(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2014-219548
(P2014-219548A)
(43)公開日　平成26年11月20日(2014.11.20)

(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　請求項の数 5　ＯＬ　（全 16 頁）

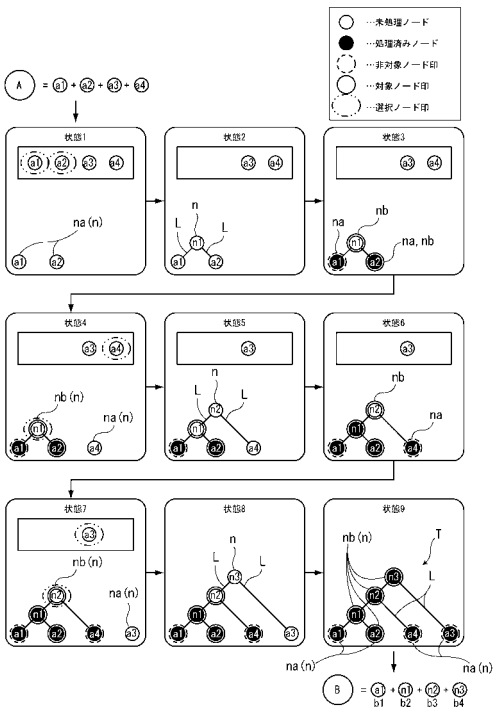
(21)出願番号　特願2013-98521(P2013-98521)
(22)出願日　平成25年5月8日(2013.5.8)

(71)出願人　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

Ｆターム(参考)　5J104 AA18　JA03　JA13　PA07

(54)【発明の名称】暗号処理システム

(57)【要約】
【課題】　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に、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。
【解決手段】　暗号化又は復号化に用いる暗号鍵は、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、暗号文を上記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ｎとして配置可能な木構造データＴ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Ｔの対応するノードｎ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む。
【選択図】　図３



【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む
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
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
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、
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

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ＤＥＳ（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】
【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に、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】
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むことを特徴とする。
【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。
【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。

【００１０】

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、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】

【００１１】

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】

【００１２】

【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。

【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。

【図１０】（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。

【発明を実施するための形態】

【００１３】

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。

【００１４】

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、ＡＴ互換機と呼ばれる通常のＰＣであり、ＣＰＵ３と、ＲＡＭ４と、ＳＳＤやＨＤＤ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ＣＤ－Ｒ等の記憶媒体からデータを読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ＰＣとＴＣＰ／ＩＰ通信可能なネットワークインターフェイス１１とを備えている。

【００１５】
上記ＲＡＭ４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及び復号化装置２として機能する。
【００１６】
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）１２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）１３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１４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）１６と、パスワード取得部１６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）１７と、該入力部１２から入力された平文を、暗号鍵提供部１３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）１８と、該暗号化部１８で暗号化されたデータである暗号文を出力する出力部１９とを備えている。
【００１７】
入力部１２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１８に渡される。
【００１８】
暗号鍵生成部１７は、自身で生成した暗号鍵を、記憶部１４に記憶するか、或いは暗号鍵提供部１３に直接渡す。暗号鍵提供部１３は、上記のように暗号鍵生成部１７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１４に記憶された暗号鍵を読込む。なお、暗号鍵提供部１３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１１から暗号鍵を取得してもよい。
【００１９】
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）２１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）２２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２３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）２４と、パスワード取得部２４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）２６と、該入力部２１から入力された暗号文を、暗号鍵提供部２２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）２７と、該復号化部２７で復号化されたデータである平文を出力する出力部２８とを備えている。
【００２０】
暗号鍵生成部２６は、自身で生成した暗号鍵を、記憶部２３に記憶するか、或いは暗号鍵提供部２２に直接渡す。暗号鍵提供部２６は、上記のように暗号鍵生成部２６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２３に記憶された暗号鍵を読込む。なお、暗号鍵提供部２２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１１から暗号鍵を取得してよい。
【００２１】
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。
【００２２】
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「ａ１」，「ａ２」，「ａ３」，「ａ４」）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。
【００２３】
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。
【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「ａ２」，「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。
【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。
【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。
【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。

