

(19) 日本国特許庁(JP)

再公表特許(A1)

(11) 国際公開番号

W02008/114829

発行日 平成22年7月8日(2010.7.8)

(43) 国際公開日 平成20年9月25日(2008.9.25)

(51) Int.Cl. F I テーマコード(参考)
G09C 1/00 (2006.01) G09C 1/00 610A 5J104

審査請求 未請求 予備審査請求 有 (全 30 頁)

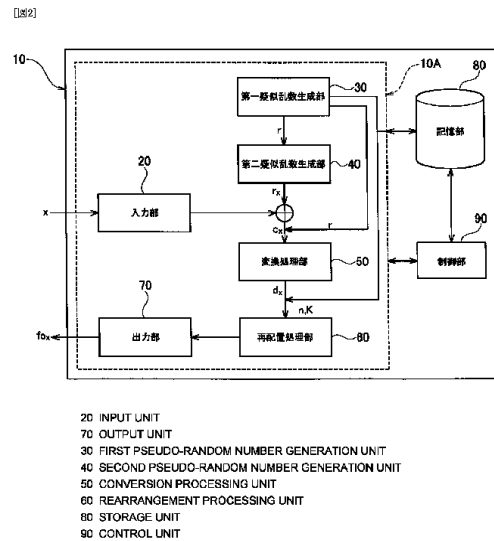
<p>出願番号 特願2009-505246 (P2009-505246)</p> <p>(21) 国際出願番号 PCT/JP2008/055132</p> <p>(22) 国際出願日 平成20年3月19日(2008.3.19)</p> <p>(31) 優先権主張番号 特願2007-71432 (P2007-71432)</p> <p>(32) 優先日 平成19年3月19日(2007.3.19)</p> <p>(33) 優先権主張国 日本国(JP)</p>	<p>(71) 出願人 800000068 学校法人東京電機大学 東京都千代田区神田錦町2-2</p> <p>(74) 代理人 100083806 弁理士 三好 秀和</p> <p>(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</p> <p>Fターム(参考) 5J104 JA03 NA02 NA09 NA37</p>
---	---

最終頁に続く

(54) 【発明の名称】 暗号装置、復号装置、暗号プログラム、復号プログラム、及び記録媒体

(57) 【要約】

平文 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、ヘッダ r を初期値として擬似乱数列 r_x を生成する第二擬似乱数生成部40と、平文 x と擬似乱数列 r_x との排他的論理和をとった結果 c_x とヘッダ r との組 (r, c_x) をS-BOXを用いて変換する変換処理部50と、一体化されたデータ d_x を n 個のブロック $b_i (i = 0, \dots, n - 1)$ に分割し、秘密鍵 K を用いて n 個のブロック $b_i (i = 0, \dots, n - 1)$ を再配置することにより暗号文 $f c_x$ を生成する再配置処理部60と、暗号文 $f c_x$ を出力する出力部70とを備える。



【特許請求の範囲】**【請求項 1】**

平文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、
前記平文と前記第三の擬似乱数列とを排他的論理和した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データを所定の非線形関数を用いて変換する変換処理部と、
前記変換された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データを暗号文として生成する再配置処理部と、
前記生成された暗号文を出力する出力部と、
を備えたことを特徴とする暗号装置。

10

【請求項 2】

請求項 1 に記載の暗号装置において、前記変換処理部は、前記所定の非線形関数として前記再配置表を用いて変換を行うことを特徴とする暗号装置。

【請求項 3】

平文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平文をブロック暗号化するブロック暗号文生成部と、
前記ブロック暗号化された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データを所定の非線形関数を用いて変換する変換処理部と、
前記変換された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データを暗号文として生成する再配置処理部と、
前記生成された暗号文を出力する出力部と、
を備えたことを特徴とする暗号装置。

20

【請求項 4】

請求項 3 に記載の暗号装置において、前記変換処理部は、前記所定の非線形関数として前記再配置表を用いて変換を行うことを特徴とする暗号装置。

30

【請求項 5】

暗号文を入力する入力部と、
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に戻す逆再配置処理部と、
所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第 1 のデータと残りの第 2 のデータと分離する逆変換処理部と、
前記第 1 のデータを初期値として擬似乱数列を生成する公開可能な擬似乱数生成部と、
生成された擬似乱数列と前記第 2 のデータとを排他的論理和したデータを平文として出力する出力部と、
を備えたことを特徴とする復号装置。

40

【請求項 6】

請求項 5 に記載の復号装置において、前記逆変換処理部は、前記所定の非線形関数として前記所定の再配置表を用いて逆変換を行うことを特徴とする復号装置。

【請求項 7】

暗号文を入力する入力部と、
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に戻す逆再配置処理部と、
所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先

50

頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理部と、

擬似乱数列を生成する秘密の擬似乱数生成部と、

生成された擬似乱数列を鍵として用いることにより前記第2のデータをブロック復号化したデータを平文として出力する出力部と、

を備えたことを特徴とする復号装置。

【請求項8】

請求項7に記載の復号装置において、前記逆変換処理部は、前記所定の非線形関数として前記所定の再配置表を用いて逆変換を行うことを特徴とする復号装置。

【請求項9】

コンピュータを、

平文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、

前記平文と前記第三の擬似乱数列とを排他的論理和した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データを所定の非線形関数を用いて変換する変換処理手段と、

前記変換された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データを暗号文として生成する再配置処理手段と、

前記生成された暗号文を出力する出力手段と、

して機能させることを特徴とする暗号プログラム。

【請求項10】

請求項9に記載の暗号プログラムにおいて、前記変換処理手段は、前記所定の非線形関数として前記再配置表を用いて変換を行うことを特徴とする暗号プログラム。

【請求項11】

コンピュータを、

平文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手段と、

前記第一の擬似乱数列を鍵として前記平文をブロック暗号化するブロック暗号文生成手段と、

前記ブロック暗号化された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データを所定の非線形関数を用いて変換する変換処理手段と、

前記変換された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データを暗号文として生成する再配置処理手段と、

前記生成された暗号文を出力する出力手段と、

して機能させることを特徴とする暗号プログラム。

【請求項12】

請求項11に記載の暗号プログラムにおいて、前記変換処理手段は、前記所定の非線形関数として前記再配置表を用いて変換を行うことを特徴とする暗号プログラム。

【請求項13】

コンピュータを、

暗号文を入力する入力手段と、

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に戻す逆再配置処理手段と、

所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理手段と、

前記第1のデータを初期値として擬似乱数列を生成する公開可能な擬似乱数生成手段と

10

20

30

40

50

生成された擬似乱数列と前記第2のデータとを排他的論理和したデータを平文として出力する出力手段と、

して機能させることを特徴とする復号プログラム。

【請求項14】

請求項13に記載の復号プログラムにおいて、前記逆変換処理部は、前記所定の非線形関数として前記所定の再配置表を用いて逆変換を行うことを特徴とする復号プログラム。

【請求項15】

コンピュータを、

暗号文を入力する入力手段と、

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に戻す逆再配置処理手段と、

所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理手段と、

擬似乱数列を生成する秘密の擬似乱数生成手段と、

生成された擬似乱数列を鍵として用いることにより前記第2のデータをブロック復号化したデータを平文として出力する出力手段と、

して機能させることを特徴とする復号プログラム。

【請求項16】

請求項15に記載の復号プログラムにおいて、前記逆変換処理部は、前記所定の非線形関数として前記所定の再配置表を用いて逆変換を行うことを特徴とする復号プログラム。

【請求項17】

請求項9乃至16のいずれか1項に記載の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、既知平文攻撃を排除可能とする暗号装置及び暗号プログラム、これらの暗号装置及び暗号プログラムによって作成された暗号文を復号する復号装置及び復号プログラム、及びこれらのプログラムを記録する記録媒体に関する。

【背景技術】

【0002】

従来使用されてきた暗号方式（共通鍵暗号方式：ブロック暗号、ストリーム暗号）は、既知平文攻撃を許容する（非特許文献1）。既知平文攻撃とは、平文とその暗号文との組を多数用いて暗号鍵を特定する攻撃のことであり、線形攻撃や差分攻撃などがその代表的な手法である。DES（Data Encryption Standard）やAES（Advanced Encryption Standard）などの標準的なブロック暗号方式では、鍵スケジュールと10ラウンド以上のデータ攪拌とによって、これらの攻撃に対抗してきた。

【非特許文献1】“Cryptography: Theory and practice, 3rd-Ed.”, Stinson, D.R., Chapman & Hall/CRC Press Inc. 2006.

