

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6296589号
(P6296589)

(45) 発行日 平成30年3月20日 (2018. 3. 20)

(24) 登録日 平成30年3月2日 (2018. 3. 2)

(51) Int. Cl. F 1
 H04L 9/06 (2006.01) H04L 9/00 611Z
 G09C 1/00 (2006.01) G09C 1/00 610A

請求項の数 5 (全 17 頁)

<p>(21) 出願番号 特願2013-98521 (P2013-98521) (22) 出願日 平成25年5月8日 (2013. 5. 8) (65) 公開番号 特開2014-219548 (P2014-219548A) (43) 公開日 平成26年11月20日 (2014. 11. 20) 審査請求日 平成28年4月11日 (2016. 4. 11)</p>	<p>(73) 特許権者 504155293 国立大学法人島根大学 島根県松江市西川津町1060 (74) 代理人 100081673 弁理士 河野 誠 (74) 代理人 100141483 弁理士 河野 生吾 (72) 発明者 六井 淳 島根県松江市西川津町1060 国立大学 法人島根大学内 審査官 中里 裕正</p>
--	---

最終頁に続く

(54) 【発明の名称】 暗号処理システム

(57) 【特許請求の範囲】

【請求項 1】

平文を暗号文に暗号化するか、或いは暗号文を平文に復号化する暗号処理システムであって、

データが入力される入力部と、

暗号鍵を提供する提供部と、

前記入力部から入力された平文を提供部から提供された暗号鍵に基づいて暗号化するか、或いは入力部から入力された暗号を、提供部から提供された暗号鍵に基づいて復号化する変換部と、

該変換部で変換されたデータを出力する出力部とを備え、

上記暗号鍵は、

平文を所定データ長毎に分割して得られる複数の平文分割データと、暗号文を所定データ長毎に分割して得られる複数の暗号文分割データとを、全てノードとして配置可能な木構造データの該リンク情報と、

各平文分割データから各暗号文分割データを求めることが可能であるとともに各暗号文分割データから各平文分割データを求めることが可能なように木構造データの対応するノードにランダムに配置された各平文分割データ及び各暗号文分割データの該配置情報とを含み、

前記木構造データは、子ノードのみとして機能するリーフノード以外のノードは、該ノードに接続された下層の複数のノードに対して親ノードとなり、該複数の下層ノードは該

親ノードの子ノードとなり、

前記木構造データは、任意の親ノードと、該親ノードにリンクを介して接続された複数の子ノードとにおいて、一のノードに格納されたデータを、該一のノード以外の複数のノードに格納されたデータから求めることが可能なように構成された

暗号処理システム。

【請求項 2】

前記変換部は入力部から入力される平文を暗号化する暗号化部であり、
該暗号化部は、
入力された平文を上記所定データ長毎に分割する分割処理と、
分割処理によって分割された各平文分割データから、提供部により提供される暗号鍵に
基づいて、各暗号文分割データを算出するメイン処理と、
メイン処理によって求められた各暗号文分割データを結合して暗号化された暗号文を生成する結合処理とを行う

10

請求項 1 に記載の暗号処理システム。

【請求項 3】

前記変換部は入力部から入力される暗号文を復号化する復号化部であり、
該復号化部は、
入力された暗号文を上記所定データ長毎に分割する分割処理と、
分割処理によって分割された各暗号文分割データから、提供部により提供される暗号鍵
に基づいて、各平文分割データを算出するメイン処理と、
メイン処理によって求められた各平文分割データを結合して復号化された平文を生成する結合処理とを行う

20

請求項 1 または 2 の何れかに記載の暗号処理システム。

【請求項 4】

平文分割データ又は暗号文分割データの数を 11 以上とした

請求項 1 乃至 3 の何れかに記載の暗号処理システム。

【請求項 5】

暗号鍵を生成する生成部と、

パスワードを取得する取得部とを備え、

上記木構造データのリーフノードは、分割された複数の平文分割データと同数設けられ、
該複数のリーフノードには、対応する平文分割データがそれぞれ格納され、
該木構造データの任意の親ノードは、リンクを介して接続される 2 つの子ノードを有し

30

前記生成部は、

上記取得部で取得されるパスワードに基づいて再現可能な擬似乱数を発生させる発生処理と、

発生処理によって発生させた擬似乱数に基づいて木構造データの上記リンク情報及び配置情報を再現可能にランダムに決定する決定処理と、

上記木構造データのリンク情報及び配置情報に基づいて暗号鍵を成形する生成処理とを行う

40

請求項 1 乃至 4 の何れかに記載の暗号処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

この発明は、平文の暗号化又は暗号文の復号化を行う暗号処理システムに関する。

【背景技術】

【0002】

情報化社会の到来に伴って、通信内容や記憶装置に記憶されたデータの秘密保持を目的とした暗号技術の研究が盛んに行われ、実用化されている。この一例として、暗号化及び復号化のための情報が示された暗号鍵を用いて、平文の暗号化や暗号文の復号化を行う D

50

ES (Data Encryption Standard) や AES (Advanced Encryption Standard) 等の暗号方式を用いた暗号処理システムが開発され、公知になっている (例えば、特許文献 1, 2 を参照。)。

【先行技術文献】

【特許文献】

【0003】

【特許文献 1】特開 2000 - 162965 号公報

【特許文献 2】特許第 5042272 号公報

【発明の概要】

【発明が解決しようとする課題】

10

【0004】

上記文献に示す暗号処理システムでは、解読の困難性を高めて安全性を確保する場合、暗号鍵のビット長が長く設定するか、或いは複数回の暗号化処理を行う等によって、解読のために必要な計算量を増加させる必要があるので、安全性と、暗号化又は復号化の処理の高速化とを両立することが困難である。

【0005】

本発明は、平文の暗号化の処理又は暗号文の復号化の処理を高速で行うことが可能であるととも、安全性も高い暗号処理システムを提供することを課題とする。

【課題を解決するための手段】

【0006】

20

本発明の暗号処理システムは、平文を暗号文に暗号化するか、或いは暗号文を平文に復号化する暗号処理システムであって、データが入力される入力部と、暗号鍵を提供する提供部と、前記入力部から入力された平文を提供部から提供された暗号鍵に基づいて暗号化するか、或いは入力部から入力された暗号を、提供部から提供された暗号鍵に基づいて復号化する変換部と、該変換部で変換されたデータを出力する出力部とを備え、上記暗号鍵は、平文を所定データ長毎に分割して得られる複数の平文分割データと、暗号文を所定データ長毎に分割して得られる複数の暗号文分割データとを、全てノードとして配置可能な木構造データの該リンク情報と、各平文分割データから各暗号文分割データを求めることが可能であるとともに各暗号文分割データから各平文分割データを求めることが可能なように木構造データの対応するノードにランダムに配置された各平文分割データ及び各暗号文分割データの該配置情報とを含み、前記木構造データは、子ノードのみとして機能するリーフノード以外のノードは、該ノードに接続された下層の複数のノードに対して親ノードとなり、該複数の下層ノードは該親ノードの子ノードとなり、前記木構造データは、任意の親ノードと、該親ノードにリンクを介して接続された複数の子ノードとにおいて、一のノードに格納されたデータを、該一のノード以外の複数のノードに格納されたデータから求めることが可能なように構成されたことを特徴とする。