【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。

【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。

【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。

【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。

【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）に「子ノード」として配置し、ステップＳ４に進む。

【００３９】

ステップＳ４では、そのループ処理のステップＳ１及びステップＳ３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」となる一対のノード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Ｓ５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」となる一対のノード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Ｓ５では、その回のループ処理でステップＳ１及びステップＳ３で「子ノード」として選択した一対のノード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Ｓ６に進む。

【００４１】

ステップＳ６では、その回のループ処理で「子ノード」」とした一対のノード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Ｓ７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Ｓ７では、その回のループ処理でステップＳ１及びステップＳ３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Ｓ４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Ｓ１に処理を戻して、次の回のループ処理に移行する。

【００４３】

ステップＳ５において、その回のループ処理で「子ノード」とした一対のノード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Ｓ８に進む。ステップＳ８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Ｓ７に進む。

【００４４】

ステップＳ５において、その回のループ処理で「子ノード」として選択した一対のノードｎが何れも「葉ノード」でない場合には、ステップＳ９に進む。ステップＳ９では、「親ノード」を「非対象ノード」とし、ステップＳ７に進む。

【００４５】

また、ステップＳ２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Ｓ１０に進む。ステップＳ１０では、その回のループ処理で、ステップＳ１で「子ノード」として選択されたノードｎを「根ノード」にするとともに、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。

【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Ｓ１→ステップＳ２→ステップＳ３の処理に対応し、状態２は、ステップＳ４の処理に対応し、状態３は、ステップＳ５→ステップＳ６→

ステップＳ７の処理に対応し、状態４は、ステップＳ１→ステップＳ２→ステップＳ３の処理に対応し、状態５は、ステップＳ４の処理に対応し、状態６は、ステップＳ５→ステップＳ８→ステップＳ７の処理に対応し、状態７は、ステップＳ１→ステップＳ２→ステップＳ３の処理に対応し、状態８は、ステップＳ４の処理に対応し、状態９は、ステップＳ５→ステップＳ８→ステップＳ７→ステップＳ１→ステップＳ２→ステップＳ１０の処理に対応している。

【００４８】

なお、上記したステップＳ１、ステップＳ３、ステップＳ６及びステップＳ８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。

【００４９】

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、フィボナッチＬＦＳＲ（Ｌｉｎｅａｒ　Ｆｅｅｄｂａｃｋ　Ｓｈｉｆｔ　Ｒｅｇｉｓｔｅｒ）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ＲＡＭ４上に実行される前記プロフラムによって実装される他、上記フィボナッチＬＦＳＲ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。

【００５０】

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つづつずれるとともに、全ビットＸから予め選定された複数の各ビット（タップ）Ｘｓ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。

【００５１】

なお、このフィボナッチＬＦＳＲ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。

【００５２】

これに加えて、タップＸｓの選定や、排他的論理和を求めるＸＯＲゲート３１ａ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。

【００５３】

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Ｓ１→ステップＳ２→ステップＳ３→ステップＳ４→ステップＳ５→ステップＳ６→ステップＳ７と処理を進む場合、ステップＳ１、ステップＳ３及びステップＳ６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。

【００５４】

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。

【００５５】

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、復号化のための暗号鍵を生成可能であり、利便性が高い。

【００５６】

次に、図６及び図７に基づいて、暗号化部１８の構成を詳述する。

【００５７】

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１８は、入力部１２から入力された平文を受取ると、ステップＳ１１から処理を開始する。ステップＳ１１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Ｓ１２に進む。

【００５８】

ステップＳ１２では、暗号鍵提供部１３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ｎａに格納し（図６の「暗号化処理１」）、この各平文用ノードｎａ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ｎｂ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Ｓ１２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Ｓ１３に進む。

【００５９】

ステップＳ１３では、直前のステップＳ１２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。

【００６０】

次に、図６及び図８に基づいて、復号化部２７の構成を詳述する。

【００６１】

図８は、復号化部の処理フロー図である。復号化部２７は、入力部１２から入力された暗号文を受取ると、ステップＳ２１から処理を開始する。ステップＳ２１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Ｓ２２に進む。

【００６２】

ステップＳ２２では、暗号鍵提供部１３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ｎｂに格納し（図６の「復号化処理２」）、この各暗号文用ノードｎｂ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ｎａ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Ｓ２２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Ｓ２３に進む。