【発明の開示】

【0003】

上記の手法においてセキュリティレベルを引き上げるためには、暗号鍵の長さを長くしたり、データ攪拌処理において使用されるS-BOXの非線形性を強くしたりする必要がある。しかし、その分、暗号化の速度が低下し、共通鍵暗号方式の従来利点を損なう。また、既知平文攻撃に対して強い特別なS-BOXを設計する必要があり、そのための研究開発費や計算機資源も増大することになる。

【0004】

本発明は、上記の実情を鑑みて為されたものであり、既知平文攻撃を排除可能とすると

10

20

30

40

50

共に、高速度で暗号化を行うことのできる暗号装置及び暗号プログラム、これらの暗号装置及び暗号プログラムによって作成された暗号文を復号化する復号装置及び復号プログラム、及びこれらのプログラムを記憶する記憶媒体を提供することを目的とする。

【0005】

本発明の第1の側面によれば、平文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、前記平文と前記第三の擬似乱数列とを排他的論理和した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データを所定の非線形関数を用いて変換する変換処理部と、前記変換された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データを暗号文として生成する再配置処理部と、前記生成された暗号文を出力する出力部とを備えたことを特徴とする暗号装置が提供される。

10

【0006】

本発明の第2の側面によれば、平文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平文をブロック暗号化するブロック暗号文生成部と、前記ブロック暗号化された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データを所定の非線形関数を用いて変換する変換処理部と、前記変換された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データを暗号文として生成する再配置処理部と、前記生成された暗号文を出力する出力部とを備えたことを特徴とする暗号装置が提供される。

20

【0007】

本発明の第3の側面によれば、暗号文を入力する入力部と、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に戻す逆再配置処理部と、所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理部と、

前記第1のデータを初期値として擬似乱数列を生成する公開可能な擬似乱数生成部と、生成された擬似乱数列と前記第2のデータとを排他的論理和したデータを平文として出力する出力部とを備えたことを特徴とする復号装置が提供される。

30

【0008】

本発明の第4の側面によれば、暗号文を入力する入力部と、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に戻す逆再配置処理部と、所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理部と、擬似乱数列を生成する秘密の擬似乱数生成部と、生成された擬似乱数列を鍵として用いることにより前記第2のデータをブロック復号化したデータを平文として出力する出力部とを備えたことを特徴とする復号装置が提供される。

40

【0009】

本発明の第5の側面によれば、コンピュータを、平文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、前記平文と前記第三の擬似乱数列とを排他的論理和した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データを所定の非線形関数を用い変換する変換処理手段と、前記変換された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データを暗号文として生成する再配置処理手段と、前記生成された暗号文を出力する出力手段として機能させることを特徴とする暗号プログラムが提供される。

【0010】

50

本発明の第6の側面によれば、平文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平文をブロック暗号化するブロック暗号文生成手段と、前記ブロック暗号化された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データを所定の非線形関数を用いて変換する変換処理手段と、前記変換された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データを暗号文として生成する再配置処理手段と、前記生成された暗号文を出力する出力手段として機能させることを特徴とする暗号プログラムが提供される。

【0011】

本発明の第7の側面によれば、コンピュータを、暗号文を入力する入力手段と、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に戻す逆再配置処理手段と、所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理手段と、前記第1のデータを初期値として擬似乱数列を生成する公開可能な擬似乱数生成手段と、生成された擬似乱数列と前記第2のデータとを排他的論理和したデータを平文として出力する出力手段として機能させることを特徴とする復号プログラムが提供される。

【0012】

本発明の第8の側面によれば、コンピュータを、暗号文を入力する入力手段と、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に戻す逆再配置処理手段と、所定の非線形関数を用いて逆再配置されたデータを逆変換し、逆変換されたデータを先頭から所定のデータ長を有する第1のデータと残りの第2のデータと分離する逆変換処理手段と、擬似乱数列を生成する秘密の擬似乱数生成手段と、生成された擬似乱数列を鍵として用いることにより前記第2のデータをブロック復号化したデータを平文として出力する出力手段として機能させることを特徴とする復号プログラムが提供される。

【0013】

本発明の第9の側面によれば、上記のプログラムのうちの少なくとも一つが記録されたコンピュータが読み取り可能な記録媒体が提供される。

【図面の簡単な説明】

【0014】

【図1】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おいて示した概念図である。

【図2】図2は、本発明に係る暗号装置の概略的な構成を示したブロック図である。

【図3】図3は、本発明に係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】図4は、図3に示した暗号化処理における変換処理の具体的な手順を示したフローチャートである。

【図5】図5は、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6】図6は、図4に示した変換処理の一例を示した図である。

【図7】図7は、本発明に係る復号装置の概略的な構成を示したブロック図である。

【図8】図8は、本発明に係る復号化処理の手順を示したフローチャートである。

【図9】図9は、図8に示した復号化処理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10】図10は、図8に示した復号化処理における逆変換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11は、本発明のその他の暗号装置の概略的な構成を示したブロック図である。

【図12】図12は、図11に示した暗号装置における暗号化の手順を示した概念図であ

10

20

30

40

50

る。

【発明を実施するための最良の形態】

【0015】

図1は、本発明に係る暗号方式の特徴をストリーム暗号を用いて示した概念図である。

図1に示すように、本発明の暗号方式は、

(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 で擬似乱数列 r_x を生成するステップと、

(3) 平文 x と擬似乱数列 r_x との排他的論理和をとり平文 x を暗号化するステップと

、

(4) 第一の擬似乱数列 r をヘッダとして暗号文 c_x に付加したデータ (r, c_x) を $S-BOX$ (Substitution Box) などを用いて変換 (一体化) 処理するステップと、

(5) 変換されたデータ 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、

(6) 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ることにより 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データを最終的な暗号文 $f c_x$ として生成するステップと、

を備える。

【0016】

さらに、この暗号方式では、必要に応じて、ステップ(4)~(6)における変換処理から再配置処理までの処理を数回繰り返すステップを含めてもよい。

【0017】

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。

【0018】

また、上記(3)では、平文 x と疑似乱数 r_x 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、疑似乱数 r_x との排他的論理和をとる対象は、平文 x と、平文 x のデータ長などの情報を含んだヘッダ情報と、必要に応じてパディング p とを合わせたものである。以降の実施例においては、ヘッダ情報 u と平文 x 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平文 x と見なす。

【0019】

本発明においては、この暗号方式を“再配置暗号”と称する。

【0020】

以下に、本発明の実施形態を、図面を用いて詳細に説明する。

【0021】

[暗号装置]

図2は、本発明に係る暗号装置の概略的な構成を示したブロック図である。暗号装置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、変換処理部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。

【0022】

入力部20は、送信者が平文 x を入力するための入力インターフェースである。

【0023】

第一擬似乱数生成部30は、予測困難な擬似乱数列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

10

20

30

40

50

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。

【0024】

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、暗号的には安全ではない。しかし、本発明
10
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を後述する変換処理部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。

【0025】

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。

【0026】

変換処理部50は、データを非線形的に変換する処理を行うS-BOXなどを含めた関数系から構成されるものであり、後述する変換処理を行う。

【0027】

再配置処理部60は、データをブロックに分割し、当該ブロックを再配置するための再配置表を含めた関数系から構成されるものであり、後述する再配置処理を行う。
20

【0028】

出力部70は、最終的に生成された暗号文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。

【0029】

記憶部80は、入力部20～出力部70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
30
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。

【0030】

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。

【0031】

本実施例では、暗号装置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とから切り離れた構成としてもよい。いずれの構成においても、暗号装置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は、制御部90により記憶部80から読み出された暗号プログラムに従って制御される。
40

【0032】

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。
50

【 0 0 3 3 】

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。

【 0 0 3 4 】

[暗号化処理]

以上を前提として、図 2 に示した暗号装置 1 0 によって行われる再配置暗号について詳細に説明する。図 3 は、図 2 に示した暗号装置 1 0 によって行われる再配置暗号の処理手順を示したフローチャートである。

【 0 0 3 5 】

送信者により入力部 2 0 から平文 x (長さ: g ワード) が入力されると、制御部 9 0 は、これを記憶部 8 0 に記憶させ、記憶部 8 0 に格納された暗号プログラムに従い、第一擬似乱数生成部 3 0 ~ 生成部 7 0 に対して以下に示す処理を行うように促す。

10

【 0 0 3 6 】

ステップ S 1 0 において、制御部 9 0 は、第一擬似乱数生成部 3 0 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 8 0 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。