30

【0007】

前記変換部は入力部から入力される平文を暗号化する暗号化部であり、該暗号化部は、入力された平文を上記所定データ長毎に分割する分割処理と、分割処理によって分割された各平文分割データから、提供部により提供される暗号鍵に基づいて、各暗号文分割データを算出するメイン処理と、メイン処理によって求められた各暗号文分割データを結合して暗号化された暗号文を生成する結合処理とを行うものとしてもよい。

40

【0008】

前記変換部は入力部から入力される暗号文を復号化する復号化部であり、該復号化部は、入力された暗号文を上記所定データ長毎に分割する分割処理と、分割処理によって分割された各暗号文分割データから、提供部により提供される暗号鍵に基づいて、各平文分割データを算出するメイン処理と、メイン処理によって求められた各平文分割データを結合して復号化された平文を生成する結合処理とを行うものとしてもよい。

【0009】

平文分割データ又は暗号文分割データの数を 11 以上としたものとしてもよい。

50

【 0 0 1 0 】

暗号鍵を生成する生成部と、パスワードを取得する取得部とを備え、上記木構造データのリーフノードは、分割された複数の平文分割データと同数設けられ、該複数のリーフノードには、対応する平文分割データがそれぞれ格納され、該木構造データの任意の親ノードは、リンクを介して接続される2つの子ノードを有し、前記生成部は、上記取得部で取得されるパスワードに基づいて再現可能な擬似乱数を発生させる発生処理と、発生処理によって発生させた擬似乱数に基づいて木構造データの上記リンク情報及び配置情報を再現可能にランダムに決定する決定処理と、上記木構造データのリンク情報及び配置情報に基づいて暗号鍵を成形する生成処理とを行うものとしてもよい。

【 発明の効果 】

10

【 0 0 1 1 】

暗号鍵を有していれば、暗号鍵に示されたリンク情報及び配置情報に基づいて、木構造データを迅速且つ簡易的に取得可能し、この木構造データから、少ない計算量で、直ちに平文への復号化処理または暗号文への暗号化を行うことが可能であるため、処理速度の高速化を図ることが可能である。これに加えて、暗号鍵がなければ、木構造データの取得も困難であるため、暗号解読が困難であり、安全性が高い。

【 図面の簡単な説明 】

【 0 0 1 2 】

【 図 1 】 本発明を適用した暗号処理システムのブロック図ある。

【 図 2 】 暗号化装置及び復号化装置の概略図である。

20

【 図 3 】 暗号化及び復号化の具体的手段を示す概念図である。

【 図 4 】 暗号化装置及び復号化装置の暗号生成部が行う暗号鍵生成処理のフロー図である。

【 図 5 】 擬似乱数生成手段の一例の構成を示す概念図である。

【 図 6 】 暗号化部が暗号鍵に基づいて行う平文の暗号化処理の手順を示す説明図であるとともに、復号化部が暗号鍵に基づいて行う暗号文の復号化処理の手順を示す説明図である。

【 図 7 】 暗号化部の処理フロー図である。

【 図 8 】 復号化部の処理フロー図である。

【 図 9 】 (A) , (B) は、それぞれ木構造データの他例を示す概念図である。

30

【 図 1 0 】 (A) 乃至 (E) は、それぞれ本暗号処理システムの検証実験の結果を示す一覧表である。

【 図 1 1 】 (A) は暗号化する際の処理速度を示し、(B) は復号化する際の処理速度を示している。

【 発明を実施するための形態 】

【 0 0 1 3 】

図 1 は、本発明を適用した暗号処理システムのブロック図ある。

図示する暗号処理システムは、暗号化装置 1 及び復号化装置 2 の一方又は両方 (図示する例では両方) を備えている。本暗号処理システムは、通信の暗号化や保存されたデータの暗号化等、広い範囲に適用可能である。

40

【 0 0 1 4 】

図 2 は、暗号化装置及び復号化装置の概略図である。

暗号化装置 1 及び復号化装置 2 は共にコンピュータである。このコンピュータ 1、2 は、A T 互換機と呼ばれる通常の P C であり、C P U 3 と、R A M 4 と、S S D や H D D 等の記憶装置 6 と、キーボードやマウス等の入力インターフェイス 7 と、モニタに画像を出力するビデオカード 8 と、フロッピー (登録商標) や C D - R 等の記憶媒体からデータを読み込むとともに該記憶媒体にデータを書込む外部記憶装置 9 と、他の P C と T C P / I P 通信可能なネットワークインターフェイス 1 1 とを備えている。

【 0 0 1 5 】

上記 R A M 4 上に実行されるプログラムによって、コンピュータが暗号化装置 1 及び復

50

号化装置 2 として機能する。

【 0 0 1 6 】

図 1 に示す通り、上記暗号化装置 1 は、暗号化されていないデータである平文が入力される入力部（入力手段）1 2 と、暗号化のための暗号鍵を提供する暗号鍵提供部（提供部、暗号鍵提供手段）1 3 と、前記記憶装置 6 または記憶媒体が設置された外部記憶装置 9 からなる記憶部 1 4 と、複数の英数字や記号からなる文字列であるパスワードが入力インターフェイス 7 等を介して入力された際にこのパスワードを取得するパスワード取得部（取得部、パスワード取得手段）1 6 と、パスワード取得部 1 6 で取得されたパスワードから暗号鍵を生成する暗号鍵生成部（生成部、暗号鍵生成手段）1 7 と、該入力部 1 2 から入力された平文を、暗号鍵提供部 1 3 から提供される暗号鍵に基づいて暗号化する暗号化部（変換部、暗号化手段）1 8 と、該暗号化部 1 8 で暗号化されたデータである暗号文を出力する出力部 1 9 とを備えている。

10

【 0 0 1 7 】

入力部 1 2 から入力される平文は、通常のブロック暗号やストリーム暗号とは異なり、そのままのサイズで、暗号化部 1 8 に渡される。

【 0 0 1 8 】

暗号鍵生成部 1 7 は、自身で生成した暗号鍵を、記憶部 1 4 に記憶するか、或いは暗号鍵提供部 1 3 に直接渡す。暗号鍵提供部 1 3 は、上記のように暗号鍵生成部 1 7 から直接暗号鍵を受取るか、或いは、上記のようにして予め記憶部 1 4 に記憶された暗号鍵を読み込む。なお、暗号鍵提供部 1 3 が公開鍵等で暗号化された通信経路を介してネットワークインターフェイス 1 1 から暗号鍵を取得してもよい。

20

【 0 0 1 9 】

上記復号化装置 2 は、暗号化されたデータである暗号文が入力される入力部（入力手段）2 1 と、復号化のための暗号鍵を提供する暗号鍵提供部（提供部、暗号鍵提供手段）2 2 と、前記記憶装置 6 または記憶媒体が設置された外部記憶装置 9 からなる記憶部 2 3 と、複数の英数字や記号からなる文字列であるパスワードが入力インターフェイス 7 等を介して入力された際にこのパスワードを取得するパスワード取得部（取得部、パスワード取得手段）2 4 と、パスワード取得部 2 4 で取得されたパスワードから暗号鍵を生成する暗号鍵生成部（生成部、暗号鍵生成手段）2 6 と、該入力部 2 1 から入力された暗号文を、暗号鍵提供部 2 2 から提供される暗号鍵に基づいて復号化する復号化部（変換部、暗号化手段）2 7 と、該復号化部 2 7 で復号化されたデータである平文を出力する出力部 2 8 とを備えている。

