

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5489115号
(P5489115)

(45) 発行日 平成26年5月14日(2014.5.14)

(24) 登録日 平成26年3月7日(2014.3.7)

(51) Int. Cl. F I
HO4L 9/32 (2006.01) HO4L 9/00 675B
GO6F 21/64 (2013.01) GO6F 21/24 167A

請求項の数 5 (全 38 頁)

(21) 出願番号	特願2009-263838 (P2009-263838)	(73) 特許権者	800000068
(22) 出願日	平成21年11月19日(2009.11.19)		学校法人東京電機大学
(65) 公開番号	特開2011-109510 (P2011-109510A)		東京都足立区千住旭町5番
(43) 公開日	平成23年6月2日(2011.6.2)	(74) 代理人	100083806
審査請求日	平成24年10月23日(2012.10.23)		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) 【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

10

20

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成する関数系であることを特徴とする原本性保証装置。

10

【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、

20

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、

30

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成する関数系であることを特徴とする原本性保証装置。

40

【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、

50

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、

10

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、

20

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成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、

30

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、

40

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

50

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成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。

10

【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 P K I (Public Key Infrastructure) である。P K I 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 M D 5 や S H A 1 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。

20

【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を用いられているが、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。

30

40

【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、P K I 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、P K I そのものの信頼性が揺らいでしまう。

【0007】

50

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 $S - B O X$ (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 ($i = 0, 1, \dots, n - 1$) 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 ($i = 0, 1, \dots, n - 1$) 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 $f c_x$ 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、 $0, 1, \dots, n - 1$ 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 G_0 が生成する第二の擬似乱数列 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. Lindell, 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

10

20

30

40

50

符号 M A C (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ふ。

【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。

10

【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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成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。

20

30

【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

40

50

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成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。

10

【 0 0 1 5 】

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、前記第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成する関数系であることを特徴とする原本性

20

30

保証プログラムである。

【 0 0 1 6 】

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

40

50

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 - B O X の関数を用いて構成される第一の非線形関数系と、前記第一の非線形関数系の逆変換として形を有する第二の非線形関数系とを直列に結合した関数系であり、第三の電子データを入力とする前記第一の非線形関数系からの出力を、前記第二の非線形関数系の入力として前記第四の電子データを生成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。

10

【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、認証局を介することなく、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。

20

【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。

30

【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。

40

【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

50

な構成を示したブロック図である。

【図 1 3】図 1 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概念図である。

【図 1 4】図 1 2 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 1 5】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 1 6】PKIにおける原本性保証方法を示す図である。

【図 1 7】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】

10

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、

20

(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 - B O X 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 ($i = 0, 1, \dots, n - 1$) 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 ($i = 0, 1, \dots, n - 1$) 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 $f c_y$ として生成するステップと、

を備える。

【0021】

30

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、 $0, 1, \dots, n - 1$ 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、 $K = (k[0], k[1], \dots, k[n - 1])$ 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列 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 と第三の擬似乱数列 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、第三の擬似乱数列 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。

40

【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

50

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。

10

【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。

20

【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。

30

【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。

40

【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 f_{c_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 f_{c_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】

50

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。

10

【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。

20

【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。

30

【0040】

送信者により入力部 20 から電子文章 (平文) \times (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。

40

【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
```

50

```
for i:=0 to a-1 do
  r[i]:=G0;
```

次に、ステップ S 2 0 において、制御部 9 0 は、記憶部 8 0 から予め格納された分割数 n を読み出し、第一疑似乱数生成部 3 0 に対して、 $0, 1, \dots, n-1$ 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 $K = (k[0], k[1], \dots, k[n-1])$ として記憶部 8 0 に記憶させる。

【 0 0 4 4 】

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、第一の疑似乱数列 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、第一の疑似乱数列 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。

10

【 0 0 4 5 】

< 再配置表 K の生成アルゴリズムの実施例 >

ステップ S 2 0 における第一疑似乱数生成部 3 0 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。

【 0 0 4 6 】

```
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;
```

20

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、第一疑似乱数生成部 3 0 は、毎回1ワードの疑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。

30

【 0 0 4 7 】

また、第一疑似乱数生成部 3 0 において、ある周期の疑似乱数からより長い周期の疑似乱数を生成することも考えられる。例えば、第一疑似乱数生成部 3 0 において、8バイトの疑似乱数列から256バイトの第二の疑似乱数列（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。

【 0 0 4 8 】

< 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
```