【 0 0 3 7 】

< ヘッダ r の生成アルゴリズムの実施例 >

ステップ S 1 0 における第一擬似乱数生成部 3 0 によるヘッダ r の生成アルゴリズムは、以下のように記述される。

20

```
r: array[0..a-1] of the word;
Randomize; //initialize G0 by the clock;
for i:=0 to a-1 do
```

```
  r[i]:=G0;
```

【 0 0 3 8 】

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、第一擬似乱数生成部 3 0 に対して、 $0, 1, \dots, n-1$ までの n 個の整数の疑似乱数列 (第二の疑似乱数列) 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 3 9 】

30

< 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 における第一擬似乱数生成部 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。

```
k: array[0..n-1] of the word;
Randomize; //initialize G0 by the clock;
for i:=0 to a-1 do
  k[i]:=i;
for i:=0 to N do
begin
  s:=G0 mod n;
  t:=G0 mod n;
  x:=k[s];
  k[s]:=k[t];
  k[t]:=x;
end;
```

40

ここで、1 ワードは 8 ビット、16 ビット、または 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 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。

【 0 0 4 0 】

なお、一般には、第一擬似乱数部 3 0 が生成する疑似乱数列は同じ数字の重複を含む。

50

このとき、第二の擬似乱数列から生成される再配置表 K における n 個の数字 (0 , 1 , . . . , n - 1) の並びには同じ数字の重複が生じてしまう。具体的には、分割数 n が十分に大きければ、第一擬似乱数生成部 30 が生成する第二の擬似乱数列には同じ数字の重複が存在することはないと考えられるが、分割数 n が小さい場合には、その可能性が生じる。そのような場合には、n 個の数字の並びが重複せず、かつ十分ランダムに配列されるように工夫して再配置表 K を作成する必要がある。

【 0 0 4 1 】

その場合、後述する [具体的な実装] における擬似乱数生成器 G 1 を使い、生成した擬似乱数列を用いて数万回並べ替えることが考えられる。しかしながら、それよりも効率的な方法として、第一擬似乱数生成部 30 が生成する擬似乱数列を周期的に拡張して、拡張した擬似乱数列から再配置表 K を生成し、さらに、生成された再配置表 K の統計的な偏りを補正することにより最終的に所望の再配置表 K を得るという方法が考えられる。

10

【 0 0 4 2 】

この方法は、次の 3 つのステップから構成される。

【 0 0 4 3 】

{ ステップ 1 }

第一擬似乱数生成部 30 において、短い擬似乱数列から 256 バイトの擬似乱数列を作成する。例えば、32 バイトの擬似乱数列を 256 バイトの擬似乱数列に拡張するアルゴリズムは、以下のように記述される。

< 実施例 >

20

```
procedure set32byte;
var
  i, j, a: integer
  d: array[0..255] of byte;
begin
  Randomize;
  for i:=0 to 31 do
  begin
    j:=random(256);
    d[i]:=j;
  end;
  for i:=32 to 255 do
  begin
    a:= i mod 32;
    d[i]:=d[a];
  end;
end;
```

30

{ ステップ 2 }

【 0 0 4 4 】

次に、記憶部 80 の擬似乱数列のメモリ領域に 256 バイトの擬似乱数列を順次格納していき、同じ数字が出たら次のメモリに格納するといった処理を繰り返すことにより再配置表 K を生成する。例えば、256 バイトの擬似乱数列から 256 バイトの再配置表 K を生成するアルゴリズムは、以下のように記述される。

40

< 実施例 >

```
procedure generate256k;
var
  d, st, k: array[0..255] of byte;
  i: integer;
  procedure set i; //dからkを作成する。
var
```

50

```

    j: integer;
begin
  if st[d[i]]=0 then
    begin
      k[d[i]]:=i;
      st[d[i]]:=1;
    end
  else
    begin
      j:=(d[i]+19) and 255;
      while st[j]=1 do
        begin
          j:=(j+67) and 255;
        end;
      k[j]:=i;
      st[j]:=1;
    end;
  end;
begin
  read_d;
  clear_st;
  for i:=0 to 255 do
    begin
      set i;
    end;
end;
{ ステップ 3 }

```

10

20

30

40

50

【 0 0 4 5 】

次に、ステップ 2 で作成された再配置表 K の統計的な偏りを、例えば、以下に示すようなアルゴリズムを用いて補正する。

< 実施例 >

```

procedure revise;
var
  k, k2: array[0..255] of byte;
  i: integer;
begin
  read_k;
  for l:=0 to 255 do
    begin
      k2[i]:=k[255-k[i]];
    end;
  end;
end;

```

【 0 0 4 6 】

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 から入力された平文 x を読み込み、これを平文 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) と共にこれを記憶部 8 0 に記憶させる。ここで、ヘッダ情報 u の長さ q は 4 ワードの符号なし整数として設定することができる。

【 0 0 4 7 】

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、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、平文

x と同じ長さ ($nm - a - 1$ ワード)の第二の擬似乱数列 $r_x = (r_0, r_1, \dots, r_{nm-a-1})$ を生成させ、これを記憶部80に記憶させる。

【0048】

なお、このステップにおいて、入力部20から入力される平文 x の長さ g が分割数 n の倍数ではない場合には、制御部90は、 $v - a - q - g \pmod{n}$ 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、平文 x 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 、パディング p を合わせたデータを改めて平文 $x = (u, x, z) = (x_0, x_1, \dots, x_{nm-a-1})$ 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、 $m = (a + q + g + v) / n$ として算出される。

10

【0049】

次に、ステップS50において、制御部90は、記憶部80から平文 $x = (x_0, x_1, \dots, x_{nm-a-1})$ と第二の擬似乱数列 $r_x = (r_0, r_1, \dots, r_{nm-a-1})$ とを読み出し、両者の排他的論理和をとる($c_i = x_i \text{ XOR } r_i$ ($i = 0, 1, \dots, nm - a - 1$))ことにより暗号文 $c_x = (c_0, c_1, \dots, c_{nm-a-1})$ を生成し、これを記憶部80に記憶させる。

【0050】

なお、本実施例では、このステップにおいて第二の擬似乱数列 r_x と平文 x 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S40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S50において平文 x 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。

20

【0051】

次に、ステップS60において、制御部90は、記憶部80から第一の擬似乱数列 r と暗号文 $c_x = (c_0, c_1, \dots, c_{nm-a-1})$ とを読み出し、第一の擬似乱数列 r を暗号文 c_x のヘッダとして付加したデータ $(r, c_x) = (c_0, c_1, \dots, c_{nm-1})$ を生成させ、これを改めて $c_x = (r, c_x) = (c_0, c_1, \dots, c_{nm-1})$ として記憶部80に記憶させる。

【0052】

次に、ステップS70において、制御部90は、ステップS80~S100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($Ct = 1$)、ステップS80へ処理を進める。

30

【0053】

ステップS80において、制御部90は、記憶部80からデータ $c_x = (c_0, c_1, \dots, c_{nm-1})$ を読み出し、変換処理部50に対して、例えば、図4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を用いてバイト単位で変換する変換処理を行わせ、変換されたデータを $d_x = F(r, c_x) = (d_0, d_1, \dots, d_{nm-1})$ として記憶部80に記憶させる。ここで、“ d ”はブロック差分と呼ばれる整数であり、 $0 < d < n$ の範囲で予め決められているものとする。また、図4に示した変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。

