(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2011-109510
(P2011-109510A)
(43)公開日　平成23年6月2日(2011.6.2)

(51)Int.Cl.		F I			テーマコード（参考）
H04L 9/32	(2006.01)	H04L	9/00	675B	5B017
G06F 21/24	(2006.01)	G06F	12/14	560C	5J104

審査請求 未請求　請求項の数 5　ＯＬ　（全 38 頁）

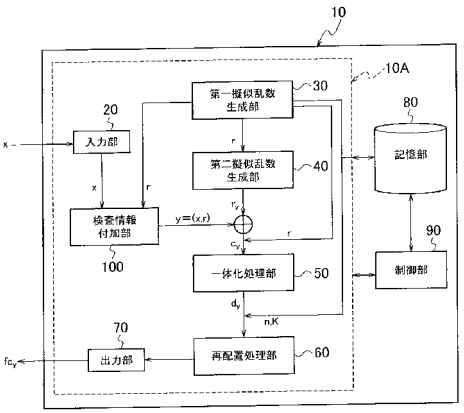
(21)出願番号　特願2009-263838 (P2009-263838)
(22)出願日　平成21年11月19日 (2009.11.19)

(71)出願人　800000068
学校法人東京電機大学
東京都千代田区神田錦町２－２
(74)代理人　100083806
弁理士　三好　秀和
(74)代理人　100100712
弁理士　岩▲崎▼　幸邦
(74)代理人　100095500
弁理士　伊藤　正和
(74)代理人　100101247
弁理士　高橋　俊一
(74)代理人　100098327
弁理士　高松　俊雄

最終頁に続く

(54)【発明の名称】原本性保証装置、原本性保証プログラム、及びこのプログラムを記録する記録媒体

(57)【要約】
【課題】第三者機関による認証を不要とし、量子コンピュータによる攻撃にも耐性を有する電子文章の原本性保証装置及び暗号プログラムを提供する。
【解決手段】電子文章ｘを入力する入力部２０と、秘密鍵Ｋとヘッダｒとをそれぞれ擬似乱数列として生成する第一擬似乱数生成部３０と、電子文章ｘに検査情報としてヘッダｒを付加して新たに電子文章ｙを生成する検査情報付加部１００と、ヘッダｒを初期値として擬似乱数列r_yを生成する第二擬似乱数生成部４０と、電子文章ｙと擬似乱数列r_yとの排他的論理和をとった結果c_yとヘッダｒとの組（ｒ，c_y）を対象化一体化関数系を用いて一体化する一体化処理部５０と、一体化されたデータ$d_y=S(r, c_y)$をｎ個のブロックb_iに分割し、秘密鍵Ｋを用いてｎ個のブロックb_iを再配置することにより暗号文fc_yを生成する再配置処理部６０と、暗号文fc_yを出力する出力部７０とを備える。
【選択図】図２



【特許請求の範囲】
【請求項１】
電子文章を入力する入力部と、
第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成部と、
前記第一の擬似乱数列を初期値として第三の擬似乱数列を生成する公開可能な擬似乱数生成部と、
前記電子文章に検査情報として前記第一の擬似乱数列を付加した第一の電子データを生成する検査情報付加部と、
前記第一の電子データと前記第三の擬似乱数列とを排他的論理和した第二の電子データに前記第一の擬似乱数列をヘッダとして付加した第三の電子データを対称化一体化関数系を用いて一体化して第四の電子データを生成する一体化処理部と、
前記第四の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第五の電子データを暗号文として生成する再配置処理部と、
を有する暗号文生成部と、
暗号文を受信する受信部と、
前記暗号文を所定の分割数に分割し、秘密鍵としての前記再配置表を用いて分割したデータを元の配置に戻して第六の電子データを生成する逆再配置処理部と、
前記対称化一体化関数系の逆関数系を用いて前記第六の電子データを逆一体化して第七の電子データを生成し、前記第七の電子データを先頭から所定のデータ長を有する第八の電子データと残りの第九の電子データとに分離する逆一体化処理部と、
前記第八の電子データを初期値として前記公開可能な擬似乱数生成部にて生成された擬似乱数列と前記第九の電子データとを排他的論理和した第十の電子データを後ろから前記所定のデータ長を有する第十一の電子データと残りの第十二の電子データとに分離し、前記第十一の電子データと前記逆一体化処理部から出力された前記第八の電子データとを比較して、両者の値が一致した場合に前記第十二の電子データの原本性を保証する検査情報検証部と、
を有する復号検証部と、
を備え、
前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証装置。
【請求項２】
電子文章を入力する入力部と、
第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成部と、
前記第一の擬似乱数列を鍵として前記電子文章をブロック暗号化して第一の電子データを生成するブロック暗号文生成部と、
前記第一の電子データに前記第一の擬似乱数列をヘッダとして付与した第二の電子データを対称化一体化関数系を用いて一体化して第三の電子データを生成する一体化処理部と、
前記第三の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第四の電子データを暗号文として生成する再配置処理部と、
を有する暗号文生成部と、
暗号文を受信する受信部と、
前記暗号文を所定の分割数に分割し、秘密鍵としての所定の再配置表を用いて分割されたデータを元の配置して第五の電子データを生成する逆再配置処理部と、
前記対称化一体化関数系の逆関数系を用いて前記第五の電子データを逆一体化して第六の電子データを生成し、前記第六の電子データを先頭から所定のデータ長を有する第七の電子データと残りの第八の電子データと分離する逆一体化処理部と、

前記秘密の擬似乱数生成部にて生成された第一の擬似乱数列を鍵として用いることにより前記第八の電子データをブロック復号化した第九の電子データを後ろから前記所定のデータ長を有する第十の電子データと残りの第十一の電子データとに分離し、前記第十の電子データと前記逆一体化処理部から出力された前記第七の電子データとを比較して、両者の値が一致した場合に前記第十一の電子データの原本性を保証する検査情報検証部と、
を有する復号検証部と、
を備え、
前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証装置。

【請求項３】
コンピュータを、
電子文章を入力する入力手段と、
第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成手段と、
前記第一の擬似乱数列を初期値として第三の擬似乱数列を生成する公開可能な擬似乱数生成手段と、
前記電子文章に検査情報として前記第一の擬似乱数列を付加した第一の電子データを生成する検査情報付加手段と、
前記第一の電子データと前記第三の擬似乱数列とを排他的論理和した第二の電子データに前記第一の擬似乱数列をヘッダとして付加した第三の電子データを対称化一体化関数系を用いて一体化して第四の電子データを生成する一体化処理手段と、
前記第四の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第五の電子データを暗号文として生成する再配置処理手段と、
を有する暗号文生成手段と、
暗号文を受信する受信手段と、
前記暗号文を所定の分割数に分割し、秘密鍵としての前記再配置表を用いて分割したデータを元の配置に戻して第六の電子データを生成する逆再配置処理手段と、
前記対称化一体化関数系の逆関数系を用いて前記第六の電子データを逆一体化して第七の電子データを生成し、前記第七の電子データを先頭から所定のデータ長を有する第八の電子データと残りの第九の電子データとに分離する逆一体化処理手段と、
前記第八の電子データを初期値として前記公開可能な擬似乱数生成手段にて生成された擬似乱数列と前記第九の電子データとを排他的論理和した第十の電子データを後ろから前記所定のデータ長を有する第十一の電子データと残りの第十二の電子データとに分離し、前記第十一の電子データと前記逆一体化処理手段から出力された前記第八の電子データとを比較して、両者の値が一致した場合に前記第十二の電子データの原本性を保証する検査情報検証手段と、
を有する復号検証手段と、
して機能させ、
前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証プログラム。

【請求項４】
コンピュータを、
電子文章を入力する入力手段と、
第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成手段と、
前記第一の擬似乱数列を鍵として前記電子文章をブロック暗号化して第一の電子データを生成するブロック暗号文生成手段と、
前記第一の電子データに前記第一の擬似乱数列をヘッダとして付与した第二の電子データを対称化一体化関数系を用いて一体化して第三の電子データを生成する一体化処理手段

と、
前記第三の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第四の電子データを暗号文として生成する再配置処理手段と、
を有する暗号文生成手段と、
暗号文を受信する受信手段と、
前記暗号文を所定の分割数に分割し、秘密鍵としての所定の再配置表を用いて分割されたデータを元の配置して第五の電子データを生成する逆再配置処理手段と、
前記対称化一体化関数系の逆関数系を用いて前記第五の電子データを逆一体化して第六の電子データを生成し、前記第六の電子データを先頭から所定のデータ長を有する第七の電子データと残りの第八の電子データと分離する逆一体化処理手段と、
前記秘密の擬似乱数生成手段にて生成された第一の擬似乱数列を鍵として用いることにより前記第八の電子データをブロック復号化した第九の電子データを後ろから前記所定のデータ長を有する第十の電子データと残りの第十一の電子データとに分離し、前記第十の電子データと前記逆一体化処理手段から出力された前記第七の電子データとを比較して、両者の値が一致した場合に前記第十一の電子データの原本性を保証する検査情報検証手段と、
を有する復号検証手段と、
して機能させ、
前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証プログラム。
【請求項５】
請求項３又は４項に記載の原本性保証プログラムが記録されたコンピュータが読み取り可能な記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、電子文章の原本性を保証する装置、電子文章の原本性の保証をコンピュータに実行させるプログラム、及びこのプログラムを記録する記録媒体に関する。
【背景技術】
【０００２】
インターネット上での安全な商取引を行うためのインフラとして現在注目されているのが、公開鍵暗号方式に基づくＰＫＩ（Public Key Infrastructure）である。ＰＫＩでは、電子文章の原本性を保証するために、電子文章に公開鍵暗号によるデジタル署名がなされる（非特許文献１）。この技術では、図１６に示すように、送信装置は、送信者が送りたい電子文章からＭＤ５やＳＨＡ１などのハッシュ関数を用いて第一のハッシュ値を生成し、それを秘密鍵で暗号化して、暗号化された第一のハッシュ値を電子文書に添付した電子データを受信装置（受信者）に送信する。受信装置は受け取った電子データの第一のハッシュ値を送信装置の公開鍵を用いて復号化すると共に、受け取った電子データの電子文章から先ほどのハッシュ関数を用いて第二のハッシュ値を生成する。すると、送信装置から受信装置に至る経路において第三者により電子文章が改竄された場合には、受信装置で復号した第一のハッシュ値と受信装置において生成した第二のハッシュ値とが一致しない。逆に言えば、この状況において、第一のハッシュ値と第二のハッシュ値とが一致すれば、送信者にとって電子文章の原本性が保証されたことになる。
【０００３】
しかしながら、この技術には、以下に示す３つの課題がある。
【０００４】
［課題１］この技術では、原本性の保証にハッシュ関数が用いられているが、近年、ハッシュ関数の多くが偽造可能であることが報告されている（非特許文献２及び非特許文献

３）。このことはハッシュ関数の安全性が不十分であることを意味するので、デジタル署名の信頼性が揺らいでしまう。

【０００５】

［課題２］この技術の基盤である公開鍵暗号は中間者攻撃を許す。これを防ぐためには、ＰＫＩにおける認証局が発行したデジタル証明書などを利用して送信装置の公開鍵が正規のものかどうかを確認する必要がある（図１５）。しかし、認証局に不正があった場合には、デジタル証明書そのものが信頼できなくなる（非特許文献１）。

【０００６】

［課題３］近年研究が盛んな量子コンピュータが実現すると、この技術の基盤である公開鍵暗号そのものが解読される可能性が飛躍的に高まる（非特許文献４）。この場合、ＰＫＩそのものの信頼性が揺らいでしまう。

【０００７】

これら３つの課題に関し、課題３を解決することを目的の一つとした技術として、再配置暗号方式という暗号化技術が開示されている（特許文献１及び非特許文献５）。図１７は、再配置暗号の特徴をストリーム暗号により示したものである。本図に示すように、再配置暗号化の方法は、

（Ａ）秘密の擬似乱数発生器G_0で第一の擬似乱数列ｒと第二の擬似乱数列Ｒとを生成するステップと、

（Ｂ）第一の擬似乱数列ｒを初期値として公開可能な擬似乱数発生器G_1で擬似乱数列r_xを生成するステップと、

（Ｃ）平文ｘと擬似乱数列r_xとの排他的論理和をとり第一のデータc_xを生成するステップと、

（Ｄ）第一の擬似乱数列をヘッダｒとして第一のデータc_xに付加したデータ（ｒ，c_x）をＳ－ＢＯＸ（Substitution Box）を含む非線形関数系を用いて一体化して第二のデータ$d_x = F(r, c_x)$を生成するステップと、

（Ｅ）第二のデータd_xをｎ個のブロックb_i（ｉ＝０，１，・・・，ｎ－１）に分割するステップと、

（Ｆ）第二の擬似乱数列Ｒに基づき生成された再配置表Ｋを秘密鍵として用いてｎ個のブロックb_i（ｉ＝０，１，・・・，ｎ－１）を重複することなく再配置させた第三のデータを暗号文fc_xとして生成するステップと、

を備えている。

【０００８】

ここで、再配置表Ｋとは、０，１，・・・，ｎ－１までのｎ個の整数の重複のない十分にランダムな並び替えを表す表（再配置表）のことである。この再配置表Ｋは、暗号化処理に先立ち、秘密の擬似乱数生成器Ｇ０が生成する第二の擬似乱数Ｒに基づき作成されるものであり、ｎ！通りの可能性の中から一つが秘密鍵として選ばれる。

【先行技術文献】

【特許文献】

【０００９】

【特許文献１】国際公開第２００８－１１４８２９号

【非特許文献】

【００１０】

【非特許文献１】Introduction to Modern Cryptography, J. Katz and Y. Lindel, Chapman & Hall/CRC (2008).

【非特許文献２】“Finding collisions in the SHA-1”, X. Wang, Y. L. Yin, and H. Yu, In Advances in Cryptology - Crypto 2005, Lecture Notes in Computer Science, Springer, Vol. 3621, pp 17-36(2005).

【非特許文献３】“How to break MD5 and other hash functions”, X. Wang and H. Yu, In Advances in Cryptology - Eurocrypt 2005, Lecture Notes in Computer Science, Springer, Vol. 3494, pp 19-35 (2005).