40

50

```

    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;
    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ラウンド攪拌して再配置表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
,238,214,167,6,71,247,232,112,221,148,73,58,201,207,2,146,55,90,102,162,33,103,1
09,39,85,13,105,63,189,11,178,215,220,255,181,0,23,37,114,171,202,96,72,164,188,
62,223,7,51,27,144,147,16,21,203,163,101,175,192,46,48,18,108,126,229,43,230,160
,118,117,15,234,154,155,111,231,219,9,227,132,53,110,121,136,107,65,64,47,239,21
6,128,198,76,183,68,29,141,56,69,125,50,142,138,209,70,99,211,81,150,42,35,185,2
24,225,222,240,104,5,139,93,179,1,129,83,19,248,115,67,36,242,12,174,123,236,54,
151,120,60,24,182,84,38,254,208,86,116,82,244,41,217,165,161,122,75,20,190,26,17
3,195,88,187,172,210,87,145,32,97,124,119,137,196,204,205,79,135,134,191,127,133
,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ですが、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ラウンド以上攪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 1 6 2 5 6 程度の周期になり、1 6 バイトの場合は 4 1 9 4 1 7 6 程度の周期になる。この周期は、 $\text{mod } 2$ の多項式の因数分解のされ方によって多少変化する。

【 0 0 5 3 】

次に、ステップ S 3 0 において、制御部 9 0 は、送信者により入力部 2 0 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 8 0 に記憶させる。

【 0 0 5 4 】

次に、ステップ S 4 0 において、制御部 9 0 は、第一の擬似乱数列 r を記憶部 8 0 から読み出し、検査情報付加部 1 0 0 に対して、入力部 2 0 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 8 0 に記憶させる。

【 0 0 5 5 】

次に、ステップ S 5 0 において、制御部 9 0 は、第一の擬似乱数列 r を記憶部 8 0 から読み出し、第二擬似乱数生成部 4 0 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 $r_y = (r_0, r_1, \dots, r_{nm-a-1})$ を生成させ、これを記憶部 8 0 に記憶させる。

【 0 0 5 6 】

なお、このステップにおいて、入力部 2 0 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 9 0 は、 $v = 2a - q - g \pmod{n}$ 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 3 0 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 9 0 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 $y = (u, x, z) = (x_0, x_1, \dots, x_{nm-a-1})$ として記憶部 8 0 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。

【 0 0 5 7 】

次に、ステップ S 6 0 において、制御部 9 0 は、記憶部 8 0 から第一の電子データ $y = (x_0, x_1, \dots, x_{nm-a-1})$ と第三の擬似乱数列 $r_y = (r_0, r_1, \dots, r_{nm-a-1})$ とを読み出し、両者の排他的論理和をとる ($c_i = x_i \text{ XOR } r_i$ ($i = 0, 1, \dots, nm - a - 1$)) ことにより第二の電子データ $c_y = (c_0, c_1, \dots, c_{nm-a-1})$ を生成し、これを記憶部 8 0 に記憶させる。

【 0 0 5 8 】

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 9 0 は、ステップ S 5 0 において第二の擬似乱数生成部 4 0 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 S 6 0 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。

【 0 0 5 9 】

次に、ステップ S 7 0 において、制御部 9 0 は、記憶部 8 0 から第一の擬似乱数列 r と第二の電子データ $c_y = (c_0, c_1, \dots, c_{nm-a-1})$ 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 $(r, c_y) = (c_0, c_1, \dots, c_{nm-1})$ を生成させ、これを改めて $c_y = (r, c_x$

10

20

30

40

50

) = ($c_0, c_1, \dots, c_{nm-1}$) として記憶部 80 に記憶させる。

【 0060 】

次に、ステップ S 80 において、制御部 90 は、ステップ S 90 ~ S 120 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 S 90 へ処理を進める。

【 0061 】

ステップ S 90 において、制御部 90 は、記憶部 80 から第三の電子データ $c_y = (c_0, c_1, \dots, c_{nm-1})$ 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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ をバイト単位で変換して一体化して第四の電子データ $d_y = S (r, c_y) = (d_0, d_1, \dots, d_{nm-1})$ 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 - B O X 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 S 91 - 1 ~ S 99 - 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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ に対して、 $c_{(dm+i) \bmod nm} = c_{(dm+i) \bmod nm} + c_{i \bmod nm} (i = 0, 1, \dots, nm-1)$ として加算した値 (左辺の $c_{(dm+i) \bmod nm}$) を S - B O X 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 - B O X 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 - B O X 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 S 91 - 2 ~ S 99 - 2 に示すように、 $g (i s, k, d)$ と書くことができる。ここで、i s は S - B O X 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 $F (i s, m, d)$ 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 $g (i s, m, d)$ と図 11 (A) の $F (i s, m, d)$ とが対応する。

【 0066 】

ここで、 $f (s, m, d)$ と $g (i s, m, d)$ 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 - B O X の具体例を表している。この例では、301 個のデータ配列 $c [0], c [1], \dots, c [300]$ 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 $c [0], c [1], \dots, c [300]$ 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

10

20

30

40

50

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。

10

【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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ をバイト単位で変換して一体化して第四の電子データ $d_y = S(r, c_y) = (d_0, d_1, \dots, d_{nm-1})$ 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。

20

【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 $P \leq e^{-2} / 2^m$ を満たす

30

【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 $S_n = \{1, 2, \dots, n\}$ 上の対称群とし、 $F(\cdot) = \{x \in S_n \mid x = (x)\}$ ($\subset S_n$) とする。ここで、自明度 $T: S_n \rightarrow \mathbb{N}$ を $T(\cdot) = \#F(\cdot)$ ($\subset S_n$) 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 $1 - 1/m! < P(T < m) < 1 - 1/(e \cdot m!)$ が成り立つ。

40

【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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ を読み出

50

し、一体化処理部 50 に対して、図 6 (A) のステップ S 9 1 - 3 ~ S 9 7 - 3 に示すように、 $c_{(d m+i) \bmod n m} = c_{(d m+i) \bmod n m} + c_{i \bmod n m}$ ($i = 0, 1, \dots, n m - 1$) として加算した値 (左辺の $c_{(d m+i) \bmod n m}$) を S - B O X 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 S 9 0 - 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 S 9 1 - 4 ~ S 9 1 7 - 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 - B O X を用いてバイト単位で変換して、 $c_{(d m+i) \bmod n m} = c_{(d m+i) \bmod n m} - c_{i \bmod n m}$ ($i = 0, 1, \dots, n m - 1$) 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 S 9 0 - 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 $d_x = S (r, c_x) = (d_0, d_1, \dots, d_{n m - 1})$ を記憶部 80 に記憶させる。

10

【 0 0 7 6 】

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。

【 0 0 7 7 】

