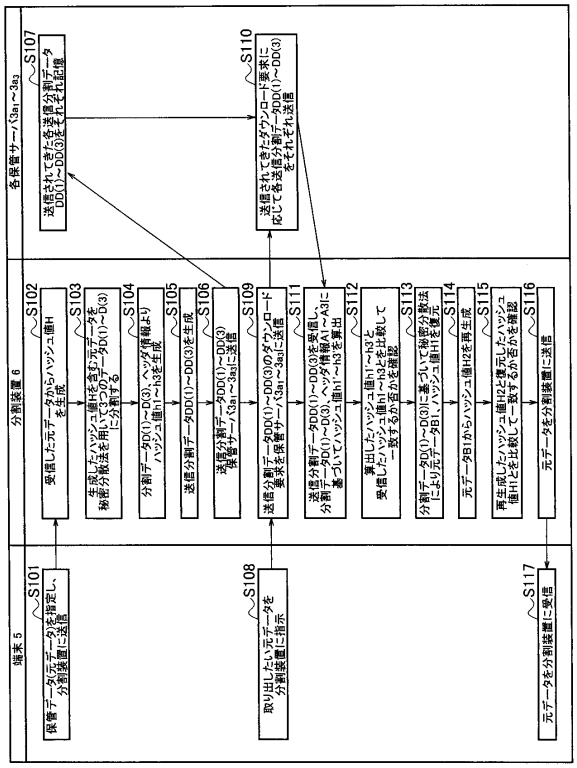
【図５】


5
端末
31
データ送受信部
N
ネットワーク
保管サーバ
分割データDD(1)
3a1
保管サーバ
分割データDD(2)
3a2
保管サーバ
分割データDD(3)
3a3
6
分割装置
32
ハッシュ値生成部
33
分割データ生成部
34
乱数発生部
35
ハッシュ値確認部
36
元データ復元部
37
データ送受信部
10
メモリ
元データB
データ原本性確保用プログラム
P3
4 データ原本性確保システム

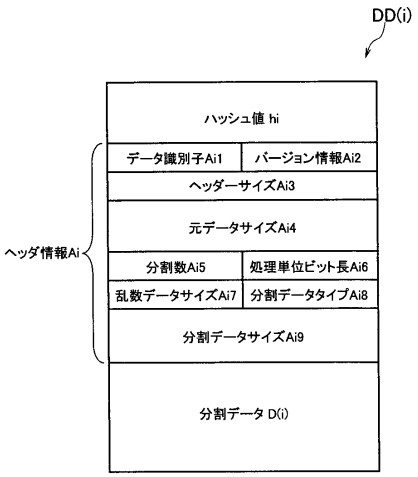
【図６】


S102
ハッシュ値を生成
S103
S104
ハッシュ値を生成
S109、S110
DD(1)
DD(2)
DD(3)
S111
再度ハッシュ値を生成
S112
一致するかどうか確認
S113
S114
再度ハッシュ値を生成
S115
一致するかどうか確認
元データB
ハッシュ値H
分割データD(1)
ヘッダ情報A1
ハッシュ値h1
分割データD(2)
ヘッダ情報A2
ハッシュ値h2
分割データD(3)
ヘッダ情報A3
ハッシュ値h3