【非特許文献４】“Algorithms for Quantum Computation: Discrete Logarithms and Factorings”, Shor, P., Proceedings 35th Annual Symposium on Foundations of Computer Science, pp124-134 (1994).
【非特許文献５】“NP completeness of relocation cipher”, Suzuki, S., Far East Journal of Applied Mathematics, Vol. 33, Issue 2, pp219-236 (2008).
【発明の概要】
【発明が解決しようとする課題】
【００１１】
再配置暗号方式は、公開鍵暗号方式と比較して、暗号化の計算量が少なく、高速であり、さらに暗号化されたデータの解読問題はＮＰ完全であることが期待されるが、平文を攪拌する演算回数が少ないので、暗号文の改竄の有無を検出する能力は高くない。これに関し、特許文献１に開示された再配置暗号方式では、鍵付きハッシュ関数でメッセージ認証符号ＭＡＣ（Message Authentication Codes）を実現し、それを用いて電子文章の原本性を保証する応用例が開示されている。しかしながら、この応用例に開示された技術においても、ハッシュ関数が利用されているので、やはり課題１が残されることになる。この課題を克服するためにハッシュ表を大きくすると、その分ハッシュ値の計算速度が遅くなってしまう。
【課題を解決するための手段】
【００１２】
本発明は、このような実情を鑑みて為されたものであり、上記の課題１～３を解決することができる原本性保証装置、原本性保証プログラム、及びこのプログラムを記録する記録媒体を提供することを目的とする。
【００１３】
上記の目的を達成するため、請求項１に記載の発明は、電子文章を入力する入力部と、第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成部と、前記第一の擬似乱数列を初期値として第三の擬似乱数列を生成する公開可能な擬似乱数生成部と、前記電子文章に検査情報として前記第一の擬似乱数列を付加した第一の電子データを生成する検査情報付加部と、前記第一の電子データと前記第三の擬似乱数列とを排他的論理和した第二の電子データに前記第一の擬似乱数列をヘッダとして付加した第三の電子データを対称化一体化関数系を用いて一体化して第四の電子データを生成する一体化処理部と、前記第四の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第五の電子データを暗号文として生成する再配置処理部と、を有する暗号文生成部と、暗号文を受信する受信部と、前記暗号文を所定の分割数に分割し、秘密鍵としての前記再配置表を用いて分割したデータを元の配置に戻して第六の電子データを生成する逆再配置処理部と、前記対称化一体化関数系の逆関数系を用いて前記第六の電子データを逆一体化して第七の電子データを生成し、前記第七の電子データを先頭から所定のデータ長を有する第八の電子データと残りの第九の電子データとに分離する逆一体化処理部と、前記第八の電子データを初期値として前記公開可能な擬似乱数生成部にて生成された擬似乱数列と前記第九の電子データとを排他的論理和した第十の電子データを後ろから前記所定のデータ長を有する第十一の電子データと残りの第十二の電子データとに分離し、前記第十一の電子データと前記逆一体化処理部から出力された前記第八の電子データとを比較して、両者の値が一致した場合に前記第十二の電子データの原本性を保証する検査情報検証部と、を有する復号検証部と、を備え、前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証装置である。
【００１４】
請求項２に記載の発明は、電子文章を入力する入力部と、第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成部と、前記第一の擬似乱数列を鍵として前記電子文章をブロック暗号化して第一の電子データを生成するブロック暗号文生成部と、前記第一の電子デ

ータに前記第一の擬似乱数列をヘッダとして付与した第二の電子データを対称化一体化関数系を用いて一体化して第三の電子データを生成する一体化処理部と、前記第三の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第四の電子データを暗号文として生成する再配置処理部と、を有する暗号文生成部と、暗号文を受信する受信部と、前記暗号文を所定の分割数に分割し、秘密鍵としての所定の再配置表を用いて分割されたデータを元の配置して第五の電子データを生成する逆再配置処理部と、前記対称化一体化関数系の逆関数系を用いて前記第五の電子データを逆一体化して第六の電子データを生成し、前記第六の電子データを先頭から所定のデータ長を有する第七の電子データと残りの第八の電子データと分離する逆一体化処理部と、前記秘密の擬似乱数生成部にて生成された第一の擬似乱数列を鍵として用いることにより前記第八の電子データをブロック復号化した第九の電子データを後ろから前記所定のデータ長を有する第十の電子データと残りの第十一の電子データとに分離し、前記第十の電子データと前記逆一体化処理部から出力された前記第七の電子データとを比較して、両者の値が一致した場合に前記第十一の電子データの原本性を保証する検査情報検証部と、を有する復号検証部と、を備え、前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証装置である。

【００１５】

請求項３に記載の発明は、コンピュータを、電子文章を入力する入力手段と、第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成手段と、前記第一の擬似乱数列を初期値として第三の擬似乱数列を生成する公開可能な擬似乱数生成手段と、前記電子文章に検査情報として前記第一の擬似乱数列を付加した第一の電子データを生成する検査情報付加手段と、前記第一の電子データと前記第三の擬似乱数列とを排他的論理和した第二の電子データに前記第一の擬似乱数列をヘッダとして付加した第三の電子データを対称化一体化関数系を用いて一体化して第四の電子データを生成する一体化処理手段と、前記第四の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第五の電子データを暗号文として生成する再配置処理手段と、を有する暗号文生成手段と、暗号文を受信する受信手段と、前記暗号文を所定の分割数に分割し、秘密鍵としての前記再配置表を用いて分割したデータを元の配置に戻して第六の電子データを生成する逆再配置処理手段と、前記対称化一体化関数系の逆関数系を用いて前記第六の電子データを逆一体化して第七の電子データを生成し、前記第七の電子データを先頭から所定のデータ長を有する第八の電子データと残りの第九の電子データとに分離する逆一体化処理手段と、前記第八の電子データを初期値として前記公開可能な擬似乱数生成手段にて生成された擬似乱数列と前記第九の電子データとを排他的論理和した第十の電子データを後ろから前記所定のデータ長を有する第十一の電子データと残りの第十二の電子データとに分離し、前記第十一の電子データと前記逆一体化処理手段から出力された前記第八の電子データとを比較して、両者の値が一致した場合に前記第十二の電子データの原本性を保証する検査情報検証手段と、を有する復号検証手段と、して機能させ、前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証プログラムである。

【００１６】

請求項４に記載の発明は、コンピュータを、電子文章を入力する入力手段と、第一及び第二の擬似乱数列を生成する秘密の擬似乱数生成手段と、前記第一の擬似乱数列を鍵として前記電子文章をブロック暗号化して第一の電子データを生成するブロック暗号文生成手段と、前記第一の電子データに前記第一の擬似乱数列をヘッダとして付与した第二の電子データを対称化一体化関数系を用いて一体化して第三の電子データを生成する一体化処理手段と、前記第三の電子データを複数のブロックデータに分割し、前記第二の擬似乱数列

に基づき生成された再配置表を秘密鍵として用いることにより前記複数のブロックデータを重複することなく再配置した第四の電子データを暗号文として生成する再配置処理手段と、を有する暗号文生成手段と、暗号文を受信する受信手段と、前記暗号文を所定の分割数に分割し、秘密鍵としての所定の再配置表を用いて分割されたデータを元の配置して第五の電子データを生成する逆再配置処理手段と、前記対称化一体化関数系の逆関数系を用いて前記第五の電子データを逆一体化して第六の電子データを生成し、前記第六の電子データを先頭から所定のデータ長を有する第七の電子データと残りの第八の電子データと分離する逆一体化処理手段と、前記秘密の擬似乱数生成手段にて生成された第一の擬似乱数列を鍵として用いることにより前記第八の電子データをブロック復号化した第九の電子データを後ろから前記所定のデータ長を有する第十の電子データと残りの第十一の電子データとに分離し、前記第十の電子データと前記逆一体化処理手段から出力された前記第七の電子データとを比較して、両者の値が一致した場合に前記第十一の電子データの原本性を保証する検査情報検証手段と、を有する復号検証手段と、して機能させ、前記対称化一体化関数系は、Ｓ－ＢＯＸを含む第一の非線形関数系からの出力を、前記第一の非線形関数系の逆変換としての形を有する第二の非線形関数系の入力とする関数系であることを特徴とする原本性保証プログラムである。

【００１７】

請求項５に記載の発明は、請求項３又は４項に記載の原本性保証プログラムが記録されたコンピュータが読み取り可能な記録媒体である。

【発明の効果】

【００１８】

本発明によれば、電子文章の高速な暗号化を可能とすると共に、その原本性を、認証局を介すことなく、当事者間で完結して保証できる。従って、本発明を、ネットワーク上にデータを置くクラウドコンピューティングにて運用すれば、その信頼性を高めることができる。また、本発明においては、暗号文に証拠能力が保証されたデジタル署名を付与できるので、法律的な証拠として活用できる。また、本発明においては、暗号文の解読問題がＮＰ完全であることが期待されることから、量子コンピュータが実用化されたとしても原本性の保証が可能となる。

【図面の簡単な説明】

【００１９】

【図１】本発明の一実施の形態に係る暗号方式の特徴をストリーム暗号により示した概念図である。

【図２】本発明の一実施の形態に係る送信装置としての原本性保証装置の暗号化処理に関わる処理部の概略的な構成を示したブロック図である。

【図３】図２に示した原本性保証装置における暗号化処理の手順を示したフローチャートである。

【図４】暗号化処理時における一体化処理と復号化処理時における逆一体化処理との関係を示す図である。

【図５】図４に示した暗号化処理と復号化処理との非対称性を示すシミュレーション結果を示す図である。

【図６】図３に示した暗号化処理における一体化処理の具体的な手順を示したフローチャートである。

【図７】図３に示した暗号化処理における再配置処理の具体的な手順を示したフローチャートである。

【図８】本発明の一実施の形態に係る受信装置としての原本性保証装置の復号化処理に関わる処理部の概略的な構成を示したブロック図である。

【図９】図８に示した原本性保証装置における復号化処理及び検証処理（以後、復号検証処理と称する）の手順を示したフローチャートである。

【図１０】図８に示した原本性保証装置における逆再配置処理の具体的な手順を示したフローチャートである。

【図１１】図８に示した原本性保証装置における逆一体化処理の具体的な手順を示したフローチャートである。
【図１２】図２に示した原本性保証装置の一変更例の暗号化処理に関わる処理部の概略的な構成を示したブロック図である。
【図１３】図１２に示した原本性保証装置における暗号化処理の手順を示した概念図である。
【図１４】図１２に示した原本性保証装置の復号化処理に関わる処理部の概略的な構成を示したブロック図である。
【図１５】図２に示した原本性保証装置を用いた原本性保証方法を示す図である。
【図１６】ＰＫＩにおける原本性保証方法を示す図である。
【図１７】従来の再配置暗号方式の特徴をストリーム暗号により示す概念図である。
【発明を実施するための形態】
【００２０】
図１は、本発明に係る暗号方式の特徴をストリーム暗号により示した概念図である。図１に示すように、この暗号方式は、再配置暗号を用いたものであり、
（１） 秘密の擬似乱数発生器G_0で第一の擬似乱数列ｒと第二の擬似乱数列Ｒとを生成するステップと、
（２） 第一の擬似乱数列ｒを初期値として公開可能な擬似乱数発生器G_1で第三の擬似乱数列r_yを生成するステップと、
（３） 電子文章ｘに検査情報として第一の擬似乱数列ｒを付加し第一の電子データｙ＝（ｘ，ｒ）を生成するステップと、
（４） 第一の電子データｙと第三の擬似乱数列r_yとの排他的論理和をとり第二の電子データc_yを生成するステップと、
（５） 第一の擬似乱数列ｒをヘッダとして第二の電子データc_yに付加した第三の電子データ（ｒ，c_y）をＳ－ＢＯＸを含む対称化一体化関数系を用いて一体化して第四の電子データd_yを生成するステップと、
（６） 第四の電子データd_yをｎ個のブロックb_i（ｉ＝０，１，・・・，ｎ－１）に分割するステップと、
（７） 第二の擬似乱数列Ｒに基づき生成された再配置表Ｋを秘密鍵として用いてｎ個のブロックb_i（ｉ＝０，１，・・・，ｎ－１）を重複することなく再配置させた第五の電子データを暗号文fc_yとして生成するステップと、
を備える。
【００２１】
さらに、この暗号方式では、必要に応じて、ステップ（５）～（７）における一体化から再配置までの処理を数回繰り返すステップを含めてもよい。
【００２２】
なお、再配置表Ｋは、０，１，・・・，ｎ－１までのｎ個の整数の重複のない十分にランダムな並び替えを表す表（再配置表）のことであり、以降では、Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）と表すことにする。再配置表Ｋは、暗号化処理を行うたびに、秘密の擬似乱数生成器G_0が生成する第二の擬似乱数Ｒに基づき作成されるものであり、ｎ！通りの可能性の中から一つが秘密鍵として選ばれる。
【００２３】
なお、ステップ（４）では、第一の電子データｙと第三の疑似乱数r_yとの排他的論理和をとったが、正確には、図１に示すように、第三の疑似乱数r_yとの排他的論理和をとる対象は、第一の電子データｙと、電子文章ｘのデータ長などの情報を含んだヘッダ情報ｕと、必要に応じてパディングｐとを合わせたものである。以降においては、ヘッダ情報ｕと第一の電子データｙと、必要に応じてパディングｐとを合わせたデータを改めて第一の電子データｙと見なして説明する。
【００２４】
以下に、本発明の実施の形態を、図面を用いて詳細に説明する。

【００２５】
［送信装置としての原本性保証装置］
図２は、本発明の一実施の形態に係る送信装置としての原本性保証装置の暗号化処理に関わる処理部の概略的な構成を示したブロック図である。原本性保証装置１０は、入力部２０と、第一擬似乱数生成部３０と、第二擬似乱数生成部４０と、一体化処理部５０と、再配置処理部６０と、出力部７０と、記憶部８０と、制御部９０と、検査情報付加部１００とを備える。このうち、記憶部８０と制御部９０とを除く部分を暗号文生成部１０Ａと称することにする。
【００２６】
入力部２０は、送信者が電子文章ｘを入力するための入力インターフェースである。
【００２７】
第一擬似乱数生成部３０は、予測困難な擬似乱数列（第一及び第二の擬似乱数列ｒ，Ｒ）を生成する擬似乱数生成器であり、暗号装置１０のシステムクロックや入力部２０からの入力タイミングなどを利用することができるが、より乱数に近いものとして熱雑音などを用いることもできる。第一擬似乱数生成部３０として熱雑音による擬似乱数生成器を用いた場合、その他の場合と比べてコストを低減できる。第一擬似乱数生成部３０としては、毎回異なる擬似乱数列を生成することが重要であるので、使い捨て擬似乱数生成器を用いてもよい。第一擬似乱数生成部３０に使用される擬似乱数生成器は送信者と受信者の間で秘密にされる。第一擬似乱数生成部が生成する擬似乱数列については、送信者、受信者を含め誰も一切知っている必要がない。
【００２８】
第二擬似乱数生成部４０は、統計的に偏りのない擬似乱数列（第三の擬似乱数列r_y）を生成する擬似乱数生成器であり、メルセンヌ・ツイスターを用いることができる。メルセンヌ・ツイスターは、統計学的に優れた擬似乱数列を生成できるが、暗号学的には安全ではない。しかし、本発明においては、第一擬似乱数生成部３０で生成した第一の擬似乱数列ｒを後述する一体化処理部５０による非線形変換と後述する再配置処理部６０による再配置により秘匿にすることができるので、メルセンヌ・ツイスターの使用が可能である。第二擬似乱数生成器４０に使用される擬似乱数生成器は送信者と受信者以外に対して公開してもよい。
【００２９】
もちろん、第一擬似乱数生成部３０としてメルセンヌ・ツイスターを使用してもよい。
【００３０】
検査情報付加部１００は、入力部２０から入力された電子文章ｘに検査情報を付加して第一の電子データｙを生成する処理部である。本実施の形態においては、電子文章ｘの原本性を保証するために、検査情報付加部１００は、検査情報として第一擬似乱数生成器３０にて生成された第一の擬似乱数列ｒを電子文章ｘに付加する。さらに、本実施の形態においては、検査情報付加部１００は、検査情報としての第一の擬似乱数列ｒを電子文章ｘのフッタとして付加して第一の電子データｙ＝（ｘ，ｒ）を生成する。
【００３１】
一体化処理部５０は、第二擬似乱数生成４０にて生成された第三の擬似乱数列r_yと検査表情付加部１００にて生成された第一の電子データｙとの排他的論理和をとることにより生成した第二の電子データc_yに第一の擬似乱数列ｒをヘッダとして付加した第三の電子データ（ｒ，c_y）をＳ－ＢＯＸを含む非線形関数系を用いて、第三の電子データ（ｒ，c_y）に後述する一体化処理を行うことにより第四の電子データd_y＝Ｓ（ｒ，c_y）を生成する。
【００３２】
再配置処理部６０は、第四の電子データd_yをｎ個のブロックに分割し、第二の擬似乱数列Ｒからｎ個のブロックを再配置する再配置表を用いて第四の電子データd_yに後述する再配置処理を行うことにより第五の電子データを暗号文fc_yとして生成する。
【００３３】

出力部７０は、最終的に生成された暗号文ｆｃ$_y$を受信者へ出力するための出力インターフェースである。

【００３４】

記憶部８０は、入力部２０～出力部７０、及び検査情報付加部１００から成る暗号生成部１０Ａが生成した各種のデータの格納を行うサブメモリと、後述する暗号化処理の各ステップを実行するためのコンピュータに読み取り可能な暗号プログラムを格納するメインメモリとから構成される。記憶手段８０は、ＲＡＭ（Random Access Memory）やＲＯＭ（Read Only Memory）などから構成される。さらに、記憶部８０のサブメモリとメインメモリとを別体として構成し、メインメモリ部分を磁気ハードディスク、フロッピー（登録商標）ディスク、CD-ROMなどの光ディスク、磁気テープ、メモリチップ等に記憶させてもよい。

【００３５】

制御部９０は、記憶部８０から読み出した暗号プログラムに従って、入力部２０～記憶部８０を制御するＣＰＵ(Central Processing Unit)を備える。

【００３６】

本実施の形態では、原本性保証装置１０を、暗号文生成部１０Ａ及び制御部９０と、記憶部８０とを一体化した構成としたが、記憶部８０を独立した記憶装置として暗号文生成部１０Ａ及び制御部９０とから切り離した構成としてもよい。いずれの構成においても、原本性保証装置１０はコンピュータによって実現されるものであり、入力部２０～出力部７０、及び検査情報付加部１００は、制御部９０により記憶部８０から読み出された暗号化プログラムに従って制御される。