40

```
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。

【0054】

50

図6は、図4に示した変換処理のフローチャートにおいて、 $n = 3$, $m = 2$, $d = 1$ の場合を視覚化したものである。この例では、記号A1～A3の処理で示したように、長さ2ワードのブロックデータを縦方向に順次積層し、記号A4の処理において積層した値をブロック単位で足し合わせ、最後に記号A5の処理において、ブロック単位でS-BOXを用いて変換することにより変換処理を行っている。

【0055】

なお、この変換処理は、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、ここで用いた加算の代わりに減算を用いてもよい。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 $K = (k[0], k[1], \dots, k[n-1])$ 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。

10

【0056】

<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 $K = (k[0], k[1], \dots, k[n-1])$ 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。

($n \leq 256$ 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。

$ct := 0;$

for $i := 0$ to $n - 1$ do

begin

 if $(k[i] - e \geq 0)$ and $(k[i] - e < 256)$ then do

 begin

$s[ct] := k[i] - e;$

$ct := ct + 1$

 end;

end;

20

30

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。

【0057】

次に、ステップS90において、制御部90は、記憶部80から変換されたデータ $d_x = (d_0, d_1, \dots, d_{nm-1})$ 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 $d_x = (b_0, b_1, \dots, b_{n-1})$ として記憶部80に記憶させる。

40

【0058】

次に、ステップS100において、制御部90は、記憶部80から分割されたデータ $d_x = (b_0, b_1, \dots, b_{n-1})$ 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5に示すように、分割されたデータ $d_x = (b_0, b_1, \dots, b_{n-1})$ を、再配置表 $K = (k[0], k[1], \dots, k[n-1])$ に基づき、 $d_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ のように再配置させ、これを記憶部80に記憶させる。なお、図5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

50

レスヘzバイトの記憶内容をコピーする処理を表す。

【0059】

次に、ステップS110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 = 2）、ステップS120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。

【0060】

ステップS120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S130に処理を進める。この場合、制御部90は、記憶部80から再配置されたデータ $d_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ を読み出し、最終的な暗号文 $fc_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ として出力部70に出力する。そして、出力部70は、後述する復号装置に対して暗号文 $fc_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ を送信してすべての処理を終了する。

10

【0061】

一方、ステップS120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S80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～S100までの処理を繰り返した後、すべての処理を終了する。

【0062】

[暗号装置10の効果]

暗号装置10は、実質的に二つの鍵を用いて平文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、暗号装置10は、以下のような効果をもたらす。

20

【0063】

(1) 第一擬似乱数生成部30は、平文を暗号化するたびに異なる第一の擬似乱数列を生成することができる。異なる第一の擬似乱数列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で生成される第二の擬似乱数列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。

30

【0064】

(2) 第一擬似乱数生成部30が生成する第二の擬似乱数列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^{168} 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。

【0065】

(3) 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。

40

【0066】

(4) 上記(2)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。

【0067】

(5) 上記(4)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。

50

【 0 0 6 8 】

(6) また、暗号装置 1 0 は、再配置処理部 6 0 を設けたことにより、同じ平文 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、同じ平文 x と同じ第一の鍵 r とから、毎回異なる暗号文が無数にできることを意味する。

【 0 0 6 9 】

(7) さらに、暗号装置 1 0 においては、変換処理部 5 0 で用いられる $S - B O X$ 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 $S - B O X$ の構成よりも簡単であり、 $S - B O X$ の研究開発にコストがかからない。

10

【 0 0 7 0 】

[復号装置]

図 7 は、本発明に係る復号装置の概略的な構成を示したブロック図である。復号装置 1 0 0 は、入力部 1 2 0 と、公開可能な第二擬似乱数生成部 1 4 0 と、逆変換処理部 1 5 0 と、逆再配置処理部 1 6 0 と、出力部 1 7 0 と、記憶部 1 8 0 と、制御部 1 9 0 とを備える。このうち、記憶部 1 8 0 と制御部 1 9 0 とを除く部分を復号文生成部 1 0 0 A と称することにする。

【 0 0 7 1 】

入力部 1 2 0 は、送信者から送られてきた暗号文 $f c x$ を受信者が受け取るための入力インターフェースである。第二擬似乱数生成部 1 4 0 は、暗号装置 1 0 の第二擬似乱数生成部 4 0 と同様の構成を有する。逆再配置処理部 1 6 0 は、暗号装置 1 0 の再配置処理部 6 0 と同様の構成を有し、後述する逆再配置処理を行う。逆変換処理部 1 5 0 は、暗号装置 1 0 の変換処理部 5 0 と同様の構成を有し、後述する逆変換処理を行う。出力部 1 7 0 は、最終的に復号された復号文 (平文 x) を出力するための出力インターフェースである。

20

【 0 0 7 2 】

記憶部 1 8 0 は、入力部 1 2 0 ~ 出力部 1 7 0 から成る復号文生成部 1 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、後述する復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。

30

【 0 0 7 3 】

制御部 1 9 0 は、記憶部 1 8 0 から読み出した復号プログラムに従って、入力部 1 2 0 ~ 記憶部 1 8 0 を制御する CPU を備える。

【 0 0 7 4 】

本実施例では、復号装置 1 0 0 を、復号文生成部 1 0 0 A 及び制御部 1 9 0 と、記憶部 1 8 0 とを一体化した構成としたが、記憶部 1 8 0 を独立した記憶装置として復号文生成部 1 0 0 A 及び制御部 1 9 0 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、復号装置 1 0 0 はコンピュータによって実現されるものであり、入力部 1 2 0 ~ 出力部 1 7 0 は、制御部 1 9 0 により記憶部 1 8 0 から読み出された暗号プログラムに従って制御される。

40

【 0 0 7 5 】

[復号化処理]

以上を前提として、図 7 に示した復号装置 1 0 0 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 8 は、図 7 に示した復号装置 1 0 0 によって行われる復号化の処理手順を示したフローチャートである。

【 0 0 7 6 】

暗号装置 1 0 から送信された暗号文 $f c x = (f_0, f_1, \dots, f_{n m - 1})$ が入力部 1 2 0 から入力されると、制御部 1 9 0 は、これを記憶部 1 8 0 に記憶させ、記憶部 1 8 0 に格納された暗号プログラムに従い、第二擬似乱数生成部 1 4 0 ~ 出力部 1 7 0 に対して以下に示す処理を行うように促す。

50

【 0 0 7 7 】

ステップ S 2 0 0 において、制御部 1 9 0 は、暗号装置 1 0 から送信された暗号文の長さ $n m$ を入力部 1 2 0 から読み込ませ、これを記憶部 1 8 0 に記憶させる。

【 0 0 7 8 】

次に、ステップ S 2 1 0 において、制御部 1 9 0 は、記憶部 1 8 0 から暗号文 $f c_x$ の長さ $n m$ と予め格納された分割数 n とを読み出す。

【 0 0 7 9 】

次に、ステップ S 2 2 0 において、制御部 1 9 0 は、読み出した暗号文 $f c_x$ の長さ $n m$ と分割数 n とから、ブロックデータの長さ m を $m = n m / n$ として算出する。

【 0 0 8 0 】

次に、ステップ S 2 3 0 において、制御部 1 9 0 は、ステップ S 2 4 0 ~ S 2 6 0 で行われる分割処理と逆再配置処理と逆変換処理とを 1 セットとした処理のラウンド数を表すフラグ $C t$ を立て ($C t = 1$)、ステップ S 2 4 0 へ処理を進める。

【 0 0 8 1 】

ステップ S 2 4 0 において、制御部 1 9 0 は、記憶部 1 8 0 から暗号文 $f c_x = (f_0, f_1, \dots, f_{n m - 1})$ 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 $d_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ として記憶部 1 8 0 に記憶させる。

【 0 0 8 2 】

次に、ステップ S 2 5 0 において、制御部 1 9 0 は、記憶部 1 8 0 からデータ $d_x = (b_{k[0]}, b_{k[1]}, \dots, b_{k[n-1]})$ 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 1 6 0 に対して、図 9 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、逆配置されたデータを $d_x = (b_0, b_1, \dots, b_{n-1})$ として記憶部 1 8 0 に記憶させる。

【 0 0 8 3 】

次に、ステップ S 2 6 0 において、制御部 1 9 0 は、記憶部 1 8 0 から逆配置されたデータ $d_x = (b_0, b_1, \dots, b_{n-1})$ を読み出し、逆変換処理部 1 5 0 に対して、例えば、図 1 0 に示すように、これを $S - B O X$ を用いて逆変換させ、さらに、逆変換されたデータを $d_x = (d_0, d_1, \dots, d_{n m - 1})$ を $c_{(d m + i) \bmod n m} = c_{(d m + i) \bmod n m - c_{i \bmod n m}}$ ($i = n m - 1, n m - 2, \dots, 1, 0$) として減算する逆変換処理を行わせ、逆変換されたデータを $c_x = (c_0, c_1, \dots, c_{n m - 1})$ として記憶部 1 8 0 に記憶させる。

【 0 0 8 4 】

ここで、逆変換とは、暗号装置 1 0 の $S - B O X$ の非線形変換を表す関数 s の逆関数 i_s 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。