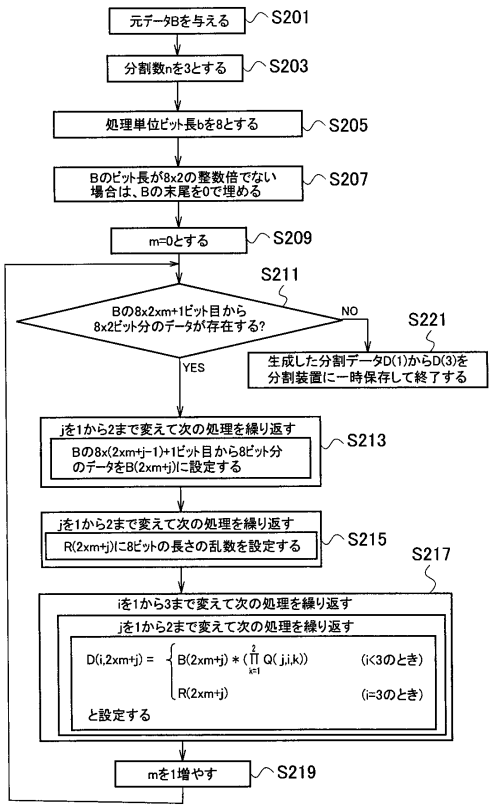
【図７】


各保管サーバ3a1～3a3
S107
送信されてきた全送信分割データDD(1)～DD(3)をそれぞれ記憶
S110
送信されてきたダウンロード要求に応じて各送信分割データDD(1)～DD(3)をそれぞれ送信
分割装置 6
S102
受信した元データからハッシュ値Hを生成
S103
生成したハッシュ値Hを含む元データを秘密分散法を用いてn個のデータD(1)～D(3)に分割する
S104
分割データD(1)～D(3)、ヘッダ情報よりハッシュ値h1～h3を生成
S105
送信分割データDD(1)～DD(3)を生成
S106
送信分割データDD(1)～DD(3)を保管サーバ3a1～3a3に送信
S109
送信分割データDD(1)～DD(3)のダウンロード要求を保管サーバ3a1～3a3に送信
S111
送信分割データDD(1)～DD(3)を受信し、分割データD(1)～D(3)、ヘッダ情報Ai1～Ai3に基づいてハッシュ値h1'～h3'を算出
S112
算出したハッシュ値h1'～h3'と受信したハッシュ値h1～h3とを比較して一致するか否かを確認
S113
分割データD(1)～D(3)に基づいて秘密分散法により元データBを復元
S114
元データBからハッシュ値H2を再生成
S115
再生成したハッシュ値H2と復元したハッシュ値H1とを比較して一致するか否かを確認
S116
元データを分割装置に送信
端末 5
S101
保管データ、元データ名を指定し、分割装置に送信
S108
取り出したい元データを分割装置に指示
S117
元データを分割装置に受信

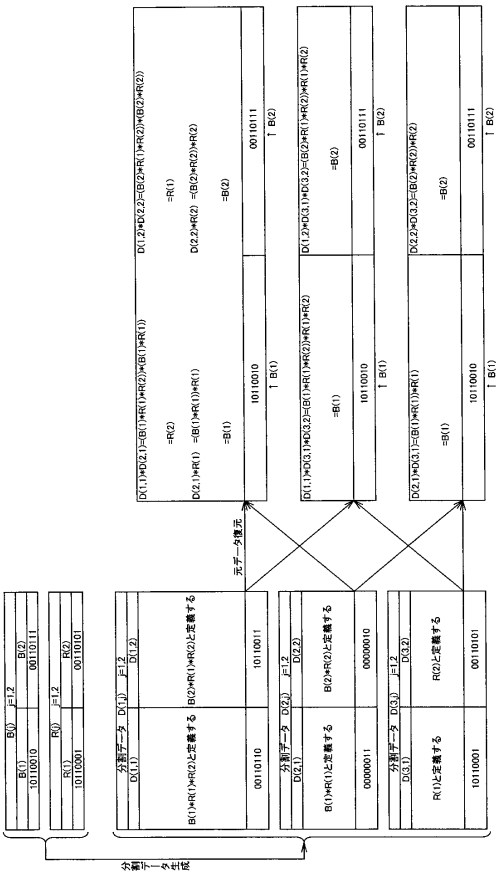
【図８】


DD(i)
ハッシュ値 hi
データ識別子Ai1
バージョン情報Ai2
ヘッダーサイズAi3
元データサイズAi4
分割数Ai5
処理単位ビット長Ai6
乱数データサイズAi7
分割データタイプAi8
分割データサイズAi9
分割データ D(i)
ヘッダ情報Ai

【図９】


元データBを与える
S201
分割数nを3とする
S203
処理単位ビット長bを8とする
S205
Bのビット長が8x2の整数倍でない場合は、Bの末尾を0で埋める
S207
m=0とする
S209
S211
Bの8x2xm+1ビット目から8x2ビット分のデータが存在する?
NO
S221
生成した分割データD(1)からD(3)を分割装置に一時保存して終了する
YES
jを1から2まで変えて次の処理を繰り返す
Bの8x(2xm+j-1)+1ビット目から8ビット分のデータをB(2xm+j)に設定する
S213
jを1から2まで変えて次の処理を繰り返す
R(2xm+j)に8ビットの長さの乱数を設定する
S215
S217
iを1から3まで変えて次の処理を繰り返す
jを1から2まで変えて次の処理を繰り返す
D(i,2xm+j) = B(2xm+j) * (∏ Q(j,i,k)) (i<3のとき)
R(2xm+j) (i=3のとき)
と設定する
mを1増やす
S219