【００３７】

ここで、コンピュータとは、構造化された入力を所定の規則に従って処理し、処理した結果を構造化して出力する装置のことを指し、例えば、汎用コンピュータ、スーパーコンピュータ、メインフレーム、ワークステーション、マイクロコンピュータ、サーバ等が含まれる。また、通信ネットワーク（例えば、イントラネット、ローカルエリアネットワーク（LAN）、ワイドエリアネットワーク（WAN）、及びこれらの組み合わせから成る通信ネットワーク）を介して接続された２つ以上のコンピュータから成る構成（例えば、分散コンピュータシステム）であってもよい。

【００３８】

また、ここでのコンピュータには、携帯電話やモバイル端末、家電製品や自動車などの制御チップ、コントローラ、ＩＣカードに組み込まれた演算装置なども含まれる。

【００３９】

［暗号化処理］

以上を前提として、図２に示した原本性保証装置１０によって行われる暗号化処理について詳細に説明する。図３は、図２に示した原本性保証装置１０によって行われる暗号化処理の手順を示したフローチャートである。

【００４０】

送信者により入力部２０から電子文章（平文）ｘ（長さ：ｇワード）が入力されると、制御部９０は、これを記憶部８０に記憶させ、記憶部８０に格納された暗号化プログラムに従い、第一擬似乱数生成部３０～生成部７０、及び検査情報付加部１００に対して以下に示す処理を行うように促す。

【００４１】

ステップＳ１０において、制御部９０は、第一擬似乱数生成部３０に対して、第一の擬似乱数列ｒ（長さ：ａワード）を生成させ、これを記憶部８０に記憶させる。なお、第一の擬似乱数列ｒの長さａは任意に設定することができる。

【００４２】

＜擬似乱数ヘッダｒの生成アルゴリズムの実施例＞

ステップＳ１０における第一擬似乱数生成部３０による第一の擬似乱数列ｒの生成アルゴリズムは、以下のように記述される。

【００４３】
```
r: array[0..a-1] of the word;
Randomize; //initialize G0 by the clock
for i:=0 to a-1 do
  r[i]:=G0;
```
次に、ステップＳ２０において、制御部９０は、記憶部８０から予め格納された分割数ｎを読み出し、第一擬似乱数生成部３０に対して、０，１，・・・，ｎ－１までのｎ個の整数から成る第二の疑似乱数列Ｒを生成させ、これを再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）として記憶部８０に記憶させる。

【００４４】
なお、再配置表Ｋは、暗号化の度に生成する必要はない。図３のフローチャートでは、第一の擬似乱数列ｒの後に再配置表Ｋを生成するようにしているが、基本的には、第一の擬似乱数列ｒの生成と再配置表Ｋの作成とは独立した処理として規定される。

【００４５】
＜再配置表Ｋの生成アルゴリズムの実施例＞
ステップＳ２０における第一擬似乱数生成部３０による再配置表Ｋの生成アルゴリズムは、以下のように記述される。

【００４６】
```
k: array[0..n-1] of the word;
Randomize; //initialize G0 by the clock
for i:=0 to N-1 do
  k[i]:=i;
for i:=0 to rn-1 do
begin
  for j:=0 to n-1 do
  begin
  s:=G0 mod n;
  x:=k[j];
  k[j]:=k[s];
  k[s]:=x;
  end
end;
```
ここで、１ワードは８ビット、１６ビット、又は３２ビットの符号なし整数を表す。また、ここでは、第一擬似乱数生成部３０は、毎回１ワードの擬似乱数を出力するものとしている。このアルゴリズムは、鍵の長さをｎとしたときＯ（ｎ）の計算量でランダムな置換を生成する高速なアルゴリズムである。

【００４７】
また、第一擬似乱数生成部３０において、ある周期の擬似乱数からより長い周期の擬似乱数を生成することも考えられる。例えば、第一擬似乱数生成部３０において、８バイトの擬似乱数列から２５６バイトの第二の擬似乱数列（再配置表Ｋ）を生成するアルゴリズムは、以下のように記述される。

【００４８】
＜実施例＞
```
procedure set 8byte;
var
  d: array [0..7] of byte;
  k: array [0..255] of byte;
  i, j, z: integer
  x: byte;
  function g: byte;
```

```
    var
      a, b: integer;
    begin
      a:=(j+7) and 7;
      b:=j and 7;
      d[b]:=d[b]+d[a];
      g:=d[b];
    end;
  begin
    for i :=0  to 255 do
    begin
      k[i]:=i;
    end;
    read_d;
    for i :=0  to 1 do //２ラウンド攪拌
    begin
      for j :=0  to 255 do
      begin
      z:=g and 255;
      x:=k[j];
      k[j]:=k[z];
      k[z]:=x;
      end;
    end;
  end;
```

一例として、このアルゴリズムにて、８バイトの擬似乱数“１０４，１２７，１５６，１６４，９，２４６，９９，２１０”から２５６バイトの擬似乱数を生成し、２ラウンド撹拌して再配置表Ｋを生成した結果は、次のようになる。

【００４９】

K=(159,251,3,153,44,233,98,40,193,66,169,200,184,253,206,212,17,45,246,30,250,199,177,34,235,197,95,243,180,131,176,61,52,237,157,228,78,213,106,80,166,186,22,226,74,149,14,218,170,94,100,59,140,31,10,143,249,130,152,4,91,57,49,156,77,241,238,214,167,6,71,247,232,112,221,148,73,58,201,207,2,146,55,90,102,162,33,103,109,39,85,13,105,63,189,11,178,215,220,255,181,0,23,37,114,171,202,96,72,164,188,62,223,7,51,27,144,147,16,21,203,163,101,175,192,46,48,18,108,126,229,43,230,160,118,117,15,234,154,155,111,231,219,9,227,132,53,110,121,136,107,65,64,47,239,216,128,198,76,183,68,29,141,56,69,125,50,142,138,209,70,99,211,81,150,42,35,185,224,225,222,240,104,5,139,93,179,1,129,83,19,248,115,67,36,242,12,174,123,236,54,151,120,60,24,182,84,38,254,208,86,116,82,244,41,217,165,161,122,75,20,190,26,173,195,88,187,172,210,87,145,32,97,124,119,137,196,204,205,79,135,134,191,127,133,28,89,8,25,92,252,245,158,113,168,194)。

【００５０】

この関数ｇの周期は比較的長いが、統計的な性質はあまり良くないかも知れない。しかし、この関数ｇは、計算速度は高速であり、２つずつメモリ内容を置換する処理が非線形であるため、２ラウンド以上撹拌することでこの欠点を補うことができる。

【００５１】

また、８バイトではなく一般のｍバイトの擬似乱数列からより長い擬似乱数列を生成する関数gは、次のように書ける。

【００５２】

＜関数ｇの別の実施例＞

```
function g: byte;
var
  a,b: integer;
begin
  a:=(j+m-1) mod m;
  b:=j mod m;
  d[b]:=d[b]+d[a];
  g:=d[b];
end;
```

この関数ｇは、多くの場合、2^{7}（$2^{m-1}-1$）程度の周期になる。つまり、８バイトの場合は１６２５６程度の周期になり、１６バイトの場合は４１９４１７６程度の周期になる。この周期は、ｍｏｄ ２の多項式の因数分解のされ方によって多少変化する。

【００５３】

次に、ステップＳ３０において、制御部９０は、送信者により入力部２０から入力された電子文章ｘを読み込み、これを電子文章ｘのデータ長ｇなどの情報を含んだヘッダ情報ｕ（長さ：ｑワード）と共に記憶部８０に記憶させる。

【００５４】

次に、ステップＳ４０において、制御部９０は、第一の擬似乱数列ｒを記憶部８０から読み出し、検査情報付加部１００に対して、入力部２０から入力された電子文章ｘにヘッダ情報ｕを付加したものに、検査情報としての第一の擬似乱数列ｒをフッタとして付加して第一の電子データｙ＝（ｕ，ｘ，ｒ）を生成させ、これを記憶部８０に記憶させる。

【００５５】

次に、ステップＳ５０において、制御部９０は、第一の擬似乱数列ｒを記憶部８０から読み出し、第二擬似乱数生成部４０に対して、第一の擬似乱数列ｒを初期値として、第一の電子データｙと同じ長さ（ｎｍ－ａ－１ワード）の第三の擬似乱数列$r_y=(r_0, r_1, \cdots, r_{nm-a-1})$を生成させ、これを記憶部８０に記憶させる。

【００５６】

なお、このステップにおいて、入力部２０から入力される電子文章ｘと検査情報としての第一の擬似乱数列ｒとの和の長さｇ＋ａが分割数ｎの倍数ではない場合には、制御部９０は、$v \equiv -2a-q-g \pmod{n}$となるような最小の非負整数ｖを算出し、パディングｐとしてｖワードの長さの擬似乱数列ｚを第一擬似乱数生成部３０に生成させ、第一の電子データｙの最後に付加する処理を行う。そして、制御部９０は、ヘッダ情報ｕ、平文ｘ、検査情報としての第一の擬似乱数ｒ、及びパディングｐを合わせたデータを改めて第一の電子データ$y=(u, x, z)=(x_0, x_1, \cdots, x_{nm-a-1})$として記憶部８０に記憶させる。また、この場合、再配置処理において分割される各ブロックの長さを表す整数“ｍ”は、$m=(2a+q+g+v)/n$として算出される。

【００５７】

次に、ステップＳ６０において、制御部９０は、記憶部８０から第一の電子データ$y=(x_0, x_1, \cdots, x_{nm-a-1})$と第三の擬似乱数列$r_y=(r_0, r_1, \cdots, r_{nm-a-1})$とを読み出し、両者の排他的論理和をとる（$c_i=x_i \ \mathrm{XOR}\ r_i$（$i=0, 1, \cdots, nm-a-1$））ことにより第二の電子データ$c_y=(c_0, c_1, \cdots, c_{nm-a-1})$を生成し、これを記憶部８０に記憶させる。

【００５８】

なお、本実施の形態では、このステップにおいて第三の擬似乱数列r_yと第一の電子データｙとの間で一度に排他的論理和する構成としたが、その代わりに、制御部９０は、ステップＳ５０において第二の擬似乱数生成部４０に対して１ワードづつ擬似乱数を生成させ、そのつど、ステップＳ６０において第一の電子データｙの１ワードと逐次的に排他的論理和する構成としてもよい。

【００５９】

次に、ステップＳ７０において、制御部９０は、記憶部８０から第一の擬似乱数列ｒと

第二の電子データc_y＝（c_0，c_1，・・・，c_{nm-a-1}）とを読み出し、第一の擬似乱数列ｒを第二の電子データc_yのヘッダとして付加した第三の電子データ（ｒ，c_y）＝（c_0，c_1，・・・，c_{nm-1}）を生成させ、これを改めてc_y＝（ｒ，c_x）＝（c_0，c_1，・・・，c_{nm-1}）として記憶部８０に記憶させる。

【００６０】

次に、ステップＳ８０において、制御部９０は、ステップＳ９０～Ｓ１２０で行われる変換処理と分割処理と再配置処理とを１セットとした処理のラウンド数を表すＣｔを立て（Ｃｔ＝０）、ステップＳ９０へ処理を進める。

【００６１】

ステップＳ９０において、制御部９０は、記憶部８０から第三の電子データc_y＝（c_0，c_1，・・・，c_{nm-1}）を読み出し、一体化処理部５０に対して、２つの非線形関数系から構成された対称化一体化関数系を用いて第三の電子データc_y＝（c_0，c_1，・・・，c_{nm-1}）をバイト単位で変換して一体化して第四の電子データd_y＝Ｓ（ｒ，c_y）＝（d_0，d_1，・・・，d_{nm-1}）を生成させ、これを記憶部８０に記憶させる。ここで、対称化一体化関数系について説明する。

【００６２】

［対称化一体化関数系の定義と特徴］

図４は、暗号化処理時における一体化処理と復号化処理時における逆一体化処理との関係を示す図である。

【００６３】

図４（Ａ）は、暗号化処理時におけるＳ－ＢＯＸを用いた一体化処理の最も単純な例を示したものである。この例では、ステップＳ９１－１～Ｓ９９－１に示すように、一体化処理部５０は、記憶部８０から読み出された第三の電子データc_y＝（c_0，c_1，・・・，c_{nm-1}）に対して、$c_{(dm+i)\bmod nm}$＝$c_{(dm+i)\bmod nm}$＋$c_{i \bmod nm}$（ｉ＝０，１，・・・，ｎｍ－１）として加算した値（左辺の$c_{(dm+i)\bmod nm}$）をＳ－ＢＯＸを用いてバイト単位で非線形変換し一体化する処理を行う。この一体化処理を行う非線形関数系をｆ（ｓ，ｍ，ｄ）と書くことにする。ここで、ｓはＳ－ＢＯＸの関数を表し、ｍは一体化（暗号化）処理におけるラウンド数を表し、ｄはブロック差分（ｄは０＜ｄ＜ｎを満たす整数）を表している。

【００６４】

一方、図４（Ｂ）は、復号化処理時におけるＳ－ＢＯＸを利用した逆一体化処理の最も簡単な例を表したものである。この図において、逆一体化（復号化）処理を行う非線形関数系は、ステップＳ９１－２～Ｓ９９－２に示すように、ｇ（ｉｓ，ｋ，ｄ）と書くことができる。ここで、ｉｓはＳ－ＢＯＸの関数ｓの逆関数を表し、ｋは逆一体化処理におけるラウンド数を表し、ｄはブロック差分（ｄは０＜ｄ＜ｎを満たす整数）を表している。

【００６５】

また、図６（Ａ）は、特許文献１に示されている暗号化処理時における一体化処理の非線形関数系Ｆ（ｓ，ｍ，ｄ）を示したものであり、図１１（Ａ）は特許文献１に示されている復号化処理時における逆一体化処理の非線形関数系Ｆ（ｉｓ，ｍ，ｄ）を示したものである。従って、図４（Ａ）のｆ（ｓ，ｍ，ｄ）と図６（Ａ）のＦ（ｓ，ｍ，ｄ）とが対応し、図４（Ｂ）のｇ（ｉｓ，ｍ，ｄ）と図１１（Ａ）のＦ（ｉｓ，ｍ，ｄ）とが対応する。

【００６６】

ここで、ｆ（ｓ，ｍ，ｄ）とｇ（ｉｓ，ｍ，ｄ）はそれぞれ暗号化処理時の一体化処理を行う非線形関数系と復号化処理時の逆一体化処理を行う非線形関数系なので、関数系ｆを用いた一体化処理による攪拌効果とｇを用いた一体化処理による攪拌効果とは直感的には対称であることが期待される。しかし、以外なことにこれらの攪拌効果は全く非対称である。その実例を図５に表す。図５（Ａ）は、シミュレーションで用いたＳ－ＢＯＸの具体例を表している。この例では、３０１個のデータ配列ｃ［０］，ｃ［１］，・・・，ｃ［３００］の最初の１０個にだけ擬似乱数を格納し、残りは全て０を格納している。この

データ配列ｃ［０］，ｃ［１］，・・・，ｃ［３００］に対してｆ（ｓ，５，ｄ）を作用させた値をプロットすると図５（Ｂ）に示すようになる。ところが、同じデータ配列に対してｇ（ｓ，５，ｄ）を作用させたものをプロットすると図５（Ｃ）に示すようになってしまう。

【００６７】

このことは、暗号化処理時における一体化処理を行う非線形関数系ｆ（ｓ，ｍ，ｄ）と復号化処理時における逆一体化処理を行う非線形関数系を逆に用いて一体化処理を行う非線形関数系ｇ（ｓ，ｍ，ｄ）とではその攪拌性能が大きく異なることを意味する。これより、図６（Ａ）に示した特許文献１の暗号化処理時における一体化処理の非線形関数系Ｆ（ｓ，ｍ，ｄ）は、攪拌性能には優れているが、改竄検出能力は劣っていることがわかる。同様に、図１１（Ａ）に示した特許文献１の復号化処理時における逆一体化処理を行う非線形関数系を逆に用いて一体化処理を行う非線形関数系Ｇ（ｓ，ｍ，ｄ）は、攪拌性能には劣っているが、改竄検出能力には優れていることがわかる。