30

【 0 0 2 0 】

暗号鍵生成部 2 6 は、自身で生成した暗号鍵を、記憶部 2 3 に記憶するか、或いは暗号鍵提供部 2 2 に直接渡す。暗号鍵提供部 2 6 は、上記のように暗号鍵生成部 2 6 から直接暗号鍵を受取るか、或いは、上記のようにして予め記憶部 2 3 に記憶された暗号鍵を読み込む。なお、暗号鍵提供部 2 2 が公開鍵等で暗号化された通信経路を介してネットワークインターフェイス 1 1 から暗号鍵を取得してよい。

【 0 0 2 1 】

本暗号処理システムでは、平文を暗号化して暗号文を作成する際に用いる暗号鍵と、該暗号文を復号化して平文を作成する際に用いる暗号鍵とが同一であり、このため、該暗号鍵は、共通鍵となる。

40

【 0 0 2 2 】

図 3 は、暗号化及び復号化の具体的手段を示す概念図である。

平文のデータ（例えば、図 3 の「A」）は、所定のデータ長を有する複数の平文分割データ（例えば、図 3 の「a 1」, 「a 2」, 「a 3」, 「a 4」）に分割される。平文を分割する数（分割数）N は、任意の値に設定できるが、好ましくは 1 1 以上に設定し、さらに好ましくは 1 2 以上に設定する。分割された各平文分割データ同士は、それぞれ異なるデータ長としてもよいし、或いは互いに同一データ長としてもよい。

【 0 0 2 3 】

50

同様に、暗号文のデータ（例えば、図3の「B」）は、所定のデータ長を有する複数の暗号文分割データ（例えば、図3の「b1」、「b2」、「b3」、「b4」）に分割される。暗号文の分割数Nも、任意の値に設定できるが、好ましくは11以上に設定し、さらに好ましくは12以上に設定する。分割された各暗号文分割データ同士は、それぞれ異なるデータ長としてもよいし、或いは互いに同一データ長としてもよい。

【0024】

ちなみに、本例では、平文を暗号化することにより暗号文と、該暗号文を復号化することにより得られる平文とは、共に、等分に分割され、分割数Nも同数に設定されている。すなわち、平文と暗号文のデータサイズは同一で、各分割データも互いに全て同一である。

10

【0025】

本暗号処理システムでは、全ての平文分割データを個別に格納するとともに、全ての暗号文分割データを個別に格納することが可能な数のノードnを有する木構造データT（同図の状態9参照）を規定する。そして、木構造データTにおいて、平文分割データを格納するノードnである平文用ノードna（例えば、図3の状態9の「a1」、「a2」、「a3」、「a4」）と、暗号文分割データを格納するノードnである暗号文用ノードnb（例えば、図3の状態9の「n1」、「n2」、「n3」）とは、一部で重複してもよいが、全てで重複してはならず、平文分割データ又は暗号文分割データの何れも格納されないノードも存在しないようにする。

【0026】

この木構造データTは、最上層に配置されたノードn（例えば、図3の状態9の「n3」）が「根ノード（ルートノード）」になり、下層のノードnである「子ノード」から上層のノードnである「親ノード」に延びるリンクLは必ず1つであり、「親ノード」となるノードnから「子ノード」となるノードnに延びるリンクLは必ず複数であり、これを言換えると、「子ノード」になるノードnと、該ノードnの「親ノード」になるノードnとの関係は、多対一になる。最下層に配置されたノードn（例えば、図3の状態9の「a1」、「a2」、「a3」、「a4」）が「葉ノード（リーフノード）」になる。

20

【0027】

あるノードnに着目すると、このノードnを「親ノード」とした場合における該ノードnの「子ノード」となる全てのノードnに格納されたデータに基づいて、この「親ノード」となるノードnに格納されるデータの情報が求められる。例えば、「親ノード」となるノードn（例えば、図3の状態9の「n1」）に対して、「子ノード」となる2つのノードn（例えば、図3の状態9の「a1」、「a2」）を考えた場合、「子ノード」となる一方のノードnに格納されたデータと、他方のノードnに格納されたデータとの差分を求め、この差分データを「親ノード」となるノードnに格納してもよい。

30

【0028】

また、「親ノード」となる1つのノードnに対して、「子ノード」となるノードnが3つ以上ある場合には、「子ノード」となる3つ以上のノードnから選択した2つのノードnの差分を求め、この差分と、残りの「子ノード」となるノードnから選択された1つのノードnとからさらに差分を求め、以下、「子ノード」となる全てのノードnが選択されるまで順次この処理を繰返して、「親ノード」となるノードnに格納するデータを求めてもよい。

40

【0029】

さらに、「親ノード」となるノードn及び該ノードnにリンクLで繋がれた「子ノード」となる全てのノードnに対し、これらのノードnの何れか1つが、格納されるデータが未知である「未知ノード」であるとともに、残りが、格納されるデータが既知である「既知ノード」である場合、この「既知ノード」となるノードnに格納された既知データから、「未知ノード」となるノードnに格納された未知データが求められれば、「子ノード」の格納データから「親ノード」に格納されるデータを算出する手段は、差分に限定されることもなく、その他の乗算や除算や減算を用いてもよい。

50

【0030】

そして、このようなリンク構造（ツリー構造）を有する木構造データTの全ノードnのうちから、各平文分割データを個別に格納するノードnを、平文用ノードnaとして、ランダムに選択するとともに、各暗号文分割データを個別に格納するノードnも、暗号文用ノードnbとして、ランダムに選択する。

【0031】

ただし、この平文用ノードnaのランダム選択にあたっては1つの条件が課せられる。具体的には、各平文用ノードnaに格納された平文分割データから、木構造データTの平文用ノードna以外の各ノードnに格納されたデータを求めることができるようにして、平文用ノードnaのランダム選択が行われる。図示する例では、「葉ノード」となる各ノードnが平文用ノードnaになっている。

10

【0032】

同様に、この暗号文用ノードnbを、木構造データTの全ノードnのうちからランダムに選択する際にも同一の条件が課せられる。具体的には、各暗号文用ノードnbに格納された暗号文分割データから、木構造データTの暗号文用ノードnb以外の各ノードnに格納されたデータを求めることができるようにして、暗号文用ノードnbのランダム選択が行われる。

【0033】

暗号鍵は、この木構造データの木構造及び配置情報を有し、この暗号鍵の情報に基づいてデータの暗号化及び復号化が行われる。

20

【0034】

以下、図3乃至図5に基づいて、上記木構造データのアルゴリズムを用いた暗号鍵生成の具体的手段を説明する。

【0035】

図4は、暗号化装置及び復号化装置の暗号生成部が行う暗号鍵生成処理のフロー図である。

暗号鍵生成部17, 26は、暗号鍵を生成する処理が開始されると、平文から分割した複数の平文分割データを各別に格納する複数のノードn（すなわち、平文用ノードna）を、それぞれ「未処理ノード」として選択範囲に加え、ステップS1に進む。ちなみに、このようにして「未処理ノード」としてセットされたノードnは、後述する処理（具体的には、ステップS6の処理）によって、順次、「処理済みノード」にセットされる。