【００６３】

ステップＳ２３では、直前のステップＳ２２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。

【００６４】

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。

【００６５】

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。

【００６６】
なお、擬似乱数生成手段２９による乱数生成の手段として、本例では、フィボナッチＬＦＳＲを用いているが、これに限定されるものではなく、ガロアＬＦＳＲ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。
【００６７】
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。
【００６８】
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。
【００６９】
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）。
【００７０】
次に、図１０及び図１１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。
【００７１】
図１０（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。
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５～１０の範囲で１つづつ変化させ、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。
【００７２】
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ＡＴ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６３ビット）の２進数表示して用いる。
【００７３】
【数１】

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

【００７４】
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。
【００７５】
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１１になると、解析が困難になり、分割数Ｎが１２になると、解析に要する時間が３５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。
【００７６】
続いて、図１０（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。
【００７７】
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、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→１

２８ＭＢ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】

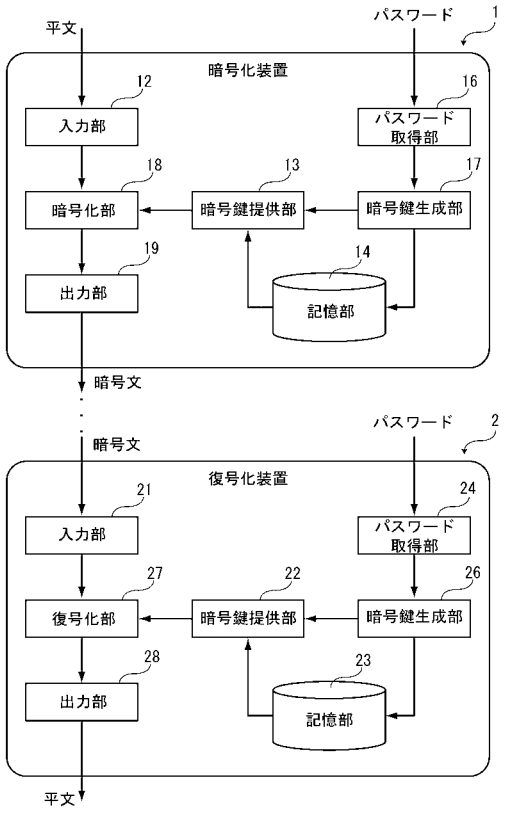
このグラフに示されるように、概ね１４５０Ｍｂｉｔ／ｓｅｃの処理速度が確保され、実用に耐え得る良好な結果となった。