【００６８】

そこで、この事実を逆手にとって、本実施の形態においては、図６（Ａ）及び６（Ｂ）に示した２つの一体化処理の非線形関数系Ｆ（ｓ，ｍ，ｄ）とＧ（ｓ，ｋ，ｄ）とを、図６（Ｃ）に示すように配置して構成した関数系を用いて、第三の電子データc_y＝（c_0，c_1，・・・，c_{nm-1}）をバイト単位で変換して一体化して第四の電子データd_y＝Ｓ（ｒ，c_y）＝（d_0，d_1，・・・，d_{nm-1}）を生成させる。ここでｋ＜ｍである。このようにして構成された関数系のことを対称化一体化関数系と称し、これを構成する２つの非線形関数系Ｆ（ｓ，ｍ，ｄ）とＧ（ｓ，ｋ，ｄ）のことをそれぞれ第一の一体化関数系及び第二の一体化関数系と称する。

【００６９】

このようにして構成された対称化一体化関数系を用いることにより、電子文章ｘの原本性保証に重要となる復号化処理時における攪拌効果が十分に得られることになる。ただし、例えば、対称化一体化関数系におけるＳ－ＢＯＸの周期が２となった場合、一体化関数系を対称化したことにより、暗号化処理時に復号化処理が含まれることになるので、暗号化の強度が弱まることが考えられる。しかし、その確率は非常に小さいことが以下の定理により数学的に保証される。

【００７０】

＜定理１＞

２ｍ分割の再配置暗号の鍵が周期２になる確率Ｐは、不等式$P \ll e^{\sqrt{2}}/2^m$を満たす。

【００７１】

定理１は、対称化一体化関数系を用いても暗号化の強度が弱くはならないことを示している。

【００７２】

また、再配置暗号の分割数をｎとし、S_nをΩ＝｛１，２，・・・，ｎ｝上の対称群とし、Ｆ（σ）＝｛ｘ∈Ω｜ｘ＝σ（ｘ）｝（σ∈S_n）とする。ここで、自明度Ｔ：S_n→ＮをＴ（σ）＝＃Ｆ（σ）（σ∈S_n）で定義すると、次の定理が成り立つことが証明できる。

【００７３】

＜定理２＞

十分大きなｎと０≪ｍ≪ｎを満たす任意のｍに対して、不等式１－１／ｍ！＜Ｐ（Ｔ＜ｍ）＜１－１／（ｅ・ｍ！）が成り立つ。

【００７４】

定理２は、再配置暗号の鍵には自明なものがほとんどないことを示していて、再配置暗号がＮＰ完全であることが期待されるが、より平均的な意味においても安全性が高いことを示している。定理１及び２が意味するのは、再配置暗号の鍵は不動点が少なく、周期の短いものも非常に少ないということである。

【００７５】
ここで、ステップＳ９０の処理の説明に戻る。ステップＳ９０において、制御部９０は、記憶部８０から第三の電子データ$c_y=(c_0, c_1, \cdots, c_{nm-1})$を読み出し、一体化処理部５０に対して、図６（Ａ）のステップＳ９１－３～Ｓ９７－３に示すように、$c_{(dm+i) \bmod nm}=c_{(dm+i) \bmod nm}+c_{i \bmod nm}$（ｉ＝０，１，・・・，ｎｍ－１）として加算した値（左辺の$c_{(dm+i) \bmod nm}$）をＳ－ＢＯＸを用いてバイト単位で変換して一体化する処理を第一の一体化関数系Ｆ（ｓ，ｍ，ｄ）を用いて行わせる（図６（Ｃ）のステップＳ９０－１）。その後、制御部９０は、一体化処理部５０に対して、図６（Ｂ）のステップＳ９１－４～Ｓ９１７－４に示すように、第一の一体化関数系Ｆ（ｓ、ｍ、ｄ）にて一体化された電子データをＳ－ＢＯＸを用いてバイト単位で変換して、$c_{(dm+i) \bmod nm}=c_{(dm+i) \bmod nm}-c_{i \bmod nm}$（ｉ＝０，１，・・・，ｎｍ－１）として減算する処理を第二の一体化関数系Ｇ（ｓ，ｋ，ｄ）を用いて行わせる（図６（Ｃ）のステップＳ９０－２）。そして、制御部９０は、このようにして生成された第四の電子データ$d_x=S(r, c_x)=(d_0, d_1, \cdots, d_{nm-1})$を記憶部８０に記憶させる。
【００７６】
また、図６（Ａ）及び６（Ｂ）に示した一体化換処理のフローチャートの中の関数ｗは、ワードとバイト配列の共用体であり、次の形式で記憶部８０に記憶される。
【００７７】
```
Tunion=record
  case integer of
  1: (d:Word);
  2: (h:array[0..3] of byte);
end;
w: Tunion;
```
ここでは、１ワードを４バイトとして扱っている。
【００７８】
なお、この一体化処理における変換処理は、可逆な変換（１対１対応の変換）であればどのような処理を行ってもよい。例えば、図６（Ａ）で用いた加算の変わりに減算を用いてよい。また、Ｓ－ＢＯＸについても、非線形な変換であればどのような関数系を用いてもよい。しかしながら、本発明の主旨からは、再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）から次のようにして生成されるＳ－ＢＯＸを用いることが好ましい。
【００７９】
＜再配置表ＫによるＳ－ＢＯＸの生成アルゴリズムの実施例＞
再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）からは様々な方法でＳ－ＢＯＸを生成することができるが、ここでは簡単な実施例をあげる。
【００８０】
```
（n<=256のとき）
e:=256-n;
for i:=0 to e-1 do
  s[i]:=n+i;
for i:=e to 255 do
  s[i]:=k[i-e];
（n>256のとき）
```
n-256≧e≧0となる整数eを一つ決める。
【００８１】
```
ct:=0;
for i:=0 to n-1 do
begin
  if (k[i]-e>=0) and (k[i]-e<256) then do
```

```
    begin
      s[ct]:=k[i]-e;
     ct:=ct+1
    end;
  end;
```

ここで、ｎ＝２５６のときはｓ＝Ｋになり、変換処理における非線形な変換を施すために、再配置表Ｋそのものが利用できる。つまり、このときには、Ｓ－ＢＯＸは、０，１，・・・，２５５をランダムに並び替えたｓ＝（ｋ［０］，ｋ［１］，・・・，ｋ［２５５］）となる。

【００８２】

次に、ステップＳ１００において、制御部９０は、記憶部８０から変換された第四のデータ$d_y=(d_0, d_1, \cdots, d_{nm-1})$を読み出し、再配置処理部６０に対して、これを長さｍワードのｎ個のブロックデータ$b_i=(d_{mi}, d_{mi+1}, \cdots, d_{mi+m-1})$（ｉ＝０，１，・・・，ｎ－１）に分割させ、分割されたデータを新たに第四の電子データ$d_x=(b_0, b_1, \cdots, b_{n-1})$として記憶部８０に記憶させる。

【００８３】

次に、ステップＳ１１０において、制御部９０は、記憶部８０から分割された第四のデータ$d_y=(b_0, b_1, \cdots, b_{n-1})$と再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）とを読み出し、再配置処理部６０に対して、図７に示すように、分割されたデータ$d_y=(b_0, b_1, \cdots, b_{n-1})$を、再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）に基づき、$d_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$のように再配置させ、これを第五の電子データとして記憶部８０に記憶させる。なお、図７において、命令Ｍｏｖｅ（ｘ［ｉ］，ｙ［ｊ］，ｚ）は、ｘ［ｉ］のアドレスからｙ［ｊ］のアドレスへｚバイトの記憶内容をコピーする処理を表す。

【００８４】

次に、ステップＳ１２０において、制御部９０は、フラグＣｔの値をインクリメントし（Ｃｔ＝１）、ステップＳ１３０において、インクリメントされた値が所定のラウンド回数ｈを越えたか否かを判定する。

【００８５】

そして、ステップＳ１３０において、制御部９０によりインクリメントされた値が所定のラウンド回数ｈを越えたと判定された場合は、ステップＳ１４０に処理を進める。この場合、制御部９０は、記憶部８０から第五の電子データ$d_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$を読み出し、最終的な暗号文$fc_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$として出力部７０に出力する。そして、出力部７０は、後述する受信装置に対して暗号文$fc_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$を送信してすべての処理を終了する。

【００８６】

一方、ステップＳ１３０において、制御部９０によりインクリメントされた値が所定のラウンド回数ｈを越えていないと判定された場合は、ステップＳ９０に戻って、インクリメントされた値が所定のラウンド回数ｈを越えるまで、ステップＳ９０～Ｓ１２０までの処理を繰り返した後、ステップＳ１４０に進む。

【００８７】

［送信装置としての原本性保証装置１０の効果］

原本性保証装置１０は、実質的に二つの鍵を用いて電子文章ｘを暗号化処理している。第一の鍵は、第一擬似乱数生成部（秘密の擬似乱数生成器）３０により生成され、平文ｘにヘッダとして付加される第一の擬似乱数列ｒである。第二の鍵は、同じく第一擬似乱数生成部３０により生成され、再配置処理の際に使用される第二の疑似乱数列Ｒから生成される再配置表Ｋである。これらの鍵を用いることによって、原本性保証装置１０は、以下のような効果をもたらす。

【００８８】
（Ｅ１）第一擬似乱数生成部３０は、電子文章ｘを暗号化するたびに異なる第一の擬似乱数列ｒを生成することができる。異なる第一の擬似乱数列ｒを初期値として第二擬似乱数生成部（公開可能な擬似乱数生成器）４０で生成される第三の擬似乱数列$ｒ_{y}$の間には相関がほとんどないので、本暗号を既知平文攻撃することは極めて難しい。
【００８９】
（Ｅ２）第一擬似乱数生成部３０が生成する第二の擬似乱数列Ｒに同じ数の重複する配列がない場合、第一擬似乱数生成部３０はｎ！通りの可能性の中から秘密鍵として一つの再配列表Ｋを生成することができる。例えば、ｎ＞４０の場合、その組み合わせは$２^{１５９}$よりも大きくなり、さらに、最も実用的なｎ＝２５６の場合、その組み合わせの数は$２^{１６８３}$を越えることになるので、鍵の全探索は事実上不可能になる。
【００９０】
（Ｅ３）分割数ｎを大きくすることに計算上のコストはかからない。また、本暗号は、擬似乱数の生成、整数の加算、メモリ内容のコピーといった高速処理が可能な演算のみから構成されているので、暗号化の実現速度は、現在標準の共通鍵方式であるＡＥＳと比較して極めて高速である。また、全体の演算回数も十分の一程度以下になる。
【００９１】
（Ｅ４）（Ｅ２）で述べたように、本暗号の安全性は、データを分割し再配置したものを元に戻すことの計算量的困難さに基づいている。このことを考慮すると、本発明は長期間同じ鍵を使用しても高いセキュリティレベルが維持できる。
【００９２】
（Ｅ５）（Ｅ４）の利点は、本暗号が長期間のデータ保存に適していることを意味する。このことから、本暗号は、従来の暗号化方式では対応できなかった分野、例えば、医療データ等の個人情報の長期保存にも適用できる。
【００９３】
（Ｅ６）また、原本性保証装置１０は、再配置処理部６０を設けたことにより、同じ電子文章ｘと同じ第一の鍵（第一の擬似乱数列）ｒ（長さ：ｋビット）とから、$２^{ｋ}$通りの暗号文を生成できる。このことは、同じ電子文章ｘと暗号化の度に毎回変化する第一の鍵ｒとから、毎回異なる暗号文が（ｒの長さも変動するならば）無数にできることを意味する。
【００９４】
（Ｅ７）さらに、原本性保証装置１０においては、一体化処理部５０で用いられるＳ－ＢＯＸとして再配置表Ｋを利用することにより、その構成は０，１，・・・，ｎ－１のｎ個の整数をランダムに配置するだけのものになるので、従来の暗号化方式におけるＳ－ＢＯＸの構成よりも簡単であり、Ｓ－ＢＯＸの研究開発にコストがかからない。
【００９５】
（Ｅ８）さらに、原本性保証装置１０で採用した再配置暗号方式による暗号文の解読問題はＮＰ完全であることが予想される。
【００９６】
現在、世界標準の暗号として公開鍵暗号が広く採用されている。公開鍵暗号は、巨大数の素因数分解が現在のコンピュータ（ノイマン型コンピュータ）の能力では現実的な時間では行えないこと(素因数分解問題)などを安全性の根拠としている。しかし、近年急速に研究開発が進められている量子コンピュータを使うと、公開鍵暗号を解くために必要な素因数分解問題と離散対数問題を高速に解くことができることが証明されている（非特許文献４）。このことは、将来、量子コンピュータが実用化されると、公開鍵暗号は、標準的な暗号方式としては実質的に使用できなくなることを意味する。
【００９７】
しかし、当業者の間では、ＮＰ完全性を有する問題であれば量子コンピュータでも解く事ができないと考えられている。これに関し、原本性保証装置１０で採用した再配置暗号方式による暗号文の解読問題はＮＰ完全であることが以下のようにして予想される。予想

に当たって、まず次の３つの問題を設定する。

【００９８】

（Ｐ１）ナップサック問題（部分和問題）Z_0：決定問題としてのナップサック問題とは、容量ＣのナップサックとN個の品物A_i（容量c_i，価値v_i）がある場合（$i=1, 2, \cdots, N$）に、このナップサックに詰め込める品物の組み合わせの中で価値の総計が所定値Vとなる組み合わせがあるか否かを判定する問題である。ここで、C，N，c_i，v_i，Vはすべて自然数である。この問題において、すべての品物について$c_i=v_i$が成り立つ場合、これを部分和問題といい、以下のように定式化できる。

【００９９】

＜部分和問題＞

与えられた自然数x_1，x_2，・・・，x_N，yに対し、ある部分集合$I \subset \{1, 2, \cdots, N\}$が存在し、$y=\Sigma_{i \in I} x_i$とできるか？

（Ｐ２）和ジグソーパズル問題Z_1：和ジグソーパズル問題と称する問題を新たに設定する。

【０１００】

＜和ジグソーパズル問題＞

与えられた自然数x_1，x_2，・・・，x_N，yに対し、ある置換（Ｓ－ＢＯＸ）$s \in S_N$と自然数mが存在し、$y=\Sigma_{i=1}^{m} x_{s(i)}$とできるか？

（Ｐ３）再配置暗号ジグソーパズル問題Z_2：再配置暗号ジグソーパズル問題と称する問題を新たに設定する。

【０１０１】

＜再配置暗号ジグソーパズル問題＞

与えられた自然数の配列$X=(x_1, x_2, \cdots, x_N)$と自然数（平文）$W$に対して、再配置暗号のある秘密鍵Ｋ（Ｓ－ＢＯＸ又は再配置表）が存在し、$D_K(X)=W$とできるか？ここで、D_Kは秘密鍵Ｋを用いた再配置暗号の復号化関数である。

【０１０２】

このとき、再配置暗号による暗号文の解読問題がＮＰ完全であることの予想は、上記の３つの問題を多項式時間に帰着させることで次のようになされる。

【０１０３】

まず、ナップサック問題（部分和問題）Z_0はＮＰ完全であることが既に知られている。そして、明らかに、Z_0はZ_1に多項式時間に帰着できる。つまり、$Z_0 <_p Z_1$。

【０１０４】

次に、関数$f(X, y, s, m)=(E_s(X, y, m), (X, \Sigma_{i=1}^{m} x_{s(i)}, m), s)$を定義する。ここで、$E_s$は、$s$を用いた再配置暗号の暗号化関数である。関数$f$は、再配置関数$s$と暗号化関数$E_s$で計算されるわけだが、その計算時間はＯ（$n$）程度である場合には、$Z_1$は$Z_2$に多項式時間に帰着されることになる。つまり、この場合には、関数fは多項式時間計算可能関数であり、$Z_1 <_p Z_2$となる。ここで、$X=(x_1, x_2, \cdots, x_N)$への$s$の作用を$s(X)=(x_{s(1)}, x_{s(2)}, \cdots, x_{s(N)})$とし、$n$を入力のサイズ（バイト数）とする（$N \ll n$）。

【０１０５】

このようにして、Z_0，Z_1，Z_2の各問題を$Z_0 <_p Z_1 <_p Z_2$の順番で多項式時間に帰着させることができる場合には、Z_0及びZ_1はＮＰ完全であるので、結論として、Z_2もＮＰ完全であることが予想される。

【０１０６】

[受信装置としての原本性保証装置]