```
for i:=0 to 255 do
  is[s[i]]:=i;
```

【 0 0 8 5 】

次に、ステップ S 2 7 0 において、制御部 1 9 0 は、フラグ $C t$ の値をインクリメントし ($C t = 2$)、ステップ S 2 8 0 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。

【 0 0 8 6 】

ステップ S 2 8 0 において、制御部 1 9 0 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 S 2 9 0 に処理を進める。

【 0 0 8 7 】

ステップ S 2 9 0 において、制御部 1 9 0 は、記憶部 1 8 0 から逆変換されたデータ $c_x = (c_0, c_1, \dots, c_{n m - 1})$ と予め格納された数値 “ a ” とを読み出し、逆変換されたデータ $c_x = (c_0, c_1, \dots, c_{n m - 1})$ の先頭から a ワードをヘッダ r として規定し、逆変換されたデータ c_x を改めて $c_x = (r(\text{ヘッダ}), c_x(\text{残りのデータ}))$ として記憶部 1 8 0 に記憶させる。

10

20

30

40

50

【0088】

次に、ステップS300において、制御部190は、記憶部180から逆一体化された $c_x = (r, c_x)$ を読み出し、第二疑似乱数生成部140に対して、ヘッダ r を初期値とした $nm - a$ ワードの疑似乱数列 $r_x = (r_0, r_1, \dots, r_{nm-a-1})$ を生成させる。

【0089】

そして、ステップS310において、制御部190は、逆一体化された残りのデータ $c_x = (c_{a+1}, c_{a+2}, \dots, c_{nm-a-1})$ と生成された疑似乱数列 $r_x = (r_0, r_1, \dots, r_{nm-a-1})$ とを排他的論理和する ($x_{a+i} = c_{a+i} \oplus r_i$ ($i = 0, 1, \dots, nm-a-1$)) ことにより、データ $x = (x_a, x_{a+1}, \dots, x_{nm-1})$ を算出し、次いで、データ x の先頭から q ワードを平文 x のデータ長 g などの情報を含んだヘッダ情報 u として切り取り、さらに、残りのデータの先頭から g ワードのみを平文 x として復号し、これを記憶部180に記憶させる。

10

【0090】

なお、最後に切り捨てられた $nm - q - g$ ワードのデータはパディングである。

【0091】

そして最後に、ステップS320において、制御部190は、記憶部180から平文 x を読み出し、出力部170に出力させて、全ての処理を終了する。

【0092】

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～S260までの処理を繰り返した後、すべての処理を終了する。

20

【0093】

このように、復号装置100は、暗号装置10から送信されてきた暗号文 fc_x 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 とを暗号装置10と共有している。これにより、復号装置100は、復号の過程で暗号文 fc_x の本当の鍵とも言える疑似乱数列のヘッダ r が入手でき、平文 x の長さ g も同様に入手できるので、疑似乱数列のパディング p を切り捨てることができる。

30

【0094】

[暗号装置と復号装置のその他の構成]

上記においては暗号装置と復号装置とを別体として説明したが、これはあくまで各装置の構成及び機能の説明を容易にするためである。上記の説明から明らかなように、本発明においても、従来の共通鍵方式と同様、暗号化処理と復号化処理とは可逆の関係にあるので、暗号装置と復号装置とを一体の構成とした装置に適用できることは明らかである。

【0095】

[具体的な実装]

本発明に係る暗号装置及び復号装置のプログラム上の実装例を示す。ここでは、1ワードを1バイトとし、疑似乱数のヘッダ r は256ビットを使用する。従って、 $a = 32$ である。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、暗号装置のシステムクロックや送信者による入力部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をヘッダ r として使用する。また、分割数 n は $n = 256$ 、平文のデータ長 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。

40

【0096】

平文 x 、ヘッダ情報 u 、疑似乱数のヘッダ r をあわせた全データが1024バイトの倍

50

数になるように、平文 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、LongWord (32ビット符号なし整数) である。

【0097】

なお、以降のプログラムの変数の中には、上記の実施例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。

【0098】

< 公開可能な疑似乱数生成部 40 の実装例 >

```
d: array[0..7] of LongWord;
```

```
i; integer;
```

```
i:=0;
```

```
function g1: LongWord;
```

```
begin
```

```
  w.d:=d[i and 7]+d[(i-1) and 7];
```

```
  d[i and 7]:=w.d;
```

```
  w.b[0]:=k[w.b[0]]; w.b[1]:=k[w.b[1]];
```

```
  w.b[2]:=k[w.b[2]]; w.b[3]:=k[w.b[3]];
```

```
  g1:=w.d; //va;ue of g1
```

```
  i:=(i+1) and 7;
```

```
end;
```

ここで、 w は LongWord (32ビット符号なし整数) d と 4 バイトの配列 $b[0]$, $b[1]$, $b[3]$ の共用体である。

【0099】

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_x と平文の疑似乱数のヘッダ r 以降の部分を実行的論理和して暗号化したものを、LongWord の配列として、改めて $x = (x[0], x[1], \dots, x[v-1])$ ($v = 256m$) とする。ここで、 m は、ブロックデータの長さを LongWord の個数で表したものである。

【0100】

< 変換処理の実装例 >

```
i; integer;
```

```
begin
```

```
  for i:=0 to v-m-1 do
```

```
    begin
```

```
      x[i+m]:=x[i+m];x[i];
```

```
      w.d:=x[i+m];
```

```
      w.b[0]:=k[w.b[0]]; w.b[1]:=k[w.b[1]];
```

```
      w.b[2]:=k[w.b[2]]; w.b[3]:=k[w.b[3]];
```

```
      x[i+m]:=w.d;
```

```
    end;
```

```
  for i:=v-m to v-1 do
```

```
    begin
```

```
      x[i+m-v]:=x[i+m-v]+x[i];
```

```
      w.d:=x[i+m-v];
```

```
      w.b[0]:=k[w.b[0]]; w.b[1]:=k[w.b[1]];
```

```
      w.b[2]:=k[w.b[2]]; w.b[3]:=k[w.b[3]];
```

```
      x[i+m-v]:=w.d;
```

```
    end;
```

```
end;
```

【0101】

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

10

20

30

40

50

、 $y_i = (x[m_i], x[m_i + 1], \dots, x[m_i + m - 1])$ ($i = 0, 1, \dots, 255$) である。

【0102】

< 逆変換処理の実装例 >

まず、逆置換 ik を次のように計算しておく。

```
for i:=0 to 255 do
```

```
  ik[k[i]]:=i;
```

逆置換 ik を用いて逆変換を次のように計算する。

```
  i; integer;
```

```
  begin
```

```
    for i:=v-1 downto v-m do
```

```
      begin
```

```
        w.d:=x[i+m-v];
```

```
        w.b[0]:=k[w.b[0]]; w.b[1]:=k[w.b[1]];
```

```
        w.b[2]:=k[w.b[2]]; w.b[3]:=k[w.b[3]];
```

```
        x[i+m-v]:=w.d;
```

```
        x[i+m-v]:=x[i+m-v]-x[i];
```

```
      end;
```

```
    for i:=v-m-1 downto 0 do
```

```
      begin
```

```
        w.d:=x[i+m]
```

```
        w.b[0]:=k[w.b[0]]; w.b[1]:=k[w.b[1]];
```

```
        w.b[2]:=k[w.b[2]]; w.b[3]:=k[w.b[3]];
```

```
        x[i+m]:=w.d;
```

```
        x[i+m]:=x[i+m]-x[i];
```

```
      end;
```

```
    end;
```

10

20

【0103】

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。

30

【0104】

< 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。

```
  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], 4*v);
```

```
  end;
```

40

【0105】

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。