```
Tunion=record
  case integer of
    1: (d:Word);
    2: (h:array[0..3] of byte);
  end;
w: Tunion;
```

20

ここでは、1ワードを4バイトとして扱っている。

【 0 0 7 8 】

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) で用いた加算の代わりに減算を用いてもよい。また、S - B O X 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 $K = (k [0], k [1], \dots, k [n - 1])$ から次のようにして生成される S - B O X を用いることが好ましい。

【 0 0 7 9 】

< 再配置表 K による S - B O X の生成アルゴリズムの実施例 >

再配置表 $K = (k [0], k [1], \dots, k [n - 1])$ からは様々な方法で S - B O X 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。

30

【 0 0 8 0 】

```
( 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 )
```

40

n - 256 e 0 となる整数 e を一つ決める。

【 0 0 8 1 】

```
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
```

50

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、 $0, 1, \dots, 255$ をランダムに並び替えた $s = (k[0], k[1], \dots, k[255])$ 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 $d_y = (d_0, d_1, \dots, d_{nm-1})$ 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 $b_i = (d_{mi}, d_{mi+1}, \dots, d_{mi+m-1})$ ($i = 0, 1, \dots, n-1$) 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 $d_x = (b_0, b_1, \dots, b_{n-1})$ として記憶部 80 に記憶させる。

10

【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 $d_y = (b_0, b_1, \dots, b_{n-1})$ と再配置表 $K = (k[0], k[1], \dots, k[n-1])$ 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 $d_y = (b_0, b_1, \dots, b_{n-1})$ を、再配置表 $K = (k[0], k[1], \dots, k[n-1])$ に基づき、 $d_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ 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。

20

【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 $d_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ を読み出し、最終的な暗号文 $fc_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ 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 $fc_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ を送信してすべての処理を終了する。

30

【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 $S90 \sim S120$ 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擬似乱数列 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。

40

【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

50

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^{15}
 9 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 $0, 1, \dots, n-1$ 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 - B
 OX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（部分和问题） Z_0 ：決定問題としてのナップサック問題と

10

20

30

40

50

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 ($i = 1, 2, \dots, N$) 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、これを部分和问题といい、以下のように定式化できる。

【0099】

<部分和问题>

与えられた自然数 x_1, x_2, \dots, x_N, y に対し、ある部分集合 $I = \{1, 2, \dots, N\}$ が存在し、 $y = \sum_{i \in I} x_i$ 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。 10

【0100】

<和ジグソーパズル問題>

与えられた自然数 x_1, x_2, \dots, x_N, y に対し、ある置換 (S-BOX) s と自然数 m が存在し、 $y = \sum_{i=1}^m x_{s(i)}$ 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 $X = (x_1, x_2, \dots, x_N)$ 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。 20

【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 (部分和问题) 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 $f(X, y, s, m) = (E_s(X, y, m), (X, \sum_{i=1}^m x_{s(i)}, m), s)$ 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、 $X = (x_1, x_2, \dots, x_N)$ への s の作用を $s(X) = (x_{s(1)}, x_{s(2)}, \dots, x_{s(N)})$ とし、 n を入力サイズ (バイト数) とする ($N = n$)。 30

【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。 40

【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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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。

10

【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。

20

【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 とから切り離れ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。

30

【0113】

送信装置から送信された暗号文 $f c_y = (f_0, f_1, \dots, f_{nm-1})$ 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。

40

【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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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

50

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 $f_{cy} = (f_0, f_1, \dots, f_{nm-1})$ 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 $d_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ として記憶部80に記憶させる。

10

【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 $d_y = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ と秘密鍵 $K = (k[0], k[1], \dots, k[n-1])$ 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 $d_y = (b_0, b_1, \dots, b_{n-1})$ 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 $d_y = (b_0, b_1, \dots, b_{n-1})$ 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、 $c_{(dm+i) \bmod nm} = c_{(dm+i) \bmod nm} + c_{i \bmod nm}$ ($i = 0, 1, \dots, nm-1$)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 $d_y = (d_0, d_1, \dots, d_{nm-1})$ を $c_{(dm+i) \bmod nm} = c_{(dm+i) \bmod nm} - c_{i \bmod nm}$ ($i = nm-1, nm-2, \dots, 1, 0$)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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ 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)。

20

30

【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。

40

【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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ 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 $c_y = (c_0, c_1, \dots, c_{nm-1})$ 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。

50

【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 $r_y = (r_0, r_1, \dots, r_{nm-a-1})$ 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 $c_y = (c_{a+1}, c_{a+2}, \dots, c_{nm-a-1})$ と生成された擬似乱数列 $r_y = (r_0, r_1, \dots, r_{nm-a-1})$ とを排他的論理和する ($x_{a+i} = c_{a+i} \text{ XOR } r_i$ ($i = 0, 1, \dots, nm-a-1$)) ことにより、第一の電子データ $y = (x_a, x_{a+1}, \dots, x_{nm-1})$ 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。

10

【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。

20

【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 $f c_y$ を復号するための情報として、秘密鍵としての再配置表 $K = (k[0], k[1], \dots, k[n-1])$ 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 $f c_y$ 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。

30

【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。

40

【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

50

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。

【 0 1 3 4 】

[受信装置としての原本性保証装置 1 0 の効果]

このように、原本性保証装置 1 0 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 1 0 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。

【 0 1 3 5 】

受信装置としての原本性保証装置 1 0 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 1 5 0 の特徴にある。この特徴は、逆一体化処理部 1 5 0 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。

【 0 1 3 6 】

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^{-8^a} 程度でしかない。例えば、第一の擬似乱数列 r の長さを 1 2 8 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、 $2^{-8} \times 128 = 2^{-1024}$ 程度と限りなく小さい。

【 0 1 3 7 】

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を確認できてしまう。

【 0 1 3 8 】

[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 1 0 A と復号化処理部 1 0 B とを同じ原本性保証装置 1 0 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 1 0 A と復号化処理部 1 0 B とを別体の装置として構成してもよい。

【 0 1 3 9 】

[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 1 2 8 バイトを使用する。第一疑似乱数生成部 3 0 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 2 0 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 $K = (k[0], k[1], \dots, k[255])$ として表される再配置表であり、 $k[i]$ ($i = 0, 1, \dots, 255$) には、 $0, 1, \dots, 255$ を並べ替えた値が格納されている。

【 0 1 4 0 】

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。

【 0 1 4 1 】

疑似乱数列のヘッダ r 、ヘッダ情報 u 、電子文章 x 、検査情報としての疑似乱数列 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 $x = (x_0, x_1, \dots, x_{256m-1})$ とする。ここで、 x_i ($i = 0, 1, \dots, 255$)は、Long Word (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;
```