図８は、本発明の一実施の形態に係る原本性保証装置の復号検証処理に関する処理部の概略的な構成を示したブロック図である。ここでは、原本性保証装置１０が上記のようにして暗号化された暗号文fc_yを受信して復号化する機能を備えるものとして、原本性保証装置１０の受信装置としての側面について説明する。従って、以降の説明では、ある送信装置にて上記のようにして暗号化された暗号文fc_yが当該送信装置から原本性保証装

置１０に送信されたことを前提とする。また、以下では、原本性保証装置１０の暗号化処理に関する処理部と同じ機能を有する構成要素については同じ符号を付すことにする。

【０１０７】

原本性保証装置１０は、入力部（受信部）１２０と、公開可能な第二擬似乱数生成部４０と、逆一体化処理部１５０と、逆再配置処理部１６０と、出力部１７０と、記憶部８０と、制御部９０と、検査情報検証部１１０とを備える。このうち、記憶部８０と制御部９０とを除く部分を復号文生成部１０Ｂと称することにする。

【０１０８】

入力部１２０は、送信装置から送られてきた暗号文fc_yを受信するための入力インターフェースである。逆再配置処理部１６０は、再配置処理部６０と同様の構成を有し、後述する逆再配置処理を行う。逆一体化処理部１５０は、一体化処理部５０と同様の構成を有し、後述する逆一体化処理を行う。出力部１７０は、最終的に復号された復号文（電子文章ｘ）を出力すると共に、暗号文fc_yに含まれる検査情報ｒを検査情報検証部１１０へ出力するための出力インターフェースである。

【０１０９】

記憶部８０は、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、第二擬似乱数生成部４０、出力部１７０、及び検査情報検証部１１０から成る復号文生成部１０Ｂが生成した各種のデータの格納を行うサブメモリと、後述する復号検証処理の各ステップを実行するためのコンピュータに読み取り可能な暗号プログラムを格納するメインメモリとから構成される。

【０１１０】

制御部９０は、記憶部８０から読み出した復号検証プログラムに従って、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、第二擬似乱数生成部４０、出力部１７０、記憶部８０、及び検査情報検証部１１０を制御するＣＰＵを備える。

【０１１１】

本実施の形態では、原本性保証装置１０を、復号文生成部１０Ｂ及び制御部９０と、記憶部８０とを一体化した構成としたが、記憶部８０を独立した記憶装置として復号文生成部１０Ｂ及び制御部９０とから切り離した構成としてもよい。いずれの構成においても、原本性保証装置１０はコンピュータによって実現されるものであり、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、第二擬似乱数生成部４０、出力部１７０、及び検査情報検証部１１０は、制御部９０により記憶部８０から読み出された復号検証プログラムに従って制御される。

【０１１２】

［復号化処理］

以上を前提として、図９に示した原本性保証装置１０によって行われる復号化処理について詳細に説明する。図９は、図８に示した原本性保証装置１０における復号検証処理の手順を示したフローチャートである。

【０１１３】

送信装置から送信された暗号文$fc_y=(f_0, f_1, \cdots, f_{nm-1})$が入力部１２０から入力されると、制御部９０は、これを記憶部８０に記憶させ、記憶部８０に格納された復号検証プログラムに従い、逆再配置処理部１６０、逆一体化処理部１５０、第二擬似乱数生成部４０、出力部１７０、及び検査情報検証部１１０に対して以下に示す処理を行うように促す。

【０１１４】

ステップＳ２００において、制御部９０は、送信装置から送信された暗号文fc_yの長さｎｍを入力部２０から読み込ませ、これを記憶部８０に記憶させると共に、暗号文fc_yを第五の電子データとして記憶部９０に記憶させる。

【０１１５】

次に、ステップＳ２１０において、制御部９０は、記憶部８０から暗号文fc_yの長さｎｍと予め格納された分割数ｎとを読み出す。

【０１１６】
次に、ステップＳ２２０において、制御部９０は、読み出した第五の電子データとしての暗号文fc_yの長さｎｍと分割数ｎとから、ブロックデータの長さｍをｍ＝ｎｍ／ｎとして算出する。
【０１１７】
次に、ステップＳ２３０において、制御部９０は、ステップＳ２４０～Ｓ２７０で行われる分割処理と逆再配置処理と逆一体化処理とを１セットとした処理のラウンド数を表すフラグＣｔを立て（Ｃｔ＝０）、ステップＳ２４０へ処理を進める。
【０１１８】
ステップＳ２４０において、制御部９０は、記憶部８０から第五の電子データとしての暗号文$fc_y=(f_0, f_1, \cdots, f_{nm-1})$を読み出し、これをｎ個のブロックデータに分割して、分割されたデータを$d_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$として記憶部８０に記憶させる。
【０１１９】
次に、ステップＳ２５０において、制御部９０は、記憶部８０からデータ$d_y=(b_{k[0]}, b_{k[1]}, \cdots, b_{k[n-1]})$と秘密鍵Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）とを読み出し、逆再配置処理１６０に対して、図１０に示すように、ｋ［ｉ］番目のブロックデータ$b_{k[i]}$をb_iへと逆配置させたデータ$d_y=(b_0, b_1, \cdots, b_{n-1})$を第四の電子データとして記憶部８０に記憶させる。
【０１２０】
次に、ステップＳ２６０において、制御部９０は、記憶部８０から第四の電子データ$d_y=(b_0, b_1, \cdots, b_{n-1})$を読み出し、逆一体化処理部１５０に対して、図１１（Ｂ）のステップＳ２６１－２～Ｓ２６４－２に示すように、$c_{(dm+i) \bmod nm}=c_{(dm+i) \bmod nm}+c_{i \bmod nm}$（ｉ＝０，１，・・・，ｎｍ－１）として加算した値（左辺の$c_{(dm+i) \bmod nm}$）をＳ－ＢＯＸの逆関数を用いてバイト単位で変換して逆一体化する処理を第二の逆一体化関数系Ｇ（ｉｓ，ｍ，ｄ）により行わせる（図１１（Ｃ）のステップＳ２６０－１）。その後、制御部９０は、逆一体化処理部１５０に対して、図１１（Ａ）に示すように、第二の逆一体化関数系Ｇ（ｉｓ，ｌ，ｄ）により逆一体化された電子データをＳ－ＢＯＸの逆関数を用いてバイト単位で逆一体化して、逆一体化されたデータを$d_y=(d_0, d_1, \cdots, d_{nm-1})$を$c_{(dm+i) \bmod nm}=c_{(dm+i) \bmod nm}-c_{i \bmod nm}$（ｉ＝ｎｍ－１，ｎｍ－２，・・・，１，０）として減算する処理を第一の逆一体化関数系Ｆ（ｉｓ，ｍ，ｄ）を用いて行わせ、逆一体化されたデータ$c_y=(c_0, c_1, \cdots, c_{nm-1})$を第三の電子データとして記憶部１８０に記憶させる（図１１（Ｃ）のステップＳ２６０－２）。
【０１２１】
ここで、逆一体化とは、Ｓ－ＢＯＸの関数ｓの逆関数ｉｓを用いた逆変換のことであり、次のプログラムで実現される。
【０１２２】
```
for i:=0 to 255 do
  is[s[i]]:=i;
```
次に、ステップＳ２７０において、制御部９０は、フラグＣｔの値をインクリメントし（Ｃｔ＝１）、ステップＳ２８０において、インクリメントされた値が所定のラウンド数ｈを越えたか否かを判定する。
【０１２３】
ステップＳ２８０において、制御部９０によりインクリメントされた値が所定のラウンド数ｈを越えたと判定された場合には、ステップＳ２９０に処理を進める。
【０１２４】
ステップＳ２９０において、制御部９０は、記憶部８０から第三の電子データ$c_y=(c_0, c_1, \cdots, c_{nm-1})$と予め格納された数値“ａ”とを読み出し、第三の電

子データc_y＝（c_0，c_1，・・・，c_{nm-1}）の先頭からａワードを擬似乱数列のヘッダｒとして規定し、第三の電子データc_yを改めてc_y＝（ｒ（擬似乱数列のヘッダ），c_y（残りのデータ：第二の電子データ））として記憶部８０に記憶させる。

【０１２５】

次に、ステップＳ３００において、制御部９０は、記憶部８０から第三の電子データc_y＝（ｒ，c_y）を読み出し、第二擬似乱数生成部１４０に対して、擬似乱数列のヘッダｒを初期値としたｎｍ－ａワードの擬似乱数列r_y＝（r_0，r_1，・・・，r_{nm-a-1}）を生成すると共に、擬似乱数列のヘッダｒを検査情報検証部１１０に出力する。

【０１２６】

そして、ステップＳ３１０において、制御部９０は、第二の電子データc_y＝（c_{a+1}，c_{a+2}，・・・，c_{nm-a-1}）と生成された擬似乱数列r_y＝（r_0，r_1，・・・，r_{nm-a-1}）とを排他的論理和する（$x_{a+i}=c_{a+i}$ ＸＯＲ r_i（ｉ＝０，１，・・・，ｎｍ－ａ－１））ことにより、第一の電子データｙ＝（x_a，x_{a+1}，・・・，x_{nm-1}）を算出し、次いで、第一の電子データｙの先頭からｑワードを電子文章ｘのデータ長ｇなどの情報を含んだヘッダ情報ｕと規定し、さらに、残りのデータの先頭からｇワードのみを電子文章ｘと規定し、その次のａワードを検査情報ｒと規定し、これらを記憶部１８０に記憶させる。

【０１２７】

なお、最後に残ったｎｍ－ｑ－ｇ－ａワードのデータはパディングである。

【０１２８】

次に、ステップＳ３２０において、制御部９０は、記憶部８０から電子文章ｘと検査情報ｒとを読み出し、出力部７０に出力させ、検査情報ｒをさらに検査情報検証部１１０に出力する。

【０１２９】

そして、ステップＳ３３０において、制御部９０は、検査情報検証部１１０に対して、擬似乱数列のヘッダｒと検査情報ｒとを比較させ、それらの値が一致するか否かを検証する。

【０１３０】

一方、ステップＳ２８０において、制御部１９０によりインクリメントされた値が所定のラウンド数ｈを越えていないと判定された場合には、ステップＳ２４０に戻って、インクリメントされた値が所定のラウンド回数ｈを越えるまで、ステップＳ２４０～Ｓ２７０までの処理を繰り返した後、ステップＳ２９０へ進む。

【０１３１】

このように、復号文生成部１０Ｂは、送信装置から送信されてきた暗号文fc_yを復号するための情報として、秘密鍵としての再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）と、分割数ｎと、ヘッダの長さａと、ブロック差分ｄとを暗号文生成部１０Ａと共有している。これにより、復号化処理部１０Ｂは、復号検証処理の過程で暗号文fc_yの本当の鍵とも言える擬似乱数列のヘッダｒが入手でき、電子文章ｘの長さｇも同様に入手できるので、検査情報ｒと擬似乱数列のパディングｐとを規定することができる。

【０１３２】

［原本性保証］

図１５は、図２に示した原本性保証装置１０を用いた電子文章の原本性保証の方法を示す図である。本図において、送信装置及び受信装置は共に原本性保証装置１０を用いて構成されているものとする。なお、以下の説明においては、図２に示したヘッダ情報ｕとパディングｐとは本質ではないので無視する。

【０１３３】

始めに、送信装置は、送信者が送りたい電子文章ｘに秘密の擬似乱数列ｒをヘッダ情報として付加すると共に、検査情報として秘密の擬似乱数列ｒを電子文章ｘのフッタ情報として付加する。次に、送信装置は、このようにしてヘッダ情報とフッタ情報として秘密の擬似乱数列ｒが付加された電子文章ｘを擬似乱数列ｒと再配置表Ｋとを用いて再配置暗号

化した暗号データを生成し、この暗号データを送信装置（送信者）に送信する。すると、受信装置は、受け取った暗号データを再配置表Ｋを用いて復号化して、復号化されたデータからヘッダ情報とフッタ情報とを切り出し、両者を比較する。すると、送信装置から受信装置に至る経路において第三者により電子文章が改竄された場合には、両者の値は一致しない。逆に言えば、ヘッダ情報とフッタ情報とが一致すれば、電子文章ｘの原本性が送信者にとって保証されたことになる。

【０１３４】

［受信装置としての原本性保証装置１０の効果］

このように、原本性保証装置１０による原本性保証にはハッシュ関数を用いる必要がない。さらに、原本性保証装置１０による再配置暗号方式は共通鍵方式に属するので、ＰＫＩのような認証局による鍵の認証は不要である。

【０１３５】

受信装置としての原本性保証装置１０に上記のような原本性保証を可能とさせているものは、復号時に十分な攪拌が行われるという逆一体化処理部１５０の特徴にある。この特徴は、逆一体化処理部１５０における逆一体化処理が対称性一体化関数系の逆関数系を用いて行われることによる。この特徴ゆえ、検査情報が偶然一致する確率を極めて低いものにすることができる。

【０１３６】

再配置暗号方式では、第一の（秘密の）擬似乱数列ｒを電子文章ｘに添付すると共に、第一の擬似乱数列ｒを初期値として第三の（公開可能な）擬似乱数列r_yを生成して電子文章ｘを変換するので、第一の擬似乱数列ｒと電子文書ｘに添付した検査情報ｒが復号時に偶然一致する確率は、第一の擬似乱数列ｒの長さがａバイトであるとき、2^{-8a}程度でしかない。例えば、第一の擬似乱数列ｒの長さを１２８バイトとすると、暗号文を改竄して第一の擬似乱数列ｒと電子文書ｘに添付した検査情報ｒが復号時に偶然一致する確率は、$2^{-8\times128}=2^{-1024}$程度と限りなく小さい。

【０１３７】

また、再配置暗号方式では、同一の電子文章ｘでも暗号化の度に異なるランダムな暗号文に変換されるので、第三者が偽造に成功したかどうかを確認できない。この点、公開鍵暗号方式を用いた原本性保証では、第三者が偽造に成功したかどうかが確認できてしまう。

【０１３８】

［原本性保証装置のその他の構成］

上記した実施の形態においては、暗号化処理部１０Ａと復号化処理部１０Ｂとを同じ原本性保証装置１０の中で実現する構成としたが、これは暗号化処理と復号化処理とが可逆の関係にあるからである。しかし、必要に応じて、暗号化処理部１０Ａと復号化処理部１０Ｂとを別体の装置として構成してもよい。

【０１３９】

［具体的な実装例］

上記した実施の形態における暗号化プログラム及び復号化プログラムの実装例を示す。ここでは、１ ワードを１バイトとし、疑似乱数列のヘッダｒは１２８バイトを使用する。第一疑似乱数生成部３０で生成する疑似乱数としては、コンピュータプログラミング環境で使用できる疑似乱数を用いる。具体的には、原本性保証装置のシステムクロックや送信者による入力部２０からの入力のタイミングなどを使用して再現しにくい疑似乱数列をヘッダrとして使用する。また、分割数ｎはｎ＝２５６、電子文章ｘのデータ長ｇなどの情報を含むヘッダ情報ｕはｕ＝４、ブロック差分ｄ＝１とする。このとき、秘密鍵ＫはＫ＝（ｋ［０］，ｋ［１］，・・・，ｋ［２５５］）として表される再配置表であり、ｋ［ｉ］（ｉ＝０，１，・・・，２５５）には、０，１，・・・，２５５を並べ替えた値が格納されている。

【０１４０】

なお、以降のプログラムの変数の中には、上記した実施の形態で使用した変数名が異な

るものもあるが、混乱することはないはずである。

【０１４１】

疑似乱数列のヘッダｒ、ヘッダ情報ｕ、電子文章ｘ、検査情報としての擬似乱数列ｒを合わせた全データが２５６バイトの倍数になるように、電子文章ｘの末尾に適当な長さｖの疑似乱数をパディングする。そして、これを改めてｘ＝（x_0，x_1，・・・，x_{256m-1}）とする。ここで、x_i（ｉ＝０，１，・・・，２５５）は、ＬｏｎｇＷｏｒｄ（４バイト符号なし整数）である。

【０１４２】

＜公開可能な疑似乱数生成部４０の実装例＞

```
noise: array[0..127] of byte;
i: integer; //iはグローバル変数
i:=0;
function g1: byte; //g1はPascal のローカル関数
var
  c,cc,r: byte;
begin
  c:=i and 127;
  cc:=(i+127) and 127;
  r:=noise[c]+noise[cc];
  noise[c]:=r;
  g1:=k[r];
end;
```