30

【0036】

ステップS1では、選択範囲の中から、1つのノードnを「選択ノード」としてランダムに選択し、この選択したノードnが「処理済みノード」であれば、再度、選択範囲中からランダムに1つのノードnを「選択ノード」として選択し、以下、「未処理ノード」が選択されるまで、このステップS1の処理を繰返し、「未処理ノード」が「選択ノード」としてランダムに選択された時点で、該選択されたノードnを、左右一方側（図示する例では左側）に「子ノード」として配置し、ステップS2に進む。

【0037】

ステップS2では、選択範囲に含まれるノードnの中に、ステップS2でランダム選択したノードn以外で、「未処理ノード」が存在するか否かをチェックし、「未処理ノード」が存在すれば、ステップS3に進む。

40

【0038】

ステップS3では、選択範囲の中から、1つのノードnを「選択ノード」としてランダムに選択し、この選択されたノードnが「処理済みノード」である場合、或いは該ノードnがその回のループのステップS1で「選択ノード」とされている場合、再度、選択範囲中からランダムに1つのノードnを「選択ノード」として選択し、「未処理ノード」で且つその回のループのステップS1で「選択ノード」にされていないノードnを選択するまで、このステップS3の処理を繰返し、該条件を満たすノードnが「選択ノード」としてランダム選択された時点で、該選択されたノードnを、左右他方側（図示する例では右側）

50

に「子ノード」として配置し、ステップS 4に進む。

【0039】

ステップS 4では、そのループ処理のステップS 1及びステップS 3で選んだ2つのノードnを、「子ノード」に有する「親ノード」と、「子ノード」となる一对のノードnからこの「親ノード」となるノードnに繋がるリンクLを生成し、ステップS 5に進む。ちなみに、「親ノード」となるノードnに格納されるデータは、後述したように全ての子ノードnから求められるものであれば何れでもよいが、ここでは、「子ノード」となる一对のノードnの差分データを、「親ノード」となるノードnに格納する。

【0040】

ステップS 5では、その回のループ処理でステップS 1及びステップS 3で「子ノード」として選択した一对のノードnがそれぞれ「葉ノード」であるか否かを確認し、両方とも「葉ノード」であれば、ステップS 6に進む。

10

【0041】

ステップS 6では、その回のループ処理で「子ノード」とした一对のノードn及び「親ノード」とした1つのノードnのうちからランダムに選択された2つのノードnを、「対象ノード」とするとともに、残りの1つのノードnを、暗号文分割データが格納されない「非対象ノード」として、ステップS 7に進む。ちなみに、この「対象ノード」となるノードnは、上述した暗号文用ノードnbである。

【0042】

ステップS 7では、その回のループ処理でステップS 1及びステップS 3で「子ノード」としてランダム選択した2つの各ノードnを、「処理済みノード」にセットするとともに、その回のループのステップS 4で「親ノード」として生成したノードnを、「未処理ノード」にセットして上述した選択範囲に追加し、その回のループ処理を終了させ、ステップS 1に処理を戻して、次の回のループ処理に移行する。

20

【0043】

ステップS 5において、その回のループ処理で「子ノード」とした一对のノードnの一方が「葉ノード」であり、他方が「葉ノード」でない場合、ステップS 8に進む。ステップS 8では、上記「葉ノード」且つ「子ノード」であるノードnと、その回のループ処理で生成された「親ノード」であるノードnとの何れか一方を、ランダムに選択して「対象ノード」とするとともに、他方を、上述した「非対象ノード」として、ステップS 7に進む。

30

【0044】

ステップS 5において、その回のループ処理で「子ノード」として選択した一对のノードnが何れも「葉ノード」でない場合には、ステップS 9に進む。ステップS 9では、「親ノード」を「非対象ノード」とし、ステップS 7に進む。

【0045】

また、ステップS 2において、選択範囲中のノードnが全て「処理済みノード」である場合には、ステップS 10に進む。ステップS 10では、その回のループ処理で、ステップS 1で「子ノード」として選択されたノードnを「根ノード」とするとともに、該ノードnを、「処理済みノード」にセットし、これによって木構造データTの木構造及び配置情報を格納された暗号鍵が生成され、以上をもって暗号鍵を生成する処理を終了させる。

40

【0046】

以上のようにして、上述した木構造及び配置の条件を満たす木構造データTがランダムに生成され、この木構造及び配置情報が暗号鍵に格納され、これらの情報に基づいて、データの暗号化及びデータの復号化が行われる。

【0047】

ちなみに、図3の状態1は、ステップS 1 ステップS 2 ステップS 3の処理に対応し、状態2は、ステップS 4の処理に対応し、状態3は、ステップS 5 ステップS 6 ステップS 7の処理に対応し、状態4は、ステップS 1 ステップS 2 ステップS 3の処理に対応し、状態5は、ステップS 4の処理に対応し、状態6は、ステップS 5 ステ

50

ステップ S 8 ステップ S 7 の処理に対応し、状態 7 は、ステップ S 1 ステップ S 2 ステップ S 3 の処理に対応し、状態 8 は、ステップ S 4 の処理に対応し、状態 9 は、ステップ S 5 ステップ S 8 ステップ S 7 ステップ S 1 ステップ S 2 ステップ S 10 の処理に対応している。

【 0 0 4 8 】

なお、上記したステップ S 1、ステップ S 3、ステップ S 6 及びステップ S 8 の処理（決定処理）によって、木構造データ T の木構造及び配置状態をランダムに選択して決定する際、本例では擬似乱数を用いて再現可能にランダム選択を行う。

【 0 0 4 9 】

この擬似乱数の生成には、従来公知の種々の手段を用いることができるが、ここでは、10
フィボナッチ L F S R (L i n e a r F e e d b a c k S h i f t R e g i s t e r) からなる擬似乱数生成手段 29 (図 5 参照) を用いる。擬似乱数生成手段 29 は R A M 4 上
に実行される前記プログラムによって実装される他、上記フィボナッチ L F S R については従来公知であるため、詳細は割愛するが、その構成を以下に簡単に説明する。

【 0 0 5 0 】

図 5 は、擬似乱数生成手段の一例の構成を示す概念図である。生成する乱数は、予め定めた所定（図示する例では M ビットであり、さらに具体的には 63 ビット）ビット長のデータサイズを有し、ある状態からその次の状態への移行（以下、「次状態への移行」）にあたっては、各ビット X の値が、隣接する最終のビット X（図 5 における最右端の M 番目のビット X）側のビット X に 1 つずつずれるとともに、全ビット X から予め選定された複20
数の各ビット（タップ） X s の排他的論理和が順次求められ、その最終結果が最初のビット X（図 5 における最左端の 1 番目のビット X）にフィードバックされる。このようにして、周期の長い再現性のある乱数が順次擬似的に生成されていく。

【 0 0 5 1 】

なお、このフィボナッチ L F S R では、初期値 R の決定も非常に重要になるが、この初期値 R を、パスワード取得手段 16, 24 で取得されるパスワード及び上述した分割数 N の一方又は両方（図示する例では両方）に基づいて決定する。

【 0 0 5 2 】

これに加えて、タップ X s の選定や、排他的論理和を求める X O R ゲート 31 a からなる論理回路 31 の構成を、パスワード取得手段 16, 24 で取得されるパスワード及び上述した分割数 N の一方又は両方に依存させてもよい。30