10

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、検査情報としての疑似乱数列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、Long Wordの配列として、改めて $x = (x[0], x[1], \dots, x[v-1])$ ($v = 256m$)とする。ここで、mは、ブロックデータの長さをLong Wordの個数で表したものである。

20

【0143】

なお、次の実装例では、図1の表記を合わせて、疑似乱数のヘッダ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;
```

40

< 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]];
end;
```

50

```
x[0]:=x[0]-x[128+4+g+256+128+v];
```

```
end;
```

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

10

【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;
```

20

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;
```

30

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。

40

【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。

50

【 0 1 5 1 】

記憶部 8 0 は、入力部 2 0、第一擬似乱数生成部 3 0、一体化処理部 5 0、再配置処理部 6 0、出力部 7 0、検査情報付加部 1 0 0、ブロック暗号生成部 2 4 0 から成る暗号生成部 2 0 0 A 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。

【 0 1 5 2 】

また、制御部 9 0 は、記憶部 8 0 から読み出した暗号化プログラムに従って、入力部 2 0、第一擬似乱数生成部 3 0、一体化処理部 5 0、再配置処理部 6 0、出力部 7 0、検査情報付加部 1 0 0、ブロック暗号生成部 2 4 0、記憶部 8 0 を制御する CPU を備える。

10

【 0 1 5 3 】

なお、記憶部 8 0 及び制御部 9 0 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。

【 0 1 5 4 】

[変形例に係る暗号化処理]

以上を前提として、図 1 2 に示した原本性保証装置 2 0 0 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 1 3 は、図 1 2 に示した原本性保証装置 2 0 0 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。

【 0 1 5 5 】

原本性保証装置 2 0 0 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。

20

【 0 1 5 6 】

従って、ブロック暗号文生成部 2 4 0 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 9 0 は、 $v = 2a - g - q \pmod{n}$ 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 3 0 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。

30

【 0 1 5 7 】

次のステップとして、制御部 9 0 は、記憶部 8 0 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 2 4 0 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。

【 0 1 5 8 】

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。

40

【 0 1 5 9 】

[変更例に係る送信装置としての原本性保証装置 2 0 0 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のだが、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。