【符号の説明】

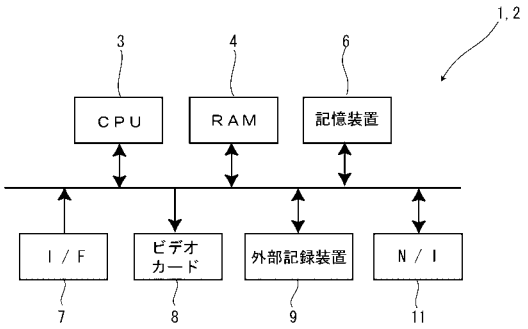
【００８８】

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

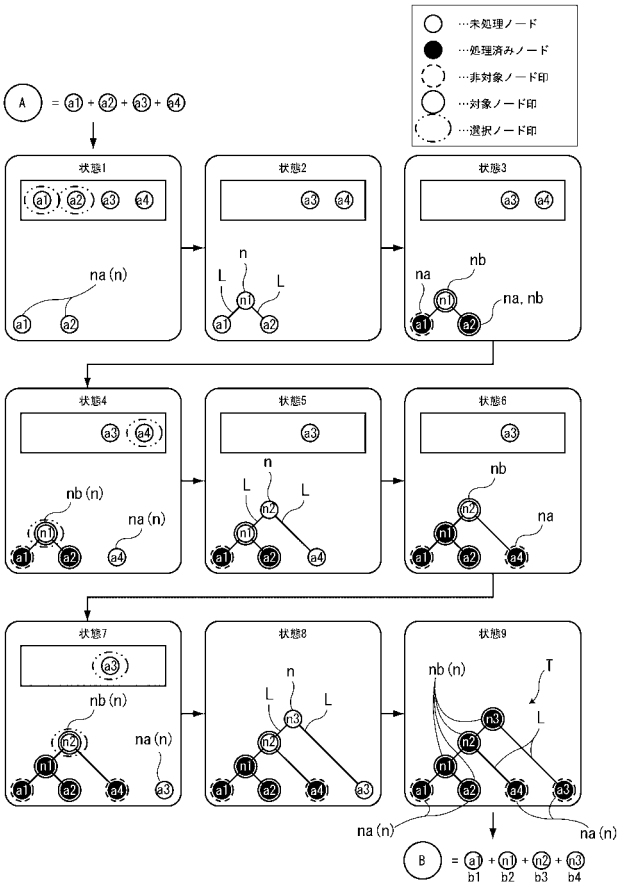
【図１】



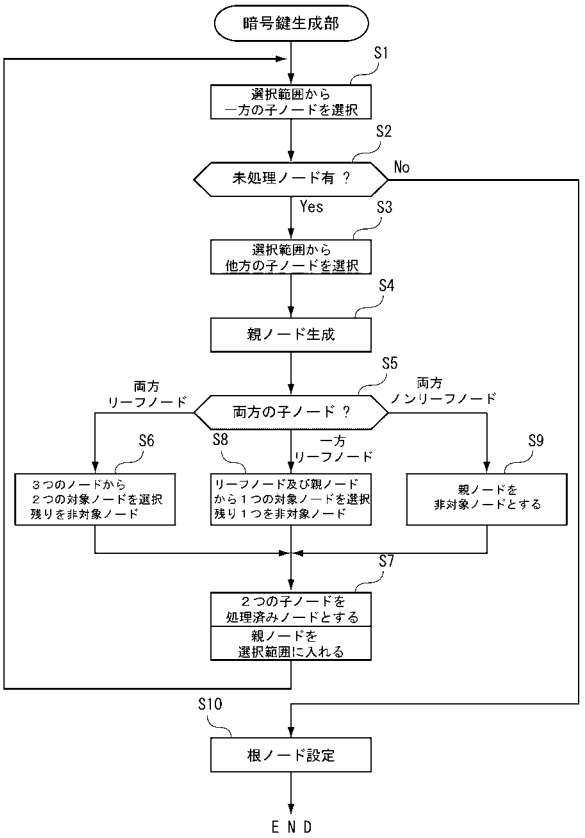
【図２】



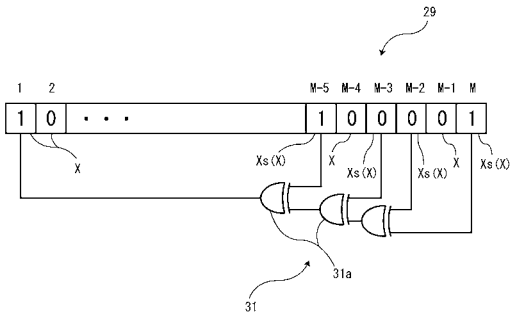
【図３】


…未処理ノード
…処理済みノード
…非対象ノード印
…対象ノード印
…選択ノード印
状態1
状態2
状態3
状態4
状態5
状態6
状態7
状態8
状態9
na (n)
nb (n)
nb
na
na, nb
n
T
L
Xs (X)
X
29
31
31a