このような疑似乱数生成部４０から生成される疑似乱数r_yと、ヘッダ情報ｕ、電子文章ｘ、検査情報としての擬似乱数列ｒ、パディングｐと、を排他的論理和して暗号化したものを、ＬｏｎｇＷｏｒｄの配列として、改めてｘ＝（ｘ［０］，ｘ［１］，・・・，ｘ［ｖ－１］）（ｖ＝２５６ｍ）とする。ここで、ｍは、ブロックデータの長さをＬｏｎｇＷｏｒｄの個数で表したものである。

【０１４３】

なお、次の実装例では、図１の表記を合わせて、擬似乱数のヘッダｒの長さはａ＝１２８バイト、ヘッダ情報ｕの長さはｑ＝４バイト、電子文章ｘの長さはｇバイト、パディングｐの長さはｖバイトとしている。また、繰り返し回数ｒｎは適宜指定することができるが、ｒｎ＝４程度に設定すればよい。

【０１４４】

＜一体化処理の実装例＞

```
  for j :=0  to rn-1 do
  begin
    x[0]:=x[0]+x[128+4+g+256+128+v];
    x[0]:=k[x[0]];
    for i :=1  to 128+4+g+256+128+v do
    begin
      x[i]:=x[i]+x[i-1];
      x[i]:=k[x[i]];
    end;
  end;
```

＜逆一体化処理の実装例＞

```
 for j :=0  to rn-1 do
 begin
   for i :=128+4+g+256+128+v  downto 1 do
   begin
     x[i]:=k[x[i]];
```

```
      x[i]:=x[i]-x[i-1];
    end;
    x[0]:=k[x[0]];
    x[0]:=x[0]-x[128+4+g+256+128+v];
  end;
```

この処理の後、ｘをｘ＝（y_0，y_1，・・・，y_{255}）と２５６分割する。ここで、y_i＝（ｘ［ｍｉ］，ｘ［ｍｉ＋１］，・・・，ｘ［ｍｉ＋ｍ－１］）（ｉ＝０，１，・・・，２５５）である。

【０１４５】

＜再配置処理の実装例＞

ｘと同じ長さの配列ｙを準備し、以下のように再配置処理する。ここで、命令Ｍｏｖｅ（ｘ［ｉ］，ｙ［ｊ］，ｚ）は、ｘ［ｉ］のアドレスからｙ［ｊ］のアドレスへzバイトの記憶内容をコピーする処理を表す。この処理によって、配列ｘの内容を再配置表Ｋによってブロック単位で並べ替えることができる。

【０１４６】

```
  i: integer;
  begin
    for i:=0 to 255 do
    begin
      Move(x[i*m], y[k[i]*m], m);
    end;
    Move(y[0], x[0], v);
  end;
```

なお、復号時に使用する逆再配置処理は以下のようにすればよい。

【０１４７】

```
  i: integer;
  begin
    for i:=0 to 255 do
    begin
      Move(x[k[i]*m], y[i*m], m);
    end;
    Move(y[0], x[0], 1);
  end;
```

この実装例は、１６８３ビットの鍵長のブロック暗号程度の安全性を持ち、ＡＥＳの１０分の１程度の演算回数で暗号化処理できる。

【０１４８】

［変更例に係る送信装置としての原本性保証装置］

上記した実施の形態では、第二擬似乱数生成部４０を用いたストリーム暗号によって高速な原本性装置１０をデザインした。その変更例として、ストリーム暗号をブロック暗号に変えた構成を有する原本性保証装置が考えられる。

【０１４９】

図１２は、図２に示した原本性保証装置の一変更例の暗号化処理に関する処理部の概略的な構成を示したブロック図である。原本性保証装置２００は、入力部２０と、第一擬似乱数生成部３０と、ブロック暗号文生成部２４０と、一体化処理部５０と、再配置処理部６０と、出力部７０と、記憶部８０と、制御部９０と、検査情報付加部１００とを備える。このうち、記憶部８０と制御部９０とを除く部分を暗号文生成部２００Ａと称することにする。

【０１５０】

このうち、入力部２０と、第一擬似乱数生成部３０と、一体化処理部５０と、再配置処理部６０と、出力部７０と、検査情報付加部１００とは、それぞれ、図２に示した原本性

保証装置１０の入力部と、第一擬似乱数生成部と、一体化処理部と、再配置処理部と、出力部と、検査情報付加部と同様の機能を有するので、同じ符号を付すことにより、それらの構成及び機能の説明を省略する。

【０１５１】

記憶部８０は、入力部２０、第一擬似乱数生成部３０、一体化処理部５０、再配置処理部６０、出力部７０、検査情報付加部１００、ブロック暗号生成部２４０から成る暗号生成部２００Ａが生成した各種のデータの格納を行うサブメモリと、後述する暗号化処理の各ステップを実行するためのコンピュータに読み取り可能な暗号プログラムを格納するメインメモリとから構成される。

【０１５２】

また、制御部９０は、記憶部８０から読み出した暗号化プログラムに従って、入力部２０、第一擬似乱数生成部３０、一体化処理部５０、再配置処理部６０、出力部７０、検査情報付加部１００、ブロック暗号生成部２４０、記憶部８０を制御するＣＰＵを備える。

【０１５３】

なお、記憶部８０及び制御部９０の構成は、上記した実施の形態と同様の構成なので、それらの構成及び機能の説明を省略する。

【０１５４】

［変形例に係る暗号化処理］

以上を前提として、図１２に示した原本性保証装置２００によって行われる暗号化処理について上記した実施の形態と異なる部分についてのみ説明する。図１３は、図１２に示した原本性保証装置２００によって行われる暗号化処理の手順を示した概念図である。

【０１５５】

原本性保証装置２００は、上記した実施の形態におけるストリーム暗号の代わりにブロック暗号を用いて暗号化処理を実現するものである。

【０１５６】

従って、ブロック暗号文生成部２４０において電子文章ｘをブロック暗号化する際に、電子文章ｘと検査情報としての第一の擬似乱数ｒとの和の長さｇ＋ａがブロック長の倍数とならない場合には第一の電子データｙの最後にパディングをする必要が生じる。このとき、制御部９０は、ｖ≡－２ａ－ｇ－ｑ（ｍｏｄ　ｎ）となるような最小の非負整数ｖを算出し、パディングｐとしてｖワードの長さの擬似乱数列ｚを第一擬似乱数生成部３０に生成させ、第一の電子データｙの最後に付加する処理を行う。そして、電子文章ｘのデータ長ｇなどの情報を含んだヘッダ情報ｕ（長さ：ｑワード）、電子文章ｘ、検査情報としての第一の擬似乱数列ｒ、パディングｑを合わせたデータを改めて第一の電子データｙ＝（ｙ，ｚ）とした後、ブロック暗号化のステップへと処理を進める。この場合、再配置処理において分割される各ブロックの長さを表す整数“ｍ”は、ｍ＝（２ａ＋ｑ＋ｇ＋ｐ）／ｎとして算出される。

【０１５７】

次のステップとして、制御部９０は、記憶部８０から電子文章ｘ（ｇワード）、電子文章ｘのデータ長などの情報を含んだヘッダ情報（ｑワード）、検査情報としての第一の擬似乱数列ｒ（ａワード）、パディング（ｐワード）を読み出し、ブロック暗号文生成部２４０に対して、これらのデータを第一の擬似乱数列ｒを鍵として、公知のブロック暗号化を行わせる。

【０１５８】

以降の処理は、ストリーム暗号を用いた上記した実施の形態と同様なので、説明を省略する。

【０１５９】

［変更例に係る送信装置としての原本性保証装置２００の効果］

変形例の一部で使用したブロック暗号化の手法自体は、ＮＭＲ量子コンピュータにおけるGroverのアルゴリズムで攻撃されることが知られている。例えば、ＡＥＳは128ビットの鍵の場合、古典的なコンピュータでは全数探索では2^{128}通りの鍵を確かめなければ

ならないわけだが、ＮＭＲ量子コンピュータを使うと2^{64}通りの鍵を探索する計算量で暗号を破ることができる。

【０１６０】

一方、本変形例に係る再配置暗号方式は分割数を簡単に増加できるので、Groverのアルゴリズムに対しても十分な強度を維持できる。例えば、標準的な２５６分割の再配置暗号でも鍵の総数は２５６！　2^{1684}通りあり、ＮＭＲ量子コンピュータによる攻撃を受けても、その計算量は（２５６！）$^{1/2}$　2^{842}となり、実際問題として攻撃は全く成功しない。

【０１６１】

また、その他にもっと効率的なアルゴリズムが出現したとしても、例えば５１２分割の再配置暗号の速度はほとんど変化しないが、鍵の総数は５１２！　2^{3875}通りとなり、量子コンピュータの計算量は（５１２！）$^{1/2}$　2^{1938}となってしまう。

【０１６２】

このようにどんな攻撃方法を考案しても、再配置暗号では分割数を増加すると、暗号の強度が指数関数的に増大して攻撃を振り切ってしまうと考えられ、再配置暗号の解読問題はＮＰ完全性をもつと期待される状況にある。従って、変形例に係る送信装置としての原本性保証装置２００は、ブロック暗号と比較した場合でも、十分な効果を持つと言える。

【０１６３】

[変更例に係る受信装置としての原本性保証装置]

図１４は、図１２に示した原本性保証装置の復号化処理に関わる処理部の概略的な構成を示したブロック図である。ここでは、原本性保証装置２００が上記のようにして暗号化された暗号文fc_yを受信して復号化する機能を備えるものとして、原本性保証装置２００の受信装置としての側面について説明する。従って、以降の説明では、ある送信装置にて上記のようにして暗号化された暗号文fc_yが当該送信装置から原本性保証装置２００に送信されたことを前提とする。また、以下では、原本性保証装置２００の暗号化処理に関する処理部と同じ機能を有する構成要素については同じ符号を付すことにする。

【０１６４】

原本性保証装置２００は、入力部（受信部）１２０と、逆再配置処理部１６０と、逆一体化処理部１５０と、ブロック暗号文復号部３４０と、出力部１７０と、検査情報検証部１１０と、記憶部８０と、制御部９０と、を備える。このうち、記憶部８０と制御部９０とを除く部分を復号文生成部２００Ｂと称することにする。

【０１６５】

入力部１２０は、送信装置から送られてきた暗号文fc_yを受信するための入力インターフェースである。逆再配置処理部１６０は、再配置処理部６０と同様の構成を有し、後述する逆再配置処理を行う。逆一体化処理部１５０は、一体化処理部５０と同様の構成を有し、後述する逆一体化処理を行う。ブロック暗号文復号部３４０は、ブロック暗号文生成部２４０と同様の構成を有し、後述するブロック暗号文を復号化する。出力部１７０は、最終的に復号された復号文（電子文章ｘ）を出力すると共に、暗号文fc_yに含まれる検査情報ｒを検査情報検証部１１０へ出力するための出力インターフェースである。

【０１６６】

記憶部８０は、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、ブロック暗号文復号部３４０、出力部１７０、及び検査情報検証部１１０から成る復号文生成部１０Ｂが生成した各種のデータの格納を行うサブメモリと、後述する復号検証処理の各ステップを実行するためのコンピュータに読み取り可能な暗号プログラムを格納するメインメモリとから構成される。

【０１６７】

制御部９０は、記憶部８０から読み出した復号検証プログラムに従って、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、ブロック暗号文復号部３４０、出力部１７０、記憶部８０、及び検査情報検証部１１０を制御するＣＰＵを備える。

【０１６８】

本実施の形態では、原本性保証装置２００を、復号文生成部２００Ｂ及び制御部９０と、記憶部８０とを一体化した構成としたが、記憶部８０を独立した記憶装置として復号文生成部２００Ｂ及び制御部９０とから切り離した構成としてもよい。いずれの構成においても、原本性保証装置２００はコンピュータによって実現されるものであり、入力部１２０、逆再配置処理部１６０、逆一体化処理部１５０、ブロック暗号文復号部３４０、出力部１７０、及び検査情報検証部１１０は、制御部９０により記憶部８０から読み出された復号検証プログラムに従って制御される。

【０１６９】

［変形例に係る復号化処理］

以上を前提として、図１４に示した原本性保証装置２００によって行われる復号化処理について説明する。原本性保証装置２００によって行われる復号検証処理は、ストリーム暗号がブロック暗号に変わっただけで、本質的な部分は、図８に示した原本性保証装置１０における復号検証処理の方法と同じであるので、図８を参照して、説明を簡略化する。

【０１７０】

送信装置から送信された暗号文$fc_y=(f_0, f_1, \cdots, f_{nm-1})$が入力部１２０から入力されると、制御部９０は、これを記憶部８０に記憶させ、記憶部８０に格納された復号検証プログラムに従い、逆再配置処理部１６０、逆一体化処理部１５０、ブロック暗号文復号部部３４０、出力部１７０、及び検査情報検証部１１０に対して以下に示す処理を行うように促す。

【０１７１】

まず、ステップＳ２００～２９０において、制御部９０は、逆再配置処理部１６０、逆一体化処理部１５０に対して、図８に示したステップＳ２００～２９０の処理を行わせる。

【０１７２】

次に、制御部９０は、記憶部８０から第三の電子データ$c_y=(r, c_y)$を読み出し、擬似乱数列のヘッダｒを鍵として、第二の電子データ$c_y=(c_{a+1}, c_{a+2}, \cdots, c_{nm-a-1})$を公知のブロック復号化して第一の電子データ$y=(x_a, x_{a+1}, \cdots, x_{nm-1})$を算出すると共に、擬似乱数列のヘッダｒを検査情報検証部１１０に出力する。

【０１７３】

次に、制御部９０は、第一の電子データｙの先頭からｑワードを電子文章ｘのデータ長ｇなどの情報を含んだヘッダ情報ｕと規定し、さらに、残りのデータの先頭からｇワードのみを電子文章ｘと規定し、その次のａワードを検査情報ｒと規定し、これらを記憶部１８０に記憶させる。なお、最後に残ったｎｍ－ｑ－ｇ－ａワードのデータはパディングである。

【０１７４】

次に、制御部９０は、記憶部８０から電子文章ｘと検査情報ｒとを読み出し、出力部７０に出力させ、検査情報ｒをさらに検査情報検証部１１０に出力する。

【０１７５】

そして、制御部９０は、検査情報検証部１１０に対して、擬似乱数列のヘッダｒと検査情報ｒとを比較させ、それらの値が一致するか否かを検証する。

【０１７６】

一方、ステップＳ２８０において、制御部９０によりインクリメントされた値が所定のラウンド数ｈを越えていないと判定された場合には、ステップＳ２４０に戻って、インクリメントされた値が所定のラウンド回数ｈを越えるまで、ステップＳ２４０～Ｓ２７０までの処理を繰り返した後、ステップＳ２９０へ進む。

【０１７７】

このように、復号文生成部２００Ｂは、送信装置から送信されてきた暗号文fc_yを復号するための情報として、秘密鍵としての再配置表Ｋ＝（ｋ［０］，ｋ［１］，・・・，ｋ［ｎ－１］）と、分割数ｎと、ヘッダの長さａと、ブロック差分ｄとを暗号文生成部２

００Ａと共有している。これにより、復号化処理部１０Ｂは、復号検証処理の過程で暗号文ｃｆ$_y$の本当の鍵とも言える擬似乱数列のヘッダｒが入手でき、電子文章ｘの長さｇも同様に入手できるので、検査情報ｒと擬似乱数列のパディングｐとを規定することができる。

【０１７８】

この変更例は暗号化処理にブロック暗号を用いているので、上記した実施の形態のようにストリーム暗号を用いた場合と比べて暗号化の実行速度は遅くなるが、その代わりに、従来使用されているＡＥＳなどの共通鍵方式を用いた暗号装置への実装が容易であるといった利点がある。

【０１７９】

［変更例に係る受信装置としての原本性保証装置２００の効果］

上記の変更例においては、電子文章の原本性保証のために、再配置暗号方式をストリーム暗号の変わりにブロック暗号において用いた。ＡＥＳを含む従来のブロック暗号では、暗号化モジュールを納入するソフトウェア業者に不正があった場合、これを避けることが困難であるといった問題がある。これをインサイダー攻撃と称する。