```
  i; integer;
```

```
  begin
```

```
    for i:=0 to 255 do
```

50

```

begin
  Move(x[k[i]*m], y[i*m], m);
end;
Move(y[0], x[0], 4);
end;

```

【 0 1 0 6 】

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倍程度の暗号化の実行速度が達成できる。

【 0 1 0 7 】

[その他の実施例]

上記の実施例では、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暗号装置10をデザインした。その他の実施例として、ストリーム暗号をブロック暗号に変えた構成を有する暗号装置が考えられる。

【 0 1 0 8 】

図11は、その他の暗号装置の概略的な構成を示したブロック図である。暗号装置200は、入力部220と、第一擬似乱数生成部230と、ブロック暗号文生成部290と、変換処理部250と、再配置処理部260と、出力部270と、記憶部280と、制御部300とを備える。このうち、記憶部280と制御部300とを除く部分を暗号文生成部200Aと称することにする。

【 0 1 0 9 】

このうち、入力部220と、第一擬似乱数生成部230と、変換処理部250と、再配置処理部260と、出力部270とは、それぞれ、入力部20と、第一擬似乱数生成部30と、変換処理部50と、再配置処理部60と、出力部70と同様の機能を有するので、説明を省略する。

【 0 1 1 0 】

記憶部280は、入力部220～出力部270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。

【 0 1 1 1 】

また、制御部300は、記憶部280から読み出した暗号プログラムに従って、入力部220～記憶部280を制御するCPUを備える。

【 0 1 1 2 】

なお、記憶部280及び制御部300のハードとして構成は、上記実施例と同様の構成なので、説明を省略する。

【 0 1 1 3 】

[暗号化処理]

以上を前提として、図11に示した暗号装置200によって行われる再配置暗号について上記の実施例と異なる部分のみ説明する。図12は、図11に示した暗号装置200によって行われる再配置暗号の処理手順を示した概念図である。

【 0 1 1 4 】

暗号装置200は、上記の実施例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本発明の再配置暗号を実現するものである。

【 0 1 1 5 】

従って、ブロック暗号文生成部290において平文xをブロック暗号化する際に、平文xの長さがブロック長の倍数とならない場合には平文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300は、 $v - a - g - q \pmod{n}$ 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230に生成させ、平文xの最後に付加する処理を行う。そして、平文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)、平文x、パディングqを合わ

10

20

30

40

50

せたデータを改めて平文 $x = (x, z)$ 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、 $m = (a + q + g + p) / n$ として算出される。

【0116】

次のステップとして、制御部 300 は、記憶部 280 から平文 x (gワード)、平文 x のデータ長などの情報を含んだヘッダ情報 (q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 290 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。

【0117】

以降の処理は、ストリーム暗号を用いた上記の実施例と同様なので、説明を省略する。

10

【0118】

このように、本発明の暗号をブロック暗号に用いた場合、ストリーム暗号に用いた場合よりも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。

【0119】

[再配置暗号の NP 完全性]

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 (Peter W. Shor, "Algorithms for Quantum Computation: Discrete Logarithms and Factoring", In Proceeding of 35th IEEE FOCS, pp.124-134, Santa Fe, NM, Nov 20-22, 1994.)。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。

20

【0120】

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、本発明の再配置暗号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示される。

【0121】

証明に当たって、まず次の 3 つの問題を設定する。

30

【0122】

(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。

<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?

40

【0123】

(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。

<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 S_N と自然数 m が存在し、 $y = \sum_{i=1}^m x_{s(i)}$ とできるか?

【0124】

(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。

50

<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。

【0125】

このとき、本願発明の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証明される。

【0126】

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 に多項式時間帰着できる。つまり、

$$Z_0 <_p Z_1$$

【0127】

次に、関数 $f(X, y, s, m) = (E_s(X, y, m), (X, \prod_{i=1}^m x_{s(i)}, m), s)$ 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 $O(n)$ 程度であるので、 Z_1 は Z_2 に多項式時間帰着されることになる。つまり、関数 f は多項式時間計算可能関数であり、

$$Z_1 <_p Z_2$$

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 ($N = n$)。

【0128】

以上のことより、 Z_0, Z_1, Z_2 の各問題は、

$$Z_0 <_p Z_1 <_p Z_2$$