【 0 0 5 3 】

そして、この擬似乱数生成手段 29 は、ランダム選択毎に「次状態への移行」の処理を繰返して、擬似的に乱数を生成する処理（発生処理）を行い、このように生成した乱数を用いてランダム選択処理（決定処理）を行う。例えば、ステップ S 1 ステップ S 2 ステップ S 3 ステップ S 4 ステップ S 5 ステップ S 6 ステップ S 7 と処理を進む場合、ステップ S 1、ステップ S 3 及びステップ S 6 の計 3 回、ランダム選択が行われ、このランダム選択の処理毎に「次状態への移行」の処理が行われる。

【 0 0 5 4 】

ちなみに、このランダム選択毎に行われる「次状態への移行」の処理は、必ずしも 1 回40
づつである必要はなく、予め定めた所定回数実行して、最後の「次状態への移行」の処理により擬似生成される乱数を、ランダム選択に用いてもよい。このようにランダム選択に用いない擬似乱数生成を行う「空回し処理」によって、より解読され難い乱数を擬似的に生成することが可能になる。

【 0 0 5 5 】

ランダム選択の際、このように再現可能な擬似乱数を用いた場合、同一のパスワードを用いることにより同一の暗号鍵（具体的には、木構造データの同一の木構造及び配置情報）を暗号化装置 1 及び復号化装置 2 の両方で生成できる。このため、暗号化装置 1 において上記パスワードを入力するのみによって、暗号化のための暗号鍵を生成可能であるとともに、復号化装置 2 において上記パスワードを入力するのみによって、復号化のための暗50

号鍵を生成可能であり、利便性が高い。

【 0 0 5 6 】

次に、図 6 及び図 7 に基づいて、暗号化部 1 8 の構成を詳述する。

【 0 0 5 7 】

図 6 は、暗号化部が暗号鍵に基づいて行う平文の暗号化処理の手順を示す説明図であるとともに、復号化部が暗号鍵に基づいて行う暗号文の復号化処理の手順を示す説明図であり、図 7 は、暗号化部の処理フロー図である。暗号化部 1 8 は、入力部 1 2 から入力された平文を受取ると、ステップ S 1 1 から処理を開始する。ステップ S 1 1 では、上記分割数 N 個に平文を分割して、平文分割データを生成する分割処理を行い、処理が終了すると、ステップ S 1 2 に進む。

10

【 0 0 5 8 】

ステップ S 1 2 では、暗号鍵提供部 1 3 から提供される暗号鍵を受取り、この暗号鍵から、木構造データ T のリンク情報及び配置情報を取得し、まず、各平文分割データを、木構造データ T の平文用ノード n a に格納し（図 6 の「暗号化処理 1」）、この各平文用ノード n a に格納されたデータから、残りのノード n に格納されたデータを求め、上記暗号鍵から、木構造データ T の各暗号文用ノード n b に格納されたデータを、暗号文分割データとして取得する（同図の「暗号化処理 2」）。このステップ S 1 2 の処理が暗号化のためのメイン処理になり、このメイン処理が終了すると、ステップ S 1 3 に進む。

【 0 0 5 9 】

ステップ S 1 3 では、直前のステップ S 1 2 で取得した各暗号文分割データを結合させて暗号文を生成する結合処理を行い、処理を終了させる。

20

【 0 0 6 0 】

次に、図 6 及び図 8 に基づいて、復号化部 2 7 の構成を詳述する。

【 0 0 6 1 】

図 8 は、復号化部の処理フロー図である。復号化部 2 7 は、入力部 1 2 から入力された暗号文を受取ると、ステップ S 2 1 から処理を開始する。ステップ S 2 1 では、上記分割数 N 個に暗号文を分割して、暗号文分割データを生成する分割処理を行い、分割処理が終了すると、ステップ S 2 2 に進む。

【 0 0 6 2 】

ステップ S 2 2 では、暗号鍵提供部 1 3 から提供される暗号鍵を受取り、この暗号鍵から、木構造データ T のリンク情報及び配置情報を取得し、まず、各暗号文分割データを、木構造データ T の暗号文用ノード n b に格納し（図 6 の「復号化処理 2」）、この各暗号文用ノード n b に格納されたデータから、残りのノード n に格納されたデータを求め、上記暗号鍵から、木構造データ T の各平文用ノード n a に格納されたデータを、平文分割データとして取得する（同図の「復号化処理 2」）。このステップ S 2 2 の処理が復号化のためのメイン処理になり、このメイン処理が終了すると、ステップ S 2 3 に進む。

30

【 0 0 6 3 】

ステップ S 2 3 では、直前のステップ S 2 2 で取得した各平文分割データを結合させて平文を生成する結合処理を行い、処理を終了させる。

【 0 0 6 4 】

以上のように構成される暗号処理システムによれば、木構造データ T のリンク情報及び配置状態を、暗号鍵によって取得できれば、少ない処理工程で、迅速に暗号化及び復号化を行うことができる一方で、上記リンク情報及び配置情報を取得できなければ、暗号文の復号化は困難であり、処理の迅速化と、暗号強度の向上とを両立できる。

40

【 0 0 6 5 】

しかも、擬似乱数によって、分割数 N やパスワードによる再現性を持たせて暗号鍵を生成するため、暗号鍵をデータとして、保持しておく必要がなく、利便性の暗号強度も両立可能になる。

【 0 0 6 6 】

なお、擬似乱数生成手段 2 9 による乱数生成の手段として、本例では、フィボナッチ L

50

F S Rを用いているが、これに限定されるものではなく、ガロア L F S R や、その他の擬似乱数を生成する公知の手段を用いてもよい。

【 0 0 6 7 】

また、パスワードによる再現性を持たせることなく、暗号鍵を生成する場合には、擬似乱数ではなく、ランダム選択による決定処理時に真正乱数を用いて、木構造データ T を生成し、これに基づいて暗号鍵を生成する生成処理を行ってもよい。この場合には、暗号化及び復号化にあたって、暗号鍵をデータとして保持する必要はあるが、暗号文を解読される可能性はさらに低くなる。

【 0 0 6 8 】

さらに、木構造データ T の構成も図 3 及び図 6 に示すような構成に限定されない。

10

【 0 0 6 9 】

図 9 (A) , (B) は、それぞれ木構造データの他例を示す概念図である。木構造データ T は、「リーフノード」以外のノード n に平文分割データを配置してもよく(同図(A)参照)、さらには、「親ノード」となるノード n が「子ノード」となるノード n を 3 個以上有していてもよい(同図(B)参照)。

【 0 0 7 0 】

次に、図 1 0 及び図 1 1 に基づき、本暗号処理システムの検証結果について説明する。

【 0 0 7 1 】

図 1 0 (A) 乃至(E) は、それぞれ本暗号処理システムの検証実験の結果を示す一覧表である。

20

まず、同図(A)の一覧表に示す検証では、木構造データ T の木構造を総当りにより解析した。具体的には、パスワードが 5 桁で英数字が混在するもの(同表の「 A 」で示すもの)と、英数字が混在するもの(同表の「 B 」で示すもの)と、英字のみのもの(同表の「 C 」で示すもの)と、数字のみのもの(同表の「 D 」で示すもの)と、英数字が混在するもの(同表の「 E 」で示すもの)とのそれぞれに対して、分割数 N を 5 ~ 1 0 の範囲で 1 つずつ変化させ、各分割数 N に対して、総当りにより、木構造データ T の木構造の解析を行った。