【 0 1 6 0 】

50

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 $256! \cdot 2^{1684}$ 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 $(256!)^{1/2} \cdot 2^{842}$ 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 $512! \cdot 2^{3875}$ 通りとなり、量子コンピュータの計算量は $(512!)^{1/2} \cdot 2^{1938}$ 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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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 $f c_y$ 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とから切り離れた構成としてもよい。いずれの構成におい

10

20

30

40

50

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 $f c_y = (f_0, f_1, \dots, f_{nm-1})$ 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 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 $c_y = (c_{a+1}, c_{a+2}, \dots, c_{nm-a-1})$ を公知のブロック復号化して第一の電子データ $y = (x_a, x_{a+1}, \dots, x_{nm-1})$ 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 $f c_y$ を復号するための情報として、秘密鍵としての再配置表 $K = (k[0], k[1], \dots, k[n-1])$ 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 200A 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 $c f_y$ 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

10

20

30

40

50

る。

【 0 1 7 8 】

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 A E S 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。

【 0 1 7 9 】

[変更例に係る受信装置としての原本性保証装置 2 0 0 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代わりにブロック暗号において用いた。A E S 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。

10

【 0 1 8 0 】

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。

20

【 0 1 8 1 】

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 C B C 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 Z 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、C B C 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ふ。特に、C B C 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。