【０１８０】

例えば、ブロック暗号を通常モードで使用する場合を想定してみる。電子文章ｘを秘密鍵Ｋで暗号化して暗号文Ｙ＝Ｅ（Ｋ，ｘ）を出力する暗号化モジュールを納入するとき、その代わりに業者が次のような暗号化モジュールを納入するとする。それは、適当な条件下で暗号文としてＹではなく特定のブロックＺを追加した、（Ｙ，Ｚ）を出力するような暗号化モジュールである。ここで、ブロックＺは、特定の暗号によって秘密鍵Ｋを暗号化したブロックである。ユーザが業者からこのようなインサイダー攻撃を受けると、ワンタイムパッドなどを使用した場合でも、リアルタイムで暗号が破られてしまう。この状況をユーザから見た場合、暗号文を復号する際に、時々文字化けが起こるようにしか見えない。従って、ユーザはそれを通信回線の事情による文字化けと区別できない。さらに、この攻撃を圧縮技術などと組み合わせると、ユーザはブロックの長さを調べたとしてもインサイダー攻撃と通信回線の事情による文字化けとの識別ができない。また、たとえ暗号化モジュールの異常を察知したとしても、ウィルスなどによって持ち込まれたモジュールと区別することができない。

【０１８１】

これに対する対応策として考えられるのは、ユーザが暗号化モジュールを購入の際に、業者にソースファイルを納入させて、プログラムを十分確認した上で、コンパイルして使用するという方法である。しかし、それでも上記のようなインサイダー攻撃を完全に排除することは現実問題として容易ではない。従来のブロック暗号においても、ブロック暗号を通常モードではなくＣＢＣモードで使用すれば、このような攻撃はある程度排除できる。しかし、（Ｚ，Ｙ）のように暗号文Ｙのヘッダとしてブロックを乗せられると攻撃が成功する。これに対処するためには、さらに、ＣＢＣモードも２周以上暗号化する必要がある。しかし、それをした場合、ブロック暗号は現在の使用速度よりも２倍以上遅くなってしまう。特に、ＣＢＣモードの初期ベクトルとしてブロックＺを用いる方法に対しては、その対応が難しい。

【０１８２】

しかし、上記のような再配置暗号を用いれば、このようなインサイダー攻撃を原理的に検出できる。なぜならば、これは暗号文の改竄に該当するため、原本性が保証できないからである。すなわち、第三者が原本性をチェックするプログラムを公開すれば、本発明に対するインサイダー攻撃は排除できる。

【０１８３】

［デジタル証拠としての利用］

上記した実施の形態における原本性保証方法をユーザ個人と認証局とで重複して用いることで、例えば以下のようにしてデジタル証拠を実現することができる。

【０１８４】

（ステップ１）何らかの方法（量子鍵配送など）で認証局Ａとユーザ Ｕとが再配置暗号の共通鍵K_0を共有する。

【０１８５】

（ステップ２）ユーザＵが秘密鍵K_Uを用いて電子文章ｘを再配置暗号化して、暗号文Ｃ＝Ｅ（K_U，ｘ）を生成する。

【０１８６】

（ステップ３）ユーザＵは、共通鍵K_0を用いて暗号文Ｃを認証局Ａに暗号化して送信する。

【０１８７】

（ステップ４）認証局Ａは、ヘッダとして受信日時、時間など証拠として必要な情報（証拠情報）Ｈを付加して認証局の鍵K_Aを用いて暗号化し、デジタル証拠Ｍ＝Ｅ（K_A，Ｈ，Ｃ）を作成する。

【０１８８】

（ステップ５）必要があれば、認証局Ａは、デジタル証拠Ｍを復号してＣのメッセージダイジェストｍを作成し、鍵K_0を用いて暗号化してユーザＵに送信する。

【０１８９】

（ステップ６）認証局Ａは、鍵K_0を用いてデジタル証拠Ｍを暗号化してユーザＵに送信する。

【０１９０】

（ステップ７）ユーザＵは、ＰＫＩにてデジタル証拠Ｍを公開したり、複数のオンラインストレージなどに保存する。

【０１９１】

（ステップ８）ユーザＵは、裁判などでデジタル証拠Ｍが必要になったら、裁判所にデジタル証拠Ｍと秘密鍵K_Uを提出する。

【０１９２】

（ステップ９）裁判所は、認証局Ａにデジタル証拠Ｍを復号させて、証拠情報Ｈとユーザ Ｕの秘密鍵K_Uを入手する。

【０１９３】

（ステップ１０）裁判所は、Ｄ（K_U，Ｃ）＝ｘを計算し、証拠情報Ｈと電子文章ｘとを入手し、証拠を確認する。

【０１９４】

このようにすれば、公開鍵暗号が量子コンピュータで攻撃された場合であってもなお有効なデジタル証拠を構成できることが期待される。それは、既に述べたように、再配置暗号を解かずにデジタル証拠Ｍを改竄することは不可能だからである。また、再配置暗号方式では、同一の電子文章でも毎回異なる暗号文が生成されるので、第三者は改竄に成功したかどうかを確認することもできない。

【０１９５】

［検査情報を用いた暗号内通信］

上記した実施の形態においては、電子文章の原本性保証のために検査情報として第一の擬似乱数ｒを用いたわけだが、これ以外に検査情報として以下のような情報を電子文章ｘに付加することにより、再配置暗号に様々な検査能力を付与することができる。このような情報通信は暗号内での通信なので、暗号化した本人の同意がなければ確認できない。

【０１９６】

（実施例１）電子文章の信頼性保証技術

検査情報としてＭＡＣアドレス、ホスト名、ＩＰアドレス、鍵ファイル名などを電子文章ｘに付加すれば、暗号の管理状況を確認することが可能となり、電子文章の信頼性を保証する技術が実現する。これにより、厳格に管理されている秘密鍵を使用した文章であることが確認できる。

【０１９７】

（実施例２）ソフトウェア不正使用防止技術

検査情報としてＭＡＣアドレス、ユーザ名、コンピュータ名などを電子文章ｘに付加すれば、再配置暗号を含むシステムを使用したユーザを特定する証拠を残すことができる。これにより、システムの使用許諾の有無を、本人の同意の上、秘密鍵の提供の下で確認できる。

【０１９８】

［メッセージ認証符号を用いた電子文章の原本性保証方法との比較］

最後に、再度、メッセージ認証符号を用いた電子文章の原本性保証方法との比較した場合の、上記実施の形態及びその変更例の効果を箇条書きにて示す。

【０１９９】

（効果１）上記実施の形態及びその変更例では、脆弱性が指摘されているハッシュ関数を使用せずに原本性保証を実現している。

【０２００】

（効果２）上記実施の形態及びその変更例では、毎回異なる暗号文が生成されるため衝突が確認できず攻撃が困難である。

【０２０１】

（効果３）上記実施の形態及びその変更例では、演算回数が少なく高速である。演算回数はＡＥＳの十分の一程度なので、原本性保証技術としてはハッシュ関数の計算まで含めると従来技術の二十分の一以下の演算回数で原本性保証機能を実現している。もちろん、この比率はハッシュ関数や暗号の実装の仕方にも依存するが、従来技術と比較して著しく高速であることには変わりはない。

【０２０２】

（効果４）上記実施の形態及びその変更例では、擬似乱数ヘッダｒを長くすることで原本性保証の信頼性も制限無く高めることができる。これが、メッセージ認証符号を用いた従来の原本性保証方法と異なる点である。

【０２０３】

（効果５）上記実施の形態及びその変更例では、上記したような暗号内通信の技術を確立できるので、これによって様々な機能を実現できる。例えば、メッセージ認証符号による原本性保証方法は、公開鍵暗号を用いた署名による方法のように送信者が送信内容を後で否認できない機能を実現できない。しかし、上記実施の形態及びその変更例では、暗号化に鍵Ｋ１（受信者と共有）を使用し、受信者の知らない別の鍵Ｋ２で氏名、日付などを暗号化して、これを検査情報として添付することができる。このとき、鍵Ｋ２を第三者に供託すれば、送信者は後で送信事実と送信内容を否認できなくなる。

【０２０４】

なお、本発明は、上述した実施形態及びその変形例に限定されるものではなく、その要旨を逸脱しない範囲でその他の構成にても具現化することができる。

【産業上の利用可能性】

【０２０５】

本発明は、電子文章の高速な暗号化を可能とすると共に、その原本性を、認証局を介すことなく、当事者間で完結して保証できる原本性保証装置を提供することができる。この原本性保証装置を、ネットワーク上にデータを置くクラウドコンピューティングにて運用すれば、その信頼性を高めることができる。また、この原本性保証装置は、暗号文に証拠能力が保証されたデジタル署名を付与できるので、法律的な証拠として活用できる。また、この原本性保証装置は、暗号文の解読問題がＮＰ完全であることが期待されることから、量子コンピュータが実用化されたとしても原本性の保証が可能となる。

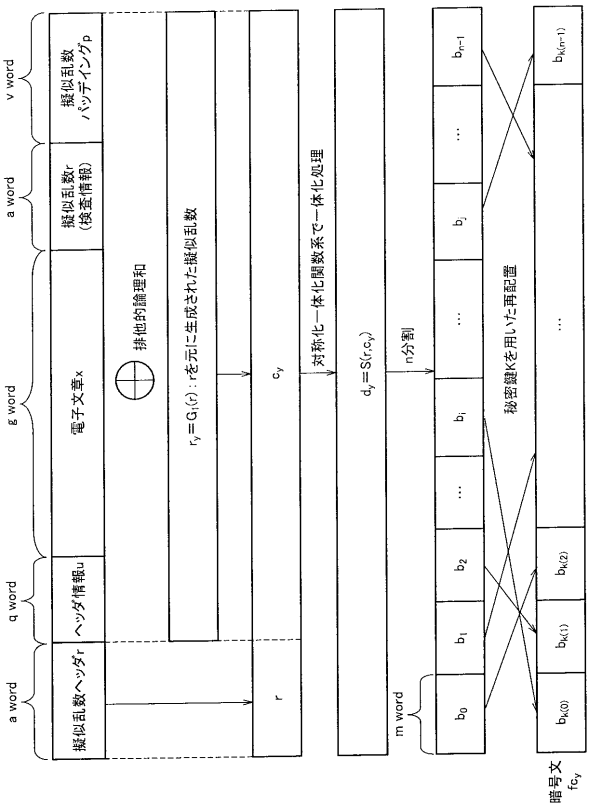
【符号の説明】

【０２０６】

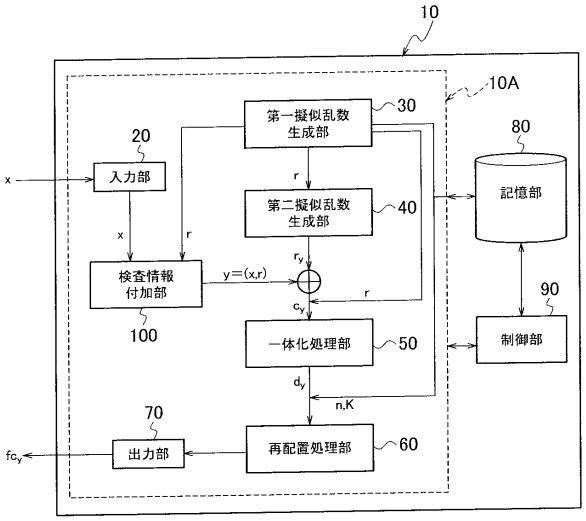
原本性保証装置　１０，２００
入力部　２０，１２０
第一擬似乱数生成部　３０
第二擬似乱数生成部　４０

一体化処理部　５０
再配置処理部　６０
出力部　７０，１７０
記憶部　８０
制御部　９０
検査情報付加部　１００
検査情報検証部　１１０
ブロック暗号文生成部　２４０
ブロック暗号文復号部　３４０
逆一体化処理部　１５０
逆再配置処理部　１６０

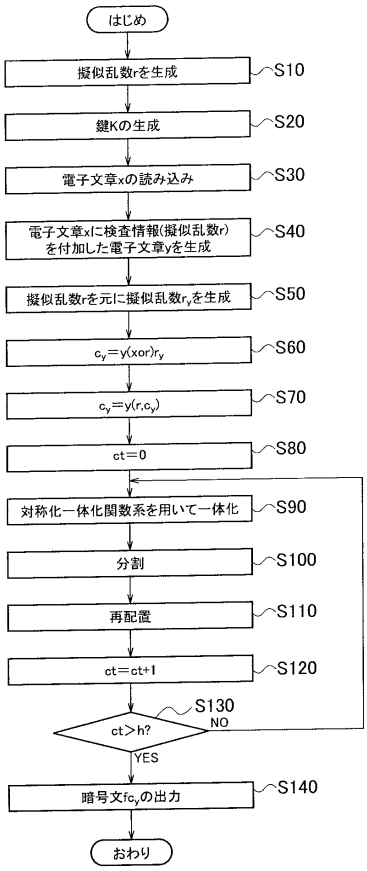
【図１】



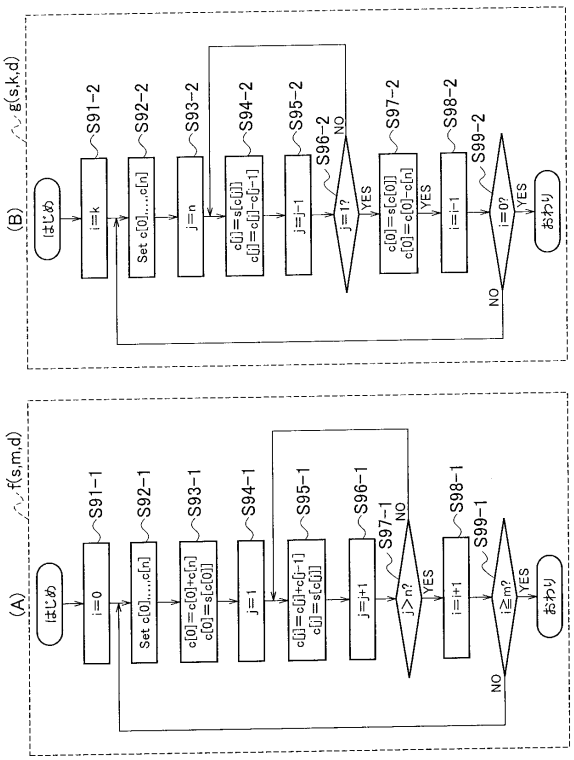
【図２】



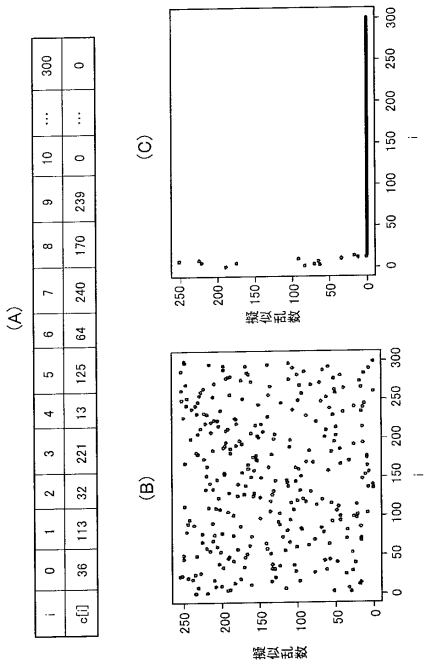
【図３】


はじめ
擬似乱数rを生成
S10
鍵Kの生成
S20
電子文章xの読み込み
S30
電子文章xに検査情報(擬似乱数r)を付加した電子文章yを生成
S40
擬似乱数rを元に擬似乱数r_yを生成
S50
$c_y=y(xor)r_y$
S60
$c_y=y(r,c_y)$
S70
ct=0
S80
対称化一体化関数系を用いて一体化
S90
分割
S100
再配置
S110
ct=ct+1
S120
ct>h?
S130
NO
YES
暗号文fc_yの出力
S140
おわり