【 0 0 7 2 】

ちなみに、用いるコンピュータは、本願の出願時点で高性能される A T 互換機のコンピュータを用い、初期値 R は、パスワードを用いて以下の式から求め、求めた値を、X ビット(具体的には 6 3 ビット)の 2 進数表示して用いる。

30

【 0 0 7 3 】

【数 1】

$$R = N + 1 \times PW[0] + 2 \times PW[1] + \dots + (m - 1) \times PW[m - 2] + m \times PW[m - 1]$$

【 0 0 7 4 】

N は上述した分割数であり、m はパスワードの文字数であり、PW [i - 1] は、パスワードにおける i 番目の文字の文字コードである。

【 0 0 7 5 】

その結果は、同表に示す通りであり、解析に要する時間及び回数は、分割数 N の上昇に伴って飛躍的に上昇している。そして、同表には示していないが、分割数 N が 1 1 になると、解析が困難になり、分割数 N が 1 2 になると、解析に要する時間が 3 5 年以上となるという結果になり、事実上、解析が不可能になった。

40

【 0 0 7 6 】

続いて、図 1 0 (B) の一覧表に示す検証では、乱数再現により、木構造データ T の木構造の解析を行った。その他の条件は、同図(A)と同様である。

【 0 0 7 7 】

その結果は、同表に示す通りであり、上述したノード n の総当りによる解析によりも少ない時間や回数で解析を行うことができる場合が多くあり、全体的には、解析に要する時間や回数は減少し、効率的な解析が行われている。ちなみに、パスワード自体の解読は、

50

乱数の再現に帰着されるので、乱数発生時に乱数値に基づいて上述の「空回し処理」を行うことにより、再現されるリスクを大幅に低減することが可能になる。その詳細は、後述する。

【0078】

続いて、図10(C)の一覧表に示す検証では、木構造データTのリンク構造(木構造)が既知であると仮定し、総当りにより、暗号文用ノードnbを特定する解析を行った。用いるパスワードの種類は、同図(A)に示すものと同様であり、分割数Nを8 12 16 20 21 22と変化させ、各分割数Nに対して、総当りにより、暗号文用ノードnbを特定する解析を行った。

【0079】

その結果は、同表に示す通りであり、分割数Nが少ない場合には、少ない時間及び回数が解析可能であったが、分割数Nが増加ほど解析は困難になっている。そして、分割数Nが16を超えたあたりから、解析の難易度が極端に上がり、同表には示していないが、分割数Nが26に達した段階で、解析は不可能になった。

【0080】

続いて、図10(D)の一覧表に示す検証では、木構造データTのリンク構造(木構造)が既知であると仮定し、乱数再現により、分割用ノードnbを特定する解析を行った。その他の条件は、同図(C)に示すものと同様である。

【0081】

その結果は、同表に示す通りであり、同図(C)に示す総当り解析に比べて、概ね解析に要する時間や回数が減少している。

【0082】

続いて、同図(E)の一覧表に示す検証では、木構造データTの木構造を総当りにより解析した。ただし、木構造データTの生成にあたっては、擬似乱数生成手段29により「次状態への移行」の処理を行う際、上述した処理に加えて、乱数発生回数がj回である場合、パスワードのj番目の文字コードをMビットの2進数表示して、各ビットXに加え、これを擬似的に生成した乱数として用い、上述した「空回し処理」を行う。ちなみに、jは、パスワードの文字数以上にはならず、パスワードの文字数を超えた場合には、その値が1に戻される。その他の条件は同図(A)と同様である。

【0083】

その結果は、同表に示す通りであり、解析に要する時間や回数は、同図(A)に示す総当り解析と同程度になっている。すなわち、総当り解析に対しては、「空回し処理」の効果は殆ど見られないことが分った。一方、この木構造データTの木構造の解析を、乱数再現により行うと、分割数Nが4の段階で、解析が困難になり、この「空回し処理」は、乱数再現による解析に対して、有効であることが分った。

【0084】

図11(A)は暗号化する際の処理速度を示し、(B)は復号化する際の処理速度を示している。同図(A)は、分割数Nが80と、96と、112と、128のそれぞれの場合において、平文のデータサイズを1MB 4MB 16MB 32MB 64MB 128MBと変化させ、この平文を暗号鍵により暗号化する処理に要す時間から、処理速度を求め、これをグラフ化したものである。暗号化装置1としては、上述の検証と同一のコンピュータを用いた。

【0085】

このグラフに示されるように、概ね1700Mbit/secの処理速度が確保され、実用に耐え得る良好な結果となった。

【0086】

また、同図(B)は、分割数Nが80と、96と、112と、128のそれぞれの場合において、暗号文のデータサイズを1MB 4MB 16MB 32MB 64MB 128MBと変化させ、この暗号文を暗号鍵により復号化する処理に要す時間から、処理速度を求め、これをグラフ化したものである。復号化装置2としては、上述の検証と同一の

10

20

30

40

50

コンピュータを用いた。

【0087】

このグラフに示されるように、概ね1450Mbit/secの処理速度が確保され、実用に耐え得る良好な結果となった。

【符号の説明】

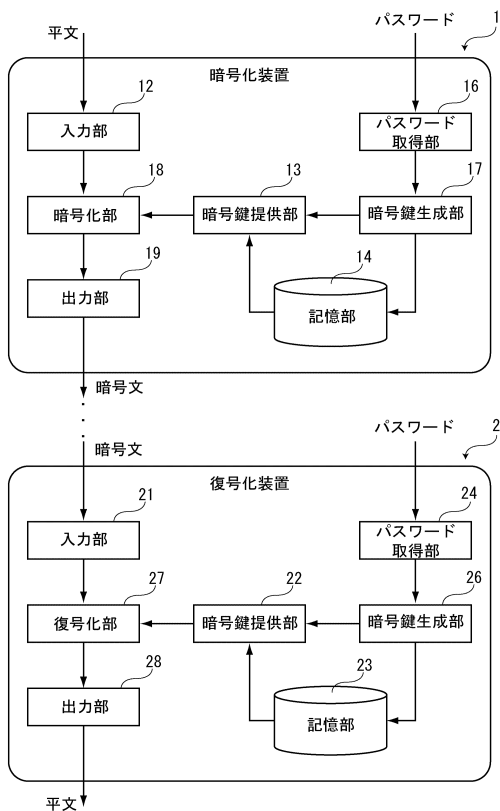
【0088】

- 12 入力部（入力手段）
- 13 暗号鍵提供部（提供部，暗号鍵提供手段）
- 16 パスワード取得部（取得部，パスワード取得手段）
- 17 暗号鍵生成部（生成部，暗号鍵生成手段）
- 18 暗号化部（変換部，暗号化手段）
- 19 出力部（出力手段）
- 21 入力部（入力手段）
- 22 暗号鍵提供部（提供部，暗号鍵提供手段）
- 24 パスワード取得部（取得部，パスワード取得手段）
- 26 暗号鍵生成部（生成部，暗号鍵生成手段）
- 27 暗号化部（変換部，暗号化手段）
- 28 出力部（出力手段）
- n ノード
- T 木構造データ