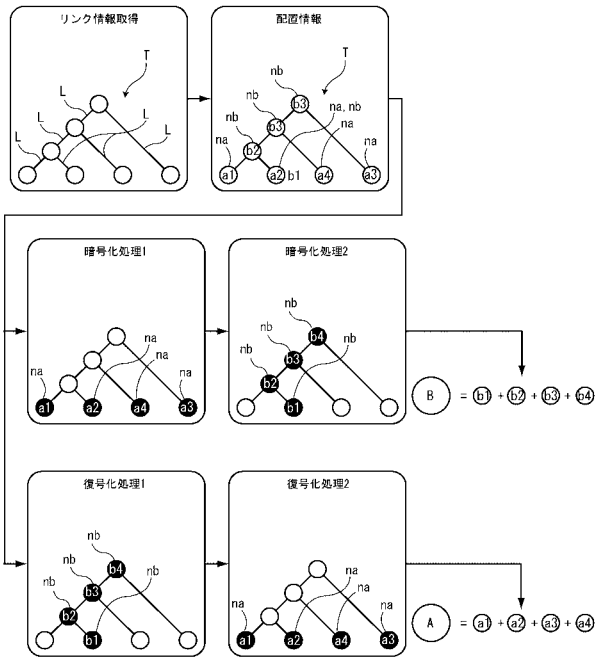
【図４】


暗号鍵生成部
S1
選択範囲から
一方の子ノードを選択
S2
未処理ノード有 ？
No
Yes
S3
選択範囲から
他方の子ノードを選択
S4
親ノード生成
S5
両方
リーフノード
両方の子ノード ？
両方
ノンリーフノード
S6
３つのノードから
２つの対象ノードを選択
残りを非対象ノード
S8
一方
リーフノード
リーフノード及び親ノード
から１つの対象ノードを選択
残り１つを非対象ノード
S9
親ノードを
非対象ノードとする
S7
２つの子ノードを
処理済みノードとする
親ノードを
選択範囲に入れる
S10
根ノード設定
E N D

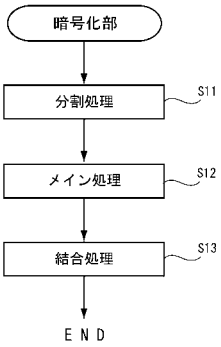
【図５】


29
1
2
M-5
M-4
M-3
M-2
M-1
M
1
0
0
0
0
1
X
Xs (X)
31
31a

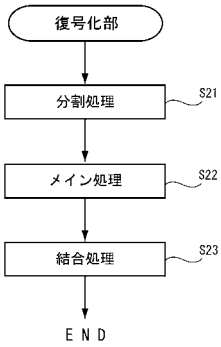
【図６】


リンク情報取得
配置情報
暗号化処理1
暗号化処理2
復号化処理1
復号化処理2
T
L
nb
na
na, nb

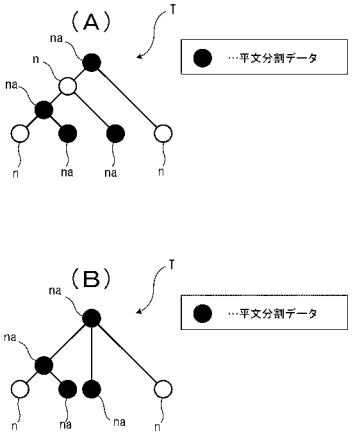
【図７】



【図８】



【図９】



【図１０】

(A)

分割数	5	6	7	8	9	10
A 評価総数	157	48,394	1,022,977	118,542,255	13,361,656,785	26,561,058,487
A 時間 (sec)	0.000	0.000	0.078	10.655	1358.435	2,971.041
B 評価総数	1,805	41,499	2,677,833	161,333,816	5,850,758,026	911,613,071,077
B 時間 (sec)	0.000	0.000	0.250	15.679	661.613	112,687.470
C 評価回数	1,112	10,810	777,693	178,836,563	9,110,674,918	1,124,547,911,542
C 時間 (sec)	0.000	0.000	0.078	18.689	1038.369	137,390.270
D 評価回数	2,593	44,749	2,595,645	125,977,537	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	417.303	58583.688
平均評価回数	1,588	39,286	1,895,177	133,030,533	7,593,776,050	630,493,850,507
平均時間 (sec)	0.000	0.000	0.172	13.182	832.208	77,577.511
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 評価総数	9,710	456,393	18,497	67,256,489	1,155,934	N/A
A 時間 (sec)	0.016	0.421	0.016	86.705	1.654	N/A
B 評価総数	2,218	10,359,946	271,927,308	2,919,338,317	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	2,153,389,569
C 時間 (sec)	0.031	1.997	0.281	51.028	510.448	3,585.760
D 評価回数	148	136,078	32,265	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	37	679	33,927	367,335	447,801,835	12,244,665
E 時間 (sec)	0.000	0.000	0.047	0.437	583.406	19.796
平均評価回数	5,083	2,623,205	54,454,628	605,533,591	196,351,436	728,651,958
平均時間 (sec)	0.009	2.434	65.427	841.182	299.189	1,212.897