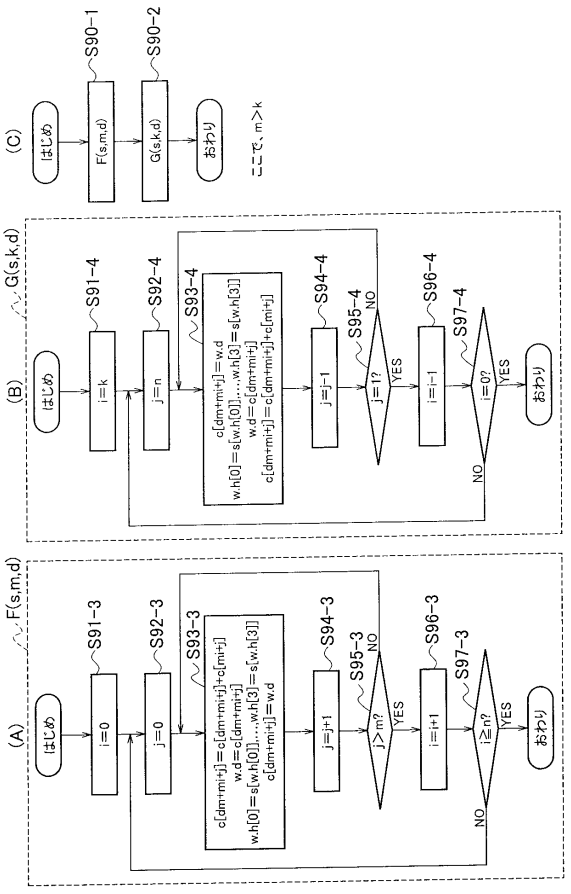
【図４】


g(s,k,d)
(B)
はじめ
i=k
S91-2
Set c[0],...,c[n]
S92-2
j=n
S93-2
c[j]=s[c[j]]
c[j]=c[j]-c[j-1]
S94-2
j=j-1
S95-2
j=1?
S96-2
NO
YES
c[0]=s[c[0]]
c[0]=c[0]-c[n]
S97-2
YES
i=i-1
S98-2
i=0?
S99-2
YES
NO
おわり
f(s,m,d)
(A)
はじめ
i=0
S91-1
Set c[0],...,c[n]
S92-1
c[0]=c[0]+c[n]
c[0]=s[c[0]]
S93-1
j=1
S94-1
c[j]=c[j]+c[j-1]
c[j]=s[c[j]]
S95-1
j=j+1
S96-1
j>n?
S97-1
NO
YES
i=i+1
S98-1
i≧m?
S99-1
YES
NO
おわり

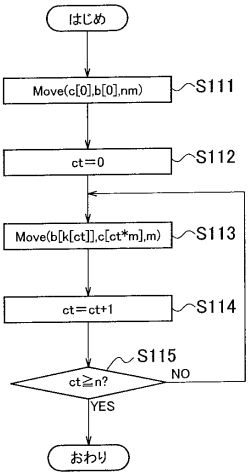
【図５】


(A)
i
c[i]
0 1 2 3 4 5 6 7 8 9 10 ... 300
36 113 32 221 13 125 64 240 170 239 0 ... 0
(C)
(B)
擬似乱数
i
0 50 100 150 200 250 300

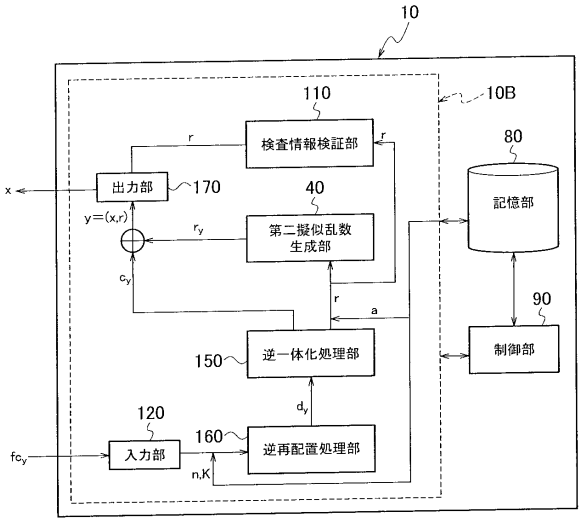
【図６】


(C)
はじめ
F(s,m,d)
S90-1
G(s,k,d)
S90-2
おわり
ここで、m>k
G(s,k,d)
(B)
はじめ
i=k
S91-4
j=n
S92-4
w.h[0]=s[w.h[0]],...,w.h[3]=s[w.h[3]]
w.d=c[dm+mi+j]
c[dm+mi+j]=c[dm+mi+j]+c[mi+j]
S93-4
j=j-1
S94-4
j=1?
S95-4
NO
YES
i=i-1
S96-4
i=0?
S97-4
NO
YES
おわり
F(s,m,d)
(A)
はじめ
i=0
S91-3
j=0
S92-3
c[dm+mi+j]=c[dm+mi+j]+c[mi+j]
w.d=c[dm+mi+j]
w.h[0]=s[w.h[0]],...,w.h[3]=s[w.h[3]]
c[dm+mi+j]=w.d
S93-3
j=j+1
S94-3
j>m?
S95-3
NO
YES
i=i+1
S96-3
i≧n?
S97-3
NO
YES
おわり

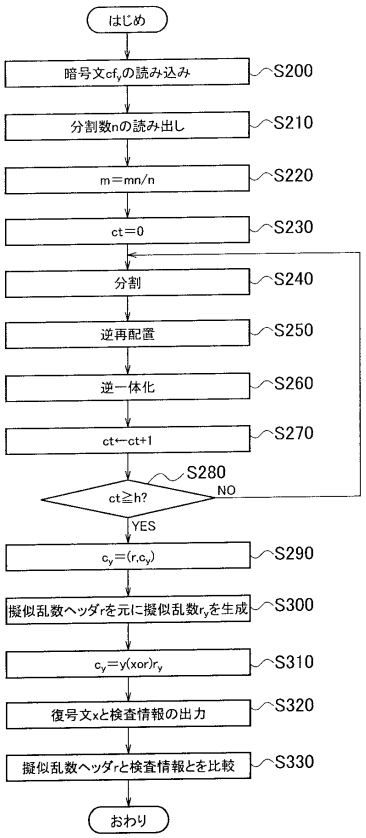
【図７】


はじめ
Move(c[0],b[0],nm)
S111
ct＝0
S112
Move(b[k[ct]],c[ct*m],m)
S113
ct＝ct+1
S114
S115
ct≧n?
NO
YES
おわり

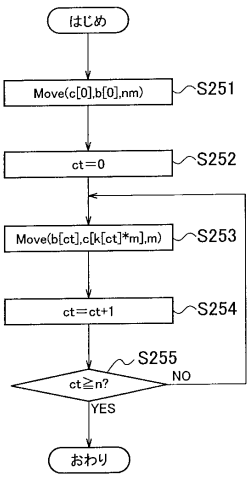
【図８】


10
10B
110
検査情報検証部
r
80
記憶部
x
出力部
170
40
第二擬似乱数
生成部
y＝(x,r)
r_y
c_y
r
a
90
制御部
150
逆一体化処理部
d_y
120
160
逆再配置処理部
fc_y
入力部
n,K

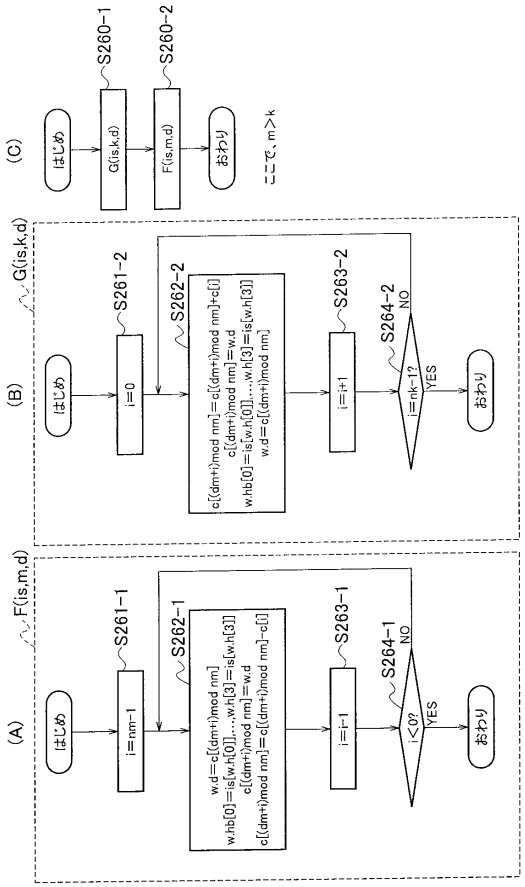
【図９】


はじめ
暗号文cf_yの読み込み
S200
分割数nの読み出し
S210
m＝mn/n
S220
ct＝0
S230
分割
S240
逆再配置
S250
逆一体化
S260
ct←ct+1
S270
S280
ct≧h?
NO
YES
c_y＝(r,c_y)
S290
擬似乱数ヘッダrを元に擬似乱数r_yを生成
S300
c_y＝y(xor)r_y
S310
復号文xと検査情報の出力
S320
擬似乱数ヘッダrと検査情報とを比較
S330
おわり

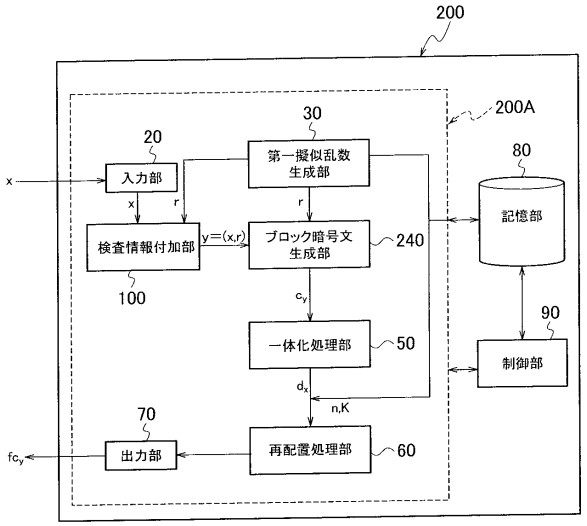
【図１０】


はじめ
Move(c[0],b[0],nm)
S251
ct＝0
S252
Move(b[ct],c[k[ct]*m],m)
S253
ct＝ct+1
S254
S255
ct≧n?
NO
YES
おわり

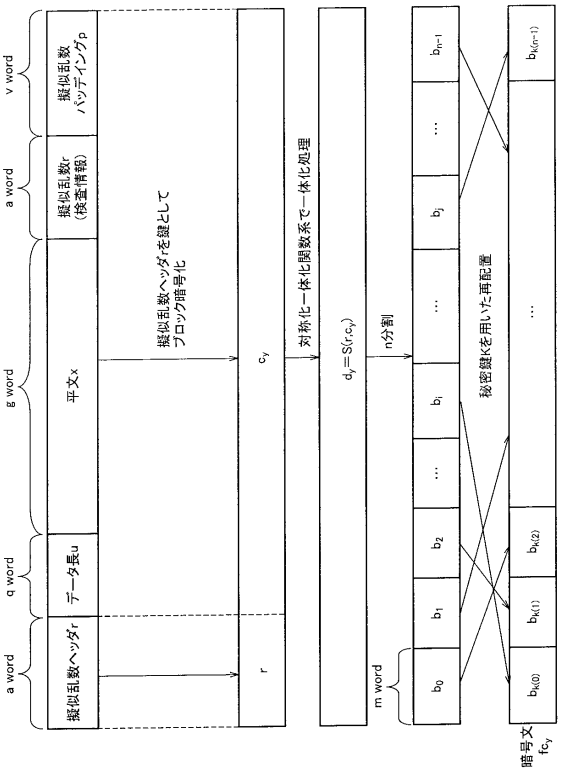
【図１１】



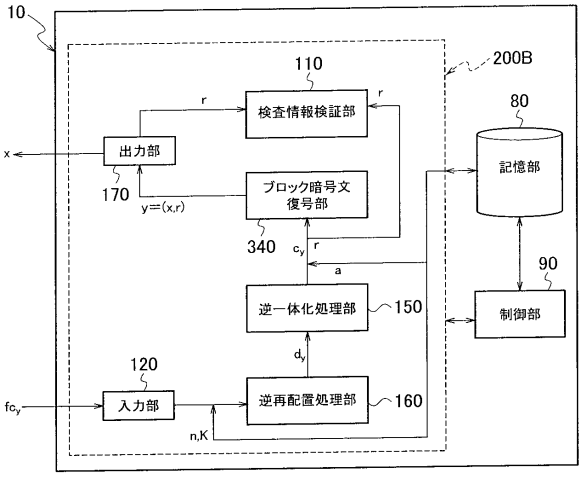
【図１２】



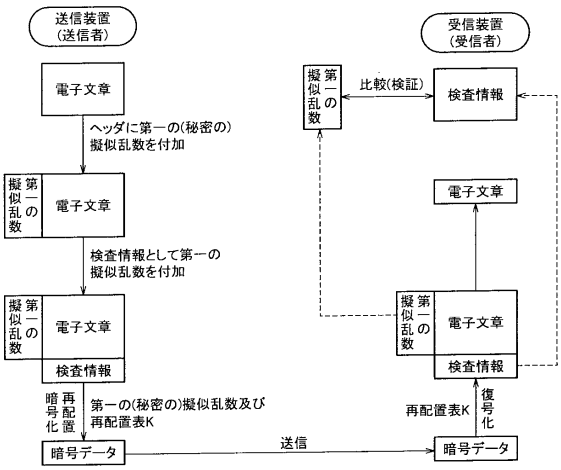
【図１３】



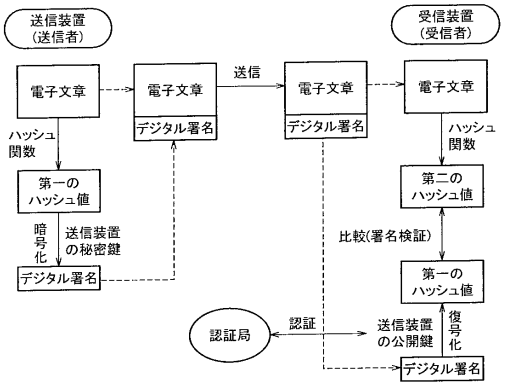
【図１４】



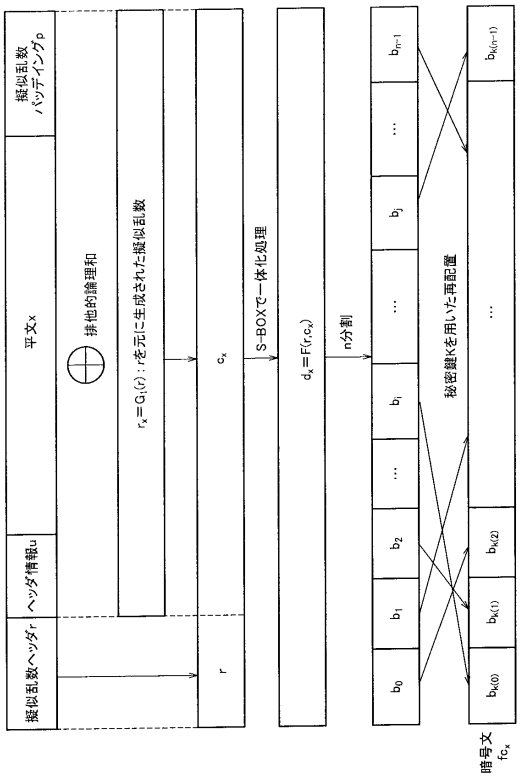
【図１５】


送信装置
(送信者)
受信装置
(受信者)
電子文章
ヘッダに第一の(秘密の)
擬似乱数を付加
第一の
擬似乱数
電子文章
検査情報として第一の
擬似乱数を付加
検査情報
暗号化
再配置
第一の(秘密の)擬似乱数及び
再配置表K
暗号データ
送信
比較(検証)
検査情報
電子文章
再配置表K
復号化
暗号データ

【図１６】


送信装置
(送信者)
受信装置
(受信者)
電子文章
電子文章
デジタル署名
送信
電子文章
デジタル署名
電子文章
ハッシュ
関数
ハッシュ
関数
第一の
ハッシュ値
第二の
ハッシュ値
暗号化
送信装置
の秘密鍵
比較(署名検証)
デジタル署名
第一の
ハッシュ値
認証局
認証
送信装置
の公開鍵
復号化
デジタル署名

【図１７】


擬似乱数
パッディングp
平文x
ヘッダ情報u
擬似乱数ヘッダr
排他的論理和
$r_x = G_1(r)$: rを元に生成された擬似乱数
c_x
r
S-BOXで一体化処理
$d_x = F(r, c_x)$
n分割
b_{n-1}
b_j
b_i
b_2
b_1
b_0
秘密鍵Kを用いた再配置
$b_{K(n-1)}$
$b_{K(2)}$
$b_{K(1)}$
$b_{K(0)}$
暗号文
rc_x

フロントページの続き

(72)発明者　鈴木　秀一
東京都千代田区神田錦町２丁目２番地　学校法人東京電機大学内

Ｆターム(参考) 5B017 AA08 CA16
5J104 AA08 EA02 EA13 LA03 LA06 NA02 NA09 NA12 NA27 NA38
PA14