30

【 0 1 8 2 】

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。

40

【 0 1 8 3 】

[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。

【 0 1 8 4 】

(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。

【 0 1 8 5 】

50

(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

10

20

30

40

50

できる。

【 0 1 9 8 】

[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。

【 0 1 9 9 】

(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。

【 0 2 0 0 】

(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。

【 0 2 0 1 】

(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。

【 0 2 0 2 】

(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。

【 0 2 0 3 】

(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。

【 0 2 0 4 】

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】

【 0 2 0 5 】

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】

【 0 2 0 6 】

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

10

20

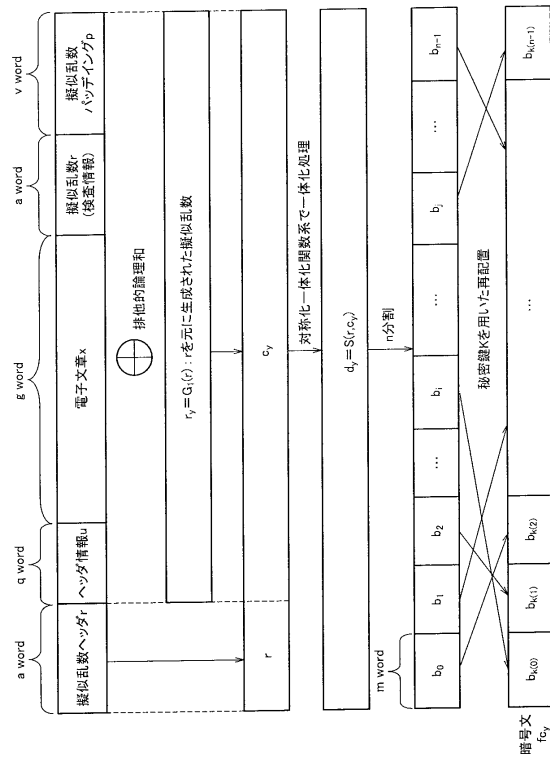
30

40

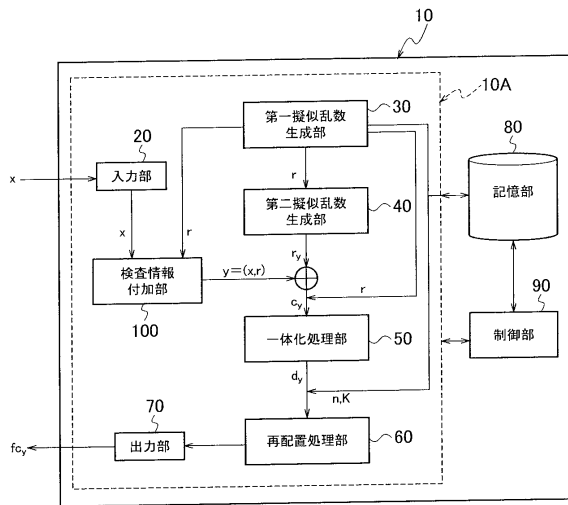
50

- 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

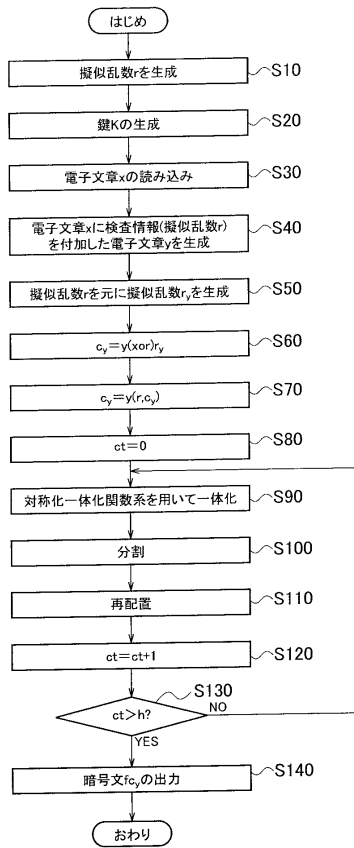
【図1】



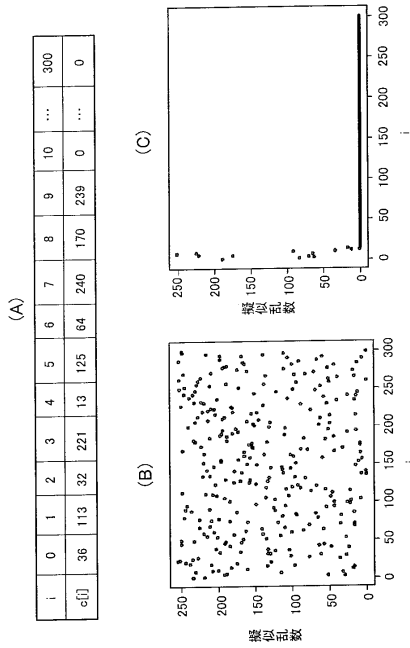
【図2】



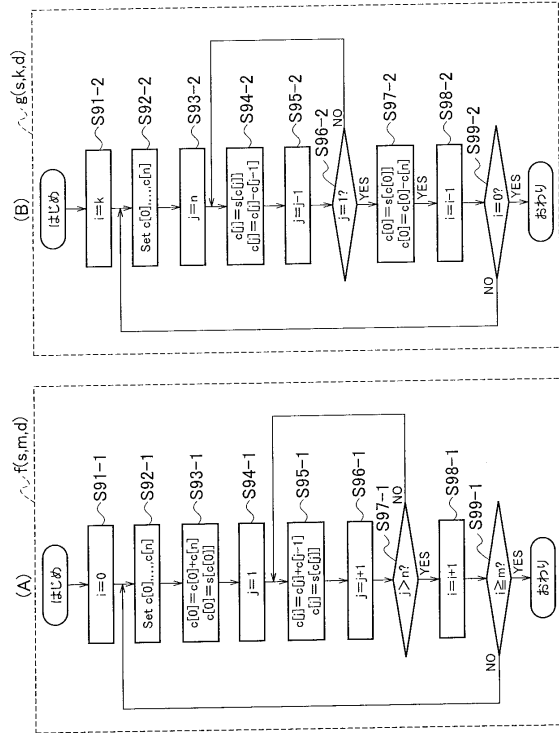
【図3】



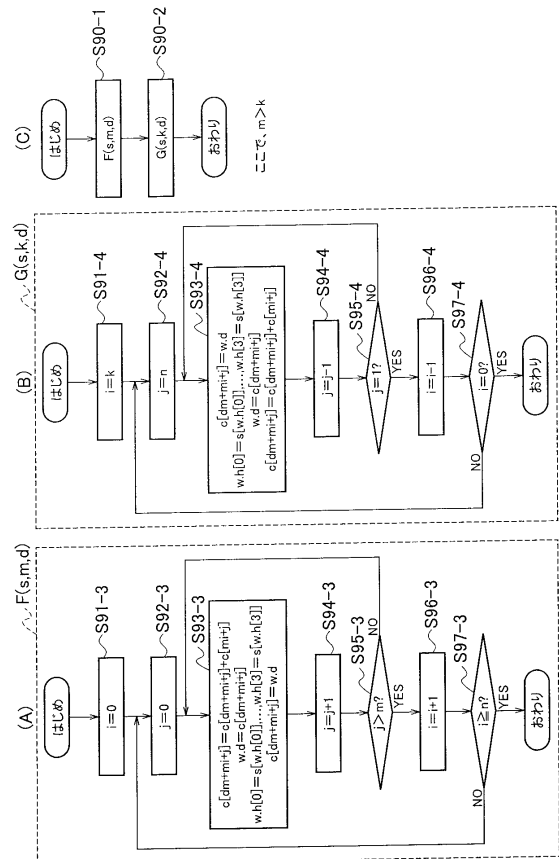
【図5】



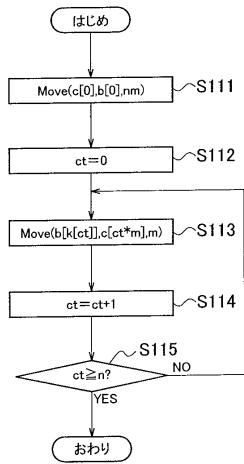
【図4】