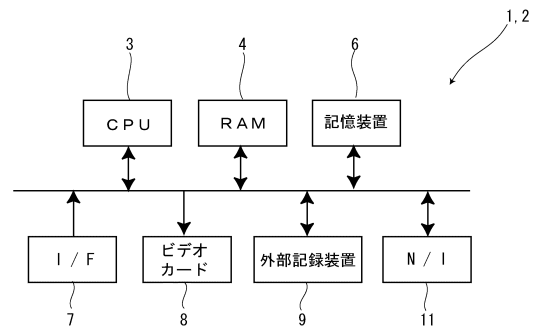
10

20

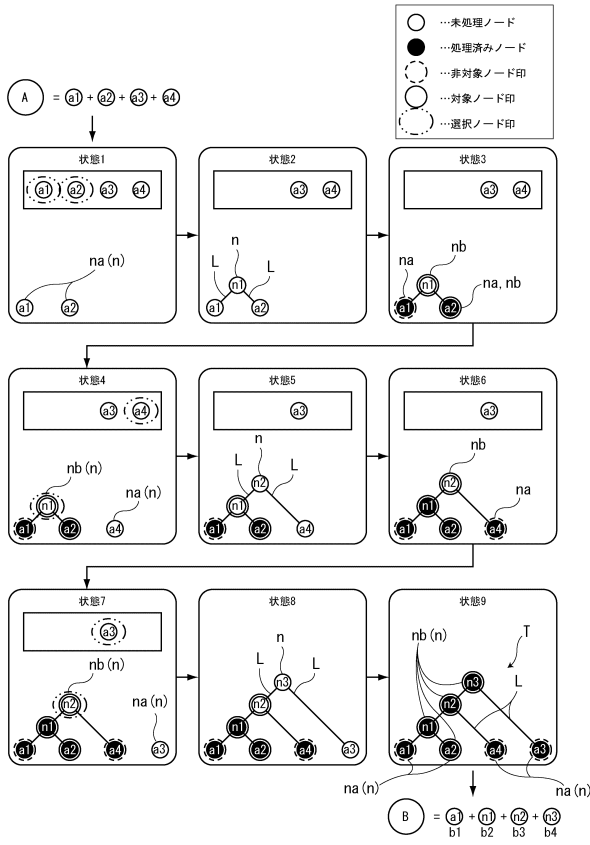
【図1】



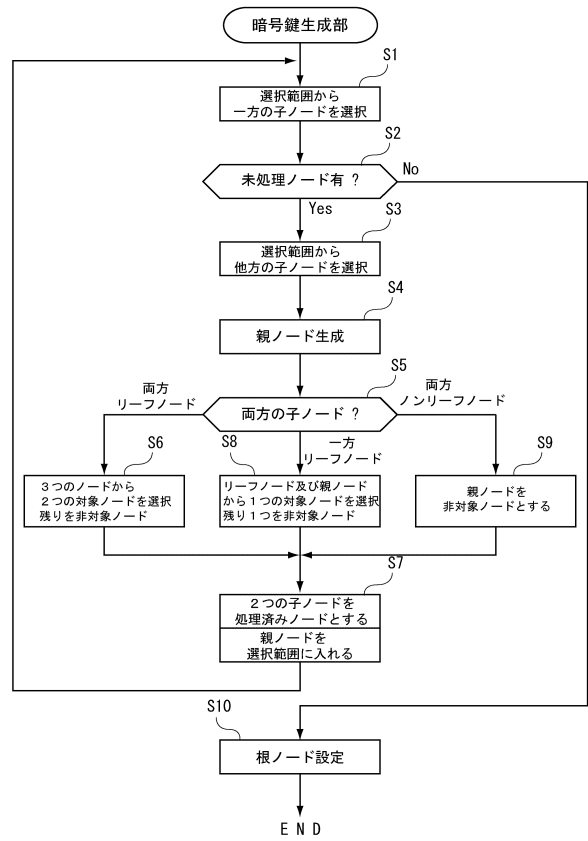
【図2】



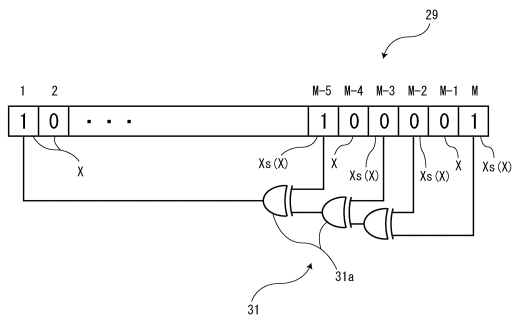
【図3】



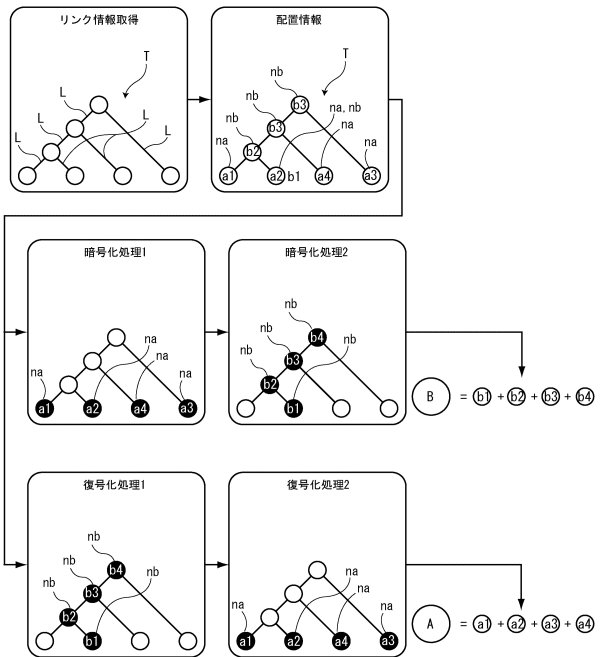
【図4】



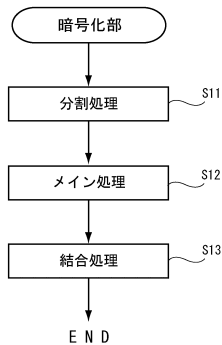
【図5】



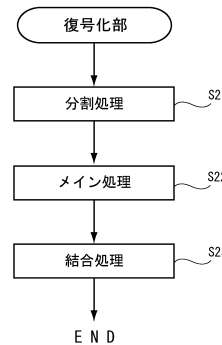
【図6】



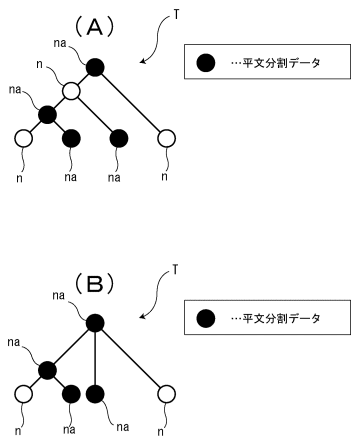
【図7】



【図8】



【図9】



【図10】

(A)