の順番に多項式時間帰着させることができることがわかる。上述したように、 Z_0 及び Z_1 は NP 完全であるので、結論として、 Z_2 も NP 完全であることが示されたことになる。

【0129】

このことより、統計的に偏りのない擬似乱数発生器を用いた場合、本発明の再配置暗号は量子コンピュータに対する耐性を持つと考えられる。実際に使用される擬似乱数発生器は統計的な偏りを持つわけだが、例えば、256個に分割された再配置暗号のブロックの順番を統計的に割り出すことは困難であると考えられる。

【0130】

[再配置暗号を用いたメッセージ認証]

上記で説明した実施形態においては、暗号文が改竄された場合、復号した平文から改竄を検知する機能は高くない。しかし、本暗号をハッシュ関数と組み合わせて使用することにより、この機能を補完することができる。

【0131】

これに関し、上述したように公開鍵暗号が量子コンピュータによって攻撃されるのと同様に、メッセージ認証に使われる鍵付きハッシュ関数 (MAC: Message Authentication Code) も攻撃されると考えられる。

【0132】

これに対抗するために、本発明においては、次に示すように幾つかの応用例が考えられる。

【0133】

<応用例1>

h を MD5 や SHA1 などのハッシュ関数とし、送信するメッセージ m を n バイトの配

10

20

30

40

50

列 $X = (x_0, x_1, \dots, x_{n-1})$ に格納することを考える。このとき、次の手順で認証機能付きメッセージ M を作成する。

【0134】

(S1) 再配置暗号の秘密鍵 K (S - BOX 又は再配置表) を用いて、配列 $X = (x_0, x_1, \dots, x_{n-1})$ を $(K[x_0], K[x_1], \dots, K[x_{n-1}])$ と置換する。

【0135】

(S2) $m' = E_K(h(X)) = h_K(m)$ を計算する。

【0136】

(S3) オリジナルのメッセージ m に対して、認証機能付きメッセージ M を $M = (m, m')$ とする。 10

【0137】

このようにして作成された認証機能付きメッセージ M が、共通鍵として秘密鍵 K を共有する通信において、攻撃者によって改竄されたか否かを、次のようにして確認することができる。

【0138】

メッセージ M を受信した受信者は、 $h(X)$ と $D_K(m')$ を計算して、それらが一致するか否かを調べる。このとき、一致すればメッセージ M は改竄されていないことが分かる。

【0139】

この応用例の利点として、通常、ハッシュ関数は必ず衝突があるので、小さい確率ではあるが、メッセージを改竄することができる。しかし、本応用例では、オリジナルのメッセージ m 及び秘密鍵 K が同じでも、それから作成された m' は無数にあるので、攻撃者はメッセージの改竄に成功したか否かを確認することはできない。 20

【0140】

< 応用例 2 >

応用例 1 の方向とは異なり、本発明の再配置暗号を利用して全く新しい MAC を作成することも可能である。いま、再配置暗号の秘密鍵 K を、0, 1, 2, ..., 255 を並べ替えた 256 バイトの S - BOX とする。そして、メッセージ m を n バイトの配列 $X = (x_0, x_1, \dots, x_{n-1})$ に格納することを考える。このとき、次の手順で認証機能付きメッセージ M を作成する。 30

【0141】

(S1) $i = 0, j = 0$ とする。

【0142】

(S2) 秘密鍵 K を用いて、配列 $X = (x_0, x_1, \dots, x_{n-1})$ を $(K[x_0], K[x_1], \dots, K[x_{n-1}])$ と置換する。

【0143】

(S3) $x_{(j+1) \bmod n} = x_{(j+i) \bmod n} + x_j$ とし、その後、 j をインクリメントする。

【0144】

(S4) $j < n$ ならば、(S3) へ行く。 40

【0145】

(S5) i をインクリメントする。

【0146】

(S6) $i < R$ ならば、 $j = 0$ として (S3) へ行く。ここで、 R は攪拌の回数を表す所定値であり、通常 3 回ぐらいに設定する。

【0147】

(S7) $m' = E_K(x_0, x_1, \dots, x_{n'-1}) = h_K(m)$ を計算する。

【0148】

ここで、 n' は n よりも小さい自然数である。つまり、本応用例においては、一体化さ 50

れたデータの一部を暗号化し、ハッシュ表として使用するので、このハッシュ表は平文の長さよりもずっと短い。

【0149】

この場合も、秘密鍵 K を共有する通信で、メッセージ $M = (m, m')$ を受信した受信者は、(S1) ~ (S6) を実行し、 $(x_0, x_1, \dots, x_{n-1})$ と $D_K(m')$ が一致するか否かを調べることでメッセージ認証を行うことができる。この応用例はかなり高速で行うことができ、やはり、攻撃者がメッセージの改竄に成功したか否かを確認することはできない。

【0150】

< 応用例 3 >

また、別の応用例として、メッセージの受信者は、ランダムな再配置暗号の秘密鍵 K' を用いて、次の手順に従うことによって、送信者が送信内容を否認できないメッセージ認証を行うことができる。

【0151】

(S1) 送信者は、< 応用例 2 > の MAC を計算し、 $M = (m, m')$ を受信者に送信する。

【0152】

(S2) 受信者は、ランダムな再配置暗号の秘密鍵 K' を作成し、 $y = E_{K'}(m, h_{K'}(m))$ を作成して、送信者に送り返す。

【0153】

(S3) 送信者は、 $D_{K'}(y) = (m, h_{K'}(m))$ を計算し、正当な通信であることを確認した後、 $m'' = E_{K'}(h_{K'}(m))$ を計算し、 (m, m'') を受信者に送り返す。

【0154】

(S4) 受信者は、 $D_{K'}(m'')$ を計算し、 $h_{K'}(m)$ と一致するか否かを調べ、一致した場合には (m, K', m'') を保存する。

【0155】

この応用例における MAC の衝突は秘密鍵 K' に依存する。送信者は、 K' を知らないため、 m'' の衝突を計算できない。従って、送信者は、メッセージ m の送信を否認できない。もちろん、第三者がこの通信内容を改竄することもできない。

【0156】

以上、幾つかの例を挙げて説明したように、本発明の再配置暗号を用いると、量子コンピュータによっても攻撃されないメッセージ認証を行うことができるので、これを電子決済の基本として電子商取引システムに適用すれば、将来的に量子コンピュータに対して耐性を有する電子商取引システムを開発することができる。その意味において、本発明の再配置暗号は、今後さらに活発化するであろう電子商取引システムの基盤と十分に成り得る暗号方式であると考えられる。

【0157】

なお、本発明は、上述した実施形態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構成要素を変形して具現化することができる。また、上記した実施形態に開示された構成要素を適宜組み合わせることにより各種の発明を形成することができる。例えば、各実施形態に示される構成要素から幾つかの構成要素を削除してもよい。また、ある実施形態に示される構成要素を別の実施形態に追加してもよい。さらに、異なる実施形態に亘って構成要素を適宜組み合わせてもよい。

【産業上の利用可能性】