【図１０】


元データ復元
分割データ

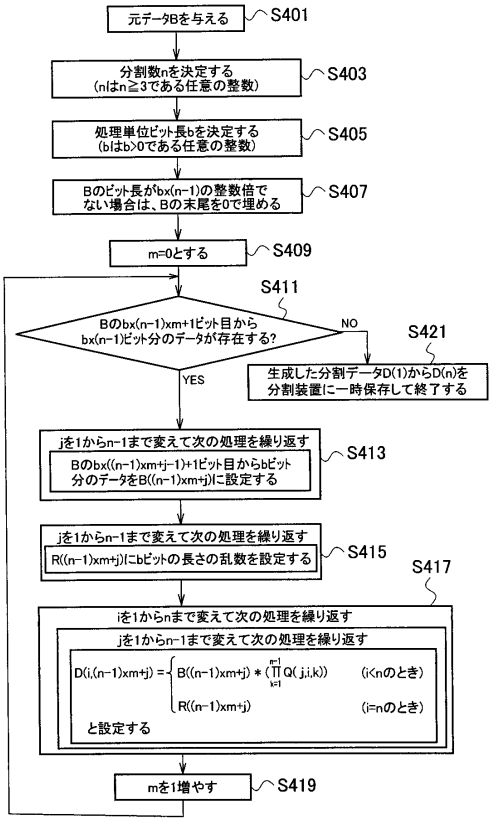
【図１１】

3分割 (n=3)
任意の2つの分割データから元データが復元可能。

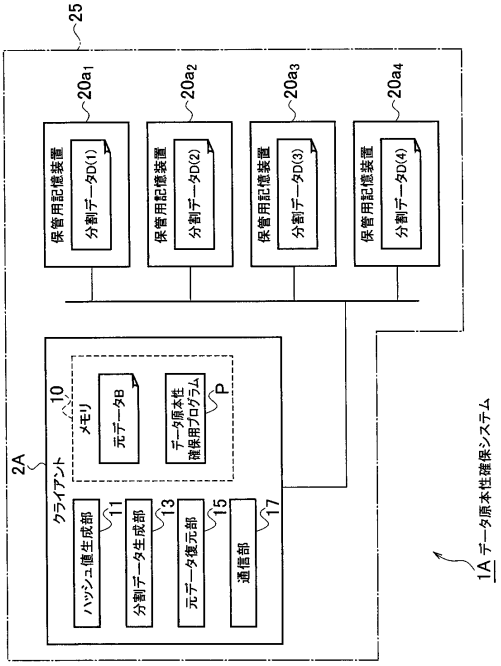
jの値		1	2	…	j=2×m+1	j+1	…
元データ	B(j)	B(1)	B(2)	…	B(j)	B(j+1)	…
乱数	R(j)	R(1)	R(2)	…	R(j)	R(j+1)	…
分割データD(1,j)		B(1)*R(1)	B(2)*R(2)	…	B(j)*R(j)	B(j+1)*R(j+1)	…
分割データD(2,j)		B(1)*R(1)	B(2) *R(2)	…	B(j)*R(j)	B(j+1) *R(j+1)	…
分割データD(3,j)		R(1)	R(2)	…	R(j)	R(j+1)	…

(mは$m>0$の任意の整数)　→元データBの末尾まで続く

【図１２】



【図１３】



フロントページの続き

(72)発明者　野村　進
東京都千代田区内幸町一丁目１番６号　エヌ・ティ・ティ・コミュニケーションズ株式会社内

審査官　速水　雄太

(56)参考文献　特開２００２－２１７８９１（ＪＰ，Ａ）
特開昭６０－２４７６８３（ＪＰ，Ａ）
特開２００２－３３０１２９（ＪＰ，Ａ）
特開２００２－１３５２４７（ＪＰ，Ａ）
特開２００１－０８６１１０（ＪＰ，Ａ）
特開２００１－００５７８１（ＪＰ，Ａ）
米国特許第０５９９１４１４（ＵＳ，Ａ）
国際公開第００／０４５３５８（WO，Ａ１）
特開２００３－１８８８６７（ＪＰ，Ａ）
尾形わかは，岡田光司，黒澤馨，秘密分散共有法（特集 ネットワークシステムと暗号 電子化社会の危機管理），Computer Today，株式会社サイエンス社，１９９８年　7月　1日，第１５巻，第4号，第１８－２３頁
岡本栄司，暗号理論入門，共立出版株式会社，１９９３年　2月２５日，第１２９－１３２頁
Martin Tompa and Heather Woll, How to Share a Secret with Cheaters, Advances in Cryptology - CRYPTO '86, LNCS 263, １９８７年, pp. 261-265
Ernest F. Brickell, Douglas R. Stinson, The Detection of Cheaters in Threshold Schemes, Advances in Cryptology - CRYPTO '88, LNCS Vol.403, １９９０年, pp. 564-577

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｌ　　９／０８
Ｈ０４Ｌ　　９／１０