(C)

分割数	8	12	16	20	21	22
A 評価総数	3,379	1,333,331	227,079,304	61,387,189,559	262,425,795,930	807,225,342,710
A 時間 (sec)	0.000	0.047	6.755	2,072.484	9,381.870	27,827.656
B 評価回数	5,138	1,331,647	202,278,727	52,667,127,315	239,953,370,817	978,106,546,287
B 時間 (sec)	0.000	0.062	7.784	2,297.213	10,795.141	44,920.066
C 評価回数	1,710	1,125,747	190,697,923	68,509,105,832	112,297,321,961	336,890,890,615
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	662,748	153,145,433	67,783,576,635	44,216,255,031	1,023,612,452,521
D 時間 (sec)	0.000	0.031	6.443	3,169.520	1,792.989	48,079.121
E 評価回数	6,145	1,140,381	300,394,213	55,285,701,374	205,230,493,078	866,246,336,861
E 時間 (sec)	0.000	0.047	13.307	2,398.270	9,185.655	39,390.038
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	269,128,937,220	1,052,049,481,860

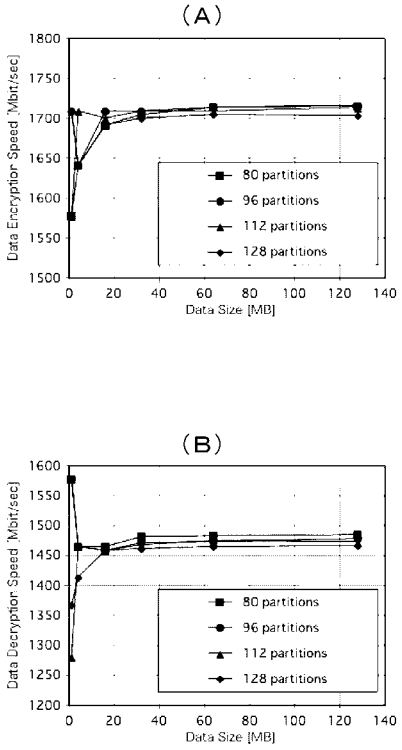
(D)

分割数	8	12	16	20	21	22
A 評価回数	8,809	8,809	10,801,735	10,802,470	10,801,735	1,343,815
A 時間 (sec)	0.000	0.000	8.393	7.472	8.377	1.030
B 評価回数	13	164	39,223,460	2,162,852	1,208,650,613	1,207,984,282
B 時間 (sec)	0.000	0.000	31.716	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	2,286,945,386
D 時間 (sec)	0.000	0.016	0.000	281.331	2,232.567	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	40,056.346	1,202.715	42,737.929
平均評価回数	1,998	6,188	10,005,490	1,072,155,529	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	11,864.445

(E)

分割数	5	6	7	8	9	10
A 評価回数	1,889	8,638	1,510,129	125,624,257	9,667,858,084	569,077,892,305
A 時間 (sec)	0.000	0.000	0.156	12.527	1,054.562	60,897.597
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	0.390	12.074	336.461	64,810.392
C 評価回数	1,576	36,106	1,351,664	75,463,004	13,805,582,589	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	1,852	33,968	795,637	59,610,970	4,595,050,916	280,997,006,425
D 時間 (sec)	0.000	0.016	0.078	5.834	544.784	36,113.923
E 評価回数	2,628	36,142	1,066,166	143,990,535	13,175,101,533	754,645,977,019
E 時間 (sec)	0.000	0.016	0.094	14.118	1,436.853	88,864.854
平均評価回数	1,546	30,688	1,308,901	85,138,123	7,341,246,870	353,686,642,945
平均時間 (sec)	0.000	0.005	0.143	8.894	835.790	48,018.239
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】


(A)
Data Encryption Speed [Mbit/sec]
1800
1750
1700
1650
1600
1550
1500
0
20
40
60
80
100
120
140
Data Size [MB]
80 partitions
96 partitions
112 partitions
128 partitions
(B)
Data Decryption Speed [Mbit/sec]
1600
1550
1500
1450
1400
1350
1300
1250
1200
0
20
40
60
80
100
120
140
Data Size [MB]
80 partitions
96 partitions
112 partitions
128 partitions