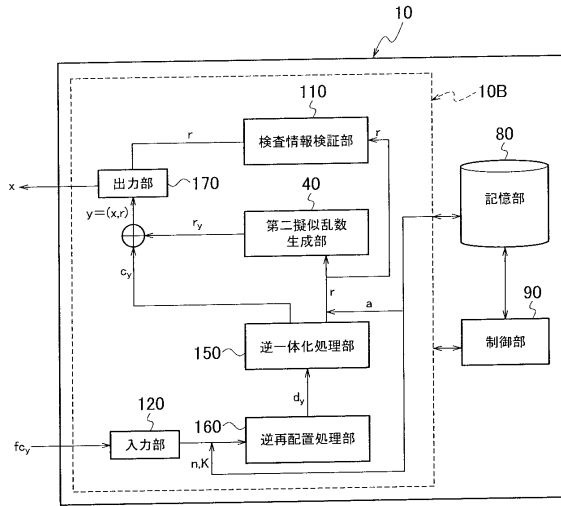
【図6】



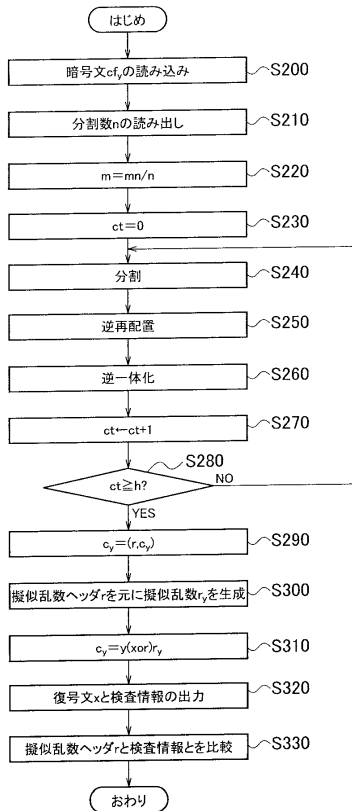
【 図 7 】



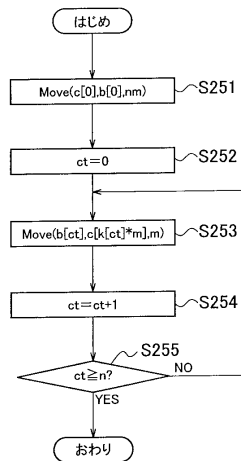
【 図 8 】



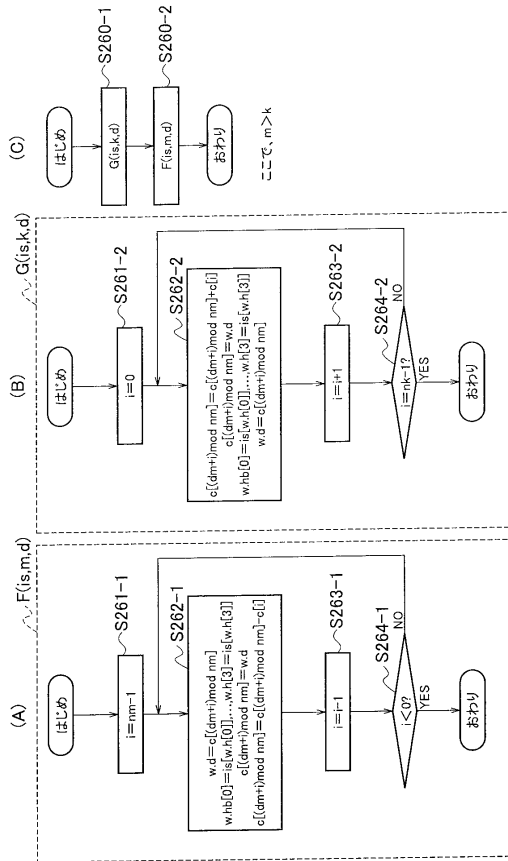
【 図 9 】



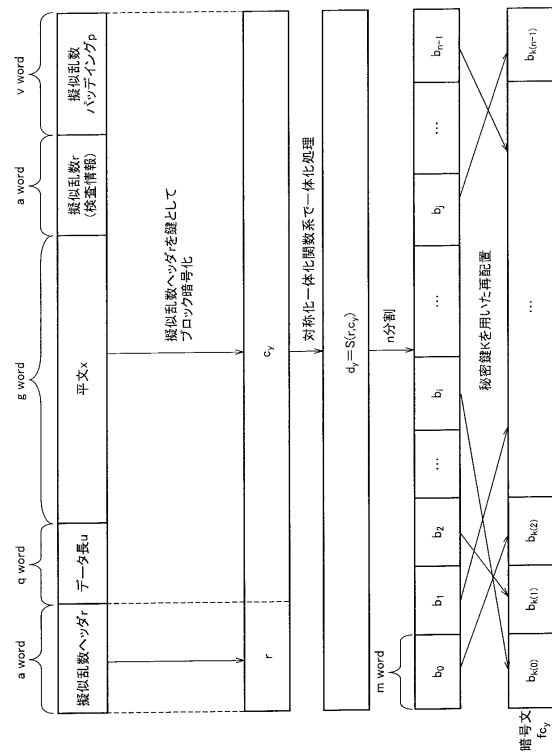
【 図 10 】



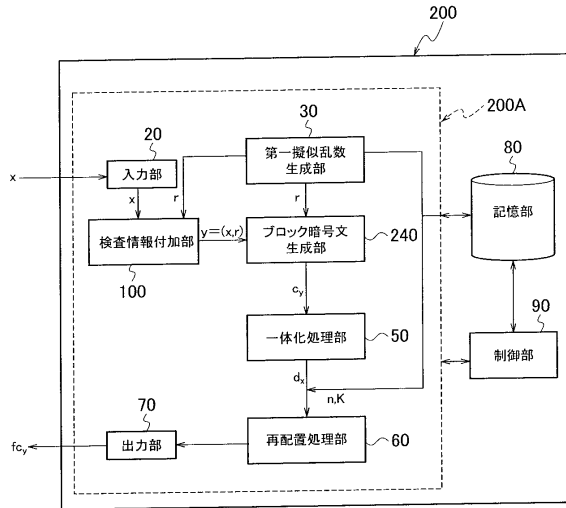
【図 1 1】



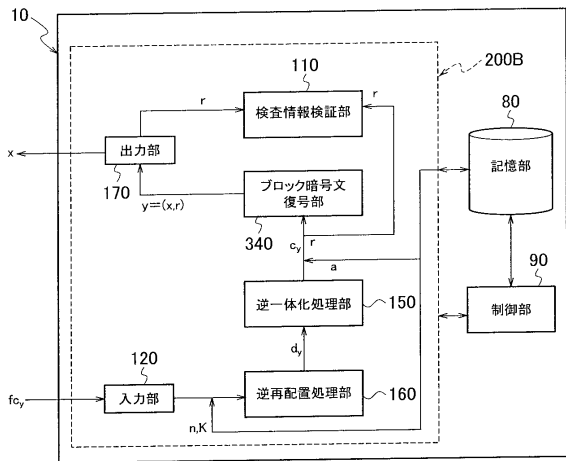
【図 1 3】



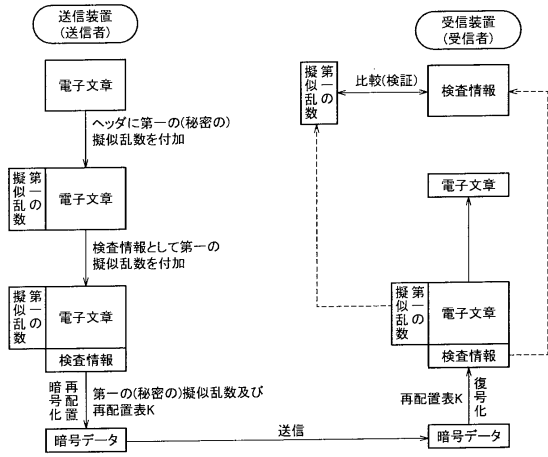
【図 1 2】



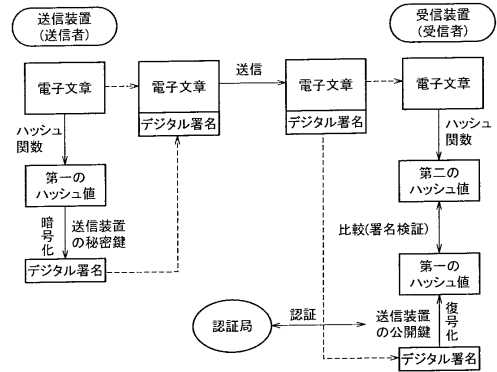
【図 1 4】



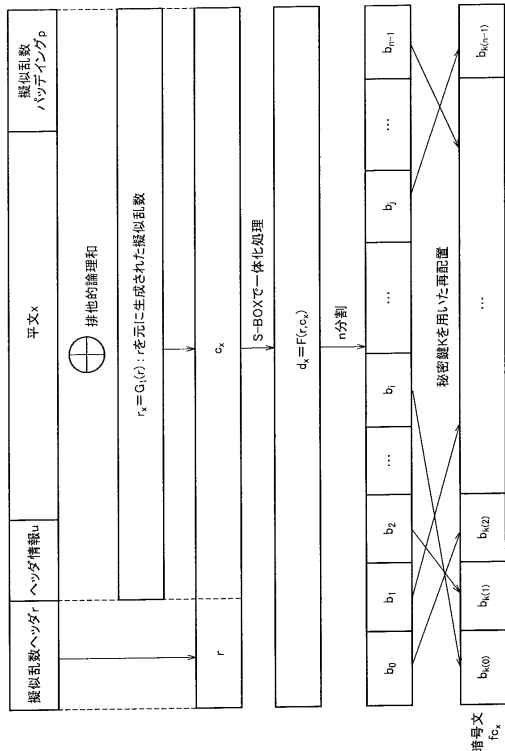
【図15】



【図16】



【図17】



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

審査官 金沢 史明

(56)参考文献 国際公開第2008/114829(WO, A1)

特開2006-339898(JP, A)

(58)調査した分野(Int.Cl., DB名)

H04L 9/32

G09C 1/00