【0158】

本発明によれば、既知平文攻撃を極めて困難にすると共に、暗号化の実行速度を極めて速くすることができる暗号装置及び暗号プログラム、これらの暗号装置及び暗号プログラムによって作成された暗号文を復号化する復号装置及び復号プログラム、及びこれらのプログラムを記憶する記憶媒体を提供することができる。

10

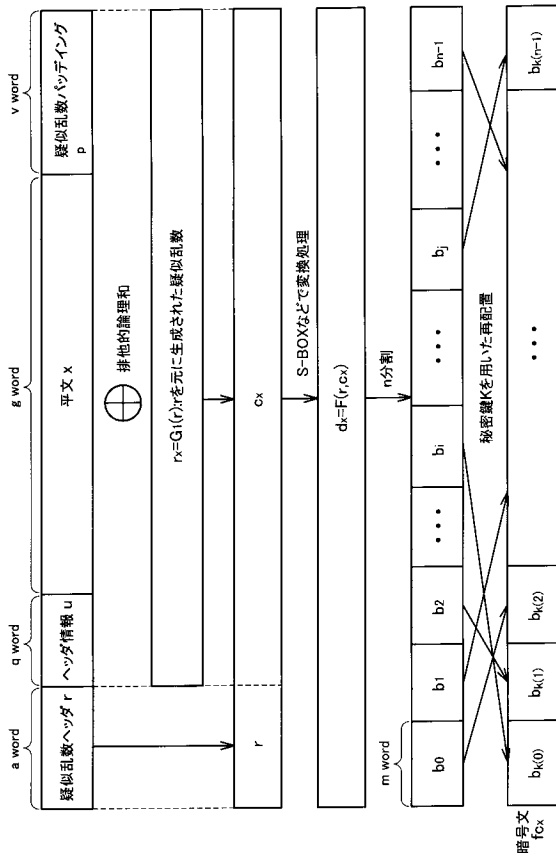
20

30

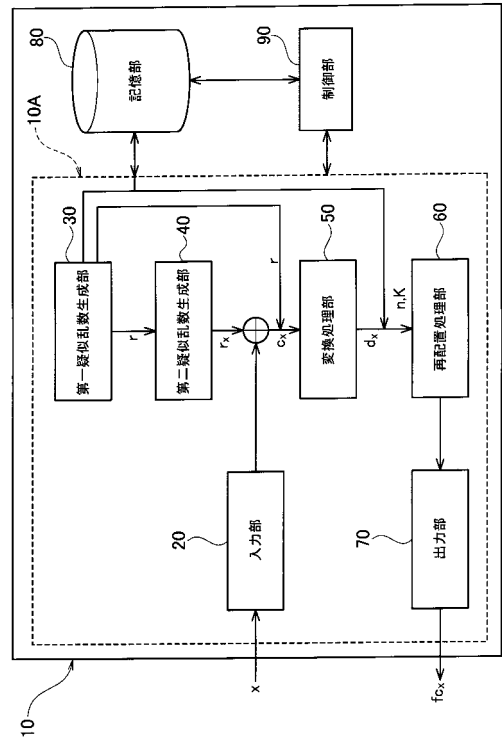
40

50

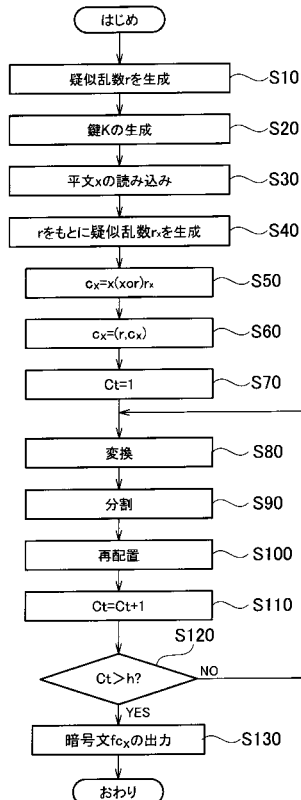
【図 1】



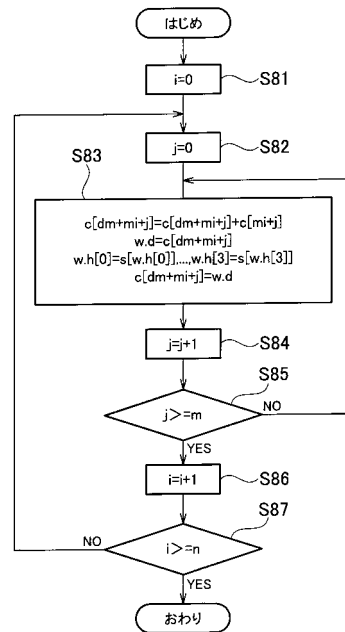
【図 2】



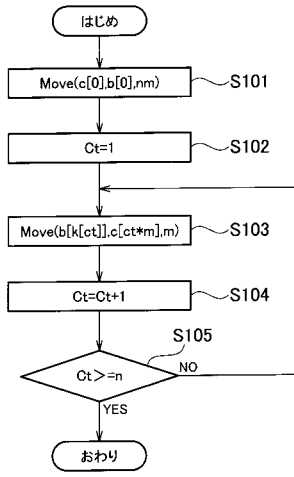
【図 3】



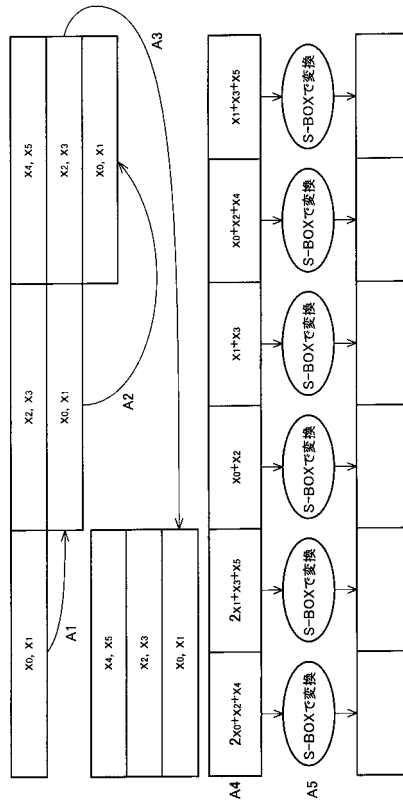
【図 4】



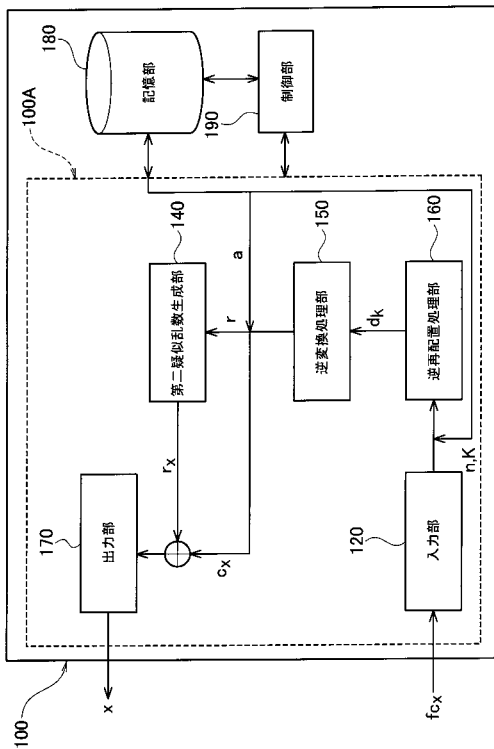
【 図 5 】



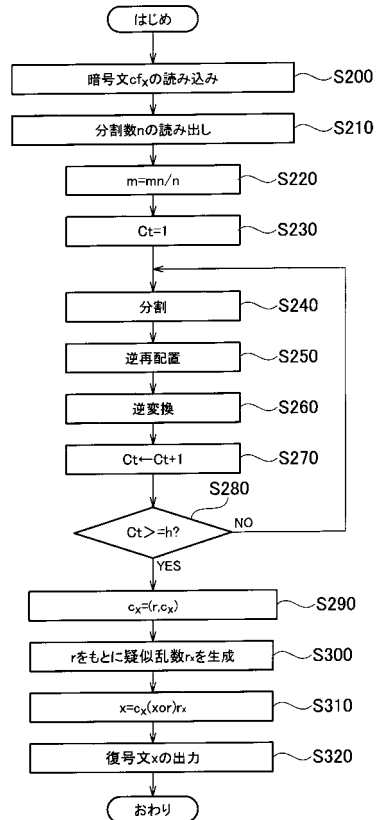
【 図 6 】



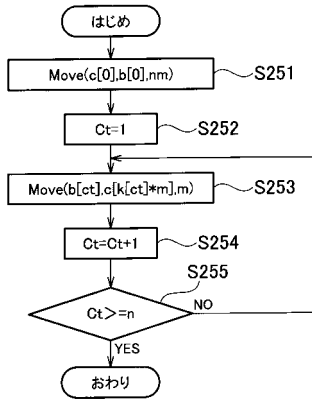
【 図 7 】



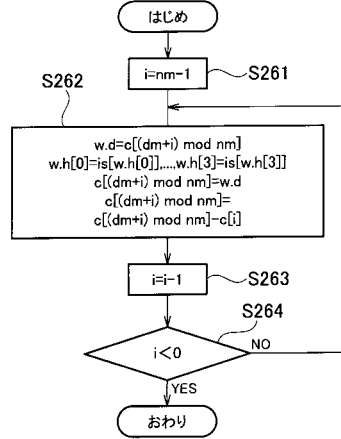
【 図 8 】



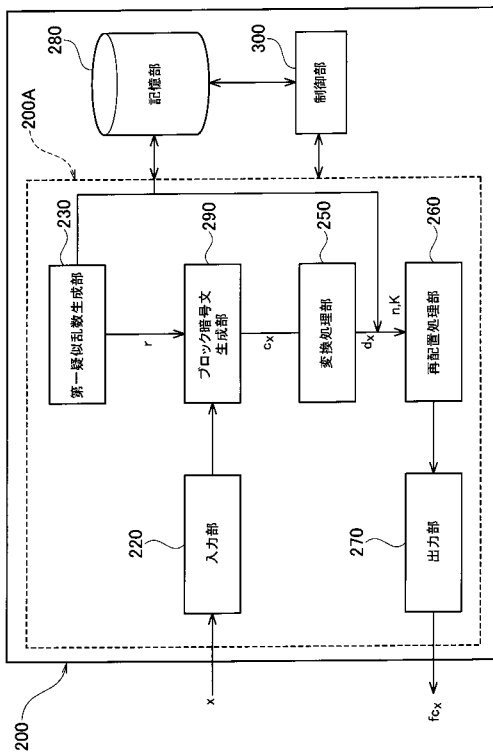
【図9】



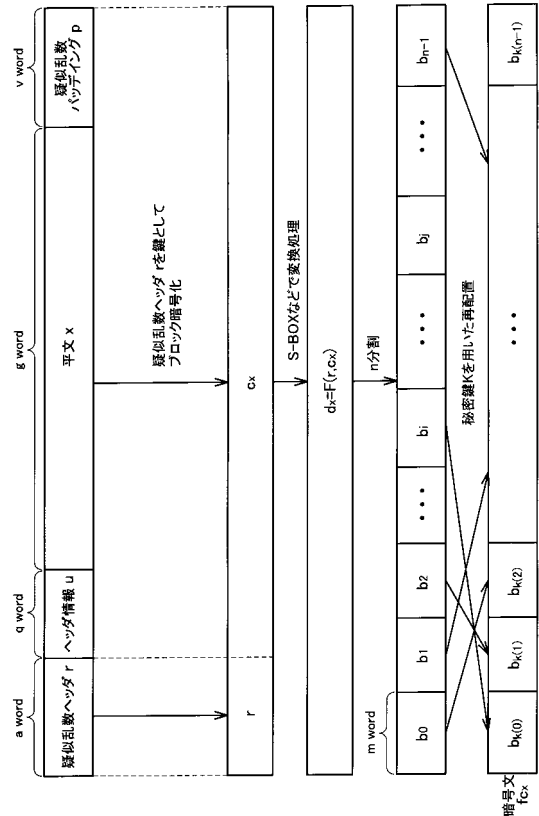
【図10】



【図11】



【図12】



【 国際調査報告 】

INTERNATIONAL SEARCH REPORT		International application No. PCT/JP2008/055132
A. CLASSIFICATION OF SUBJECT MATTER G09C1/00 (2006.01) i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) G09C1/00		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2008 Kokai Jitsuyo Shinan Koho 1971-2008 Toroku Jitsuyo Shinan Koho 1994-2008		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP 2006-191626 A (Toshiba Corp., Toshiba Tec Corp.), 20 July, 2006 (20.07.06), Par. Nos. [0016] to [0028] & US 2006/0153374 A1	1-17
Y	JP 62-81145 A (Hitachi, Ltd.), 14 April, 1987 (14.04.87), Page 2, upper left column, lines 9 to 15; lower right column, lines 15 to 18 (Family: none)	1-17
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 12 May, 2008 (12.05.08)		Date of mailing of the international search report 27 May, 2008 (27.05.08)
Name and mailing address of the ISA/ Japanese Patent Office		Authorized officer
Facsimile No.		Telephone No.

国際調査報告		国際出願番号 PCT/JP2008/055132									
A. 発明の属する分野の分類 (国際特許分類 (IPC)) Int.Cl. G09C1/00(2006.01)i											
B. 調査を行った分野 調査を行った最小限資料 (国際特許分類 (IPC)) Int.Cl. G09C1/00											
最小限資料以外の資料で調査を行った分野に含まれるもの <table border="0"> <tr> <td>日本国実用新案公報</td> <td>1922-1996年</td> </tr> <tr> <td>日本国公開実用新案公報</td> <td>1971-2008年</td> </tr> <tr> <td>日本国実用新案登録公報</td> <td>1996-2008年</td> </tr> <tr> <td>日本国登録実用新案公報</td> <td>1994-2008年</td> </tr> </table>				日本国実用新案公報	1922-1996年	日本国公開実用新案公報	1971-2008年	日本国実用新案登録公報	1996-2008年	日本国登録実用新案公報	1994-2008年
日本国実用新案公報	1922-1996年										
日本国公開実用新案公報	1971-2008年										
日本国実用新案登録公報	1996-2008年										
日本国登録実用新案公報	1994-2008年										
国際調査で使用了電子データベース (データベースの名称、調査に使用了用語)											
C. 関連すると認められる文献											
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号									
Y	JP 2006-191626 A (株式会社東芝、東芝テック株式会社) 2006.07.20, 段落【0016】 - 【0028】 & US 2006/0153374 A1	1-17									
Y	JP 62-81145 A (株式会社日立製作所) 1987.04.14, 第2頁左上欄 第9-15行, 右下欄第15-18行 (ファミリーなし)	1-17									
<input type="checkbox"/> C欄の続きにも文献が列挙されている。		<input type="checkbox"/> パテントファミリーに関する別紙を参照。									
* 引用文献のカテゴリー 「A」特に関連のある文献ではなく、一般的技術水準を示すもの 「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」口頭による開示、使用、展示等に言及する文献 「P」国際出願日前で、かつ優先権の主張の基礎となる出願		の日の後に公表された文献 「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」同一パテントファミリー文献									
国際調査を完了した日 12.05.2008		国際調査報告の発送日 27.05.2008									
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 速水 雄太	5S 4180								
		電話番号 03-3581-1101	内線 3546								

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(注) この公表は、国際事務局(WIPO)により国際公開された公報を基に作成したものである。なおこの公表に係る日本語特許出願(日本語実用新案登録出願)の国際公開の効果は、特許法第184条の10第1項(実用新案法第48条の13第2項)により生ずるものであり、本掲載とは関係ありません。