分割数	5	6	7	8	9	10
A 評価回数	157	48,394	1,022,977	118,242,255	13,861,656,785	26,561,058,487
A 時間 (sec)	0.000	0.000	0.078	10.855	1388.435	2,971.041
B 評価回数	1,805	41,499	2,677,833	161,333,616	5,850,768,008	911,613,071,977
B 時間 (sec)	0.000	0.000	0.260	16.079	661.613	112,687.470
C 評価回数	1,112	10,810	777,693	178,836,653	9,110,674,918	1,124,547,911,542
C 時間 (sec)	0.000	0.000	0.078	18.489	1038.369	137,950.370
D 評価回数	2,593	44,749	2,605,645	126,977,537	5,971,190,401	615,737,161,153
D 時間 (sec)	0.000	0.000	0.218	12.964	685.262	76255.087
E 評価回数	2,275	50,977	2,491,738	80,462,504	3,674,560,139	474,010,049,377
E 時間 (sec)	0.000	0.000	0.234	7.925	417.383	6553.988
平均評価回数	1,588	39,286	1,895,177	133,030,533	7,993,770,050	630,493,800,507
平均時間 (sec)	0.000	0.000	0.172	13.182	832.208	77,677.511
組み合わせ数	2,880	86,400	3,628,800	203,212,800	14,631,321,600	1,316,818,944,000

(B)

分割数	5	6	7	8	9	10
A 評価回数	9,710	456,393	18,497	67,256,489	1,155,934	N/A
A 時間 (sec)	0.016	0.421	0.015	88.705	1.654	N/A
B 評価回数	2,218	10,350,946	271,927,305	2,919,335,317	N/A	N/A
B 時間 (sec)	0.000	9.625	326.744	4,064.244	N/A	N/A
C 評価回数	13,314	2,162,927	261,122	38,010,926	335,551,877	2,163,389,569
C 時間 (sec)	0.031	1.997	0.281	51.028	510.448	3,585.760
D 評価回数	1,146	136,078	92,285	2,694,888	896,104	20,321,641
D 時間 (sec)	0.000	0.125	0.047	3.245	1.248	33.134
E 評価回数	27	679	33,927	367,385	447,801,835	12,244,665
E 時間 (sec)	0.000	0.000	0.047	0.437	683.406	19.796
平均評価回数	5,083	2,625,265	54,454,628	606,530,601	190,351,433	728,651,958
平均時間 (sec)	0.009	2.434	65.427	841.132	299.130	1,212.897

(C)

分割数	8	12	16	20	21	22
A 評価回数	3,579	1,333,331	227,079,204	61,387,183,559	292,425,795,930	807,226,342,710
A 時間 (sec)	0.000	0.047	6.755	2,072.464	9,361.670	27,827.656
B 評価回数	5,138	1,331,647	202,278,727	52,667,127,315	239,893,370,817	978,106,546,287
B 時間 (sec)	0.000	0.062	7.784	2,297.213	10,795.141	64,930.066
C 評価回数	1,710	1,123,747	190,697,993	65,509,105,832	112,297,321,961	336,880,896,615
C 時間 (sec)	0.000	0.047	7.472	3,104.671	4,765.169	14,214.683
D 評価回数	4,273	692,748	163,145,433	67,783,576,635	44,216,255,031	1,023,612,452,821
D 時間 (sec)	0.000	0.031	6.443	3,169.320	1,792.989	28,679.121
E 評価回数	6,145	1,140,281	300,294,212	55,285,701,374	305,230,493,078	866,246,538,861
E 時間 (sec)	0.000	0.047	13.307	2,998.270	9,185.653	39,390.638
平均評価回数	4,129	1,124,771	214,699,100	61,126,538,943	156,042,466,500	742,249,894,099
平均時間 (sec)	0.000	0.047	8.352	2,608.428	6,276.371	34,094.614
組み合わせ数	6,435	1,352,078	300,540,195	68,923,264,410	369,138,937,220	1,052,040,481,860

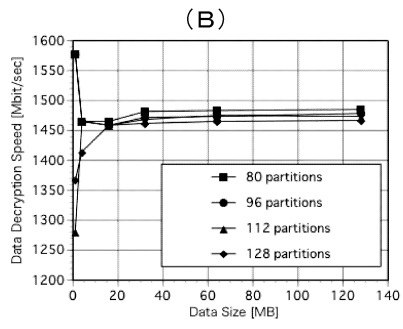
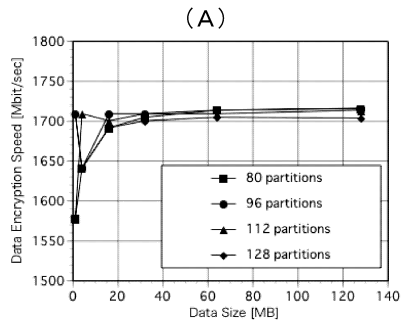
(D)

分割数	8	12	16	20	21	22
A 評価回数	8,809	8,809	10,801,735	10,800,470	10,801,735	1,943,815
A 時間 (sec)	0.000	0.000	8.388	7.472	6.377	1.030
B 評価回数	13	164	39,223,460	2,162,852	1,298,685,613	1,207,984,292
B 時間 (sec)	0.000	0.000	31.715	1.700	1,167.303	1,210.952
C 評価回数	40	1,126	1,126	N/A	N/A	N/A
C 時間 (sec)	0.000	0.000	0.000	N/A	N/A	N/A
D 評価回数	1,127	18,535	1,127	268,585,063	2,286,945,383	2,286,945,383
D 時間 (sec)	0.000	0.016	0.000	281.331	2,282.567	2,307.868
E 評価回数	2	2,307	2	4,007,372,931	1,293,947,010	4,407,372,931
E 時間 (sec)	0.000	0.000	0.000	40,056.345	1,292.715	42,713.229
平均評価回数	1,998	6,188	10,005,490	1,072,155,829	1,200,093,685	1,975,911,605
平均時間 (sec)	0.000	0.003	8.022	10,086.712	1,152.741	11,564.445

(E)

分割数	5	6	7	8	9	10
A 評価回数	1,889	8,838	1,510,129	125,624,267	9,667,938,084	509,077,892,305
A 時間 (sec)	0.000	0.000	0.156	12.527	1,064.562	60,397.597
B 評価回数	1,832	69,273	3,190,388	106,140,872	2,803,787,897	520,446,474,577
B 時間 (sec)	0.000	0.000	0.350	12.074	336.461	64,810.392
C 評価回数	1,576	35,106	1,351,054	75,465,004	13,805,582,080	296,953,207,342
C 時間 (sec)	0.000	0.000	0.140	7.613	1,640.078	37,922.668
D 評価回数	1,252	33,968	795,637	59,610,070	4,598,050,916	280,997,096,425
D 時間 (sec)	0.000	0.016	0.078	5.534	544.784	361,115.923
E 評価回数	2,628	36,142	1,006,165	14,990,535	13,175,101,683	794,645,277,010
E 時間 (sec)	0.000	0.016	0.094	14.118	1,438.853	88,864.854
平均評価回数	1,548	50,688	1,308,901	85,138,123	7,341,246,870	383,886,642,945
平均時間 (sec)	0.000	0.000	0.163	6.684	635.760	45,013.239
組み合わせ数	2,880	86,400	3,628,800	203,212,800	14,631,321,600	1,316,818,944,000

【 図 11 】



フロントページの続き

(56)参考文献 特開平09 - 200196 (JP, A)
特開平11 - 187013 (JP, A)
特開2006 - 279644 (JP, A)

(58)調査した分野(Int.Cl., DB名)

H04L 9/06

G09C 1/00

JSTPlus/JMEDPlus/JST7580(JDreamIII)